Image analysis using vector models and similarity measures

Group Members:

1. Dhruvil Deepakbhai Shah / dshah47@asu.edu

2. Deep Vijaykumar Patel / dpatel94@asu.edu

3. Nihar Samirbhai Patel / npatel69@asu.edu

4. Manikanta Mudara / mmudara@asu.edu

5. Pramukh Belam / pbelam@asu.edu

6. Vijay Aravindh Viswanathan / vviswa19@asu.edu

Abstract

The Olivetti dataset is consisting of 400 images. These images are of 40 different persons. There are 10 unique images per person. These images vary in many aspects like lightning, facial expression (smile or no smile), facial details (glasses or no glasses) etc. All the images are with a dark uniform background with the person in front position. In this phase of the project, initially during Task-1 and Task-2, the features of the images are extracted and stored using three models: Color moments, Extended local binary pattern and Histogram of oriented gradients given a folder of images. After the extraction of the features, a program function is implemented in Task-3 that returns k nearest images given a base image using the similarity measures like “Cosine similarity” and “Earth movers distance”. Finally, in Task-4, a program is generated that combines the decisions of all three models to find the k nearest images given a base image. During each task, a separate .txt file is generated to store the outputs.

Keywords

Color moments, Extended local binary patterns, Histogram of oriented gradients, Feature extraction, Olivetti dataset

Introduction

The Olivetti dataset is a comprehensive dataset comprising 400 facial images in grayscale space from 40 different persons. The images are of 64 x 64 dimensions and have a constant black background. The images differ in various aspects. The dataset includes the metadata explaining the dimensionality of the dataset and how the images are taken. The metadata is used in the Task-0 to familiarize with the dataset. There are 3 types of features extracted from the images: Color moments, Local binary patterns and Histogram of oriented gradients.

The features of all the images in the folder/dataset are extracted and they are used to compute the distance score (similarity score) given a base image. There are two distance functions/ measures of similarity used in this phase of the project: Earth mover’s distance (EMD) and Cosine similarity.

Terminology

This section shades light on frequently used phrases and discusses the techniques used in this project phase.

Grayscale and RGB images:

These images consist of only 1 channel. Values are in 0 to 255 range and represents the intensity of pixels. RGB images contain 3 channels (Red, Green, Blue) and values in these channels depicts the intensity of the pixel in the respective channel.

Color Moments:

Color moments are mostly employed as features in image retrieval applications for color indexing in order to compare how similar two images are based on color. In order to identify and retrieve a comparable image, one image is usually compared to a database of digital images with pre-computed characteristics. Each picture comparison yields a similarity score, the lower the number, the more similar the two photos are expected to be. Color moments comprise the Mean, Standard deviation and Skewness of the image since the low-order moments contain the majority of the color distribution information. Color moments are an excellent feature to utilize in changing lighting circumstances because they encode both shape and color information, but they can't manage occlusion very well. Each channel has three color moments i.e. Grayscale images have 3 color moments (one channel) while RGB color model has nine moments and CMYK color model has twelve moments.

Extended Local Binary Pattern:

A representation of an image based on the neighbors of a chosen pixel is known as a Local Binary Pattern (LBP). Each pixel is compared to the values of all the other pixels within a specified radius R, yielding a binary representation (encoded as decimal) for each pixel. As a result, a 64 X 64 picture yields a 64 X 64 LBP matrix. The LBP feature vector is the histogram returned by the LBP function.

In its most basic form, the LBP feature vector is produced as follows:

* Divide the window you're looking at into cells (e.g. 16x16 pixels for each cell).
* Compare each pixel in a cell to each of its eight neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels in a clockwise or counter-clockwise direction around a circle.
* Write "0" where the value of the center pixel is larger than the neighbor's value. If not, write "1." This yields a binary number of eight digits (which is usually converted to decimal for convenience).
* Create a histogram of the frequency of each "number" occurring in each cell (i.e., each combination of which pixels are smaller, and which are greater than the center). A 256-dimensional feature vector may be observed in this histogram.
* Normalize the histogram if desired.
* All of the cells' histograms should be concatenated (normalized). This generates a feature vector for each window/grid (of length 256).

Histogram of Oriented Gradients

The histogram of oriented gradients (HOG) is a feature descriptor for object recognition in computer vision and image processing. The method counts the number of times a gradient orientation appears in a certain area of a picture. This version employs 9 orientations with an 8 × 8 cell size and a 2 x 2 block size.

Similarity Metrics

This is used to determine how similar two data items are. Euclidean, Earth Movers distance, Pearson Correlation, KL Divergence, Mahanalobis Distance, Cosine Similarity, and other distance measuring techniques are among them.

Earth mover’s distance

The earth mover's distance (EMD) is a statistic that calculates the distance between two probability distributions across a given region D. This is known as the Wasserstein metric in mathematics. The Earth Mover's Distance (EMD) is a method for assessing dissimilarity between two multi-dimensional distributions in a feature space when a distance measure between single features, referred to as the ground distance, is provided. This distance is 'lifted" by the EMD from individual features to entire distributions.

Cosine similarity

**Cosine similarity is a measure of similarity that quantifies the cosine of the angle between two non-zero vectors in an inner product space. The cosine of 0° equals 1, and for every angle in the range (0,] radians, it is less than 1. It is therefore a judgment of orientation rather than magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors orientated at 90° to each other have a similarity of 0, and two vectors diametrically opposite have a similarity of -1, regardless of magnitude.**

Problem Specification

This phase of the project consists of 4 tasks:

**Task 1:** Implement a program that extracts, and outputs (in a human readable way) the appropriate feature descriptors given an image ID and one of the following models:

Color moments, Extended local binary patterns, Histogram of oriented gradients.

**Task 2:** Create a program that, given an image folder, extracts and saves feature descriptions for all of the photos in the folder.

**Task 3:** Implement a program that retrieves and visualizes the most identical k photos based on the related visual descriptors, given a folder with images and an image ID, a model, and a value "k." List the overall matching score for each match.

**Task 4:** Implement a program that retrieves and visualizes the most comparable k photos based on all associated visual descriptors, provided a folder with images and an image ID and a value "k." List the total matching score as well as the contributions of the individual visual models for each match.

Assumptions

The following assumptions are made during the implementation of this phase of the project.

1. All images in the folder are of the same size and shape 64 x 64.
2. The images in the folder are grayscale.
3. The libraries used to compute the distance give accurate results.
4. If two photos have the same distance/similarity score, the image with the lowest numerical image ID is picked as the more similar.

Description of the proposed solution/implementation

In the proposed solution, the Earth mover’s distance and Cosine Similarity are used as an measure of similarity between images.

**Task 1**

In this task, the features of the images are extracted using three models: Color moments, Extended local binary patterns, Histogram of oriented gradients.

**Task 1.1** Color moments:

Function: ColorMoments()

Input: A single array containing array(s) of image(s)

Output: Array comprising features of the image(s)

First, the images were split into 8 x 8 grids resulting in 64 8 x 8 grids (per image) and Mean, Standard deviation and Skewness were calculated for each one of them.

Mean values of the 64 8X8 windows result into a matrix of shape 8x8:

Standard Deviation values of the 64 8X8 windows result into a matrix of shape 8x8:

Skewness values of the 64 8X8 windows result into a matrix of shape 8x8:

Hence, we get 8 x 8 x 3 (For Mean, Standard Deviation, Skewness) matrix for each image.

**Task 1.2** Extended local binary patterns:

Function: localBinPattern()

Input: A single array containing array(s) of image(s)

Output: Array comprising features of the image(s)

The default library “local\_binary\_pattern” from skimage.feature is used for computing the extended local binary patterns of an image. The parameters passed in this default method: Image array containing the pixel values, 8 (Number of neighbors of the cell), 1 (Radius), ‘ror’ (Rotational invariant).

The shape of the output is (m, 256) where m is the number of images in the folder. The 2nd dimension (256) depicts that there are 256 values (256 bins in histogram). So the image is represented by 1-D vector of 256 length.

A collage of a person wearing glasses

Description automatically generated with medium confidence

**Task 1.3** Histogram of oriented gradients:

Function: histOrientedGrad

Input: A single array containing array(s) of image(s)

Output: Array comprising features of the image(s)

The default library “hog” from skimage.feature is used to compute the Histogram of oriented gradients features. The parameters passed into the method: Image array, Orientations = 9, Pixels per cell = (8,8), Cells per block = (2,2), Normalization = ‘L2-Hys’. The output feature descriptor contains 1764 values. For m images in the folder, the final returned array has the shape of (m, 1764).

A collage of a person's face

Description automatically generated with medium confidence

FINAL FUNCTION TASK1:

Input: Path of the image, Model function

Output: A text file (TASK1.txt) containing the feature descriptors is generated in the local directory

Function call:

TASK1('image-0.png', ColorMoments)

TASK1.txt:

Image image-0.png :

<function ColorMoments at 0x7fa4bcb06cb0> :

1.470781250000000000e+02

4.150425338745117188e+01

-5.505471229553222656e-01

1.960937500000000000e+02

4.512893676757812500e+00

-7.120791077613830566e-01

2.039218750000000000e+02

3.696724414825439453e+00

**Task 2**

FINAL FUNCTION TASK2:

Inputs: An array containing the lists of the images in each subfolder (Shape of this array will be (m, ) for a folder containing m subfolders, each of this 3 arrays/lists have shape of (n, 64, 64) where n=number of images in that subfolder)

Output: Text files containing the feature descriptors of the images for the m sets(subfolders) for all 3 models are created in each subfolder. Each set will contain the following text file: TASK2outputFeatures.txt after executing this function with proper parameters. This text file contains the features for all the images in the set(subfolder).

Text

Description automatically generated

Function call:

TASK2(all\_data)

TASK2outputFeatures.txt (for first image in set1):

Image image-0.png :

Color momemts [Mean, Standard deviation, Skewness]: (Shape:(8, 8, 3))

[[[ 1.47078125e+02 4.15042534e+01 -5.50547123e-01]

[ 1.96093750e+02 4.51289368e+00 -7.12079108e-01]…]…]

LBP feature values: (Shape:(256,))

[3.22265625e-02 5.98144531e-02 0.00000000e+00 6.37207031e-02

0.00000000e+00 6.10351562e-03 0.00000000e+00 1.37695312e-01

0.00000000e+00 7.32421875e-04 … ]

HOG feature values: (Shape:(1764,))

[0.34298747 0.34298747 0.01171484 ... 0.01057377 0.04454767 0.2046834 ]

**Task 3**

FINAL FUNCTION TASK3:

Inputs: Folder path containing the subfolders containing the images, Name of the image to be compared, Model function, Value of k

Output: Each subfolder in the folder will have 3 text files namely, TASK3KNearestColorMoments.txt, TASK3KNearesthistOrientedGrad.txt, TASK3KNearestlocalBinPattern.txt comprising the names of the k-nearest images and their respective distances with the base image.

Color moments:

The earth mover’s distance is used for computing the distance between images using color moments features. The color moments features are stored as 1D array of shape: (192, ). The shape of the distance vector will be (m, ) where m is number of images.

Extended local binary patterns:

The cosine similarity is used to calculate the distance between images using the local binary patterns features. The LBP features are stored as 1D array of shape: (256, ). The shape of the distance vector will be (m, ) where m is number of images.

Histogram of oriented gradients:

The earth mover’s distance is used for computing the distance between images using histogram of oriented gradients features. The HOG features are stored as 1D array of shape: (1764, ). The shape of the distance vector will be (m, ) where m is number of images.

Function call:

TASK3('images/set3/', 'image-0.png', ColorMoments, 4)

TASK3KNearestlocalBinPattern.txt (for set1):

Distance between image-0.png and image-6.png : 0.0010641066735831428

Distance between image-0.png and image-4.png : 0.0023483014619594123

Distance between image-0.png and image-2.png : 0.0030176596182959203

Distance between image-0.png and image-3.png : 0.003063406130106139

Output of the code for image-0.png in set3 (Visualization and printing the distance of k nearest images):

**A collage of a person's face

Description automatically generated with medium confidence**

**Task 4**

FINAL FUNCTION TASK4:

Inputs: Folder path containing the subfolders containing the images, Name of the image to be compared, Model function, Value of k

Output: Each subfolder in the folder will have 3 text files named TASK4outputFeatures.txt comprising the names of the k-nearest images and their respective distances with the base image.

The weightage given to all three model functions are as follows:

Color moments model (ColorMoments): 0.45 (45%)

Extended local binary pattern (localBinPattern): 0.1 (10%)

Histogram of oriented gradients (histOrientedGrad): 0.45 (45%)

Color moments:

The earth mover’s distance is used for computing the distance between images using color moments features. The color moments features are stored as 1D array of shape: (192, ). The shape of the distance vector will be (m, ) where m is number of images.

Extended local binary patterns:

The cosine similarity is used to calculate the distance between images using the local binary patterns features. The LBP features are stored as 1D array of shape: (256, ). The shape of the distance vector will be (m, ) where m is number of images.

Histogram of oriented gradients:

The earth mover’s distance is used for computing the distance between images using histogram of oriented gradients features. The HOG features are stored as 1D array of shape: (1764, ). The shape of the distance vector will be (m, ) where m is number of images.

Function call:

TASK4('images/set1', 'image-0.png', 4)

TASK4outputFeatures.txt (for set1): ­­­­

Distance between image-0.png and image-2.png : 0.21778326496256237

Distance between image-0.png and image-6.png : 0.23122469688387892

Distance between image-0.png and image-8.png : 0.2369807701134989

Distance between image-0.png and image-5.png : 0.24155092408073192

Output of the code (Visualization and printing the distance of k nearest images):

A collage of a person's face

Description automatically generated

Interface Specifications

Run the “CSE515 project phase1.ipynb” file into Google colab because there will be no dependency and environment issues and code will run smoothly.

System requirements /installation and execution instructions:

The solution will work errorless with Python 3.6+.

System: Google colab (Tested only on this platform)

Requirements:

numpy==1.19.5

scipy==1.4.1

scikit-image==0.16.2

scikit-learn==0.22.2.post1

matplotlib==3.2.2

Execution steps:-

1. Open the CSE515 project phase1.ipynb file into Google colab (recommended) and **upload two files in the local directory: test\_imgage\_sets.zip and image-0.png**
2. Hit “**Run all**” in Google colab from “Runtime” tab.
3. After successful run, TASK 0 cell will print the meta data of the dataset.
4. After successful run, TASK 1 cell will generate the text file (TASK1.txt) in the local directory.
5. After successful run, TASK 2 cell will generate the text file (TASK2outputFeatures.txt) in each set.
6. After successful run, TASK 3 cell will generate the text files (TASK3KNearestColorMoments.txt, TASK3KNearesthistOrientedGrad.txt, TASK3KNearestlocalBinPattern.txt) in set1, set2 and set3.
7. After successful run, TASK 4 cell will generate the text file (TASK4outputFeatures.txt) in each set.

**Conclusions**

It can be observed from the results of this study's research that different feature models contain variable quantities of information, and that information loss is critical for similarity comparisons. Varied models provide different outcomes from time to time, and the discrepancies are attributable to the intrinsic loss induced by the model selected.

Bibliography

1. T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971-987, July 2002, doi: 10.1109/TPAMI.2002.1017623.
2. <https://en.wikipedia.org/wiki/Cosine_similarity>
3. <https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/RUBNER/emd.htm>
4. <https://en.wikipedia.org/wiki/Earth_mover%27s_distance>
5. <https://en.wikipedia.org/wiki/Color_moments>
6. <https://en.wikipedia.org/wiki/Local_binary_patterns>
7. <https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients>

Appendix

Specific roles of the group members

Each team member was responsible for this part of the project on their own. The collaboration was confined to the overall solution design and idea sharing.