iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** X **Month of publication:** October 2020

DOI: https://doi.org/10.22214/ijraset.2020.31767

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue X Oct 2020- Available at www.ijraset.com

Expression Recognition using Support Vector Machines and OpenCV

Abhishek Dogra[1], Akshita Paul[2], Atanu Chakraborty[3], Dr. Purnendu Bikash Acharjee[4]
[1, 2, 3, 4]*School of Computing Sciences Kaziranga University Jorhat, India*

Abstract: ***Computers were invented as a means of easing the manual activities performed by humans, from simple calculations to performing weather predictions. Every organization and business are in need of automation and are willing to invest a ton of money to bring a little comfort to their customers, all for their satisfaction. Satisfaction can be pointed out by the expression on the person's face. However, businesses tend to organize satisfaction surveys manually, which can be time consuming. Our proposed method aims to use the automation brought about by computers to solve this issue through various methods. It makes use of the power provided by computers to automate the process of surveying customer satisfaction and bring about an easier solution to a once convoluted task. For our method, we're utilizing the facial recognition technology to implement expression recognition and output a result of the expressions. To achieve this, we will be implementing OpenCV in Python and a camera for the vision of the faces. We'll be using Haar classifiers to detect faces and the classification between different expressions is performed by a deep learning algorithm known as Support Vector classifier. The result was a system that could detect faces in video frames, recognize their expressions and label them with an accuracy of 75% as well as provide an output chart of the various expressions and their percentages for the businesses all the while uploading the useful data to a database periodically.***

I. INTRODUCTION

In the present time, a lot of organisations are aware that customer satisfaction is the most important asset they have. However, gathering feedback from customers manually on a regular basis is not always seen as a possibility. The need for customer satisfaction survey stands most important for business organisations in order to understand the customers and improve loyalty. Also, gathering customer satisfaction survey on existing products and services will provide the organisations with the insight to drive future decisions, resulting in a truly customer-oriented business. Earlier, customer satisfaction surveys focused on active listening during one-on-one sessions with customers, which was ineffective, time consuming, and no small feat. The online satisfaction survey next came into focus which brought automation and turned everything from manual to automatic. A set of survey questions is sent out to a target sample and the members of this sample can respond to the questions over the World Wide Web. Customer reviews and ratings help an organisation in the ease of data gathering in minimal costs. But again, this method is not applicable for surveys that require respondents who do not have an access to the internet. Some examples of these respondents include the elderly and people who reside in remote areas. Besides, there also might be an inability to reach a challenging population. To solve these problems, we have developed a simple method that detects and counts the number of people in an image, be it still or real time. This not only just has applications in public gatherings, but also private organizations that might want to keep track of the people attending functions. Further, our proposal also focuses on expression recognition, which can be used to track a person's interest in a particular event, and most importantly, it can perform customer satisfaction survey. We focus on making customer satisfaction survey an important and automatic process, just with the help of expression recognition of the customers at different levels. Perceiving whether someone is sad, happy, or angry by the way he/she turns up his/her nose or knits his/her eye-brows comes naturally to humans. Recognizing and identifying facial expressions are tasks in the realm of perception where, traditionally, human performs better than computers. Or, rather, this used to be the case. Our system, now, can recognize facial expressions and classify them into three categories, i.e. neutral, happy and frowning with the highest possibility of accuracy only by training our datasets in the database. High accuracy allows avoiding false identification. Security levels will be significantly improved, the integration process is easy and flawless, the system will be fully automated and also, time fraud will be excluded.

Overall, we have developed a system that can detect faces, recognize expressions and match them with the existing facial expression data in the dataset in our database.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue X Oct 2020- Available at www.ijraset.com

II. METHODOLOGY

The process is divided into three steps.

1) Facial data extraction
2) Training of the model
3) Expression recognition in a video stream

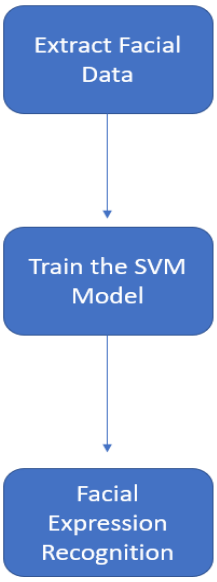


Fig. The three phases

A. Facial Data Extraction

The facial data extraction is required to build the model. The facial data is acquired from a dataset of images that make up the different expressions. For example, images of faces smiling, faces with a frown etc. Each of these expressions serve as distinct classes for the model. The python script glosses over each image in the dataset under the named folder. The name of the folder is used as the name of the class itself, while the images inside the folders "define" those classes.

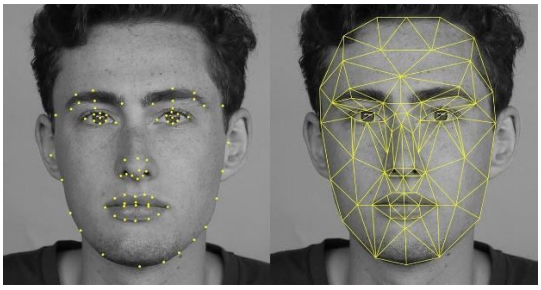


Fig. 2.1 Facial Vectors (Courtesy: Cole Murray)

The figure describes the type of data that is extracted from the images. The data is stored in a serialized manner under the class label.

B. Extraction of face Embeddings

The python script for this step requires the following arguments:

1) Dataset: The path to our dataset.
2) Embeddings: The path to our output embeddings file. The script will extract the embeddings and store them serially on the file.
3) Detectors: The path to OpenCV's deep learning face detector model. This will actually localize the face in the image.
4) Embedding Model: Path to the OpenCV's deep learning embedding model. This will allow us to extract a 128-D vector.

The script quantifies each image present in the dataset directory. The label of the directories inside the dataset are used to represent the person that is recognized using the set of the images present within the particular directory.

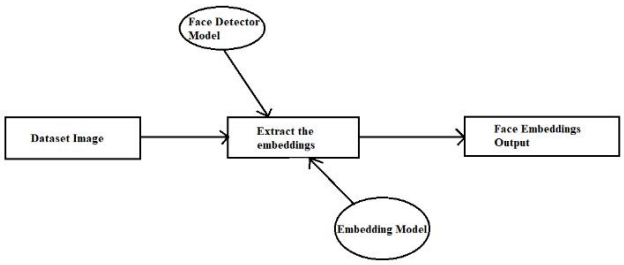


Fig. 2.2 Facial Data Extraction

C. Training of the Model

The classification medium used in the method is a simple Support Vector Classifier provided in the SCIKIT-LEARN library of Python. The SVC is trained to recognize an expression in the frame and label is accordingly. The labels and the classification come from the pickle file generated in the previous step.

The python script for this step requires three arguments:

1) *Embeddings:* The path to the output embeddings file from the previous step. The file now has embeddings from the faces in the dataset.

2) *Recognizer:* This will be our output model which will be used to recognize expressions. It is based on SVM.

3) *LE:* This will be the label encoder. The labels of the face directories will be now be used for face recognition. The name of the directory will be the name of the expression detected.

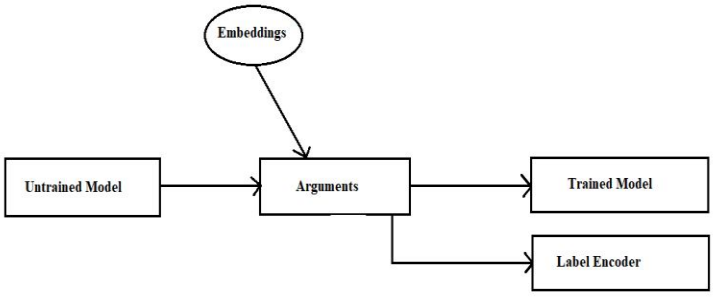


Fig. 2.3 Training the model

D. Expression Recognition

In this phase, we apply OpenCV and deep learning:

1) Detect Images

2) Compute embeddings to quantify a face

3) Train a support vector machine (SVM) on top of the embeddings

4) Recognize faces and their expressions

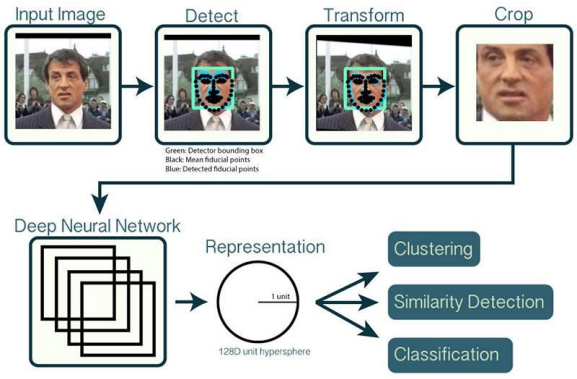


Fig. 2.4 An overview of the workflow of OpenCV's Face recognition

The python script responsible for this phase turns on the concerned camera and starts taking input frames. Each frame is processed one by one. The speed of frames being processed is an average of 25 frames per second, which varies from system to system.
Each frame is treated as an individual image and the processing is similar to the extraction of facial data part. Faces are detected in each frame and the face with the highest confidence has its ROIs extracted. The classification model trained in the last phase is loaded into the script and a comparison between the current data and the model's data is performed. The end result is the classification of expressions that is labelled on top of the person's face using an image write function in OpenCV.
The python script requires a connected camera to the system. It can be a camera module for Raspberry Pi or a webcam for a desktop. The VideoStream() method is used to capture frames from the camera and FPS method gives us the current FPS.
The python script will use four arguments for it to work:

a) Detector: The path to OpenCV's deep learning face detector. It detects the face ROI (Regions of interest).
b) Embedding Model: Path to OpenCV's deep learning face embedding model. We use this to extract the face embeddings.
c) Recognizer: The path to the recognizer model that was trained in the previous step, which was an SVM.
d) LE: The path to our label encoder. This contains the labels like 'happy'.

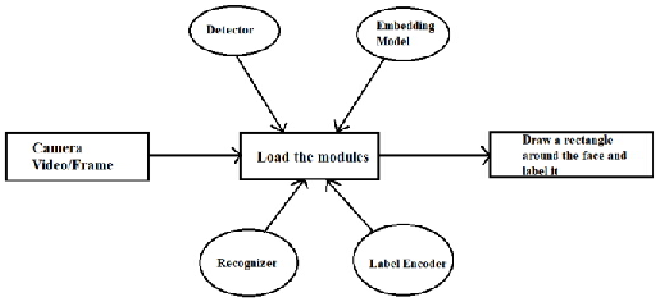


Fig. 2.5 Work-flow of the expression recognition process

All of these scripts are executed one-by-one. Once the model is created, there's no need to execute the first two scripts again, unless, there's new dataset being added to the previous. The data extraction has to be executed and the model has to be rebuilt in case new data is added. After the satisfactory execution of all the scripts, pressing the quit key will generate a pie chart based on the expressions. The data of the chart will also be added to a database to maintain it. This allows the method to be used as a survey for customer satisfaction that can be employed by restaurants, malls or a business of any type.

III. RESULTS AND DISCUSSION

The method was tested with a dataset of 500 images per expression. The minimum threshold was set to 75% and it was constantly being crossed, i.e., the accuracy was 75% at most of the time. Accuracy can be increased furthermore if the dataset is increased or if some tweaks are made to the methods.
Below are a few test cases for the expression recognition.
The images contain multiple text indicators. There are the three different expression counts, maximum number of faces counted, number of faces currently in the image, a unique id for each face in the frame, the current emotion and a frame count. The frame count is a simple counter which, when it reaches 100, uploads the current expression counts and number of faces into the database for studying.

Fig. 2.6 Test Case 1

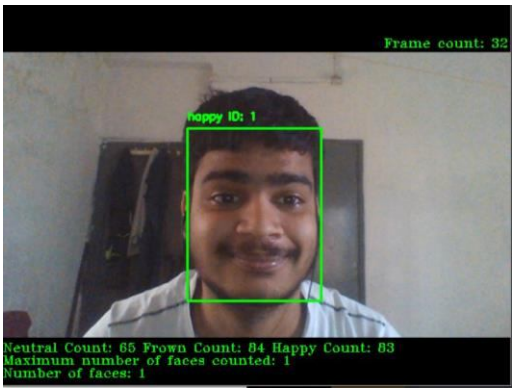


Fig. 2.7 Test Case 2

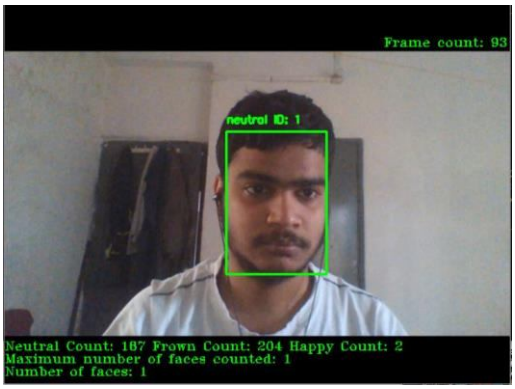


Fig. 2.8 Test Case 3

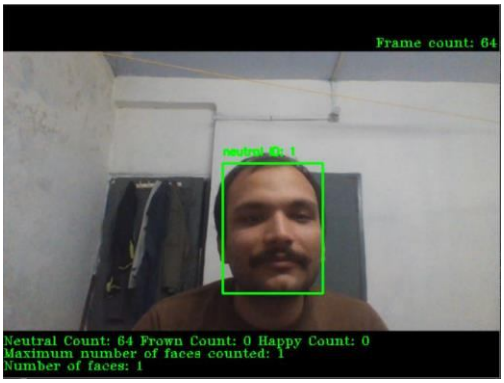


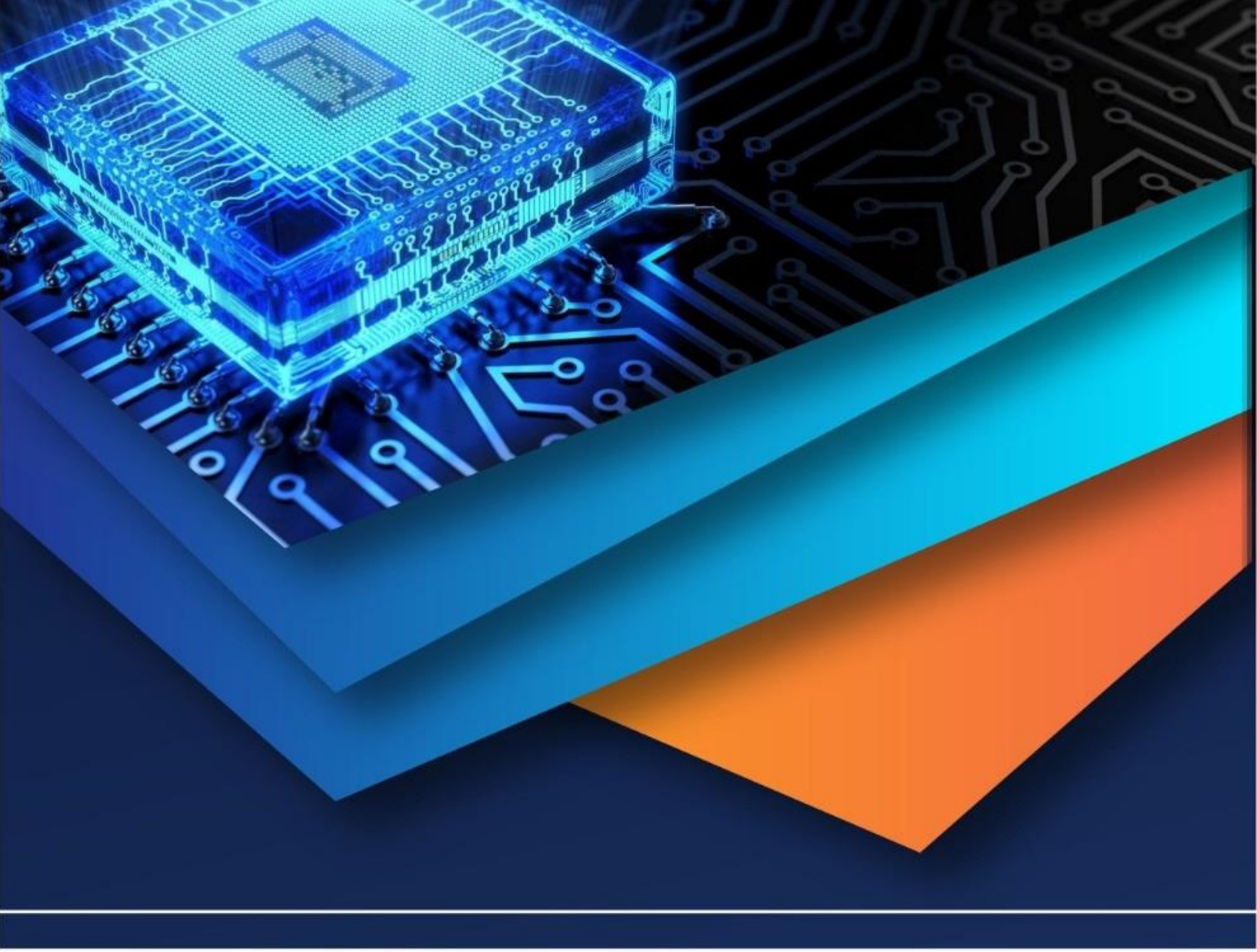
Fig. 2.9 Test Case 4

IV. CONCLUSION

In this paper, we proposed a novel method for facial expression recognition. The method made use of OpenCV and SVM to achieve the goal. The method achieved expression recognition which was nigh instantaneous with an accuracy of 75% per expression. Haar Cascades for face localization in an image were a significant contributor to the speed of the whole process and with further optimizations can be made even better. The facial recognition modules provided in OpenCV were made for exactly that but a slight change to the dataset and modules turned it into an expression recognition system. It made use of a linear kernel SVC which provided more speed albeit with a slight but negligible loss to accuracy. The data provided by the application at the end is in a pie chart form but is easily convertible to different other forms when required. It is also uploaded constantly to a database for easier access and survey. The end result is a system that is able to accurately distinguish between different facial expressions using easy-to-understand methods and algorithms and provides an efficient way to perform satisfaction surveys for businesses to know what keeps their customers satisfied

REFERENCES

[1] Philipp Michel, Rana El Kaliouby, "Real time facial expression recognition in video using support vector machines", ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces, November 2003, Pages 258–264, https://doi.org/10.1145/958432.958479
[2] Dr. Shaik Asif Hussain, Ahlam Salim Abdallah Al Balushi, "A real time face emotion classification and recognition using deep learning model", Journal of Physics: Conference Series, ICE4CT 2019, doi: 10.1088/1742- 6596/1432/1/012087
[3] Chandan G, Ayush Jain, Harsh Jain, Mohana, "Real time object detection and tracking using deep learning and OpenCV", Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2018), IEEE Xplore Compliant Part Number: CFP18N67- ART, ISBN: 978-1-5386-2456-2
[4] Shah, Parin M., "Face Detection from Images using Support Vector Machines" (2012). Master's Projects. 321. DOI: https://doi.org/10.31979/etd.wg5s-gyqn
[5] Abhishek Dogra, Akshita Paul, Atanu Chakraborty, Nitish Kumar Boruah, "Expression Recognition and Satisfaction Survey", Kaziranga University, Jorhat.
[6] T.S. Hai, L.M. Triet, L.H. Thai, N.T. Thuy, "Real time burning image classification using support vector machines", EAI Endorsed Transactions on context-aware systems and applications, 2017, doi 10.4108/eai.6-7-2017.152760.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY