A collage of a person's face

Description automatically generated with medium confidence

|  |
| --- |
|  |
| **Arabic Sign Language Model**  Neural Network Model – Projrct Final Report |
| |  |  |  | | --- | --- | --- | | Hayat Aldhahri & Juri AlSayigh | 11/25/21 | Neural Network Model | |

# **Overview**

Sign language recognition is an important pillar for the development of societies. There are multiple solutions and approaches to improve and develop sign language recognitions systems. One approach is through machine learning. Developing Neurol Network Model a robust visual recognition algorithm for ArSL will aid improving the communities with hearing disabilities living in a smart city.

# **Data Description**

The dataset contains 1280 images of Arabic Sign Language alphabets, 40 images for each sign. The images were performed of 32 Arabic signs and alphabets (class). The images were split into 980 traning and 240 for test.

A collage of a person's face

Description automatically generated with medium confidence

Figure 1: Example of the 32 ArSL

# **Algorithm**

First, we define Images to Collect in order to make the model recognize images. We started by capturing images for Arabic sign language and storing it on local drive. There were 32 ArSL defined signs where each sign has 40 images captured. Following that, a label has been defined for each ArSL image so that the model can recognize the image. The images with the corresponding label are fed into the model for it to start learning.

Graphical user interface, text, application, email

Description automatically generated

Figure 2: Code for image capturing

## **Model training and evaluation**

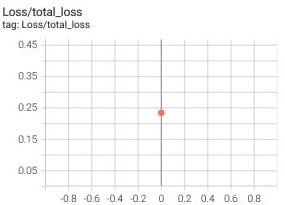
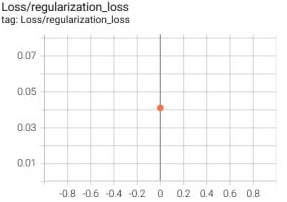
The below steps are followed for model setup and training ;

* Setup paths
* Download TF Models Pretrained Models from Tensorflow model zoo and install TFD
* Create Label Map
* Create TF Records
* Copy model configuration to training folder
* Update configuration for transfer learning
* Train the model
* Evaluate the model
* Load the train model from checkpoint
* Detect from an image
* Real time detection from a webcam
* Freezing the graphs
* Conversion to TFJS
* Conversion to TFLite
* Zip and Export Model

## **Model Evaluation**

The below figure displays the model total loss. these figures were prepared using TenserFlow for machine learning. As shown below, the total loss is approximately 0.23, which is quite low number and reflects the model accuracy.

Figure 3: Model Total Loss



# **Tools**

# The tools used to accomplish the seven project are provided as modules in Python such as Pandas, Matplotlib, Seaborn, NumPy, Librosa, IPython, Keras, Scipy, and sklearn, Opencv-python (Video Capturing), LabelImg library (Training Images Labeling tool), Tensorflow (Tensorflow Object Detection), sounddevice.

The python libraries support multiple data analysis and cleaning methods to ensure the data is clean and ready to be visualized. In order to make the model more user-friendly, Flask or Android Studio might be deployed as tools to develop a web or mobile application.

# **Conclusion**

Neural network modeling will be used to develop a visual recognition system for ArSL. The model will aid in improving the research in sign language recognition and can be implemented as part of the smart city project.