Evolution in Electrical and Electronic Engineering Vol. 4 No. 1 (2023) 477-486

EEEE

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/eeee
e-ISSN : 2756-8458

Face Recognition Attendance System

Surenthiran Segran[1], Mohamad Hairol Jabbar[1]*

[1]Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Parit Raja, 86400, Johor, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/eeee.2023.04.01.056
Received 17 January 2023; Accepted 04 February 2023; Available online 30 April 2023

Abstract: One of the most effective image processing applications is face recognition, which is essential in this technical world. For the purpose of verification, particularly in the context of student attendance, the recognition of the human faces is an important concern. Attendance system using facial recognition is a step of identifying students by using face biometrics based on high-definition monitoring and other computer technologies. The implementation of this system aims to digitally transform the established method of recording attendance by calling names and keeping handwritten records. The methods used today to take attendance are difficult and time-consuming. Manual recording of attendance can be used to simply modify records which will lead to attendance proxy. This is the main reason to develop the facial recognition attendance system. The developed project uses Raspberry Pi as the main processing unit with Raspberry Pi camera module. This developed project uses a cloud computing platform (Microsoft Azure) for face services and to retrieve data from the database. For the database of the student, SQLite were used in the developed project. The developed project was tested in terms of face detection, face recognition and attendance marking. The developed project was able to detect, recognize and mark the attendance of the student up to 90% accuracy.

Keywords: Face Recognition, Microsoft Azure, SQLite

1. Introduction

Attendance is crucial in administration but can be tedious and prone to errors. The traditional method of verbally calling names and recording them can be ineffective, particularly in large classes. To address this issue, some organizations have implemented document-based systems or digital methods such as biometric fingerprinting or card scanning [1]. While these solutions may improve attendance tracking, they can also be limiting as they can be slow and require students to carry identification cards. With the advancement of technology, a newer biometric approach, facial recognition, has been proven to be an effective method for taking attendance [1].

Corresponding author: hairol@uthm.edu.my

The Automated Attendance System utilizes facial recognition technology to automatically identify whether students are present or absent in the classroom. One of the key aspects of this system is the use of deep learning, which allows the system to train itself using a dataset and produce accurate results through various learning methods. Attendance is considered a vital aspect in academic environments, both for students and instructors. With the advancements in deep learning technology, the system can now also monitor students' attendance performance and keep track of the collected data. However, previous facial recognition attendance systems had some limitations such as misidentification of faces, camera placement and lighting. The proposed project aims to address these limitations to develop a more accurate facial recognition attendance system [2].

This proposed project comes with several objectives. The first objective of the project was to design an attendance system based on face recognition. The second objective of the proposed project was to build a prototype of the system and the third objective is to test the system in terms of detection and attendance record.

2. Materials and Methods

The development process of Facial Recognition Attendance System using Neural Network will be discussed in this section. To ensure the success of this project, a methodology is an important aspect. The methodology is typically a system of guidelines for resolving a problem, achieving our goal, and researching and studying all project data in order to select and develop an appropriate method. This section explains the stages involved in creating and developing the project, starting from the beginning, and progressing through many critical steps until the end.

2.1 Block Diagram

Figure 1 shows the block diagram of Facial Recognition Attendance System.

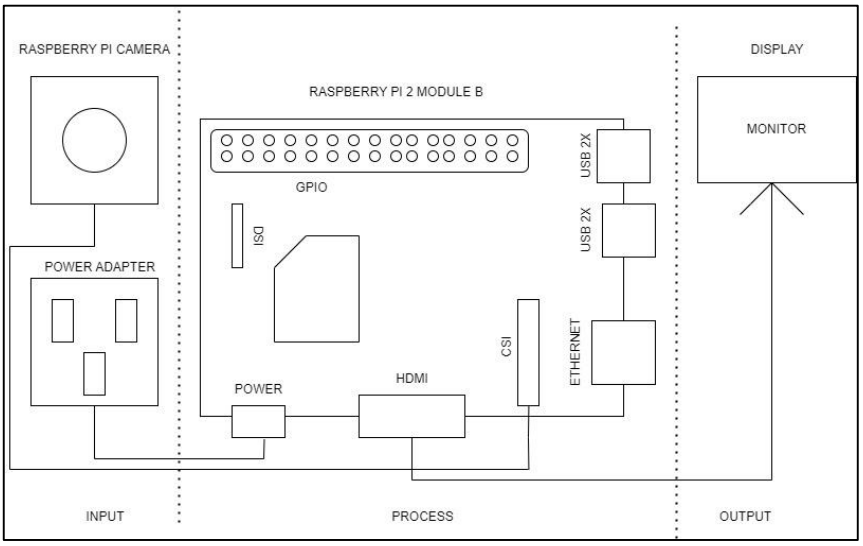

Figure 1: Block Diagram of Facial Recognition Attendance System

2.2 Methods

The process of developing Facial Recognition Attendance System starts from the literature

review. The next step was to determine the objective, scope, and suitable method of implementation for the project. From the next step, the designing and development phase starts from the registration process. The first step of the design and development process is to register the students into the system. This step includes taking a picture of a particular student as the dataset for the students. Using the dlib's frontal face detector, the raspberry pi camera will take 20 sample data of each student to be registered. Before capturing the picture, the students will need to update their name and matrix number of them.

Once the Raspberry Pi cam captures 20 datasets of the student's face, the dataset has stored in a specific folder with a specific user Id. Once the dataset is stored in the folder, a specific Id was extracted called PersonGroupId for the particular person. This step will be done by the Microsoft Azure face services. Once Microsoft Azure is done giving PersonGroupId, a unique Id for each dataset was extracted as well in the folder called FaceId. All these Ids will be used to train the dataset with the PersonGroupId.

After that, the dataset was sent to the training process. The datasets will train by Microsoft Azure and give the status of the training once we run the get status command. If the training is successfully done, it will display "succeeded" in the terminal. After this step, the detection procedure will start, where the students will need to stand in front of the Raspberry Pi camera to detect the total number of student faces. The detected number of student's face was displayed in the terminal. Once the student's face is detected, Microsoft Azure will start the recognition procedure where it will compare the detected face with the trained dataset. Once the face is identified, the terminal displays the name of the students which have already recognized.

The data will then be transferred to the database and to the spreadsheet. The recognized faces will be marked "1" on the spreadsheet with the name and the matrix number of the student. Figure 2 shows the project flowchart.

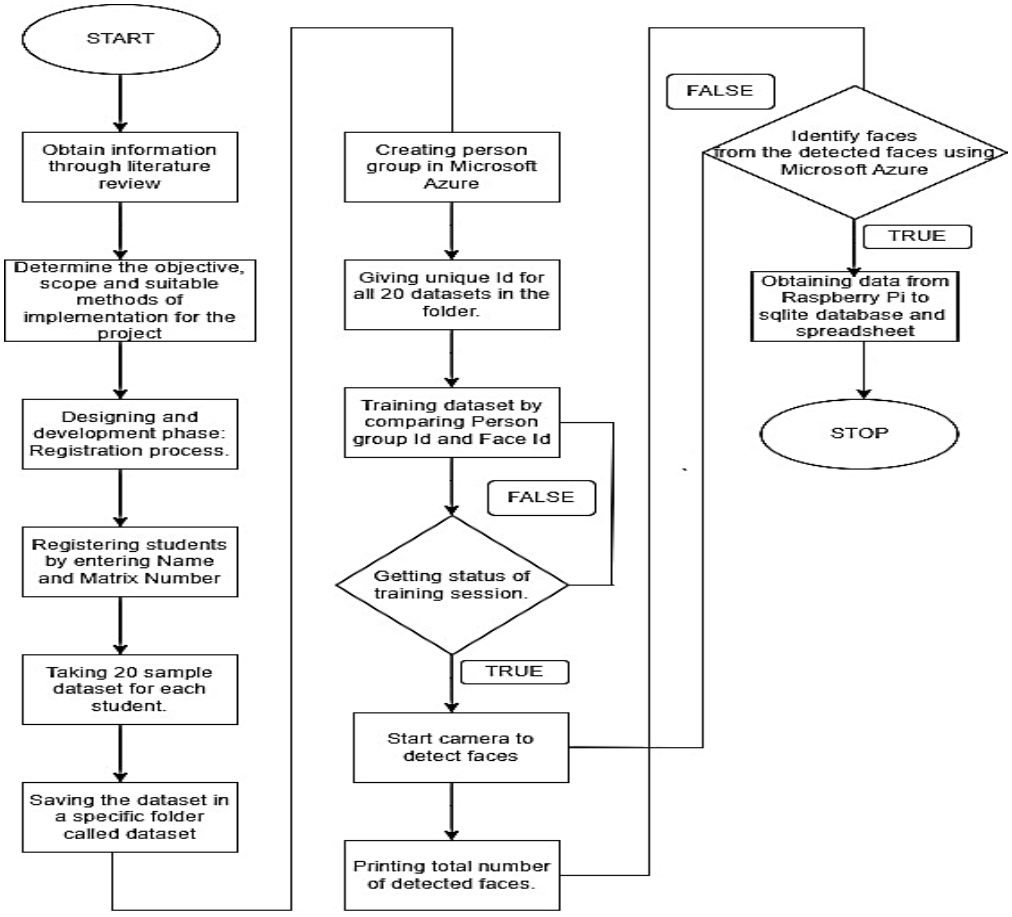

Figure 2: Flowchart of Facial Recognition Attendance System

2.3 Microsoft Azure

Microsoft Azure is a cloud-based platform and infrastructure created by Microsoft that allows for building, deploying, and managing applications and services through a global network of data centers managed by Microsoft. One of the services offered by Azure is Azure Face API, which is a facial recognition service that uses machine learning algorithms like CNN to identify and verify people in images and videos and analyze facial attributes such as age, gender, and emotions. However, the CNN architecture and algorithms used by Microsoft Azure was not mentioned and explained in detail by any website. To use Azure Face API, one must sign up for an Azure account and create a Face API resource in the Azure portal. Then, the API can be used by sending HTTP requests to the API endpoint, passing in the necessary parameters, and image or video data. Figure 3 shows Microsoft Azure Machine Learning Architecture.

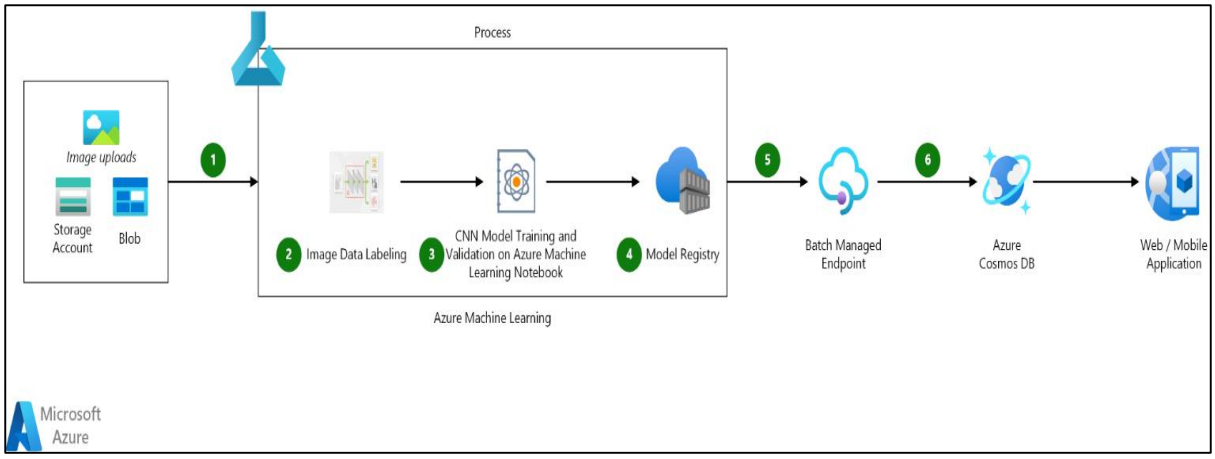

Figure 3: Microsoft Azure Machine Learning Architecture

The increasing use of technology such as IoT and AI has resulted in a massive generation of data. Sifting through the data to find useful information has become a significant problem. Image classification is a useful technique for determining what an image depicts. By categorizing images in high volumes, it can help in handling the data. Convolutional neural networks (CNNs) are efficient in handling image datasets and have been instrumental in advancing the image classification process to its current state-of-the-art level.

There are three main types of layers in CNNs:

- Convolutional layers
- Pooling layers
- Fully connected layers.

The first layer of a Convolutional Neural Network (CNN) is the Convolutional layer. It can be followed by additional Convolutional layers or Pooling layers. The final layer of a CNN is typically the Fully Connected layer. As the number of layers in the CNN increases, the model's complexity also increases, allowing it to identify more parts of an image. The initial layers concentrate on basic features, such as edges. As the image data progresses through the layers of the CNN, it starts to identify more complex elements or shapes in the object, eventually leading to the recognition of the desired object.

2.4 Hardware Setup

The overall connections of peripherals such as monitor, keyboard, mouse and Raspberry Pi with Raspberry Pi Camera Module have been showed in the Figure 4.

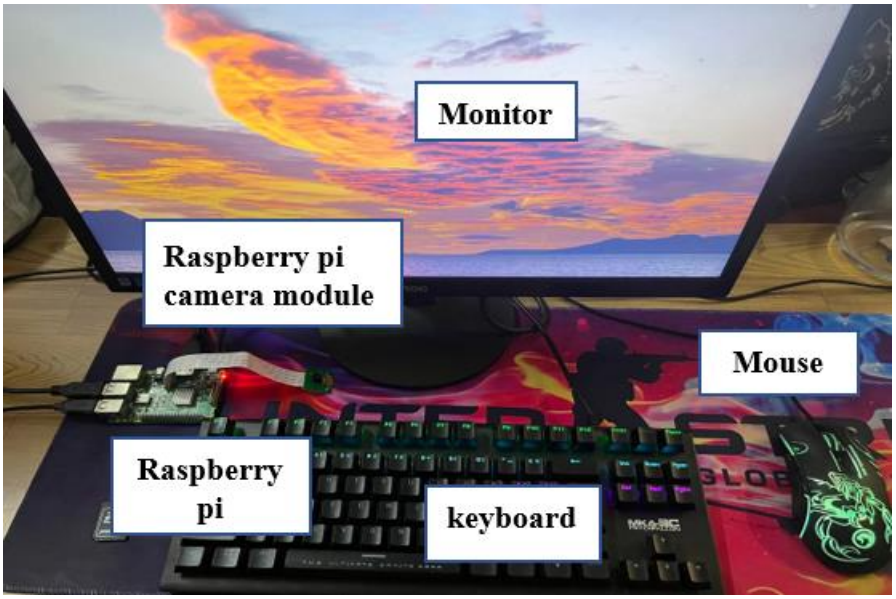

Figure 4: Facial Recognition Attendance System

There are some peripherals connected to the Raspberry Pi such as mouse, keyboard, and monitor. The function of mouse and keyboard are used for registration phase where it will need to enter the students name and matrix number. Besides, the monitor is used for the display of the whole system.

3. Results and Discussion

The result of this project is shown in Figure 5, 6 and 7 from the registration phase to the face detection phase then the Face recognition phase and finally the attendance marking phase.

```
pi@raspberrypi:~ $ source ~/.profile
pi@raspberrypi:~ $ workon cv
(cv) pi@raspberrypi:~ $ cd Autoattendance-Cognitive-master/
(cv) pi@raspberrypi:~/Autoattendance-Cognitive-master $ python add_student.py
Enter student's name : SURENTHIRAN A/L SEGRAN
Enter student's Matric Number : CE190116
```

Figure 5: Terminal output during the registration phase.

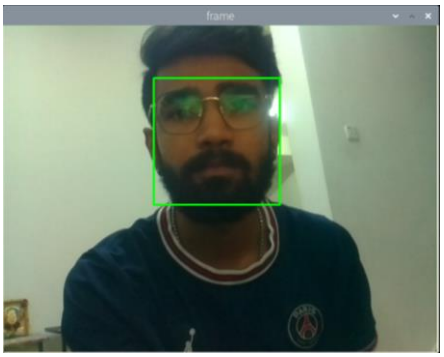

Figure 6: Taking picture from Pi Cam

Figure 7: Dataset of student.

Figures 6 and 7 show the result obtained from the python code. The frame displays the face of the student in a live stream mode where it captured 20 sample data for the training set. Using the OpenCV library, a green square will form around the face of the students to capture the frontal face of the student. The faces of the students were automatically cropped using the dlib frontal face detection algorithm.

```
cap = cv2.VideoCapture(0)
detector = dlib.get_frontal_face_detector()
```

Figure 8: Python code to set up a video capture using dlib

Figure 9 shows the code to be training a face recognition model using the Microsoft Cognitive Services Face API to train the dataset. This will initiate training of the face recognition model for the person group Id and PersonId. Figure 10 shows Python code to detect faces.

train.py - C:\Users\SUREN\Desktop\fyp2\CODE\train.py (3.10.5)

File Edit Format Run Options Window Help

```
import cognitive_face as CF
from global_variables import personGroupId

Key = 'b28418cced3642a485e0244ca0f40af1'
CF.Key.set(Key)
BASE_URL = 'https://liverpoolfc.cognitiveservices.azure.com/'
CF.BaseUrl.set(BASE_URL)

res = CF.person_group.train(personGroupId)
print res
```

Figure 9: Python code to train the dataset using Microsoft Azure

detect.py - C:\Users\SUREN\Desktop\fyp2\CODE\detect.py (3.10.5)

File Edit Format Run Options Window Help

```
import cv2
import dlib
import os
import sys
import sqlite3

detector = dlib.get_frontal_face_detector()

if len(sys.argv) is not 1:
        img = cv2.imread(str(sys.argv[1]))
        dets = detector(img, 1)
        if not os.path.exists('./Cropped_faces'):
                os.makedirs('./Cropped_faces')
        print "detected = " + str(len(dets))
        for i, d in enumerate(dets):
                cv2.imwrite('./Cropped_faces/face' + str(i + 1) +
                        '.jpg', img[d.top():d.bottom(), d.left():d.right()])
```

Figure 10: Python code to detect faces.

Libraries like OpenCV and dlib are imported to detect the faces of the student from the input image. The specific library from dlib, which is frontal face detector was used to detect the total number of faces detected. The number of detected faces were listed in the output terminal once the code has debugged. This part of the code was used as part of a facial recognition system to pre-process images of faces before they are passed to a face recognition model for identification. It could also be used to

extract multiple faces from a group photo for individual recognition. Figure 11 shows Database of student with personId in SQLite.

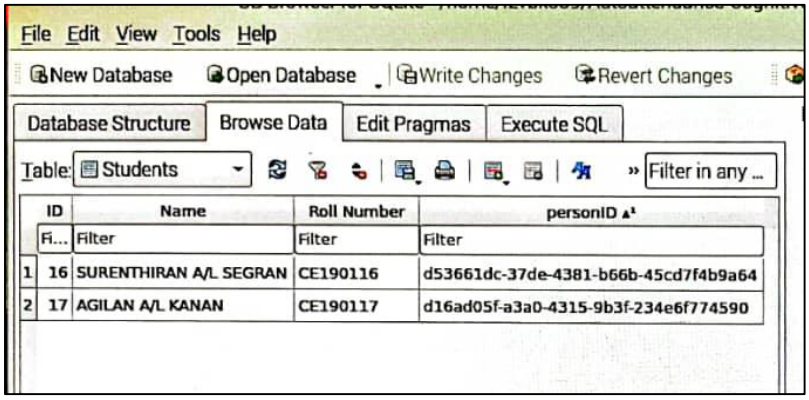

Figure 11: Database of student with personId in SQLite

The SQLite database which has already created before was connected to the Facial Recognition Attendance System in order to update the names and the roll number of the students. The updated names of the student were displayed in the spreadsheet in order to mark the attendance. In the SQlite database, the PersonId were displayed as well. Figure 12 shows the updated data in the spreadsheet.

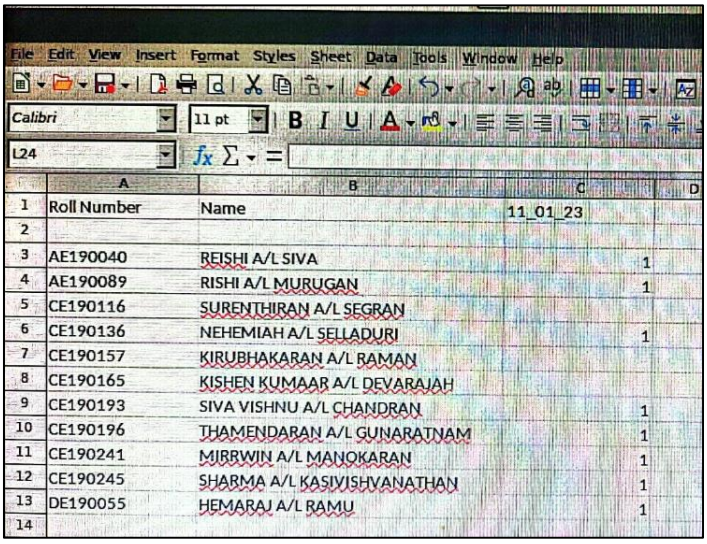

Figure 12: Updated data in the spreadsheet

Figure 12 shows the student's name that has already been recognized and marked the attendance column with "1". For the students who have registered and don't attend the class, the attendance column will not be updated.

3.1 Analysis of detection accuracy in various conditions

The detection accuracy of a Raspberry Pi camera is determined by various factors such as the camera's resolution and quality, lighting conditions, and the distance of the student from the camera. While the

Raspberry Pi camera has the good image quality and is suitable for a variety of applications, including facial recognition, its detection accuracy may not be as high as that of higher-resolution cameras. This is because the Raspberry Pi camera has a lower resolution of 5 megapixels, which makes it harder to identify small details in an image. Furthermore, lighting conditions can significantly influence the detection accuracy of the Raspberry Pi camera, poor lighting can make it difficult to detect and recognize faces if the image is too dark or has too much noise. Also, the performance of the facial recognition algorithm used to analyze the images can affect the overall detection accuracy. If the algorithm is not optimized for the specific application or if it is not trained on a diverse and representative dataset, the detection accuracy may be lower. The overall percentage for the detection accuracy for this project will be 80% which can be improved more after considering the factors that has been mentioned above. Figure 13 shows the unclear image under poor lighting.

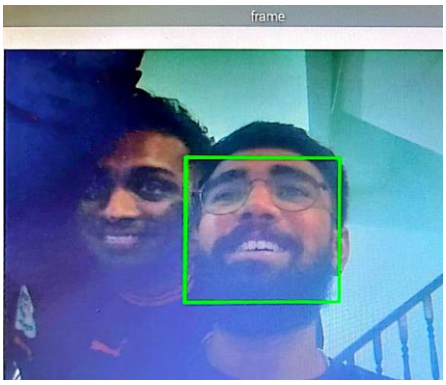

Figure 13: Unclear image under poor lighting

Figure 13 illustrates an image taken under inadequate lighting conditions, where only one face can be detected by the Raspberry Pi camera. The detected face will be outlined with a green square, while the absence of a square indicates that the face was not detected. Similarly, the distance of the student to the camera can also affect the detection accuracy. Generally, the Raspberry Pi camera is most effective for capturing images and videos at moderate to close distances. At these distances, the camera can take clear and detailed images with good contrast and color accuracy. However, as the distance increases, the image quality may decrease due to lens distortion and other factors.

4. Conclusion

From the testing and evaluation of the project, it can be concluded that the development of the Facial Recognition Attendance System has been successful. This was achieved by creating a Facial Recognition Attendance System using various libraries and software specifically designed for facial recognition. The information in the spreadsheet will assist lecturers in monitoring and keeping track of daily attendance lists of students. With the high accuracy of the system, the possibility of proxy attendance is eliminated. Additionally, the system reduces the time required for recording attendance and prevents wastage of time. The accuracy of the Facial Recognition Attendance System can be further improved by increasing the amount of training datasets and using more precise algorithms. To sum up, the objectives of the project were met by designing an attendance system based on facial recognition, constructing a prototype of the system, and testing the system in terms of detection and attendance record accuracy.

Acknowledgement

The authors would also to thank the Faculty of Electrical and Electronic Engineering, University Tun Hussein Onn Malaysia for its support.

References

[1] P. S. Smitha and A. Hegde, "Face Recognition based Attendance Management System," 2020. [Online]. pp .1190-1192. Available: www.researchgate.net/publication/326261079_Face_detection_

[2] D. Nandhini R and S. P. C. N, "Student attendance system in classroom using face recognition technique," in 2016 International Conference on Information and Communication Technology Convergence, ICTC 2016, Nov. 2019, pp. 574–576. doi: 10.1109/ICTC.2016.7763360.