**Machine Learning – IT4060**

**Assignment II**

A picture containing icon

Description automatically generated

**B.Sc. (Hons) in Information Technology Specializing in Software Engineering**

**Department of Computer Science and Software Engineering Sri Lanka Institute of Information Technology**

**Sri Lanka**

**May 2022**

**Contents**

[**Introduction** 3](#_Toc104759925)

[**Dataset** 4](#_Toc104759926)

[**Methodology** 5](#_Toc104759927)

[**Results and Discussion** 11](#_Toc104759928)

[**Limitations and Future Work** 11](#_Toc104759929)

[**Individual Contributions** 12](#_Toc104759930)

[IT19067902 YOGANATHAN J.A. 12](#_Toc104759931)

[IT19004914 JAYASINGHA J.M.M.M 12](#_Toc104759932)

[IT19020990 WIJESINGHE M.K 12](#_Toc104759933)

[IT19089300 RAMYATHILAKE S.H.M. 12](#_Toc104759934)

[**OneDrive Link to Demonstration Video** 13](#_Toc104759935)

[**References** 14](#_Toc104759936)

# **Introduction**

A normal person with a hearing threshold of 20db or greater in both ears is considered as a person without any hearing disabilities, anyone having a lesser hearing threshold is considered as a person with a mild, moderate, severe, or profound hearing loss. Persons ranging from mild to severe hearing impairments are categorized as ‘Hard of Hearing’. They communicate through spoken language and with the help of hearing aids. Persons with profound hearing loss is categorized as ‘Deaf’. Deaf people have extremely weak hearing or rather no hearing at all and they communicate using sign language.

Graphical user interface

Description automatically generatedSign language is a non-verbal communication medium which consists of facial expressions, hand signals, gestures, and body language. According to the World Health Organization over 5% of the world’s population are affected by hearing loss and deafness. This includes approximately 430 million adults and 34 million children. Even though hearing disability is a familiar topic in the society, there is less awareness regarding the difficulties faced by this community.

Figure - WHO Statistic of Hearing Loss

The main dilemma faced by the hearing-impaired community is the prevailing communication barrier between them and the people who communicate with spoken languages. Majority of the persons without any hearing disabilities lacks sign language knowledge. This situation makes the lives of the hearing-impaired community much more complicated as they are unable to effectively interact with the society to fulfil their needs and wants.

The proposed solution is a minor attempt to bridge the prevailing communication barrier between the two communities. We have implemented a model using machine learning techniques which is capable of recognizing sign language gestures that represent letters of the English alphabet. Any sign gesture representing a letter of the English alphabet can be presented in front of a camera and the corresponding letter would appear on the screen of the user. This solution can be used by both communities for communication as well as for educational purposes.

# **Dataset**

The model was trained using a dataset that was downloaded from Kaggle which contains images of the English alphabet relevant to the American sign language. The images have been separated into 29 different folders representing various classes. The dataset has been categorized as ‘Training’ and ‘Testing’ data. The training dataset comprises of images representing the 26 letters of the English alphabet and 3 other gestures representing ‘Delete’, ‘Nothing’ and ‘Space’. Each class comprises of over 7000 elements. The testing dataset consists of 29 images that can be used to test the speed and accuracy of the model.

<https://www.kaggle.com/datasets/debashishsau/aslamerican-sign-language-aplhabet-dataset>

A picture containing graphical user interface

Description automatically generated

Figure - Kaggle Dataset

# **Methodology**

VGG 16 algorithm was used to train the model. VGG 16 is a convolutional neural network model which was used to win the Imagenet competition in 2016. VGG 16 is considered to be an excellent object detection model. The VGG 16 model consists of convolutional layers of 3x3 filters throughout the entire network with the stride of 1 pixel and maxpool layer of 2x2 of stride 2. The entire architecture comprises follows this order of convolutional and maxpool layers. Finally, the output will be generated by a softmax layers which is followed by 2 fully connected layers. The 16 in VGG 16 represents the 16 layers that have weights.

Diagram

Description automatically generated

Figure - VGG 16 Architecture

The dataset has been downloaded from Kaggle and saved in the relevant drive folder

Graphical user interface, text

Description automatically generated

Figure - Jupyter Notebook Screenprint

Afterwards the required libraries have been imported to the project

Graphical user interface, text

Description automatically generated with medium confidence

Figure - Jupyter Notebook Screenprint

Importing the required libraries

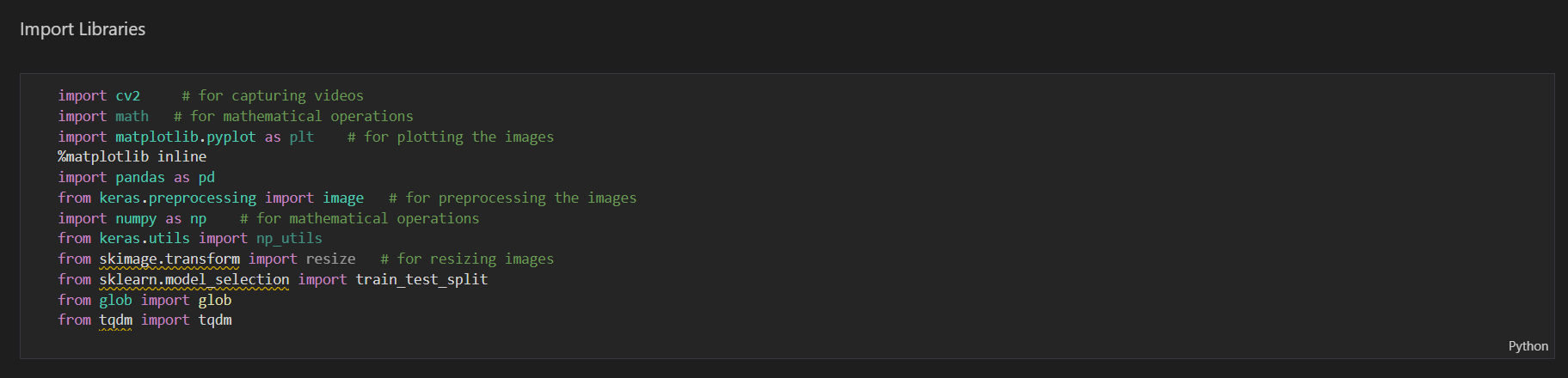


Figure - Jupyter Notebook Screenprint

Renaming the images extracted from the dataset and generating a single text file

Text

Description automatically generated

Figure - Jupyter Notebook Screenprint

Assigning images into training and testing datasets

Graphical user interface, text, application

Description automatically generated

Figure - Jupyter Notebook Screenprint

Creating tags to identify training and testing images

Graphical user interface

Description automatically generated with medium confidence

Figure - Jupyter Notebook Screenprint

Adding all the training images into a single CSV file

Text

Description automatically generated

Figure - Jupyter Notebook Screenprint

Assigning training images into training array

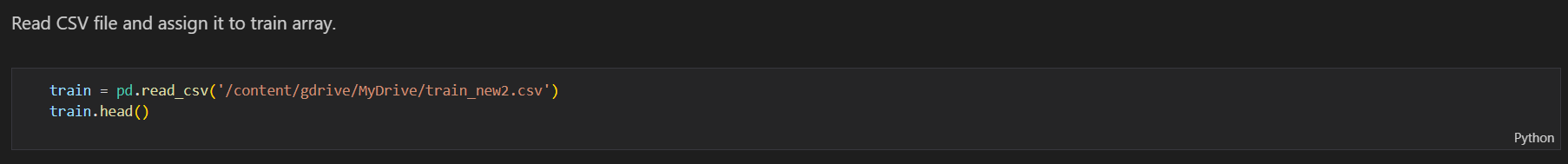


Figure - Jupyter Notebook Screenprint

Preprocessing images and converting into an array list,

Converting images into HSV color space, blurring the image using Gaussian blur technique, detecting edges of the image using Sobel edge detection

Text

Description automatically generated

Figure - Jupyter Notebook Screenprint

Assigning the VGG 16 model as the base model

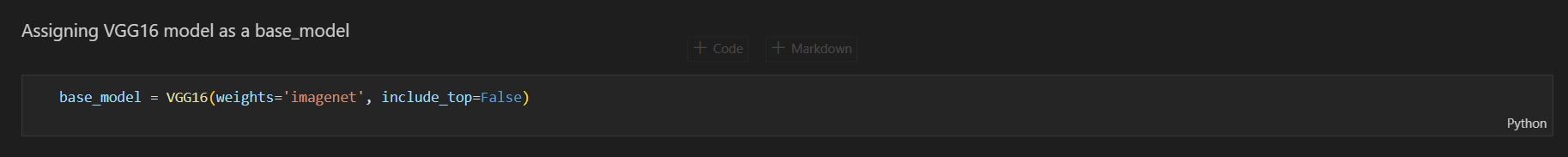


Figure - Jupyter Notebook Screenprint

Reshaping the training and validation frames into a single dimension and normalizing the pixel values

Graphical user interface, background pattern

Description automatically generated

Figure - Jupyter Notebook Screenprint

Defining the model architecture

Graphical user interface

Description automatically generated with low confidence

Figure - Jupyter Notebook Screenprint

Defining a function to save the model with the best weights based on the validation loss value. Then, the model is compiled and the training process will be started.

A screenshot of a computer screen

Description automatically generated with medium confidence

Figure - Jupyter Notebook Screenprint

# **Results and Discussion**

Graphical user interface, application

Description automatically generatedThe accuracy and the validation loss of the model is shown in the below diagrams.

Figure - Validation Loss

Graphical user interface

Description automatically generated with medium confidence

Figure - Classification accuracy

The model has been evaluated under the below 4 metrics,

Classification accuracy, precision, recall and F1 score. The below table depicts the values attained for each of these metrics when tested on letter A, S, P, D and W. Additionally, the pre-processing steps have contributed towards improving the accuracy of the predictions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Letter** | **Precision** | **Recall** | **F1 score** |
| A | 1.0 | 0.95 | 0.98 |
| S | 0.94 | 0.92 | 0.93 |
| P | 0.89 | 1.00 | 0.94 |
| D | 1.00 | 0.74 | 0.83 |
| W | 0.77 | 0.74 | 0.75 |

Figure - Metrics of Model

# **Limitations and Future Work**

As the initial step, the proposed model is developed to identify only the letters of the English alphabet. However, as future work, we can focus on expanding the dataset in order to identify other sign gestures including gestures that involve a movement. In addition, real-time identification of gestures along with facial expression detection can be explored.

# **Individual Contributions**

## **IT19067902 YOGANATHAN J.A.**

1. Understanding the problem and the gathering background information relating to the research problem [4][1][2]
2. Understanding the VGG 16 algorithms and feature extraction and preprocessing techniques that can be incorporated into the algorithm [5]
3. Creating the report and updating the sections of the report
4. Developed a basic UI along with an API to display the prediction of the model
5. Rearranging dataset to suit the VGG 16 algorithm
6. Developing model and training model

## **IT19004914 JAYASINGHA J.M.M.M**

1. Understanding the problem and the gathering background information relating to the research problem [4][1][2]
2. Understanding the VGG 16 algorithms and feature extraction and preprocessing techniques that can be incorporated into the algorithm [5]
3. Creating the report and updating the sections of the report
4. Developed a basic UI along with an API to display the prediction of the model
5. Rearranging dataset to suit the VGG 16 algorithm
6. Developing model and training model

## **IT19020990 WIJESINGHE M.K**

1. Understanding the problem and the gathering background information relating to the research problem [4][1][2]
2. Understanding the VGG 16 algorithms and feature extraction and preprocessing techniques that can be incorporated into the algorithm [5]
3. Creating the report and updating the sections of the report
4. Developed a basic UI along with an API to display the prediction of the model
5. Rearranging dataset to suit the VGG 16 algorithm
6. Developing model and training model

## **IT19089300 RAMYATHILAKE S.H.M.**

1. Understanding the problem and the gathering background information relating to the research problem [4][1][2]
2. Understanding the VGG 16 algorithms and feature extraction and preprocessing techniques that can be incorporated into the algorithm [5]
3. Creating the report and updating the sections of the report
4. Developed a basic UI along with an API to display the prediction of the model
5. Rearranging dataset to suit the VGG 16 algorithm
6. Developing model and training model

# **OneDrive Link to Demonstration Video**

<shorturl.at/sDF01>

# **References**

K. Bantupalli and Y. Xie, "American Sign Language Recognition using Deep Learning and Computer Vision," 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 4896-4899, doi: 10.1109/BigData.2018.8622141 [1]

Mazhar, Osama, Sofiane Ramdani, and Andrea Cherubini. 2021. "A Deep Learning Framework for Recognizing Both Static and Dynamic Gestures" Sensors 21, no. 6: 2227. [2]

<https://www.analyticsvidhya.com/blog/2019/09/step-by-step-deep-learning-tutorial-video-classification-python/> [3]

[https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss](https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss%20%5b4)  [4]

<https://www.youtube.com/watch?v=mjk4vDYOwq0&t=286s> [5]