SCHULICH
School of Engineering

UNIVERSITY OF CALGARY

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING

ENEL 645 - Data Mining & Machine Learning

Team No.: 15

Project: Isolated Sign Language Recognition using CNN

Submitted to:
Dr. Roberto Medeiros de Souza
Dept. of Electrical and Computer Engineering
University of Calgary

Submitted by:
Fahimul Haque (UCID # 30025654)
Devang Jigneshbhai Madhani (UCID # 30122189)
Muhammad Younus (UCID # 30050057)
Aditya Jariwala (UCID # 30135898)

Team Member’s Contribution

All members have contributed to the project as follows:

Fahimul Haque has performed literature review and written the final project report. He also collected datasets, designed two mentioned models/algorithms, and written codes for data capturing and to implement aforementioned models.

Devang Jigneshbhai Madhani has collected datasets, and designed and coded one mentioned model/algorithm.

Muhammad Younus has collected datasets and wrote coding for post-training recognition file.

Aditya Jariwala has assisted Devang in data collection and code writing and video compilation.

Consensus Score Table

Name	Score
Fahimul Haque **UCID: 30025654**	3
Devang Jigneshbhai Madhani **UCID: 30122189**	3
Muhammad Younus **UCID: 30050057**	3
Aditya Jariwala **UCID: 30135898**	3

Abbreviations

SLR	Sign language recognition
ML	Machine learning
DL	Deep learning
CNN	Convolutional neural networks
FCNN	Fully connected neural network
ReLU	Rectified linear unit
ASL	American Sign Language
HSV	Hue, Saturation, Value (Color space)

Github Repository: https://github.com/fahimul-haque/Isolated-Sign-Language-Recognition-using-CNN_ENEL-645-Final-Project

Isolated Sign Language Recognition using CNN

1. Motivation and Significance

Automatic sign language recognition (SLR) is an arising human computer interaction area that assists hearing or speech impaired people. In addition, gesture recognition is one of the important methods to build user-friendly interfaces. It is a technique that is used to understand and analyze the human body language and interact with the user accordingly. In recent years, using machine learning (ML) techniques, researchers have introduced sensor-based and vision-based SLR systems, where both systems are constrained by various limitations [1]. Proposed sensor-based SLR systems considered a number of ideas, including gloves with accelerometer and tilt sensor [2] or multi-colored glove [3], accelerometer and multichannel electromyography [4], Kinect [5, 6], leap motion sensor [7], to estimate hand and/or fingers positions. Whereas proposed vision-based systems applied techniques such as dynamic skin for segmentation and/or hand movement detection, facial expression, skin tone detection using background removal technique [8, 9, 10, 11]. Compared to sensor-based techniques, vision-based methods offer more mobility and low-cost system; however, these methods suffer from change in illuminations and lack of depth information when considering 2-dimensional images. Apart from these two classifications, the methods for hand gesture recognition categorized based on the ML models used, e.g. neural network, 3D convolutional neural network, support vector machine, hidden markov model, k-means clustering [4, 6, 12, 13, 14, 15]. These proposed methods vary in terms of achieved accuracy, computational efficiency, number of samples used for performance analysis, flexibility and so on.

Our project aims to perform isolated SLR using supervised deep learning (DL) models. In this report, a fully connected neural network (FCNN) and two convolutional neural network (CNN) models, including a design inspired from VGG16, have been implemented and their performance have been analyzed for SLR. These classification techniques can be useful for building a system for Gesture Navigation and communication with hearing or speech impaired people.

2. Methodology

The goal is to build DL models to classify the signs, from the images, which are captured through a webcam. To design this project there are three main steps which are as follows: capturing images for creating dataset, training the model, and prediction of sign/gestures.

2.1 Creating Dataset

For this project, signs from American Sign Language (ASL) have been considered. Dataset for total 36 classes (i.e. sign languages) have been recorded that represent digits 0-9 and alphabets A-Z. In total, 200 images for each class and 9000 images considering all 36 classes have been captured. Out of 9000 images, 5400 images have been used for training, 1800 images for validation and 1800 images for testing. It should be also noted here that since some gestures look exactly same (e.g. ‘0’ and ‘O’) or almost similar (e.g. ‘1’ and ‘D’) in ASL, to simplify the process, new gestures have been assigned for this cases.

Images have been captured through a webcam via OpenCV with a homogeneous background and stored after some processing according to associated classes. During image capturing stage, image segmentation is applied based on skin tone to differentiate region of interest (ROI), i.e. hand from the background. To do that the color space of the image has been converted to HSV (Hue, Saturation, Value) color space from RGB color space and specific range of HSV values for skin color has been applied as threshold to differentiate the image pixels of hand from the background.

In addition, trackbars have been added to adjust HSV range in real-time to detect skin tone without closing the running program. The trackbars represent the lowest and highest values of the Hue, Saturation and Value of the HSV color space and adjustment to the values may be required for different background or webcam quality. Pixels falling within the range of threshold values are converted to white color and those which do not fall within this range are converted to black. Lastly, the captured images are stored as 64 × 64 images under associated class folder directories. In short, the process for creating dataset is as follows:

- Capture an image from the webcam.
- Know/adjust the range of HSV values for skin color and use them as threshold values.
- Conversion of image from RGB color space to HSV color space.
- If pixels fall within the range of threshold values, convert them to white.
- If pixels do not fall within the range of threshold values, convert them to black.
- Lastly, save the segmented image.

Figure 1 represents some examples of the captured images.

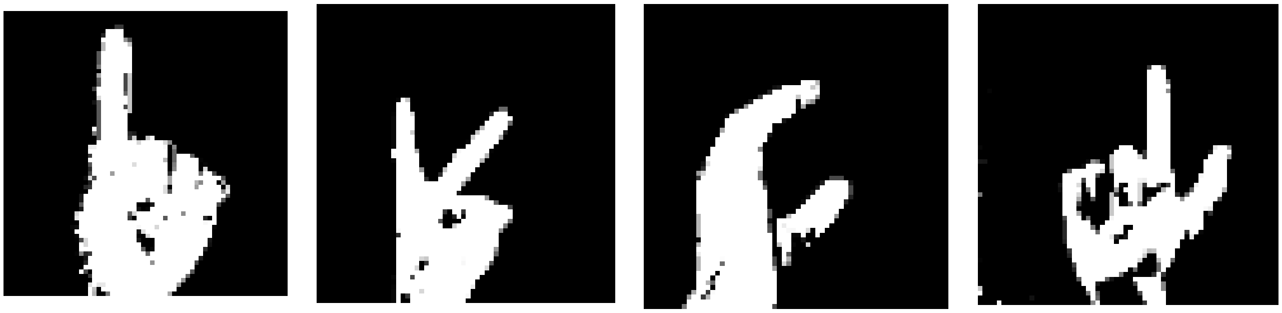


Figure 1: Examples of captured images (from left: sign for 1, 2, alphabet C and L) using OpenCV

2.2 Model Training

The second stage of the project involves selecting models for SLR and train these using the stored images from the previous step. Three deep learning models, namely FCNN, CNN, and modified VGG-16, have been trained for this project. Where, for our implementation, all designed models contain one input layer, multiple hidden layers, and an output layer. The stored images are fed to the system as inputs and the 36 classes for these images are considered as outputs.

The proposed FCNN for this project contains three hidden layers, the first two layers include 512 neurons and the third layer consist of 256 neurons. To prevent the algorithm from overfitting, a dropout value of 0.4 has also been used for the first two hidden layers. The final layer has 36 nodes for 36 classes of datasets. To generate activation map the hidden layers use Rectified linear unit (ReLU) function and the output layer uses Softmax function to predict the best possible output.

For the first CNN model, four convolutional layers are used, where the first three layers use 32 filters and the last layer includes 64 filters, each filter having 3 × 3 kernels. Same padding has been applied to keep the size of the output-feature maps same as input-feature maps and ReLU functions is used for generating activation map. After each convolution layer (except for the first layer), Max Pooling layer of 2 × 2 pooling window is introduced to reduce the dimensionality of our images. The outputs from last pooling layer are then fed to fully connected layers. To do that the dimension of the data is first converted to a single dimension vector using flattening operation. After flattening operation, three fully connected layers are added with 784, 256, and 36 nodes/neurons, where the output layer is the one with 36 nodes.

Lastly, another model has been implemented that uses similar architecture as popular CNN model VGG-16 [16]. However, instead of feeding the model with 224×224 RGB images as in traditional VGG-16 model, the model is fed with 64×64 images in HSV color space. In the hidden layers, total thirteen convolutional layers with 3×3 kernels are added along with 5 stages of Max Pooling with 2×2 pooling window. The first convolutional layers included 64 filters, after which a Max Pooling layer is added to reduce the size of the photo into half. Afterwards, two more convolutional layers with 128 filters are implemented along with another Max Pooling layer. After that three stages each containing three convolutional layers and one max pooling layer is added, where the convolutional layers in these three stages contain 256, 512, and 512 filters, respectively. Lastly, after flattening the dimension, three FCN layers are added, where the first two FCN layers include 4096 neurons and the last one (output layer) contains 36 nodes. In all convolutional layers and FCN layers (except the output layer), ReLU function is used for generating activation map. Softmax function is used to create activation map for the last layer.

To train models data augmentation, such as rescale, perception angles, has been applied using ImageDataGenerator of Keras. Fine tuning such as adjusting learning rate, dropping rate, number of layers, batch size, optimizers have also been implemented to determine desired result. All the models are trained over 50 epochs, with a learning rate of 0.00001. The implemented codes for these models can be found here: https://github.com/fahimul-haque/Isolated-Sign-Language-Recognition-using-CNN_ENEL-645-Final-Project

2.3 Prediction of signs

The last stage includes recognition of sign languages, where the trained model receives a gesture given by user as an image and classifies it as one of the stored classes. For this, two different routes are considered. The first one includes use of trained models to recognize test images that are stored in the very first stage. The test accuracy are discussed in Results section. The second route includes detecting hand gesture in real-time using live webcam streaming and giving the output as text/speech. Figure 2 demonstrates the flow diagram of the proposed SLR system.

3. Results

Performances of the three models that have been implemented for this project are compared in this section. To evaluate the performances the accuracy metric is used. TABLE 1 shows the training, validation, and test accuracy for the implemented models.

TABLE 1
Training, Validation, and Testing Accuracy of Implemented Models

	FCNN	CNN	VGG-16 based model
Training Accuracy	96%	99.7%	100%
Validation Accuracy	89%	93.4%	92.1%
Testing Accuracy	87.2%	90.6%	91.3%

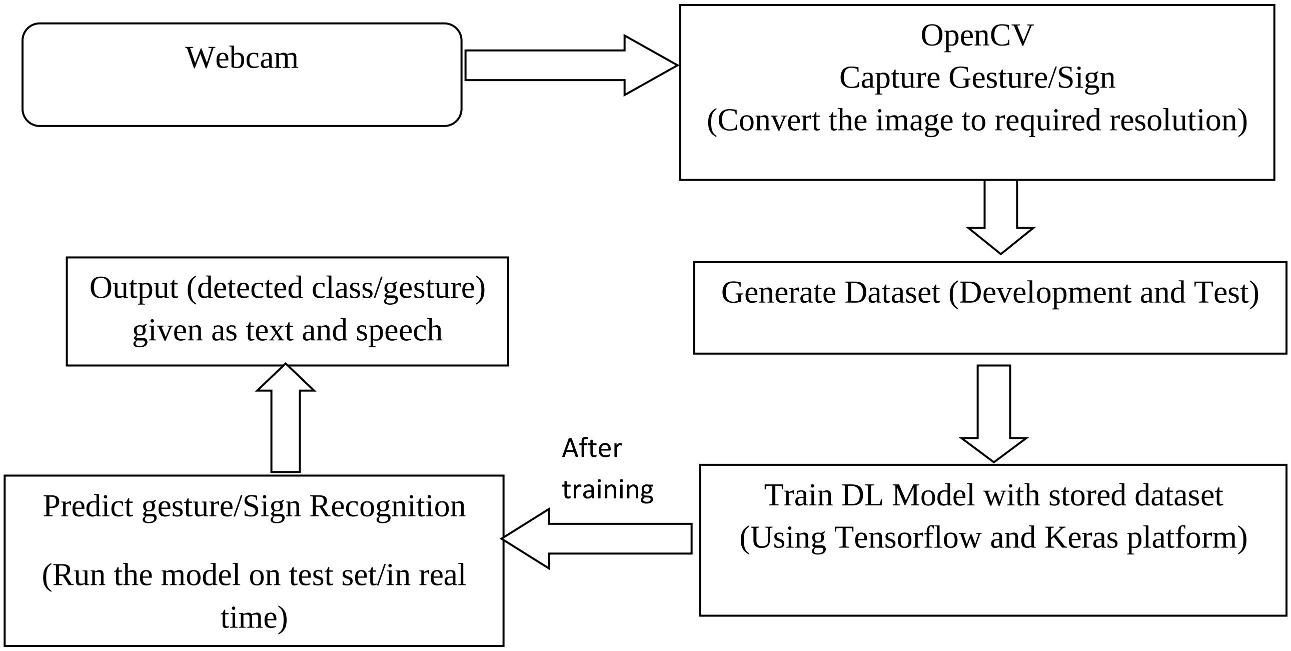

Webcam
OpenCV
Capture Gesture/Sign
(Convert the image to required resolution)
Generate Dataset (Development and Test)
Train DL Model with stored dataset
(Using Tensorflow and Keras platform)
After training
Predict gesture/Sign Recognition
(Run the model on test set/in real time)
Output (detected class/gesture) given as text and speech

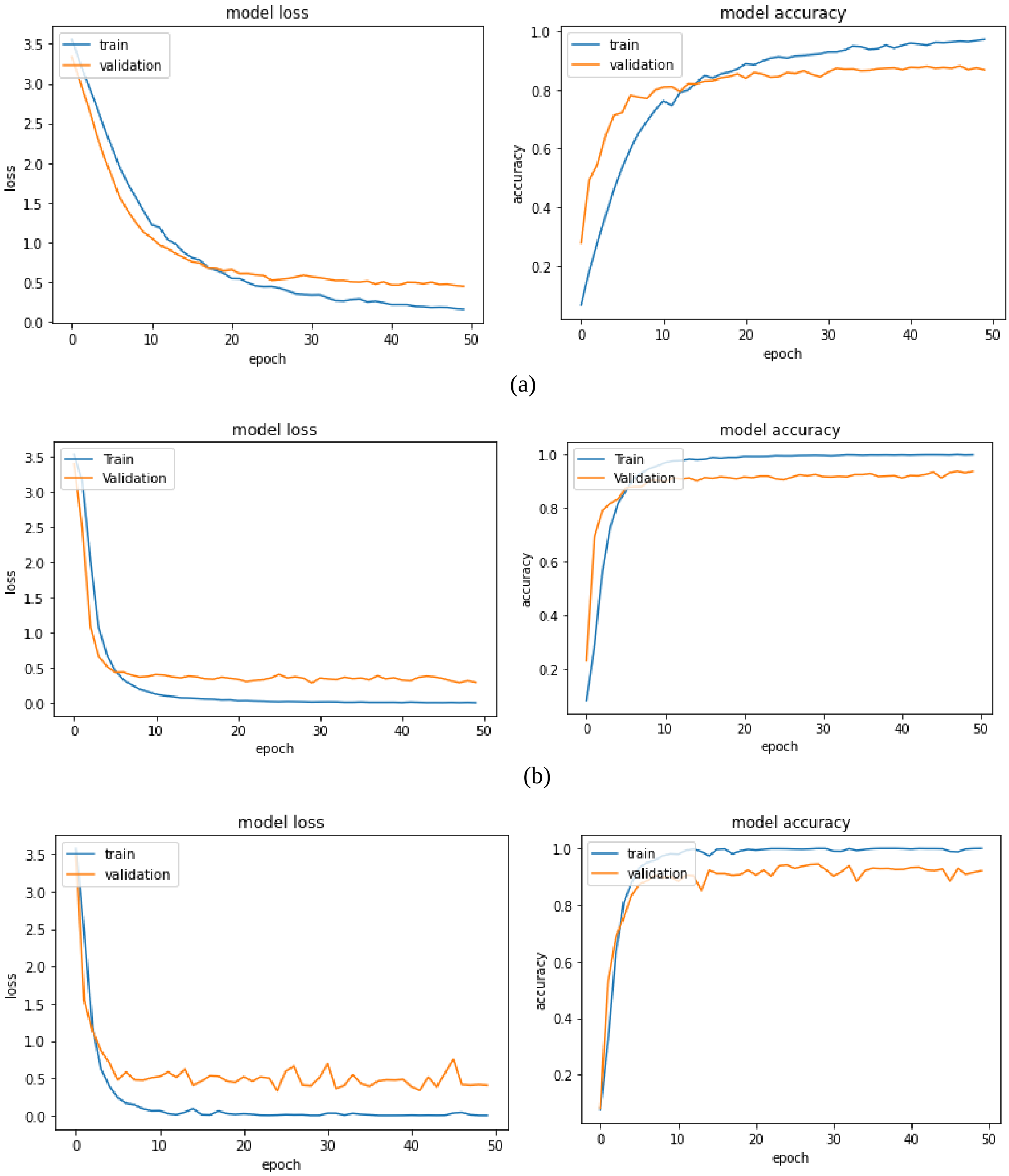
Figure 2: Flow diagram of proposed SLR system

As can be seen from TABLE 1, the VGG-16 architecture-based model outperforms the other two models in terms of training and testing accuracy, but it is to be noted here that this model is also more computationally complex compared to the other two models. As expected, the CNN models outperform FCNN for the particular architecture used and the lowest accuracy has been achieved from the FCNN model.

Further, the model loss and accuracy for training and validation sets for all models have been plotted for 50 epochs and showed on Figure 3. As can be observed from the figure, as expected, the model loss decreases over number of epochs and the model accuracy increases with an increase in number of epochs used for training and validating the models. It can also be observed from all the graphs that the models converge to a solution after approximately 15-20 epochs.

4. Conclusion and Future Work

This project focused on isolated sign recognition using deep learning techniques. From experimental work, it has been observed that CNN models outperformed FCNN model, and more complex model provided better accuracy. For future work, we can work on deep learning techniques (i.e. Recurrent neural networks) that work along a temporal sequence. This can allow the system to recognize words/sentences that are made by the user.



(a)

(b)

(c)

Figure 3: Model loss and accuracy for (a) FCNN, (b) CNN, and (c) modified VGG-16

References

[1] R. Elakkiya, "Machine learning based sign language recognition: a review and its research frontier," *Journal of Ambient Intelligence and Humanized Computing,* 2020.

[2] A. Z. Shukor, M. F. Miskon, M. H. Jamaluddin, F. b. Ali, I. M. F. Asyraf and M. B. b. Bahar, "A New Data Glove Approach for Malaysian Sign Language Detection," *Procedia Computer Science,* vol. 76, 2015.

[3] R. Y. Wang and J. Popović, "Real-Time Hand-Tracking with a Color Glove," *ACM Transactions on Graphics,* vol. 28, no. 3, 2009.

[4] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang and J. Yang, "A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans,* vol. 41, no. 6, pp. 1064 - 1076, 2011.

[5] I. C. o. A. I. a. S. Computing, "Sign Language Recognition Using Kinect," in *International Conference on Artificial Intelligence and Soft Computing*, Simon Lang; Marco Block; Raúl Rojas, 2012.

[6] P. Molchanov, S. Gupta, K. Kim and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* Boston, MA, USA, 2015.

[7] C.-H. Chuan, E. Regina and C. Guardino, "American sign language recognition using leap motion sensor," in *2014 13th International Conference on Machine Learning and Applications*, Detroit, MI, USA, 2014.

[8] N. B. Ibrahim, M. M.Selim and H. H. Zayed, "An Automatic Arabic Sign Language Recognition System (ArSLRS)," *Journal of King Saud University - Computer and Information Sciences,* vol. 30, no. 4, pp. 470-477, 2018.

[9] G. Caridakis, S. Asteriadis and K. Karpouzis, "Non-manual cues in automatic sign language recognition," *Personal and Ubiquitous Computing volume,* vol. 18, pp. 37-46, 2014.

[10] N. B. Ibrahim, M. M. Selim and H. H. Zayed, "A dynamic skin detector based on face skin tone color," in *2012 8th International Conference on Informatics and Systems (INFOS),* Giza, Egypt, 2012.

[11] J. Han, G. Awad and A. Sutherland, "Automatic skin segmentation and tracking in sign language recognition," *IET Computer Vision,* vol. 3, no. 1, pp. 24-35, 2009.

[12] R. Akmeliawati, F. Dadgostar, S. Demidenko, N. Gamage, Y. Kuang, C. Messom, M. Ooi, A. Sarrafzadeh and G. SenGupta, "Towards real-time sign language analysis via markerless gesture tracking," in *2009 IEEE Instrumentation and Measurement Technology Conference,* Singapore,

2009.

[13] M. Elmezain, A. Al-Hamadi, J. Appenrodt and B. Michaelis, "A Hidden Markov Model-based continuous gesture recognition system for hand motion trajectory," in *2008 19th International Conference on Pattern Recognition*, Tampa, FL, USA, 2008.

[14] M. Elmezain, A. Al-Hamadi, J. Appenrodt and B. Michaelis, "A Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition," *International Journal of Electrical, Electronic and Communication Sciences,* vol. 2, no. 5, pp. 760 - 767, 2008.

[15] M. M. Zaki and S. I. Shaheen, "Sign language recognition using a combination of new vision based features," *Pattern Recognition Letters,* vol. 32, no. 4, pp. 572-577, 2011.

[16] K. Simonyan and A. Zisserman, "VERY DEEP CONVOLUTIONAL NETWORKS FORLARGE-SCALE IMAGE RECOGNITION," in *The International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.