Loyalty customer detection and personalized offers using age and gender prediction implemented using Dlib, Keras, Raspberry Pi and Apache Kafka

Sreekiran A R

***Abstract*—This paper presents an approach to find the loyalty customers in brick and mortar stores without compromising their identity, by assigning them system generated ID's and provide them with personalized offers, suitable for their age and gender using camera based data. Here, a raspberry Pi camera is used as input sensor which can be placed at the bill counters, where it sends high quality video to a high performing server via Apache Kafka [1], where the algorithm checks for the detected faces and see if they match the training data, if match is not found, append the training data with a new ID, the customer's age and gender also will be predicted and the data along with timestamp is sent to a Kafka server and from there to an SQL database which can be used for visualization. The same algorithm can be applied for personalized digital signage, can be used to track people in the store by linking multiple cameras. Thus, this solution has an impact on customer retention, smart advertising and people tracking, also it helps the supply chain for smart inventory management by getting insights using age-gender bucket data.**

I. Index Terms

Computer vision, age and gender prediction, operationalize IoT, Apache Kafka, facial detection and tracking, transfer learning, Raspberry Pi, deep learning, retail solutions.

II. Introduction

IoT technologies are being relied on by retailers to catch shopper's eyes and increase their sales in the process. The market for IoT retail solutions is slated to grow by approximately 20% in the coming year. The main reason for this increased demand is the digital signage as retailers seek new ways to keep their sales growing. Micro-processors and ubiquitous Internet connectivity has resulted in a technology transition in the retail world. Large number of IoT retail devices are being deployed and connected, and they have the ability to offer reports on various customer habits. Some of the features of IoT in retail is inventory tracking and management, mobile payments, product enhancements and customized marketing.

According to studies done by Bain & Company [2], increasing customer retention by 5% can lead to an increase in profits of 25%-95%, and the likelihood of converting an existing customer into a repeat customer is 60%-70%, while the probability of converting a new lead is 5%-20%, at best and giving them personalized services is one of the efficient ways to retain customers. However, in many cases, companies are unable to identify loyal customers, let alone provide them with suitable offers. This may be because the customer either provide different contact information or may not be willing to give at all.

Computer vision plays a big role in providing friction-less retail experience for the customers, but as the use of cameras becomes pervasive, physical retailers will have to be transparent in their data collection practices, respect individual's security and privacy concerns.

In this paper, we are trying to build a solution for finding loyalty customers in brick and mortar stores without compromising their identity, by assigning them system generated ID's and provide them with personalized offers, suitable for their age and gender using camera based data. Here, a Raspberry Pi is used as input sensor, the Kafka producer, which encodes and sends high quality video to an Apache Kafka server, which is then received and decoded by a high performing system, the Kafka consumer, in real-time where the received images are passed through a facial detection algorithm which detects all the faces and passes all the faces in the given threshold size to a facial recognition model using Dlib [3] and age and gender prediction model using Keras. In the facial recognition model [4], initially, the training data will be null, so the first face passed will be assigned a unique ID and their encoded face features will be appended to the training data. The ID along with their predicted age and gender will be sent to different Kafka topic along with timestamp, which is received by another server and stores it in MySQL database which is later used for UI development.

The same algorithm can be applied for personalized digital signage, display advertisements according to the age and gender of the customers in front of the monitors, can be used to track people in the store by linking multiple cameras and thereby get many insights like customer dwell time, customer path, heat map etc. Thus, this solution has an impact on customer retention, smart advertising and people tracking, also it has an impact on the supply chain wing by providing smart inventory management by getting insights using age gender bucket data.

III. Functionality

The functionality of many computer vision solutions involves a data extraction part, an object detection part and a classification part. Here, both the facial recognition and age and gender prediction models also goes through these steps to get the desired output.

Sreekiran A R is with Innovation and Engineering, Umiya Business Bay, Tower 2, Bridgei2i Analytics Solutions, 2nd Floor, Cessna Business Park, Kadubeesanahalli, Outer Ring Road, Bengaluru, 560103, Karnataka, India, Tel.: 080674 22100, e-mail: sreekiranar@gmail.com

A. Functions

The input camera data from the Raspberry Pi is encoded and send to the Kafka server, which is then received by the deep learning sever in which our solution algorithm is deployed. The solution decodes the image data and executes the below steps frame by frame.

•Face detection: The primary step is to detect the faces in the data which is done with the help of OpenCV cascade classifier.
•The detected faces are cropped and passed through two algorithms; Facial recognition and age and gender prediction.
•Face recognition algorithm : Each face is passed through a facial recognition algorithm with the help of Dlib library in Python and initially, the training data will be null. When the first face comes, it will be assigned the first ID and now whenever a new face is coming it will be first compared with the training data and if a match is found, the ID of the matched face is returned or else, a new ID is given. If the server was shut down and algorithm has to restart, the code will check for the previous data saved in the hard disk and load the training data with face encoding and IDs and continue so that no data is lost.
•Age and gender prediction : A deep learning model using WideResNet architecture trained with 500K+ IMDB data set is used to predict the age and gender of the detected face.
•Finally, the detected face ID, age and gender will be sent to another Kafka producer topic, along with time stamp.
•The output is then received from Kafka and stored in MySQL database for UI purposes. We can get insights from the data about number of customers, age and gender buckets, number of repeated customers etc using a single camera, customer dwell time, customer tracking, heat map etc. linking multiple cameras.

The algorithms are further explained in the next section.

B. Working Algorithm

As mentioned above, the working of the algorithm in the three phases are explained below.

1) Data Extraction: The first step of this solution is to start an Apache Kafka Server for real-time data ingestion. This is done with the help of Spotify-Kafka docker. The next part of algorithm is executed in the Raspberry Pi system. Here, the images are captured from the camera at 12 frames per second at 1280x720 resolution with a 5MP camera with the help of OpenCV2.4.10 library in python. The data rate is limited to 12 frames per second due to the hardware limitation of deployment of algorithm in real-time. The obtained image is first converted to black and white as color is not required for face detection and classification and also for the faster transfer of data, it is further encoded to string to make it one dimensional and then to bytes for transferring the image using Kafka Producer to a specific topic. Thus, the data is transferred to a Kafka topic and any authorized person can access the data real-time.

2) Face detection: This part can be done using methods like Histogram of Oriented Gradients [5]. In our case, we are using the OpenCV default function HaarCascade face detector function to detect multiple faces in an image. The detected objects are then cropped by giving an additional 10% from the obtained coordinates to account for the whole face and then each cropped image is passed for further classification.

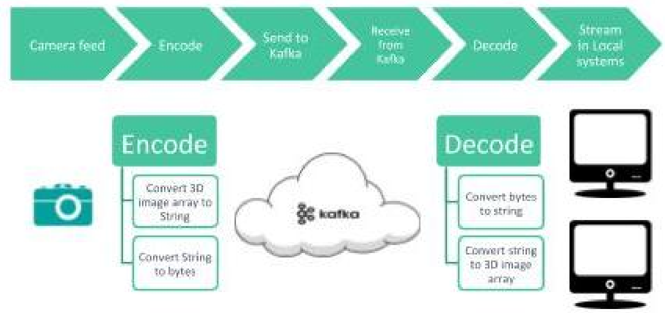

Fig. 1. Work flow of data extraction

3) Classification: The cropped faces are now passed through face recognition algorithm and also age and gender prediction algorithm.

a) Facial recognition: The cropped images are first aligned using face landmark estimation. There are lots of ways to do this, but we are going to use the approach invented in 2014 by Vahid Kazemi and Josephine Sullivan [6]. The basic idea is we will come up with 68 specific points (called landmarks) that exist on every face — the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then we will train a machine learning algorithm to be able to find these 68 specific points on any face. Now that we have aligned the face according to the landmarks, next step is to extract features of the face. Feature creation is done with the help of neural networks which will convert each face into a vector of 128 measures called embedding. The exact approach for faces we are using was invented in 2015 by researchers at Google [7] Adamam but many other approaches exist. Once the measurements are obtained the first face array will be saved as a list of array and an ID will also be saved with the same index. Now, when a second face comes and goes through all these steps and the embedding is extracted, first, it compares it with the first face array and check for match using matrix distance and if the second face is not close to first face, it will be appended to the training data, else it will return the face id as first image as it is recognized by the system as the same person. Thus, with just one image of the person, the system is able to recognize that person with good accuracy. Thus, we got the ID for the customer, next step is to predict the age and gender.

b) Age and Gender prediction: In this model we are using pre-trained weights for a neural network with WideResNet architecture [8], first, we convert the cropped faces into 64x64 resolution as the model was trained with an input requirement in that resolution. The model is a classification model with 100 classes age 1-100. The model gives the age with the highest probability and also there is a binary classifier which decides the gender. The weights for the model was obtained by training from 500K images from the IMDB data set. Thus, now we have the face ID, age and gender of the given face. Now we send the data along with the current timestamp to a different

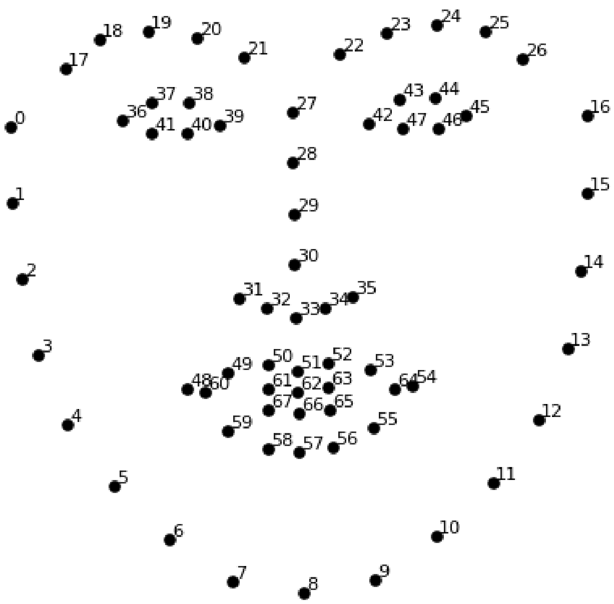

Fig. 2. The 68 landmarks we will locate on every face. This image was created by Brandon Amos of CMU who works on OpenFace.

Kafka topic as our output. Thus, the output is also in the Kafka server which can be accessed by authorized person in real-time. The output is then stored to a MySQL database for UI purposes.

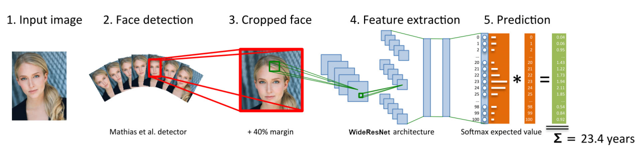

Fig. 3. Working of neural network in age prediction

IV. HARDWARE & SOFTWARE

Apache Kafka server is set up on a dedicated system to ensure the availability of the system 24x7 with specification 8GB RAM, 4 Cores and 100GB storage. The Kafka server is hosted using Spotify Kafka [10] docker and is adjusted to have a maximum data retention period of 1 hour for the topic to which video is sent and a maximum data retention size of 2 GB is given so as not to prevent crashing of server due to insufficient space. The Kafka producer and consumer is invoked using Kafka-python [11] 1.3.5. We are using a 5MP camera module for Raspberry Pi with capacity of 1080p30 Video connected to a Raspberry Pi 3model B+, 1.4GHz 64- bit quad-core processor, dual band wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and Power over-Ethernet support (with separate PoE HAT) with Raspbian [12] OS as our platform for getting camera data. All the systems are connected to same network using 100Mbps LAN cable. The connected camera is accessed using Python libraries PiCamera and OpenCV2.4.10. The algorithms are deployed in a deep learning server which has 20 Core, 120 GB RAM and 2TB hard disk which also stores the training data. All the algorithms are made in Python 2.7 platform. The received image in bytes from Kafka is decoded using OpenCV3.3.0 and also face detection is done using HaarCascade multi detect function of OpenCV [13] and the face data is aligned and converted to a 128 dimensional vector using library Dlib. The deep learning network used for age and gender prediction is done using library Keras [14] 1.3.5 and tensorflow [15]1.5.0. The obtained result is stored in MySQL server using mysql-connector [16] 1.1.6. All the platforms except Raspberry Pi and UI server uses Ubuntu 14.04 OS. The UI is still a work in progress so the versions and specifications is avoided.

V. RESULTS

Thus, the solution gives a basic output of timestamp, face ID, age and gender which is regulated every frame. This can be used to get the gender ratio, different age buckets of customers. The first occurrence of every unique face will be saved in a folder to see the unique customers, and each occurrence of that particular face will be saved in a folder named that particular face ID. Thus, we can find dwell time of the customers and also track a person, if connected to different cameras.

TABLE I
SAMPLE OUTPUT TABLE

timestamp	age	gender	face_id
2018-03-09 16:14:46	23.3692	M	0
2018-03-09 16:14:47	20.2888	M	0
2018-03-09 16:14:48	20.9699	M	0
2018-03-09 16:14:49	21.4732	M	0

Here, the screenshots of real-time execution of solution is attached. For showing the impact of the solution, a demo video of 2 minutes 42 seconds duration is passed instead of live video. As a result, 126 faces were detected and the age-gender distributions are shown below.

VI. CONCLUSION

In this solution, we have implemented real-time data streaming using Apache Kafka, Facial detection and recognition, age and gender prediction using deep learning and transfer learning.

This solution makes an impact for identifying regular customers and retain them by providing them with unique offers specific to their age and gender bucket without compromising their identities.

This solution can be further improved in the future to get customer customized digital signage, dwell time, track customers, get heat map of the store and thus provide massive impact in creating a frictionless shopping experience for customers in brick and mortar stores.

ACKNOWLEDGMENT

This work would not have been possible without the immense support and guidance of Jatan Sharma, Manager, Innovation and Engineering Department and Kirti Ranjan, Manager, Innovation and Engineering Department, who guided me in the right path and inspired me to achieve this result.

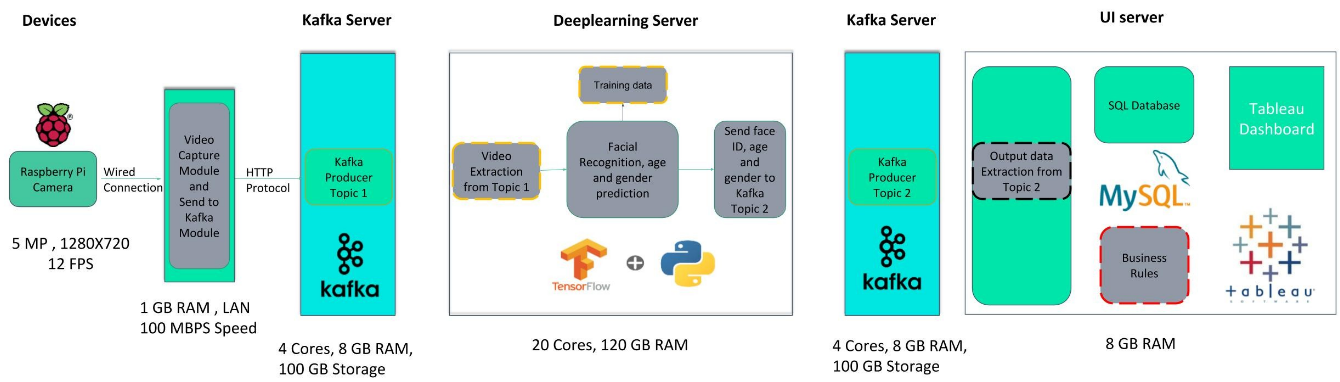

Fig. 4. Architecture/ Workflow of solution

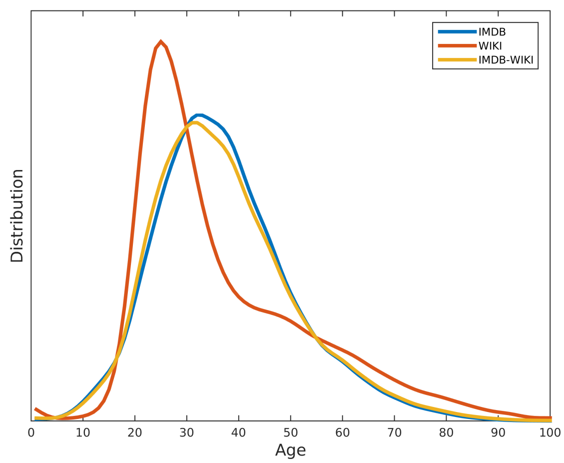

Fig. 5. IMDB-WIKI [9] data set distribution in which the age prediction was pre-trained, 460,723 from IMDB and 62,328 from Wikipedia

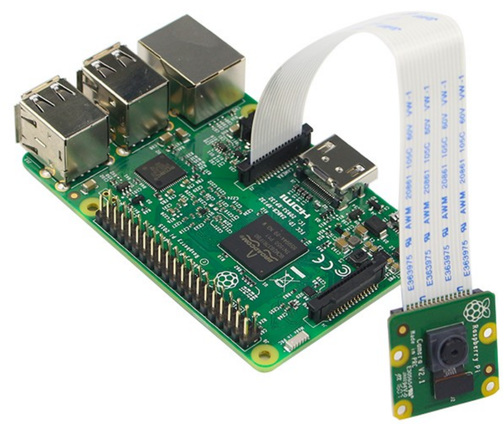

Fig. 6. Raspberry Pi 3 model B+ with 5 MP Camera Module support.

I am indebted to Vaibhav Singhal, Sri Harsha, Swapnil Singhal and Arpan Naik who were part of implementing this solution.

I am also grateful to my brother, Sreeragh A R, Software Engineer, Qburst for providing me guidance and unending inspiration.

REFERENCES

[1] “Apache Kafka.” [Online]. Available: http://kafka.apache.org/
[2] “Value of Keeping Right Customers.” [Online]. Available: https://hbr.org/2014/10/the-value-of-keeping-the-right-customers
[3] “dlib,” dlib. [Online]. Available: http://dlib.net/
[4] “Facial recognition,” https://towardsdatascience.com/facial-recognition-using-deep-learning-a74e9059a150.
[5] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection.” [Online]. Available: http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf
[6] V. Kazemi and J. Sullivan, “One Millisecond Face Alignment with an Ensemble of Regression Trees.” [Online]. Available: http://www.csc.kth.se/~vahidk/papers/KazemiCVPR14.pdf
[7] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering,” in *Computer Vision Foundation*. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/app/1A_089.pdf
[8] A. Geitgey, “Age and gender prediction.” [Online]. Available: https://www.dlology.com/blog/easy-real-time-gender-age-prediction-from-webcam-video-with-keras/
[9] R. Rothe, R. Timofte, and L. V. Gool, “DEX: Deep EXpectation of apparent age from a single image Rasmus Rothe, Radu Timofte, Luc Van Gool Computer Vision Lab.” [Online]. Available: https://www.vision.ee.ethz.ch/en/publications/papers/proceedings/eth_biwi_01229.pdf
[10] “Spotify-Kafka,” https://github.com/spotify/docker-kafka.
[11] “Kafka Python,” https://github.com/dpkp/kafka-python.
[12] “Raspbian,” https://www.raspberrypi.org/downloads/raspbian/.
[13] “Open CV,” http://opencv-python-tutroals.readthedocs.io/en/latest/.
[14] “keras,” https://keras.io/.
[15] “Tensorfllow,” https://www.tensorflow.org/.
[16] “mysql.” [Online]. Available: https://dev.mysql.com/doc/connector-python/en/

Sreekiran A R One year Experience in working with computer vision, IoT and Big data as part of Innovation and Engineering department at Bridgei2i Analytics Solutions.

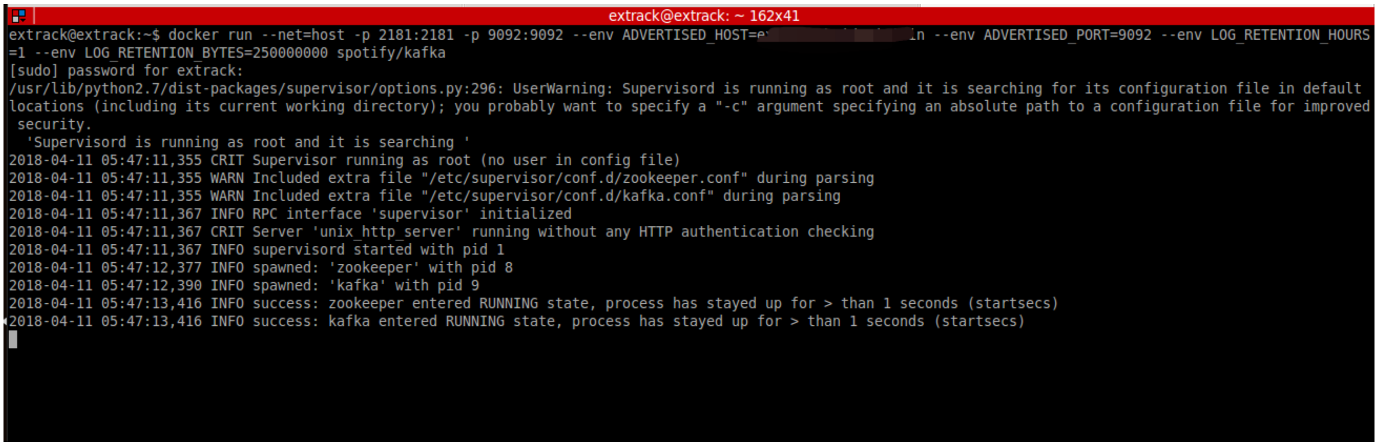

Fig. 7. Apache Kafka docker started

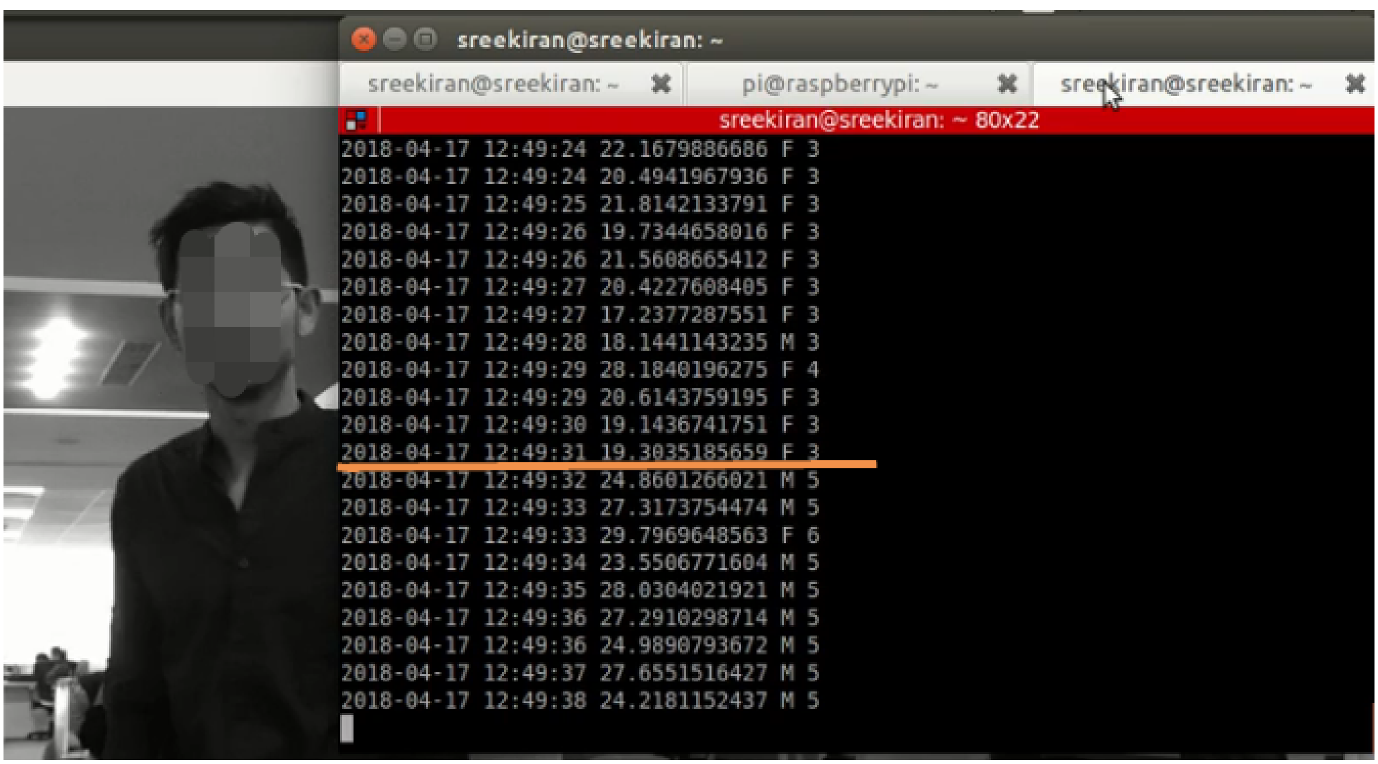

Fig. 8. Algorithm running in real-time. Face is detected as Male with approx. age 25 and given an ID 5

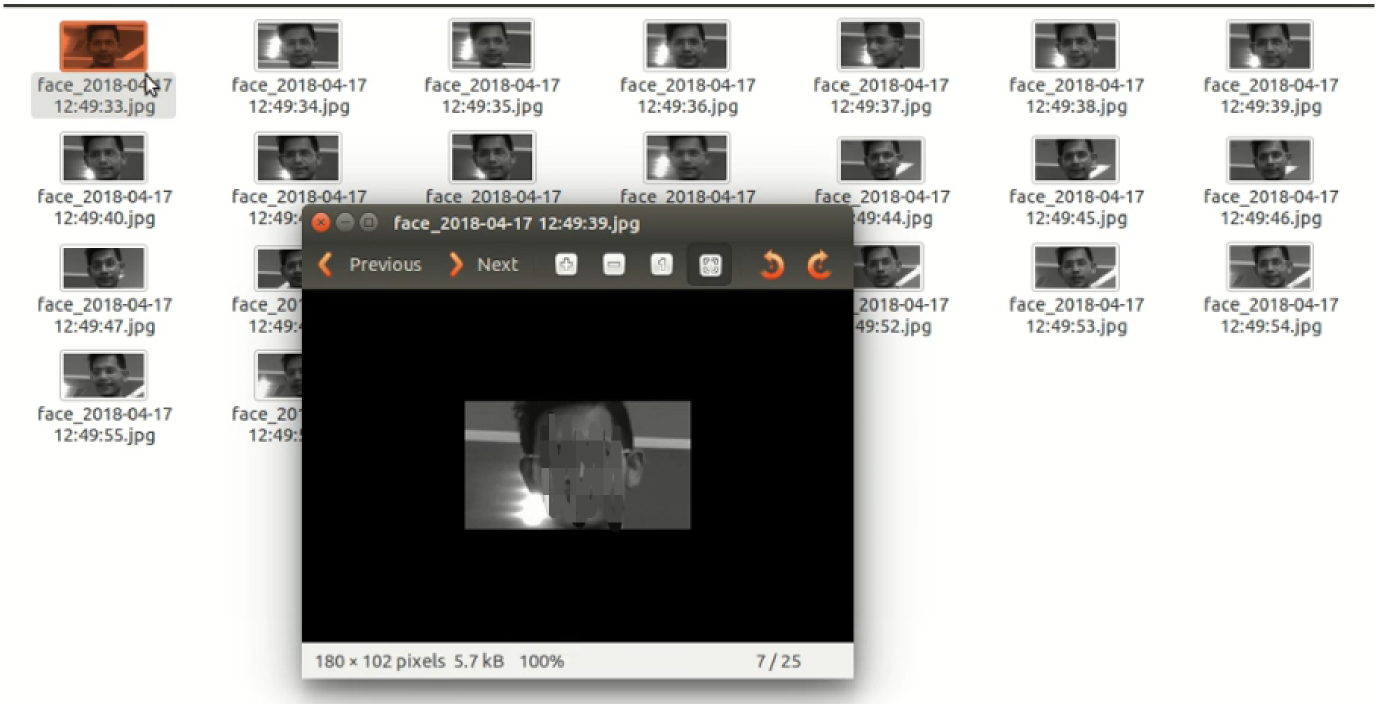

Fig. 9. Output folder of ID 5, all the occurrence of ID 5 is recorded with timestamp.

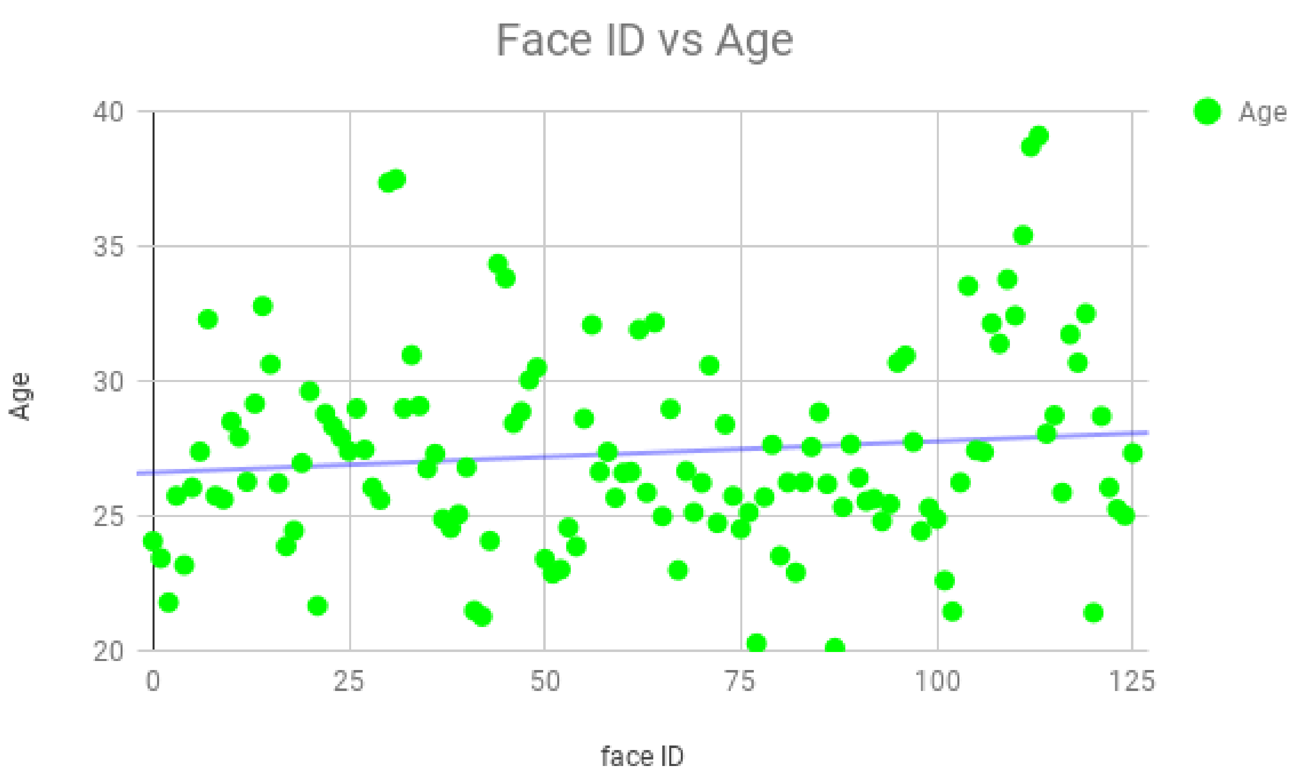

Fig. 10. A total of 126 faces were detected and the age distribution is plotted.

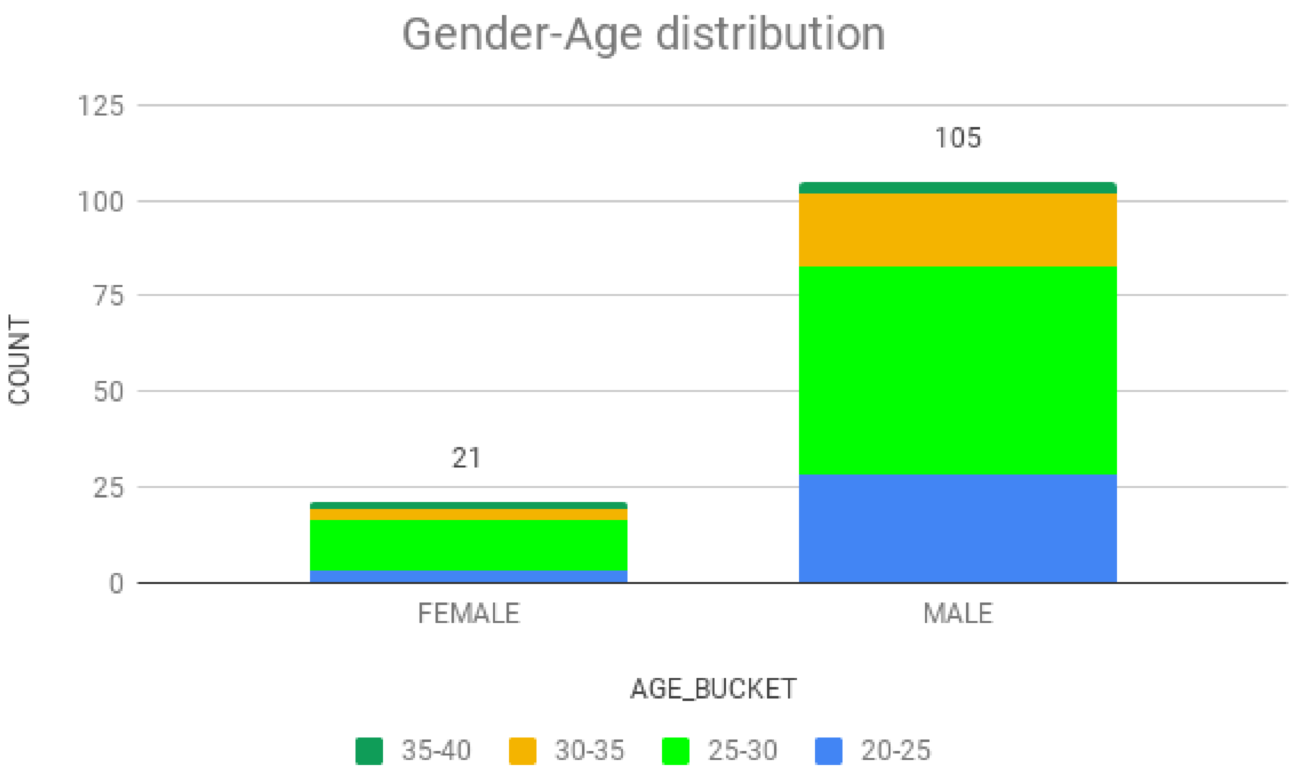

Fig. 11. Gender-Age distribution of output