Image Segmentation Techniques Based On SVM

Yu-Yao Huang[1] , *Jian-Jiun Ding*[2]
Department of Electrical Engineering, National Taiwan University
E-mail address: b05901174@ntu.edu.tw[1] , jjding@ntu.edu.tw[2]

ABSTRACT

Image segmentation is the foundation of many image processing tasks such as object detection, image recognition and image compression. Nowadays, most advanced segmentation methods are super-pixel-based. Meanwhile, although many methods based on machine learning (ML) has been developed. However, a large proportion of ML-based image segmentation algorithms are pixel-wise. For the irregular shape and size, it is hard to apply ML-based algorithms on super-pixels. In this paper, we propose a novel method which combines the traditional super-pixel algorithms and the support vector machine (SVM), and achieves state of the art in BSDS300 dataset in GCE measurement.

Index Terms – Image segmentation, super-pixel, support vector machine

1. INTORDUCTION

Image segmentation is a classic task in the field of image processing. Many applications rely on qualify segmentation results. Traditionally, people used to cluster pixels of the image to get super-pixels. For example, the mean shift [1] and the watershed algorithm [2] are the classic super-pixel algorithms.

Recently, machine learning is widely-used and achieves great success in many fields, like natural language processing and speech signal processing. Computer vision is no exception. Some researchers aimed to apply machine learning algorithms on image segmentation. However, it is difficult to apply ML-based algorithm on super-pixels due to irregular shapes. It is a pity because we may improve the performance via combining the merit of traditional super-pixel methods and ML-based approaches.

If this paper, we proposed a new framework and successfully applied the SVM on super-pixels. Finally, the experiments on BSDS300 dataset showed the effectiveness of our proposed framework.

2. RELATED WORK

Super-pixel algorithm is a common pre-processing in many image segmentation algorithms. This is because most super-pixel algorithms focus on grouping pixels with similar relatively low-level features, like color and texture. As a result, the image will be over-segmented by super-pixel algorithm. In this section, a brief introduction of some widely-used super-pixel algorithm and super-pixel-based segmentation algorithms will be presented.

2.1. Mean Shift

The main idea of the mean shift algorithm [1] assumes that the density of data consists of several kernel distribution. Therefore, the mean shift is a simple iterative procedure that shifts every data to the mean of data in certain area. It is shown that mean shift is a mode-seeking process on a surface constructed with a “shadow” kernel. For Gaussian kernels, mean shift is a gradient mapping. Convergence is studied for mean shift iterations. Cluster analysis is treated as a deterministic problem of finding a fixed point of mean shift that characterizes the data.

For the application of the mean shift, an iterative mode mode-seeking procedure for locating local maxima of a density function, is applied to find modes in the color or

intensity feature space of an image. Pixels that converge to the same mode define the super-pixels.

2.2. Watershed Algorithm

The final purpose of the watershed algorithm [2] is to find the watershed line in an image. The watershed performs a gradient ascent starting from local minima to produce watershed line which separates catchment basins. The resulting super-pixels of the watershed algorithm are often irregular in size and shape and do not exhibit good boundary adherence.

The watershed transform is one of the classic methods for image segmentation. The basic idea behind watershed segmentation is to consider the regions to be extracted as catchment basins in topology. The watershed lines are the boundaries of catchment basins. First, the watershed algorithm initializes the points by finding the minimum intensity pixels and then the water will progressively fill up all the different catchment basins. If two pixels is merged by the raising water level, these two pixels will be classified to a super-pixel. At some point the water basin will start to merge with water from neighboring regions. This merging can be prevented by constructing dams at high altitude.

The segmentation produced by a naive application of the watershed algorithm is often inadequate: the image is usually over-segmented into a large number of regions because each potential minima could become a region in an image, in other words, the number of regions roughly equals to the number of minima in the image. As a result, several post-processing steps have been proposed to produce results that match human perception.

2.3. Multiscale Normalized Cut (MNCut)

Cour et al. propsed a multiscale spectral image segmentation algorithm named as MNCut [4], to segment large images. In contrast to most multiscale image processing, this algorithm works on multiple scales of the image in parallel, without iteration, to capture details from coarse to fine. The algorithm is computationally efficient, allowing to segment large images. It uses the Normalized Cut graph partitioning framework to construct a graph encoding pairwise pixel affinity, and partition the graph for image segmentation. They demonstrate that large image graphs can be compressed into multiple scales capturing image structure at increasingly large neighborhood. They also show that the decomposition of the image segmentation graph into different scales can be determined by ecological statistics on the image grouping cues. Images that could not be processed due to their size in the past, have been accurately segmented via this method.

3. ADOPTED TECHINQUE

3.1. Mean Shift Super-pixel

The super-pixels we adopt are generated by the mean shift method [1]. It equips with the properties for good boundaries maintenance and its segmented result lies between over-segmentation and over-merging.

3.2. Saliency Detection

Saliency detection is to distinguish the parts which human usually focus on from other insignificant parts in the image. It generates a saliency map which represents the importance of each pixel. The regions with larger saliency values are more likely to be the human interested regions.

In our framework, we adopt two kinds of saliency detection proposed by Zhu [6] and by Kim [7], respectively. Kim *et al*. proposed a saliency detection that utilize high-dimensional color transform (HDCT) to combine global and local salient region detection by random forest algorithm. Instead of focusing on one single color space, the HDCT applies multiple color space representation like RGB, CIELAB and HSV to solve the color ambiguities within regions. We can derive the difference of saliency values between two regions R_1 and R_2 from the computed

saliency map.

$$dSV(R_1, R_2) = |SV(R_1) - SV(R_2)| \quad (1)$$

where $SV(R_1)$ represents the mean saliency value of region R_1.

3.3. Edge Detection

In our method, we adopt the fast edge detection algorithm proposed by Dollar [8] and apply the structured edge (SE) detection to generate an edge map. This edge map is performed with binarization and forms the binary contour map. we define the *Contour-Rate* as:

$$ContourRate(i,j) = \frac{\#\ of\ pixels\ of\ long\ contours\ on\ Bnd\ (i,j)}{\#\ of\ pixels\ of\ Bnd(i,j)} \quad (2)$$

where *Bnd(i, j)* is the boundary between two adjacent super-pixels *i* and *j*. We further define edge strength to compensate the lack of long contours by:

$$ES(i,j) = \frac{\sum_{p \in Bnd(i,j)} edge\ value\ of\ p}{\#\ of\ pixels\ of\ Bnd(i,j)} \quad (3)$$

where edge value of *p* denotes the edge response value calculated by structured edge detector of pixel *p* on the boundary between super-pixel *i* and *j*.

3.3. Lab Color Histogram

We uniformly quantize each color channel into 32 levels and then the histogram of each super-pixel is calculated in the feature space of 32×32×32 bins. We define the similarity measure $\rho(R, Q)$ between two super-pixels *R* and *Q* as

$$\rho(R,Q) = \sum_{u=1}^{u} \sqrt{Hist_R^u * Hist_Q^u} \quad (4)$$

where $Hist_R$ is the normalized histogram of super-pixel *R*, *U* is the number of bins of quantization level, and *u* means the u^{th} element in normalized histogram. Therefore, the coefficient ρ is defined as the cosine of the unit vectors:

$$(\sqrt{Hist_R^1} \ldots \sqrt{Hist_R^U})^T \ and \ (\sqrt{Hist_Q^1} \ldots \sqrt{Hist_Q^U})^T \quad (5)$$

3.4. CIEDE2000 Color Difference

We choose the CIELAB color space and the CIEDE2000 color differences [9] to match human perception and sensitivity. Given two super-pixels *R1* and *R2*, we calculate the mean Lab values *MR1 = (L1, a1, b1)* and *MR2 = (L2, a2, b2)* and determine their difference:

$$de00(R_1, R_2) = \triangle E_{00}(M_{R1}, M_{R2}) \quad (6)$$

where $\triangle E_{00}$ is color difference from CIEDE2000

3.5. Texture Features

We choose the Log-Gabor filter [10] to extract the texture features of each super-pixel. The Log-Gabor filter has a null DC component and can be constructed with arbitrary bandwidth and the bandwidth can be optimized to produce a filter with minimal spatial extent. In this paper, we utilize the Log-Gabor filter with 2 scales and 4 orientations to produce texture maps. Then, the difference of textures between two adjacent super-pixel *i* and *j* is determined from:

$$dTex(i,j) = \sqrt{\sum_{k=1}^{8} (T_k(i) - T_k(j))^2} \quad (7)$$

where *k* denotes the k^{th} combination of scale and orientation.

3.6. Texture Information Enforcement

We apply the 2D-DCT to each super-pixel in grayscale image to extract more texture information. Since the range of 2D-DCT is within a rectangular, we must adjust the shape of super-pixels for computation to cope with the irregular shape of super-pixels. Therefore, for each super-pixel, we pad the minimum bounding box with mean intensity value of super-pixel.

The stronger response will occur in middle frequency band with more repeating pattern within regions. Hence, we define the texture strength by measuring the middle band response after we filter the DC term.

$$stTex(R) = \frac{\#\ of\ pixels \in midband}{\#\ of\ pixels\ of\ X_{p,q}} \quad (8)$$

where the *midband* is area on $X_{p,q}$ with DC term filtered. We further capture the texture information using gradient information by measuring the histogram of gradient within each super-pixel. Then, we compute the texture from

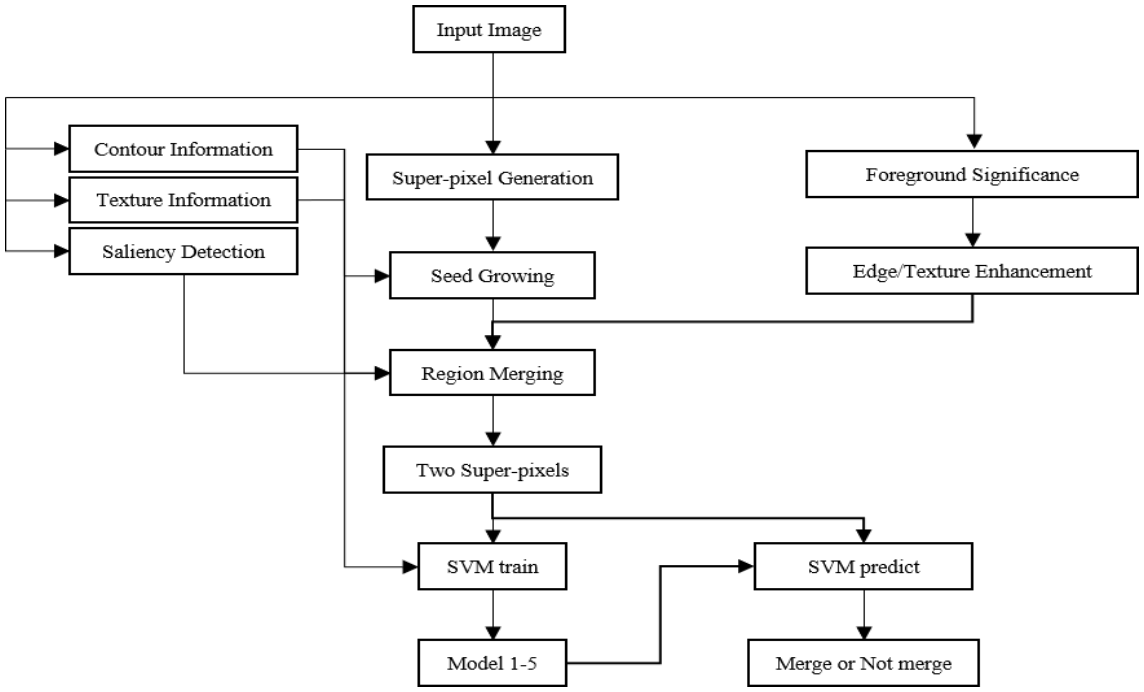

Fig. 1. The block diagram of the proposed method

gradient histogram by:

$$hisTex(R) = \frac{\#\ of\ pixels \in histogramLevel(6\sim30)}{\#\ of\ pixels\ in\ R} \qquad (9)$$

3.7. Support Vector Machine (SVM)

Instead of using the support vector machine built-in MATLAB, we use LIBSVM proposed by Chih-Jen Lin [5]. The LIBSVM is a simple, easy-to-use and efficient software for SVM classification and regression. Besides, the LIBSVM can modify the parameter to fit the model we need. We can set the mode of SVM and the mode of kernel. In out algorithm, we set the function as C-SVM and apply the RBF function $e^{-r|u-v|^2}$.

4. PROPOSED METHOD

In this section, we cover the main concept of the proposed method. Our proposed algorithm can be divided into two parts: the pre-processing steps and the support vector machine. Based on various similarity measurement in Section 3, we use them to help the pre-processing steps to generate super-pixels and use them as the features of each super-pixel. We pair two adjacent super-pixels and use the features of these two super-pixels as the input of SVM trainer. On the other side, we set the label which determines whether these two super-pixels should be merged or not as output of SVM trainer. The block diagram of the proposed method is shown in Fig. 1.

At the beginning, we use "Advanced image segmentation techniques with and without pre-assigned region number" [3] as the pre-processing step in our algorithm. We use the mean shift to generate the initial super-pixels, and then under the constraint of intensity of edge with the color and texture information, the super-pixels go through a growing process. Following, we apply a foreground significance estimation to determine if the edge/texture enhancement should be used or not.

Second, after noted above, the image has been merged into less super-pixels. We get various dissimilarity information of each super-pixels like color, texture, edge, area and saliency. For each super-pixel, we pair it with one of its adjacent super-pixels and then we calculate its dissimilarity by the information mentioned above. We use

this difference as input and whether there two super-pixels should be merged or not as output to train the SVM model.

We train several models for different number of super-pixels in an image. Besides, we use contact rate and difference of saliency value as the constraint to prevent from over-merging. This procedure will continue until every super-pixel pair should not be merged.

4.1. Super-pixel Growing

With the similarity measure defined in Section 3.4, we use such constraint to determine whether to merge or not. If the super-pixel *R* and its adjacent super-pixel *Q* are the closest in the histogram measurement space to each other, they should be merged. On the other word, we can represent the neighbor of super-pixel *Q* as $\{S_i^Q\}_{(i=1,2,\ldots,r)}$. Obviously, *R* is within $\{S_i^Q\}_{(i=1,2,\ldots,q)}$ since they are adjacent, therefore, *Q* is also a member of $\{S_i^Q\}_{(i=1,2,\ldots,r)}$. Finally, the merging between super-pixel *R* and *Q* will happen if the following conditions are satisfied:

$$\rho(R,Q) = \max_{i=1,2,\ldots,r} \rho(R,S_i^R) \quad (10)$$

and

$$\rho(R,Q) = \max_{i=1,2,\ldots,q} \rho\left(Q,S_i^Q\right) \quad (11)$$

Although we define a strict constraint to merge two super-pixels, there are some situations that could cause over-merging. Therefore, we set up some rules to further prevent such situation from happening. That is, we use *dTex(R,Q)* and *ContourRate(R,Q)* as threshold to reinforce the constraint. After the initial merging of super-pixels, we consider the super-pixels with more than two merging times as the initial seeds and perform seed growing process. The process uses the distance defined as:

$$Dist(i,j) = a * dTex(i,j) + b * (1 - \rho(i,j)) \quad (12)$$

where *i* is the initial seeds, and j is its adjacent super-pixels. Therefore, the seed growing process will merge the seeds with its neighbor that is closest to it in the *Dist(i,j)* measurement. In this paper, we use (a,b) = (3,0.5) to perform the seed growing. However, we use *ContourRate(i,j)* as the additional limitation of the growing process.

4.2. Support Vector Machine Trainer

After super-pixel growing, the image has been classified into adequate super-pixels. Then, we pair each super-pixel with its all adjacent super-pixels. The information of the pair of super-pixels is the input and the output of the support vector machine.

According to the ground truth, we know whether these pairs should be merged or not. Therefore, we set the label 0 to represent the pair of super-pixels should not be merged and on the contrary, set the label 1 to represent the pair of super-pixels should be merged. These labels are the output of the support vector machine.

On the other side, we use the similarity measurement mentioned in Section 3 as the features of the pair of super-pixels including *Hist*, *dTex*, *de00*, *Edge*, *Contact_Rate* and *dSV*.

Instead of using a model from beginning to end, we design a five-stage merging procedure according to the number of super-pixels. These five stages is corresponding to five support vector machine and corresponding to five ranges of number of super-pixels: ~150, 150~100, 100~75, 75~40, 40~. Besides, different models are trained with different features. We constructed many experiments to find out the fittest features.

4.3. Support Vector Machine Tester

For the first four SVMs, instead of merging the super-pixels according to the label of prediction, we use the accuracy of the label as reference. Every time we merge super-pixels, we only choose the top ranks of the accuracy to merge. Besides, we set some threshold to preventing from merging incorrectly. For each stage, repeatedly predict and merge until no further super-pixel can be merged or the number of super-pixels is out of range.

For the last step, we directly use the prediction of the

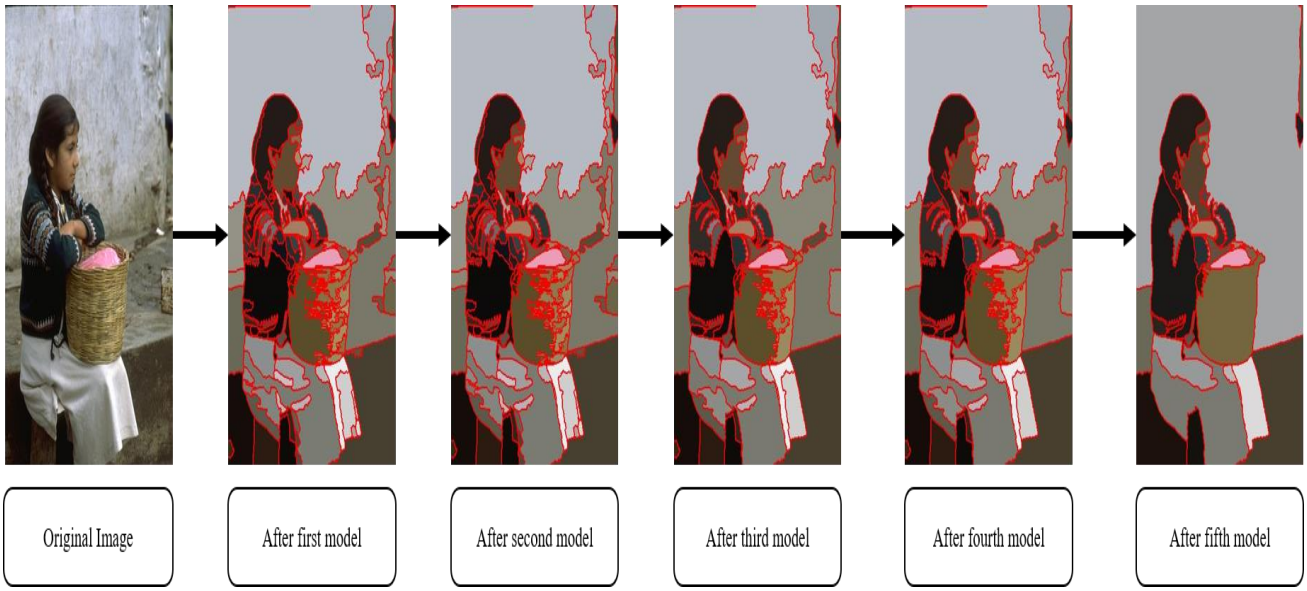

Fig. 2. The image after each step instead of the model

SVM. Different from first four models, we only predict one time. According to the prediction, the image merges the last time. Fig. 2. represents the image after each step.

5. EXPERIMENT

We perform our experiments on the Berkeley Segmentation Data Set 300 (BSDS300), which consist of 300 color images with at least 4 annotated ground truth images for each image. As for evaluation metrics in BSDS 300, we adopt Probabilistic Rand Index (PRI), Variation of Information (VoI), Global Consistency Error (GCE), and Boundary Displacement Error (BDE) to measure our segmentation results. Among all the evaluation metrics, the performance is better if PRI is larger while other three are smaller.

We compare our method with the other algorithms like Ncut [11], JSEG [12], NTP [13], MNcut , saliency driven total variation (SDTV) [14], TBES [15], UCM [16], MLSS [17], and SAS [18], global/local affinity graph for image segmentation (GL-graph) [19], and SCS [20]. Table 1 shows the performance of each segmentation algorithm. We can see that our proposed algorithm achieves state of the art in the measurement of CGE, and outperforms many methods in other three measurements.

Table 1. Performance of the proposed method compares to other methods over the BSDS300

Method	PRI	VoI	GCE	BDE
Ncut	0.7242	2.9061	0.2232	17.15
JSEG	0.7756	2.3217	0.1989	14.40
MNcut	0.7559	2.4701	0.1925	15.10
NTP	0.7521	2.4954	0.2373	16.30
SDTV	0.7758	1.8165	0.1768	16.24
TBES	0.80	1.76	N/A	N/A
UCM	0.81	1.68	N/A	N/A
MLSS	0.8146	1.845	0.1809	12.21
SAS	0.8319	1.6849	0.1779	11.29
GL-graph	0.8384	1.8012	0.1934	10.6633
SCS	0.8414	1.6573	0.1795	10.8783
Proposed	0.8239	1.8490	0.1638	10.6972

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel framework which applies the SVM on super-pixels. It is a great progress while most previous works can only apply ML methods on

regular pixels. we believe this is a worthwhile direction for more researchers to work on. Last but no least, the experiments on BSDS300 show the effectiveness of our proposed method.

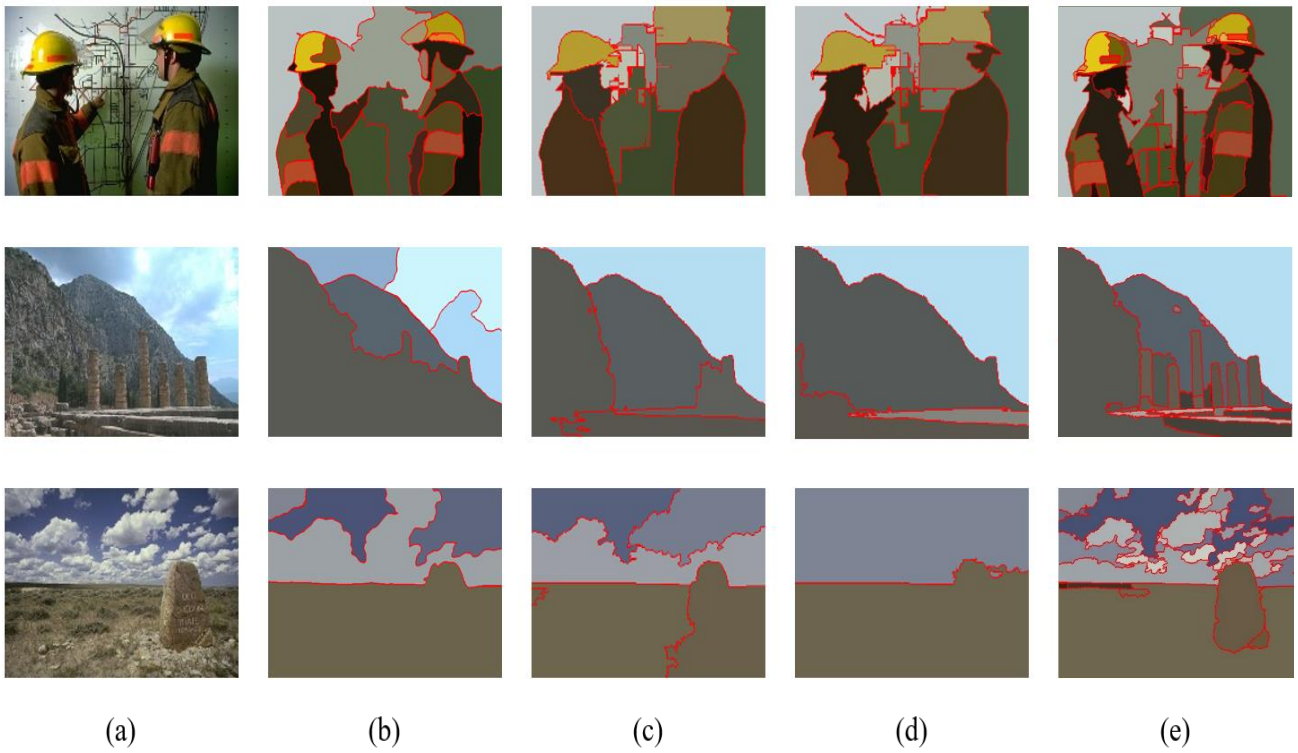

Fig. 3. Visual comparison on BSDS300 (a) Input images. (b) TBES. (c) MLSS (d) SAS. (e) Ours

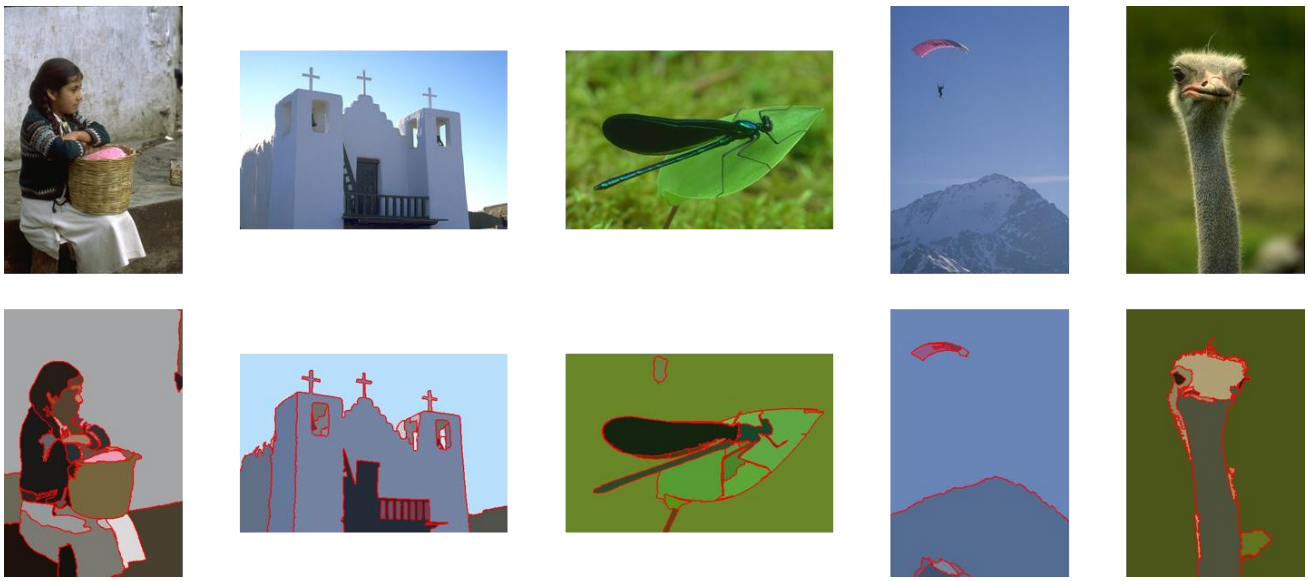

Fig. 4. BDSD300 dataset after our proposed segmentation

REFERENCES

[1] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, May 2002.

[2] L. Vincent and P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, no. 6, pp. 583-598, June 1991.

[3] T. C. Lin, Y. Y. Huang, I. F. Lu, H. Y. Ko, J. J. Ding, J. Y. Huang, aa P. H. Chen, "Advanced image segmentation techniques with and without pre-assigned region number", International Congress on Engineering and Information, Fukuoka, Japan, 2020.

[4] T. Cour, F. Benezit, J. Shi, "Spectral segmentation with multiscale graph decomposition," in CVPR, pp. 1124-1131, 2005.

[5] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[6] Zhu, W., Liang, S., Wei, Y., & Sun, J.(2014) Saliency optimization from robust background detection, in IEEE Conf. Computer Vision and Pattern Recognition, pp. 2814-2821.

[7] Kim, J., Han, D., Tai Y.W., & Kim, J.(2014) Salient region detection via high-dimensional color transform, in IEEE

Conf. Computer Vision and Pattern Recognition, pp. 883-890.

[8] Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: ICCV (2013)

[9] Sharma, G., Wu, W., & Dalal, E.(2005) The CIEDE2000 color-difference formula: Implementation notes, upplementary test data, and mathematical observations, Color Research & Application, 30(1), 21-30.

[10] Field, D.J.(1987) Relations between the statistics of natural images and the response properties of cortical cells, J. Opt. Soc. Am. A, 4(12), 2379-2394.

[11] Shi, J., & Malik, J.(2000) Normalized cuts and image segmentation, IEEE Trans. Pattern Analysis and Machine Intelligence, 22(8), 888-905.

[12] Deng, Y., & Manjunath, B.S.(2001) Unsupervised segmentation of color-texture regions in images and video, IEEE Trans. Pattern Analysis and Machine Intelligence, 23(8), pp. 800-810.

[13] Wang, J., Jia, Y., Hua, X.S., Zhang, C., & Quan, L.(2008) Normalized tree partitioning for image segmentation," in IEEE Conf. Computer Vision and Pattern Recognition, pp. 1-8.

[14] Donoser, M., Urschler, M., Hirzer M., & Bischof, H.(2009) Saliency driven total variation segmentation, in IEEE Int. Conf. Computer Vision, pp. 817-824.

[15] Rao, S. R., Mobahi, H., Yang, A.Y., Sastry, S.S., & Ma, Y.(2009) Natural image segmentation with adaptive texture and boundary encoding, in ACCV, pp. 135-146.

[16] Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J.(2011) Contour detection and hierarchical image segmentation, IEEE Trans. Pattern Analysis and Machine Intelligence, 33(5), pp. 898-916.

[17] Kim, T., & Lee, K.(2010) Learning full pairwise affinities for spectral segmentation, in IEEE Conf. Computer Vision and Pattern Recognition, pp. 2101-2108

[18] Li, Z., Wu, X.M., & Chang, S.F.(2012) Segmentation using superpixels: A bipartite graph partitioning approach, in IEEE Conf. Computer Vision and Pattern Recognition, pp. 789-796.

[19] Wang, X., Tang, Y., Masnou, S., & Chen, L.(2015) A global/local affinity graph for image segmentation, IEEE Trans. Image Processing, 24(4), pp. 1399-1411.

[20] Yang, Y., Wang, Y., & Xue, X.(2016) A novel spectral clustering method with superpixels for image segmentation, Optik-International Journal for Light and Electron Optics, 127(1), pp. 161-167