Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 18 No. 13 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijim.v18i13.49277

SHORT PAPER

Visual Programming for Human Detection Using FaceNet in Pocket Code

Md. Salah Uddin[1,2](✉), **Wolfgang Slany**[2]

[1]Multimedia and Creative Technology, Daffodil International University, Dhaka, Bangladesh

[2]Institute of Software Technology, Graz University of Technology, Graz, Austria

salah.mct@diu.edu.bd

ABSTRACT

Pocket Code is a visual programming-based mobile application for creating games, animations, music, videos, and other types of applications. This paper presents the integration of face recognition capabilities into Pocket Code through a visual programming interface based on the FaceNet architecture. The FaceNet dataset is used to train and deploy a face recognition model in Pocket Code visual programming. Integration of the FaceNet algorithm into Pocket Code aims to enhance the accessibility and simplicity of facial recognition technology for students and developers. Building face recognition applications typically involves writing complex code, which can be challenging for both beginners and experienced developers. The Pocket Code visual programming blocks have simplified the process, enabling anyone to easily incorporate face detection into their projects, irrespective of their coding experience. The article discusses the implementation and performance assessment of the FaceNet algorithm in Pocket Code visual programming.

KEYWORDS

artificial intelligence, face recognition, FaceNet algorithm, mobile learning, Pocket Code, visual programming

1 INTRODUCTION

Currently, there is a focus on simplifying programming for young students to learn. Visual programming languages (VPLs) such as Pocket Code [1] have emerged as an engaging method for teaching programming concepts. They enable users to create games, animations, and videos on mobile devices using a visual coding interface with draggable blocks. Face recognition is now widely used in biometric identification and security systems [2] due to advancements in computer vision and facial recognition technologies. However, integrating any face recognition algorithm into an application requires advanced programming skills, which can be a barrier for beginners. This paper discusses integrating the FaceNet face training and detection mechanism into Pocket Code in a more accessible way through simple drag-and-drop code blocks.

Uddin, M.S., Slany, W. (2024). Visual Programming for Human Detection Using FaceNet in Pocket Code. *International Journal of Interactive Mobile Technologies (iJIM)*, 18(13), pp. 195–202. https://doi.org/10.3991/ijim.v18i13.49277

Article submitted 2024-03-23. Revision uploaded 2024-05-08. Final acceptance 2024-05-09.

This paper also discusses the integration of a human detection algorithm based on FaceNet [3] into Pocket Code. This integration enables users to develop applications with face recognition capabilities using a user-friendly visual programming interface. FaceNet is a neural network-based system developed by Google that utilizes deep learning algorithms. It enables users to incorporate human detection mechanisms into their projects. FaceNet directly learns a mapping between face images and a compact Euclidean space, in which the distances represent a measure of similarity between faces. Once this space is created, feature vectors from FaceNet embedding can be used to easily implement tasks such as face recognition, verification, and clustering. Additionally, a novel online triplet mining technique can be used to generate training triplets of roughly aligned matching or non-matching face patches. The method's advantages include state-of-the-art facial recognition performance with only 128 bytes used per face and significantly higher representational efficiency. The system achieves an accuracy of 100% in Pocket Code visual programming blocks.

2 BACKGROUND STUDY

Visual programming languages are programming environments that express programming objects and concepts using graphical components such as blocks, icons, or diagrams [4]. They aim to enhance the usability and accessibility of programming, especially for students and inexperienced programmers. VPLs are becoming increasingly common in educational settings because they offer a practical and interesting approach to learning programming topics and improving computational thinking abilities [5].

The Graz University of Technology in Austria developed a VPL-based application called Pocket Code for mobile devices [1].

Pocket Code is an open-source program that allows users to create interactive stories, games, and animations on tablets or smartphones using a drag-and-drop interface similar to the Scratch programming language [6].

Google researchers developed FaceNet, a deep learning model for facial recognition and computer vision. It achieved high accuracy on various facial recognition benchmarks by calculating the Euclidean distance between the embeddings of two face images and learning a compact Euclidean embedding for each image.

Scratch Extensions [7] allow users to integrate visual programming-based machine learning and computer vision features into applications such as object detection and facial recognition. However, these extensions necessitate the installation of additional software and may not be compatible with tablets or mobile phones.

3 RESEARCH TECHNIQUES AND IMPLEMENTATION

There are several essential elements involved in implementing the FaceNet-based human identification algorithm in the Pocket Code environment:

3.1 FaceNet model integration

The FaceNet model, pre-trained on a large dataset of face images, is integrated into the Pocket Code application. This model is responsible for extracting facial embeddings from input images, which are then used for face recognition and human detection tasks. FaceNet utilizes a deep convolutional neural network (CNN) and triplet loss. The input is a person's image, and the output is a 128-dimensional vector representing the key features of the face. These embeddings compress the face image into a vector and are similar for the same person. This technology enables the compression of high-dimensional image data into low-dimensional embeddings.

These embeddings can be plotted on an x-y axis coordinate system, enabling easy visualization of similar faces within the same cluster. FaceNet embeds faces into Euclidean spaces to measure similarity for facial recognition and clustering. It utilizes these embeddings for standard verification with a specified threshold value. The range of distances for detecting face identity ranges from 0.0 (identical) to 4.0 (different). Exact grouping is ensured by a threshold of 1.1. Faces positioned next to each other have modest gaps between them, whereas faces positioned next to other people have large distances.

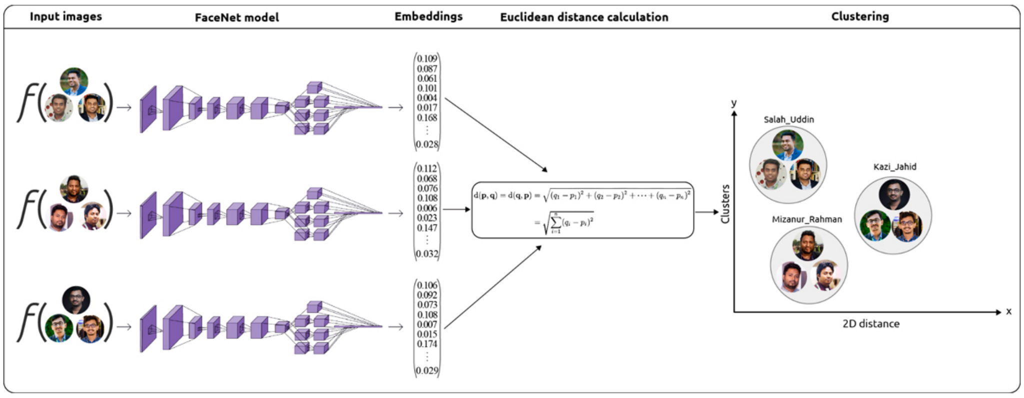

Fig. 1. FaceNet model for image training

3.2 Face recognition and human detection

In order to identify and extract facial areas from the input photos, preprocessing is conducted before the images are input into the FaceNet model. The next step is to calculate the Euclidean distance between the input image's embeddings and a reference set of embeddings after obtaining the face embeddings from the FaceNet model. The face identifies the person as known if the distance is less than a certain threshold. Otherwise, it is classified as an unknown or new face, demonstrating the detection of a human.

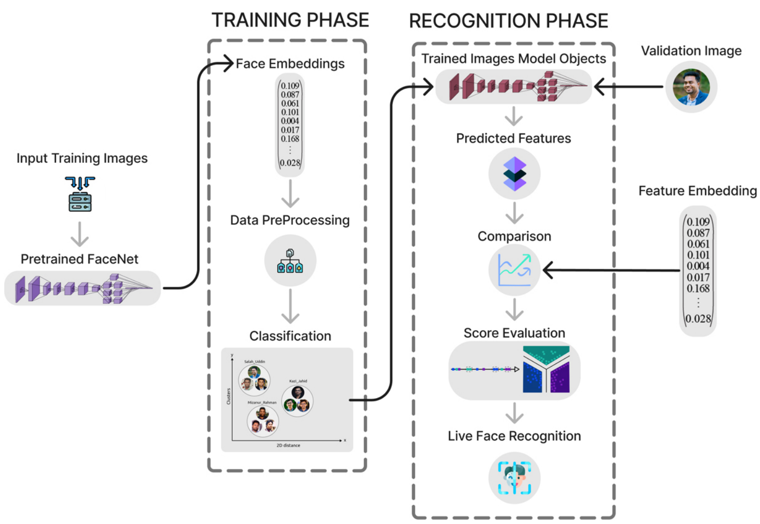

Fig. 2. Image train and recognition process

Algorithm 1: FaceNet Model Creation

input: minimum 2 images per person

repeat
  randomly initialize the FaceNet parameteres or load the saved checkpoint.
  for each image **do**
    preprocess images to 128x128
  generate random embeddings 128-d
  form triplets:
    randomly select anchor image
    select negative and positive image
    adjust facenet parameteres:

$$f(x_i) = [128d]$$

  for each triplet **do**
    triplet loss function:

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

    for fast convergence **do**
      select triplets:

$$HardPositive : Argmax \|f(x_i^a) - f(x_i^p)\|_2^2$$

$$HardNegative : Argmin \|f(x_i^a) - f(x_i^n)\|_2^2$$

    train on Inception Network Architecture
  save the checkpoints
until loss $\geq$ stop_threshold [convergence point]
return saved checkpoint

3.3 Integration with pocket code projects

Visual programming blocks for human detection integrated into the Pocket Code environment enable users to train and recognize human faces in their projects. Users can create interactive applications, games, or educational projects by combining these blocks with other programming constructs and multimedia elements without needing to know complex code.

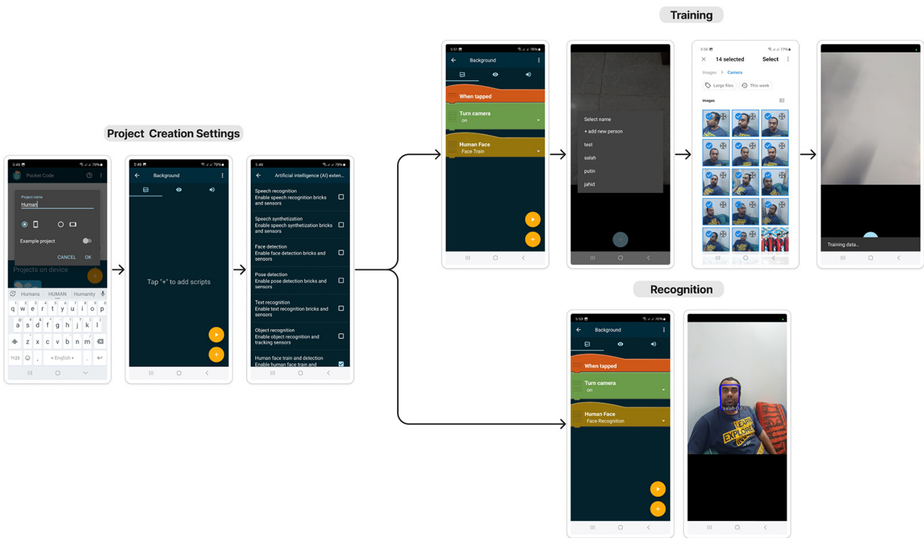

Fig. 3. Visual programming for Pocket Code human face train and recognition

4 EXPERIMENTAL RESULTS AND EVALUATION

To assess the accuracy, processing time, and resource utilization of the Pocket Code human recognition algorithm, it was tested in various ways using a variety of input images and configurations.

Accuracy: The accuracy of the human detection algorithm was assessed by comparing the detected faces with ground-truth annotations. The testing dataset consisted of images with a variety of human faces, different poses, and diverse backgrounds.

The dataset consists of 80% training data, 20% testing data from the training set, and 10 pictures from the test set. Correct predictions can be calculated using the confusion matrix.

Accuracy = 10 / 10 × 100% = 100%

Three-fold cross-validation is used to assess the SVM classification during validation to obtain the optimal model. The dataset is divided into three subgroups, each containing ten images, for testing using three-fold cross-validation.

Table 1 displays the outcomes of the three-fold cross-validation test after the repetition has been completed three times. Using the confusion matrix and three-fold cross-validation, the SVM validation results in the dataset yielded 100% accuracy.

Table 1. Threefold cross-validation data train accuracy in Pocket Code

Fold	Accuracy (FaceNet)
1	100%
2	100%
3	100%

The FaceNet algorithm demonstrated highest accuracy in both recognized and unrecognized faces during system testing, as illustrated in Table 2.

Table 2. Familiar image recognition accuracy in Pocket Code

Name	Image Recognition		Probability
	True	False	
salah	yes		0.41–0.43
jahid	yes		0.39–0.41
hasan	yes		0.35–0.38

Accuracy = 3/3 × 100% = 100%

The results of testing FaceNet on unrecognized faces are shown in Table 3.

Table 3. Unfamiliar image recognition accuracy in Pocket Code

Name	Image Recognition		Probability
	True	False	
Unfamiliar1		yes	0.12–0.15
Unfamiliar2		yes	0.16–0.17
Unfamiliar3		yes	0.14–0.15

Accuracy = 3/3 × 100% = 100%

Training Processing Time: The training time was measured for different stages of the FaceNet model algorithm. To evaluate the accuracy rate of the proposed algorithm in Pocket Code for training datasets in mobile applications in this research, a phone with the following specifications was used: CPU-Octa-core (2 × 2.4 GHz Cortex-A78 & 6 × 2.0 GHz Cortex-A55), RAM – 6GB. Finally, the following result was observed:

Table 4. Image training time and accuracy in Pocket Code

Number of Image	Training Accuracy	Time
15 (size:3000*4000)	100%	4.2s
30 (size:3000*4000)	100%	7.5s
50 (size:3000*4000)	100%	9.3s

5 EXAMPLES AND USE CASES

The visual programming method for human recognition used by Pocket Code has many potential applications.

Interactive games and experiences: Users can create interactive games that respond to faces, allowing game characters to behave differently based on whether their identities or facial expressions are recognized.

Educational projects: By utilizing human detection in Pocket Code, educational institutions can provide students with a practical way to learn about computer vision, facial recognition, and human-computer interaction.

Security and access control: The basic security or access control systems, which determine access based on identifying authorized and unauthorized individuals, can be established with the assistance of Pocket Code.

Social and communication applications: Face recognition software considers automatically naming individuals in pictures or videos, along with personalized greetings, which can enhance friendly relationships and partnerships.

Creative projects: Pocket Code has simplified the use of human detection for creatives, designers, and artists in interactive art installations, music videos, and multimedia projects.

6 CONCLUSION

The paper introduces the integration of the FaceNet human detection algorithm into Pocket Code. This integration allows users to develop mobile applications with face recognition and human detection capabilities directly on their mobile devices and tablets using visual programming.

The proposed approach aims to make face recognition technology accessible to a broader audience. This will enable users to integrate human detection into Pocket Code projects, facilitating exploration and experimentation with computer vision and machine learning concepts in a practical and engaging way. The human detection algorithm in Pocket Code demonstrates promising results in accuracy, processing time, and resource utilization, rendering it suitable for real-time applications, interactive games, educational projects, security and access control systems, augmented reality experiences, and creative multimedia projects. It may inspire creativity, innovation, and advancements in computer vision and artificial intelligence education if face recognition technology is made more accessible through visual programming.

Future studies should examine the integration of sophisticated models and algorithms as well as the effectiveness of visual programming environments in teaching students computer vision and machine learning principles.

7 REFERENCES

[1] W. Slany, "Pocket Code: A scratch-based integrated development environment for mobile devices," in *Proceedings of the IEEE Global Engineering Education Conference (EDUCON)*, 2014, pp. 35–36. https://doi.org/10.1145/2660252.2664662

[2] R. Jafri and H. R. Arabnia, "A survey of face recognition techniques," *Journal of Information Processing Systems*, vol. 5, no. 2, pp. 41–68, 2009.

[3] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823. https://doi.org/10.1109/CVPR.2015.7298682

[4] M. M. Burnett, "Visual programming," *Encyclopedia of Electrical and Electronics Engineering*, pp. 275–283, 1999. https://doi.org/10.1002/047134608X.W1707

[5] C. Kelleher and R. Pausch, "Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers," *ACM Computing Surveys (CSUR)*, vol. 37, no. 2, pp. 83–137, 2005. https://doi.org/10.1145/1089733.1089734

[6] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, and Y. Kafai, "Scratch: Programming for all," *Communications of the ACM*, vol. 52, no. 11, pp. 60–67, 2009. https://doi.org/10.1145/1592761.1592779

[7] Scratch Extensions. (n.d.). https://extensions.scratch.mit.edu/

[8] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. 511–518. https://doi.org/10.1109/CVPR.2001.990517

[9] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016. https://doi.org/10.1109/LSP.2016.2603342

[10] F. Cahyono, W. Wirawan, and R. Fuad Rachmadi, "Face recognition system using facenet algorithm for employee presence," in *2020 4th International Conference on Vocational Education and Training (ICOVET)*, 2020, pp. 57–62. https://doi.org/10.1109/ICOVET50258.2020.9229888

[11] M. Müller, C. Schindler, and W. Slany, "Pocket Code – A mobile visual programming framework for app development," in *2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, Montreal, QC, Canada, 2019, pp. 140–143. https://doi.org/10.1109/MOBILESoft.2019.00027

[12] A. C. Petri, C. Schindler, W. Slany, and B. Spieler, "Game design with Pocket Code: Providing a constructionist environment for girls in the school context," in *Proceedings Constructionism in Action 2016*, Bangkok, Thailand, 2016, pp. 111–118. http://e-school.kmutt.ac.th/constructionism2016/Constructionism%202016%20Proceedings.pdf

[13] F. D. Adhinata, D. P. Rakhmadani, and D. Wijayanto, "Fatigue detection on face image using FaceNet algorithm and k-nearest neighbor classifier," *Journal of Information Systems Engineering and Business Intelligence*, vol. 7, no. 1, pp. 22–30, 2021. https://doi.org/10.20473/jisebi.7.1.22-30

[14] M. A. Lazarini, R. Rossi, and K. Hirama, "Systematic literature review on the accuracy of face recognition algorithms," *EAI Endorsed Trans IoT*, vol. 8, no. 30, p. e5, 2022. https://doi.org/10.4108/eetiot.v8i30.2346

[15] S. Munasinghe, C. Fookes, and S. Sridharan, “Human-level face verification with intra-personal factor analysis and deep face representation,” *IET Biom.*, vol. 7, no. 5, pp. 467–473, 2018. https://doi.org/10.1049/iet-bmt.2017.0050
[16] Pandit, Anjali, Nikalje, Ritesh, Vishwakarma, Neha, Vishwasrao, Vaishnavi, and Khose, “Face authentication using MTCNN and FaceNet,” *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 11, no. XI, pp. 1140–1143, 2023. https://doi.org/10.22214/ijraset.2023.56679
[17] A. Abozaid, A. Haggag, H. Kasban, and M. A. Eltokhy, “Multimodal biometric scheme for human authentication technique based on voice and face recognition fusion,” *Multimedia Tools and Applications*, vol. 78, pp. 16345–16361, 2018. https://doi.org/10.1007/s11042-018-7012-3
[18] R. Goel, I. Mehmood, and H. Ugail, “A study of deep learning-based face recognition models for sibling identification,” *Sensors*, vol. 21, no. 15, p. 5068, 2021. https://doi.org/10.3390/s21155068

8 AUTHORS

Md. Salah Uddin is an Assistant Professor and the head of the Department of Multimedia and Creative Technology at Daffodil International University. He is a PhD researcher at the Technical University of Graz in Austria. His research interests include machine learning, data science, big data, cloud computing, game theory, and AR/VR. He can be contacted at salah.mct@diu.edu.bd.

Wolfgang Slany is the Professor of the Institute of Software Technology at Graz University of Technology and the head and founder of the Catrobat nonprofit free open-source project. His research interests include poverty alleviation through coding education for teenagers, with a focus on girls, refugees, and adolescents in developing countries. Slany received a Ph.D. in applied computer science and artificial intelligence from the Vienna University of Technology. He conducts research, teaches, and consults on sustainable large-scale agile software development and user experience topics for mobile platform projects.