CpSc863 PAPER DEVELOPMENT

Project Document Cover Sheet

PROJECT REPORT

Project

Course	CpSc863	**Team ID**	Team 9
Project Title	Exploring Principles-of-Art Features For Image Emotion Recognition		
Start Date	January 2015	**End Date**	April 2015
Lead Institution	Clemson University		
Project Advisor	James Wang		
Project Team Member	Zian Chen Shengying Liu		

Table of Contents

Acknowledgements

This project was conducted under the guide of Dr. James Wang. During the study and implementation of this paper[1], we get a lot of help from Dr. Wang and we really appreciate his guidance and help.

The project would particularly like to acknowledge the help and support of the following:

- Dr. James(Zijun) Wang, Clemson University
- Dr. Allan Hanbury, Vienna University of Technology
- Jana Machajdik, Vienna University of Technology

Executive Summary

Paper research and reference study

The primary aim of the paper was to use principle-of-art emotion feature extraction to get feature vector, and then use these feature vector as training data of libsvm, after libsvm get training model, we use another part of data set as test data and let libsvm get test result for us. In this stage we investigate all the feature vectors this paper need. Since this paper just give us brief introduction of these vectors without many details of how we should calculate these vectors, so we read a lot of reference paper the author provided in order to know the exact algorithms of each feature vector.

Extract feature vector based on reference paper's approach

In this stage, we follow reference paper's approach to write our feature vector extraction code on Matlab. Some of the test images are too large to calculate, so we did resize on these images so that they didn't have so much feature points to be calculate.

Training and testing data use Libsvm

With help of Dr. Allan Hanbury, we captured some train and test image from http://sosphotos.deviantart.com and then extracted feature vector based on that. For classification, we got 84 images for training set and 24 images for testing set.

Test result and emotion prediction analysis

If we set label on our test image feature vector, we will get accuracy percentage of how many labels of image we got it right in classification. Besides that, Libsvm also gave its own prediction on each test image we provided, by using these prediction label, we can see if the model libsvm generated based on our algorithm is reasonable or not.

Background

With the increasing demand of image understanding by the public, the elements-of-art- based low-level visual analysis was not enough anymore. On the other hand, digital photography technology has been improved a lot through these years, which let high level image analysis based on image emotion becomes feasible.

The elements-of-art, such as shape, texture, color, line, etc. are used for most of the previous works, which are all low level visual features. However they cannot maintain constant in situation such as different arrangements. What's more, their have weak connection to emotions, but as we know, element arrangements are vary among meanings and arouse different emotions. At this point of view, we could not investigate image emotions based on one particular element. The proper way should be carefully arrange these elements and organize them into meaningful regions and images to describe specific emotions and semantics. The guidelines or rules of organizing or arranging associated elements are known as the principles-of-art, which consider various artistic aspects including proportion, movement, balance, harmony, emphasis, and variety.

According to above analysis and related art theory and computer vision research, we decide to study, formulate, and implement the principles-of-art systematically. We combine each principle quantization values together to build image emotion features. Unlike low-level features mentioned before, Principles-of-art-based emotion features (PAEF)will arrange and organize different elements and then used to do image emotional classification and prediction.

Methodology

Overall Approach

Basically our approach can be divided into three parts, which is shown in figure 1 below,

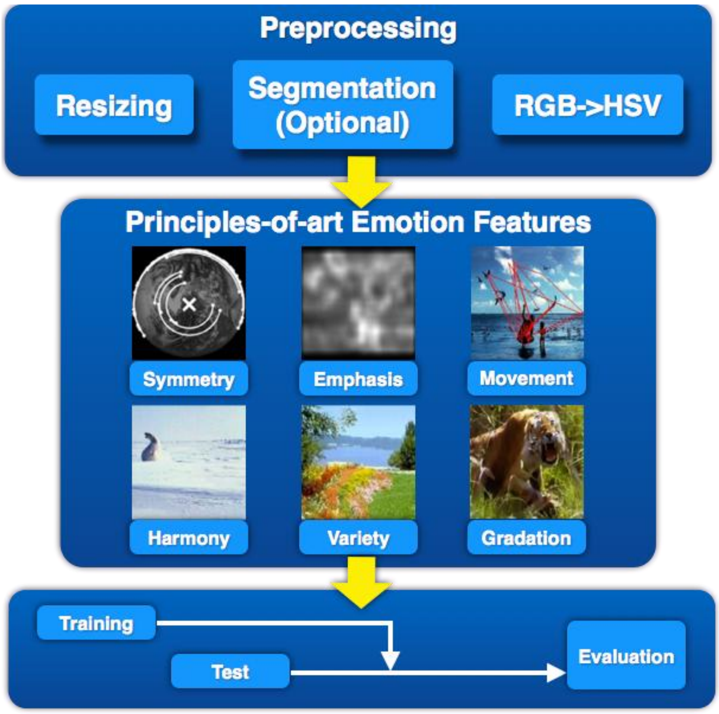


Figure 1 approach for PAEF emotion recognition

1)Firstly we extract one thousand photos from classification image dataset. Classify them manually and treat these classification results as frame of reference. Then we extract six hundred photos from these one thousand photos, do resizing and segmentation on these photos, cut them into segments, and then, for each segment, transfer them from RGB to HSV.

principles	measurement	#	Short description
Balance	Bilateral symmetry	12	Symmetry number, Max radius, angle, strength
	Rotational symmetry	12	Symmetry number, Max center, strength
	Radial symmetry	36	Distribution of symmetry map after radial transformation
Emphasis	Item color contrast	15	Avg contrast of saturation, contrast of light and dark, extension, complements, hue, warm and cold, simultaneous contrast
	RFA	3	Rate of focused attention based on saliency map and subject mask
Harmony	Range ability of hue and gradient direction	2	First and second max of local max hues and gradient direction in relative histograms of an image patch, and their difference; the combination of all patches of an image

Variety	Color names	12	Color types of black, blue, brown, gray, green, orange, pink, purple, red, white, yellow and their amount
	Distribution of gradient	48	The distribution of gradient on eight scales of direction and eight scales of length
Gradation	Absolute and relative variation	9	Pixel-wise windowed total variation, windowed inherent variation in x and y direction respectively, and relative total variation
Movement	Gaze scan path	16	The distribution of gaze vector

Table 1 Summary of the measurements for principles of art

2)Secondly, we apply formula of six principles and calculate every principles of art using measurement in below table 1. After all the calculation, we will get feature vectors,then we combine the representation of the six principles into one feature vector consistently. The dimensions of these principles are 60, 18, 2, 60, 9 and 16 respectively.

3)After doing this, we use feature vectors we generated as input of pattern recognition tools named LIBSVM to let it do training step. Since LIBSVM will build the classification principles according to the feature vectors we provides automatically, what kind of feature vectors we calculated will determine what kind of classification principles LIBSVM will build.

4) Then, after using six hundreds of photos as training material, we use other four hundreds to do the test on LIBSVM's classification principles. And see whether these testing photos will be classified correctly by LIBSVM. As we know, there should always have deviations between our approach and the manually classification results, LIBSVM will generatethe deviations between these two solution and we can calculate MSE (standard deviation) upon it.

Implementation

As we can see from figure 1, the most timing consuming part for the whole project is feature vector extraction. So we introduce this part as detailed as we can.

1. Balance

The balance features are divided into 3 parts:

Bilateral symmetry

1. Generated a set of mirrored feature descriptors mi. Here mi describes a mirrored version of the image patch associated with feature ki. The choice of mirroring axis is arbitrary owing to the orientation normalization in the generation of the descriptor. mi is generated by directly modifying this feature descriptor,this can be achieved simply by reordering the elements of the descriptor vector so they represent the original image patch flipped about the axis aligned with the dominant orientation.

2. Sort matches and generate match pairs (pi,pj) that are potentially symmetry features.

3. Calculate relative location, orientation and scale of pi and pj. Determine which contribute most to the symmetry and get all the symmetry axises.

4. Reject the irrelative points and get the final results.

There are 4 features in total:

1) Symmetry number

2) Maximum symmetry radius

3) Symmetry angle

4) Symmetry strength

Figure 2 are some sample results of bilateral symmetry.

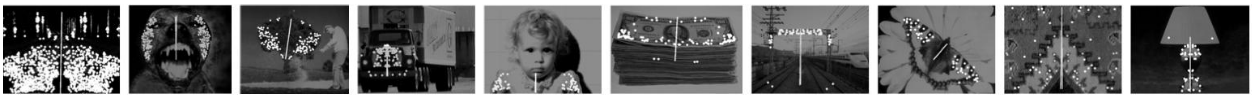


Figure 2 Bilateral Symmetry

Rotation symmetry

Unlike bilateral symmetry detection, detecting rotational symmetry does not require the manufacture of additional feature descriptors, and is detected by simply matching the features ki against each other. Each match defines a pair of point vectors (pi,pj).

There are 3 features in total :

1.Maximum symmetry center.x

2.Maximum symmetry center.y

3.Maximum symmetry strength

Figure 3 are some sample results of bilateral symmetry.

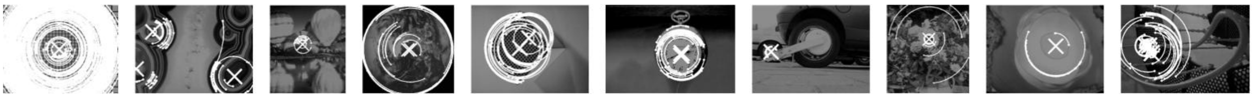


Figure 3 Rotation Symmetry

3.Radio Symmetry

We use the code on Gareth Loy website to calculate the radio symmetry,we get 1 feature here :

Radio symmetry strength

2. Emphasis

As we already know, emphasis, also known as contrast, is used to stress difference of certain elements. In this paper, the author adopt Itten’s color contrasts[2] and Sun’s rate of focused attention(RFA)[3] to measure the principle of emphasis. Basically emphasis calculation can be divided into three parts. Calculate saliency map of the image, calculate threshold mask of this image, and use formula (1) to calculate RFA.
When calculate saliency map, we did some improvement on this step. We used Graph-Based Visual Saliency(gbvs) Map[4] instead of author’ s ltten’ s color contrasts.
A new bottom-up visual saliency model, Graph-Based Visual Saliency (GBVS), is proposed. It consists of two steps: first forming activation maps on certain feature channels, and then normalizing them in a way which highlights conspicuity and admits combination with other maps. The model is simple, and biologically plausible insofar as it is naturally parallelized. This model powerfully predicts human fixations on 749 variations of 108 natural images, achieving 98% of the ROC area of a human-based control, whereas the classical algorithms of Itti & Koch ([5],[6], [7]) achieve only 84%.
In our algorithm, we used six steps to get our gvbs map from original image:

Step 1: compute raw feature maps from images
Step 2: compute activation maps from feature maps
Step 3: normalize activation maps.
Step 4: average across maps within feature channel.
Step 5: sum across feature channels
Step 6: blur for better results.

Figure 4 shows us a sample example of gbvs map we get.

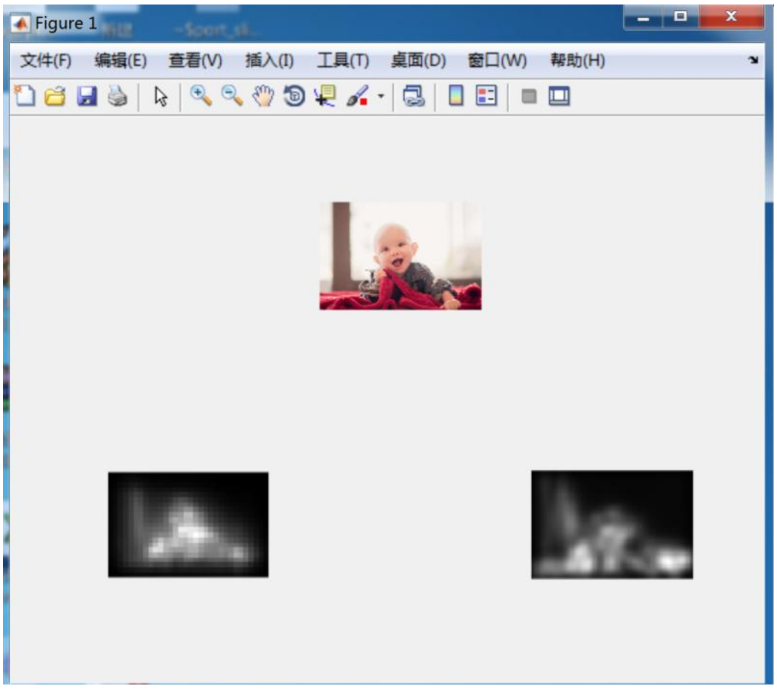


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)

Figure 4 sample result of gbvs map

The top center image is the original one , the bottom left corner is the result using our gbvs algorithm, and the bottom right corner is the result using ltten’ s algorithm. Obviously we can draw a conclusion that gbvs gets more accuracy ROI result compared with ltten’ s result.
After we get saliency map, we use code below to get threshold mask of the image, of course we need to transfer the original image from RGB to HSV, and in this case we only need to use V dimension to calculate mask.

```
% RGB->HSV
HSV = rgb2hsv(img);
H = HSV(:, :, 1);
S = HSV(:, :, 2);
V = HSV(:, :, 3);

double_V = double(V);

% calculate threshold and mask
threshold = graythresh(double_V);
Mask = im2bw(double_V,threshold);
```

The last step is using formula(1) in the paper to calculate RFA. The formula is like below

$$RFA(i) = \frac{\sum_{x=1}^{Wid}\sum_{y=1}^{Hei} Saliency(x,y)Mask_i(x,y)}{\sum_{x=1}^{Wid}\sum_{y=1}^{Hei} Saliency(x,y)}$$

Based on this formula, we calculate RFA using below code

```
SM_multiple = saliency_gbvs.master_map_resized.*Mask;
SM_sum = sum(SM_multiple(:));
S_sum = sum(saliency_gbvs.master_map_resized(:));
RFA = SM_sum/S_sum;
```

3. Variety

Each color has a special meaning and is used in certain ways by artists. We count how many basic colors (black, blue, brown, green, gray, orange, pink, purple, red, white, and yellow) are present and the pixel amount of each color.

Figure 5&6 represent a sample result of variety features.

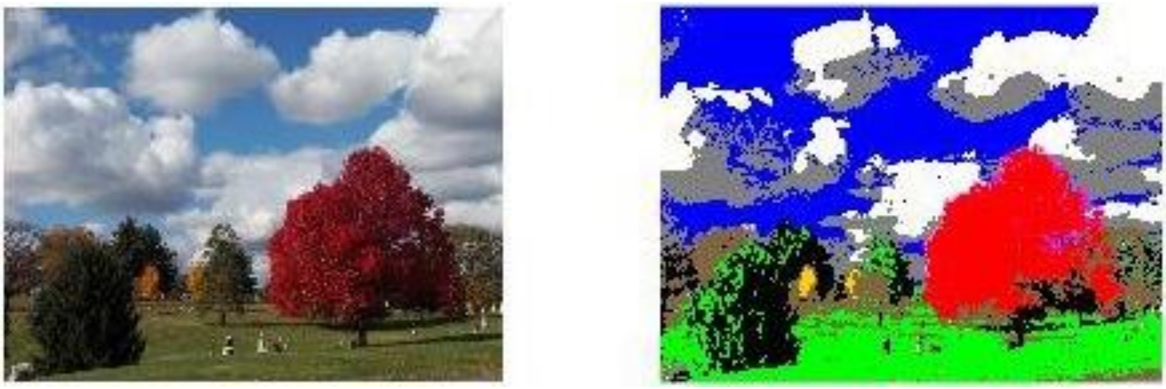


Figure 5. Calculate Color Weight

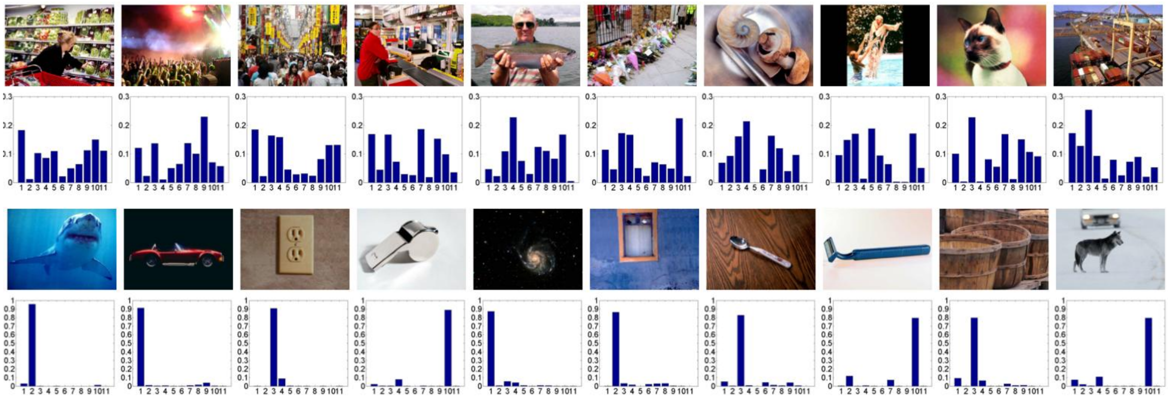


Figure 6. Color Weighting

4. Gradation

Here we use 3 steps to get the gradation features:

1. Get the gradient of the image in x and y directions

2. Calculate the windowed inherent variation and windowed total variation

3. Calculate the relative gradation and absolute gradation of the image

Actually we need to calculate 5 feature vectors using below four formulas:

$$D_x(p) = \sum_q g_{p,q} |(\partial_x I)_q|, \; D_y(p) = \sum_q g_{p,q} |(\partial_y I)_q|,$$

$$g_{p,q} = exp \left(-\frac{(x_p - x_q)^2 + (y_p - y_q)^2}{2\sigma^2} \right),$$

$$L_x(p) = |\sum_q g_{p,q} (\partial_x I)_q|, \; L_y(p) = |\sum_q g_{p,q} (\partial_y I)_q|.$$

$$RG = \sum_p RTV(p) = \sum_p \left(\frac{D_x(p)}{L_x(p) + \varepsilon} + \frac{D_y(p)}{L_y(p) + \varepsilon} \right),$$

$$AGT_x = \sum_p D_x(p), AGT_y = \sum_p D_y(p),$$

$$AGI_x = \sum_p L_x(p), AGI_y = \sum_p L_y(p).$$

We get 5 features here :

1.Pixel-wise windowed total variation

2.windowed inherent variation in x direction

3.windowed inherent variation in y direction

4.relative total variation in x direction

And figure 7 shows sample result we get for Gradation feature vectors:

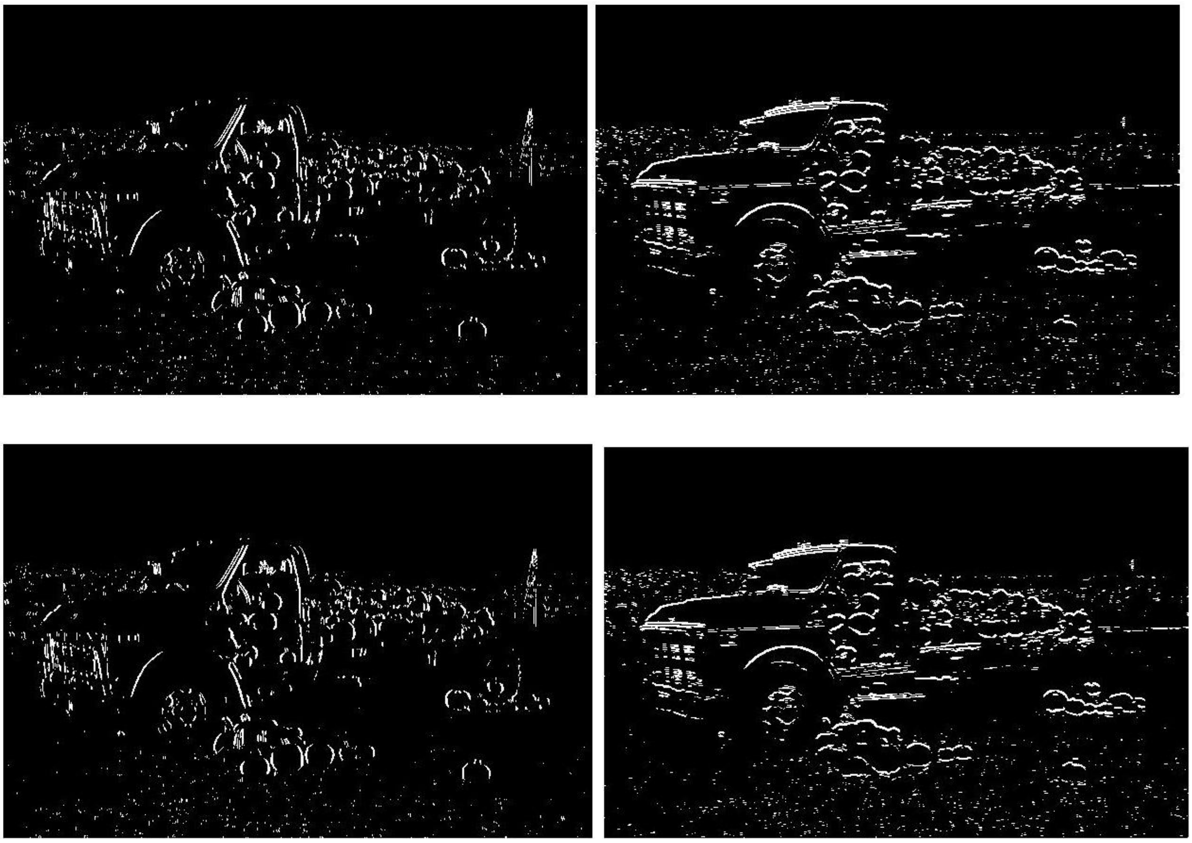


Figure 7. Gradation Features(Dx,Dy,Lx,Ly)

5. Movement

Movement is used to create the look and feel of action. Basically in this stage, we use gaze selection and eye scan path to indicate human’s eye moving action. Based on that we get our gaze vector from gaze selection. Visual data is represented as an ensemble of small image patches. Kurtosis maximization is adopted to search for the Super Gaussian Component (SGC). A response map is then obtained by filtering the original image with the found SGC. Based on the response map, we adopt a well known principle named winner-takes all (WTA) to select and locate the simulated fixation point. Gram-Schmidt orthogonal method is applied at the beginning of each selection to avoid convergence at the same location. Along with the saccadic simulation, a saliency map can also be estimated using either the selected fixations or the response maps. The proposed framework enables fast selection of a small number of fixations, which give processing priority to the most important components of the visual input. Different from low-level feature-based saliency driven approaches, the proposed gaze selection method is guided by high-level feature-independent statistical cues, which is supported by findings observed from real-world fixation analysis.
Below figure 8 shows up the overall procedure of the movement vector extraction.

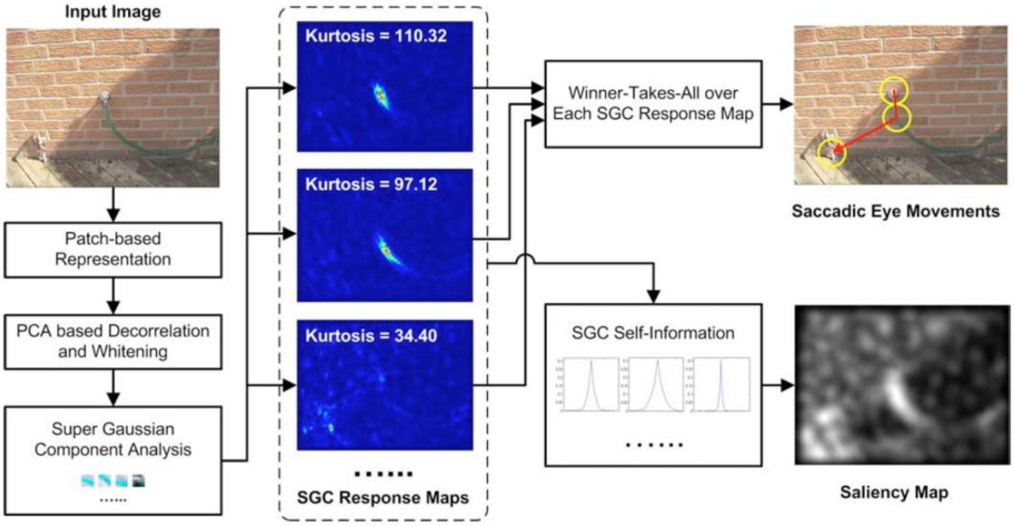
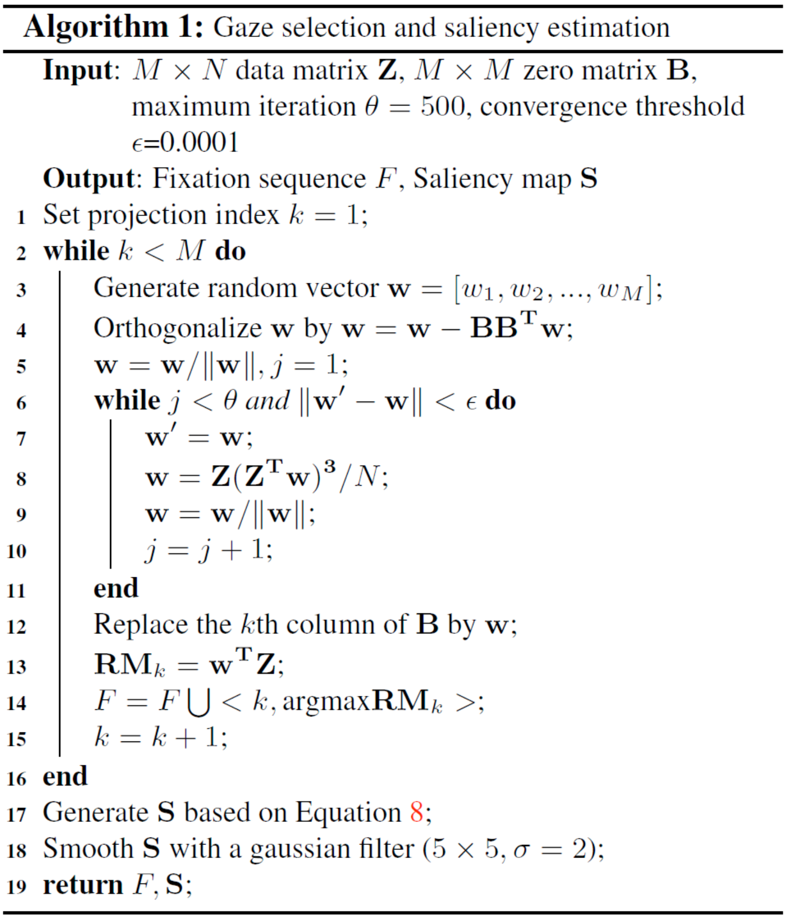


Figure 8. Movement Vector Extraction

And our code was mainly based on algorithm shown in figure 9



Algorithm 1: Gaze selection and saliency estimation

```
Input: $M \times N$ data matrix $\mathbf{Z}$, $M \times M$ zero matrix $\mathbf{B}$,
       maximum iteration $\theta = 500$, convergence threshold
       $\epsilon$=0.0001
Output: Fixation sequence $F$, Saliency map $\mathbf{S}$
Set projection index $k = 1$;
while $k < M$ do
    Generate random vector $\mathbf{w} = [w_1, w_2, ..., w_M]$;
    Orthogonalize $\mathbf{w}$ by $\mathbf{w} = \mathbf{w} - \mathbf{B}\mathbf{B}^{\mathrm{T}}\mathbf{w}$;
    $\mathbf{w} = \mathbf{w}/\|\mathbf{w}\|, j = 1$;
    while $j < \theta$ and $\|\mathbf{w}' - \mathbf{w}\| < \epsilon$ do
        $\mathbf{w}' = \mathbf{w}$;
        $\mathbf{w} = \mathbf{Z}(\mathbf{Z}^{\mathrm{T}}\mathbf{w})^3/N$;
        $\mathbf{w} = \mathbf{w}/\|\mathbf{w}\|$;
        $j = j + 1$;
    end
    Replace the $k$th column of $\mathbf{B}$ by $\mathbf{w}$;
    $\mathbf{RM}_k = \mathbf{w}^{\mathrm{T}}\mathbf{Z}$;
    $F = F \bigcup < k, \mathrm{argmax}\mathbf{RM}_k >$;
    $k = k + 1$;
end
Generate $\mathbf{S}$ based on Equation 8;
Smooth $\mathbf{S}$ with a gaussian filter $(5 \times 5, \sigma = 2)$;
return $F, \mathbf{S}$;
```

Figure 9. Gaze selection and saliency estimation

And figure 10 shows us one sample result of movement eye scan path on original image.

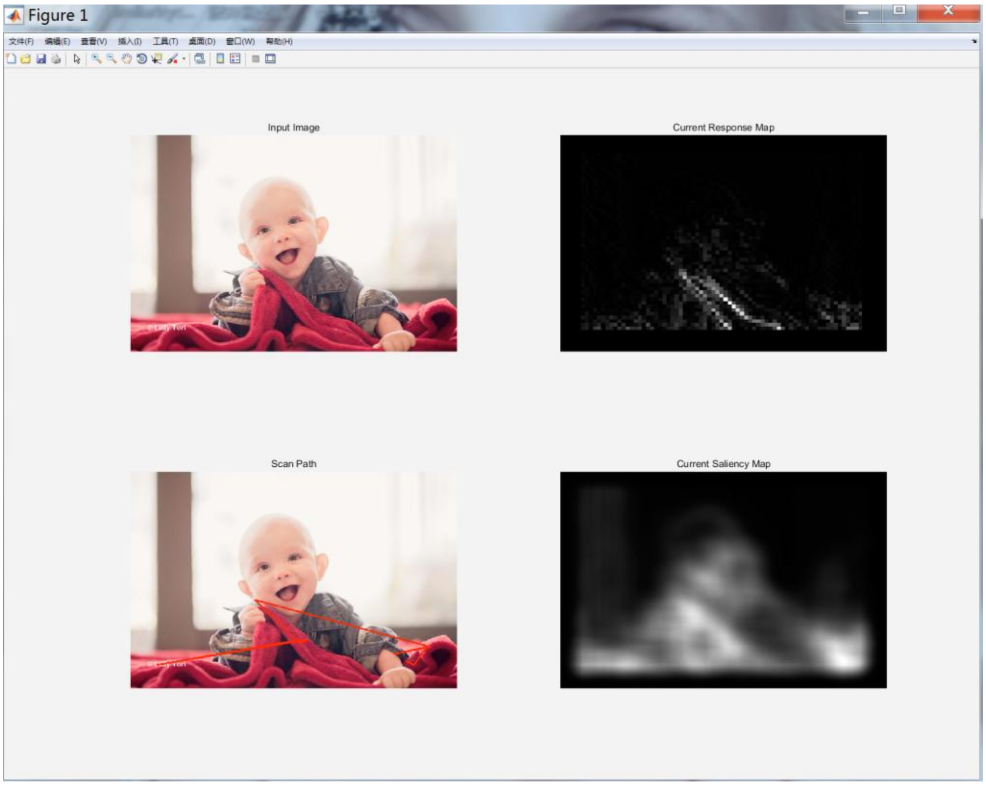


Figure 10. Gaze selection result

Outputs and Results

After extracting all the feature vectors, we get our training vector collection and testing vector collection in hand. Please note that when we use Libsvm for classification and regression, we give both training collection and testing collection with labels. The reason why we did it this way is because when we do the training, we need to let Libsvm ‘study’ what kind of vector represent sadness, contentment, fear, etc. And also, when we do the testing part, we need Libsvm to generate its own judger, which is the training model it generated after training process. And use it to ‘judge’ whether our label set on each vector row is correct or not. More importantly, we focus attention on its prediction came together with its accuracy quota. And we use our human intuition to see if its prediction is more reasonable than our label on testing images. If its prediction is more reasonable, than we can see the model generated based on our algorithm is good.

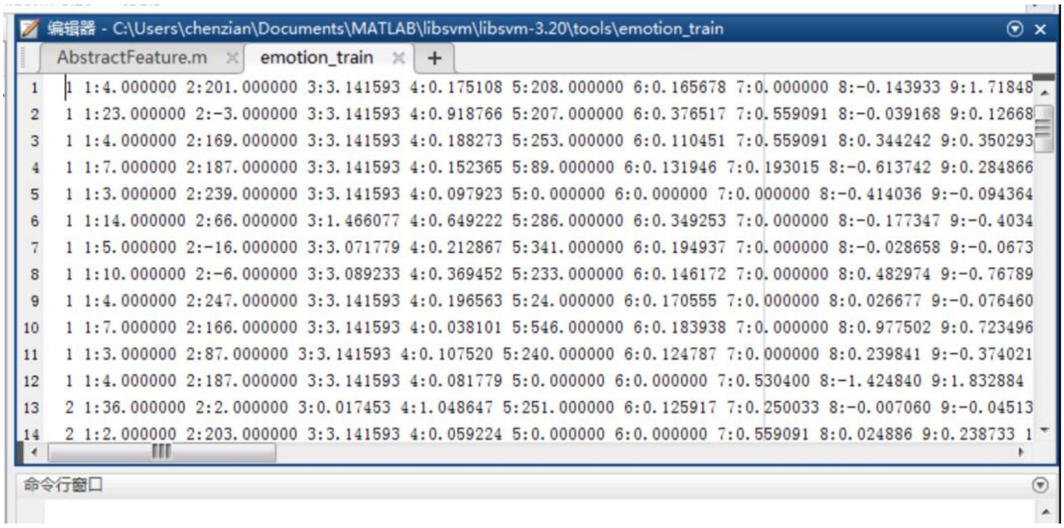
Classification

Classification is basically we train Libsvm to get the training model, and then use these training model to classify our test image set. In the evaluation of this classification part, we firstly get a very low accuracy, only 2.37%. The reason of these low accuracy result is because of these two main reasons,

- We didn’t set the best Libsvm parameters before we did our first classification. Since Libsvm is a support vector machine, the appropriate kernel function chosen, the correct SVM setting, and the degree of kernel function, the gamma parameter, all these things did influence the SVM model a lot. So How could we get these parameters all correct? Luckly, we have a python script named “easy.py” which can analysis the training dataset, and then set all these parameters correct for us by running a gnuplot process. **gnuplot** is a command-driven interactive function plotting program. It can be used to plot functions and data points in both two- and three-dimensional plots in many different formats. It is designed primarily for the visual display of scientific data.
- The second reason, is that we need to keep training set and testing set in the same style. For example, if we treat disgust images as something that will make the observer feel uncomfortable, then the test image should also use this kind of images. Or if we treat disgust images as something that shows disgust emotion expression within the image(like a face showing disgust expression), then the test images should also follow this rules. **The key point here is, no matter how you understand the meaning of the emotion, you should keep consistency between training images and testing images.**

After fixing these two issues, our accuracy increased from 2.37% to almost 81%! That is really a big progress. And we learned a lot from this process.

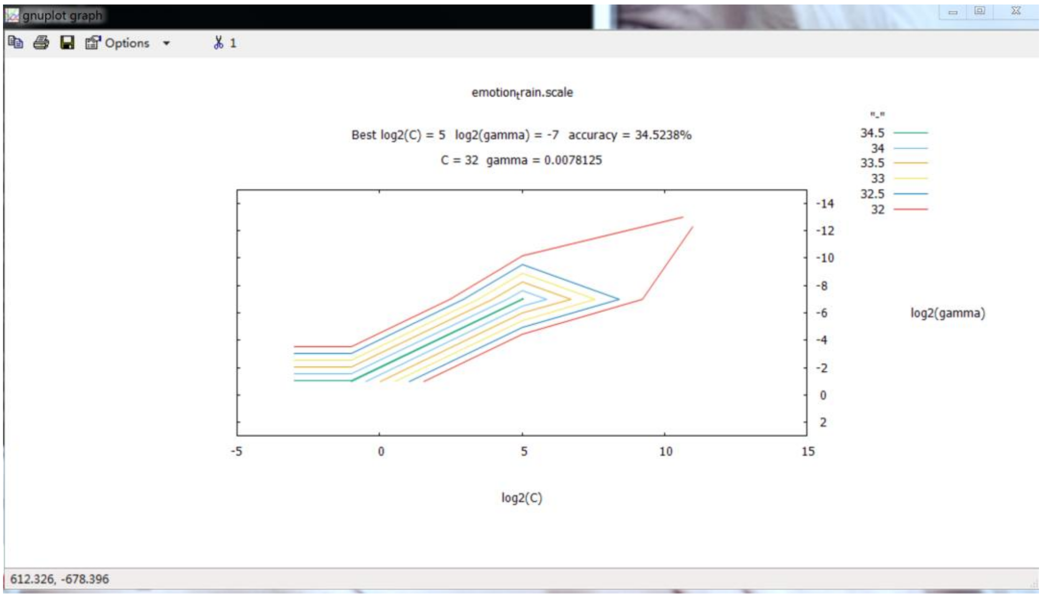
Figure 11 is the snapshot of our emotion_train vector collection for classification. The emotion_test is just the same format.


编辑器 - C:\Users\chenzian\Documents\MATLAB\libsvm\libsvm-3.20\tools\emotion_train
AbstractFeature.m
emotion_train
+
命令行窗口

```
1 1:4.000000 2:201.000000 3:3.141593 4:0.175108 5:208.000000 6:0.165678 7:0.000000 8:-0.143933 9:1.71848
1 1:23.000000 2:-3.000000 3:3.141593 4:0.918766 5:207.000000 6:0.376517 7:0.559091 8:-0.039168 9:0.12668
1 1:4.000000 2:169.000000 3:3.141593 4:0.188273 5:253.000000 6:0.110451 7:0.559091 8:0.344242 9:0.350293
1 1:7.000000 2:187.000000 3:3.141593 4:0.152365 5:89.000000 6:0.131946 7:0.193015 8:-0.613742 9:0.284866
1 1:3.000000 2:239.000000 3:3.141593 4:0.097923 5:0.000000 6:0.000000 7:0.000000 8:-0.414036 9:-0.094364
1 1:14.000000 2:66.000000 3:1.466077 4:0.649222 5:286.000000 6:0.349253 7:0.000000 8:-0.177347 9:-0.4034
1 1:5.000000 2:-16.000000 3:3.071779 4:0.212867 5:341.000000 6:0.194937 7:0.000000 8:-0.028658 9:-0.0673
1 1:10.000000 2:-6.000000 3:3.089233 4:0.369452 5:233.000000 6:0.146172 7:0.000000 8:0.482974 9:-0.76789
1 1:4.000000 2:247.000000 3:3.141593 4:0.196563 5:24.000000 6:0.170555 7:0.000000 8:0.026677 9:-0.076460
1 1:7.000000 2:166.000000 3:3.141593 4:0.038101 5:546.000000 6:0.183938 7:0.000000 8:0.977502 9:0.723496
1 1:3.000000 2:87.000000 3:3.141593 4:0.107520 5:240.000000 6:0.124787 7:0.000000 8:0.239841 9:-0.374021
1 1:4.000000 2:187.000000 3:3.141593 4:0.081779 5:0.000000 6:0.000000 7:0.530400 8:-1.424840 9:1.832884
2 1:36.000000 2:2.000000 3:0.017453 4:1.048647 5:251.000000 6:0.125917 7:0.250033 8:-0.007060 9:-0.04513
2 1:2.000000 2:203.000000 3:3.141593 4:0.059224 5:0.000000 6:0.000000 7:0.559091 8:0.024886 9:0.238733 1
```

Figure 11. Training sample for classification

Figure 12 show how the easy.py used gnuplot to get the correct parameter for LibSVM.



Best log2(C) = 5 log2(gamma) = -7 accuracy = 34.5238%

C = 32 gamma = 0.0078125

Figure 12. gnuplot for Libsvm parameter setting

Regression

SVR(Support Vector Regression) is another important experiment that we take in our method. Different from the SVM(Support Vector Machine),the SVR only output a single value that represents the predicted value of the testing vector. So, the SVR is used for predicting value rather than classification.

Each image have can be valued in two parameters -Valence and Arousal.

Arousal: The physiological and psychological state of being reactive to stimuli. It results in an observable change in the physical state of the body which causes you to become alert and a ready to move and respond.

Valence: This is the fluffy-cognitive-psychology one, and why I keep putting words like "good" in quotes. Valence is the "intrinsic attractiveness or aversiveness" of an emotion. Which is cognitive-psychologist speak for whether or not people would want to feel something. We put the feature vectors together with the valence and arousal values into SVR to training. Then, we put our testing vector and default valence and arousal into the SVR to prediction.

We take 3 steps to predict the testing value

1. Get the vectors of training samples.

We give 2 values(valence & arousal) to each image that we will use for training by manual, the distribution of the training samples is represented in the figure below. Then we calculate the feature vector by PAEF, we have the valence, arousal and feature vectors in this step.

2. Training & Testing

Considering the SVR can only output one value at the same time, we separate the expedient in 2 parts: predict valence and predict arousal. The difference between these two experiences is we combine the valence value with the feature vector in the valence prediction, and opposite in the arousal prediction. For example, we put the feature vectors with the valence value into the SVR getting the training model, then put the testing vectors which also combined with the valence value. At last, we get the predicted valence value of the testing vectors. After we get both of the valence and arousal values, we will put them together to calculate the accuracy of our method.

3. Evaluation

We combine the valence and arousal value together to evaluate the accuracy of our method.

The result were not as good as we have expected,

- The predicted valance values are aggregated in a small area(valence=2,valence=9)
- The accuracy is 20% for the valence and 0% for the arousal

The reasons why we got these kind of results are:

- The limitation of the training samples, we only take 60 images for training, and the variety of the training samples is pretty low.
- The valence and arousal values that generated manually were not accurate enough.
- The number of the testing samples is small, we only take 10 images for prediction.

So for the reason of data size limitation, our regression accuracy may not be very good, but since we get very good result on classification part, we can say that our algorithms are good for this image emotion recognition. If we could get much bigger dataset in the future, we have confidence we could get much better regression results.

Conclusions

By implement and test the approach of this paper, we have these conclusion below,

- Basically the approach author provided in this paper can get good results in image emotion recognition.
- By using gbvs instead of Ltten’s color of contrast approach in emphasis emotion vector generation, we get better result than Ltten’s. This is our improvement upon author’s original approach.
- Under condition of small image dataset, we can still get great result in image classification, but we couldn’t get reasonable result in regression part. The main reason is that, during regression process, we need to extract a regression function by large enough data points in A-V dimension space.
- If we want to get better results using LibSVM, we need to keep consistency between training images and testing images.

References

[1] Zhao, Sicheng, et al. "Exploring principles-of-art features for image emotion recognition." Proceedings of the ACM International Conference on Multimedia. ACM, 2014.

[2] J. Itten. The art of color: the subjective experience and objective rationale of color. Wiley, 1974.

[3] X. Sun, H. Yao, R. Ji, and S. Liu. Photo assessment based on computational visual attention model. In ACM MM, 2009.

[4] J. Harel, C. Koch, and P. Perona, "Graph-Based Visual Saliency", Proceedings of Neural Information Processing Systems (NIPS), 2006.

[5] L. Itti, C. Koch, & E. Niebur "A model of saliency based visual attention for rapid scene analysis", IEEE Transactions on Pattern Analysis and Machine 1998

[6] L. Itti & C. Koch "A saliency-based search mechanism for overt and covert shifts of visual attention", Vision Research, 2000

[7] L. Itti, & P. Baldi "Bayesian Surprise Attracts Human Attention", NIPS*2005