IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 2, February 2022

DOI: 10.17148/IJARCCE.2022.11220

Sign language Recognition

Uma Thakur[1], Pariksheet Shende[2], Rajat Bais[3], Priyanka Karamkar[4], Rushika Bhave[5], Jayesh Mankavade[6]

Professor, Computer Science and Engineering Department, Priyadarshini College of Engineering, Nagpur, India[1]

Research Scholar, Computer Science and Engineering Department. Priyadarshini College of Engineering, Nagpur, India[2,3,4,5,6]

Abstract: The goal of this research is to construct an accurate sign language recognition model by experimenting with several segmentation methodologies and unsupervised learning algorithms. We tested with just up to 10 different classes/letters in our self-made dataset instead of all 26 potential letters to make the problem easier to approach and produce reasonable results. Using a Microsoft Kinect, we acquired 12000 RGB photos and their related depth data. The autoencoder was used to extract features from up to half of the data, while the other half was used for testing. Using our trained model, we were able to attain a classification accuracy of 98 percent on a randomly selected set of test data. We built a live demo version of the project in addition to the work we did on static photos.Techniques for colour and depth segmentation were the most reliable.

Keywords: RGB, Kinect, sign language, accuracy, algorithm, reasonable, dataset.

I. INTRODUCTION

The topic we're looking into is unsupervised feature learning for sign language recognition. Recognizing sign language is a fascinating computer vision problem that is also incredibly beneficial for deaf persons who want to communicate with people who don't understand American Sign Language (ASL). To begin, we created a data collection that included the various ASL alphabet hand motions that we wanted to identify. After that, only the hand region of each image was segmented, and the data was used for unsupervised feature learning with an autoencoder, followed by training a softmax classifier to determine which letter was being displayed. We attempted a variety of segmentation techniques until we determined that the skin was the most difficult to segment.

II. RESEARCH METHODOLOGY

2.1. Problem Description

2.1.1 Background: The CVPR gesture workshop from 2011 is a wonderful resource for learning about modern gesture recognition models and how to incorporate various learning methods. We initially looked at some previous work1 related to our topic, such as segmentation-robust modelling for sign language recognition and sign language and human activity recognition, but we ultimately used largely our own technique to sign language recognition. The MNIST handwritten digits identification problem2 provided inspiration for our learning model, which used a similar unsupervised feature learning and classification approach. We also looked into the usage of convolutional neural networks for feature learning, based on a Microsoft Research study on visual document processing.

2.1.2 Dataset: The data for training and testing came from a dataset that we created ourselves. The Kinect was used to capture 1200 samples of each of the 10 signed letters (a, b, c, d, e, f, g, h, I l), as well as the accompanying depth data. A person signs the corresponding letter while facing directly at the Kinect camera in each sample. A total of 6000 photos were utilised for training and another 6000 images were used for testing in this dataset.

III. INFORMATION SOURCES

3.1 Data Gathering:

The ASL Alphabet, a constructed dataset of American Sign Language (ASL) by Kaggle user , was the primary source of data for this project. The collection contains 87,000 photos at a resolution of 200x200 pixels. There are 29 total classes with 3000 images each, 26 for letters A-Z and three for space, delete, and nothing. The images acquired from his laptop's webcam are purely of the user gesturing in ASL. Following that, the photographs were cropped, rescaled, and labelled for use.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 2, February 2022

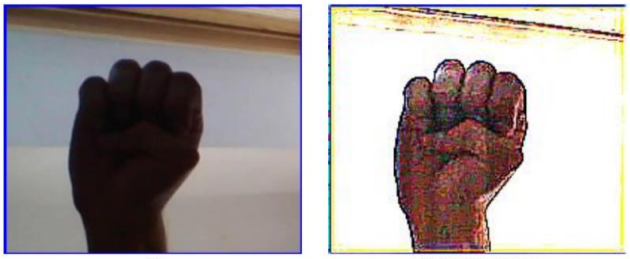
DOI: 10.17148/IJARCCE.2022.11220

(a) ASL letter A (b) ASL letter E (c) ASL letter H (d) ASL letter Y

3.2 Image Enhancement: The photographs were enhanced using a combination of brightness, contrast, sharpness, and colour enhancement. For example, the contrast and brightness were adjusted so that fingers could be distinguished when they were placed on the screen.

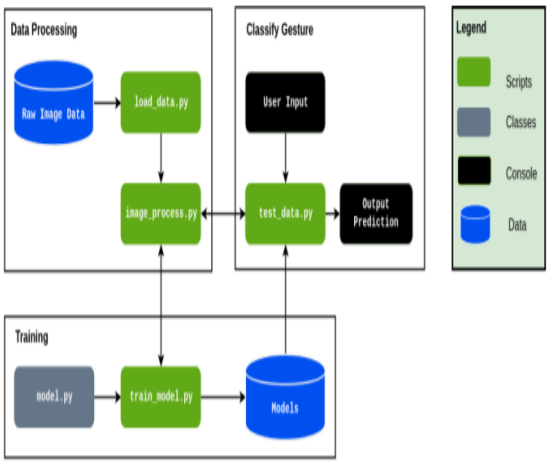
3.3 Edge Enhancement: Edge enhancement is a type of picture filtering that improves the definition of edges. This is accomplished by increasing contrast in a local picture region that is detected as an edge. The boundary of the hand and fingers vs the background becomes considerably more obvious and prominent as a result of this. This may aid the neural network in recognising the hand and its limits.

3.4 Image Whitening: Image whitening, also known as ZCA, is a technique that use a matrix's singular value decomposition. This method cleans up the data by removing any redundant or obvious information. This enables the neural network to search for more complicated and sophisticated associations as well as the underlying structure of the data.



(a) (b) Preprocessed Image

IV. A DESCRIPTION OF THE SOFTWARE'S OVERALL STRUCTURE



The project will be divided into three independent functional blocks, as indicated in Figure 1, Data Processing, Training, and Classify Gesture. To abstract some of the intricacy, the block diagram has been simplified in detail:

Data Processing: This is where the data is loaded. The raw image data is loaded into the py script, and the image data is saved as numpy arrays into file storage. The information collected during the process. The image data will be loaded from data.npy, and the image will be pre-processed by resizing/rescaling the image, as well as applying filters and ZCA whitening to enhance features. The processed picture data was divided into training, validation, and testing data and written to storage during training. A load dataset is also used during training. A python script that loads the essential data, which is divided into a Dataset class is a type that represents a collection of data. An individual image is loaded and processed from the disc for use with the trained model in categorising gestures.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 2, February 2022

DOI: 10.17148/IJARCCE.2022.11220

Model training: train model.py contains the model's training loop. The learning rate, batch size, image filtering, and number of epochs are all acquired from a config file that is used to train the model. The model architecture and the parameters used to train it are retained for future review and adjustment for better outcomes. The training and validation datasets are loaded as Data loaders in the training loop, and the model is trained with Adam Optimizer with Cross Entropy Loss. Every epoch on the validation set, the model is evaluated, and the model with the best validation accuracy is chosen was saved to storage to be evaluated and used later. The training and validation error and loss, as well as a plot of error and loss over training, are stored to the disc once the training is completed.

Categorize Gesture: Once a model has been trained, it may be used to classify a new ASL gesture that has been saved to the file system. The user enters the gesture image's file path as well as the test data. To handle data, the py script will pass the file location. py to load and pre-process the file similarly to how the model was learned.

V. MODELS OF MACHINE LEARNING

5.1 Overarching Framework: The model utilised in this classification job is a simple Convolutional Neural Network implementation (CNN). Because the project necessitates picture categorization, a CNN is the preferred architecture. The publication Using Deep Convolutional Networks for Gesture Recognition in American Sign Language provided the foundation for our model design. Consider an ASL Gesture Classification job similar to this. Convolutional blocks with two 2D Convolutional Layers with ReLU activation, followed by Max Pooling and Dropout layers made up this model.

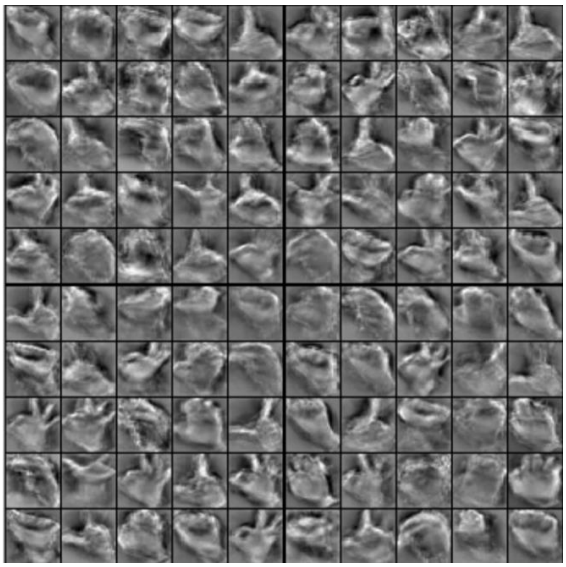


Figure: Visualization of sparse auto encoder features.

These convolutional blocks are repeated three times before being followed by Fully Connected layers that classify the data into the appropriate categories. Throughout the model, the kernel sizes are kept constant at 3 X 3.The next stage is to classify the ten different letters using the auto encoder training characteristics. The output of the autoencoder's hidden layer is sent into a softmax classifier, which divides the data into ten groups. The L-BFGS optimization function is used to train the softmax classifier once more. After around 40 iterations, this algorithm converges. We used the remaining 600 photographs per letter (for a total of 6000 images) as our test set to determine the system's accuracy.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 2, February 2022

DOI: 10.17148/IJARCCE.2022.11220

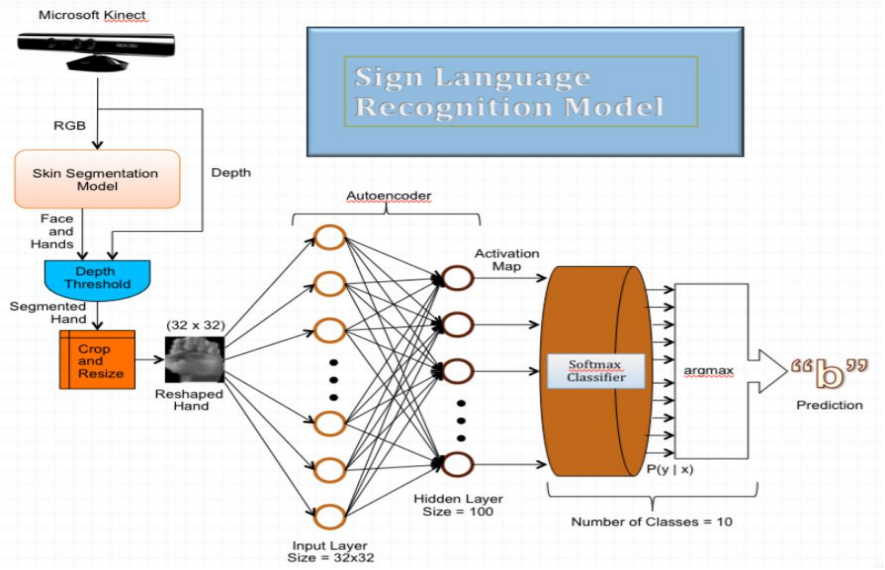


Figure: A block diagram depicting our method for recognising sign languages.

As a final stage in our research, we successfully implemented our complete system in real time, allowing hand motions made in front of the Kinect connected to our computer to be recognised. The image recorded by the kinect, the segmentation hand gesture, and the output of our classifier, which is one of the ten letters in our dataset, were all shown directly by CompuCom. Per frame, the review procedure takes less than 2 seconds. The screenshots of our real-time implementation and the results gained are shown in the following images. The original image is shown in each screenshot, along with the segmentation result and the anticipated result to the right.

Figure: Screenshots from a real-time live demonstration of a sign language recognition system.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 2, February 2022

DOI: 10.17148/IJARCCE.2022.11220

VI. FINAL THOUGHTS AND FUTURE WORK

We used methods learned in computer vision and machine learning to construct an autonomous sign language gesture recognition system in real-time in this project. We discovered that sometimes simple solutions are more effective than complex solutions. Despite using a sophisticated segmentation technique, the most effective skin masks were extracted using a simple skin segmentation model. We also recognised the time restrictions and challenges of building a dataset from the ground up. Looking back, having a dataset to work with would have been preferable. In our live demonstration, some letters were more difficult to classify, such as "a" vs. I because they only differ by one letter. By a very slight margin (the pinky of the I points up). Despite the fact that our classification approach works very well, as evidenced by the tables and photographs, there is still a lot of room for future research.

VII. REFERENCES

[1] D. Metaxas , CVPR hosts a workshop on gesture recognition.M. Ranzato.

[2] Energy-based paradigm for efficient sparse representation learning, 2006. The Courant Institute of Mathematical Sciences could be a mathematical sciences research institute.

[3] S. Sarkar. The CVPR Workshop on Gesture Recognition is co-authored by Barbara Loeding, Ruiduo Yang, Sunita Nayak, and Ayush Parashar.

[4].Teng, X. April 2005: Local linear embedding was supported by a hand gesture detection system. Visual Languages and Computers may be a specialised journal to visual languages and computing.

