Image Sorting Using Object Detection and Face Recognition

Shaikh Arbaz
Dept of Computer Engineering
Kalsekar technical campus
Mumbai University
Navi Mumbai, India

Shaikh Sohail
Dept of Computer Engineering
Kalsekar technical campus
Mumbai University
Navi Mumbai, India

Shaikh Rehan
Dept of Computer Engineering
Kalsekar technical campus
Mumbai University
Navi Mumbai, India

Mubashir Khan
dept of computer science
Kalsekar technical campus
Mumbai University
Navi Mumbai, India

Abstract:- The user generates a bulk amount of multimedia data among which one of the most used files is images. The average user does not bother to organize the images. This application is used to organize the images based on the object and faces of a person present in the image. It uses object detection, face detection, face recognition to categorize the images into its respective directories. Object Detection uses the YOLO algorithm to detect an object, for face detection it uses Haar Cascade and for face recognition LBPH algorithm. Using the above-mentioned algorithm images are categorized into directories.

Keywords:- *Object Detection, Face Detection, Face Recog-nition, YOLO, Haar cascade, LBPH.*

I. INTRODUCTION

Image is one of the most used multimedia files. Since smartphones are developed cameras as well, everyone takes images and the trillions of images stored. Since these are not going to slow down, the digital image continues to grow. 1.2 trillion images are generated by the end of 2017. 4.7 trillion photos will be stored [10]. 182 photos per month are taken by the average iOS user, whereas 111 photos are being taken by average Android user [9]. Since there are bulk amounts of photos the average user does not bother to organize the images. It takes a considerable amount of time to organize these images.

This application helps you to quickly organize the unsorted image. The basic idea is to give images as input to the program, and objects or faces are detected in the images. The image will then be moved or copied to a folder to their respective class. This allows you to rapidly go through and organize your large amounts of pictures.

The input image is fed to the application, it will detect whether there is the face or not, if the face is detected in the image then face recognition is applied to the image. For face detection ”Haar-Cascade algorithm” is used. After the face is recognized it will create a directory by the name of the given class label and for face recognition ”Linear Binary Pattern Histogram” (LBPH) algorithm is used. If a face is not found then object detection will be used on the image. For object detection YOLO algorithm is used. If any object is detected in the image, it will be moved to the respective directory by the name of the class label.

Additionally, the search by face option allows users to search images with similar faces in the directory. All the images having the same face will be displayed.

II. LITERATURE REVIEW

In this section, we reviewed three different research papers related to our System, to understand their weakness and how we can overcome them.

A. Object Recognition in Images

Object Recognition is a field of study in image processing. The process of identifying objects in video or an image is termed as Object Recognition. It has a huge number of applications in the field of activity recognition, robot localization, and automation, etc. Objects appear different when seen from a different perspective. It should be invariant to changed viewpoints, robustness, occlusion and object transformations. This task targets to perform a technique including mainly 2 stages. In the first stage, the input image is categorized using a classifier. In this paper [1] two type of classifiers are used for classifier optimization which are ”k-nearest neighbour(kNN)” and ”Support Vector Machine(SVM) classifier”. SVM classifier uses GIST features and the kNN classifier uses SIFT features. Various kernels are used in SVM such as Gaussian, Linear, and Polynomial. Feature extraction takes place, forming a similarity matrix. It is given to the kNN classifier. The comparison shows that the SVM classifier is more accurate than the kNN classifier. Coil-20P and Eth80 are the datasets used for the processing.

- *Weaknesses:*

Hand engineering of features is required before object recognition is applied.

Unable to connect Designer and Customers. Collaboration is not possible.

Cannot be viewed from Different angles.

- *How to Overcome:*

Use feature extraction techniques such as convolutional neural networks.

B. The Object Detection Based on Deep Learning

The paper [3] tells about the emergence of the object detection based on deep learning. It reviews the classical method of object detection. The paper [3] states that in deep learning methods the region selection can be achieved using particular strategies, the feature extraction can be done with the help of CNN and the classification achieved by using SVM or a special neural network. The paper reviewed two methods of deep learning namely DNN and Overfeat.

- *Weaknesses*:

Poor real time because of large number of network parameters.
Hard to train

C. Object Detection Using Convolutional Neural Networks

In this paper [5] ”Convolutional Neural Network (CNN)” is used for object detection. It uses an activation function called Rectified Linear Unit. It uses transfer learning which is a powerful deep learning technique in which pre-trained models can be used for feature extraction. It uses the Tensorflow library for high-performance numerical computation. In this paper [5] two models are compared that are ”Single Shot Multi-Box Detector” and ”Faster Region-based Convolutional Neural Network”. SSD with MobileNetv1 is used in object detection because it is lightweight, accurate and small in size and can, therefore, be used in mobile devices.

- *Weaknesses:*

SSD is less accurate

Faster R-CNN has low speed

- *How to Overcome*:

Use SSD when accuracy isn’t the top priority but speed is, as in real-time detection systems.
Use Faster R-CNN where high accuracy is required over speed such as in medical imaging.

D. Application of deep learning in object detection

This paper [2] deals with the application of deep learning in object detection. It gives a summary of some commonly used datasets such as ImageNet, PASCAL VOC, COCO. And create a new dataset for a football game. It gives a summary of the series of algorithm based on R-CNN such as R-CNN, SPP-Net, Fast R-CNN, Faster R-CNN. Faster R-CNN is used because the mAP for Faster R-CNN is 0.732 on the VOC2001 dataset. Using Faster R-CNN on a newly created dataset the mAP for 4 classes is player 0.7902, soccer goal 0.8377, corner flag 0.3508, football 0.4752.

- *Weaknesses:*

Dataset is of uneven quantity. Dataset size is uneven.

- *How to Overcome:*

Use the right evaluation metrics. Resample the training set.

- Under-sampling.
- Over-sampling.

E. Research on face recognition based on deep learning

In this era of digitalization, new technologies are flourished, deep learning is advancing in various areas. Deep Learning is the subfield of artificial neural network (ANN), it consists of different techniques/algorithms encouraged by the construction of the human brain. Handwriting recognition, image recognition, semantic analysis, weather forecasting, marketing predictions, etc uses deep learning. In the given paper [4], it mainly focuses on the complexity in deep learning regarding face recognition and their solutions to improve in the results and accuracy. Deep learning methods, its applicable knowledge along with face recognition for further delve.

- *Weaknesses:*

- How to analyze and review input when understanding the input character after learning is the optimum result?
- How to further enrich the DB resource?

III. PROPOSED SYSTEM

The proposed method is to help the user to categorize the images based on the object and faces into its respected directories. The system will create directories based on an object or user name. The resultant directory will contain the images of the related object or face. The proposed system contains the following module:

A. User Interface

The user would interact with the application using the user interface module. This interface would show all the options available to the user. The options are; search, categorize, copy/move, delete and share. The search and categorize op-tions are used when the user wants to classify the images based on either the objects or faces in them then display as a search result or categorize the images in folders, respectively. An Option flag would be used to choose between search and categorize operations.

B. Face Detection

This module is used to check whether in the input image there is a face or not. If a face is found, then the face flag is set to true and the input image is passed further to the image recognition module. If no face is found, then the image is passed on to continue with object detection. Face detection is achieved using a Haar-Cascade face detector implemented using OpenCV.

Haar-Cascade Algorithm

This module is used to check whether in the input image there is a face or not. If a face is found, then the face flag is set to true and the input image is passed further to the image recognition module. If no face is found, then the image is passed on to continue with object detection. Face detection is achieved using a Haar-Cascade face detector implemented using OpenCV.

The haar-Cascade algorithm makes use of two fundamental and time-consuming. In an image, most of the region is a non-face region. Hence, we check if the window is a face region if it does not then discard that window in a single shot. Don't process it again. Now, how do we achieve this? This is where cascade classifiers come in the picture Cascade classifiers works as follows:

Features are grouped into different stages and stages are applied one-by-one.
At first stage if the window fails, discard it. Don't consider it for remaining stages.
A region is a face region if the window passes all the stages.

E.g. For 6000+ features, divide them into stages with number of features as 1, 10, 25, 25,50 and so on, upto 38 total stages.

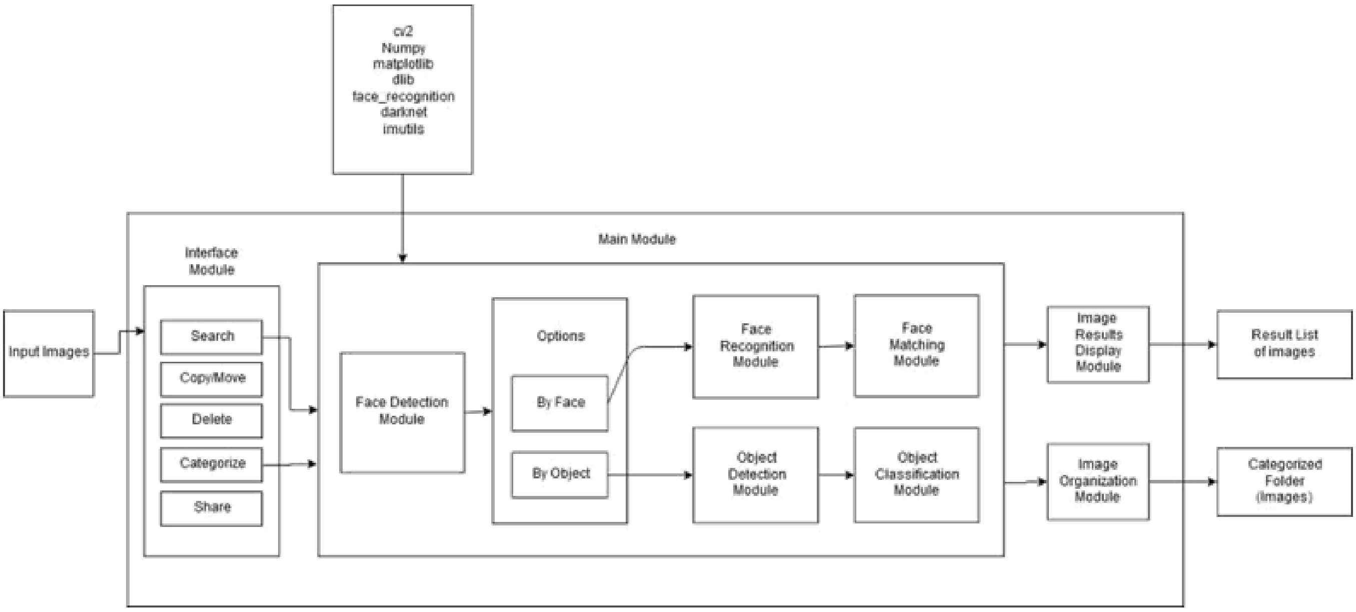


Fig 1:- System Architecture.

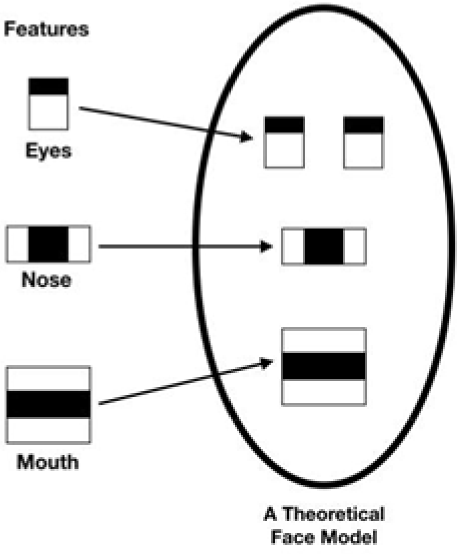


Fig 2:- Haar feature

Components which are Haar features and cascade classifier. This classifier requires lots of images with the face (positive images) as well as images without any face (negative images) to train. Each Haar feature is a single value. There are a large number of features obtained for a single image. However, most features are irrelevant for face detection. To choose the best features, the Adaboost algorithm is used. Here's how Adaboost works:

Apply every feature on all training images.

Find the best threshold value which will classify the image as positive or negative.

Select the features with minimum error rate.

Take the weighted sum of several weak classifiers to get a final strong classifier.

Even with Adaboost, the algorithm is still a little inefficient

C. Face Recognition

The face recognition module is used to recognize the faces in the image that is obtained from the face detection module. After the faces of the persons in the image have been recognized, the module would then assign a unique ID that would be named, to the image. It will then pass the unique person ID to the Face Matching Module as its input. Face recognition is achieved using the LBPH algorithm implemented through OpenCV.

LBPH Algorithm

LBPH stands for Local Binary Pattern Histogram. This algorithm uses 4 parameters:

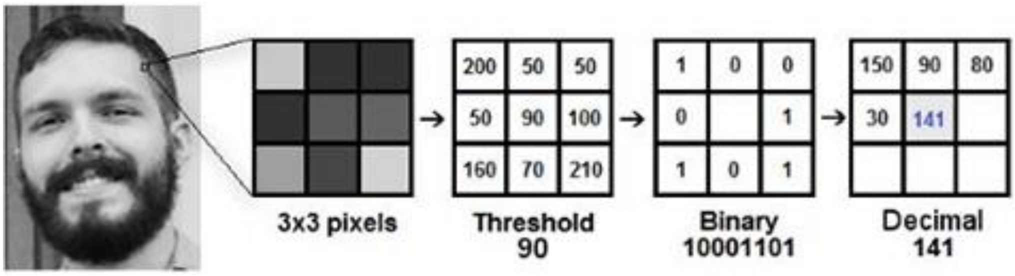
1) Radius: Set usually to 1 pixel.
2) Neighbors: Set usually to 8 pixels.
3) Grid X: Set usually to 8 cells.
4) Grid Y: Set usually to 8 cells.

Steps:

➢ Train the algorithm

- Train the dataset of facial images of people we want to recognize
- Set an ID(number or name) for each image.

➢ Apply the LBP operation



200	50	50
50	90	100
160	70	210

1	0	0
0		1
1	0	1

150	90	80
30	141	

Fig 3:- LBPH Operation

- Use sliding window based upon radius and neigh-bors parameters.
- 3 x 3 pixel matrix with gray levels of each pixel(0-255).
- Set central pixel value as the threshold.
- For each neighboring pixel set values of 0 or 1. If the value of pixel is greater than threshold value then set it as 1, else 0.
- Concatenate each binary value into a new binary value. E.g. 10101011
- Convert the new binary number to decimal and set this as the new value for the centre pixel.

➢ Extract the Histogram

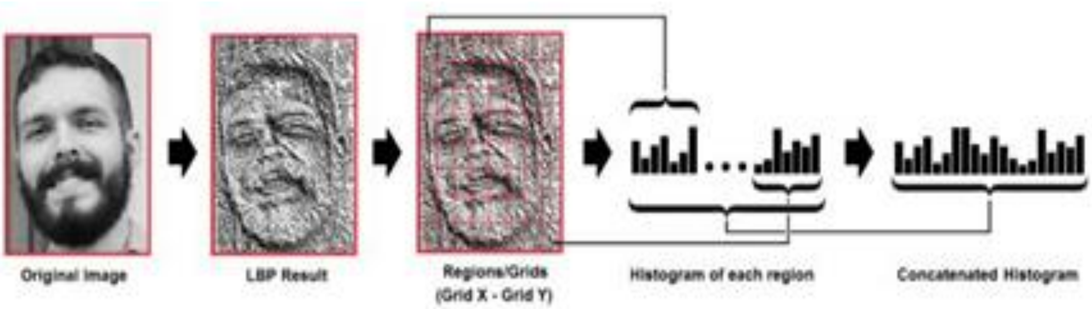


Fig 4:- Histogram Extraction

- Using the image from previous step and the parameters Grid X and Grid Y to divide the image into multiple grids.
- Extract the histograms of each region and con-catenate the histograms to create new histogram. This new histogram would represent the charac-teristics of the original image.

➢ Performing face Recognition

- Compare the input image histogram with the histogram of other images.
- For the comparison, we can use methods such as Euclidean distance, Chi square, absolute value, etc.
- Output is the ID from image with closest his-togram.
- Calculated distance can be used as confidence measure. Note that lower scores mean better confidence, since it is distance. [8]

D. *Object Detection*

The object detection module is the counterpart module to Face Detection module for objects. This module would take the image from the face detection module and apply object detection on the image. This would result in the object IDs of the objects being detected. These object IDs would be then passed onto the object classification module for further operations. The algorithm used for object detection is YOLO, which is implemented in OpenCV.

YOLO Algorithm

YOLO stands for You Only Look Once.

It is an algorithm that makes use of Convolutional Neural Network(CNN) to detect objects.

YOLO sees the whole image at once, unlike other sliding window or region-based techniques.

This helps YOLO get the required contextual information of the image, essentially halving the number of back-ground errors as compared to Fast R-CNN.

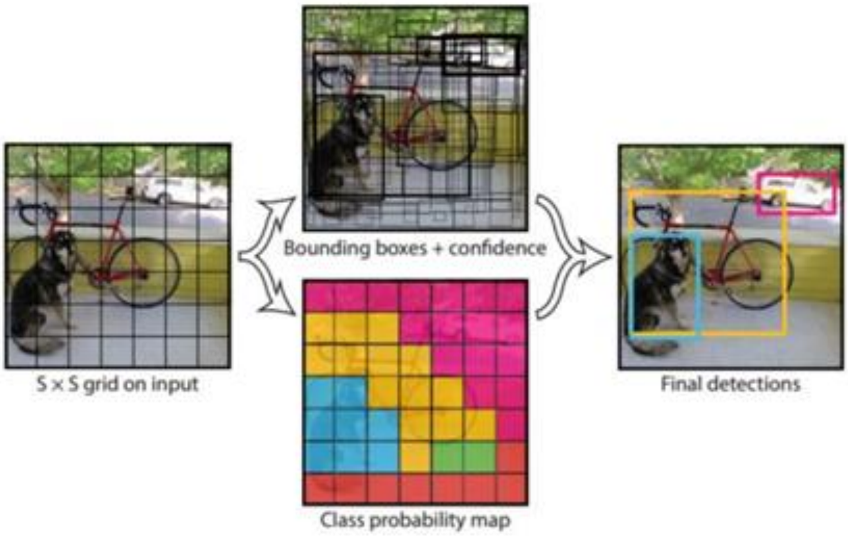


Fig 5:- YOLO Process

YOLO algorithm works as follows

1) Crop the image with equal width and height. For e.g. 416 x 416.
2) Divide the image into an S x S grid.
3) Apply the Convolutional Neural Network.
4) Calculate the predictions of Bounding box with con-fidence score and predictions of class probabilities.
5) Non-max suppression of the bounding boxes.
6) Final image with the bounding boxes, class labels and confidence scores is obtained.

Prediction of class probabilities: Confidence is given as; Confidence = Pr(Object)*IoU where,

IoU = Intersections over Union.
IoU = Highest bounding box overlap/Non-overlap.

Pr = Conditional class probability.

Prediction for Bounding box: noitemsep

– X,Y = Co-ordinates to represent center of box rela-tive to the bounds of grid cell.
– W,H = Width and Height predicted relative to the whole image.
– Confidence: Represents the IOU between the pre-dicted box and the ground truth box.

– Class specific confidence score is given as: Pr(classi)*IOU = Pr(classi—object)*(Predicted Con-fidence)
Pr(classi)*IOU = Pr(classi—object)*Pr(object)*IOU

The YOLO model is then encoded as a tensor as: S x S x (B*5 + C)

where,

C = Class probabilities, B = Bounding Boxes. [6]

E. Face Matching

Face Matching is the module following the face recognition module. This would take the person ID generated by the face recognition module and search through the current directory to find the images of faces with the same ID. If no matching IDs are found, then the problem for the same is returned as a message to the user. If matching images are found, then the current directory path and the filenames of the images are passed to either Image organization module or Image Results Display Module based on the Options flag chosen.

F. Object Classification

Object Classification is a counterpart module to the face matching module. It finds all the matching images with the same object IDs. It then takes the object IDs generated from the object detection module and then classifies them into their respective classes, and each class would be given its class name. These class names and the filenames of the matched images and would be then passed to either Image Organization or Image Results Display modules based on what the Options flag is set.

G. Image Organization

Image Organization module obtains the image filenames and their class names or person IDs from the object classification module or face matching module. It then creates new folders with the class names or persons named as the new folder names. It then moves the images belonging to a certain class or a person to their corresponding folder. The same is then performed for all the classes and persons. The output of this module would then be the newly created folders. This the module corresponds to Categorize operation.

H. Image Result Display

Image Results Display Module is a counterpart to image organization module. This module corresponds to Search op-eration. It takes all the class names or the person ID from Face Matching module or Object classification module, respectively. It would display all the images of a class or a person as a list to the user. The output would be the sorted list of images.

IV. CONCLUSION

We realized the problem of organizing images in bulk i.e sorting, moving, deleting, sharing of images and decided to come up with a system to solve the same.

Our project makes use of new and emerging technologies of deep learning and machine learning to bring a new perspective to sort and group the images.

Finally, this project is fully open source so that other people can contribute their valuable ideas and concepts for this project.

REFERENCES

[1]. Meera M K and Shajee Mohan B S, ”Object recognition in images,” 2016 International Conference on Information Science (ICIS), Kochi, 2016, pp. 126-130.doi: 10.1109/INFOSCI.2016.7845313

[2]. X. Zhou, W. Gong, W. Fu and F. Du, ”Application of deep learning in object detection,” 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), Wuhan, 2017, pp. 631-634.doi: 10.1109/ICIS.2017.7960069

[3]. C. Tang, Y. Feng, X. Yang, C. Zheng and Y. Zhou, ”The Object Detection Based on Deep Learning,” 2017 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, 2017, pp. 723-728.doi: 10.1109/ICISCE.2017.156

[4]. X. Han and Q. Du, ”Research on face recognition based on deep learning,” 2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC), Beirut, 2018, pp. 53-58.doi: 10.1109/DINWC.2018.8356995

[5]. R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra and J. M. Z. Maningo, ”Object Detection Using Convolutional Neural Networks,” TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Korea (South), 2018, pp. 2023-2027.doi: 10.1109/TENCON.2018.8650517

[6]. YOLO — You only look once, real time object detection explained towardsdatascience.com; Manish Chablani https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006

[7]. Face Detection using Haar Cascades opencv-python-tutroals.readthedocs.io; https://opencv-python-tutroals.readthedocs.io/en/latest/py$_t$utorials=py$_o$bjdetect=py$_f$

[8]. Face Recognition: Understanding LBPH Algorithm towardsdatascience.com; Kelvin Salton do Prado https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b

[9]. Special report: How we really use our camera phones gigaom.com; Janko Roettgers https://gigaom.com/2015/01/23/personal-photos-videos-user-generated-content-statistics/

[10]. How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read forbes.com; Bernard Marr https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/45474c160ba9