

Hand gesture control accuracy through increased epoch and batch size

AJAYI Oluwaseun Kayode^{1,a*}, AJAYI Oladipupo Omogbolahan^{1,b},
MALOMO Babafemi O.^{1,c}, ADEYI Abiola John^{2,d} and NWANKWO Benneth O.^{1,e}

¹Department of Mechanical Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria

²Department of Mechanical Engineering, Ladoke Akintola University of Technology, Ogbomosho, Nigeria

^aokajayi@oauife.edu.ng, ^bajayidipo@gmail.com, ^cbobmalom@yahoo.com,
^dadeyi.abiola@yahoo.com and ^enwankwobenneth22@gmail.com

Keywords: Tensor Flow, Facial Recognition, Batch Number, Epoch

Abstract. The conventional method of mechanism control which requires physical contact is being replaced with remote control, especially with the advent of the Internet of things (IoT). Facial recognition is used to identify and authenticate faces from images or videos. It has many applications, including granting or restricting access to a facility, or secured areas, or preventing unauthorized users, including selective access. The accuracy of such a recognition and control system is very important and depends on how the system was modeled and adopted. In this study, facial images and gesture images were collated and modeled in a python neural network, then optimized using TensorFlow. The algorithm was compiled unto a raspberry pi for testing on a developed automatic gate. An effective method of achieving accuracy using the epoch is presented in this work. Five (0, 5, 10, 15, 20) batch number and epoch respectively were modeled to achieve accuracy for the system. The model was trained with five gestures; fist, palm, thumb up, thumb, and last finger. The gesture recognition accuracy achieved through the epoch was maximum at 99.97 when both batch number and epoch were set to 20. However, when no epoch was set, the accuracy was below 10.2, whereas there was no accuracy below 98% when the epoch was introduced. This depicts the importance of epoch in achieving accuracy in image recognition. It was also discovered that the higher the epoch and the batch number the greater the accuracy, but the processing would require a very high processing unit.

Introduction

The conventional method of mechanism control which requires physical contact is being replaced with remote control, especially with the advent of the Internet of things (IoT). The concept of biometrics involves capturing biological images, storing them, and later retrieving them usually as a form of registering the object. Facial recognition, a biometrics concept is used to identify and authenticate faces either from images or videos. It has many applications, including granting or restricting access to a facility, secured areas or prevention of unauthorized users including selective access [1-5]. Among the pioneer works in facial recognition was the work of Sirovich and Kirby who presented a solution to solve facial algebra problems using linear algebra, but were limited in capacity to store large data [6]. Despite the early efforts, this system gained traction early 2000s with the proliferation of mobile telephony, larger computer power, and data storage [7].

To achieve facial recognition, feature extraction is paramount, which is divided into face and facial landmark extraction [8]. The accuracy of detection can be governed using Bayesian rule-based, Gaussian mixture model and the Expected-maximization (EM) algorithms [9, 10]. The outputs are in form of statistical values or otherwise referred to as probabilistic values, which depict the accuracy of the pixel detected [10]. This work presents the effect of epoch and batch

number on the detection of face and hand gestures in the control of an automatic gate mechanism. Real-time tracking of hand gestures has been worked on in recent times [11, 12], while a combination of hand gestures and head pose was adopted for a control system [13]. These two have found great applications in home security and automation [14].

Methodology

This section involves three stages; Facial recognition (data collection, data pre-processing, and model training and validation), Hand gesture recognition, and test with Automatic garage door.

Facial Recognition: This stage was achieved through; Data Collection, Data Pre-processing and Model training, and Validation.

For the data collection, data were obtained from the OpenFace database; 20 images of 5 different faces were added and classified to make 100 images processed for testing.

Data pre-processing for detection was in two phases; face detection and facial landmark detection. In face detection, the Haar cascade algorithm (OpenCV) [15] was used from the python library utilizing the Single Shot Detector (SSD) framework with the ResNet as the base network, while for face landmark detection; dlib and OpenCV were used.

Face Detection: This phase involves data collection, data pre-processing, model training and validation, and model evaluation. Deep neural network was used to train the face recognition algorithm. Images were inputted 2000 per batch which includes; anchor image, positive image, and negative images for each element of the batch.

Model training and classification: Deep neural network in python, which usually involves data entry and model training. In this procedure, triplet loss function was used for the training which has the advantage of ensuring that the anchor image is closer to the positive image (real image) than it is to the negative image. In the training, three sets of images; anchor image, positive image, and negative image constituted the inputs. In the image input, anchor and positive images (image belonging to the same person) and negative image (image belonging to another person) were inputted. 123-dimensions embedding for each face is computed and assigned weights through the network which results in larger disparity between the real images and the negative image.

$$\sum_i^N [\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha] \quad 1$$

The system was built to work real-time using the haar cascade face detection algorithm. After detection, it takes a picture of the face, runs through face encoding, comparing with existing or registered users.

Hand gesture recognition: The Convolutionary neural network deep algorithm has been handy for multiple image processing; therefore, it was adopted for the hand gesture recognition modeling. The workflow order adopted was data collection, data pre-processing and feature recognition, and training machine learning model.

First set of data was collected for open palm, fist, index finger, index finger and thumb, thumb up, thumb down and three middle fingers by taking physical images while the second batch was gotten online form Kaggle.com. Therefore, 4000 images were collected for each gesture. The data was reset to 50 x 50 pixels for uniformity. This data was used to train the system using deep neural network in python program. Two thousand images each on open palm, closed palm (fist), index finger, index finger and thumb, thumb up, thumb down, and three middle fingers were sourced making up fourteen thousand images from where three thousand were chosen for validation purposes. To make the model more robust, the position and size of the gestures were varied for each frame. These pictures were either self-taken or sourced from kaggle.com. In the data pre-processing, images were set to 50 x 50 order for uniformity. The training has input size of 2500 nodes with 25 nodes corresponding to 5 nodes of 5 hand gestures.

Automatic garage door prototype: Materials: raspberry pi, camera module, python, pc and automatic gate.

A prototype automatic garage door using the materials mentioned above was assembled to evaluate the performance of facial recognition and hand gesture intelligence. The algorithm was compiled on a raspberry pi to control the opening and closing of the gate the camera module was attached to the top of the gate to detect a registered face, the identified face is then given access to control the gate through hand gestures to open, stop or close the gate. The microcontroller sends a corresponding hand gesture signal through the Infrared module to the servo motor to either open or close the gate. Therefore, the algorithm order is; face detection, face authentication, hand gesture detection, send detected hand gesture to the servo motor and servo motor carries out sent instruction/ command. This procedure is repeated for every detection.

The trained and created model's application file was then uploaded to the raspberry pi, which has a camera module for facial and gesture detection.

Results

In training the hand gesture recognition model, the following parameters were set; 2D convolution layer applying number of convolution filters to the image: three pooling layers were used to downsample images extracted by the convolutional layers thereby reducing the dimensionality of the feature map to achieve a reduction in the processing time; introduction of dense layers which performs classification on the layers in previous procedures; the use of batch size which controls the accuracy of the estimated error gradient in training neural networks: number of epochs that determines whether the algorithm will work through the entire training dataset; and the number of trained images used per gesture to train the hand gesture model.

Tensorflow was used to define the Deep Neural Network Model used to train the hand gesture recognition model. Five input arguments were used; Data to train the model, Target to train the model, Number of epochs, validation sets/data, and batch size. The variation of batch sizes 5, 10, 15 and 20 were used against epoch variations 0, 5, 10, 15, and 20. Model accuracy and time taken to train the model were recorded and presented in Figures 1, Figure 2, and 3. The response time varied between 40.89 seconds at 5 epoch and batch sizes respectively to 1892.42 when batch size and epoch were set to 20 each. However, the accuracy of the model was highest at the maximum epoch and batch sizes with the value at 99.967 as seen in table 1.

Higher epoch and batch sizes would have been used, but for the limitation of the CPU, which has the likelihood of getting an accuracy closer to 100% for the model identification of images but with higher computation time as presented in table 2 and table 3. The facial recognition system was able to detect multiple registered users at the same time (Figure 4).

Table 1: Hand gesture model for five users

		Confidence Level				
Users	1	2	3	4	5	
1	99.92828	99.76700	99.59408	99.03502	99.57861	
2	99.05941	99.07965	99.01235	99.48297	99.83151	
3	98.68325	98.27860	99.08360	99.56101	98.64261	
4	99.86299	99.00904	99.61547	99.34412	99.50295	
5	98.15253	98.54012	98.24575	98.87617	98.59599	

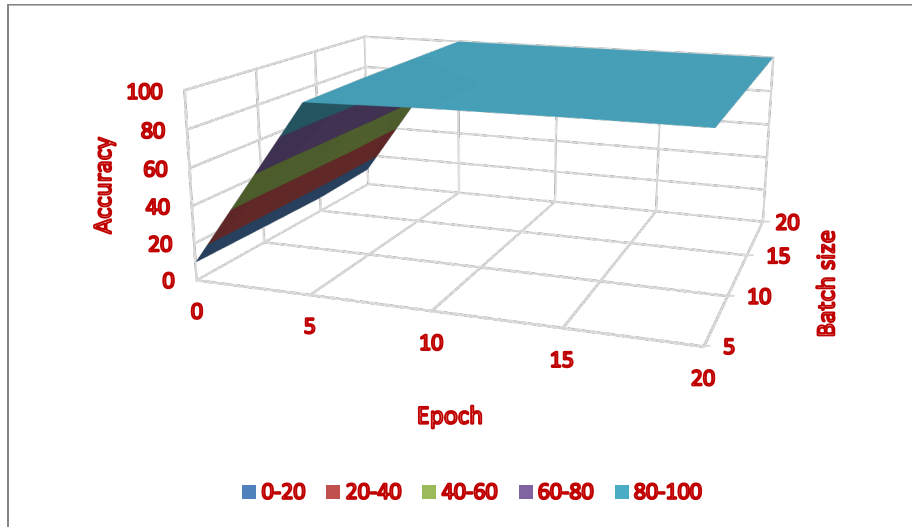


Figure 1: Accuracy level with varying batch size and epoch

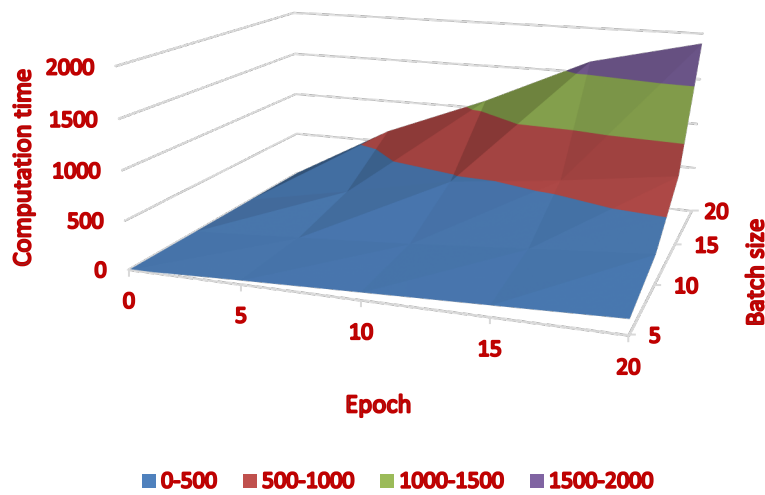


Figure 2: Computation time with varying batch size and epoch

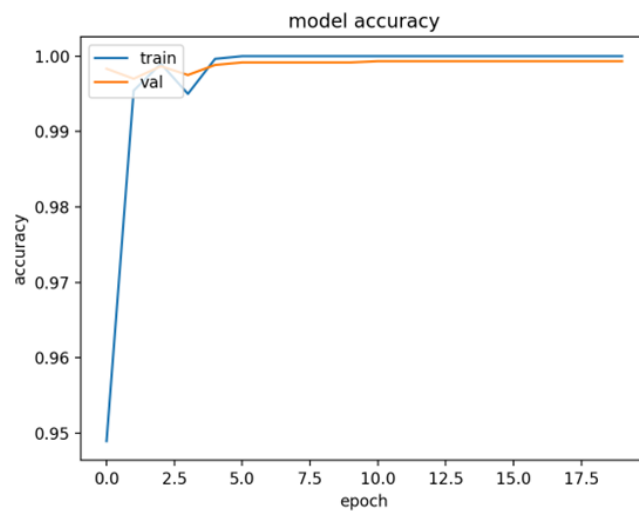


Figure 3: Plot of model accuracy with corresponding epoch

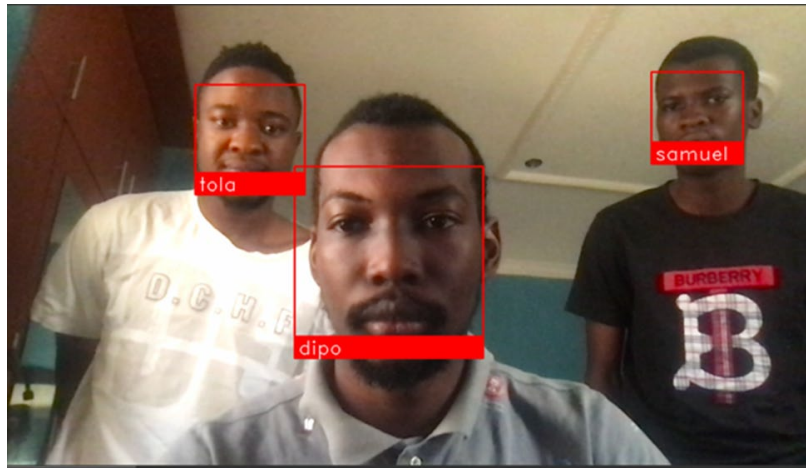


Figure 4: Detection of multiple registered users

Table 2: Face recognition test for five users

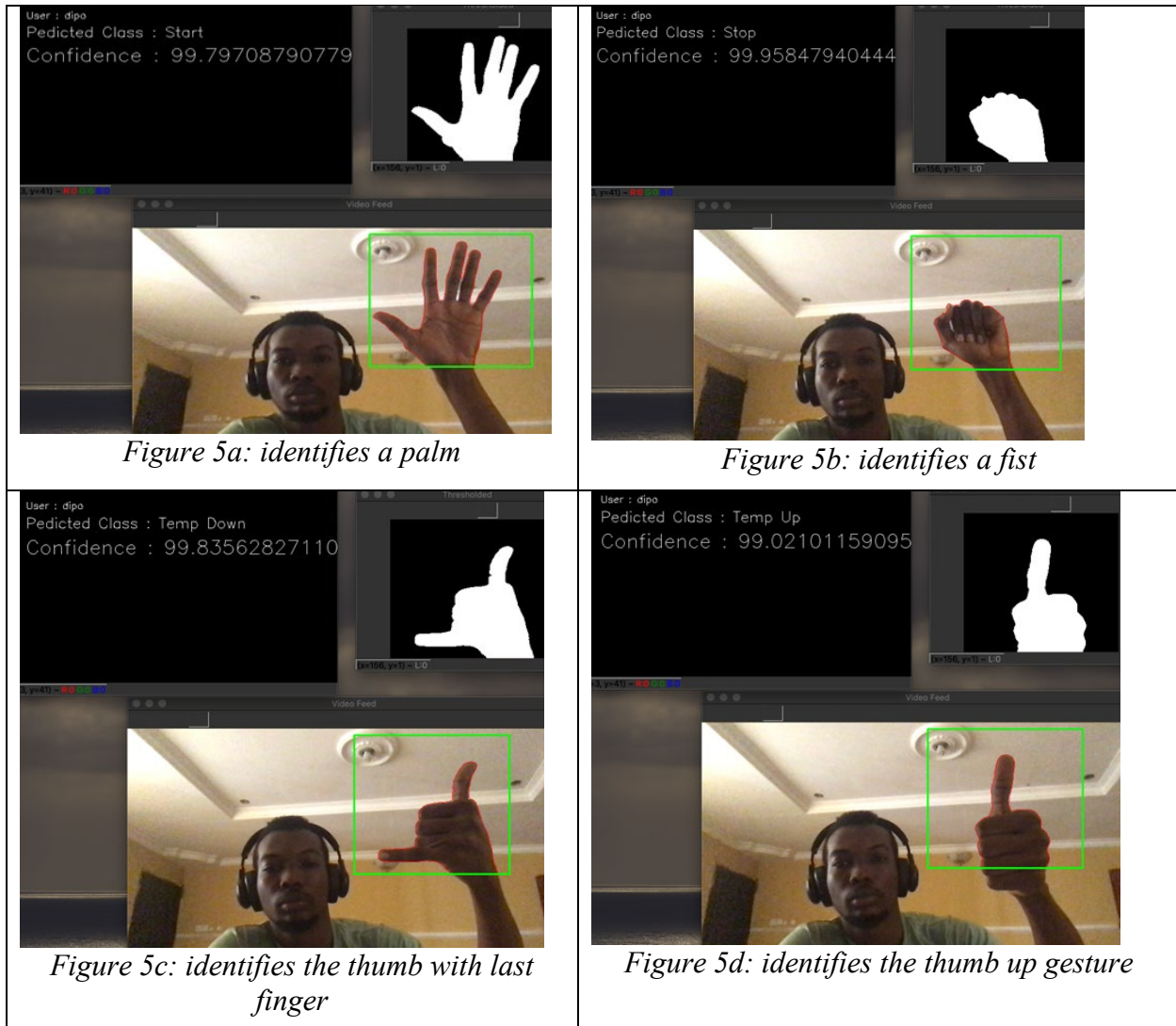
Users		Face Recognized			
1	True	True	True	True	True
2	True	True	True	True	True
3	True	True	True	True	True
4	True	True	True	True	True
5	True	True	True	True	True

Table 3: Time taken to train the model

Batch Size	Epoch				
	0	5	10	15	20
	Training time (sec)				
5	0	40.894	75.787	111.132	148.236
10	0	66.742	135.965	222.943	307.826
15	0	217.276	447.403	576.215	732.753
20	0	629.192	1088.779	1622.536	1892.426

The system was tested using five registered users, repeated five times for each. The system recognized all the users for each of the sessions. For the hand gesture, the system outputs the confidence level and a picture of the gesture captured (Figures 5(a-d)). The confidence level recorded varied between 98.15 and 99.92 throughout the experiment for the fist, palm, thumb with last finger, and thumbs up. These tests were carried out at the highest epoch value and batch number, which showed higher accuracy at the model training phase. However, when no epoch was

set, the accuracy was below 10.2, whereas there was no accuracy below 98% when epoch was introduced. This depicts the importance of epoch in achieving accuracy in image recognition.



Conclusion

Facial recognition and hand gesture algorithm was developed and implemented for automatic gate control. Facial recognition and gesture control was achieved more accurately using high epoch and batch number. The model was trained with five gestures; fist, palm, thumb up, thumb, and last finger. The gesture recognition accuracy achieved through the epoch was maximum at 99.97 when both batch number and epoch were set to 20. However, when no epoch was set, the accuracy was below 10.2, whereas there was no accuracy below 98% when the epoch was introduced. This depicts the importance of epoch in achieving accuracy in image recognition. It was also discovered that the higher the epoch and the batch number the greater the accuracy, but the processing would require a very high processing unit.

References

[1] Turk M and Pentland A. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71-86, 1991. <https://doi.org/10.1162/jocn.1991.3.1.71>

- [2] Bhatia R., Biometrics and Face Recognition Techniques. *International Journal of Advanced Research*, 3(5): 93-99, 2013.
- [3] McConnel R. K., "Method of and apparatus for pattern recognition". United States of America Patent 4,567,610., January 1986.
- [4] Deepan R., Rajavarman S. V. and Narasimhan K. Hand Gesture Based Control of Robotic Hand using Raspberry Pi Processor. *Asian Journal of Scientific*, 8(3): 392-402, 2015.
<https://doi.org/10.3923/ajsr.2015.392.402>
- [5] Chi-Man-Zhu P. and Wei H.-M. F., Real-Time Hand Gesture Recognition using Motion Tracking. *International Journal of Computational Intelligence Systems*, 4(10): 277-286, 2011.
<https://doi.org/10.1080/18756891.2011.9727783>
- [6] Sirovich L. and Kirby M. Low-dimensional procedure for the characterization of human faces. *Optical Society of America*, 4:519, 1987 <https://doi.org/10.1364/JOSAA.4.000519>
- [7] Turk M and Pentland A. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71-86, 1991. <https://doi.org/10.1162/jocn.1991.3.1.71>
- [8] Z. Sufyanu, F. S. Mohamad, A. A. Y. A. N. Musa and R. Abdulkadir. Feature extraction methods for face recognition. *International Review of Applied Engineering Research (IRAER)*, 5(3): 5658-5668, 2016.
- [9] Bhuyan M., Neog D. R. and Kar M. K. Fingertip Detection for Hand Pose. *International Journal on Computer Science and Engineering (IJCSSE)*, 4(3): 501-511, 2012.
- [10] Elmezain M., Al-Hamadi A., Sadek S. and Michaelis B. Robust methods for hand gesture spotting and recognition using Hidden Markov Models and Conditional Random Fields. 10th IEEE International Symposium on Signal Processing and Information Technology, 131-136, 2010. <https://doi.org/10.1109/ISSPIT.2010.5711749>
- [11] Deepan R., Rajavarman S. V. and Narasimhan K. Hand Gesture Based Control of Robotic Hand using Raspberry Pi Processor. *Asian Journal of Scientific*, 8(3): 392-402, 2015.
<https://doi.org/10.3923/ajsr.2015.392.402>
- [12] P. Chi-Man-Zhu and H.-M. F. Wei. Real-Time Hand Gesture Recognition using Motion Tracking. *International Journal of Computational Intelligence Systems*, 4(10): 277-286, 2011.
<https://doi.org/10.1080/18756891.2011.9727783>
- [13] Tu Y.-J., Kao C.-C., Lin H.-Y. and Chang C.-C.. Face and Gesture Based Human Computer Interaction. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8(9): 219-228, 2015. <https://doi.org/10.14257/ijsp.2015.8.9.23>
- [14] Sharma R. P. and Verma G. K.. Human Computer Interaction using Hand Gesture. Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015), India, 2015.
<https://doi.org/10.1016/j.procs.2015.06.085>
- [15] Januzaja Y., Lumaa A., Januzaja Y. and Ramaj V. Real Time Access Control Based on Face Recognition, *International Conference on Network security & Computer Science*, Antalya (Turkey), 2015.