Bachelor of Science in Computer Science & Engineering

Sign Language Recognition using Deep Learning

by

Khairul Islam Piash
Fahim Hossain Saikat
Fahimul Alam Araf
S.M. Rifatur Rana

ID:
1804076
1804077
1804089
1804098

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

07 March, 2023

Chittagong University of Engineering & Technology (CUET)
Department of Computer Science & Engineering
Chattogram-4349, Bangladesh.

Student Name	Khairul Islam Piash Fahim Hossain Saikat Fahimul Alam Araf S.M. Rifatur Rana
ID	1804076 1804077 1804089 1804098
Session	: 2020-2021
Sumbitted to **Designation**	: Animesh Chandra Roy : Assistant Professor Department of Computer Science & Engineering
Sumbitted to **Designation**	: Atiqul Islam Rizvi : Assistant Professor Department of Computer Science & Engineering
Department **Program**	: Computer Science & Engineering : B.Sc. Engineering
Tentative Title	**: Sign Language Recognition using Deep Learning**

Table of Contents

List of Figures

1 Introduction

Sign languages are visual languages that use the hand, facial and body movements as a means of communication. The movement of the arms, torso and facial emotions are all part of sign language in addition to the hands. As we know deaf people lag behind compared to people who are not physically challenged. To help them communicate and increase their self-esteem of disabled person sign language is a must tool.

Sign language detection can be achieved using deep learning by python. Here for converting the sign language to English text. The TensorFlow zoo model is used for image data processing. By applying this project detecting sign language with moderate efficiency can be achieved.

2 Literature Review

In Deep learning and computer vision sign language detection become an active area of research. Many researchers work in this field to help people in learning and understanding sign language. Now, we will discuss some prior work done on this topic.For many years, research on different sign languages in the world and most common sign languages are British Sign language, American Sign Language, Indian sign lan- guage,Russian sign language, and many more. In this paper, we focus on the American Sign Language to detect various gestures using computer vision and Tensorflow object detection. Karma Wangchuck et al. (2020) [1] have presented Real-time Bhutanese Sign Language digits recognition system using a Convolutional Neural Network. In this paper, they worked on BSL digit recognition using a convolutional neural network and create the world's first Bhutanese sign language dataset. Kumud Alok et al. (2020) [2] have presented hand sign recognition using a convolutional neural network. In their study,they made use of the convolution neural networks, to identify the images of the hand signs. Sajanraj and Beena (2018) [3].

3 Specific Objectives

The main objective of this work is to translate sign language into text. The specific objectives are:

- To assist the deaf to communicate with others and amongst themselves.
- To give deaf children the opportunity to educate themselves.
- Training and testing Tensorflow model Zoo using the collected data.
- To classify the different types of sign directions.

.

4 Required Resources

The necessary tools to implement this project can be divided into two Categories-Hardware and Software as illustrated below:

- Personal Computer with Operating System
- Python IDLE
- Images data set
- Jupyter Notebook
- WebCam

5 Methodology

We organized the experiment into five steps for this model. The collection of photos using a camera and OpenCV was the initial stage. Labeling images with labeling for detection was the second stage. We set up a Tensorflow object detection pipeline in the third step. We trained the model with transfer learning in the fourth stage. Real-time sign language detection using OpenCV and a webcam was the final stage.

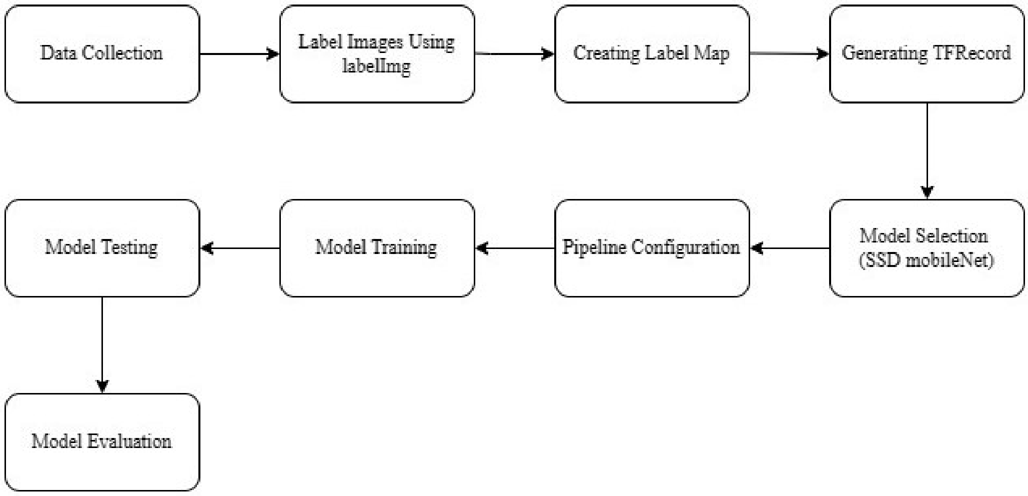


Figure 5.1: Methodology of the proposed system

The methodology for sign language detection using TensorFlow Object Detection API can be broken down into the following steps:

- Data Collection: Collect a dataset of sign language images and annotate them with bounding boxes around the signs. This dataset will be used for training and evaluating the sign language detection model.
- Data Preparation: Convert the annotated dataset into a format that can be used by TensorFlow Object Detection API, such as the TFRecord format.
- Model Selection: Choose a suitable object detection model architecture from the available models in TensorFlow Object Detection API, such as Faster R-CNN, R-FCN, or SSD.
- Pipeline Configuration: Create a pipeline configuration file that specifies the settings for training and evaluating the model, such as the model architecture, input data, and evaluation metrics.
- Model Training: Train the model on the annotated dataset using the pipeline configuration file. Monitor the training process and adjust the hyperparameters as necessary to achieve the best results.
- Model Evaluation: Evaluate the model on a test set of data to measure its performance in detecting sign language signs.

5.1 Data Collection

The dataset was collected using OpenCV and webcam. Firstly,we collected 20 images each of 5 gestures i.e.yes,no,hello,thanks,iloveyou.Then we separated into train and test in the ratio 4:1(Figs.5.2 and 5.3) . 80% of the sign language data is used for training and 20% is used for testing.

Figure 5.2: Sign language for hello

Figure 5.3: Sign language for iloveyou

5.2 Labelling Images

Labeling images is the process of annotating an image with descriptive information or metadata. This can include object labels, attributes, actions, and relationships between objects in the image.

We use labeling to annotate images.It is easy to use LabelImg tool to label images.In this,we draw a rectangle across the sign language gesture we are making and then name that gesture.This will allow us to separate each gesture into different categories.Once the annonations for images are done,we have saved the images in our train test folders.(Fig.5.4)

The XML(eXtensible Markup Language) files will be created for each data image in both test and train folder.XML files consists of the coordinator of gesture bounding boxes,category,filename,etc. for each sign language gesture inside the image.(Fig.5.5)

5.3 Create Label Map

We created a label.pbtxt file to convert label name to a numeric id.By creating a label map, the TensorFlow Object Detection API can correctly interpret the

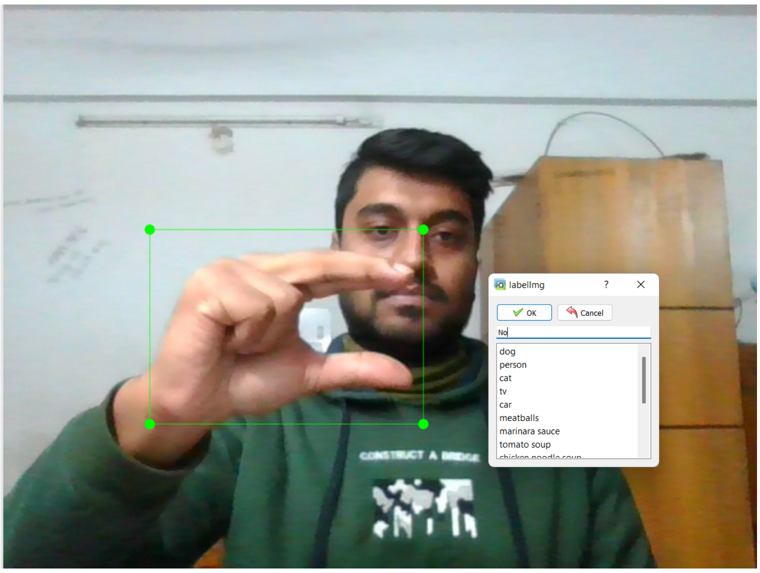
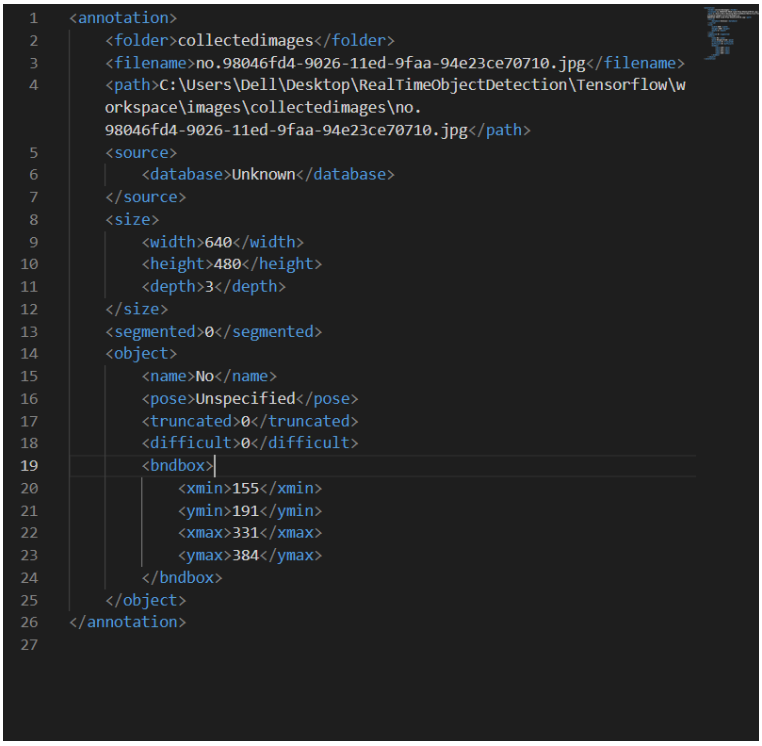


Figure 5.4: Annotated Image



```xml
<annotation>
    <folder>collectedimages</folder>
    <filename>no.98046fd4-9026-11ed-9faa-94e23ce70710.jpg</filename>
    <path>C:\Users\Dell\Desktop\RealTimeObjectDetection\Tensorflow\workspace\images\collectedimages\no.98046fd4-9026-11ed-9faa-94e23ce70710.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>640</width>
        <height>480</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>No</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>155</xmin>
            <ymin>191</ymin>
            <xmax>331</xmax>
            <ymax>384</ymax>
        </bndbox>
    </object>
</annotation>
```

Figure 5.5: XML file

class labels in the annotations and use them to train and evaluate the model. The label map is a critical component of the TensorFlow Object Detection API, and it is important to ensure that it is created accurately and correctly to ensure the best possible results.

```
item {
      name:'Hello'
      id:1
}
item {
      name:'Yes'
      id:2
}
item {
      name:'No'
      id:3
}
item {
      name:'Thanks'
      id:4
}
item {
      name:'I Love You'
      id:5
}
```

Figure 5.6: Label.pbtxt

5.4 TFRecord Generation

Tensorflow API accepts datasets in TFRecord file format,which is a sequence of binary format strings.It is an efficient and fast way to store large datasets that can be read quickly by TensorFlow for training models.

5.5 Model Selection

SSD MobileNet is a deep learning architecture that was developed specifically for object detection tasks in real-time applications.The basic idea behind SSD MobileNet is to detect objects in an image using a single shot, rather than multiple passes through the image, making it faster and more efficient than other object detection models. It is best suited for real time object detection.

5.6 Training the model

Our model is based on the Single Shot MultiBox Detector (SSD) architecture with MobileNet. We collected our own dataset and this dataset was divided into two: training and testing. We trained our model with 20,000 steps, with the training configuration and model checkpoint stored in the directory. We used SSD

MobileNet V2 model which is pre-trained model to train with our dataset. The model takes the preprocessed sign images and classifies it into the corresponding category.

5.7 Testing the model

To test and verify the model which we trained with our dataset. We tested our model with several sign langauage pcitures. Then we used 4-5 individuals for the real time sign language detection.

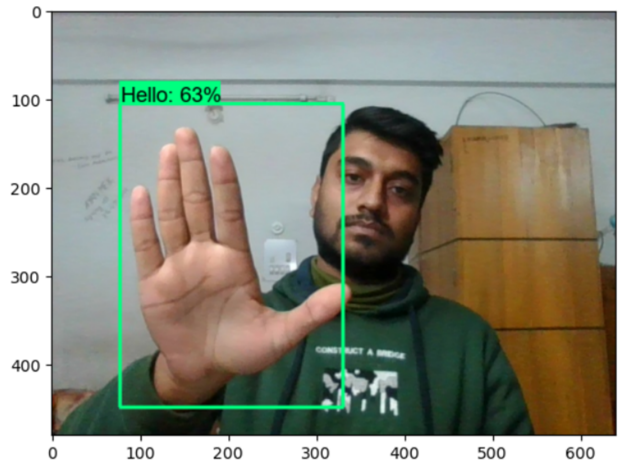


Figure 5.7: Test Image

5.8 Evaluation

Minimum average precision is a metric used to evaluate the performance of a deep learning model in TensorFlow. It measures the average precision across all classes, given a certain number of predicted labels per image. It is a commonly used metric for object detection tasks, as it provides a balance between precision and recall, considering both the number of false positives and false negatives. In TensorFlow, the metric can be calculated using the tf.metrics.AveragePrecisionAtK function. The value of k determines the number of predictions per image to consider when computing the average precision.

In deep learning, recall is an important performance metric used to evaluate the accuracy of a model's predictions. Recall measures the proportion of positive

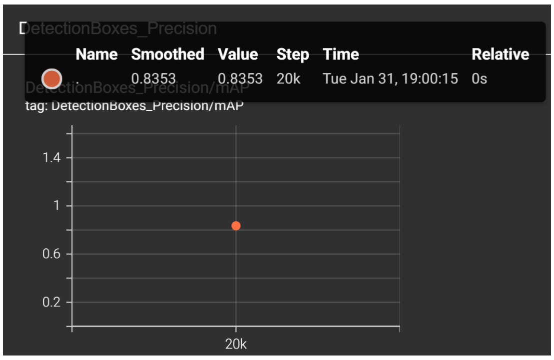


Figure 5.8: Minimum Average Precision

examples that were correctly identified by the model, also known as the true positive rate.

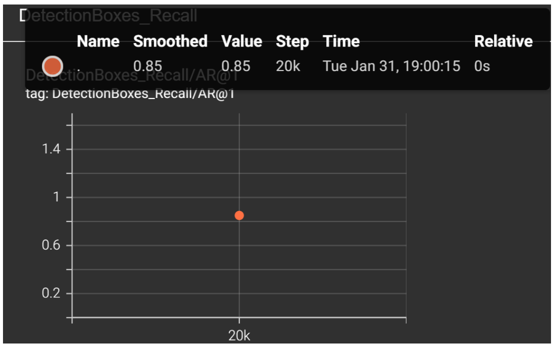


Figure 5.9: Recall Value

6 Real Time Detection

Real-time object detection refers to the ability to detect and classify objects in a video stream or camera feed in near real-time.We used OpenCv to capture the video frames and display the detection results on the screen.Some outputs are shown below:

7 Conclusion

The project classifies gestures of sign language using deep learning. For sign language, we have standardized American sign language. For better image data processing we have used the Zoo model of tensor flow. To conclude the future purpose is to enhance the recognition system by expanding its capabilities to process and recognize a wider variety of sign images.

Figure 6.10: Sign language detection model output

References

[1] K. Wangchuk, P. Riyamongkol and R. Waranusast, 'Real-time bhutanese sign language digits recognition system using convolutional neural network,' *ICT Express*, vol. 7, Sep. 2020. DOI: `10.1016/j.icte.2020.08.002` (cit. on p. 1).

[2] T. D. Sajanraj and M. Beena, 'Indian sign language numeral recognition using region of interest convolutional neural network,' in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2018, pp. 636–640. DOI: `10.1109/ICICCT.2018.8473141` (cit. on p. 1).

[3] S. Salian, I. Dokare, D. D. Serai, A. Suresh and P. Ganorkar, 'Proposed system for sign language recognition,' *2017 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC)*, pp. 058–062, 2017 (cit. on p. 1).