Photo aesthetics evaluation system: an application of CNN and SVM

Chen Qian
chenq@stanford.edu

Zhi Li
zhil@stanford.edu

***Abstract*—In this project, we applied two machine learning techniques: CNN (Convolutional Neural Network) and SVM (Support Vector machine) to build an image aesthetic evaluating system. And we have achieved an 5-folder cross validation accuracy of above 99% by using CNN implemented in Torch.**

In the 'Introduction' section, a brief background of the problem and an introduction of our system are given. In the 'Dataset and Features' section, we've discussed how we generate the training data and extract the input features for the machine learning algorithms. And in the next two sections: 'Methods' and 'Experimental Result', how we applied CNN and SVM and the performance of each algorithm have been discussed. The report ended up with a 'Conclusion' section including the summary of work and a list of future work.

Keywords—image classification, SVM, CNN, machine learning

I. Introduction (*Heading 1*)

Every day we are collecting lots of photos either taken by ourselves or from networks. Some of them are in 'good' quality in an aesthetic sense while some of them are not that 'good'. In this project we built a ranking system to automatically rank the given photos for users in a similar aesthetic view of users . We think it's useful for either classify photos w.r.t. photo quality or use the system to guide user to take good pictures in real time etc.

The input to our system is an image with specific resolution, then we use our learning algorithms to evaluate the quality of the input image as an integer ranging from 1~5. In this project, we've implemented two main type of algorithms: CNN (Convolutional Neural Network) and SVM (Support Vector Machine). And we've achieved an 5-folder cross validation accuracy of above 99% by using CNN!

II. Related Work

What we have done is actually a image rateing problem. Data set contain image and its aesthetics score and to predict some other images aesthetics score. We are referring some related work in academy. One important application is handwriting recolonization done by Yann LeCunn from NYU, Corinna Cortes from Google labs and Christopher J.C. Burges from Microsoft research. They used and compared several machine learning algorithms, like SVM, CNN, KNN and so on applied to handwriting digits recognition[1]. They have achieve 99% and above on 10 class classification accuracy by using CNN. The MNIST database has become a widely used database for training and test in the field of machine learning. There have been a number of scientific paperson attempts to achieve the lowest error rate; one paper, using a hierarchical system of convolutional neural networks, manages to get an error rate on the MNIST database of 0.23 percent.[2]The original creators of the database keep a list of some of the methods tested on it.[1]In their original paper, they use a support vector machine to get an error rate of 0.8 percent.[3]

CIFAR-10 and CIFAR-100 are another database that widely used in academy. They are 10 classes and 100 classes labeled images for objects recognition. It is done by Krizhevsky from University of Toronto[4][5].

III. Dataset and Features

In this section, we talk about our flow to get the dataset and extract the manual features used by our learning algorithms. Figure 1 on the right gives you an overview on how the flow works.

Basically, we used the AVA(a large-scale database for Aesthetic Visual Analysis) database in our project. It provides a list of image IDs under DPChallenge (with which we can construct the corresponding urls) and counts of aesthetics ratings in a range of 1~10. We then wrote a Crawler to crawl all the database (~255,000 images with their average rankings by online viewers).

After milestone, we did a better analysis on the data, and found the ranking of our previous data is not evenly distributed, this time we implemented a Sampler to pick the data uniformly distributed over the scaled ranking ranging from 1~5, so we how has 5 class data instead of 10. We did this because there's very little images range from 1~2 and 8~10, we remarked images among 1~2 as 3 and 8~10 as 7, and relabeled them as 1~5. Thus, we have 2000 images for each label class from the output of the Sampler.

These 10,000 images further went to 'Manual Feature Extractor' and 'Image Resize Engine'. The 'Manual Feature Extractor' extracts 11 features each from the original images, thus outputs a 10,000 * 11 feature matrix, these features will be

used by SVM algorithm only (please see Method Section for details). We'll discuss each feature later in this section. The 'Image Resize Engine' resizes the images into 8*8, 10*10, 16*16, 32*32, 64*64 and 128*128 versions of the 10,000 images, the 8,10,16,32 versions will be used by SVM and 32, 64, 128 versions will be used by CNN.

Both the feature matrix and the resized images went to the 'DataGen' engine to generates the input files for LIBSVM used as SVM solver, and Torch, the open framework for CNN.

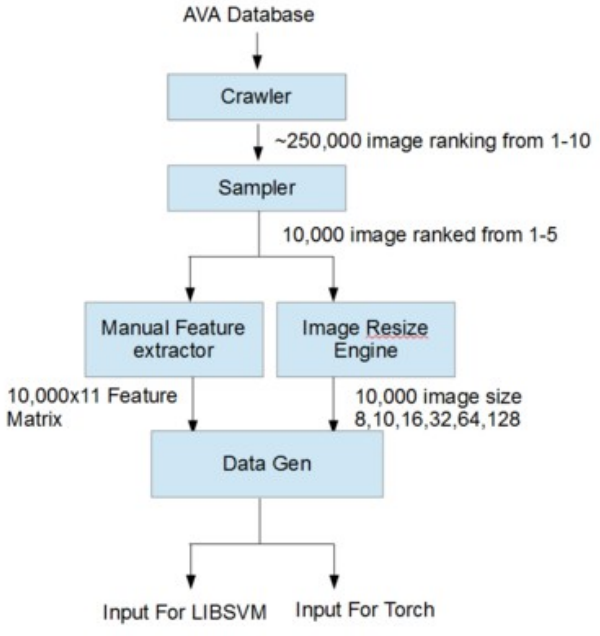

Fig. 1. Data generate flow

Now let's talk a little bit details on the manually extracted features. We think these features are typical to represent a particular image. The first 5 features are 1x1 metrics and the color centroid is a 6x1 vector feature, including one centroid (x,y) for R,G,B each. And here we assumes the image has MxN pixels.

1. Contrast: here we used the RMS(root mean square) contrast, calculated as follows:

$$I_{ij} = \frac{R_{ij} + G_{ij} + B_{ij}}{3}$$

$$\bar{I} = \frac{\sum_i \sum_j I_{ij}}{MN}$$

$$f\_contrast = \sqrt{\frac{1}{MN} \sum_i \sum_j (I_{ij} - \bar{I})^2}$$

2. Colorfulness: Hasler and Susstruck, 2003[6] suggested an algorithm to measure image colorfulness:

$$f\_color = \sqrt{Var(R-G)^2 + Var(|0.5(R+G)-B|)^2} + 0.3\sqrt{E[|R-G|]^2 + E[|0.5*(R+G)-B|]^2}$$

3. Sharpness: we used a simple way to measure image sharpness using its gray scaled image's gradient in Matlab:

$$[G_x, G_y] = gradient(gray - scale - image)$$

$$f\_sharp = \sqrt{\frac{1}{MN}(\sum_i \sum_j G_{xij}^2 + \sum_i \sum_j G_{yij}^2)}$$

4. Blurriness: Fred., Thier., 2007[7] suggested a flow to measure the blurriness of an image, and here's the flow chart:

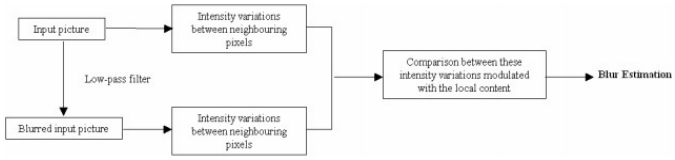

Fig. 2. Flow Chart for Measurement of Blurriness

5. Edge Detection: we define this feature as the percentage of edge pixels, and the edge pixels are detected by using 'Sobel' methods in Matlab:

$$f_{i\_edge} = \frac{\#of\_edge\_pixels}{MN}$$

6. RGB Centroids: we calculated the normalized centroid of R, G, B as our features, take R as an example:

$$(\overline{x_R}, \overline{y_R}) = (\frac{1}{M} \frac{\sum_i (i \cdot \sum_j R_{ij})}{\sum_i \sum_j R_{ij}}, \frac{1}{N} \frac{\sum_j (j \cdot \sum_i R_{ij})}{\sum_i \sum_j R_{ij}})$$

IV. Methods

A. Support Vector Machines

We used C-Support Vector Classification(C-SVC) implemented by LIBSVM for SVM. It solves the following primal problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

subject to:

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0, i = 1, \cdots, l$$

where

$$y \in R^l$$
$$y_i \in \{-1,1\}$$

where $\phi(x_i)$ maps x_i into a higher-dimensional space and $C > 0$ is the regularization parameter. Duo to the possible high dimensionality of the vector variable w, usually we solve the following dual problem.

$$\min_{w,b,\xi} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha$$

Subject to

$$y^T\alpha = 0$$
$$0 \le \alpha_i \le C, i = 1,\cdots,l$$

where $e = [1,....,1]^T$ is the vector of all ones, Q is an l by l positive semidefinte matrix,

$$Q \equiv y_i y_j K(x_i, x_j)$$
$$K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$$

is the kernal function.

$$w = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i)$$

After the dual problem is sovled, using the primal-dual relationship, the optimal w satisfies

and the decision function is

$$\text{sgn}(w^T\phi(x) + b) = \text{sgn}(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b)$$

We store label names, support vectors, and other information such as kernel parameters in the model for prediction.

Since in our problem, we are actually solving a 5-class classification problem, what it does is to use 'one against the rest' mechanism to treat one class as labeled '+1' and the rest are labeled as '-1', then we can apply the 2-class SVM above.

For the input features, we did two major experiments, for the first part, we feed the normalized image pixels as input features, e.g. for the 8*8 color image, it will has 8*8*3 = 192 features.

For the second part, we used the manual extracted features as inputs to compare the performance with the raw pixel methodology.

The experimental results will be discussed in the next section.

B. Convolutional Neural Network

In machine learning theory, convolutional neural network is a type of feed-forward artificial neural network where the individual neurons are tiled in such a way that they respond to overlapping regions in the visual field.Convolutional neural networks are inspired by biological process and they are right now, widely used in image recognition area.

1) Different types of layers

a) Convolutional layer will compute the raw pixel values of the image.

b) Sub-sampling layer will perform a fownsampling operation along the spatial dimensions, resulting in smaller dimensions data.

c) Fully connection layer will compute the class score, result in a class size vector(5x1 in this report).

2) Architecture design

We designed 3 convolutional neural network architecture to process 3 different kind of pixels image, 32x32, 64x64 and 128x128.

32x32 CNN architecture is shown as below.Input is 3 channel (RGB) of 32x32 image, it go through a convolutional layer which has 5x5 vertical and horizontal layer to produce 16x28x28 feature map. Next step is sub-sampling which takes max value of every 2x2 window to produce 6x14x14. And then go through one more convolutional layer and one more sub-sampling layer. Full connection layer is followed from the image data(16x5x5) to 5 classes which is the final output of this classification.

64x64x3 resolution image convolutional neural network shares same architecture except the full connection layer input is 16x13x13 due to larger image.

128x128x3 resolution image convolutional neural network use two more convolutional layers and subsampling layers, and the full connection layer input 64x4x4.

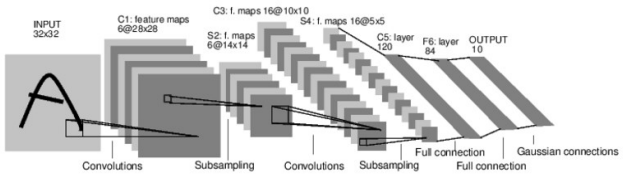

Fig. 3. Convolutional Neural Network for 32x32x3 32x32x3 image input.

V. Experiments, Results and Discussion

A. Support Vector Machines

As discussed in the previous section, for the 1st part of the experiment, we used normalized image pixels as input

features directly, and we tried 8*8, 10*10, 16*16, 32*32 dataset, we also tried 3 different kernels, the 5-folder cross validation accuracy rate are listed in the following table:

TABLE I. Test Accuracy of 5 Fold and 3 Resolution

	8x8	10x10	16x16	32x32
Linear	30.59%	37.47%	56.80%	20.34%
Polynomial	22.87%	22.77%	26.50%	17.50%
RBF	28.61%	30.98%	40.92%	19.64%

From the table, we see that the linear kernel performs best generally, the 16*16 image set gives the best performance. The result makes sense, since for low resolution images, the information within the original images for telling it's good or not is lost, think about in a extreme way, when every image becomes one pixel, it's impossible to tell whether it is good or not; and for high resolution images, then the features size will become too large, and overfitting will happen, e.g. for 32*32 images, we have 3072 features for each image, while we only have 2000 images for each label class.

Then for the second part of the experiment, we decided to use manual features extracted from the original data, and here's the result:

TABLE II. Manual Extraction Feature Result

	Manual Extraction
Linear	27.40%
Polynomial	22.87%
RBF	27.44%

From the second part we see that the accuracy lower than using raw pixels, we think this is because only 11 features are extracted and these 11 features are not guaranteed to measure photo aesthetic metrics although we think they are important features.

Overall, the best accuracy we've achieved by using SVM is 56.80%.

B. Convolutional Neural Network

We did 5 fold cross-validation on total 10000 image, they are K1, K2, K3, K4 and K5. And original image data set is extracted as 3 type of resolution, low(32x32), medium(64x64) and high(128x128).

The 5 fold experiment accuracy and run time is listed in the tables below:

TABLE III. Test Accuracy of 5 Fold and 3 Resolution

	K1	K2	K3	K4	K5
32x32	79.5%	79.85%	82.5%	79.05%	80.7%
64x64	99.7%	99.95%	99.5%	99.65%	99.2%
128x128	99.8%	99.85%	100%	99.9%	99.95%

TABLE IV. Run Time of 5 Fold and 3 Resolution

	K1	K2	K3	K4	K5
32x32	1590s	1597s	1650s	1581s	1614s
64x64	1868s	1896s	1882s	1868s	1943s
128x128	7453s	7573s	7453s	7603s	7616s

On average, CNN achieve 79.32% accuracy on 32x32 pixel input, 99.62% on 64x64 pixel input and 99.9% on 128x128 pixel input. And it takes 1606.4s to run 32x32 pixel image, 1891.4s to run 64x64 pixel image and 7539.6s to run 128x128 pixel image on Mac Pro (2.8GHz Intel Core i7 CPU) in CPU mode.

The CNN experiment shows that accuracy would increase when image resolution increase. If the image is more clear, it is easier to predict test case score.

Overall, the performance of SVM to this problem is not that good comparing with CNN algorithm.

VI. Conclusion and Future Work

A. Hybird of CNN and SVM

CNN is designed to mimic biological neural network and then applied in computer science. But in CNN, it did not consider (at least we haven't figured out) manually extracted features such as contrast, brightness, rule of thirds and so on while these features potentially contribute a lot to fthe aesthetic rank of photos. It may help improve evaluation of photos.

B. CPU vs GPU run time comparison

We run CPU mode only before milestone, so we are interested in how GPU perform and its comparison with CPU. Also the performance of parallel computing if neural network is complex. And we'll run our algorithm on the the mobile GPU board.

ACKNOWLEDGMENT

Thanks help from TA Albert Haque who provide quick and nice suggestion and explenation.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 86(11):2278-2324, November 1998.

[2] Cires¸an, Dan; Ueli Meier; Jürgen Schmidhuber (2012)."Multi-column deep neural network for image classification".*2012 IEEE Conference on Computer Vision and Pattern Recognition*

[3] LeCun, Yann; Corinna Cortes; Christopher J.C. Burges."MNIST handwriting digit database", Yann LeCun, Corina Cortes and Chris Burges. 17 August 2013.

[4] Krizhevsky, A., Sutskever, I. and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada

[5] Alex Krizhevsky "Learning Multiple layer of Feature from Tiny Images" 2009

[6] Hasler, David, and Sabine E. Suesstrunk. "Measuring colorfulness in natural images." *Electronic Imaging 2003*. International Society for Optics and Photonics, 2003.

[7] Crete, Frederique, et al. "The blur effect: perception and estimation with a new no-reference perceptual blur metric." *Electronic Imaging 2007*. International Society for Optics and Photonics, 2007.