MANTECH Publications

Journal of Research in Computer Science and Engineering
Volume 6 Issue 3
Online ISSN- 2457-0818

A Review on Artificial Neural Network and Sign Languages Detection Using Deep CNN

Al Mahmud Al Mamun[1], Md. Rakibul Hasan[2*], Md. Mokhlesur Rahman[3], Md. Mahbubur Rahman[4], Sondip Poul Singha[5], Md. Yasin Khan[6]

Islamic University (IU), Bangladesh[1, 2], Universiti Kebangsaan Malaysia[3], University of Business and Technology (BUBT), Bangladesh[4, 5, 6]

Corresponding Author's E-mail: *rhp.rakibul7@gmail.com[2*]*

Abstract

Artificial Neural Networks is a computational model inspired by the structures and functions of biological neural networks. Artificial neural networks are used in all sectors of the recent world with its life-changing applications. In this paper, we review artificial neural networks. We specially focus on the history, architecture, learning technique of the artificial neural networks. Sign language is a visual communication system between hearing-impaired or hard of hearing people. The communication foundations are based on finger shapes, hand shapes, hand movements in space concerning the body, hand orientations, and facial expressions. In this paper, we have proposed a Deep Convolutional Neural Network to recognize the hand gestures. Our system uses the pre-processed images for training and is capable of extracting signs from video sequences under minimally cluttered and dynamic backgrounds using skin color segmentation. In our method, 44 signs of the five fingers are defined. The static images have been pre-processed as grayscale images and the model is trained with 100 numbers of images for each sign. Our proposed model has achieved an accuracy rate of 95.6 % for the five test images.

Keywords: *Artificial Neural Networks, Deep Convolutional Neural Networks, Hand gestures, Sign language recognition*

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

INTRODUCTION

ARTIFICIAL Neuron Networks (ANN) is the basis of artificial intelligence (AI) to solve logical problems that are difficult for humans or statistical standards. It has self-learning capabilities like a human child that help it to provide more effective results.

ANNs have been developed as generalizations of mathematical models of human neural biology. The ANNs have been applying in all the areas of operations, such as - Email service, asset management, credit rating, e-commerce services, chatbots, deep learning, etc.

In section-II of this paper, we review artificial neural networks. In this section, we specially focus on the history, architecture, learning technique of the artificial neural networks. In section-III of this paper, we discussed the application of ANNs and the related work of several image processing and sign language detection and achieved results.

Sign language is a visual communication system between hearing-impaired or hard of hearing people. The communication foundations are based on finger shapes, hand shapes, hand movements in space concerning the body, hand orientations, and facial expressions.

In section-IV of this paper, we have proposed a model for Sign language recognition using Deep Convolutional Neural Networks [1]. Our proposed system will be easier to use and more user friendly for hard of hearing people. The system converts symbolic phoneme signs into equivalent text accurately and efficiently. The aim of this research is to achieve complete knowledge on Artificial Neuron Networks and develop a sign language recognition system for the English phonemes.

We used the Python programming language and OpenCV software library. Python [2] is an interpreter, high-level and general-purpose programming language aimed to write clear, logical code for small and large-scale projects. OpenCV (Open Source Computer Vision Library) [3] is an open-source library of programming functions especially aimed at real-time computer vision and machine learning.

We experiment our proposed model with real data and analyze the result. Our proposed model has achieved an accuracy rate of 95.6 % for the five test images. The error rate of our model is only 4.4. We

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

have tried to develop a system so that digits can be recognized in an easy, fast, and accurate way. We have tried to be dynamic so that it can be used in different critical situations.

A REVIEW ON ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANN) is a computational model inspired by structures and functions of biological neural networks [4]. It is an information processing system that has a collection of connected units (artificial neurons). ANNs have been developed as generalizations of mathematical models of human neural biology.

ANN is the basis of Artificial Intelligence (AI) to solve logical problems [5] that are difficult for humans or statistical standards. It has self-learning capabilities like a human child [6] that help it to provide more effective results.

ANNs are used in all sectors of the recent world with their life-changing applications. The ANNs have been applying in all the areas of operations [7], such as - Email service, asset management, credit rating, e-commerce services, chatbots, deep learning, etc.

A. History of ANN

Artificial Neural Networks was introduced by Warren McCulloch and Walter Pitts in 1943 [8] with the computational model of neural networks. They proposed the first artificial neuron [9] known as Threshold Logic Unit (TLU), or Linear Threshold Unit (LTU). The neuron model was a computational model of the "nerve net" in the brain. In the late 1940s, Donald Olding Hebb developed a learning hypothesis [10] based on the mechanism of neural plasticity known as Hebbian learning.

In 1954, B. G. Farley and Wesley A. Clark [11] first used computational machines to emulate the Hebbian network. In 1957, the perceptron algorithm was created by Frank Rosenblatt [12] at the Cornell Aeronautical Laboratory that was intended to be a machine indubitably than a program, and its first implementation was for the IBM 704, subsequently in custom-built hardware as the "Mark 1 perceptron". as the Group Method for Data Handling. The basics of continuous backpropagation were realized in the context of control theory by Henry J. Kelley in 1960 [13] and the principles of dynamic programming by Arthur E. Bryson in 1961 [14]. In 1965, the first functional networks with multiple layers were designed and published by

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

Alexey Ivakhnenko and Valentin Grigor′evich Lapa [15]

In 1969, Marvin Lee Minsky and Seymour Papert [16] discovered that basic perceptrons which were incompetent to process the exclusive-or circuit and those computers deficient to process useful neural networks. In 1970, explicit, efficient error backpropagation in arbitrary, discrete, possibly sparsely connected, neural networks-like networks were first described in the master's thesis of Seppo Ilmari Linnainmaa [17], he introduced the reverse mode of automatic differentiation (AD).

In 1973, Stuart Dreyfus [18] used backpropagation to adjust parameters of controllers in rate to error gradients. In 1974, Paul John Werbos first described the process of training artificial neural networks [19] through backpropagation of errors, and the backpropagation algorithm enabled practical training of multi-layer networks in 1975. He also initiates the recurrent neural networks [20].

In the 1980s, metal–oxide–semiconductor (MOS) very-large-scale integration (VLSI), complementary MOS (CMOS) technologies enabled increasing MOS transistor counts in digital electronics [21] that provided higher processing capability to develop the practical ANNs. In 1992, John (Juyang) Weng, Narendra Ahuja and Thomas S. Huang [22] introduced max-pooling to support with least-shift invariance and tolerance to deformation to assist 3D object recognition.

In the 2000s, the advent of greater computing capability and exceeding sophisticated hardware, as well as the existence of vast data sets supported computer scientists for higher-level research on ANN and AI with no end in sight. In 2006, Geoffrey Everest Hinton et al. [23] proposed a high-level learning representation using successive layers of binary with a restricted Boltzmann machine. In 2012, Dan Ciregan, Ueli Meier and Jürgen Schmidhuber [24] developed the first pattern recognizer to gain human-competitive or superhuman level performance on benchmarks.

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

Table-1: ANN contributions

Contribution	Contributor	Year
Threshold Logic Unit	Warren McCulloch Walter Pitts	1943
Hebbian learning	Donald Olding Hebb	1949
Perceptron	Frank Rosenblatt	1957
Simple/complex cell	David Hunter Hubel Torsten Nils Wiesel	1959
Lernmatrix	Karl Steinbuch	1960
ADALINE	Bernard Widrow Ted Hoff	1960
Associative memories	Karl Steinbuch	1961
Outstar Learning	Stephen Grossberg	1967
Boolean associative memory	D. J. Willshaw	1969
Boolean networks	Stuart A. Kauffman	1969
Automatic differentiation (AD)	Seppo Ilmari Linnainmaa	1970
Cerebellum	James Sacra Albus	1971
Linear associative memory	Teuvo Kohonen	1972
Neocognitron	Kunihiko Fukushima	1979
Mass action network design	Walter J. Freeman	1975
ADAPT large-scale systems	Robert Hecht-Nielsen	1977
Hopfield network	John Joseph Hopfield	1982
Learning networks	Cruz-Young	1982
Max-pooling	John (Juyang) Weng Narendra Ahuja Thomas S. Huang	1992

B. Architecture of ANN

The neurons of a network are aggregated into layers. Different layers are used to perform different type transformations based on their inputs. Typically, an artificial neural network consists of an input layer, one or more hidden layers, and an output layer [25]. The inputs travel from the input layer (first layer) to the output layer (last layer), perhaps after traversing multiple times of the layers.

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

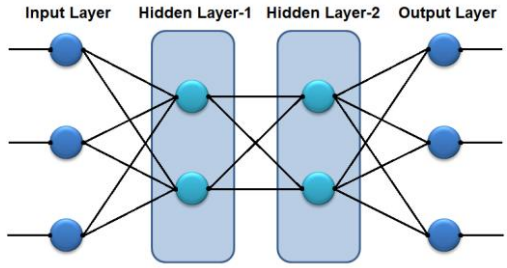


Figure 1: Basic architecture

The input layer gets the input data from the real world such as an image. The input layer passes the data through the activation functions before passing it to the next layer (first hidden layer). The input layer contains weights that perform a multiplication operation on the incoming data.

The hidden layers are placed between the input and output layers of a neural network. The hidden layer performs non-linear transformations of the input data from the previous layer. Hidden layers vary depending on the activation functions and their associated weights. The hidden layers are the layer of mathematical functions that are designed to provide a specific output for an intended result. Hidden layers endorse for the function to be broken down the input data into specific transformations.

The output layer in an ANN is responsible for computations and transferring information from the neural network to the outside world. The neurons of the output layer may be designed differently to improve the results of the repetitive processes.

An artificial neuron is a mathematical function conceived from biological neurons. It is the elementary unit of an artificial neural network. The neuron takes one or more inputs and sums them to produce an intended output result. Typically each input is individually weighted, and the sum is passed through an activation function [26] (non-linear function).

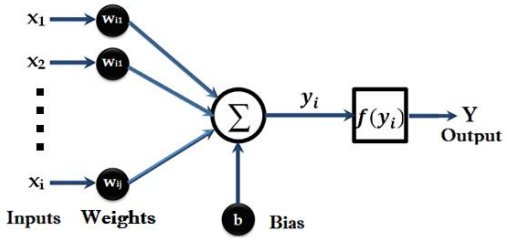


Figure 2: Artificial neuron

In the figure, inputs (x1, x2,....xi) represent the input data, weights (wi1, wi2, ... wij) represent the individual weights, and b represents the bias. The result yi represents the results after performing the summation function operation. The summation function is as following:

$$y_i = \sum_{i=1}^{n} w_{ij} x_i + b$$

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

The function f(yi) represents the activation function. The output of the neuron is represented by Y that we get after performing the activation function operation.

The activation function is a mathematical equation that is attached to the neurons and determines whether it should be fired (activated) or not, based on the input [27]. The activation functions also normalize the neuron-outputs by a range between 1 and 0 (or between -1 and 1).

There are three basic types of activation function [28], such as – (1) binary step function, linear activation function, and non-linear activation function.

The binary step function is a threshold-based activation function where the activation depends on the input value being above or below a definitive threshold. When the neuron is activated, then we get precisely the same signal as the output of the neuron. If W represent the weight values and X represent the input values, then the binary step function define as –

$$y_k = \begin{cases} 1, & if\ W^T X + b_k \geq 0 \\ 0, & if\ W^T X + b_k < 0 \end{cases}$$

The linear activation function produces an output proportional to the inputs. The linear function is more advanced than the binary step function; it allows multiple outputs, not just yes (1) and no (0). An ANN with linear activation is simply a linear regression model which is limited to handle complex parameters of input data. If W represent the weight values and X represent the input values, then the linear function define as –

$$y_k = \begin{cases} 1, & if\ W^T X + b_k \geq 0 \\ W^T X + b_k, & if\ -0.5 < W^T X + b_k < 0.5 \\ 1, & if\ W^T X + b_k \leq -0.5 \end{cases}$$

Modern ANN uses non-linear activation functions (logistic sigmoid) that allow the creation of complex mappings, which are essential for learning and modeling high dimensionality complex data, such as images, video, and audio. If a represent the slop parameter of the sigmoid function and n_k, then the non-linear function define as-

$$y_k = \frac{1}{1 + e^{-an_k}}$$

Some commonly used non-linear activation functions are - Sigmoid, Hyperbolic Tangent, ReLU (Rectified Linear Unit), Leaky ReLU, Parametric ReLU, Softmax, Swish, etc.

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

C. Learning of ANN

The most important behavior of an ANN is learning and it is necessary for every ANN by which the ANN is acquainted with the problems to solve like the brain of a human child. The predictions or classification are produced based on some collected data known as the training set. The three types of machine learning [29] are - (1) Supervised Learning, (2) Unsupervised Learning, and (3) Reinforcement Learning.

In supervised learning, the network provides both the inputs and the outputs. For any specific input, we will get a specific output. The training set of supervised learning contains inputs and desired outputs, and the network just can check that the calculated output is the same as desired output and take appropriate actions. Supervised learning is the most commonly used learning technique. The standard supervised learning problem can be generalized as- semi-supervised learning, weak supervision, active learning, structured prediction, learning to rank.

The unsupervised learning network is provided with inputs but not with desired output. The system makes the decision based on some predefined features or rules. The training set unsupervised learning contains only inputs, and the network attempts to identify similar inputs to put into categories. Unsupervised learning is biologically motivated but not suitable for all the problems.

Reinforced learning is assisted by a "critic" that provides binary feedback by informing the system whether the classification associated with a given training sample is either right or wrong. Reinforced learning is usable for over-fitting and over-generalizing the training data. The training set contains inputs and also additional information provided by the ANN during the training. The standard reinforcement learning problem can be generalized as - associative reinforcement learning, deep reinforcement learning, inverse reinforcement learning, safe Reinforcement Learning.

Other types of learning are - Multi-Instance Learning, Inductive Learning, Deductive Inference, Transductive Learning, Multi-Task Learning, Transfer Learning, Ensemble Learning, etc.

RELATED WORK

Currently, ANN is using in various industries and businesses for various purposes, such as - aircraft fault detectors

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

and simulations, aircraft control systems, auto-piloting, flight path simulations, improved guidance systems, development of power trains, virtual sensors, warranty activity analyzers, chip failure analysis, machine vision, non-linear modeling, prediction of the code sequence, process control, voice synthesis, dynamic modeling of the chemical process, machine diagnosis, product design and analysis, paper quality prediction, quality analysis of computer chips, visual quality inspection systems, condition monitoring, forklift robots, manipulator controllers, trajectory control, vision systems, ATM network control, automated information services, customer payment processing, data compression, equalizers, fault management, handwriting recognition, network design, network monitoring, real-time translation of spoken language, pattern recognition, etc.

A. Related work of Sign Languages detection

T. D. Sajanraj and M. Beena's [30] worked to provide a real-time system that could translate Indian Sign Language (ISL) to the text. Their maximal work was based on the handcrafted features. They introduced a deep learning approach to classify sign language using CNN. The system obtained a high accuracy rate for the high-quality subject.

Aditya Das et al. [31] used CNN to identify sign language gestures. The image dataset consisted of static sign language gestures captured by an RGB camera. Preprocessing was performed on the images to the clean input. The results were obtained by retraining and testing the sign language gestures dataset on a CNN model using Inception v3. The model consisted of multiple convolution filter inputs to process the same input. The system obtained an accuracy rate above 90%.

Karishma Dixit et al. [32] presented a methodology to recognize the Indian Sign Language (ISL) and converted it into a regular text. Their methodology consisted of three stages, those were - a training phase, a testing phase, and a recognition phase. The combinational parameters of Hu invariant moment and structural shape descriptors were built to form a new feature vector to recognize the sign. A multi-class Support Vector Machine (MSVM) was used for training and recognizing signs of ISL. The effectiveness of the method is validated on a dataset with 720 images. The system obtained an accuracy rate above 96%.

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

Radzi Ambar et al. [33] described the development of a translator to translate sign language into speech and text by using wearable devices. The glove-based device can read the movements of a hand and fingers. The device consists of five flex sensors to detect finger movements, and an accelerometer to detect arm movements. The device can recognize any specific gestures (correspond to words) and phrases in American Sign Language (ASL) and translate them into speech by a speaker and text displayed by an LCD screen. The device performed an average value of 0.74 s to translate a sign language to speech and text.

SIGN LANGUAGES DETECTION USING DEEP CNN

The hearing-impaired (deaf) and hard of hearing people always have problems in communicating with other people. They have problems in conveying their thoughts and ideas to normal people who have very limited or sometimes no knowledge of sign language. To overcome the problem, many sign language recognition systems are being developed by researchers, but there is still need for accurate and more effective recognition of the signs. Currently, the systems proposed by earlier researchers are based on the conversion of an action-based verb to an equivalent sign.

The challenges in sign language recognition (SLR) are video trimming, sign extraction, sign video background modeling, sign feature representation, and sign classification. The trained CNN's are tested with two discrete video sets having different signers and viewing angles with varying backgrounds. The robustness testing is performed in two cases. In case-I of testing, the same dataset i.e., already trained dataset is used and in case-II of testing, a separate dataset is used. The performance of the CNN algorithms is measured based on their accuracy in recall and recognition rates.

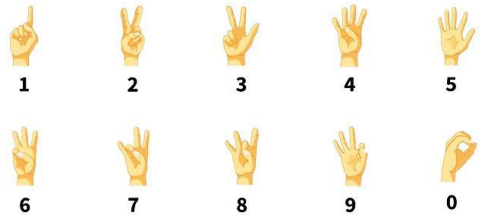


Figure 3: Numeric Sign Language



Figure 4: Alphabet Sign Language

Procedure

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

At first, we capture a basic image of the background, then choose the desired hand gesture and resize the images. Remove the background after capturing the image would preserve only the hand. The functions of the mechanism depend on the number of layers, Euclidian threshold [34], and the fraction of observed color accumulating in the background. The function removed the background but resulted in a loss in parts of the hand. This error is mostly due to the camera. It is necessary to restore the small area after the hand itself.

Then changes the image type from RGB to Binary [35] and applies a filter at different levels. Converting images to binary type is done by replacing all pixels according to the specified luminance with either white (logical 1) if the pixel is equal or greater than the level or otherwise black (logical 0). The specified level should belong to the range (0, 1). This level differs from one image to another. It can be assigned manually but using the Gray-thresh function [36] to compute it; we get better results.

The Gray-thresh function uses Otsu's system [37] to determine the threshold of the image and minimize the interclass modification of the black and white pixels. Multi-dimensional sections are altered to 2D arrays. The "Gray-thresh" function overlooks any non-zero invented part of the image. The next step is removing small connected components and objects from the binary image.

Since we are working on a 2-dimensional image from a low-quality webcam, there is no need to specify any connectivity level. Removing the noise is one of the most critical and challenging tasks for the preprocessing techniques. 2D median filtering is one of the most effective functions to overcome many filtering errors. It performs average filtering of the pixels' matrix in both directions. The output pixel comprises the average value of all pixels of the 3/3 neighboring region. Our task is to recognize a sign based on the input sequence of gestural data. This is a multiclass classification problem. We need to identify the basic characteristics of sign language gestures to choose a suitable classification method. Each sign varies in time and space. Also, the signing speed can differ significantly. After creating the database and obtaining the final filtered input image, it's time to compare the input image with all images in the database to find the closest match.

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

To ensure the best results and to receive the closest image after running the matching algorithm, matching is done via three different independent methods, which are - 2D correlation coefficient, Edge detection, and Data Histogram. The matching percentage between the input image and the image in the database is stored, and eventually, the image with the highest matching results is chosen. The correlation coefficient is an integer signifying the resemblance between two images regarding their pixel strength. The size of the two images must be identical; otherwise, the result would not be anywhere near accurate.

The algorithm calculates every pixel position in both images, the intensity value and compares it to the mean intensity of the whole image. Breaking down an image into blocks before calculating the correlation coefficient reduces the code's elapsing time if the images are relatively large. However, if the correlation coefficient is equal for two blocks of images, this doesn't necessarily indicate that images are identical.

Using small size images in the database is less likely to cause any error, even if resizing a picture may lead to information loss. It provides recognition up to 80% with some errors in extremely close signs. To increase the accuracy, we apply a second level of matching. Edge detection is an image processing procedure for finding the borders of entities within images. Edge detection is used for image dissection and data abstraction in image processing, computer visualization, and instrumental vision.

Applying this method increased matching accuracy from 60% to 80%. It divides the image pixel by pixel, finds the number of white points (logical 1), and finally determines the match depending on the average of the number found. The key objective of the edge function is to find the edge intensity of the image.

As simple as it may sound, this method is very accurate. Though two images may have the same number of white pixels, they are meant to have different edges and thus are at a certain level of dissimilarity. Using this method to compare one image with a large database of images is quite useful and precise. Once running the Matching algorithms, the maximum resemblance percentage between the input image and the images in the database will be saved and displayed.

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

To realize the specific image in the database, we use the "Find" function which is a built-in function used to determine the indices and values of the non-zero elements in the database. After recognizing the index, we need to relate it with a specific letter. For this step, another algorithm is used. For instance, input images that match with the database's image having index "0" will represent the letter "A". Words containing multiple letters and sentences containing spaces are verified using the same method but after passing through the extraction step, which separates each letter and identifies empty images as spaces.

B. Architecture

The general architecture of our system is a common CNN architecture that consists of multiple convolutional and dense layers. The architecture included two groups of two convolutional layers with ReLU activation followed by a max-pool layer and a dropout layer and two groups of fully connected layers followed by a dropout layer and one final output layer.

A convolutional neural network consists of an input, an output layer, and multiple hidden layers. The hidden layers typically consist of convolutional layers, Rectified Linear Units (ReLU) layer, pooling layers, fully connected layers, and normalization layers.

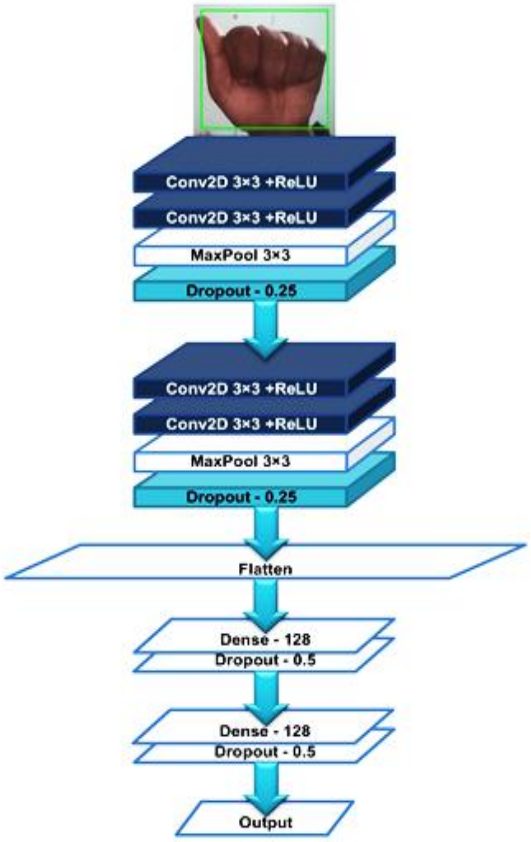


Figure 5: Architecture of CNN

Convolutional layers [38] apply a convolution operation to the input data from the input image and then pass the results to the next layer. The convolution imitates the response of an independent neuron to visual stimuli. Each processed data only for its receptive field.

The goal of the ReLU layer [39] is to incorporate nonlinearity in a system that has been calculating linear progress throughout the previous convolution layers.

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

The convolutional neural networks may include local or global pooling layers [40]. The pooling layers combine the neuron cluster's outputs at one layer into a single neuron in the next layer. In the max-pooling, the maximum value of a cluster of neurons at the prior layer. In the average pooling, the average value of a cluster of neurons at the prior layer.

Fully connected layers [41] connect every neuron in one layer into every neuron in another layer. The fully connected layer is the same as the traditional multi-layer perceptron neural network. In neural networks, each neuron receives input from some number of locations in the previous layer.

In a convolutional layer, neurons receive input from a restricted sub-area of the prior layer. In a fully connected layer, the receptive field (input area of a neuron) is the entire prior layer. In a convolutional layer, the receptive field is smaller than the entire prior layer. The neurons in a neural network compute an output value by applying some function to the input values coming from the receptive field in the previous layer.

The function that is applied to the input values is specified by a vector of weights and a bias. The learning in a neural network progresses by making incremental adjustments to the biases and weights. The vector of weights and the bias are called a filter and represent some feature of the input. The filter is replicated across the entire visual field, and these replicated units share the same parameterization (weight vector and bias) and form a feature map. This means that every neuron in a convolutional layer responds to the same feature within their specific response field. Replicating units allows for features to be detected regardless of their position in the visual field, thus constituting the property of translation invariance. The neurons are arranged in three dimensions (width, height, and depth). The neurons inside a layer are connected to only a small region (receptive field) of the layer before it. Different types of layers locally and completely connected and stacked to form CNN architecture. CNN exploits spatial locality by enforcing a local connectivity pattern between neurons of adjacent layers. The architecture ensures that the learned "filters" produce the strongest response to a spatially local input pattern. Stacking many such layers leads to non-linear filters that become increasingly global (responsive to a larger region of pixel space) so that the network first creates representations of small parts of the input,

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

then assembles representations of larger areas.

Our proposed architecture uses four convolutional layers with different sizes of windows followed by an activation function and a rectified linear unit for non-linearity. The convolutional windows are of size 16×16, 9×9, 5×5, and 5×5 from layers 1 to 4, respectively. There are three kinds of pooling [42] strategies that were tested, are - (1) Mean pooling, (2) Max pooling, and (3) Stochastic pooling. The stochastic pooling is suitable for our system. Representation of the features is done by considering two layers of stochastic pooling. Only two pooling layers are appointed to avoid the substantial information loss in feature representation in our model. The classification stage is implemented with dense/fully connected layers followed by an activation function. The SoftMax regression [43] is adopted in classification.

Table 2: Parameters of CNN Architecture

Layer no.	Layer	Function	Output Shape
1.	Input		3×640×480
2.	Conv – 1	Convolution	32×128×128
3.	Activation – 1	Activation	32×128×128
4.	Conv – 2	Convolution	32×120×120
5.	Activation – 2	Activation	32×120×120
6.	Stochastic pooling - 1	Stochastic pooling	32×60×60
7.	Dropout - 1	Dropout	32×60×60
8.	Conv - 3	Convolution	64×56×56
9.	Activation - 3	Activation	64×56×56
10.	Conv - 4	Convolution	64×52×52
11.	Activation - 4	Activation	64×52×52
12.	Stochastic pooling - 2	Stochastic pooling	64×26×26
13.	Dropout – 2	Dropout	64×26×26
14.	Flatten – 1	Flatten	43264×1×1
15.	Dense - 1	Fully connected	64×1×1
16.	Activation – 5	Activation	64×1×1
17.	Dropout - 3	Dropout	64×1×1
18.	Dense – 2	Fully connected	21×1×1
19.	Activation - 6	Activation	21×1×1
20.	Output	SoftMax Regression	21×1×1

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

C. Result

The main goal of this work is to identify correctly the sign from a selfie dataset in order to build a bridge between the hearing impaired and normal hearing people.

This paper introduces the powerful CNN tool to classify signs at a faster speed with the best recognition rates. The holding of the selfie stick in one hand and performing the sign with the other hand creates different orientations. Each sign image in the data set is pre-processed by reducing its dimensions to which will improve the computational speed of CNN.

We are satisfied to the training computation time for recognition. The time of real-time recognition is 0.4 sec per frame, and it is quite fast compared to other algorithms. A stochastic pooling adoption attained an average recognition rate of 92.88%. Implementing max pooling and mean pooling produces a recognition rate of 91.33% and 89.84% respectively.

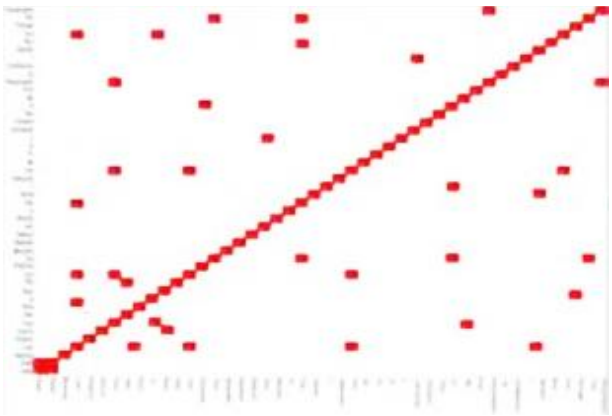


Figure 6: Average confusion matrix generated in Batch-I of training with case-I

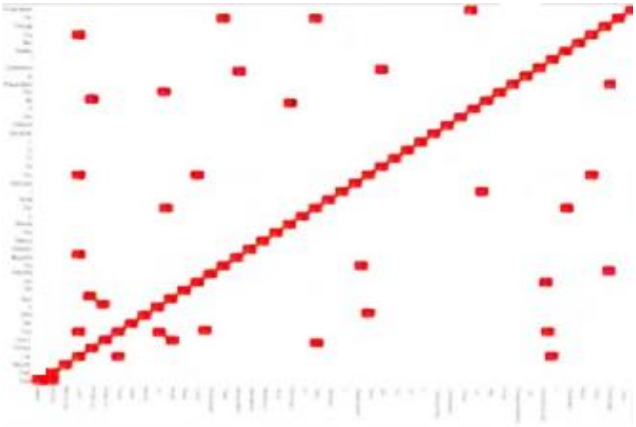


Figure 7: Average confusion matrix generated in Batch-II of training with case-I

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

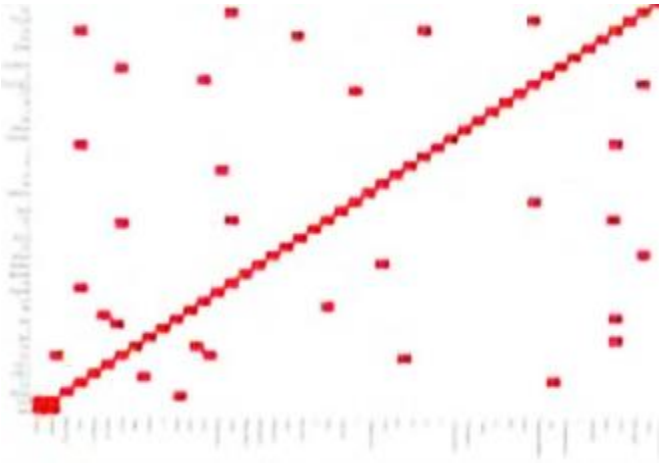


Figure 8: Average confusion matrix generated in Batch-III of training with case-I

We created 44 gesture samples using OpenCV. For each gesture, we captured 1200 images which were 50x50 pixels. All these images were in grayscale, which is stored in the gestures/ folder. The images were flipped using flip_images.py. The script flips images along the vertical axis.

We have stored the 44 gestures for 26 alphabets and 10 numbers of American Sign Language and some other gestures and trained the model on these images. We experiment with different gestures and find out the class belongs to the original alphabet.

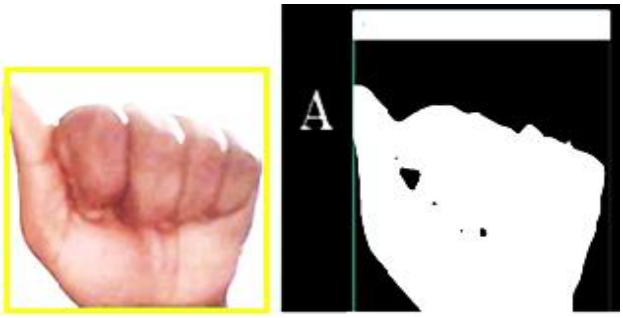


Figure 9: Test 1 (93% match with A)

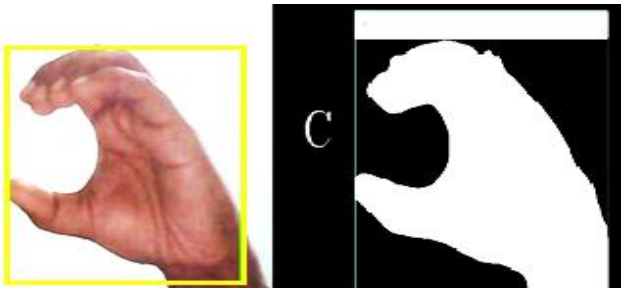


Figure 10: Test 2 (90% match with C)

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

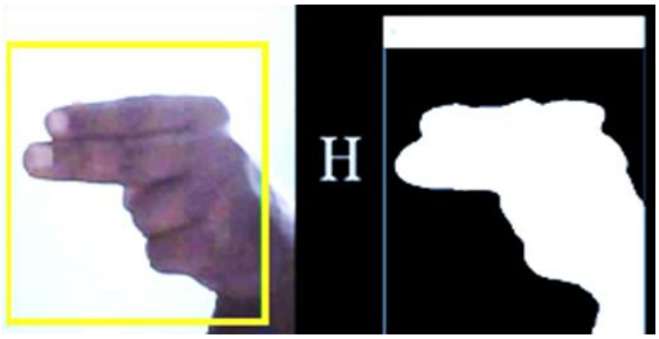


Figure 11: Test 3 (99% match with H)

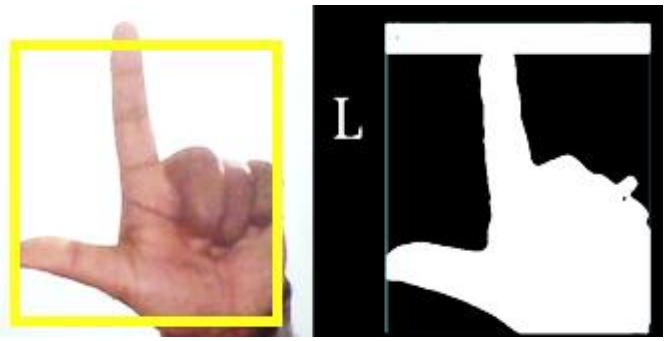


Figure 12: Test 4 (96% match with L)

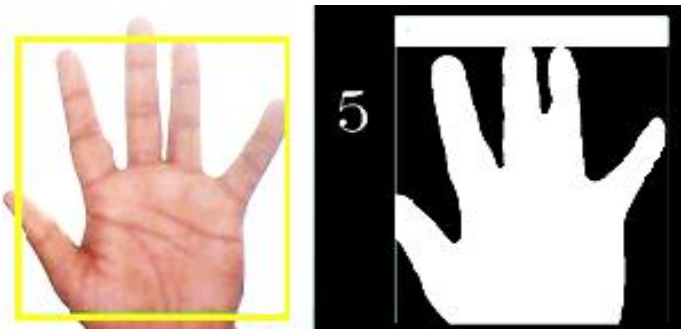


Figure 13: Test 5 (100% match with 5)

Table 3: Result Analysis

Test No.	Objects	Matching	Accuracy	Error
1.		**93% match with A**	93 %	7 %
2.		**90% match with C**	**90 %**	10 %
3.		**99% match with H**	**99 %**	1 %
4.		**96% match with L**	**96 %**	4 %
5.		**100 % match with 4**	**100 %**	0 %
Average			**95.6 %**	4.4 %

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

Therefore, our proposed model achieved an accuracy rate of 95.6 % for the five test images. The error rate of our model is only 4.4.

According to the figure and table, we have represented and analyzed the results of our experiments. We noticed that our system is very fast and generates enough accurate results. But the system also generates some results with fewer probabilities of the specific signs.

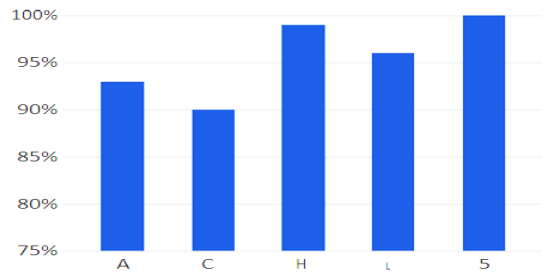


Figure 14: Result Analysis

The overall result of our system has better accuracy level. We have analyzed our system's performance based on the results and it reveals that the system works successfully.

CONCLUSION

From the review of the Artificial Neural Networks (ANN), we have discussed different usages and how it can be correlated in the real-world applications.

In real-time, image recognition is beneficial today in various tasks. Nowadays, every technology is related to some tasks that are done by image recognition and classification. According to the result and performance, we claim that our system works successfully. In the future, by gaining the expected accuracy, it will be possible to make it automated to any application that is complicated for a human to manipulate and hence complicated as well.

ACKNOWLEDGMENT

First of all, we thank the Almighty; without His mercy, it is not possible to succeed.

We would prefer to specify our sincere feelings for our team members for the incorporation, steerage, inspiration, and constructive suggestion that helped us during the preparation and attainment of this work.

REFERENCES

1. L. Lu, Y. Zheng, G. Carneiro, and L. Yang, "Deep Learning and Convolutional Neural Networks for Medical Image Computing: Precision Medicine, High Performance and Large-Scale Datasets", Springer, 2017. ISBN: 978-3-319-42999-1.

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

2. J. E. Solem, “Programming Computer Vision with Python: Tools and algorithms for analyzing images”, O'Reilly Media Inc., 2012. ISBN: 978-1-449-34193-0.
3. G. Bradski, and A. Kaehler, “Learning OpenCV: Computer Vision with the OpenCV Library”, O'Reilly Media, Inc., 2008. ISBN: 978-0-596-55404-0.
4. D.M. D’Addona, “Neural Network. In: The International Academy for Production Engineering, Laperrière L.”, G. Reinhart (eds) CIRP Encyclopedia of Production Engineering. Springer, 2014. DOI: 10.1007/978-3-642-20617-7_6563.
5. A. Ahmed, S. Elkatatny, A. Ali, M. Abughaban, and A. Abdulraheem, "Application of Artificial Intelligence Techniques in Predicting the Lost Circulation Zones Using Drilling Sensors", Journal of Sensors, 2020. DOI: 10.1155/2020/8851065.
6. B. Qolomany, A. Al-Fuqaha, A. Gupta, D. Benhaddou, et al., "Leveraging Machine Learning and Big Data for Smart Buildings: A Comprehensive Survey," in IEEE Access, vol. 7, 2019, pp. 90316-90356. DOI: 10.1109/ACCESS.2019.2926642
7. O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, “State-of-the-art in artificial neural network applications: A survey”, Heliyon,Vol. 4, No. 11, 2018. DOI: 10.1016/j.heliyon.2018.e00938.
8. W.S. McCulloch, and W. Pitts, “A logical calculus of the ideas immanent in nervous activity”, Bulletin of Mathematical Biophysics, 5, 115–133 (1943). DOI: 10.1007/BF02478259.
9. L. Troiano, A. Bhandari, P. Kriplani, and E. M. Villa, “Hands-On Deep Learning for Finance: Implement Deep Learning Techniques and Algorithms to Create Powerful Trading Strategies”, Packt Publishing, 2020. ISBN: 978-1-789-61317-9.
10. D. Hebb, “The Organization of Behavior”, New York: Wiley, 1949. ISBN: 978-1-135-63190-1.
11. B. Farley and W. Clark, "Simulation of self-organizing systems by digital computer," in Transactions of the IRE Professional Group on Information Theory, vol. 4, no. 4, 1954, pp. 76-84. DOI: 10.1109/TIT.1954.1057468.

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

12. F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain”, Psychological Review, Vol. 65, No. 6, 1958, pp. 386–408. DOI: 10.1037/h0042519.
13. H.J. Kelley, "Gradient theory of optimal flight paths", ARS Journal, Vol. 30, No. 10, 1960, pp. 947–954. DOI:10.2514/8.5282.
14. A.E. Bryson, “A gradient method for optimizing multi-stage allocation processes”, Proceedings of the Harvard Univ, Symposium on digital computers and their applications, 1961.
15. A.G. Ivakhnenko, and V.G. Lapa, “Cybernetics and Forecasting Techniques, Volume 8 of Modern analytic and computational methods in science and mathematics”, R.N. McDonough (ed), American Elsevier Publishing Company, 1967. ISBN: 978-0-444-00020-0.
16. M.L. Minsky and S. Papert, “Perceptrons; an Introduction to Computational Geometry”, MIT Press, 1969. ISBN: 978-0-262-13043-1.
17. S. Linnainmaa, “Taylor expansion of the accumulated rounding error”, BIT, 16, pp. 146–160, 1976. DOI: 10.1007/BF01931367.
18. S. Dreyfus, "The computational solution of optimal control problems with time lag", IEEE Transactions on Automatic Control, vol. 18, no. 4, 1973, pp. 383-385. DOI: 10.1109/TAC.1973.1100330.
19. P.J. Werbos, “The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting”, New York: John Wiley & Sons, 1994. ISBN: 978-0-471-59897-8
20. P. J. Werbos, "Backpropagation through time: what it does and how to do it," Proceedings of the IEEE, vol. 78, no. 10, pp. 1550-1560, 1990. DOI: 10.1109/5.58337.
21. C. Mead, and M. Ismail, “Analog VLSI Implementation of Neural Systems”, The Kluwer International Series in Engineering and Computer Science, 80, Kluwer Academic Publishers. DOI: 10.1007/978-1-4613-1639-8.
22. J. Weng, N. Ahuja and T. S. Huang, "Cresceptron: a self-organizing neural network which grows adaptively,"IJCNN International Joint Conference on Neural Networks, Baltimore, MD, USA,

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

vol.1, 1992, pp. 576-581. DOI: 10.1109/IJCNN.1992.287150.

23. R. Salakhutdinov, and G.E. Hinton, "An Efficient Learning Procedure for Deep Boltzmann Machines". Neural Computation, Vol. 24, No. 8, 2012. DOI: 10.1162/NECO_a_00311
24. J. Schmidhuber, "Deep learning in neural networks: An overview, Neural Networks, Vol. 61, 2015, pp. 85-117. DOI: 10.1016/j.neunet.2014.09.003.
25. M. Jafari, and A. Shahsavar, "The application of artificial neural networks in modeling and predicting the effects of melatonin on morphological responses of citrus to drought stress", Plos One, 2020. DOI: 10.1371/journal.pone.0240427.
26. A. Vizitiu, C.L. Niţă, A. Puiu, C. Suciu, L.M. Itu, "Applying Deep Neural Networks over Homomorphic Encrypted Medical Data", Computational and Mathematical Methods in Medicine, 2020. DOI: 10.1155/2020/3910250.
27. G. Castaneda, P. Morris, and T.M. Khoshgoftaar, "Evaluation of maxout activations in deep learning across several big data domains", Journal of Big Data 6, 72, 2019. DOI: 10.1186/s40537-019-0233-0.
28. Z. Ming, "Artificial Higher Order Neural Networks for Economics and Business", IGI Global, 2008, pp. 169. ISBN: 978-1-599-04898-7.
29. H. Tatsat, S. Puri and B. Lookabaugh, "Machine Learning and Data Science Blueprints for Finance". O'Reilly Media, Inc., 2020, pp. 7. ISBN: 978-1-492-07302-4.
30. T.D. Sajanraj, and M. Beena, "Indian Sign Language Numeral Recognition Using Region of Interest Convolutional Neural Network", Second International Conf. on Inventive Communication & Computational Technologies, Coimbatore, 2018, pp. 636-640. DOI: 10.1109/ICICCT.2018.8473141.
31. A. Das, S. Gawde, K. Suratwala, and D. Kalbande, "Sign Language Recognition Using Deep Learning on Custom Processed Static Gesture Images". International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, 2018, pp. 1-6. DOI: 10.1109/ICSCET.2018.8537248.
32. K. Dixit, and A.S. Jalal, "Automatic Indian Sign Language

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

recognition system", 3rd IEEE International Advance Computing Conference (IACC), Ghaziabad, 2013. pp. 883-887. DOI: 10.1109/IAdCC.2013.6514343.

33. R. Ambar, C.K. Fai, M.H.A. Wahab, M.M.A. Jamil, and A.A. Ma'radzi, "Development of a Wearable Device for Sign Language Recognition", 1st International Conf. on Green and Sustainable Computing (ICoGeS), 2017, IOP Conf. Series: Journal of Physics: Conf. Series 1019 (2018)012017. DOI: 10.1088/1742-6596/1019/1/012017.
34. H. Ding, S. Crozier, and S. Wilson, "Optimization of Euclidean distance threshold in the application of recurrence quantification analysis to heart rate variability studies", Chaos, Solitons & Fractals, Vol. 38, No. 5, 2008, pp. 1457-1467. DOI: 10.1016/j.chaos.2006.07.059.
35. R.D. Atmaja, M.A. Murti, J. Halomoan, and F.Y. Suratman, "An Image Processing Method to Convert RGB Image into Binary", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 3, No. 2, 2016, pp. 377-382. DOI: 10.11591/ijeecs.v3.i2.pp377-382.
36. A.K.C. Wong, and P.K. Sahoo, "A gray-level threshold selection method based on maximum entropy principle", IEEE Transactions on Systems, Man, and Cybernetics, vol. 19, no. 4, 1989, pp. 866-871. DOI: 10.1109/21.35351.
37. D. Liu, and J. Yu, "Otsu Method and K-means", Ninth International Conference on Hybrid Intelligent Systems, Shenyang, 2009, pp. 344-349. DOI: 10.1109/HIS.2009.74.
38. S. Albawi, T.A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," International Conf. on Engineering and Technology (ICET), Antalya, 2017, pp. 1-6. DOI: 10.1109/ICEngTechnol.2017.8308186.
39. Ide, H.; Kurita, T. Improvement of learning for CNN with ReLU activation by sparse regularization. 2017 International Joint Conf. on Neural Networks, Anchorage, AK, 2017, pp. 2684-2691. DOI: 10.1109/IJCNN.2017.7966185.
40. W.C.D. Melo, E. Granger, and A. Hadid, "Combining Global and Local Convolutional 3D Networks

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

for Detecting Depression from Facial Expressions", 14th IEEE International Conf. on Automatic Face & Gesture Recognition, Lille, France, 2019, pp. 1-8. DOI: 10.1109/FG.2019.8756568.

41. H. Nakahara, T. Fujii, and S. Sato, "A fully connected layer elimination for a binarizec convolutional neural network on an FPGA", 27th International Conference on Field Programmable Logic & Applications, Ghent, 2017, pp. 1-4. DOI: 10.23919/FPL.2017.8056771.
42. N. Dey, "Intelligent Speech Signal Processing", Elsevier, 2019, pp. 23. ISBN: 978-0-128-18130-0.
43. D. Lin, Z. Lin, L. Sun, K. Toh, and J. Cao, "LLC encoded BoW features and softmax regression for microscopic image classification", IEEE International Symposium on Circuits & Systems (ISCAS), Baltimore, MD, 2017, pp. 1-4. DOI: 10.1109/ISCAS.2017.8050243.

AUTHORS' PROFILE

1. Al Mahmud Al Mamun, Researcher
Department: CSE
College: Islamic University (IU), Bangladesh
Email: almamun.ow@gmail.com

2. Md. Rakibul Hasan, Researcher
Department: CSE
College: Islamic University (IU), Bangladesh
Email: rhp.rakibul7@gmail.com

3. Md. Mokhlesur Rahman, Student & Lecturer
Department: CSE
College: Universiti Kebangsaan Malaysia
Email: mmarks_cse@yahoo.com

4. Md. Mahbubur Rahman, App developer, 10 minuts school
College: University of Business and Technology (BUBT), Bangladesh
Email: mahbubmridha07@gmail.com

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved

5. Sondip Poul Singha, Lecturer

Department: CSE

College: University of Business and Technology (BUBT), Bangladesh

Email: sondipsingha@bubt.edu.bd

6. MD. Yasin, Student

Department: CSE

College: University of Business and Technology (BUBT), Bangladesh

Email: riders.mdyasinkhan1@gmail.com

Page 10-33 © MANTECH PUBLIATIONS 2021. All Rights Reserved