Chinese Sign Language Classification

Nada Abdellatef
Electrical Engineering
Ottawa University
nsedd099@uottawa.ca

Hadeer Mamdouh
Electrical Engineering
Ottawa University
hmoha156@uottawa.ca

Khaled Elsaka
Electrical Engineering
Ottawa University
kelsa047@uottawa.ca

Mostafa Nofal
Electrical Engineering
Ottawa University
mnofa091@uottawa.ca

***Abstract*— Sign language is the easiest form of communication for people who have trouble hearing. Sign language recognition is an important task when it comes to converting sign language into text. The current sign language recognition methods are broadly divided into two types: the traditional machine learning method using image features and the deep neural network-based method. The following aspects best describe the primary contributions of this study. The training and testing of the ResNet model were done by combining different feature engineering techniques. The training data represents the baseline for the classification task, and the test data is the result of fine-tuning the pre-trained model with some hidden layers to make it more accurate. In summary after applying several classification models which are linear SVM, RBF kernel SVM, Random Forest, MLP, Naïve Bayes, and XGBoost. The best model was MLP with a test accuracy of 63% and after applying hyperparameter tuning it became 64%.**

Keywords— Sign Language Classification, Image Classification, Hand Gestures

I. INTRODUCTION

Nowadays, about 5% of the population suffers from the issue of hearing loss in the world. There is also a large number of hearing-impaired people in China, number of which 27.9 million. [1] For people who have trouble hearing, sign language is the easiest form of communication. However, sign language also includes a variety of content and components, including hand form, movement, posture, and emotion. The majority of sign language simply uses the upper body, from the waist up. Additionally, a sign's shape can alter significantly depending on where it appears in the text. Conversational gestures, controlling gestures, manipulative gestures, and communicative gestures are a few categories into which hand gestures can be divided. A sort of gesture used for communication is sign language. Since sign language has a strong structural foundation, it can serve as a testing ground for computer vision algorithms. As a result, it is a relatively complex system that is difficult to learn and master. The above problems can be solved in part through sign language translation and recognition. However, the former is more expensive and requires advanced personnel arrangements, whereas the latter is gaining popularity as intelligent technology advances quickly. Chinese Deaf and Hearing-Impaired Groups are the target audience for Chinese Sign Language (CSL). It has distinctive qualities like as deep semantics, a broad scope, and a variety of expressions. Sign language recognition is an important and difficult task when it comes to converting sign language into text. The current sign language recognition methods are broadly divided into two types: the traditional machine learning method using image features and the deep neural network-based method. The former uses machine learning techniques to perform classification based on the image features of the interested area. The former uses traditional image segmentation algorithms to segment hand shapes from sign language images or video frames of sign language video. In order to represent the above CSL gestures, which can be completed with varying numbers of stretched-out fingers and in various orientations, this paper intends to construct orientation-sensitive and robust features.

II. RELATED WORK

[2] An efficient technique for the recognition of Indian Sign Language ISL letters, words, and numbers used in daily life is provided in this study. Convolutional layers are the first in the proposed CNN architecture, followed by ReLU and max-pooling layers. Different filtering window sizes are included in each convolutional layer, which helps to increase recognition speed and precision. A web camera-based dataset of 35,000 images from 100 static signs has been generated. The proposed architecture has been tested on approximately 50 deep learning models using different optimizers. The system results in the highest training and validation accuracy of 99.17% and 98.80%, respectively, with respect to different parameters such as the number of layers and filters. A variety of optimizers were used to test the suggested system, and it was discovered that SGD beat Adam and RMSProp optimizers, with training and validation accuracy on the grayscale picture dataset of 99.90% and 98.70%, respectively. The proposed system is robust enough to learn 100 different static manual signs with lower error rates. It has been found that the system outperformed other existing systems even with a smaller number of epochs. The major source of challenge in sign language recognition is the capability of sign recognition systems to adequately process many different manual signs with low error rates.

[3] To classify selfie sign language gestures, they suggested using the CNN architecture. Four convolutional layers make up the CNN architecture. The consideration of each convolutional layer with a particular filtering window size increases identification speed and precision. The benefits of the max and mean pooling techniques are combined in a stochastic pooling strategy. A total of 300000 sign video frames were produced when we generated the selfie sign language data set using 200 ISL signs with 5 signers in 5 user dependent viewing angles for 2 seconds each at 30 frames per second. To determine the reliability of the large training modes needed for CNNs, training is done in many batches. The training is carried out with three sets of data (i.e., 180000 video frames) in Batch-III in order to maximize the recognition of the SLR. This CNN architecture has higher training accuracy and validation

accuracy than the previously suggested SLR models which were Mahalanobis distance classifier (MDC), Adaboost classifier and artificial neural network (ANN). The suggested CNN architecture shows less loss during training and validation. Comparing the proposed CNN model to existing state-of-the-art classifiers, the recognition accuracy rate is higher at 92.88%.

[4] The system can detect one or two hands in a video stream in real-time. The pipeline of the experiment is as follows. Capturing frames using a Logitech HD C310 webcam. Then he used the Open-CV library to find hand contour, convex hull, and defect points, palm center localization and stabilization, fingertips identification, and finally used SVM as the classifier. After identifying individual fingertips, gestures can be classified by detecting the number of fingers. If it is five, then it is an open palm and if it is zero, then it is a closed palm. To do this, a Support Vector Machine classifier was used which acts as a separating hyperplane. Regarding the results, the best accuracy achieved was over 85%. The model worked well for every gesture. However, it is not ideal, because not all the fingers are classified correctly for every case.

[5] The authors used feature reduction by applying LDA or PCA then tried different supervised machine learning algorithms to classify Electromyography data into 7 different human hand gestures. The traditional classification algorithms were K- Nearest Neighbor (KNN), Naive Bayes, Decision Tree, and Random Forest. They used also deep learning classifiers such as Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM). For the results, the Random Forest classifier is the best model that achieves 99.43% accuracy, and the LSTM model gives 99.19% accuracy. The models are reliable and applicable because they calculated the accuracy and error graph for the train, test, and validation set. EMG data patterns for different hand gestures have differences from each other, so it is easy for the Random Forest ensemble to classify them perfectly.

III. DATA

This dataset consists of 500 RGB images of Chinese numbers in finger sign language. The Chinese numbers finger sign language consists of 10 signs in accordance with the state-issued universal sign language standard. Figure 1 demonstrates one category of Chinese finger sign language intercepted from sample images. The images were collected and taken by each member of the research team. These images were resized to 128 × 128. Then we tried various types of filters, preprocessing, and data augmentation which are explained in the methodology and challenges section. Our experiment was executed with this private dataset including 500 images. Among them, 400 images were used for training, and the rest were used for testing.

Figure 1. Chinese Numbers

IV. METHODOLOGY

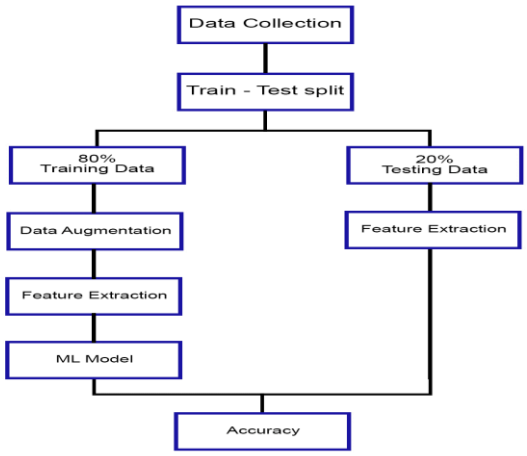

Figure 2. Overview Architecture.

Using transfer learning to fine-tune the pre-trained ResNet model by freezing the convolution layers and adding some hidden layers for the classification task thin computing the accuracy of the test data to represent our baseline figure 3.

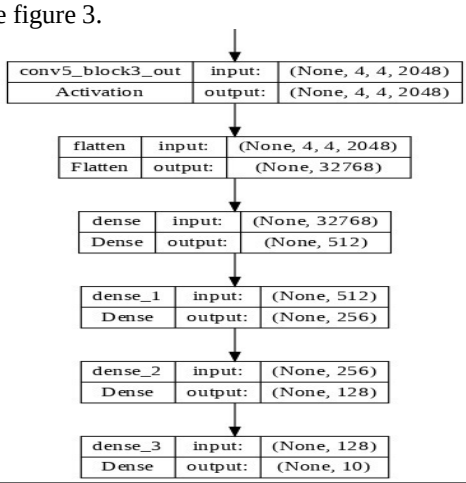

Figure 3. Baseline Architecture

Combining different feature engineering techniques:

1- We know that the convolution layers act as feature extractors, so we collect the features from the last convolution layer, then add 1024 (1*1) convolution layer to reduce the dimensions, and by applying global average pooling we got 1024 feature vector as output figure 4.

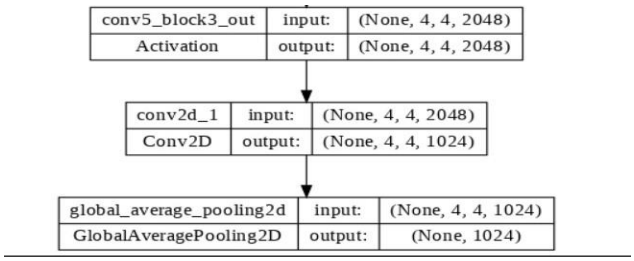

Figure 4. ResNet Feature Extractor Architecture

2- Using the Zernike [7] [13] moments after applying a binary mask on the images for hand pixels to be 1 and 0 otherwise. Zernike moment requires 2 important parameters the first one is the radius and the other one is the degree of the polynomial. The radius is for a circle that surrounds the target object and the polynomial degree represents the shape. Using the radius value to be half of our image size and the default degree of polynomial which is 8. The output number of features, in this case, is 25. More details are explained in the challenges section and the final architecture for feature extraction is in figure 9.

After concatenating the two methods we got 1049 features. Then, we feed them into ML models which are linear SVM, RBF kernel SVM, Random Forest, MLP, Naïve Bayes, and XGBoost. Selecting the champion model based on the average between the training and testing accuracy and comparing the results with the baseline performance.

The input for the ResNet model is the images without any normalization or scaling but after feature extraction, a standard scaler is applied after the concatenation and before training the classification models.

V. Challenges

To convert the images to binary and be ready for the Zernike moments we faced some challenges:

- Noisy images

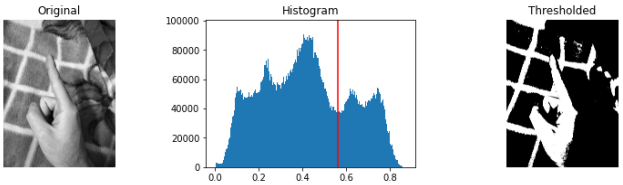

Figure 5. Otsu Thresholding [11] for noisy images

The background in figure 5 has white lines beside the hand, making it hard to separate the hands from the background noise.

- Different lightening

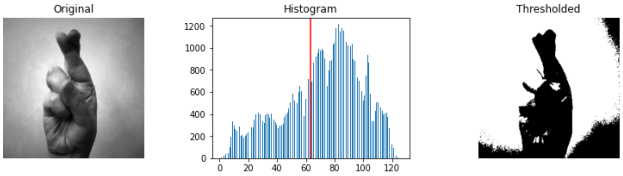

Figure 6. Otsu Thresholding with different lightening

Another problem is that we could have different lighting for example, in figure 6 light on the background, not on the hand and that leads us to change the direction of the threshold.

To solve these problems we removed the background in figure 7 using the Rembg [10]. Then it becomes easier to apply the Zernike image pre-processing [12] and get the largest contour, which corresponds to the outline of the hands. Finally, the contours were drawn as a filled-in mask with white pixels figure 8.

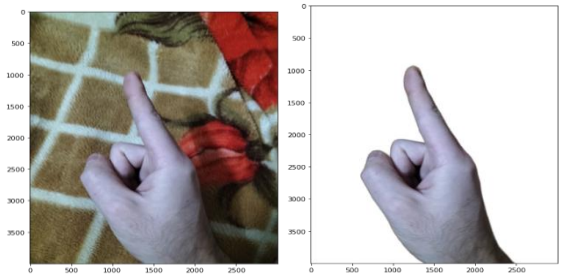

Figure 7. Remove Background

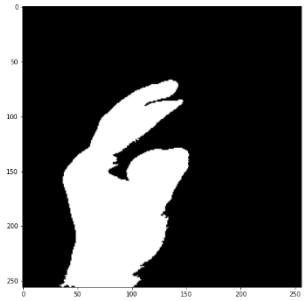

Figure 8. Zernike Moment pre-processed image

Sometimes, background removal doesn't do well if the size of the image is small and the hands aren't clear.

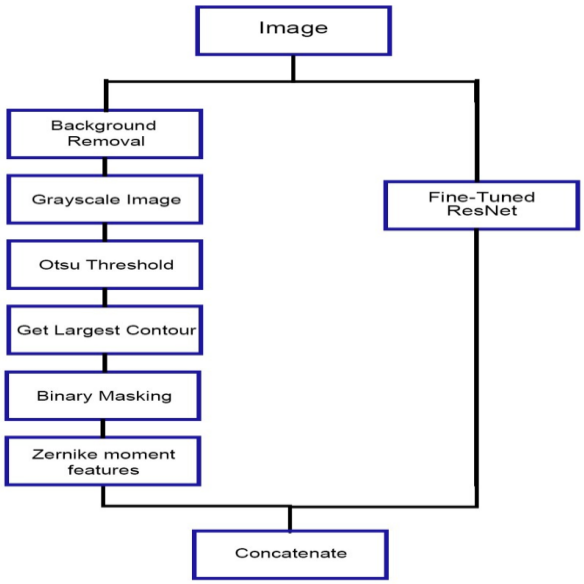

Figure 9. Feature Extraction and Concatenation Architecture

VI. Methods & Evaluation

For the baseline, we used ResNet model because it's one of the most popular pre-trained models in Image Classification and contributes to the network's ability to maintain low error rates at deeper levels.

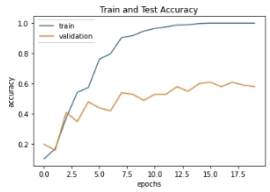

Figure 10. Train and Test Accuracy

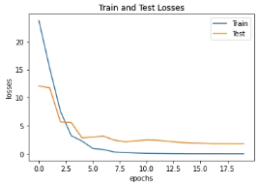

Figure 11. Train and Test Losses

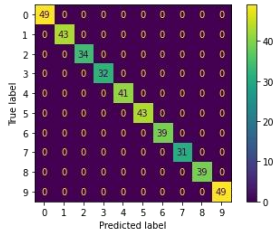

Figure 12. Train Confusion Matrix

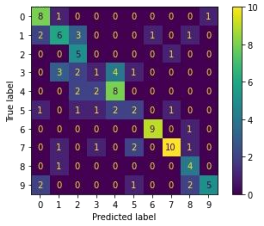

Figure 13. Test Confusion Matrix

We added some layers and fine-tuned the output from 1000 to 10 classes in our case.

We can tell from the figures that the training accuracy is around 100% and the test accuracy is 58% so there is overfitting.

After that, we combined the features from the ResNet model with the features from Zernike moments and used them to evaluate the machine-learning models.

- Accuracy Comparison of the Models

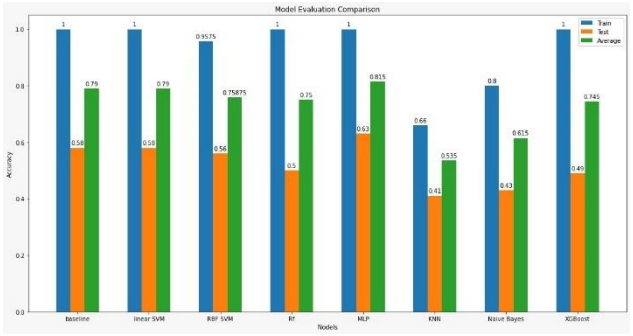

Figure 14. Model Evaluation Comparison

We tried many algorithms, and the champion model was multi-layer perception with 63% accuracy compared with the other models

We tuned the hyperparameters and found that the accuracy reached 64% and the best parameters were:

- ('activation', 'logistic')
- ('alpha', 0.003)
- ('learning_rate', constant')
- ('learning_rate_init', 0.0001)
- ('max_iter', 500)

- Photos sample visualization after data augmentation

Figure 15. Data Augmentation

Here is the Accuracy Comparison of the Models with data augmentation

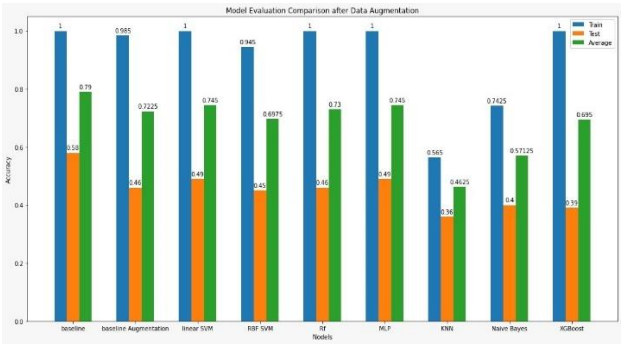

Figure 16. Models Comparison after data Augmentation

We used data augmentation to solve the problem of overfitting so some of the model's accuracies decreased but most of the results didn't change too much.

VII. Conclusion

Combining the two feature engineering techniques from the ResNet pre-trained model and Zernike moments results in improving performance over the baseline with a training accuracy of 99.5% and testing accuracy of 64% for the tuned best model MLP. The model is overfitting because the training data is very few. Although we've tried data augmentation, the results didn't meet our expectations so, we plan to increase the size of the images to have better results from the Rembg library for background removal. Also, building the MLP using Keras instead of Sklearn to have more flexibility and solve the overfitting problem, and tuning the Zernike Moments for the radius and degree of the polynomial. Another important factor that we should consider is the number of dimensions, the output from the Zernike is 25 features but the output from the ResNet is 1024, so realizing that the Zernike features represent only 2.4% leads the models to be highly biased for the ResNet features and the Zernike has a small effect. At this point, there is a variety of solutions that we could try to achieve better results from this flexible architecture, for example, reducing the number of ResNet output features to be equal to the Zernike or maybe fewer or even increasing the number of features from the Zernike by increasing the degree of the polynomial hyperparameter.

VIII. REFERENCES

[1] X. Jiang, B. Hu, S. Chandra Satapathy, S. H. Wang, and Y. D. Zhang, "Fingerspelling Identification for Chinese Sign Language via AlexNet-Based Transfer Learning and Adam Optimizer," Sci Program, vol. 2020, 2020, doi: 10.1155/2020/3291426.

[2] A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs," Neural Comput Appl, vol. 32, no. 12, pp. 7957–7968, Jun. 2020, doi: 10.1007/s00521-019-04691-y.

[3] G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," in 2018 Conference on Signal Processing And Communication Engineering Systems, SPACES 2018, Mar. 2018, vol. 2018-January, pp. 194–197. doi: 10.1109/SPACES.2018.8316344.

[4] G. N. Pham, "International Journal of Multidisciplinary Research and Growth Evaluation Experiment hand gesture recognition and classification using machine learning algorithm," vol. 2, no. 5, pp. 337–339, [Online]. Available: www.allmultidisciplinaryjournal.com

[5] A. B. Habib, F. bin Ashraf, and A. Shakil, "Finding Efficient Machine Learning Model for Hand Gesture Classification Using EMG Data," in 2021 5th International Conference on Electrical Engineering and Information and Communication Technology, ICEEICT 2021, 2021. doi: 10.1109/ICEEICT53905.2021.9667856.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks." [Online]. Available: http://code.google.com/p/cuda-convnet/

[7] Zernike, F.: Beugungstheorie des Schneidenverfahrens und seiner verbesserten Form, der Phasenkontrastmethode. Physica 1(7–12), 689–704 (1934)

[8] Barbhuiya, A.A., Karsh, R.K. & Jain, R. A convolutional neural network and classical moments-based feature fusion model for gesture recognition. Multimedia Systems 28, 1779–1792 (2022). https://doi-org.proxy.bib.uottawa.ca/10.1007/s00530-022-00951-5

[9] Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3), 1–27 (2011)

[10] danielgatis, "GitHub - danielgatis/rembg: Rembg is a tool to remove images background.," *GitHub*, Nov. 28, 2022. [Online]. Available: https://github.com/danielgatis/rembg. [Accessed: Dec. 01, 2022]

[11] "Module: filters — skimage v0.19.2 docs," *Module: filters — skimage v0.19.2 docs*. [Online]. Available: https://scikit-image.org/docs/stable/api/skimage.filters.html#skimage.filters.threshold_otsu. [Accessed: Dec. 01, 2022]

[12]A. Rosebrock, "HOW-TO: Indexing an image dataset using Zernike moments and shape descriptors.," *PyImageSearch*, Apr. 07, 2014. [Online]. Available: https://pyimagesearch.com/2014/04/07/building-pokedex-python-indexing-sprites-using-shape-descriptors-step-3-6/. [Accessed: Dec. 01, 2022]

[13] Application of Zernike moments in computer vision problems for infrared images Dmitriy Otkupman, Sergey Bezdidko, Victoria Ostashenkova
E3S Web Conf. 310 01002 (2021)
DOI: 10.1051/e3sconf/202131001002