Investigating camera calibration for eye tracking of the physically challenged

Zeenat Al-Kassim , Qurban Memon

EE Department, College of Engineering, UAE University, UAE

Abstract. This paper tries to address a problem faced by a proportion of community - the paralysed people, and investigates a solution in calibrating a camera used for the purpose of eye detection. The eye tracking is later used to read characters from a partner-assisted keyboard. As such, this keyboard solves the problem of immobility. In order to calibrate camera for the purpose of eye detection, two approaches are implemented-Artificial Neural Networks and Support Vector Machines in context of paralysed people, and are discussed in detail in this paper. The results of both approaches are compared in order to select the best approach to be used in keyboard that operates on eye tracking.

1 Introduction

According to National Health Service in the UK [1], Paralysis is loss of the ability to move muscles that may be associated with loss of feeling and other bodily functions. It can be seen as a form of nerve damage due to the inability of brain to control muscles, caused by damage in the nerves of spinal cord. Three main causes for paralysis in the U.S., according to a study done by the Christopher & Dana Reeve Foundation [2], are: stroke, spinal cord injuries and multiple sclerosis. Other causes include brain injuries, Parkinson's disease, Guillain-Barre syndrome, Spina bifida, Motorneurone disease [1]. Since sudden accidents are a major cause of paralysis, many people find themselves suddenly paralyzed. As such, the number of paralyzed people keeps on increasing every day. In the U.S., one person in every 50 suffers from paralysis [2]. This is around 6 million people (around 2% of the U.S. population) [2]. People with paralysis face many challenges in their lives. Paralysis can be in of the following four forms [1], as shown in Fig. 1:

- Monoplegia: where one limb is paralysed
- Hemiplegia: where the arm and leg on one side of the body are paralysed
- Paraplegia: where both legs are paralysed, or sometimes the pelvis and some of the lower body
- Tetraplegia/Quadriplegia: where both the arms and legs are paralysed

Figure 1: Types of Paralysis (grey: paralysed body part) [3]

People suffering from tetra/quadriplegia face difficulties even when performing simple daily life activities like communication. For instance, people during late stages of ALS [4] cannot move their entire body muscles except for the eyes muscles. As such, several techniques have been employed since ages to help those people communicate freely and easily. One such famous technique is Partner-Assisted Scanning. In this technique, the care giver presents a set of words or letters to the disabled patient, and the patient can select from among those letters or words in order to express his/her thoughts [5]. Selection, as in Fig. 2, might happen by pointing or blinking of the eye depending on the capabilities of the disabled person [5]. This method is used to help children as well as adults who suffer immobility due to late stages of ALS, multiple sclerosis, or severe injuries [5].

Figure 2. Partner Assisted Scanning [6]

In order to address this, technology has been exploited to build systems that try to simplify human life. These systems utilized concepts like human computer interaction, image processing, virtual/augmented reality, and so on, to build devices by which humans can communicate easily with the machines [7-8]. An example is the Kinect Sensor developed by Microsoft, and the Wii Remote developed by Nintendo. In this paper, a research work is reported that aims to utilise the advance technology of gesture recognition, or more precisely eye recognition to help such people. The work here is to build a keyboard, which exploits eye movement to implement the concept of Partner-Assisted Scanning on a

digital screen. The first phase of this work involves camera calibration where the system needs to detect the user's eyes. Two commonly known methods (i.e., Artificial Neural Networks (ANN) and Support Vector Machines (SVM)) were used and compared to come up with a best method to implement in camera calibration of the user's eyes in such an environment..

2 Camera Calibration

The aim is to design a system that is able to detect the eyes of the user automatically. The attempt is to train the system to detect the face of the user prior to detecting the user's eyes. When the face is detected, the eyes' location can be extracted from the face. Since the system is intended for users with mobility difficulties, the user is assumed to have his/her head placed straight in front of the system screen. Head movements and tilting are not considered since users targeted are those with physical paralysis. The system captures images by a webcam with a capacity of 30 frames per second, at a resolution of 640 by 480 pixels.

Artificial Neural Networks (ANN): ANN [9] is a concept inspired from the central nervous system in a human body, as in Fig. 3, that consists of neurons in a similar structure to a network. ANN helps in machine learning and pattern recognition[10-12]. Thus, such networks have proved successful in hand gesture and face recognition. A neural network is trained to detect a face based on common dark spots in any human's face. The advantage of this method is that once the neural network is trained, it can automatically detect any face positioned in front of the camera. Thus, training is required only once.

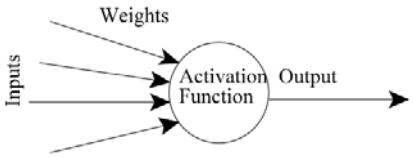

Figure 3. Artificial Neural Network

The neural network used is a two layer (one Hidden layer and one Output layer) feed-forward network trained with the Levenberg-Marquardt and scaled conjugate gradient back propagation algorithm. Thirty four (34) face samples were inserted for training: 27 (80%) samples used for training and 5 (15%) samples used for validation and the rest 2 (5%) used for testing. The hidden layer consists of 20 neurons. The hidden layer works with a sigmoid function and the output layer implements a linear function. The face samples were captured by the webcam in the original 640-by-480 resolution. The training samples consisted of faces of both male and female, adults (aged 20-50) and children (aged 5-8). In order to train the network to detect faces, some pre-processing on the face images was done, as follows:

- First, picture of the face of the training sample is captured in RGB at a resolution of 640-by-480 pixels.
- Captured RGB image is converted to grey scale.
- Adaptive Histogram Equalisation in all (R, G, and B) channels is performed.
- Cropping image to square and resizing image to 25-by-25 resolution (625 pixels).

The 625 grey scale pixel values (ranging between 0 and 255) of each image are normalised and inserted into the input file in a row. Each image occupies one row. As such, the input file for training the network is a 34×625 matrix consisting of 34 samples of 625 elements each. Participants were made to close and open eyes for capturing two images of each participant. With this, the network is trained to detect faces irrespective to whether eyes are opened or closed. The network is trained Supervised Training mode, which means desired output is provided to the network. The desired output values are four that indicate the position of the nose with respect to the eyes, as in Fig. 4.

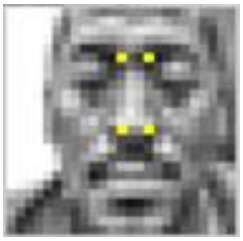

Figure 4. Desired Output of Neuron (shown in yellow dots)

These four output values are desired since the area in the middle of the two eyes can then be segmented and used for skin colour segmentation in the tracking phase. Since the aim is to extract the position of the pixels and not the value, the output is converted from grey scale value to the pixel position by the following formula:

$$Output\ value = [(row\ no - 1) \times 25] + column\ no. \quad (1)$$

The output file is 34×4 matrix consisting of 34 samples of 4 elements each, which were normalised by division with 625. The network architecture consists of 625 input nodes in the input layer, 20 hidden neurons in the hidden layer, and four output neurons in the output layer, as shown in Fig. 5. The network was trained for 1,030 iterations.

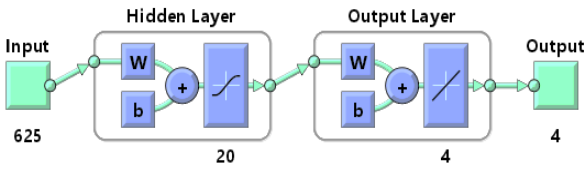

Figure 5. ANN Architecture (left is hidden layer, right is output layer)

Support Vector Machines (SVM): Support vector machines (SVM) have produced good results in pattern recognition and classification. SVMs are implemented when data is to be classified [13]. It searches for a best hyperplane, in Fig. 6, for the largest margin, to separate the two classes [13]. Support vectors are the points closest to the separating hyperplane [13]. The '+' points belong to one class, and the '-' points belong to the other. Mathematically, SVMs are implemented by a set of sequential equations. The separating hyperplane is given as:

$$\langle w, x \rangle + b = 0, \quad (2)$$

where <*w*, *x*> is a dot product of weights and data points, *b* is bias. To achieve best separating hyperplane, the following is satisfied:

$$yi(<w, xi> + b) \geq 1 \quad (3)$$

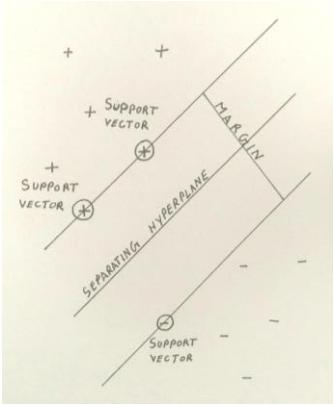

Fig. 6. Separating Hyperplane of SVM

where *y* is ±1 depending on the class. To simplify quadratic equations, Lagrange multipliers α_i are applied with the constraint and subtracted from the condition for maximal margin. Values of α_i lies in the range $0 \leq \alpha_i \leq C$, where C is a constraint which keeps values of α_i within a limited range. On the other hand, for nonlinear transformations, Kernels are often applied since a simple hyperplane cannot separate the two classes of data points apart. The concept of the Kernel method is that there exist a function φ that maps *x to a* linear space S, such that

$$K(x, y) = <\varphi(x), \varphi(y)> \quad (4)$$

This dot product takes place in the linear space S. This concept of kernels is implemented in training the support vector machines for face detection in the calibration. A polynomial kernel of order 3 was used to train the SVM. The input file is a combination of two files. One part is the positive samples consisting of 34 face samples of 625 elements (same as the training samples of the ANN discussed previously), with pixel values inserted column-wise. The other part is the negative samples consisting of 34 samples of 625 elements constituting of random non-face images taken from the internet. The output or grouping file consists of 1's for the positive samples and -1's for the negative samples. Out of the total 68 samples of the input file, 34 samples (half of which are 1's and the other half are -1's) were used for training, while the rest 34 samples were kept aside for validation of the training results.

3 Comparison: Training and Testing

Training: After 1,030 iterations in training the neural network, good results were achieved in terms of the mean squared error (mse), and the regression (i.e., correlation between outputs and targets. For the training samples, a mse of 0.00004633 and regression of 0.994613 were achieved. For the validation samples, a mse of 0.00872 and regression of 0.815132 were reached. For the testing samples, a *mse* of 0.00154 and regression of 0.999999 were achieved. Graphs showing the mse and regression numbers for all samples are in Figure 7 and 8 respectively. As for training SVM, 25 support vectors were used for training. An accuracy of 100% was reached with the 34 training samples; while an accuracy of 0.875 (87.5%) was reached with the 34 validation samples. Values of alpha were calculated during training by the kernel classifier, from which values of the weight matrix were calculated by:

$$w = \sum_i \alpha_i y_i x_i \quad (5)$$

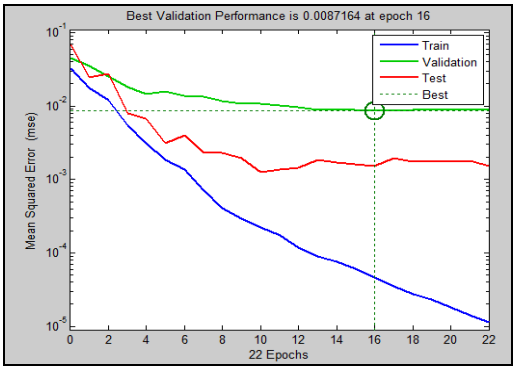

Figure 7. Error values of training, validation and testing samples

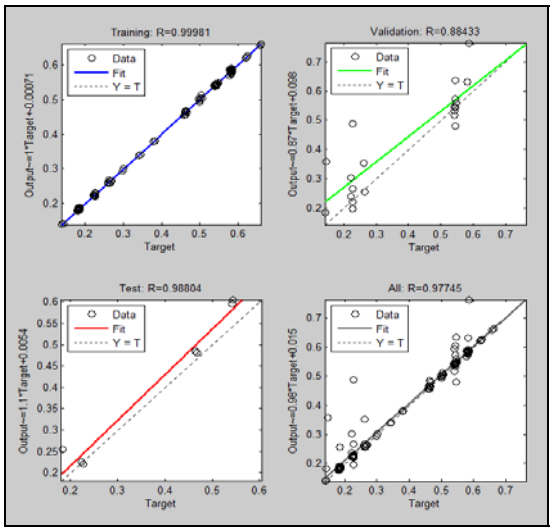

Figure 8. Regression values of training, validation and testing samples

The value of the constraint C affects the face detection results. A number of C values have been tried in training, ranging from 1000 to 0.00000001. When plotting the weight matrix after training with different values of C, it was noticed that lower the value of C, more the weight matrix looks similar to a face. Based on this, C=0.00000001 was selected as the final value for training a face detection SVM in the calibration phase.

Figure 9. Plotting of average weight matrix of SVM

Testing: Testing for both ANN and SVM, was done by inserting a random face image into the system and comparing the results with the desired results. The test image needs to undergo a few pre-processing steps before insertion into both the ANN and SVM, as discussed before in section II. As for the ANN, the following are the results in Table 1.

Table 1. ANN Testing Results

Desired Output	0.2224	0.2256	0.5424	0.5456
Network Output	0.273832	0.276808	0.571432	0.574408
MSE	0.00224493			
Regression	0.99393			

As can be noticed from Table 1, a close precision is achieved but not good enough to detect the desired area in the face. This is because when retrieving back the pixel position, a significant difference is noticed. When applying the formula as given in (1), the network four output points are (6, 21), (6, 23), (14, 7), (14, 9), instead of the desired points (6, 14), (6, 16), (14, 14), (14, 16). The network outputs correct row number, but fails in recognising the correct column numbers which leads to a quite huge difference. In testing the SVM, a few more pre-processing steps are performed on the test image as shown in Fig. 10.

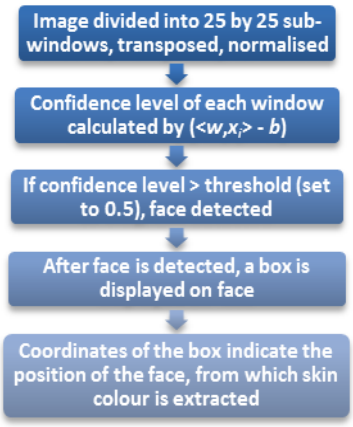

Figure 10. Steps undergone by test image before insertion into SVM

The face in the test image was detected with a confidence level of 0.73888, which is above threshold set (as 0.5). The box coordinates displayed on the detected face in the test image are [10.5000 5.5000 34.5000 29.5000]. Dividing the box width by half gives the exact position in between the two eyes (34.5+10.5 = 22.5). The 22nd column is in between the two eyes, which shows how accurately the box was placed around the detected face.

Comparing Both Approaches: While ANN and SVM are two different approaches, a comparison can be done between them based on some desired features like fast training speed and accurate results. Some features cannot be compared, like number of iterations, since SVM are trained in a different way. Alternatively, the time taken to train both ANN and SVM can be compared. This comparison is summarized in Table 2.

4 Conclusions

It was clear that SVM performed better than ANN in environment of partner-assisted eye scanning. Accurate results were reached in SVM when same training samples were used in both approaches. However, the performance of ANN can be improved because values of mean squared error reached were low, resulting in only producing accurate results in terms of row number. The performance of ANN can be improved by either increasing the number of training face samples, or by varying the training samples, as for instance, varying the lighting conditions for each training sample might produce precise column position. Overall, as a conclusion, the SVM was chosen due to its simplicity of training and good results in face detection prior to eye detection. These results are further used to build an eye tracking keyboard for people with physical challenges.

Table 2. Comparison of the Two Approaches

Feature	ANN	SVM
No. of training samples	34 face samples	68 face and non-face samples
Training accuracy	Samples: 99.4613% training , 81.5132% validation	Samples: 100% (training), 87.5% (validation)
Training time	1,030 iterations (5 sec)	No iterations, weight matrix calculated (0.6 sec)
MSE (testing)	0.00224493 (column position inaccurate)	(column position exact)

References

1. Paralysis, National Health System. Retrieved online from http://www.nhs.uk/Conditions/paralysis/
2. Reeve, C. (2002). One Degree of Separation: Paralysis and Spinal Cord Injuries in the United States. Christopher and Dana Reeve Foundation. Retrieved online from http://www.christopherreeve.org/atf/
3. Human Diseases and Conditions. Paralysis. Retrieved from http://www.humanillnesses.com/original/Pan-Pre/Paralysis.html
4. ALS Association. Online: http://www.alsa.org/about-als/what-is- als.html
5. Cincinnati Children's Hospital Medical Center. Communicating with Partner Assisted Scanning. Retrieved from http://www.youtube.com/watch?v=nGpSXQKrmR4
6. Retrieved online from http://www.surveymonkey.com/s/low-tech_AAC
7. Magrelli, S.; et al, "A Wearable Camera Detects Gaze Peculiarities during Social Interactions in Young Children with Pervasive Developmental Disorders," *IEEE Transactions on Autonomous Mental Development*, **6**(4), (2014), 274-285.
8. Xingjie W.; et al, "Dynamic Image-to-Class Warping for Occluded Face Recognition," *IEEE Transactions on*

Information Forensics and Security, , **9**(12), (2014), 2035-2050

9. Franceschini, N., "Small Brains, Smart Machines: From Fly Vision to Robot Vision and Back Again," *Proceedings of the IEEE* , **102**(.5), (2014), 751,781
10. Memon, Q., Shuja, M., "Crime investigation and analysis using neural nets," *IEEE Multi Topic Conference*, (2003), 346-350
11. S Khan, Q Memon, "Artificial Neural Network Approach to Camera Calibration and 3-D World Reconstruction for stereovision*," International Workshop on Recent Advances in Computer Vision*, (1998), 35-40
12. Memon, Q.; "Relation Based Clustering of Wear Particle Measurements for Industrial Automation", *International Journal of Automation and Control*, **1**(2-3), (2007), 207-219
13. Schölkopf, B., et al, "Estimating the Support of a High-dimensional Distribution," *Journal of Neural Computation*, **13**(7), (2001), 1443-1471.