International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 11, No.1 (Feb-2022)

https://dx.doi.org/10.12785/ijcds/110168

Assistive Technology to Communicate Through Eye Blinks: A Deep Learning Approach

Srivatsan Sridharan[1], Nirmal V[1], Sachin Soundar[1] and Maheswari N[1]

[1]*School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India*

Received 19 July 2021, Revised 26 January 2022, Accepted 28 January 2022, Published 15 February 2022

Abstract: Paralysis is the loss of muscle control in parts of or the whole body. Only 15% of people affected by paralysis are employed, due to concerns over their ability to function properly. These patients are able to think and have ideas just as well as other people but are unable to communicate these thoughts to their full extent because of a lack of an affordable medium of communication. In this paper, we propose a system for paralysis patients to communicate through Morse encoded eye blinks which can be decoded through any device equipped with a webcam. It provides a novel, affordable and practical way of communicating the entire English vocabulary along with a smaller learning curve, which is not provided in other methods. The detected pattern of blinks is converted into text for human understanding. It is compared with current AAC (Augmentative and Alternative Communication) devices and standard blink detection techniques and the results are provided for the deep learning solution on a standard dataset.

Keywords: Deep Learning, AAC, Computer Vision

1. Introduction

Paralysis affects 1 in 50 people in the United States of America, which translates to roughly 5.4 million people according to the Christopher and Dana Reeve Foundation. Paralyzed people live all around the globe with their crippled bodies but functioning brains. Numerous communication strategies have been proposed which are excessively expensive and inaccessible to patients, leaving them with a consistent battle to convey their requirements. The inability to express their thoughts to the fullest extent also takes a toll on mental health. Most non-technological strategies for paraplegics involve limiting their communication potential to a few common words and phrases(Hungry, Restroom, etc.) It might also involve a simple Yes/No based question-answer framework. In the case of retaining partial control over their fingers, they may be able to signal using the fingers through a keyboard. Digital solutions can range from being low tech, where a laser pointer is pointed on a range of common words and the patient signals when it points to their desired word, to being more innovative and high tech. The most prevalent high-tech solutions in the market are Speech Generating Devices(SGD's) which are extremely expensive and involve analyzing the breath of the user. Electroencephalography devices involve analyzing the brain waves of the user to understand if/what he is trying to communicate. As can be expected, these devices are expensive too, ranging in prices from $5000-$15000. It has been observed that most patients are able to retain control over their eye gaze and blinks. This opened up avenues to explore, related to communication through eye blinks [1]. Eye blinks initially were limited to being used just for yes or no answers. Eye gaze tracking [2]was also proposed as a means for Human-Computer Interaction. This was later advanced into the assistive technology domain to help as a means of communication. Hardware solutions exploiting the use of eye gaze and blinks suffer from the same disadvantage as their predecessors of being expensive and not easy to access [3]. Software solutions, on the other hand, do not have this issue and the rise of high-level programming languages has made it easier to program custom-built software for specific use cases. The rise and prevalence of Deep Learning in everyday life, has also entered the field of assistive technology. Eye-tracking as a tool is now more accessible than ever, and is growing in popularity amongst researchers from a whole host of different disciplines, and has the potential to become an even more important component in future perceptual user interfaces [4]. The technique is used in psychology, cognitive science, human-computer interaction, advertising, and other areas. Today, human eye-gaze, blinking, and eye movement can be recorded with high efficiency and accuracy even with very cheap and accessible equipment. Recent significant advances in Machine Learning and Computer Vision have allowed software to be custom-built for the purposes of eye tracking and blink detection and run on many mobile devices with low latency during run time with minimal

E-mail address: :s.srivatsan2017@vitalum.ac.in, nirmal.v2017@vitalum.ac.in, sachin.soundar2017@vitalum.ac.in, maheswari.n@vit.ac.in

CPU load due to lightweight models [5]. Eye Blinks can be used by paralysis patients, or anyone wanting an alternative means of communication due to the inability to converse using traditional methods, as an efficient and effective way of communicating. It does not limit the user's vocabulary to the common phrases generally associated with paraplegics and other disabled people. Instead, it opens them up to the entire dictionary and provides unlimited freedom in communicating. Encoding the blinks in Morse code allows the user to express each letter in the English Alphabet and in turn allows them to communicate anything they would normally have been able to communicate.

The rest of the paper is structured as follows. Related work is reviewed in Section 2, and some background information on the frameworks and concepts used is seen in Section 3. In Section 4, the proposed system is described in detail. The experiments and results obtained are discussed in Section 5. Conclusions and Applications are viewed in Section 6.

2. Related Work

Using modern everyday technology for unique image classification and recognition tasks is an area with booming research. Deep Learning techniques utilizing a Convolutional Neural Network have been the most efficient and accurate methods so far. This paper utilizes a FaceMesh model for facial landmarking, proposed by Kartynnik et al [6]. The model was built on top of the BlazeFace face detector, a lightweight face detection model proposed for mobile devices in [7]. The model is further explained in the later stages of this paper. Studies have used CNN's for fatigue detection from eye state. Notable improvements have been made by using projection cores along with a Fatigue Detection CNN on a standard closed eyes dataset [8]. Sanyal et al [9] have used a two-stream CNN which uses a novel eye localization technique and takes in an eye patch and a masked eye patch to predict the state of the eye and the number of blinks. They had used a Finite State Machine (FSM) to model the problem of checking whether the eyeblink falls into a 3-12 frame range for a 30 fps camera. A. J. Molina-Cantero et al have summarized blink-based communication methods in their definitive work on how blinks are used for communication by disabled people [10]. Blink detection techniques can be divided into vision-based and non-vision-based methods. Non-Vision Based methods involve the use of hardware monitoring to analyze bioelectrical signals for the related input and to confirm a blink. Some of these methods include Electroencephalography and Electrooculography. Electrooculography involves recording the potential difference using electrodes placed on the skin near the eyes, with blinks corresponding to spikes in the readings [11]. In [12], 4 types of physiological signals were analyzed, including EEG, EOG, ECG, and an alcohol sensor, in order to determine the sobriety of the user. It shows how monitoring EOG and EEG can be useful tools. There is a plethora of research on vision-based blink detection methods. Vision-based can be further split into hardware methods and software methods [11]. Hardware methods involve detecting the corneal reflection of infrared light. Al-Gawwam et al [13] trained automatic facial landmark detectors which were able to handle different lighting conditions and head orientations. It analyzed the vertical distance between the eyelids to estimate the state of the eye. The Viola-Jones algorithm has been used in many studies for detecting eyes. Background subtraction is a method that can be used for detecting eye blinks [14]. Harr transform methods have been used for the purpose of eye detection/localization [1], [15]. Motion vectors can be used to sense the closing of the eye and register the blink [15]. Frame differencing to localize the region of optical flow is an innovative technique that was used for the purpose of blink detection [16]. [17] used unsupervised learning methods in order to analyze the eye for any hidden patterns which could lead to useful information regarding the state of the subject. in [18], the shortcomings of the Eye Aspect Ratio method were discussed and a new method EARM was proposed and found to have better performance. Augmentative and Alternative Communication methods are used typically by paraplegics and other disabled people unable to communicate verbally or through typical sign language/hand signals. Monitoring brain signals through EEG with a spike corresponding to the person making a choice on a GUI was studied in detail [19]. Morse Code was used as an alternative input method for disabled persons to move their wheelchairs and improve their driving efficiency [20]. Using blinks to choose input on a keyboard and hence enabling the person to communicate was seen and tested as a viable option [1]. Using eye motion detection was proposed by Pandey et al [21] with eye motion for navigating through a GUI and blinks being used to select a choice. Blinks being mapped to a binary vector and decoded into a pre-set dictionary of words was proposed and testing different ConvNet architectures for this task was done recently by [22]. But it only allows access to the preset dictionary and not the entire language. A text entry interface using tongue gestures encoded in Morse code was explored in [23]. They also showed how input speed can scale with the user's level of expertise. Morse code entered through blinks was proposed with the authors using an infrared sensor to detect the blink and duration of it [24]. The blinks were mapped to dots and dashes in order to decode into English. This solution is hardware-based and suffers from the same issues seen earlier, that of ease of access and use. A software solution based on the same concept is proposed by Kranthi Kumar et al [25] using Eye Aspect Ratio to detect the state of the eye. The use of a CNN was observed to be more accurate than EAR in classifying the state of the eye in [26], where they built an eye-care monitoring system to detect if a user blinks sufficient times during the usage of his/her computer, to prevent Computer Vision Syndrome.

3. Background on Technologies and Frameworks

OpenCV is a research project from Intel and now available free as a toolkit of functions used for image processing applications. Originally written in C++, it now

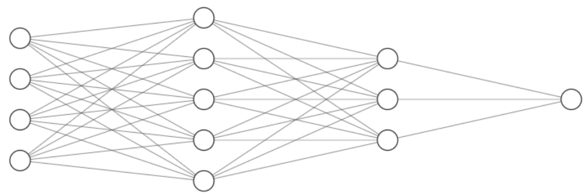

Figure 1. Neural Network with 4 input nodes, 2 hidden layers, and an output layer with 1 node

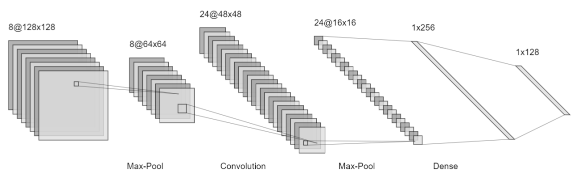

Figure 2. Convolutional Neural Network

has interfaces in Python, MATLAB, and Java too. It has functions related to image processing, image transformation, and object tracking along with others.

Neural Networks are a network of nodes, like in Figure 1, that are initialized with random weights and biases. These nodes calibrate their weights as they train to get better at the required task, either prediction tasks or classification tasks.

Convolutional Neural Networks, like in Figure 2, are specialized in image classification and object detection tasks as they have the presence of convolutional layers that can manipulate and transform the input image using kernels. Deep Learning Frameworks have risen in popularity as they have removed the need for hardcoding the training and learning algorithms, using inbuilt functions instead to do the same.

MediaPipe FaceMesh [27] is an open-source framework designed for face detection and facial landmark localization tasks. It consists of a face detector model and a facial landmark model. The face detector model, BlazeFace, is described in [7] as a ConvNet with the initial 2D convolution layer, followed by 5 BlazeBlocks and 6 double BlazeBlocks, where a BlazeBlock and a double BlazeBlock are as seen in Figure 3 and Figure 4. MobileNet [28] and MobileNetV2 [29] are neural network architectures meant to scale down and perform well on mobile and edge devices. MobileNetV2 consists of inverted residual structures inspired from ResNet and performs well on image recognition tasks with fewer parameters than other state-of-the-art networks. EfficientNet [30] was proposed as a highly efficient family of convolutional neural networks and scaling method giving state-of-the-art performance at smaller sizes. It starts at B0 and scales up till B7.

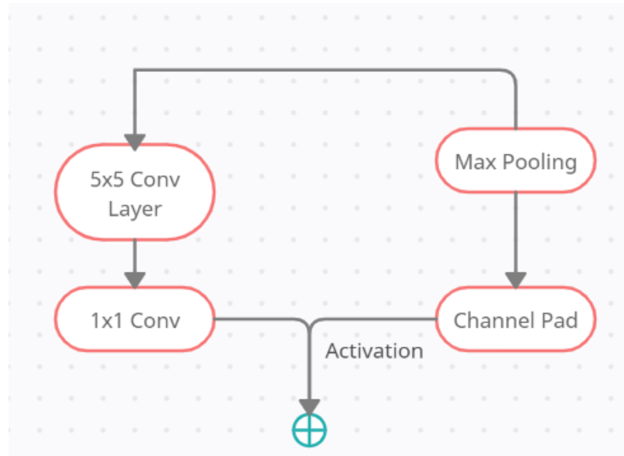

Figure 3. Single BlazeBlock

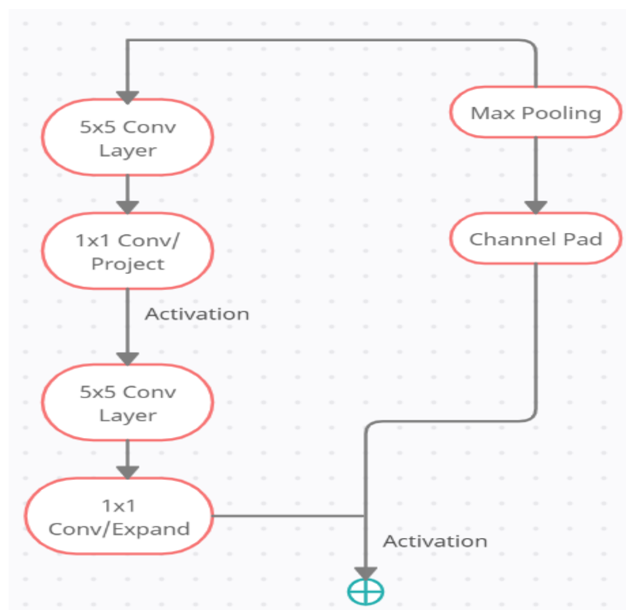

Figure 4. Double BlazeBlock

The facial landmarking model described in [6] takes in the bounding boxes predicted by BlazeFace and outputs a prediction vector of the landmark coordinates, which are mapped back onto the input image. It was trained to output 468 facial landmarks, and to work optimally on mobile and edge devices, which suits the target use case.

Morse Code, seen in Figure 5, is a method used to encode characters from the English language into a standardized pattern of dots and dashes.

4. Proposed System

We propose a system capable of detecting blinks and recording the duration of the blink. The blinks are assumed to be encoded in Morse Code, with a short blink representing a dot and a long blink representing a dash. Due

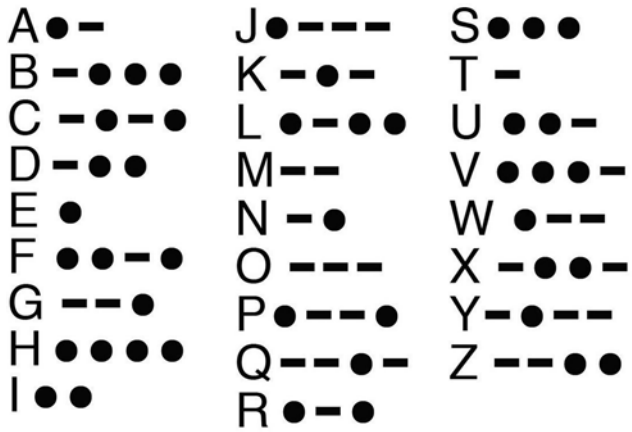

Figure 5. Morse Code

to this approach, our proposed work is able to capture the entirety of the English vocabulary through blinks, providing a solution better than other assistive technologies, which have limited vocabulary options. The blinks are converted to dots and dashes on the fly and added to the buffer. On encountering the Decode signal (a left wink), the pattern in the buffer is decoded into English if the pattern was available in the pre-defined dictionary. If not, it gives an undefined pattern error message and clears the buffer. The flow is defined in Figure 6.

The work consisted of the following parts:

1) *Extraction*: Individual frames were extracted from the streamed video. The facial keypoint model was applied to obtain the landmark coordinates and were mapped onto the original image.

2) *Cropping*: The left and right eye image areas were cropped from the whole image, by using their respective coordinates.

3) *Training*: A ConvNet was trained to classify eye images as open or closed.

4) *Classification*: The cropped eye images were fed into the trained ConvNet to classify their states.

5) *Tracking*: The Duration of the blink was tracked in terms of the number of frames. A blink lasting more than frames was a dash, and other blinks were a dot.

6) *Decoding*: The dots and dashes are added to the buffer. Encountering a left wink would result in the buffer being decoded into the English language.

A. FaceMesh Model

The eyes are detected using the FaceMesh model. It is a 12 layer model with an initial convolutional layer, followed by 5 single and 6 double BlazeBlock layers. However, the model is extremely lightweight and optimized for use on mobile devices. But a drawback of this model is the inability to distinguish between closed and open eyes. Hence, the need was seen for extraction and a separate model for distinguishing eye states. For the proposed system, only the eye areas are required. The image is cropped using slicing, where the 4 corner coordinates of the Region of Interest are extracted from each frame and sliced out. The coordinates were taken from two points, the top left and bottom right landmark coordinates for both eyes. For the right eye, this was points 441 and 448 and for the left eye, it was points 225 and 232. The cropped image is resized into a (26,34) shape.

B. Convolutional Neural Network

Taking into account the weaknesses of facial keypoint models, with respect to being unable to identify closed and open eyes, and also individual classification of eye states, we propose a secondary CNN method to combat this shortcoming. Since the landmarking model is able to identify the separate eye positions on the frame, the eye images are cropped, passed, and classified separately by a trained CNN model, to differentiate blinks, left and right winks. Due to the simplicity of the task, the model is expected to perform remarkably and combine well with the facial keypoint model for the required problem. For the model, the input layer takes gray images of the shape (26,34). The model then consists of 2 Convolutional Layers, each with a kernel of size 3x3 and a stride of 1. Each convolutional layer is followed by a MaxPooling layer which takes the max value in the kernel's neighborhood. The kernel/pool used is of size 2x2. The final Convolution layer feeds into a Dense layer of dimensions 256. Each of the previous layers, Convolutional and Dense, uses a ReLU activation function. The final output layer has only one output node, with the probability of the eye being open or closed. The final layer uses a sigmoid activation function. The layers are seen in Table I.

The model gets fed the cropped eye images and predicts their individual states. If both eyes are open, it continues tracking on the stream. If both eyes are closed in a frame, it is a blink and the duration of the closure is tracked. A closed eye pair for more than 5 frames will be added as a dash to the buffer whereas all other blinks will count as dots. The threshold of 5 frames was set after testing on a significant number of subjects at varying frame thresholds, with 5 providing the best performance for the system. Since each eye is classified separately as being closed or open, the system will be able to classify winks too. A closed left eye is the decode signal, to convert the pattern of dots and dashes in the buffer to its corresponding English alphabet. A closed right eye is a clear buffer signal, to clear the buffer in case of any mistakes when blinking. The system was tested using different subjects for the time taken to input a word and for its precision, along with other metrics.

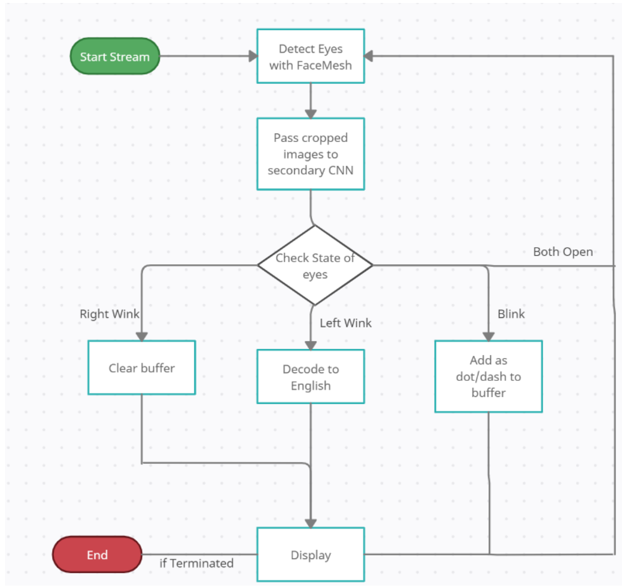

Figure 6. Process Flow

TABLE I. Convolutional Neural Network Layers

Layer	Input Size	Output Size
Input Layer	26,34,1	26,34,1
Conv2D	26,34,1	26,34,32
MaxPooling2D	26,34,32	13,17,32
Conv2D	13,17,32	13,17,64
MaxPooling2D	13,17,64	6,8,64
Flatten	6,8,64	3072
Dense	3072	256
Activation	256	256
Dense	256	1
Activation	1	1

5. Experimental Results

For comparison of different architectures to choose the best possible model for facial landmarking, three different models were tested on the Kaggle Facial Keypoint detection dataset. It was augmented using techniques like shifting and rotation to improve dataset quality. The dataset included 7049 images, which were split as 5639(80%) for training, 1410(20%) for validation, and a separate set of 1784 images for testing. The models tested were EfficientNet-B0, MobileNetV2, and FaceMesh. Seen in Figure 7 and Figure 8 are the results from testing the models on the dataset.

MobileNetV2 achieved 85% and EfficientNet-B0 achieved 87%, while FaceMesh had 95.6% accuracy. The secondary CNN model was trained on the MRL Eye Dataset which contains 84898 images of eyes obtained from the wild and under varying conditions, including lighting effects, spectacles, reflections, etc. We trained our model with a subset of the dataset, using 2874 images. The network was then tested on 1000 images. The distribution was equal, with the same number of samples for closed and open eyes. Samples from the dataset are seen in Figure 9.

All models were trained on Google Collab with a cloud GPU. The deep learning framework used was Keras and it achieved a validation accuracy of 99.96% on the cross-validation dataset which was a part of the original dataset. This is to be expected as the task is a very basic classification task. The model accuracy v epoch plot is seen in Figure 10.

The FaceMesh model's face detection algorithm can handle different orientations of the face on the video and map the facial points accurately every time, with a reported error rate of 3.96% on frontal face images which is the target use case [6]. As compared with other standard landmarking techniques and frameworks like dlib, the model was found to perform better and gave accurate consistent outputs. The system requires the subject to be the primary focus of the camera, in normal lighting conditions, for the

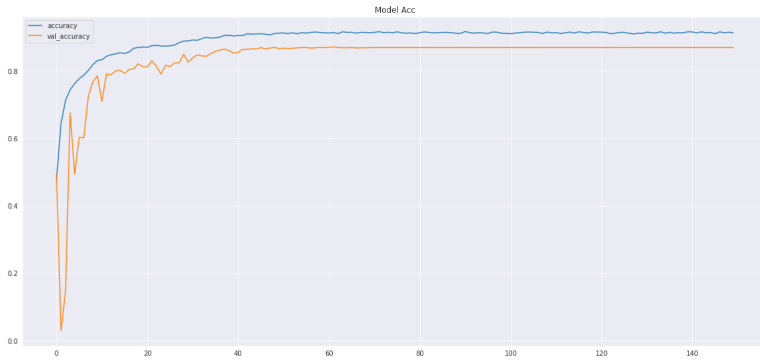

Figure 7. Mobile Net Accuracy v Epoch

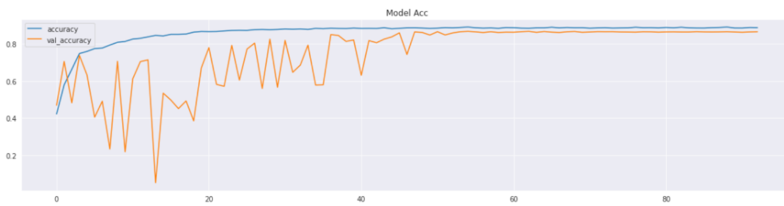

Figure 8. Efficient Net Accuracy v Epoch

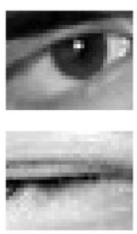

Figure 9. Samples from Eye Dataset

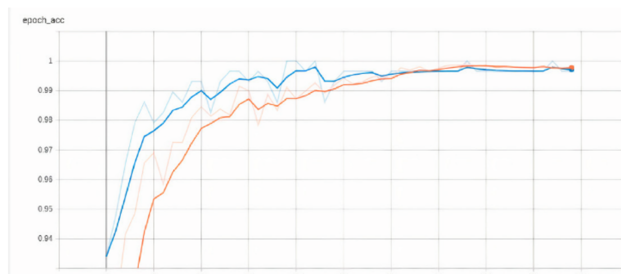

Figure 10. Training(Orange) and Validation(Blue) accuracy for the secondary CNN

FaceMesh keypoint detector to work optimally. OpenCV was used to crop the left and right eye images, as seen in Figure 8, and feed them individually into the network. The model gave out individual predictions for both eyes. The states were identified and tracked. When a left wink was detected, the pattern blinked so far was decoded and added to the current sentence. It also clears the buffer so the next letter could be typed. Since the maximum possible length of dots and dashes for a letter is 4, any patterns beyond that length will automatically be discarded and the buffer will be cleared, so the user has to type the letter again. The method was tested on the EyeBlink8 dataset, and compared against the results of other existing methods. The results for the EyeBlink8 dataset are given in Table II. The system was tested by 35 people with each person having to type in 10 words of varying lengths. It was decided to measure the system for its accuracy, by analyzing the detected blinks with the ground truth. The system detected 1734 out of the actual 1748 blinks, giving a detection accuracy of 99.19%. The measured blinks and their classifications are given in Table III, along with the ground truth values.

$$
\begin{aligned}
ClassificationAccuracy &= \frac{TruePositive + TrueNegative}{TotalBlinks} \\
&= \frac{870 + 724}{1734} \\
&= 91.9\%
\end{aligned} \tag{1}
$$

The system was also tested by the same subjects for the same words by using alternate methods of Eye Aspect Ratio(EAR) and standard CNN's instead. The accuracy and precision when using these methods were found to be lower than when using the proposed method to detect the blink. This was because in a few cases, it did not detect actual blinks as they failed to meet the required threshold. The confusion matrix for measured blinks using EAR and CNN are given in Table IV.

TABLE II. Results on Eyeblink8

Method	Precision(%)	Recall(%)	F1 Score(%)
Motion Vector Analysis [31]	94.69	91.91	93.27
Completeness Detection [32]	X	X	91.3
EfficientNet + CNN	92.73	89.7	91.19
MobileNetV2 + CNN	91.2	92.3	91.74
FaceMesh + CNN	95.64	94.25	94.93

TABLE III. Confusion Matrix for FaceMesh + CNN

	Detected as Short Blink	Detected as Long Blink
Actual Short Blink	870	47
Actual Long Blink	93	724

TABLE IV. Confusion Matrix for EAR + CNN

Method		Detected as Short Blink	Detected as Long Blink
EAR	Actual Short Blink	810	43
	Actual Long Blink	118	689
CNN	Actual Short Blink	835	54
	Actual Long Blink	105	753

TABLE V. Metrics Recorded

Metrics	Metric Value
CNN accuracy	99.96%
F1 Score	94.93%
Word Precision	91.1%
Blink Detection Accuracy	99.19%
Facial Keypoint Accuracy	95.6%
Latency	Dot - 0.5 seconds , Dash - 1 second
EyeBlink8	94.93%

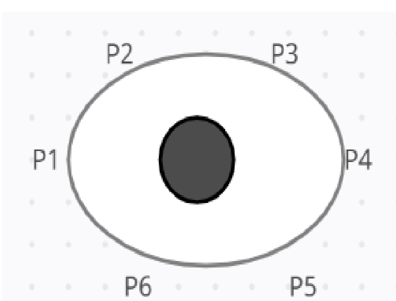

Figure 11. EAR Points

For the EAR method, seen in Figure 11, the dlib library was used for facial landmarking. It computes the ratio between upper and lower eye point positions and classifies as blink if the ratio falls below a threshold of 0.3.

$$EAR = \frac{|p_2 - p_6| + |p_3 - p_5|}{2 * |p_2 - p_4|} \quad (2)$$

The standard CNN method involved just checking the frames directly for open and closed eyes. As seen in Table 4, the total number of blinks has reduced since the system when using EAR was unable to detect actual blinks, or misclassified many long blinks as short blinks. We also measured the word precision of the system as the number of words and letters typed using the system correctly divided by the number of words typed correctly and incorrectly. The system achieved a precision value of 92.3%.

$$\begin{aligned} WordPrecision &= \frac{CorrectWords}{CorrectWords + IncorrectWords} \\ &= \frac{319}{350} \\ &= 91.1\% \end{aligned} \quad (3)$$

We have measured the time taken to type in every letter individually. Time taken was measured to average 0.5 seconds for a dot(short blink) and 1 second for a dash(long blink) with the Decode signal taking another 0.5 seconds. These values might vary depending on the person and their level of control over their eye blink. As an example, the word "HELP" was observed to take 8 seconds to type. Table V details all the metrics recorded.

Drawbacks noticed by users were regarding eye fatigue from constant blinking and the higher learning curve for Morse encoded blinks. The higher learning curve can be overcome by displaying the Morse chart on screen while waiting for input, so users are not required to memorize Morse code before using the application. In the cases of users experiencing eye fatigue, a simulation of a face was provided which can be controlled using minor gestures by the fingers. This face can be projected and recognized by the system, hence protecting the eyes from Computer Vision Syndrome too. However, it might not be useful for patients who have no motor control over their fingers. The option of a rolling keyboard is also explored, where a cursor moves across the rows and columns of an on-screen keyboard and a blink, irrespective of the type, would mean the user has chosen the letter the cursor is currently highlighting on the keyboard. This option was found to have an extremely small learning curve, as opposed to the Morse encoded blinks, but it comes at the cost of speed, since tests revealed the Morse method to be significantly faster for input.

6. Conclusions and Future Work

This paper summarizes our work on developing a system, with the disabled and paralyzed in mind, to communicate through Morse encoded eye blinks. The FaceMesh model was used to map facial features and the OpenCV library was used to crop out the eyes. The eye images were passed through a Convolutional Neural Network to determine the state and the duration of the blinks were tracked to classify the blink as a dot or dash. This pattern was decoded into English for ease in communication with other people. It is a cheap and efficient software alternative as compared to existing expensive and inaccessible hardware solutions. The system's accuracy will also scale with the users' level of expertise. Future work can include support for multiple languages, gaze tracking with the implementation of a GUI for further functionality, and integration with voice assistants like Alexa and Google Now.

References

[1] A. Królak and P. Strumiłło, *Eye-Blink Controlled Human-Computer Interface for the Disabled.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 123–133.

[2] C. H. Morimoto and M. R. Mimica, "Eye gaze tracking techniques for interactive applications," *Computer vision and image understanding*, vol. 98, no. 1, pp. 4–24, 2005.

[3] H. Singh and J. Singh, "Human eye tracking and related issues: A review," 2012.

[4] P. Majaranta and A. Bulling, *Eye Tracking and Eye-Based Human–Computer Interaction*. London: Springer London, 2014, pp. 39–65.

[5] H. Young-Joo, W. Kim, and J. Park, "Efficient eye-blinking detection on smartphones: A hybrid approach based on deep learning. mob," *Inf. Syst*, vol. 2018, p. 6929762, 2018.

[6] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-time facial surface geometry from monocular video on mobile gpus," *arXiv preprint arXiv:1907.06724*, 2019.

[7] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann, "Blazeface: Sub-millisecond neural face detection on mobile gpus," *arXiv preprint arXiv:1907.05047*, 2019.

[8] R. Huang, Y. Wang, and L. Guo, "P-fdcn based eye state analysis for fatigue detection," in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*. IEEE, 2018, pp. 1174–1178.

[9] R. Sanyal and K. Chakrabarty, "Two stream deep convolutional neural network for eye state recognition and blink detection," in *2019 3rd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*. IEEE, 2019, pp. 1–8.

[10] A. J. Molina-Cantero, C. Lebrato-Vázquez, M. Merino-Monge, R. Quesada-Tabares, J. A. Castro-García, and I. M. Gómez-González, "Communication technologies based on voluntary blinks: Assessment and design," *IEEE Access*, vol. 7, pp. 70 770–70 798, 2019.

[11] X. Kong and G. F. Wilson, "A new eog-based eyeblink detection algorithm," *Behavior Research Methods, Instruments, & Computers*, vol. 30, no. 4, pp. 713–719, 1998.

[12] N. Kurian and D. Rishikesh, "Real time based driver's safeguard system by analyzing human physiological signals," *Int. J. Eng. Trends Technol*, vol. 4, pp. 41–45, 2013.

[13] S. Al-gawwam and M. Benaissa, "Robust eye blink detection based on eye landmarks and savitzky–golay filtering," *Information*, vol. 9, no. 4, p. 93, 2018.

[14] T. S. Sharmila, R. Srinivasan, K. Nagarajan, and S. Athithya, "Eye blink detection using back ground subtraction and gradient-based corner detection for preventing cvs," *Procedia Computer Science*, vol. 165, pp. 781–789, 2019.

[15] N. Alsaeedi and D. Wloka, "Real-time eyeblink detector and eye state classifier for virtual reality (vr) headsets (head-mounted displays, hmds)," *sensors*, vol. 19, no. 5, p. 1121, 2019.

[16] T. Bhaskar, F. T. Keat, S. Ranganath, and Y. Venkatesh, "Blink detection and eye tracking for eye localization," in *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, vol. 2. IEEE, 2003, pp. 821–824.

[17] W. Han, Y. Yang, G.-B. Huang, O. Sourina, F. Klanner, and C. Denk, "Driver drowsiness detection based on novel eye openness recognition method and unsupervised feature learning," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2015, pp. 1470–1475.

[18] A. Kuwahara, R. Hirakawa, H. Kawano, K. Nakashi, and Y. Nakatoh, "Blink detection using image processing to predict eye fatigue," in *International Conference on Human Interaction and Emerging Technologies*. Springer, 2020, pp. 362–368.

[19] L. Rahman, J. Adan, Q. M. Billah, M. K. Islam, A. M. Kamal, S. Hossain, C. Shahnaz, and A. Subhana, "A low cost human computer interface for disabled people based on eye blink detection using brain signal," in *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*. IEEE, 2019, pp. 1937–1941.

[20] H. Ka and R. Simpson, “Effectiveness of morse code as an alternative control method for powered wheelchair navigation,” in *Proceedings of the Annual Conference on Rehabilitation Engineering*. RESNA, 2012.

[21] M. Pandey, K. Chaudhari, R. Kumar, A. Shinde, D. Totla, and N. Mali, “Assistance for paralyzed patient using eye motion detection,” in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE, 2018, pp. 1–5.

[22] A. A. Ramli, R. Liu, R. Krishnamoorthy, I. Vishal, X. Wang, I. Tagkopoulos, and X. Liu, “Bwcnn: Blink to word, a real-time convolutional neural network approach,” in *International Conference on Internet of Things*. Springer, 2020, pp. 133–140.

[23] L. R. Sapaico and M. Sato, “Analysis of vision-based text entry using morse code generated by tongue gestures,” in *2011 4th International Conference on Human System Interactions, HSI 2011*. IEEE, 2011, pp. 158–164.

[24] K. Mukherjee and D. Chatterjee, “Augmentative and alternative communication device based on eye-blink detection and conversion to morse-code to aid paralyzed individuals,” in *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*. IEEE, 2015, pp. 1–5.

[25] K. Kumar, V. S. Srikar, Y. Swapnika, V. S. Sravani, and N. Aditya, “A novel approach for morse code detection from eye blinks and decoding using opencv,” *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 8, 2020.

[26] G. Worah, A. Khan, M. Kothari, and M. Naik, “Monitor eye-care system using blink detection—a convolutional neural net approach,” *Int. J. Eng. Res. Technol*, vol. 6, pp. 12–16, 2017.

[27] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee *et al.*, “Mediapipe: A framework for perceiving and processing reality,” 2019.

[28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[30] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[31] A. Fogelton and W. Benesova, “Eye blink detection based on motion vectors analysis,” *Computer Vision and Image Understanding*, vol. 148, pp. 23–33, 2016.

[32] A. Fogelton and W. Benesova, “Eye blink completeness detection,” *Computer Vision and Image Understanding*, vol. 176, pp. 78–85, 2018.

Srivatsan Sridharan Srivatsan completed his B.Tech Computer Science degree from VIT University, Chennai. His research interests include AI for Healthcare, Data Privacy, and Deep Learning.

Nirmal V Nirmal V completed his B.Tech Computer Science degree from VIT University, Chennai. He currently works at Sembley as a Data Scientist. His research interests include NLP, Knowledge Graphs, and Graph Neural Networks.

Sachin Soundar Sachin Soundar finished his B.Tech degree in Computer Science from VIT University, Chennai. He is passionate about AI and Machine Learning and has previous internship experiences in these fields

Maheswari N N.Maheswari is a Professor in the School of Computer Science and Engineering at Vellore Institute of Technology (VIT), Chennai, India. She has 21 years of academic experience. Her research interests include Machine Learning and Analytics. She has authored and co-authored a large number of research articles in journals. She serves as a reviewer in national/ international conferences and journals.