Sign Language Recognition Using Deep Learning

KAVIYA BIRAVI N
II M.SC Inforamtion Technology
Avinashilingam Institute for Home Science and Higher Education for women

Dr. N. Krishnaveni M.Sc., M.Phil., SET, Ph.D,
Assitant professor,
Department of information Technology,
School of physical Sciences and computational Sciences,
Avinashilingam Institute for Home Science and Higher Education for women

ABSTRACT*Communicating with deaf and speech imp aired people has never been easier. Therefore, the use of language eventually became one of the best ways fo r deaf and speechimpaired people to communicate wit h the rest of the world. The use of language facilitates communication between disabled and healthy people a nd provides direct communication. The purpose of this article is to convert sign language into text to facilitat e communication between deaf and hearing people.*
Although this job has significant social impact, the complex hand movements and movements make it incredibly difficult. A user should be able to record the hand sign in this work and the system will anticipate and show the hand gesture's title or text output. This system goes through a number of processing steps, some of which involve various computer vision techniques.
Image pixels with values A and Z, none, deletion and s pace are the extracted features. In this project, we use a convolutional neural network (CNN) to classify symp toms and train the network on training images. When we think about how to process images and make predi ctions, we think of convolutional neural networks (CN N).

Keywords-Convolutional Neural Network (CNN), Deep Learning, Sign Language, Model and Gesture(Keywords).

1. INTRODUCTION

When considering sign language, very few people are able to help the hearing or visual impaired understand what is being communicated. In addition, despite popular belief, sign language is not a universal language. People communicate with the individual using language that is simple for him or her to understand and retain in their minds.There is no denying the importance of speaking to someone in their own language.

Since the beginning of human civilization, language has played a significant role in human interaction. Humans use language as a means of expressing their feelings and thoughts in order to comprehend the cues of the outside world. Nothing in this world has meaning without language. People take language for granted because we are so immersed in our daily routines that we fail to recognize its significance. It is extremely sad that there are people with communication impairments in our society who are frequently ignored and left behind.

1.1 PROBLEM DEFINITION

Different regions use different sign languages. Additionally, sign language differs depending on the language used in any given country. Signs for American Sign, for instance the signs of Brazilian Sign Language are distinct from language. Simply put, because every language in the world is spoken differently, every language also has a unique sign language. This makes it even more difficult for the community of people with disabilities and the community of normal people to communicate. Written communication is an alternative method of communication because verbal communication is time-consuming, impersonal, and impractical in an emergency. Additionally, compared to face-to-face communication, this alternative method of communication is difficult and slow. If we consider an emergency situation, then it is important and necessary to communicate quickly. Therefore, written communication in this situation is not well-structured and systematic.

Riya Sunil Kumar Patel et al (2021), The concept of Language Recognition (SLR) involves developing deep learning models and computer vision that can rec ognize and understand descriptions in images. The aim

is to create a system that can transform sign language into a compatible alphabet to facilitate communication between deaf or hardofhearing people and people who cannot speak. Monocular camera competition is difficu lt as background noise, occlusions, and other environm ental factors can affect motion recognition. Therefore, building accurate and reliable SLR systems requires tr aining these models using deep learning (e.g., convolut ional neural networks) and significant behavioral data. The main aim of SLR is to promote the accessibility an d participation of language users as a means of commu nication and enable them to communicate effectively with the wider community.

2. LITERATURE SURVEY

In this study, RaziehRastgooet.al (2021) examines the deep learning model for visual language recognition that has been proposed in the last five years. Although all models of the model show that symbol information has better recognition accuracy, there are still some problems that need to be solved. Convolutional neural networks (CNN) and deep Boltzmann machines (RBM) are the most advanced deep learning models for computer problems (Wu, 2019). We introduce classification to improve the information structure of discrete and continuous words. We also review the applications, literature, hybrid models, challenges, and future directions of the topic.

In this article, **NecatiCihanCamgöz et al. (2022)** improved the initial translation work and presented improved translation through markerbased annotation that can recognize multiple markers. In fact, luster tokenization is required for cuttingedge translation work. At the same time, we offer a unique Transformerbase architecture that learns sign language recognition and translation at the end of the course. This is done by combining authentication and translation issues into a unified system that uses loss of physical link (CTC). This collaborative strategy simultaneously solves the realtime learning problem and improves performance without the need for real-Timedata.

In this paper, **Yanqiu Liao et al. (2019)** proposed a more effective and efficient video-based language recognition method. This paper will introduce a new development method called BLSTM3D Residual Network (B3D ResNet) to replace the old method based on the video system. The new B3D ResNet model aims to reveal spatiotemporal features and describe the nature of individual movements in dynamics. Language skills. To define deep learning tools such as Faster RCNN, a B3D ResNet model is first used to detect the hand and separate its activity from the background. The B3D ResNet model is then used to capture and analyze the features of the input video sequence. Sequential distribution of video ideas leads to recognition of gestures and mastery of good sign language.

In this article, **Mohammad Saad Amin et al. (2022)** proposed gesture recognition in a sensor-based model using smart gloves and different sensors. These sensors include a capture and distance sensor, an accelerometer that measures the angle and degree of movement, and a tilt sensor that measures body curvature. At this stage, language is represented as numbers with unique values created using special characters. Therefore, as a model based on the image sensor, there is no need for preprocessing.

In this article**, P.K. Athira et al. (2019)** suggest that sign language interpreting systems are becoming increasingly important to reduce the social isolation of India's large deaf and hard of hearing population. It can recognize finger-spelled words in Indian Sign Language (ISL), static and dynamic single-handed gestures, and even static two-handed gestures from live video. Using Zernike moments for keyframe extraction significantly slows down the computation. In addition, he suggests better techniques for eliminating assonance when writing the alphabet. Features are extracted from the video in real-time during pre-processing using skin tone segmentation. After the joint removal step, appropriate character vectors are extracted from the gesture sequence.

In this paper, **SamiyaKabirYoume et al. (2021)** suggested that CNN outperforms advanced machine learning techniques, especially in computer vision tasks. CNNs are mostly used for object recognition and classification using visual image analysis. This study focuses on the classification of static images of Bangladeshi personality types using these deep learning architectures. Neural network architecture for this model. The family of VGG networks performs best in BSL detection among other deep learning-based architectures.

In this article, **Riya Sunil Kumar Patel et al. (2021)** has a great social effect, but complex gestures and hand movements are very difficult. This work

requires that the user can record hand movements and the system can predict and display the name or output text of the hand movement. The system goes through many processing steps, some of which involve various computer vision techniques. A function from which image pixels with values between 0 and 1 are extracted When we think of how to manipulate images and how to classify them, we think of Convolutional Neural Networks (CNN).

In this paper, **Xiao, Q., et al. (2020)** suggested that sign language production (SLG) is necessary to enable reverse communication from hearing to deaf individuals. This method can be applied to other interactions between information sequences. This method yields an approximate fitting posterior distribution that results in skeletal sequences with a variety of human-recognizable styles. A two-way correlation test was performed on a large database of 500 CSLs. Due to the short execution time of the proposed algorithm, this algorithm achieves high detection accuracy in both real and synthetic data. The generated data also helps to improve the performance of the discriminator.

In this article, **Dongxuli et al. (2020)** reported that the purpose of visual feedback is to improve deaf people's ability to communicate with others. However, most language documents currently contain only a few words. Specifically, we use and compare two different models: (1) a full viewbased approach and (b) a 2D hu manbased approach. We also proposed a new posebase d temporal graph complexity network (PoseTGCN), w hich further improves the performance of the pose algo rithm. These networks simultaneously model the spatia l and temporal dependencies of human location traject ories. Our results show that the modelbased model and the modellike model perform equally well with 10% t o 62.63% accuracy at 2000 words/total, indicating goo d performance and difficulty of information.

In this article, **Junfu Pu et al (2019)** present a n optimization method for integration for monitoring weak continuous language recognition. The above two modules are optimized in many aspects. The encoder& decoder sequence learning network consists of two dec oders: LSTM decoder and CTC decoder. Soft Dynami c Time Warp (Soft DTW) correlation with optimal lear ning is used to jointly train the two algorithms. Soft Dynamic Time Warp (Soft DTW) alignment constraints along with maximum likelihood criterion training are used to jointly train both decoders. 3D-ResNet is fine-tuned as training labels with lossy classification, using convolutional paths that represent potential alignments between input video clips and character words. In the next iteration, the encoder and decoder sequence learning networks are optimized using the extracted improved features after fine-tuning.

In this article, **Cheok, M.J. et al. (2019)** suggested that appearance-based recognition and model-based recognition use different approaches for feature extraction. Appearance-based methods extract useful information from pixels, while model-based methods include volume and skeletal modeling. Although SURF is computationally more efficient than SIFT, its performance is not as consistent as SIFT. PCA and LDA are used in a combined function to improve accuracy. Recent research on gesture recognition uses hybrid feature extraction techniques such as DTW and HMM for dynamic gestures and SVM and ANN for static gestures. This model achieved 96.5% accuracy. Sign language recognition has a large vocabulary, but scalability issues are a challenge.

In this paper, **YuecongMin et al. (2021)** proposed to identify unsegmented characters from image streams based on vision, a Continuous Sign Language Recognition (CSLR) method was developed.One of the most important problems in CSL R training is overfitting, and previous studies have sho wn that reverse training can partially solve this proble m, but more time should be devoted to training.In this work, we return to the retraining methods used in rece nt CSLR studies and conclude that specific features m ust be adequately trained to solve the overfitting proble m. That's why we want to see compatibility (VAC) to i mprove the performance of relationship managers. In p articular, the proposed VAC has two loss schedules: o ne manages the prediction matching between the featur e extractor and the relationship model, and the other on ly focuses on vision vision.

In this article,**Han,X.,et.al(2022)**Move into in dependent places and difficult times. Through the integ ration of push and selfmonitoring, the lightweight spati otemporal channel monitoring module is designed to h elp the network focus on important information based on space, time and channel length. This model include s two submodels of Channels called Temporal Attentio n and Spatiotemporal Attention. Deploying this modul e in R (2 + 1)D demonstrates the effectiveness of the p repared method by producing results comparable to or

better than state-of-the-art methods on the CLS-500, Jester, and EgoGesture datasets.

This article **Nguyen, M. Al did it. (2021),** this article reviews deep learning techniques for language recognition. In this paper, we propose a capsular network (CapsNet) that promises to achieve this goal. We also proposed an optional kernel network (SKNet) with a listening algorithm to extract spatial information. Since sign language is an important part of communication, the difficulty of realtime recognition of sign language in digital video has become a new problem in the field. The benefits of this study are as follows. (1) According to specific data, the overall recognition accuracy of CapsNet is as high as 98.72%. (2) SKNet uses tracking system to achieve the highest recognition rate reaching 98.88%.

3. METHODOLOGY AND IMPLEMETATION

3.1 METHODOLOGY

Methodology is the theoretical analysis of the methods used in some researches. It differs from a method in that the purpose of a method is not to provide a solution. Alternatively, the method provides a theoretical basis for understanding what factors, processes, or best practices can be applied to a specific situation, such as calculating specific outcomes.

It also means:

1. "Analysis of methods, rules and theories used in this discipline";
2. "The study of methods used or employed in the field";
3. "The study or explanation of a process."

Methodology is the scientific method that explains how research is conducted and describes the methods used in the study. In a sense, this described process defines the method or manner in which the data is collected, or perhaps the factors to be calculated. The methodology does not describe specific methods, but much attention is paid to its nature and the specific method or methods that pursue the goals.

3.1.1 SIGN LANGUAGE RECOGNITION SYSTEM

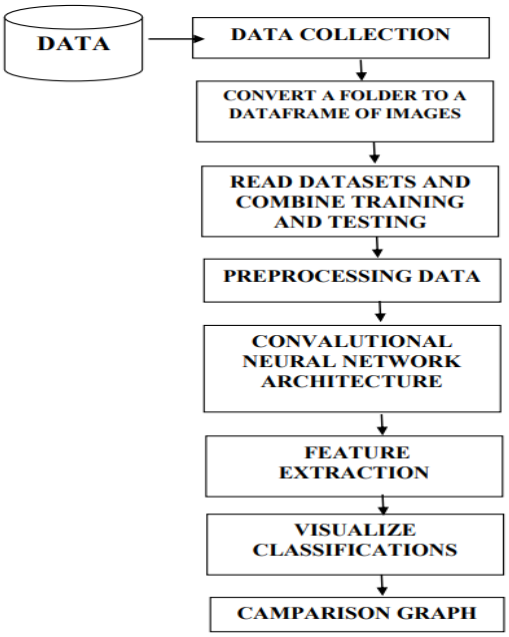

Figure 1: Sign language recognition system

3.2 DATA COLLECTION

The dataset contains images of hand gestures representing letters of the American Sign Language alphabet. This dataset is collected from kaggle website and the dataset provided in the link **"https://www.kaggle.com/code/aminesnoussi/american-sign-languagerecognition/input".** This dataset was created to help develop machine learning models for recognizing and interpreting these hand gestures.

The dataset consists of two folders, one for training and one for testing, each containing a set of images in JPG format. The training folder contains 365 images, while the testing folder contains 95 images. Each image is 200 x 200 pixels in size and is labelled with the corresponding letter of the alphabet that the hand gesture represents.

There are a total of 29 classes in the dataset, corresponding to the 26 letters of the alphabet and three additional classes for space, delete, and nothing. The "space" class represents a space between words, the "delete" class represents the action of deleting a character, and the "nothing" class represents when no hand gesture is present.

3.3 CONVERT FOLDER INTO DATAFRAME OF IMAGES

Importing various libraries related to computer vision,**cv2 and numpy**are used for computer vision and numerical computing respectively. **pandas and matplotlib** are used for data manipulation and visualization respectively. **ImageDataGenerator,**

MobileNet, and Dense are specific classes and functions within the TensorFlowKeras library that are useful for deep learning tasks involving image data.

The get_paths_labels function takes a path argument representing the directory path containing the image files to be processed, and an optional allowed extension argument that specifies the file extension of the image files to include (default is "jpg"). The function creates a directory object from the path argument using the Path class and then uses the glob method to find all files in the directory and its subdirectories with the specified file extension.The function then creates a Pandas Series object from the file paths and sets its name to "path". It also creates another Pandas Series object from the file path strings by extracting the second-to-last folder name as the label for each image, and sets its name to "label".

The function concatenates the two Series objects along the columns axis to create a DataFrame with two columns: "path" and "label". The function shuffles the rows of the DataFrame randomly using the sample method with frac=1 (which means to use all rows) and resets the index of the shuffled DataFrame. The function returns the shuffled DataFrame. The resulting DataFrame can be used as input for machine learning models that require input data in the form of file paths and their corresponding labels. This function can be useful for organizing and preprocessing large datasets of images for deep learning tasks.

3.4 READING DATASETS AND COMBINE TRAINING AND TESTING DATASETS

The Visualize_samples function takes an ImageDataGenerator object as input, randomly selects a set of image samples and their labels, and visualizes them in a grid using matplotlib. It uses the get_image function to read and convert the images to RGB format, and sets the title of each image to its corresponding label. The function also allows the user to specify the grid size and figure size of the plot. This function can be useful for quickly checking the quality and diversity of the data generated by an ImageDataGenerator.

The get_paths_labels function is used to read two image dataset and creates two separate Pandas DataFrames containing the file paths and labels of the images in each dataset. The dataframes are then concatenated into a single dataframe containing all images using the pd.concat function. Data used to train machine learning models.

```
Classes counts
A                    10
B                    19
C                    10
D                    10
E                    10
F                    13
G                    10
H                    10
I                    10
J                    10
K                    13
L                    14
M                    11
N                    13
O                    14
P                    13
Q                    10
R                    15
S                    13
T                    15
U                    15
V                    15
W                    15
X                    15
Y                    15
Z                    10
asl_alphabet_test    95
del                  10
nothing              13
space                15
Name: label, dtype: int64

Dataset size = 461 samples
```

Figure 2: Read Datasets And Counts Classes

The counts of each unique value in the second column (the class column) of the dataset and sorts them in ascending order. The resulting series is printed using the 'head' method to display all counts.

3.5 PREPROCESSING DATA

The process function processes a single image by converting its color channels from BGR to RGB, resizing it to 224x224 pixels, applying the MobileNetpreprocessing function, and adding an extra dimension to it for batch size. theKerasImageDataGenerator is used to perform data augmentation techniques such as horizontal flipping, brightness adjustment, and zooming on the input images. It also splits the dataset into training and validation subsets using a specified validation split percentage.

The flow_from_dataframe method is then used to create generator objects for both the training and validation data, with options to specify the target image size, color mode, batch size, and class mode. It takes the dataframe of image paths and corresponding labels as input, as well as the column names for the path and label data. Finally, we use a subset parameter to distinguish between training and validation sets.

Figure 3: Visualizing the Sample of Labelled Dataset

The above diagram shows the sample of labelled dataset of alphabet contains A to Z, other three classes delete, space and nothing from the training dataset. The sample images are displayed randomly based on the 4*4 matrix format.

3.6 CONVALUTIONAL NEURAL NETWORK ARCHITECTURE

The convolutional neural network architecture uses the MobileNet model as its foundation. MobileNet models are used as pre-trained models and imported from Keras application modules. The input geometry is set to (224, 224, 3). This has to do with image size and color. The "include_top" parameter is set to False. This means that the final sample set used for classification is not included. The "weight" parameter is set to "imagenet". This means that the weighted model is first learned from the Image Net dataset.

The "PretrainedModel" object is set to no training. This means that the weight of this layer does not change during operation. A dense layer with 128 neurons and ReLU activation is added to the output of the pre trained model, followed by another dense layer with 128 neurons and ReLU activation. Finally, a thick layer consisting of 29 neurons and softmax activation is added as the output layer. This corresponds to the number of groups in the file. The training value determines the size of the optimizer during training. Higher education leads to faster integration, but it can also lead to people becoming better at solving problems.

3.7 FEATURE EXTRACTION

Feature extraction is a technique used in machine learning and computer vision that involves extracting important features from raw data that can be used to improve the performance of a model. In image recognition, for instance, a feature is a measurable property of an image such as edges, corners, or textures. Feature extraction involves detecting and describing these features in a way that can be used to represent an image as a set of numerical values.

This allows the model to automatically extract high-level features from the input images without having to manually engineer them. The MobileNet model is initialized with pre-trained weights, which allows it to recognize a wide range of features such as shapes, textures, and colors.

The MobileNet model is then modified by removing the top layer and adding a few dense layers to enable it to perform the specific task at hand - classifying images into 29 different categories. Finally, the last layer is set to output a probability distribution over 29 classes using a softmax activation function.

3.8 VISUALIZE CLASSIFICATIONS

The visualize_classifications function takes a trained model, a data generator, and other optional arguments as inputs. It selects a random subset of images from the data generator, processes them using the process function, makes predictions using the provided model, and visualizes the predictions alongside the true labels. The function can be useful for assessing the performance of a model and gaining insight into which samples are misclassified.

3.9 COMPARISON

The graph is used to visualize the comparison between training and validation accuracy, loss, and mean squared error (MSE) of a neural network model. The history object, which contains the training metrics, is used to extract the values for accuracy, validation accuracy, loss, validation loss, MSE, and validation MSE. These values are plotted using Matplotlib based on the number of periods.

The first graph shows the training and validation accuracies across epochs, with the maximum validation accuracy highlighted by a green dot. Finally, the third MSE plot shows the training and

validation, with the lowest MSE validation highlighted with a green dot.

These visualizations help you understand the performance of your model over time and identify issues that need to be addressed. By observing trends in accuracy, loss, and MSE, you can decide how to modify your model to improve performance.

4. RESULTS AND DISCUSSION

4.1 RECORDING AND ANALYZING MODEL METRICS DURING TRAINING

The code metrics = pd.DataFrame(history.history) creates a DataFrame called metrics that contains the recorded metrics from the model training process. This DataFrame organizes the metrics, such as loss and accuracy, into columns, allowing for easy analysis and visualization of the model's performance over each epoch.

The model metrics are

	loss	accuracy	mean_squared_error	val_loss	val_accuracy	val_mean_squared_error
0	3.536756	0.116531	0.032667	3.456776	0.021739	0.032671
1	2.577602	0.281843	0.029047	2.446699	0.195652	0.029390
2	1.761375	0.436314	0.023681	1.652135	0.521739	0.021688
3	1.313232	0.558266	0.020198	1.062658	0.586957	0.016484
4	0.924073	0.615176	0.016576	0.576540	0.793478	0.010280
5	0.782112	0.620596	0.015434	0.387989	0.945652	0.005594
6	0.751313	0.710027	0.014353	0.592742	0.750000	0.011133
7	0.740915	0.666667	0.014639	0.652531	0.641304	0.013747
8	0.725998	0.696477	0.014252	0.461309	0.836957	0.007890
9	0.719974	0.682927	0.014698	0.366876	0.923913	0.006273

Figure 4: Recording and Analyzing Model Metrics During Training

4.2 PREDICTED SIGN LANGUAGE IMAGES

In this part, randomly selecting row_col_len^2 indexes from the data generator labels. It retrieves the classes, labels, and filepaths corresponding to the randomly selected indexes. It retrieves the images corresponding to the filepaths and applies some processing to them. It makes predictions using the provided model on the processed images and converts the predicted class indices to their corresponding class labels.

It creates a figure with row_col_len rows and columns to display the images and their classification labels. It iterates over each row and column in the figure and plots an image along with its true and predicted labels. It helps in understanding how well the model is performing on different classes.

Figure 5: Predicted Sign Language Images

This image represents the randomly predicted sign language images with true and predicted alphabet of the image.

4.3 CAMPARISON

4.3.1 Training and Validation Accuracy

Accuracy is a measure used to evaluate the eff ectiveness of classification models. It measures the per centage of samples correctly classified for the entire sa mple. To calculate accuracy, divide the number of corr ectly predicted samples by the total sample and multipl y the result by 100.However, accuracy is not always the best measure, especially when the dataset is unbalanced or the cost of misclassification varies between classes, so it is important to consider the context and specific requirements of the problem. **Accuracy = (Number of correctly predicted samples) / (Total number of samples)**

To express accuracy as a percentage, the result is often multiplied by 100.

```
Accuracy: 90.22%
```

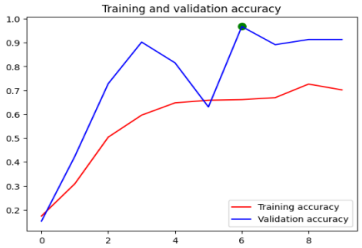

Figure 6: Training and Validation Accuracy

Figure 6: Training and Validation Accuracy shows that both the training accuracy and validation

accuracy start from a low value and gradually improve with epochs. However, towards the end of the epochs, the validation accuracy seems to plateau and does not show much improvement, while the training accuracy continues to improve. The green dots in both graphs indicate the periods with the highest credit accuracy and the lowest credit loss.

4.3.2 Training and Validation Losses

Training Loss shows how well the model fits the training data and is minimized by adjusting the model parameters during training.Validation loss will help you evaluate the generalization performance of your model and detect overfitting or underfitting. Minimizing validation losses is important to ensure good performance with new and unknown data. Batch mutual entropy:

Training Loss = -(1/N) * Σ(y_actual * log(y_pred))
Validation Loss = -(1/N_val) * Σ(y_actual * log(y_pred))

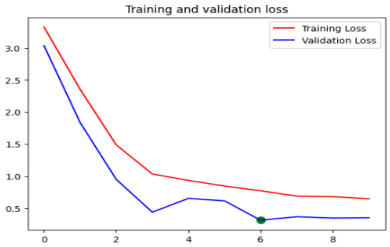

Figure 7: Training and Validation Loss

Figure 7: Training and validation losses shows that the training and validation losses both start from high values and decrease with periods. . However, at the end of the period, the credit losses start to increase while the training losses continue to decrease. The green dots in both charts show the periods of highest credit accuracy and lowest credit losses.

4.3.3 Training and Validation Mean Squared Errors (MSE)

The mean square error (MSE) calculates the differ ence between the predicted and actual values for each model, sums them, and averages them. By squaring the differences, larger errors are amplified, giving more weight to outliers. The resulting value represents the average squared error, with a lower MSE indicating a better fit to the data.

MSE is widely used because it provides a continuous and differentiable measure of error. It is particularly useful when the target variable has a wide range or contains outliers. However, MSE can be sensitive to outliers due to the squaring operation, and it doesn't provide direct interpretability of the error in the original scale of the data.

The formula for MSE is as follows:

MSE = (1/N) * Σ(y_pred - y_actual)^2

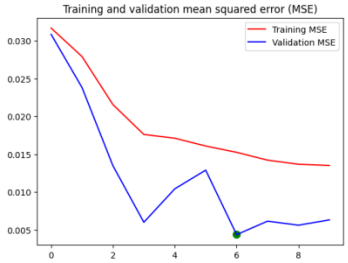

Figure 8: Training and Validation Mean Squared Errors (MSE)

The diagram shows the training and validation mean squared errors (MSE) over the epochs. The green dot indicates the lowest validation MSE achieved during the training.

4.4 CONFUSION MATRIX VISUALIZATION FOR MODEL EVALUATION

The given code calculates and visualizes the confusion matrix, By examining the confusion matrix, you can evaluate the accuracy of your model and identify misclassification patterns and areas where your model needs improvement.

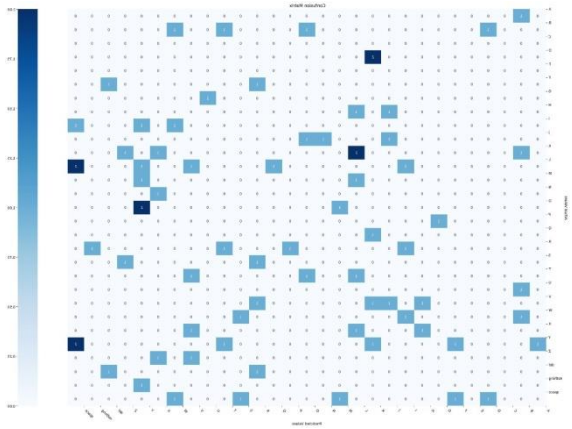

Figure 9: Confusion Matrix Visualization for Model Evaluation

The diagram shows the confusion matrix for the predicted labels with the true labels of the validation data.

5. CONCLUSION AND FUTURE SCOPE

Sign language recognition technology has come a long way and could revolutionize the way people with hearing loss communicate. Classification of the American Sign Language (ASL) alphabet and implementation of a neural network model for the

three classes of omission, gap, and no action using the MobileNet architecture. The model is trained on the ASL alphabet dataset and evaluated on the validation set.

The MobileNet model is a convolutional neural network (CNN) architecture widely used for image classification tasks. MobileNet models are enhanced with additional dense layers to learn more complex patterns and improve classification accuracy. During the accuracy training process, the model is classified using the Adam optimizer and the mutual entropy loss function. Training and validation and training and validation losses are monitored and visualized using graphs.

After training, the model achieved a prediction accuracy of 90.22% on the validation set. This accuracy value indicates the model's ability to correctly classify gestures in the ASL alphabet. It is important to note that the accuracy achieved may vary depending on factors such as dataset quality, model architecture, and training duration. Finally, we present a comprehensive pipeline for training and evaluating a MobileNet-based model for alphabetic ASL gesture classification and demonstrate the effectiveness of deep learning in sign language gesture recognition.

Future work can be expected to train this model on datasets containing continuous signs of sign language. In this way, we can identify the complete set of symbols that gives us the output of the sentence. We can also expect the system to recognize dynamic symbols to provide output with meaningful sentences for specific symbols. The system can be further expanded by adding a voice recognition feature. It helps strengthen the system by converting speech into sign language.

6. REFFERENCES

1. Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign language recognition: A deep survey. Expert Systems with Applications, 164, 113794.
2. Amin, M. S., Rizvi, S. T. H., & Hossain, M. M. (2022). A Comparative Review on Applications of Different Sensors for Sign Language Recognition. Journal of Imaging, 8(4), 98.
3. Y. Liao, P. Xiong, W. Min, W. Min and J. Lu, "Dynamic Sign Language Recognition Based on Video Sequence With BLSTM-3D Residual Networks," in IEEE Access, vol. 7, pp. 38044-38054, 2019, doi: 10.1109/ACCESS.2019.2904749.
4. Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2020). Sign language transformers: Joint end-to-end sign language recognition and translation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10023-10033).
5. Athira, P. K., Sruthi, C. J., & Lijiya, A. (2022). A signer independent sign language recognition with co-articulation elimination from live videos: an Indian scenario. Journal of King Saud University-Computer and Information Sciences, 34(3), 771-781.
6. S. K. Youme, T. A. Chowdhury, H. Ahamed, M. S. Abid, L. Chowdhury and N. Mohammed, "Generalization of Bangla Sign Language Recognition Using Angular Loss Functions," in IEEE Access, vol. 9, pp. 165351-165365, 2021, doi: 10.1109/ACCESS.2021.3134903.
7. NETWORK, U. C. N. (2021). SIGN LANGUAGE RECOGNITION SYSTEM USING CONVOLUTIONAL NEURAL NETWORK (Doctoral dissertation, CALIFORNIA STATE UNIVERSITY SAN MARCOS).
8. Xiao, Q., Qin, M., & Yin, Y. (2020). Skeleton-based Chinese sign language recognition and generation for bidirectional communication between deaf and hearing people. Neural networks, 125, 41-55.
9. Li, D., Rodriguez, C., Yu, X., & Li, H. (2020). Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In Proceedings of the IEEE/CVF winter conference on applications of computer vision (pp. 1459-1469).
10. Pu, J., Zhou, W., & Li, H. (2019). Iterative alignment network for continuous sign language recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4165-4174).
11. Cheok, M. J., Omar, Z., & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. International Journal of Machine Learning and Cybernetics, 10, 131-153
12. Min, Y., Hao, A., Chai, X., & Chen, X. (2021). Visual alignment constraint for continuous sign language recognition. In Proceedings of the

IEEE/CVF International Conference on Computer Vision (pp. 11542-11551)..

13. Gurbuz, S. Z., Gurbuz, A. C., Malaia, E. A., Griffin, D. J., Crawford, C. S., Rahman, M. M., ... & Mdrafi, R. (2020). American sign language recognition using rf sensing. IEEE Sensors Journal, 21(3), 3763-3775.
14. Han, X., Lu, F., Yin, J., Tian, G., & Liu, J. (2022). Sign language recognition based on R (2+ 1) D with spatial–temporal–channel attention. IEEE Transactions on Human-Machine Systems, 52(4), 687-698.
15. Lu, J., Nguyen, M., & Yan, W. Q. (2021). Sign language recognition from digital videos using deep learning methods. In Geometry and Vision: First International *Symposium, ISGV 2021, Auckland, New Zealand, January 28-29, 2021, Revised Selected Papers 1* (pp. 108-118). Springer International Publishing.