Image Retrieval by the Face Detection and background

Kapre Ashwin Prashant
SCOPE
Vellore Institute of Technology
Vellore, India
ashwinkapre@gmail.com

Inamdar Chaitanya Ravindra
SCOPE
Vellore Institute of Technology
Vellore, India
susmiti63@gmail.com

***Abstract*—Now a day many people have a bunch of photos stored with them. In the most of the cases, these images are never organized properly. So, when you want a particular image of a person or group of people it's very hard to find it. You have to search the whole dataset manually which very tiring and boring work. In this paper, we suggest a way to find a relevant image based on user requirements. First, we label and classify the dataset to improve time complexity. We extract the feature vector that'll represent the user requirements such as how many faces, which faces etc. Then we'll compare this feature vector with the feature vectors of the dataset. Then retrieve the relevant image based on the difference between the feature vectors. The user has the option to give feedback to improve the using relevant feedback.**

Keywords—Content based image retrieval, Face recognition, Semantic segmentation, MultiLayer Perceptron, Deep learning

I. Introduction

Dataset will be the collection of the images any individual as his/her photos. We'll use such collection of our friends. The feature vector will be divided into two parts. First part will represent the facial part of the query image and second part will represent the surrounding texture of the image apart from the facial part. The facial part will largely contain the information about the face/s that is detected in the image. Texture part will contain information about the shirt color, background texture etc.

A feature vector will be generated that will represent the number of faces in the image. Using this feature vector database will be clustered into a group of images that contain the same number of faces. Now the group of images with no face will be discarded. This will narrow down the database. The feature vector is composed of two parts, these two parts will be separately compared. Facial part will be directly compared using Euclidean distance and equal weightage will be given to each face. But in the case of the other vector weighted values will be defined based on the amount of texture in the image. Larger the contain larger the weight value. A neural network will use these weight values to produce the output. Now, these values will be directly compared using Euclidean distance. Feature extraction of the texture of the image will include the semantic concept. The vector will be generated using semantic classes. Also, the vector will have weight values based on the amount of content that'll help to remove irrelevant images that are retrieved based small details.

After this comparison, we will give relevance feedback to the output to get better results for further comparison round. This will help to learn neural network to get accurate and precise results each time. This will basically improve feature vector and extraction. First, we will detect a number of faces in each image in the dataset. Then by clustering, they will get classified into various classes which will be of no. of faces {0,1,2,3,4,5,6+}. And we will remove a class with 0 faces in the image from the database. This all basically will be unsupervised learning. After this, we will apply various features like mean etc. The performance will be measured using Precision-Recall and F-measure.

II. related work

Since the development of the deep learning techniques, different face detection models with better accuracy were introduced. We are using one of such model introduced by the Davis King [1]. This model is based on the Deep Face Recognition [2] and [3]. The pretrained model used by this example program is in the public domain. Also, the model has an accuracy of 99.38% on the standard Labeled Faces in the Wild benchmark. This is comparable to other state-of-the-art models and means that, given two face images, it correctly predicts if the images are of the same person 99.38% of the time. We have used this model to check the similarity of the faces in the 2 different images.

The idea of retrieval of the images using the content of the image than just keywords was introduced in 90s [4] [5]. The old CBIR use the basic image processing tool to extract the feature vector of the image like color histogram [4]. The recent of models of CBIR used the similar features with more sophistication [6][7]. The recent models introduced the better and new ways of indexing, use of dimension reduction methods, feature extraction [6][7]. The recent models introduced the way to build the content-based retrieval system in practical world. The recent models also used new concepts that proved to be helpful in increasing the efficiency and accuracy of the models.

The relevance feedback for the content-based image retrieval system was introduced in the late 90s [8][9]. The relevance feedback improves the model by gathering the reviews of the models and changing the model based on the reviews of the model. The content-based image retrieval system can be improved if the feature vector generation can be changed based on the review of the existing feature vector generation.

The semantics for the content-based image retrieval system was introduced in the early 2000s [10]. The semantic gap is big issue in any retrieval system. Many efforts are made to reduce the semantic gap in general. Semantic can be used at any stage of the model based on usage and requirement of the model.

The semantic segmentation proved to be technique to reduce the semantic gap. The deeplab model [11] is currently giving results with maximum accuracy. This model is based on the theoretical model [12] which labels each pixel by using deep learning models.

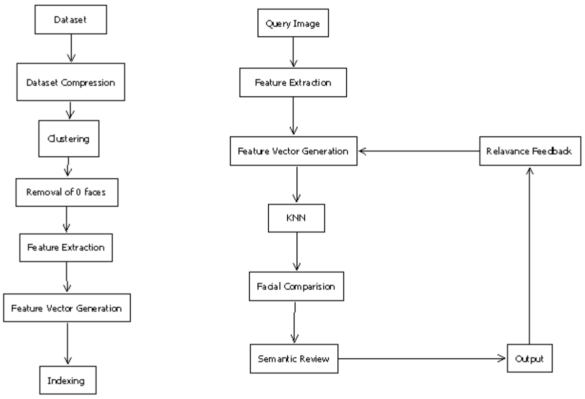
III. MOTIVATION

Multimedia database structure and also performing queries on that is very complicated problem. Generally, in our laptop we have lot of photos and if we have to find some particular photos. it's very time consuming and difficult to find. There are lot of techniques available which gives you required images but they don't take query image as an input to retrieve related images. So, we will make universal structure which can be used with different implementations.

As the result of our research we will make the system which fulfill the following requirements

- Ability to retrieve related images by taking image query as an input
- Ability to consider background also while retrieving related images.
- Potential to consider person's outfit also while retrieving related images
- It’s very easy to add new object for which there is no need of training or rebuilding the whole structure
- By semantic segmentation we will detect different objects in the image and we will store them and later we will use to compare them

IV. MODEL



A. Data pre-processing

The dataset we are using for this application is very raw and unprocessed. It will be collection of the photos taken by a user. The dataset will be totally raw without any labelling or classification. In order to get good results this dataset must processed before extracting the features from the images in the dataset. Here we propose 4 steps pre-processing that modify the dataset and convert it into better dataset consisting of clusters, low size.

- Data set compression

The first step to the data pre-processing is compression. Photo captured by the normal smartphone camera have size ranging from the 1 MB to 8 MB depending upon the quality of the camera. The captured photo contains noise and many other details that contribute to the cause of the increased size of the image.

We find the important image data that contributes to the faces and the background behind it. We use DCT [12] [13] [14] [15] [16] (Discrete Cosine Transform) to convert the image into oscillating cosine wave. The formed wave is examined and the noise and other unnecessary details are removed from the wave. The modified wave is then converted back into the image by IDCT (Inverse Discrete Cosine Transform).

- Clustering

The compressed data set is clustered using predefined clusters. These predefined clusters represent the number of faces present in the image. Using segmentation and edge detection techniques each image is analyzed. Distance of each cluster is found from each cluster and image is attached to the cluster with which it has least distance.

- Removal of 0 faces

Out of the all the cluster one cluster will be the cluster that will contain the no faces such images will be no match to any image and hence removed to reduce the computational and searching time.

B. Feature vector generation

- Facial feature extraction

The important aspect of this paper is to find the relevant images containing similar faces as that of the query image. In order to achieve this first face is detected using segmentation and boundary detection.

Once the face detected next part is to extract the facial features in order to define a face. Basic facial features are Eyes, Cheek, Chin, Forehead, Ears.

These all five features extracted using shape, color and texture feature. The color feature is extracted using color histogram, the shape feature is extracted using the Fourier transform, texture feature is extracted using mixture of GLCM and Gabor filter.

Usually in the face eye color plays more part than texture of the eye similarly shape forehead plays important part than the color of the forehead while texture of cheek plays important part than color. This means not each feature will contribute equally to the facial feature thus we create a fully trained MLP(Multi-Layer Perceptron) using basic dataset of faces that will create the facial feature using each individual features.

Background feature extraction

Even though facial features play important part in the background also contributes to relevance of the image. Background will help in retrieving even more precise and related results.

- Feature vector generation

The facial and background feature don't contribute equally to the feature vector of the image. Usually the ratio can be defined as 7:3. According to this assumption the MLP(MultiLayer Perceptron) is built to generate feature vector of the image. This perceptron can be trained using relevance feedback.

Indexing

The generated feature vector is indexed for easy and fast comparison with query image.

C. Similarity measures

- KNN

The query image will have 3 feature vector one representing the number of the faces, other two representing the facial feature and background.

In the KNN feature vector representing the number of the faces is used. In the pre-processing of the data we divided dataset into the clusters based on the number of the faces present in the image.

Now using cluster name as label of the image each image feature vector is compared with query image feature vector. Simple Euclidean distance is used to find the distance. Using proper number of K label of the query image is found out. It will the cluster the query image belongs to i.e. number of the faces query image contains.

The images in the query image cluster are further used to retrieve the relevant image since all other images irrelevant to the query image.

Facial comparison

Now on the reduced dataset of the images facial comparison is applied. It compares the feature vector generated by the MLP of the query with the feature vector of the dataset image.

The comparison will be made using modified Euclidean distance that will compare the feature vectors using higher power and different mechanism of the difference along each axis.

- Semantic review

The facial comparison will produce some results. These results will be analyzed to with semantic classes. Using this analysis semantic profile of the results will be generated. This semantic profile will be compared with the query image profile. The results with highly related with the query image will be moved up the order and vice versa.

D. Relevance feedback

In the final part of the model user will be presented with the relevance results. The user will be presented with limited set of the results if the user moves to the next set of results then former will be rated as poor. The activity of the user will be used to rate the image results on the scale of the 1 to 10. Using this rating the all the ANN present in the model are retrained to improve feature vector generation and improve the result accuracy.

V. Implementation

The specifications of the laptop and python used to implement the model.

Laptop Specifications:

HP Pavilion Notebook

- Processor: Intel(R) Core(TM) i5-5200U CPU @2.20GHz
- Installed RAM: 8.00 GB
- System Type: 64-bit Operating System, x64-based processor
- Operating System: Windows 10 Home Single Language
- Version: 1709
- OS Build: 16299.309

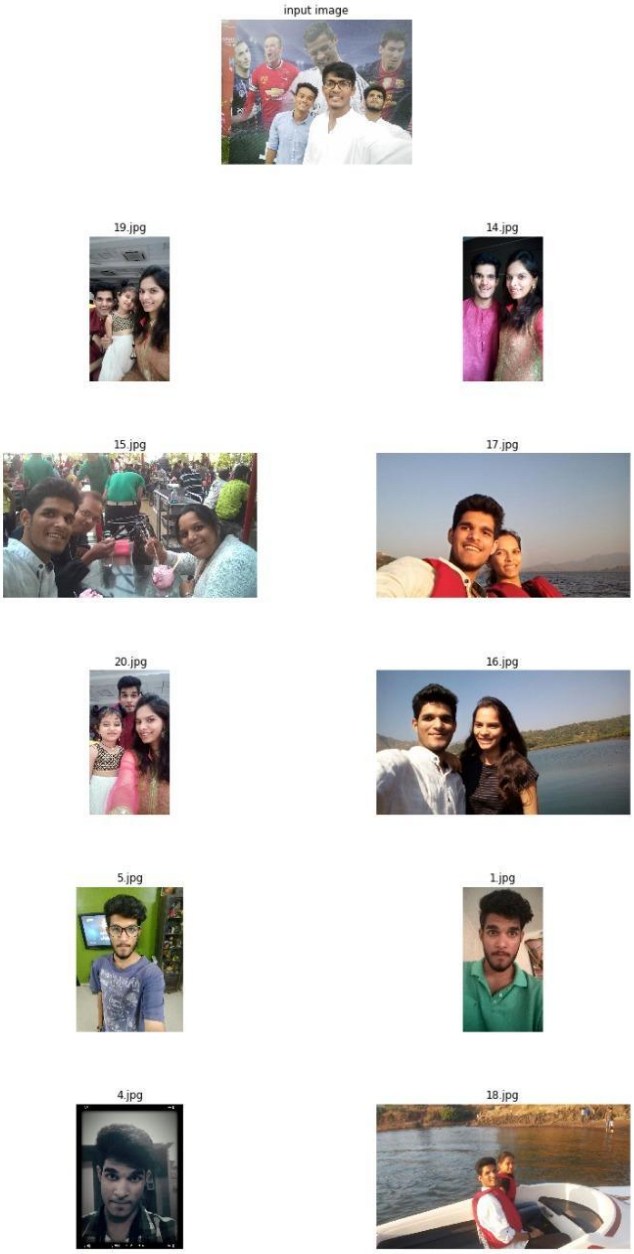
Python:

Python 3.6.3 (Miniconda3 4.3.31 64-bit) by Anaconda, Inc.

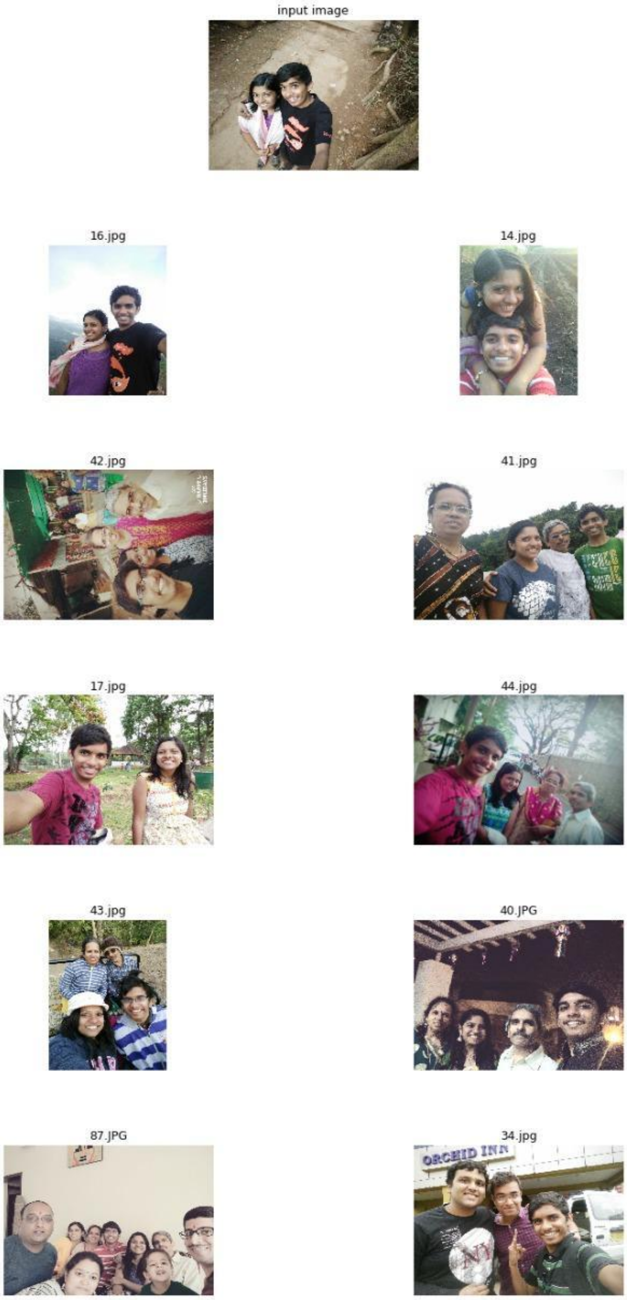
Python Libraries:

1. Numpy (version: 1.13.3)
2. PIL (version: 4.0.0)
3. Matplotlib (version: 2.0.2)
4. face_recognition (version: 0.1.0)
5. Open cv (version: 3.4.1)
6. Skimage (version: 0.13.1)

VI. RESULTS



Result-1



Result-2

VII. Analysis

The model was tested with database of 100 images. The 100 images contained different faces and different backgrounds. The model is giving good results with any number of faces. We gave query of different faces containing from selfie single face to group photo consisting more than 7 faces. In some results, an unknown face is present in the query image. The model identifies it as unknown face and ignores it.

In some cases, database contains new and old images of the person. Even though person's face is not changed much his style is changed. The model retrieves the images that are closely related to the not only face but also his style.

The model gives good results even with the query image is rotated. The model also considers background while retrieving related images as background of query image. The model also finds outs the copy of the query image present in the database using histogram.

The model doesn't retrieve images using only faces. In the case of selfies, the model finds outs the background of the image and retrieves photos containing more than one face but with same background.

The biggest problem of the model is that it takes too much time to compile and index the dataset. Even for 100 images the model took nearly 8 minutes for indexing.

The model fails to understand the semantic concepts clearly. Some results are not according to the background due to the lack of semantic understanding.

VIII. Conclusion

This model will be helpful to find related images just by giving a query image. This model will be helpful to make album containing similar images. This will help to organize bunch of photos. This model not only considers faces but also takes background in consideration. It will also help in finding image copies which are in the bunch of photos. This model considers all the features like shape, texture and color while retrieving related images which gives best results according to query image. If a face is not dataset if ignores that face doesn't compare it with others compares known faces to increase the accuracy the speed of retrieving images is much faster and takes less time to give related images. This model can work with any type of query images and also with clear and non-clear images as it removes noise from the query image. Model also can retrieve related old images over newer images by comparing the style as well as the face. Without giving any date query.

As this model has better image retrieving speed but it has poor database encoding speed as it takes lot of time to indexing of images. Many users find this model helpful for retrieving related images but they wanted to describe the query image by giving some attributes like appearance, background, location etc. because finding the perfect query images is difficult sometimes.

IX. Future Work

The future research includes following points

1. Increasing the size of database by improving indexing of database vector.
2. Expansion of query by adding combination of keywords, numbers, semantic classes
3. We can add date, time, location, device and also name of the person along with query image in the query vector
4. Even though after getting related images user finds some images bad because of some attributes like noise, disturbances and also semantic attributes like appearance, pose etc. So we can add confidence level to our model which will detect how much percent that image is good. So that we can add this semantic input like keywords such good or bad etc.
5. We can add some attributes in the query vectors which will not only retrieve related images but also will be able to retrieve all the attributes of the image like color of the outfit, weather situation etc.

Acknowledgment

We would like to thank Adam Geitgey for face recognition by compiling dlib [28].

Also, we would like to thank our professor Vijayarajan V. for helping us in this project.

References

[1] http://blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html

[2] O. M. Parkhi, A. Vedaldi, A. Zisserman Deep Face Recognition British Machine Vision Conference, 2015.

[3] H.-W. Ng, S. Winkler. A data-driven approach to cleaning large face datasets. Proc. IEEE International Conference on Image Processing (ICIP), Paris, France, Oct. 27-30, 2014

[4] Eakins, John, and Margaret Graham. "Content-based image retrieval." (1999).

[5] Gudivada, Venkat N., and Vijay V. Raghavan. "Content based image retrieval systems." Computer 28.9 (1995): 18-22.

[6] Yue, Jun, et al. "Content-based image retrieval using color and texture fused features." *Mathematical and Computer Modelling* 54.3-4 (2011): 1121-1127.

[7] Shete, Deepak S., M. S. Chavan, and Kit Kolhapur. "Content Based Image Retrieval." International Journal of Emerging Technology and Advanced Engineering 2.9 (2012): 85-90.

[8] Rui, Yong, et al. "Relevance feedback: a power tool for interactive content-based image retrieval." IEEE Transactions on circuits and systems for video technology 8.5 (1998): 644-655.

[9] Rui, Yong, Thomas S. Huang, and Sharad Mehrotra. "Content-based image retrieval with relevance feedback in MARS." Image Processing, 1997. Proceedings., International Conference on. Vol. 2. IEEE, 1997.

[10] Smeulders, Arnold WM, et al. "Content-based image retrieval at the end of the early years." IEEE Transactions on pattern analysis and machine intelligence 22.12 (2000): 1349-1380

[11] https://github.com/tensorflow/models/tree/master/research/deeplab

[12] Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, Liang-Chieh Chen+, George Papandreou+, Iasonas Kokkinos, Kevin Murphy, Alan L. Yuille (+ equal contribution). In ICLR, 2015.

[13] Watson, Andrew B. "Image compression using the discrete cosine transform." Mathematica journal 4.1 (1994): 81.

[14] Saha, Subhasis. "Image compression—from DCT to wavelets: a review." Crossroads 6.3 (2000): 12-21.

[15] Monro, D. M., and B. G. Sherlock. "Optimal quantisation strategy for DCT image compression." IEE Proceedings-Vision, Image and Signal Processing 143.1 (1996): 10-14.

[16] Cintra, Renato J., and Fábio M. Bayer. "A DCT approximation for image compression." IEEE Signal Processing Letters 18.10 (2011): 579-582.

[17] Arbelaez, Pablo, et al. "Contour detection and hierarchical image segmentation." IEEE transactions on pattern analysis and machine intelligence 33.5 (2011): 898-916.

[18] Chan, Tony F., and Luminita A. Vese. "Active contours without edges." IEEE Transactions on image processing 10.2 (2001): 266-277.

[19] Caselles, Vicent, et al. "A geometric model for active contours in image processing." Numerische mathematik 66.1 (1993): 1-31.

[20] Guttman, Michael A., Jerry L. Prince, and Elliot R. McVeigh. "Tag and contour detection in tagged MR images of the left ventricle." IEEE transactions on medical imaging 13.1 (1994): 74-88.

[21] Grigorescu, Cosmin, Nicolai Petkov, and Michel A. Westenberg. "Contour detection based on nonclassical receptive field inhibition." IEEE Transactions on image processing 12.7 (2003): 729-739.

[22] Ma, Wei-Ying, and Bangalore S. Manjunath. "EdgeFlow: a technique for boundary detection and image segmentation." IEEE transactions on image processing 9.8 (2000): 1375-1388.

[23] Mohanaiah, P., P. Sathyanarayana, and L. GuruKumar. "Image texture feature extraction using GLCM approach." International Journal of Scientific and Research Publications 3.5 (2013): 1.

[24] Gong, Peng, Danielle J. Marceau, and Philip J. Howarth. "A comparison of spatial feature extraction algorithms for land-use classification with SPOT HRV data." Remote sensing of environment 40.2 (1992): 137-151

[25] Kekre, H. B., et al. "Image Retrieval using Texture Features extracted from GLCM, LBG and KPE." International Journal of Computer Theory and Engineering 2.5 (2010): 695.

[26] Gao, Cheng-Cheng, and Xiao-Wei Hui. "GLCM-based texture feature extraction." Computer Systems & Applications 6 (2010): 048.

[27] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015

[28] https://github.com/ageitgey/face_recognition