INTERACTIVE TEDDY

Team: Bhavita Jaiswal, Rawan Alharbi, Surabhi Ravishankar, Tejaswinee Sohoni

I. INTRODUCTION

Interactive Teddy is an application that has been designed for kids. The application will include a camera inside a teddy bear which takes snapshots of objects and people around the baby. The application then classifies it as a person, dollar bill or laptop, and gives an appropriate feedback via audio. This prompts the child to learn to recognize objects around it, and potentially teaches it what to do when he sees a particular object. The application also recognizes the faces it has been programmed to recognize. This differentiation ensures that the child will be alerted when there are strangers around, and will not get friendly with them.

II. DATA SET

For the project we have used two datasets. The first one is the the CalTech dataset that was available online. The dataset consisted of images of various objects like monetary bills, laptops, faces of people, cups, etc. Each category of images consisted of more than forty images. These images were resized and cropped in order to eliminate any background noise while extracting the features. The second dataset that we used was the AT&T face database. The dataset consisted of forty different people with ten images of each person's face. Each of the ten images had various expressions and it takes into consideration facial features and other objects (e.g. spectacles). These images are in a raw format (PGM). After our program worked for the AT&T database and successfully recognized more than one face, we went on to alter the database by using our own images. These images consisted of different facial features and expressions as well. The background was not plain (e.g. there might be objects or colors in the background).

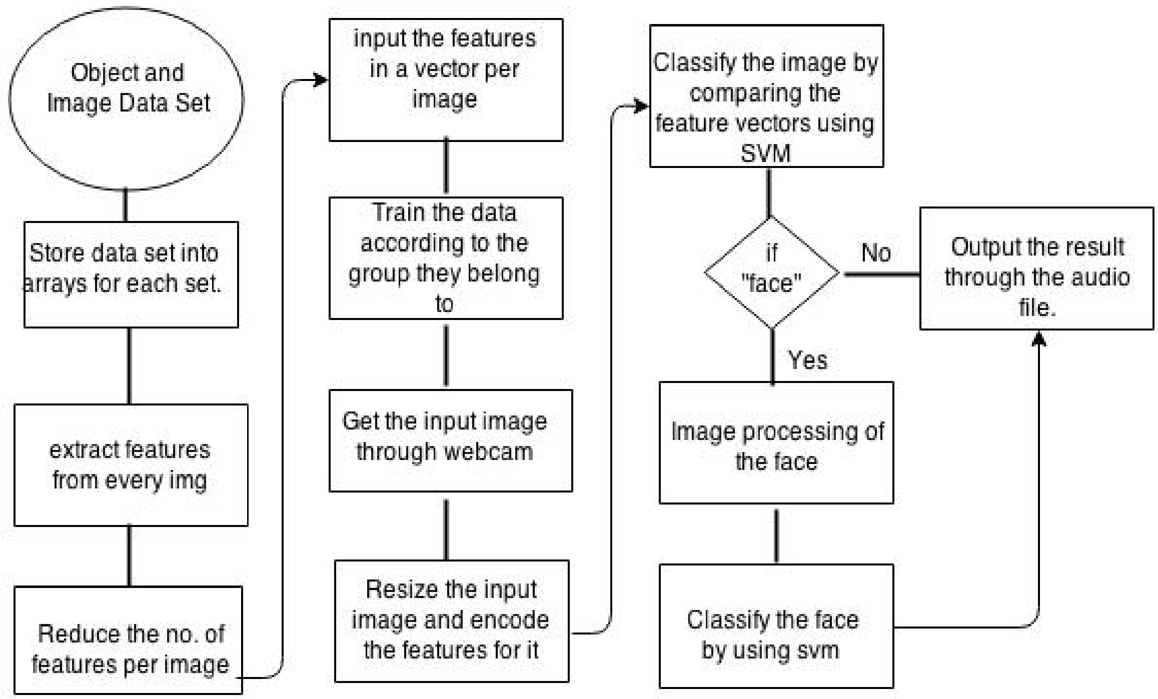
III. RELATED WORK

We started our study by exploring various models for performing object detection. Some of the models that we explored were:

- Feature-based object detection
- SVM Classification with Histogram of Oriented Gradients features
- SVM Classification with Bag of Words models with features such as SURF.

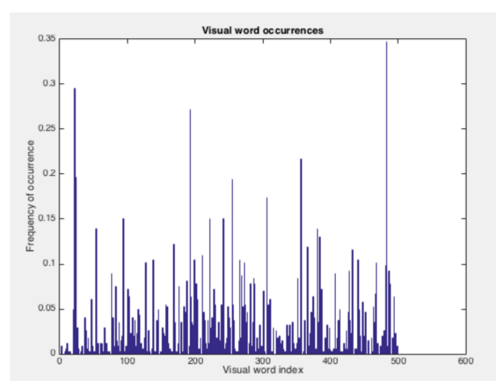
To use SVM on more than two classes we looked at some algorithms like one-vs-all and pairwise. We studied various clustering methods for reduction of the features and dimensions. We analyzed the algorithms that could give us the best results given out dataset. The Bag of Features technique, which represents an image as an orderless collection of local features (e.g. in face representation, local features can be an eye, ear, mouth, etc), overcomes the other methods for object detection. However, in face recognition, classes do not have enough of variations between them, as all images are faces, and thus classes cannot be differentiated properly. Hence, we use Histogram of Oriented Gradients (HOG) technique, instead.

IV. SYSTEM DESIGN



V. IMPLEMENTATION

1. Object Detection Module:



The object detection module consists of the following steps:

1. Importing data into image sets: A collection of image datasets which contain the description and location for each image and number of images.
2. Extracting features:
    - Construct the Vocabulary of SURF features representative of each image category.
    - Keep 80 percent of the strongest features from each Image Set. This is used for training.
3. Reducing features:
    - Balancing the number of features across all image sets to improve clustering.
    - Using K-means Clustering to create a 500 word visual vocabulary.
4. Multi-Class SVM:
    - Since SVM originally works on binary data to classify 2 classes only, we used the one-against-all method to create more than one SVM structures. The total number of SVM structures will be equal to the number of classes that we want to classify. To create these SVM structures, we trained SVM on Set 1 against Set 2 and 3 combined. Then, trained SVM on the Set 2 against Set 1 and 3 combined. Last we trained SVM on the Set 3 against Set 1 and 2 combined.

In order to train the SVM we used an optimization method to calculate the support vectors (s), weights (a) and bias (c).

$$b = a * k(s, x) + c$$ ... where k is the linear kernel

5. Testing:
   - Connect the webcam to MATLAB.
   - Capture the image from the live video stream.
   - Resize the image.
   - Encode the test image into the feature vector that represents a histogram of visual word occurrences contained in that image (Figure 5).
   - Classify the image by looping through the 3 SVM structures and plugging the features of the testing data then output the audio of that class (Figures 2 ,3 and 4).

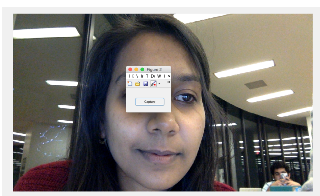


Figure 2: output is "Hello stranger"

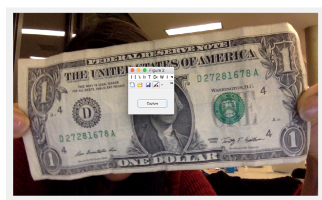


Figure 3: output is "yeah.. Mo money, no problem"

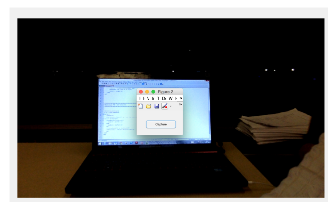


Figure 4: Output is "No no no, don't touch that!"

2. Face Recognition Module:

Face Recognition is the process of extracting features from a training set and performing any kind of classification based on the images.

The face recognition module works in the following manner,

1. Takes the face database as an input.
2. Processes the images.
3. Performs feature extraction.
4. Learns the model.
5. Classify images.
6. Give output.

This is the classic model that was taught in class and we tried implementing the same for face detection.

We used the AT&T face database for our experiment. The face database is explained in Section II. An example set of images that we used is shown in Figure 6.

The most important step that is involved in the entire module is feature extraction. There are various methods of feature extraction like SURF, Histogram of Oriented Gradients, finding the distance from the facial features, etc. The most straightforward way of performing training is by using feature extraction using Histogram of Oriented Gradients (HOG). Also, there is not enough variation between classes if we try using Bag of Features method. Therefore, we used this method. The method is based on the fact that the appearance and the shape of an image can be found using the distribution of intensity and gradients. Further, the image is divided into a number of cells where in the direction of the gradient is computed. Then HOG features are displayed in Figure 7.

Figure 6: Image set that we created.

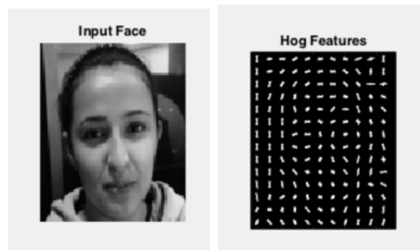


Figure 7 : Histogram of Oriented Features

There are some pitfalls in using this method for feature extraction. The more complicated the image gets, the harder it becomes to extract the HOG features. This leads to the decrease in accuracy. We tried to experiment this with the AT&T database and got about 90% accuracy. After this, we tried to experiment this with a few pictures of us that was taken. The accuracy kept dropping as we increased the database size and tried to train it with a more complicated database.

From the literature survey we learned that the best way to perform classification of more than two classes is to use a technique called binary classification. It uses 3 values (1, -1 and 0) to indicate whether the the learner models belong to the class or not, and stores this in the form of a matrix. The matlab function fitcecoc() performs this for me using SVM.

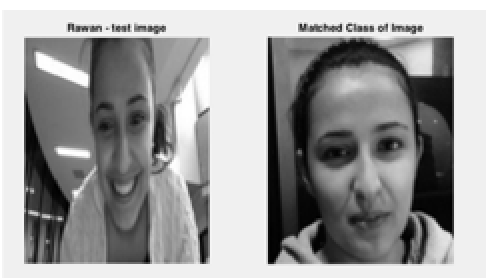


Figure 8: Recognized Rowan Correct

The next step was testing the model, wherein we first read the input image, then extracted the HOG features of the image. Once the HOG features were extracted, we created a variable that would match the class to which the extracted HOG features belonged to. Once the class is determined, the image is displayed along with the class (Figure 8). We got the following image as the final result for the database that we used.

Again, there were a few pitfalls involved in this, mostly because of the accuracy levels of the HOG features and the complexity of the dataset that we used. When we tried to recognize more faces, the accuracy levels kept dropping.

VI. RESULTS

The code will detect the image captured from the Live Video Stream from the Teddy's webcam and if the object matched with the objects of our database it will give the output as an audio depending on which object it is. If the object detected is a face then the face recognition algorithm will run and if the face matches with the face we have in the data (ideally consisting of the faces of family members), it will give the output as the name of the person.

Code: https://github.com/surabhiiyer/CreepyTeddy-
Presentation : https://www.youtube.com/watch?v=6PHNm9Nbwvc&feature=youtu.be

VII. FUTURE WORK

We will add more objects classes that the child encounters frequently. Also, as of now,we are taking the snapshot manually but in the future snapshots should be taken automatically when an object is close enough to the camera.

As for the hardware, we want to capture an image using a wireless camera embedded inside the teddy bear. When the child carries a teddy bear the snapshots will not be as stable as the ones we took using the laptop webcam. This needs to be taken into consideration.

Future work work in terms of the face recognition module revolves around making the code more accurate. There are various ways of implementing face recognition and increasing the accuracy. Some of the methods involve using Eigen Faces and Fisher Faces. The accuracy fell when we used a more complicated dataset (using our own low resolution images with different backgrounds). We will be working on making the feature extraction better.

VIII. BIBLIOGRAPHY

[1] Dardas, Nasser H., and Nicolas D. Georganas. "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques." *Instrumentation and Measurement, IEEE Transactions on* 60, no. 11 (2011): 3592-3607.
[2] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." In *Computer vision–ECCV 2006*, pp. 404-417. Springer Berlin Heidelberg, 2006.
[3]Ericson, Anton. "Object Recognition Using Digitally Generated Images as Training Data." (2013).
[4] L. R. Cerna, G. Cámara-Chávez, D. "Menotti Face Detection: Histogram of Oriented Gradients and Bag of Feature Method".
[5] Rekha, N., and M. Z. Kurian. "Face Detection in Real Time Based on HOG" International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 4, April 2014

http://www.mathworks.com/help/vision/examples/digit-classification-using-hog-features.html?prodcode=VP&language=en
http://www.mathworks.com/help/vision/examples/image-category-classification-using-bag-of-features.html?prodcode=VP&language=en
http://bytefish.de/pdf/facerec_python.pdf