CSE 515 – Group Project Phase-1

Keenan Rahman
Kunhao Zhang
Ian Bolton
Brandon Bayles
Preston Mott

Abstract

In this project we use the Olivetti Faces dataset from AT&T containing ten different images of each of 40 distinct subjects and examined how we could represent them in vector models. The mages were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position. For this phase, 3 models were implemented for modelling and extracting the feature descriptors. We then demonstrated how these vector representations could be utilized to visualize the most similar k images based on the corresponding visual descriptors.

Keywords

Database, Feature descriptors, Color moments, Local binary pattern, Histogram of oriented gradient.

Introduction

For this project phase we chose the Olivetti Faces dataset from AT&T. The original dataset was in form of “png” images. The image is quantized to 256 grey levels and stored as unsigned 8-bit integers; the loader will convert these to floating point values on the interval [0, 1], which are easier to work with for many algorithms. There are ten different images of each of 40 distinct subjects, for each of the subject, the images were taken at different times, varying lighting, facial expressions and facial details. All images were taken against a dark homogeneous background with frontal position.

The goal of this project is to experiment with image features, vector models, and similarity/distance measures.[1] To achieve these goals 3 different feature descriptors were implemented namely, Color moments, Extended local binary patterns, and Histogram of oriented gradients.

1- For color moments calculation, image is divided into 8x8 windows, color moments were calculated on each of the window, and all these results were concatenated to get the unified results. Three color moments were used for calculation (mean, standard deviation, skewness).
2- For Extended local binary pattern it is a joint distribution of gray-scale and rotational invariant LBP with the rotational invariant Variance measure. We set the value neighbor set points to 8 and radius for circle to 1.00 for computation.
3- For Histogram of oriented gradients, number of orientation bin was set to 9, with block size of 2, cell size of 8, and block norms set to L2. [1]

Furthermore, these feature descriptors were being used to help identify “k” similar images in folder. However, the challenge was to combine all these feature descriptors and to bring them on the similar scale, so calculation won’t be biased.

Implementation/Solution Description

For each of the tasks of the project a python file was created to implement the solution. In order to run each of the task the corresponding code file should be run, and the required parameters be provided to get the result. For some tasks data was displayed on terminal, while for other task a corresponding output “pdf” was created in their respective “output/{task_number}/” folder.

Task - 0

For this task, we first created a database connection between the Python and Mongo DB to store all the image data. We then take the Olivetta Faces dataset from sklearn.datasets, convert each of the array data to Binary, to store it in the database. The “id” of each of the image was set from 0-399.

Task - 1

For this task, we take 2 arguments for a function: image id and model name. Using the given image id, a corresponding image is fetched from the database using the data that was saved from running the Task – 0. A model is then executed on the corresponding model name provided as a parameter, that are: color_moment, local_binary_pattern, and histogram_of_oriented_gradients.

Color Moments

Color moments are measures that can be used differentiate images based on their features of color. [2] In order to calculate the color moments, three ways were used (mean, standard deviation, and skewness). The image matrix was divided into 8x8 box to perform calculations. All these calculations are combined to calculate the color moments of a given image.

Mean: The first color moment can be interpreted as being the average color in the given image. [2] This is calculated by formula. [2]

$$E_i = \sum_{j=1}^{N} \frac{1}{N} p_{ij}$$

Here N is the total number of points in the given box and p_{ij} is the value of the i^{th} row and j^{th} column of the box.

Standard Deviation: The second color moment is obtained by taking the square root of the variance of the color distribution. This is calculated by formula. [2]

$$\sigma_i = \sqrt{\left(\frac{1}{N} \sum_{j=1}^{N} (p_{ij} - E_i)^2\right)}$$

Here E_i is the mean value of the image.

Skewness: The third color moment measures how asymmetric the color distribution is and thus it gives information about the shape of the color distribution. This is calculated by formula. [2]

$$s_i = \sqrt[3]{\left(\frac{1}{N} \sum_{j=1}^{N} (p_{ij} - E_i)^3\right)}$$

Extended Local Binary Pattern

The extended LBP is a joint distribution of gray-scale and rotational invariant LBP with the rotational invariant Variance measure.[3] In order to calculate the ELBP we set the local neighborhood to 8.00 and the radius is set to 1.00 on a given image vector. The formula for ELBP is[3]:

$$\text{LBP} = \sum_{i=0}^{P-1} s(n_i - G_c)\,2^i$$

$$s(x) = \begin{cases} 1, & if\ x > 0 \\ 0, & otherwise \end{cases}$$

where P is the number of neighborhood pixels, n_i represents the ith neighboring pixel, and c represents the center pixel. The histogram features of size 2^P is extracted from the obtained LBP code. [3]

Histogram of Oriented Gradients

The histogram of oriented gradients is a feature descriptor used in image processing. The technique counts occurrences of gradient orientation in localized portions of an image. In order to calculate the HOG we set the number of orientation bins to 9, cell size to 8 and block size to 2. The block normalization is set to L2-norm clipping threshold set to 0.2.

$$\text{L2-norm: } f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}$$

L2-hys: L2-norm followed by clipping (limiting the maximum values of v to 0.2) and renormalizing.

For each of the model a separate function is being created that takes image id and image matrix as input and provides result for the corresponding model selected.

Task - 2

Utilizing all the model implementation from task-1 we take a folder with “png” images as input and iterate through each of the image. All models were applied on each of the image and output is being saved to “/output/task_2/” directory with file name as “{image_name}_output.pdf”. The output files contain the requested image, graphical representation of all color moments, pictorial representation of ELBP, and pictorial representation of HOG.

Task - 3

In task-3 we implemented a way for the user to input image id and a folder as an input along with 2 values, namely: model name, and k. The main objective of this task is to compare every image in a folder with the given image using the model’s name specified in the input and returns/visualizes the most similar k images based on the corresponding visual descriptors. In order to compare the image with another image we used L2-norm.

L2 Norm

The L2 norm calculates the distance of the vector coordinate from the origin of the vector space.[4] As such, it is also known as the Euclidean norm as it is calculated as the Euclidean distance from the origin. The result is a positive distance value. Formula:[4]

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^{N} |x_i|^2\right)^{1/2} = \sqrt{x_1^2 + x_2^2 + \ldots + x_N^2}$$

All the results from the images are added to a dictionary where all the distances are sorted to get the k-nearest images.

Task - 4

In task-4 we implemented a way for the user to input image id, a folder, and k-value. For this task we calculate all the values from each of the feature descriptors (color moment, extended local binary pattern, and histogram of oriented gradients) we implemented. For every image in the folder, we calculated the same feature descriptors. We use L2 norm to calculate the distance of the image with another image for each model.

The main hurdle in this task was the distance values provided from each of the models weren’t on same scale. In order to scale the values on a common scale, for each model we calculated a min and max value. For each value we subtracted the value with min of that model and divide it by the difference of max and min of that model to set the value on scale.

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Interface Specification

This program is intended to extract data from olivetta faces or the folder provided in the input and apply all or any one of the feature descriptor requested by the user. Each of the task take different sets of input and output the results as an image or a separate pdf file that are saved in their respective output/{task_number} directory.

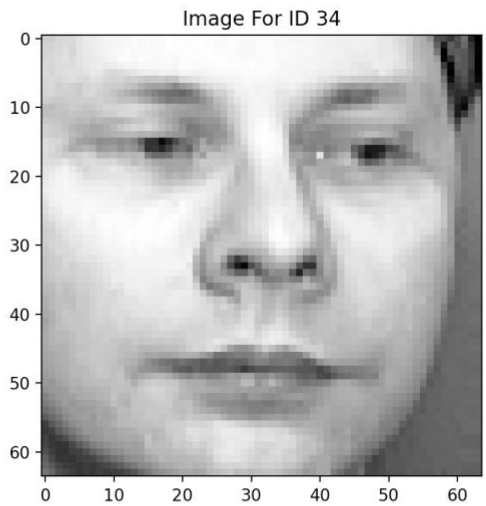
Task - 0

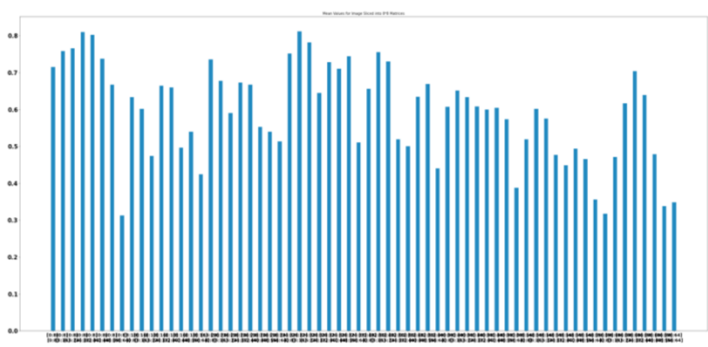
This task takes nothing as input and save the olivetta faces data fetched from scikit learn library to mongoDb server. This image matrix is first converted using pickle.dump function and then converted to Binary format.


```
{
    "_id": 0,
    "image_matrix": {
        "$binary": "gAJjbnVtcHkuY29yZS5tdWx0aWFycmF5Cl9yZWNvbnN0cnVjdApxAGNudW1weQpuZGFycmF5CnEBSwCFcQJjX2NvZGVjcwplbmNvZGUKcQNYAQAAAGJxBFgGAAAAb(
        "$type": "0"
    }
}
```

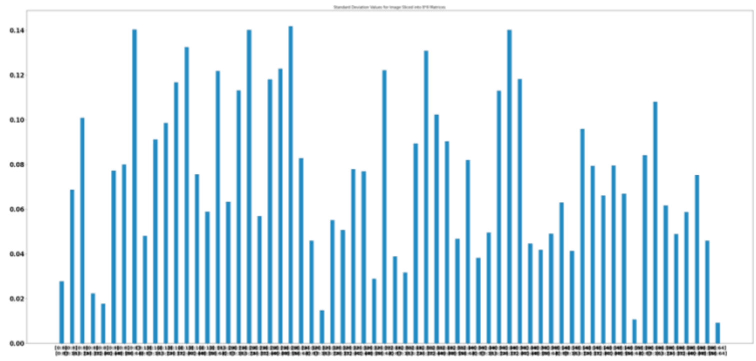

Task – 1

This task takes image id and model name as an input returns the results from requested model.

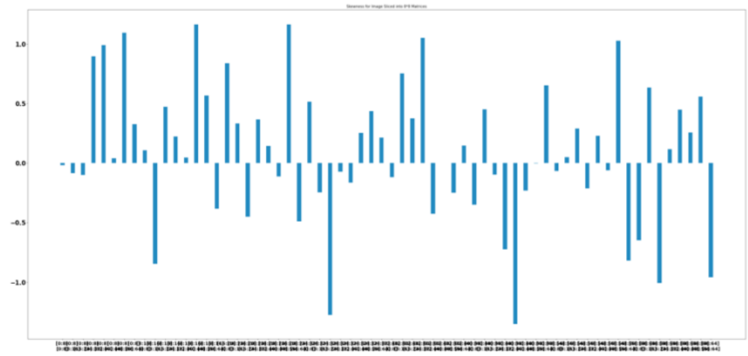




It can be seen from Mean graph that in starting blocks white color density is high, hence, the mean is also high and as it goes towards last blocks the black color density increase causing mean value to decrease.



The color variation is high in the middle part of the image whereas on the corner sides of the image the color variation is low hence the graph is oscillating between high variance to low variance

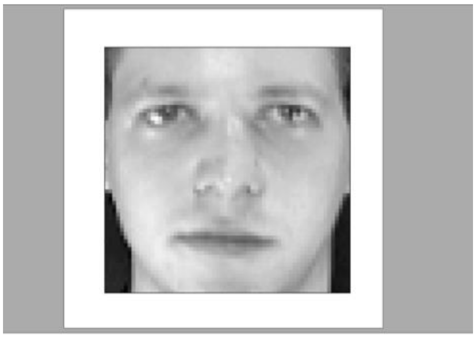


As it can be seen in the image that color is not symmetric in the middle part of the image but since it is symmetric on edges, we can see low level of skewness there.

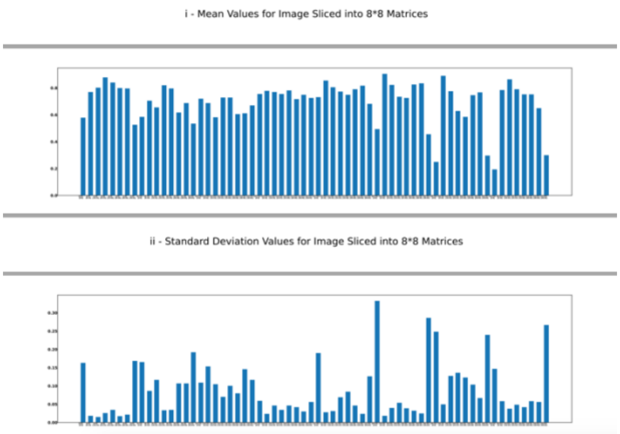
Task – 2

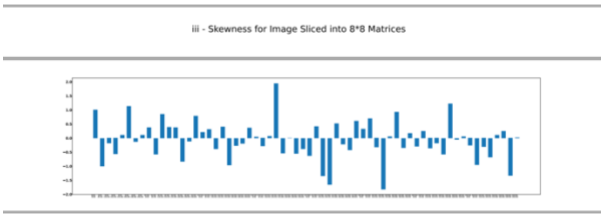
This task takes folder name as input and for each image it applies all 3 feature descriptors and create a compiled result file for each of the image in folder.

OUTPUT FOR THE image-0

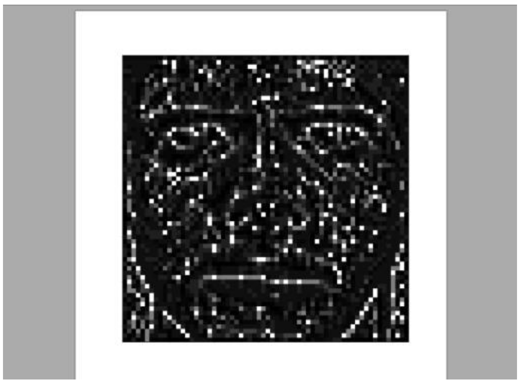


In this image we have more white color blocks and less black blocks. Other than that, we have very less variation and skewness in the image.



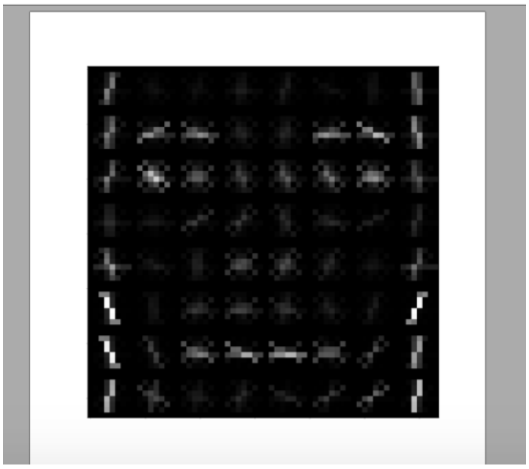


II - Local Binary Pattern Result



Since mostly the neighboring pixels have similar or near similar values, we can see more black blocks in the ELBP version of the given image/

III - Histogram of Oriented Gradients Result



Since in most part of image we have same color scheme we can see lots of black areas, that is no gradient values, but areas where we are transitioning from one color to another can be seen to have different gradient value. This value can be seen on face boundary mostly.

Task – 3

This task takes folder path, image id, model name and k-value as input and for each image it applies the requested feature descriptor and fetch k nearest images based on the descriptor output.

For given input we can see that we have darker color on edges and lighter in the other parts, when we ran HOG with k-value = 2 we got the following results.

In the matching image we can see similar color distribution pattern of having darker shades on the edges and lighter shades on the middle part of the image.

Task – 4

This task takes folder path, image id and k-value as input and for each image it applies the all the feature descriptors, combines the results of all descriptors by scaling them on a common scale and fetch k nearest images based on the scaling output.

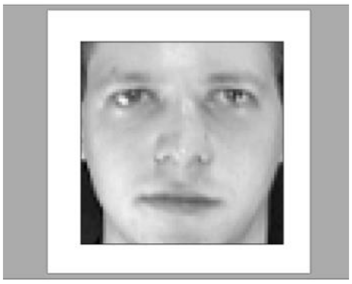
For given input we can see that we have darker color on edges and very scattered color distribution throughout the image, when we ran all feature descriptors with k-value = 2 we got the following results.

As we can see in the output that the overall score of these 2 nearest images follows a similar pattern as in the input image. The color is scattered throughout the image and the color distribution is very much like the input image, hence, we got the following output.

Output Results Explanation

Task – 2 : Set – 1 : Image – 0

OUTPUT FOR THE image-0



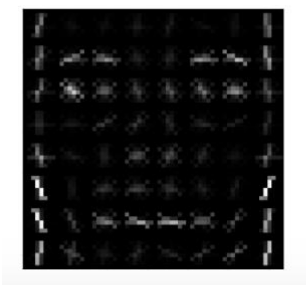
For this image we needed to calculated color moments, ELBP, and HOG

As the output of the Mean Color Moment, we can see lots of bars approaching 1, that is because we have lots of white color in the picture.

The standard deviation is high on the some of the places where we can see color changing shades. Especially in area of lips where we can see lots of different shades. The area where face ends can be seen with spike in deviation.

For Skewness, we can see very less asymmetric distribution throughout the image. However, in some areas it can be seen going +1 or -1 due to changing of color shades.

For ELBP, since mostly the neighboring pixels have similar or near similar values, we can see more black blocks in the ELBP version of the given image.

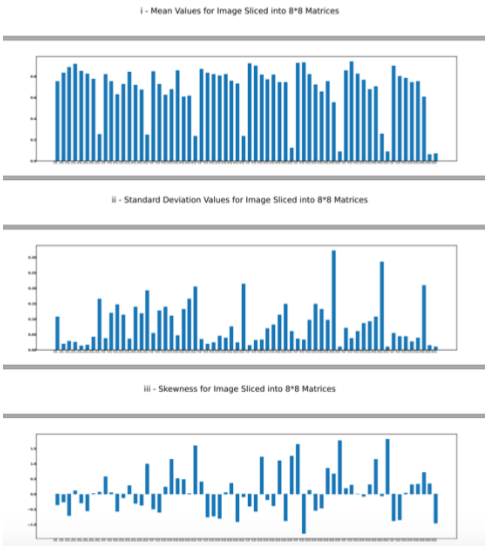


Since in most part of image we have same color scheme we can see lots of black areas, that is no gradient values, but areas where we are transitioning from one color to another can be seen to have different gradient value. This value can be seen on face boundary mostly.

Task – 2 : Set – 1 : Image – 1

OUTPUT FOR THE image-1

For this image we needed to calculated color moments, ELBP, and HOG

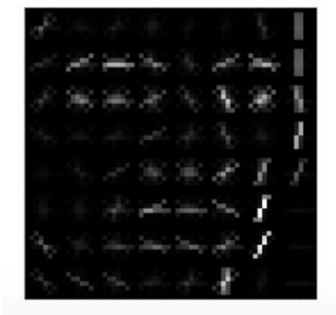


As the output of the Mean Color Moment, we can see pattern of bar going to high and gradually to low, that is because we have darker shades on right side compared to left and hence graph go from high to low in oscillating way.

The standard deviation is low and then it goes up in oscillating way, this is because we have more variation of color towards the right blocks of the image

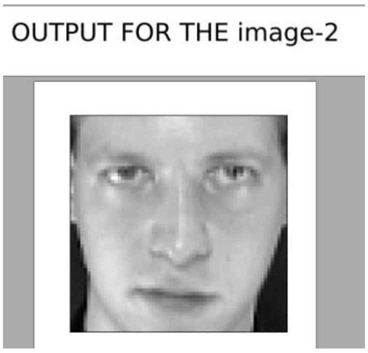
For Skewness, we can see asymmetry more towards lower area of the image that is because we have color variation for chin, lips and nose.

For ELBP, since mostly the neighboring pixels does not have very much similar color ad due to lighting effect, we have lots of color changing pattern.

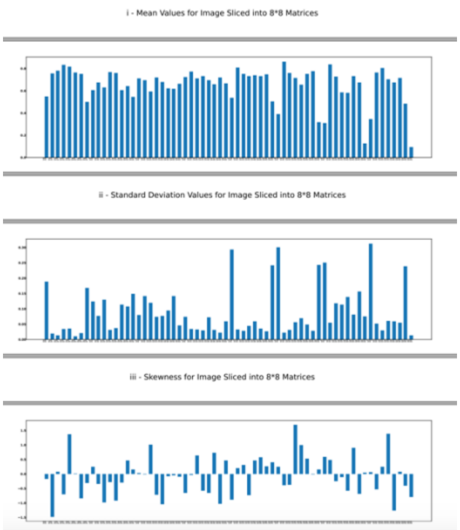


Since for the areas like eyes, nose, and face boundary have color change and color goes from light to dark we can see gradients moving down and for areas where color changes to brighter one gradient are increasing.

Task – 2 : Set – 1 : Image – 2



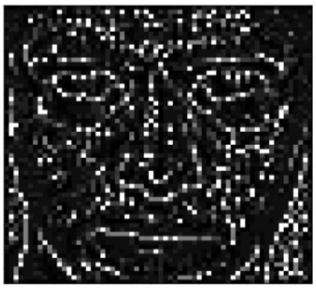
For this image we needed to calculated color moments, ELBP, and HOG



As the output of the Mean Color Moment, we can see almost all the bars in the graph are touching 1, this is because we have lots of white color in the image. To some areas that are dark we have lower median values.

The standard deviation is low throughout the image as there is not much color variation going on. In the last blocks, it gets high because colors are continuously changing.

For Skewness, we can see asymmetry more towards lower area of the image that is because we have color variation for chin, lips and nose. However, in upper and middle blocks skewness is almost nil.



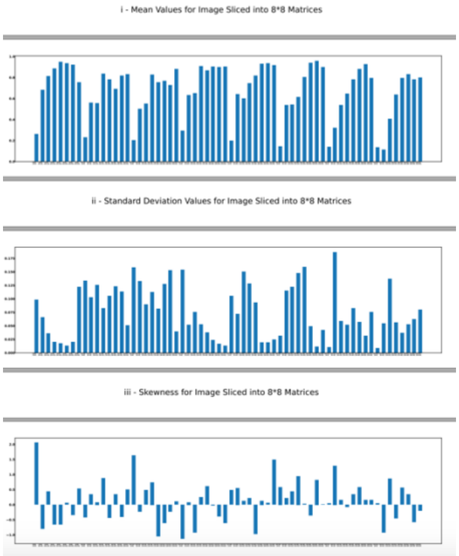
For ELBP, since mostly the neighboring pixels have lots of variation of white color, we can see lots of gradients having white values. This is because neighboring pixels have white colors.

Since for the areas like eyes, nose, and face boundary have color change and color goes from light to dark we can see gradients moving down and for areas where color changes to brighter one gradient are increasing.

Task – 2 : Set – 1 : Image – 3

OUTPUT FOR THE image-3

For this image we needed to calculated color moments, ELBP, and HOG



As the output of the Mean Color Moment, we can see pattern of bar starting from low and gradually go to high, that is because we have darker shades on left side compared to right and hence graph go from low to high in oscillating way.

The standard deviation is high and then it goes down in oscillating way, this is because we have more variation of color towards the left blocks of the image

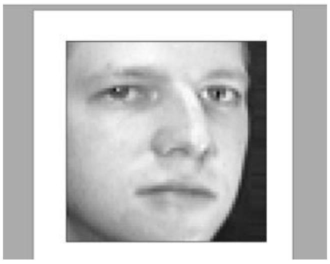
For Skewness, we can see asymmetry more towards start and middle area of the image that is because we have color variation for eyes, lips and nose.

For ELBP, since mostly the neighboring pixels does not have very much similar color and due to lighting effect, we have lots of color changing pattern.

Since for the areas like eyes, nose, and face boundary have color change and color goes from dark to light we can see gradients towards left part of the image compared to right side of image.

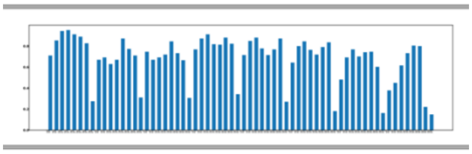
Task – 2 : Set – 1 : Image – 4

OUTPUT FOR THE image-4

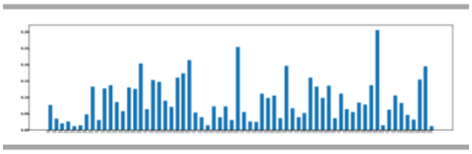


For this image we needed to calculated color moments, ELBP, and HOG

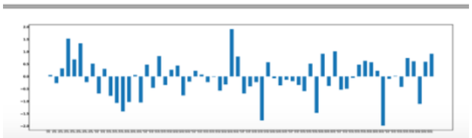
i - Mean Values for Image Sliced into 8*8 Matrices



ii - Standard Deviation Values for Image Sliced into 8*8 Matrices



iii - Skewness for Image Sliced into 8*8 Matrices



As the output of the Mean Color Moment, we can see pattern of bar starting from high and gradually go to low, that is because we have darker shades on right side compared to left and hence graph go from high to low in oscillating way.

The standard deviation is high in the middle and bottom, this is because we have more variation of color towards the left blocks of the image

For Skewness, we can see asymmetry is very less throughout the image, its high on middle and bottom part due to color variation for eyes, lips and nose.

For ELBP, since mostly the neighboring pixels does not have very much similar color and due to lighting effect, we have lots of color changing pattern. These are more towards right side as we move towards darker shades

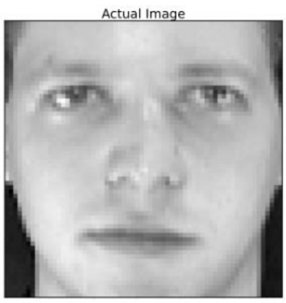
Since for the areas like eyes, nose, and face boundary have color change and color goes from dark to light we can see gradients towards right and bottom part of the image compared to left side of image.

Task – 3 : Set – 1 : Image – 0 : CM

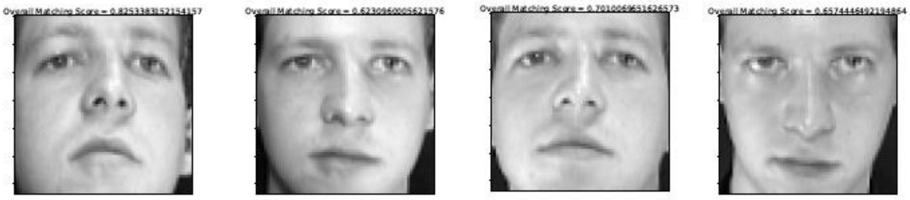
For this image we needed to find 4-nearest images using Color Moments.

For the 4 nearest image we got these following images with each having overall matching score of 6.72, 7.22, 7.27, 7.71 respectively. For all these images we have color distribution common to each other, for instance in all the images we can see dark shades towards the border of the images. The orientation of eyes and lips are at similar blocks compared to the original image. Hence, 4 nearest images were selected.

Task – 3 : Set – 1 : Image – 0 : HOG



For this image we needed to find 4-nearest images using Histogram of Oriented Gradients.



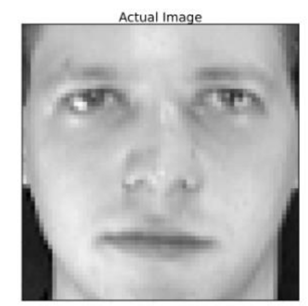
For the 4 nearest image we got these following images with each having overall matching score of 0.62, 0.65, 0.70, 0.82 respectively. For all these images we have color distribution gradients like each other we can see darker gradients on the edges and having same face orientation as the actual image. Hence, 4 nearest images were selected.

Task – 3 : Set – 1 : Image – 0 : ELBP

For this image we needed to find 4-nearest images using Extended Local Binary Patterns.

For the 4 nearest image we got these following images with each having overall matching score of 1247.8, 1284.8, 1382.7, 1393.3 respectively. For all these images we have color patterns in much similar way. We have similar lightning on the cheeks and nose part. The lips for all images are dull with bottom edges being darker compared to top edges. Hence, 4 nearest images were selected.

Task – 4 : Set – 1 : Image – 0



For this image we needed to find 4-nearest images using All models.

For the 4 nearest image we got these following images with each having overall matching score of 0.18, 0.31, 0.38, 0.41 respectively. For all these images we have color patterns in much similar way. For all images we calculate the color moments, elbp and hog and scale them all using min-max logic.

System Requirement & Execution Instruction

Install following python version and libraries in order to run the python scripts successfully. Also install latest version of MongoDb and run it on the machine, you need to update the connection URL in task_0 and task_1 code so that in can run successfully on new environment. The details of the input and output of each task is listed below.

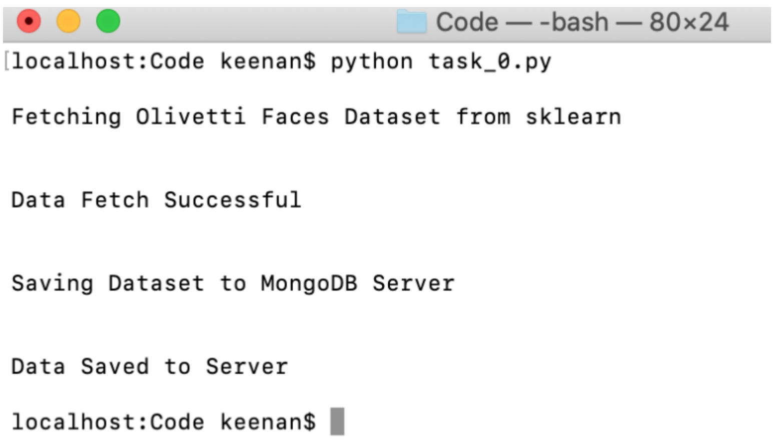
Pre-requisites

- Python 3.9.7
- Matplotlib 3.4.3
- NumPy 1.21.2
- Pymongo 3.12.0
- Scikit-Image 0.18.3
- Scikit-Learn 0.24.2
- SciPy 1.7.1

Task - 0

This task does not take anything as input and is run by executing command in the terminal.
Command: python task_0.py

Once the command is run, it displays following output.



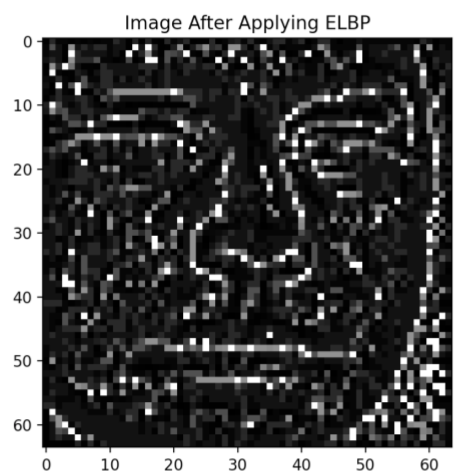
Task – 1

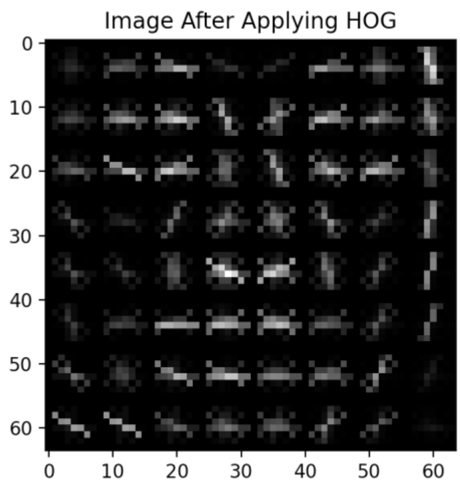
This task takes image id and model name as input and is run by executing command in the terminal.
Command: python task_1.py

Once the command is run it ask for 2 inputs as follow:

```
Code — Python task_1.py
localhost:Code keenan$ python task_1.py

Please Enter Image ID {0 - 399}: 23

Enter Model Names from
- color_moment
- local_binary_pattern
- histogram_of_oriented_gradients

Please Enter Model Name: color_moment
```

Then it displays following output:





Task - 2

This task takes only folder path as input and is run by executing command in the terminal.
Command: python task_2.py

Once the command is run it ask for 1 input as follow:

```
Code — Python task_2.py — 174×24
localhost:Code keenan$ python task_2.py

Please Enter Folder Path: /Users/keenan/Desktop/ASU/Semester 1/Multimedia and Web Databases/Project/Phase-1/set3
```

Then it displays following output and generate a pdf file:

```
Output File for "image-40" has been created

Output File for "image-30" has been created

Output File for "image-10" has been created

Output File for "image-0" has been created
```

Task - 3

This task takes folder path, image id, model name and k-value as input and is run by executing command in the terminal.
Command: python task_3.py

Once the command is run it ask for inputs as follow:

```
Please Enter Folder Path: /Users/keenan/Desktop/ASU/Semester 1/Multimedia and Web Databases/Project/Phase-1/set3

Please Enter Image ID {image-392.png}: image-30.png

Enter Model Names from
- color_moment
- local_binary_pattern
- histogram_of_oriented_gradients

Please Enter Model Name: histogram_of_oriented_gradients

Please Enter K-Value: 2
```

Then it displays following output and generates a pdf file:

```
Generating Output File for "image-30.png" using Model "histogram_of_oriented_gradients"

Calculating Color Moments for "image-40.png"

Calculating Color Moments for "image-10.png"

Calculating Color Moments for "image-0.png"

Output File Generated
```

Task - 4

This task takes folder path, image id, model name and k-value as input and is run by executing command in the terminal.
Command: python task_4.py

Once the command is run it ask for inputs as follow:

```
Please Enter Folder Path: /Users/keenan/Desktop/ASU/Semester 1/Multimedia and Web Databases/Project/Phase-1/set3

Please Enter Image ID {image-392.png}: image-40.png

Please Enter K-Value: 3
```

Then it displays following output and generates a pdf file:

```
Applying Feature Descriptors for "image-30.png"

Applying Feature Descriptors for "image-10.png"

Applying Feature Descriptors for "image-0.png"

Scaling All the Costs

Scaling All the Costs

Scaling All the Costs

Generating Output File for "image-40.png"

Output File Generated
```

Related Work

This program is a basic form of representing images using feature vectors. There are various feature descriptor algorithms in use today out which few are:

- DAISY is a feature descriptor like SIFT formulated in a way that allows for fast dense extraction.
- Peak Local Max, find peaks in an image as coordinate list or boolean mask.
- Binary Robust Independent Elementary Features is an efficient feature point descriptor. It is highly discriminative even when using relatively few bits and is computed using simple intensity difference tests.

In order to calculate the distance among the image vector we used Euclidean Distance Formula. There are other various type of distance/similarity matrix that are in use today out of which few are:

- Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space.
- Manhattan distance is a metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates.
- Minkowski distance is a generalisation of the Euclidean and Manhattan distances.
- Jaccard similarity is a statistic used for gauging the similarity and diversity of sample sets.

Conclusion

This phase of the project showed how different feature descriptors can provide different results. We used three different vector models to get the results from grayscale images and used these models simultaneously to get the k-nearest images using different distance/similarity measures. The graphs generated from color moments describe how the pixels in the image are spread out. The pictorial representation from ELBP and HOG demonstrates the use of image gradients and texture. It has further been determined that when LBP is combined with the Histogram of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.[5]

Bibliography

1. Candan, K. S. (2021). CSE 515 Multimedia and Web Databases Phase #1 Project Description (pp. 1-2). Published on Piazza
2. Noah Keen. (2005). Color Moments, https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0405/KEEN/
3. L. Mdakane and F. van den Bergh (2001). Extended Local Binary Pattern Features for Improving Settlement Type Classification of QuickBird Images, CSIR Research Space
4. David G. Lowe (2004). Distinctive Image Features from Scale-Invariant Keypoints, University of British Columbia Vancouver, B.C., Canada
5. Xiaoyu Wang, Tony X. Han, Shuicheng Yan (2009), An HOG-LBP Human Detector with Partial Occlusion Handling, ICCV

Appendix

During this phase of the project each individual was responsible for implementing the program on their own. All five members did coordinate to help validate data outputs and discuss a variety of techniques for calculating intermediary values.

Keenan Rahman - Individual Project Code and Report
Preston Mott - Individual Project Code and Report
Brandon Bayles - Individual Project Code and Report
Ian Bolton - Individual Project Code and Report
Kunhao Zhang - Individual Project Code and Report