



AMERICAN SIGN LANGUAGE DETECTION

MIHIR JOSHI

SHAMBHAVI KULKARNI

MANSI MAURYA

RESEARCH PAPER

Dynamic Tool for American Sign Language Finger Spelling Interpreter

ABSTRACT

Sign language is a type of language that uses manual communication to convey meaningful messages to other people. This includes simultaneous employing of hand gestures, movement, orientation of the fingers, arms or body, and facial expressions to convey a speaker's ideas. American Sign Language is one of the popular sign language used by most of deaf and dumb people to communicate with each other. American Sign Language is also referred to as ASL. A real-time sign language translator is required for facilitating communication between the deaf community and the general public. We propose a system called Dynamic tool for American Sign Language (ASL) finger spelling interpreter which can consistently classify the letters a–z. The dataset consists of a set of American Sign Language videos. Our approach first converts the videos into frames and then pre-processes the frames to convert them into greyscale images. Then the Convolutional Neural Network (CNN) classifier is used for building the classification model which classifies the frames into 26 different classes representing 26 English alphabets. Finally, the evaluation of the classification model is carried out with test data providing the output in the form of text or voice. The cross-validation accuracy results of 98.66% is achieved from our approach.

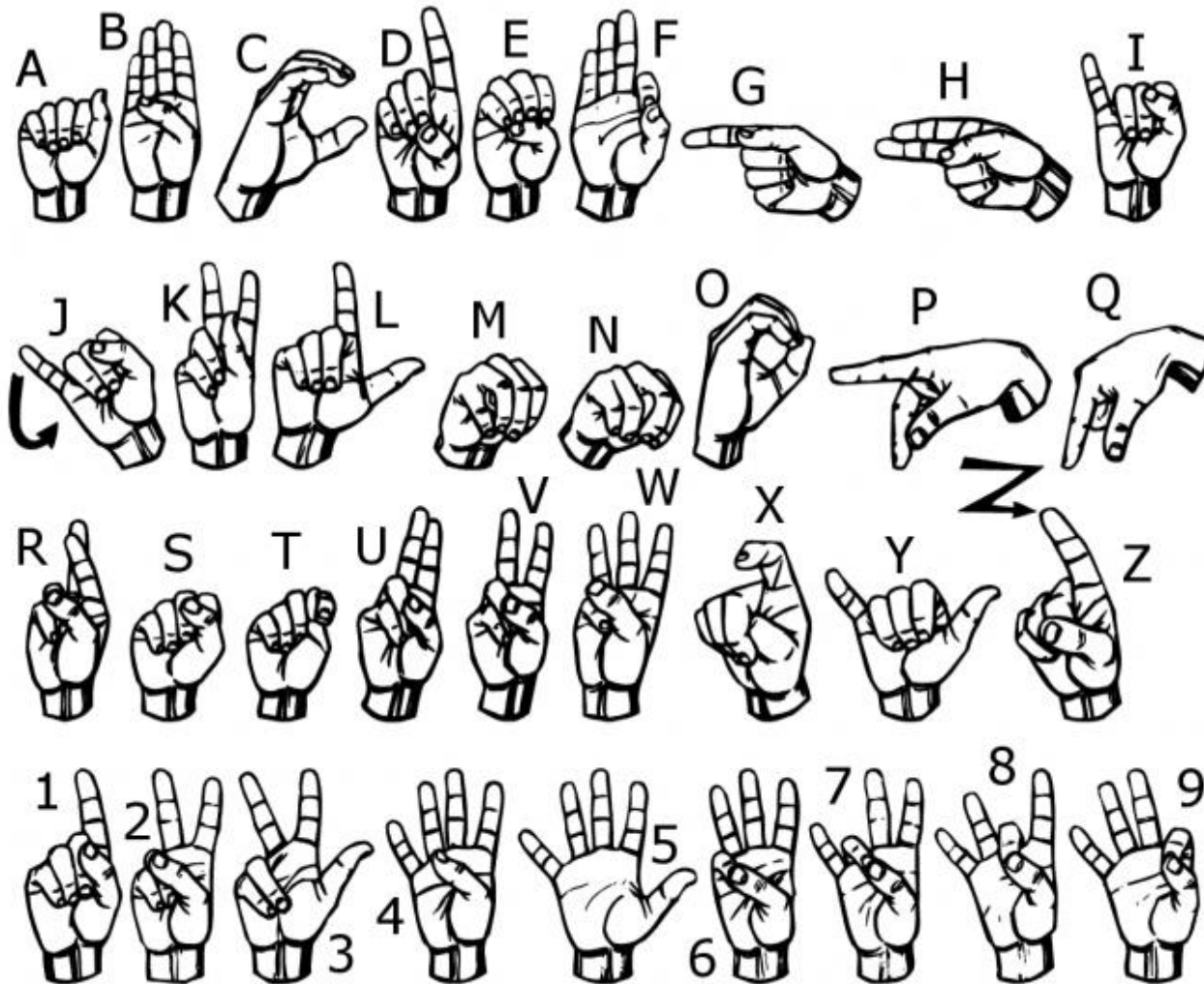
STATEMENT OF PURPOSE

Develop a Machine Learning model for detection of American Sign Language Detection and Conversion. MNIST dataset is used for training the model using Keras and Tensor flow. By applying neural network algorithms we develop a model which can identify sign language with an accuracy of 70%. We use a classification algorithm to classify the image into 26 classes.

DATASET

- MNIST DATASET

- <https://www.kaggle.com/datamunge/sign-language-mnist>



MACHINE LEARNING SCHEMA

Dataset

We will use MNIST (Modified National Institute of Standards and Technology) dataset.

Basically, our dataset consists many images of 24 (except J and Z) American Sign Language alphabets. Each image has size 28x28 pixel which means total 784 pixels per image.

Loading the dataset to Colab

Our dataset is in CSV (Comma-separated values) format. train_X and test_X contain the values of each pixel. train_Y and test_Y contains the label of image. You can use the following code to see the dataset:

Pre-processing

train_X and test_X consists an array of all the pixel pixel values. We have to create an image from these values. Our image size is 28x28 hence we have to divide the array into 28x28 pixel groups. To do that we will use the following code:

2) Build and Train the Model

We will use CNN (Convolutional Neural Network) to recognise the alphabets. We are going to use keras.

As you can observe, like any other CNN our model consists couple of Conv2D and MaxPooling layers followed by some fully connected layers (Dense).

The first Conv2D (Convolutional) layer takes input image of shape (28, 28, and 1). The last fully connected layer gives us output for 26 alphabets.

We are using a Dropout after 2nd Conv2D layer to regularise our training.

We are using softmax activation function in the final layer. Which will give us probability for each alphabet as an output.

The background is a blue gradient with faint concentric circles. White circuit-like lines with circular nodes are positioned in the corners: top-left, top-right, bottom-left, and bottom-right.

Thank You