applied sciences

MDPI

Article

Fusion of Attention-Based Convolution Neural Network and HOG Features for Static Sign Language Recognition

Diksha Kumari * and Radhey Shyam Anand

Department of Electrical Engineering, IIT Roorkee, Roorkee 247667, Uttarakhand, India; r.anand@ee.iitr.ac.in
* Correspondence: dkumari@ee.iitr.ac.in

Abstract: The deaf and hearing-impaired community expresses their emotions, communicates with society, and enhances the interaction between humans and computers using sign language gestures. This work presents a strategy for efficient feature extraction that uses a combination of two different methods that are the convolutional block attention module (CBAM)-based convolutional neural network (CNN) and standard handcrafted histogram of oriented gradients (HOG) feature descriptor. The proposed framework aims to enhance accuracy by extracting meaningful features and resolving issues like rotation, similar hand orientation, etc. The HOG feature extraction technique provides a compact feature representation that signifies meaningful information about sign gestures. The CBAM attention module is incorporated into the structure of CNN to enhance feature learning using spatial and channel attention mechanisms. Then, the final feature vector is formed by concatenating these features. This feature vector is provided to the classification layers to predict static sign gestures. The proposed approach is validated on two publicly available static Massey American Sign Language (ASL) and Indian Sign Language (ISL) databases. The model's performance is evaluated using precision, recall, F1-score, and accuracy. Our proposed methodology achieved 99.22% and 99.79% accuracy for the ASL and ISL datasets. The acquired results signify the efficiency of the feature fusion and attention mechanism. Our network performed better in accuracy compared to the earlier studies.

Keywords: sign language recognition; feature extraction; deep learning; HOG features; gestures

Citation: Kumari, D.; Anand, R.S. Fusion of Attention-Based Convolution Neural Network and HOG Features for Static Sign Language Recognition. *Appl. Sci.* **2023**, *13*, 11993. https://doi.org/10.3390/app132111993

Academic Editors: Xi-Le Zhao and Junmin Liu

Received: 28 September 2023
Revised: 27 October 2023
Accepted: 29 October 2023
Published: 3 November 2023

1. Introduction

Various software-based applications have been implanted with the advancement in computer technology, giving multiple ways of interacting with humans through computers. Nowadays, different interfaces are text or speech-based. People with speech and hearing impairments communicate and express their feelings with the help of sign language gestures. It is composed of manual and non-manual features. Manual features represent hand shape, position, palm orientation, and trajectory. Non-manual features comprise movement of the body, shoulders, and facial expressions. So, the deaf community uses gestures to convey meaningful information that is only understandable by sign language experts or the native people. Hence, there is a need to bridge this interaction gap by implementing an intelligent computer and efficient Sign Language Recognition (SLR) system. This system is required to detect the acquired data of sign gestures and provide meaningful information extracted to the machine to develop real-world applications. There are two standard methods in SLR: device-based approach and vision-based methods. The device-based methods involve sensors and data gloves, and the subjects must wear heavy devices that make them uncomfortable performing the signs. These methods enhance accuracy but could not be more efficient in real-world applications [1]. However, the vision-based methods involve non-invasive techniques like high-definition cameras like stereo, monocular, and non-invasive sensors. Vision-based methods involve various image processing techniques to enrich the data visualization for good classification results. Most earlier studies used

classical machine learning algorithms that perform feature extraction and classification as two individual steps. However, with the growth of intelligent deep neural networks and the advancement in artificial intelligence, significant improvement has been made in image and video processing. The convolutional layers in the convolutional neural network (CNN) extract many low-level features like edges, corners, curves, etc., and high-level features like object detection from images. There are various classical feature descriptors like scale-invariant feature transform (SIFT), local binary patterns (LBP), and speeded-up robust features (SURF) for the extraction of features [2–4]. However, most of these features are subject to scaling, rotation, and brightness variations. So, to solve these issues, a system is needed that uses both traditional handcrafted feature learning methods and deep learning-based models. Nowadays, most researchers use attention-based networks to enrich the model's efficiency and minimize the computational cost. The main objective of these networks is to focus on relevant features and remove redundant features to classify the gestures [5] accurately. In our proposed methodology, the sign language gesture images are pre-processed and fed to a feature extraction module that extracts specific features from two strategies: histogram of oriented gradients (HOG) and convolutional block attention module (CBAM) attention-based CNN. Finally, the merged features produce a feature vector. Combining two models aims to implement a highly robust architecture to various uncertainties and achieve reasonable accuracy. The summary of the significant contributions of this research work is as follows:

(1) We implemented an efficient feature extraction technique using the fusion of HOG feature descriptor and attention-based CNN for recognizing hand sign gestures.
(2) The CBAM module is embedded in CNN to enhance the learning of features at various scales so that the model can handle different alterations in the location and shape of hand gestures and focus on effective features.
(3) We tested the efficiency of the approach among two datasets using three evaluation metrics, namely precision, recall, and F1-score.

The remaining work of this article is organized into various sections: Section 2 describes the literature survey and the related studies for the prediction of sign language gestures. Then, in Section 3, the architecture of our designed framework is described. Section 4 discusses the overall analysis of the results. At last, we concluded the paper in the Section 5.

2. Related Work

Several researchers have proposed various methodologies to implement a system to recognize sign language gestures. We have categorized the literature survey into traditional, deep learning, and attention-based SLR methods.

Traditional SLR methods: Marin et al. extracted features based on the angle and location of the fingertips. Support vector machine (SVM) model is used to predict classes and achieved 89.70% accuracy [6].

The authors in [7] merged HOG and LBP feature descriptors for hand gesture recognition with a recognition accuracy of 92% on the National University of Singapore (NUS) dataset. In [8], the authors implemented a machine learning-based framework that applied segmentation and used discrete wavelet transform (DWT) with SURF for feature extraction.

Multiclass SVM with the bag of words technique is used to classify hand gestures with an overall accuracy of 96.5%.

The authors in [9] employed HOG features with two classifiers for hand gesture recognition with an accuracy of 96% and 98% with the K-nearest neighbor (KNN) and SVM, respectively.

A multimodal feature vector is obtained using distance transform, contour feature extraction, and K-curvature convex defects techniques for recognizing hand gesture RGB depth dataset [10]. The system acquired a recognition accuracy of 97.4%.

Deep learning based: In the paper [11], the authors used the VGG16 model to classify 36 static signs, including 26 alphabets and ten digits from the American Sign Language

(ASL) database. They fine-tuned the parameters using the transfer learning technique on the VGG16 model, which is already trained on ImageNet data and comprises more than a million images. With this methodology, they acquired an impressive accuracy of 96%. Sruthi et al. [12] proposed an Indian Sign Language (ISL) static alphabet recognition system. The face was detected using a face detection algorithm and segmented the hand region using color-based segmentation. The approach uses a deep learning algorithm to recognize 24 sign classes. The methodology obtained a validation accuracy of 98.64%.

The authors have presented a system by concatenating features from two CNN streams on the ASL alphabet dataset using various deep learning models for classification. The system achieved the best accuracy of 97.57% with ResNet50 as classifier [13].

A seven-layered CNN [14] for hand gesture recognition with data augmentation and skin segmentation has attained 96.5% and 96.57% accuracy on two benchmark datasets, respectively.

Attention-based: Nowadays, attention-based deep learning frameworks have shown tremendous growth in achieving more accurate recognition than other deep learning models. The features from the previous layers are fused into an attention block to extract knowledgeable features [15].

The authors in the paper [16] proposed a methodology that used a residual attention-based system that helped enhance feature strength. However, the residual attention module is not portable and computationally expensive.

Park Jongchan et al. [17] designed a framework applying channel and spatial attention with the bottleneck attention module (BAM). It focused on the refinement of the intermediate features of the CNN and provided better results than the other related frameworks.

Xiang Li et al. [18] developed a selective kernel networks (SkNet) model to analyze the attention process in the convolution layer. It assigned weights dynamically with different scaled convolutional kernels. This approach has shown the value of receptive fields in the recognition of images.

An attention-based network consisting of detail, subject, and channel attention modules for the enhancement of features was proposed and validated on two public datasets that achieved remarkable results compared to the baseline models [19].

The literature shows deep learning networks and traditional feature descriptors have achieved good gesture recognition results. However, most existing methods have focused only on the global image features. They have yet to consider the importance of local image features. Also, some deep learning methods are memory-intensive and computationally extensive.

We have proposed a framework using a feature extraction module involving feature fusion of classical HOG features with attention-based CNN features to recognize static sign gestures using global and spatial dependencies. The main advantage of using CBAM-based CNN with channel and spatial attention modules is focusing on potential hand gesture features.

3. Methodology

This section discusses the overall details of the datasets and the architecture of the implemented methodology with a detailed description of the significant steps to recognize static sign language gestures in the developed framework shown in Figure 1.

3.1. Dataset

The benchmark dataset, the Massey University dataset [20], consists of 2515 images of 5 individuals belonging to 36 sign classes. The images in the dataset have few changes in rotation, scale, and lighting captured from five different directions with variations in hand size with the camera. Figure 2 shows a sample of the sign gesture images taken from this image database. The static ISL dataset [21] comprises 36 sign gestures of digits (0–9) and alphabet (A–Z), with more than 1200 sample images in each class captured using a web camera. There is a total of 34,554 images in the database having a size of 250 × 250 with augmentation to introduce variability in the data. Figure 3 shows the sample images from the ISL dataset.

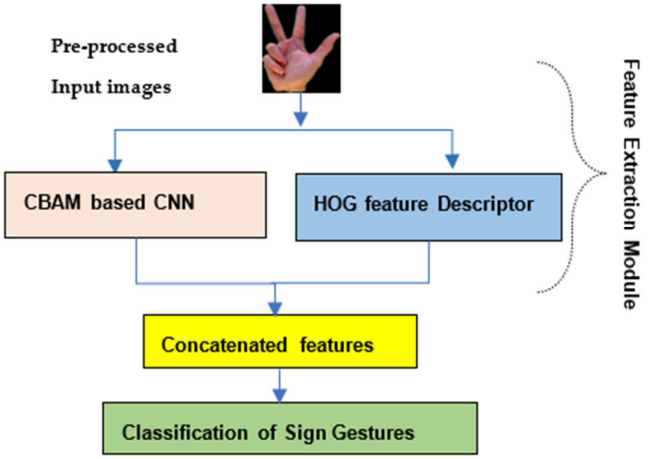

Figure 1. Proposed framework block diagram.

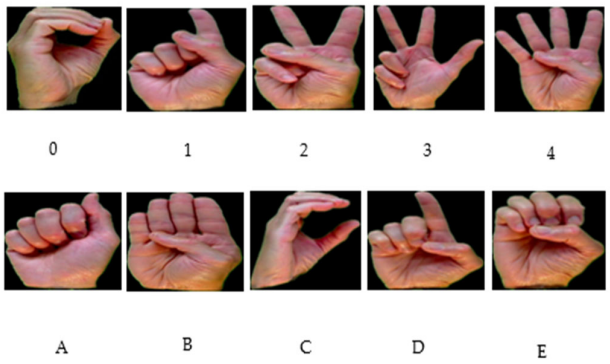

Figure 2. Sample of hand gestures of numerals (0–4) and alphabets (A–E) in Massey ASL dataset.

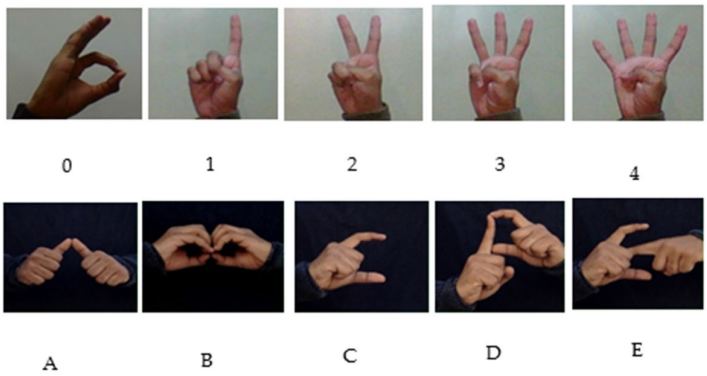

Figure 3. Sample of numerals (0–4) and alphabets (A–E) in ISL dataset for gesture recognition.

3.2. Pre-Processing

The images are first pre-processed after acquisition to prepare the images for feature extraction. The input image is scaled down to 128 × 128 to minimize the computational complexity [22]. All the input images are normalized by dividing the pixel intensities of each image by 255.

3.3. Feature Extraction Module

The process of learning meaningful features is quite challenging in image processing. Various features like edges, corners, blobs, etc., can be extracted from an image that gives a different representation and identification of an image to the computers for understanding and learning. Many features lack robustness when subjected to rotation, scaling, and

illumination variations. In the proposed structure, the handcrafted HOG-based feature vector and attention-based CNN features are used [23].

3.3.1. HOG Feature Descriptor

For image data, there is a whole branch of feature engineering that deals with the extraction of features from images. HOG computes pixel-wise gradients and orientations and plots them on a histogram. The main motive is to simplify the representation of the image by extracting the most useful information from the image and removing the redundant parts. In this way, it helps in minimizing the noise. This technique detects objects by calculating feature descriptors using horizontal and vertical gradients. The image splits into 8 × 8 cells, with each cell pixel having direction and magnitude that helps develop a compact representation that can be quickly learned by computer [24]. This method computes the gradient of each pixel in x (horizontal) and y (vertical) directions since the image is a discrete function. It is a vector having direction and magnitude given in Equations (1) and (2).

$$\theta = arctan\left(\frac{f_x}{f_y}\right) \tag{1}$$

$$mag = \sqrt{\left(f_x^2 + f_y^2\right)} \tag{2}$$

A 9-bin histogram that ranges from 0–180 degrees quantizes the orientation of the gradient with values in the range of (0–255). All histograms are merged to produce a feature vector H. The final HOG feature vector is given by Equation (3), which provides shape information of regions or points within an image [24].

$$F_{HOG} = \{f_{h1},\ f_{h2}, f_{h3}, \ldots..f_{hn}\} \tag{3}$$

We applied a Laplacian filter to sharpen the edges of the hand gestures and reduce noise. The intensity values of the images are rescaled to improve the image's contrast. This model extracted 8100 features using the HOG method from each of the input images of the dataset. Figure 4 depicts the input image and the extracted HOG image.

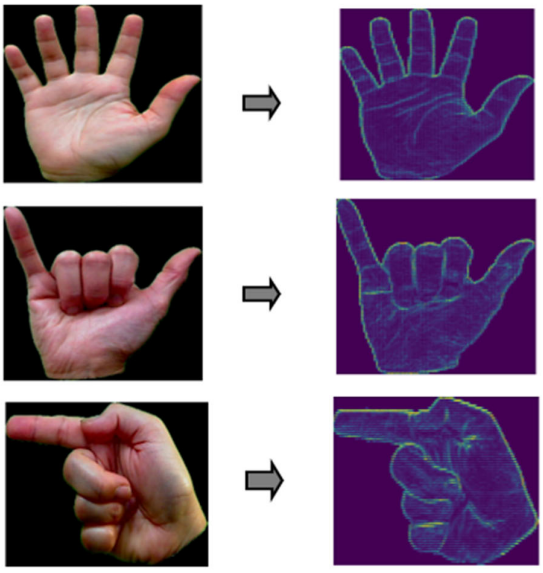

Figure 4. Processing of sign image and corresponding HOG output.

3.3.2. CBAM Attention-Based CNN Feature Extraction

In our work, we introduced a CBAM attention-based module within CNN so that the model can concentrate on particular features that carry essential information about gestures.

The architecture of the CBAM module used is shown in Figure 5. This module proposed by Woo et al. [25] aims to efficiently enhance feature extraction and representation. We can represent the intermediate feature map as $F \in \mathbb{R}^{C \times H \times B}$ where, $C \times H \times B$ is the input feature map resolution. In the channel attention block, the channel carrying meaningful information is decided based on inter-channel relationships. Two pooling functions, global average pooling (GAP) and max pooling, individually squeeze out the spatially extracted features from the intermediate feature vector. Then, these features are fed to a multilayer perceptron (MLP) and merged. Then, the sigmoid function activates the sigmoid feature map. The channel attention module gives a 1D attention map written as $M_c \in \mathbb{R}^{C \times 1 \times 1}$. Then, we will obtain a refined input feature F' by multiplying this attention map with the original output. Mathematically, the process of the channel attention module is shown in the following equations.

$$M_c(F) = f_{sigmoid}(MLP(GlobalAvgPool(F)) + MLP(MaxPool(F))) \tag{4}$$

$$F' = M_c(F) \otimes F \tag{5}$$

The spatial attention block concentrates on the inter-spatial relation of the input feature map and it is fed with this refined feature vector and two intermediate feature maps are generated using GAP and max pooling that are concatenated.

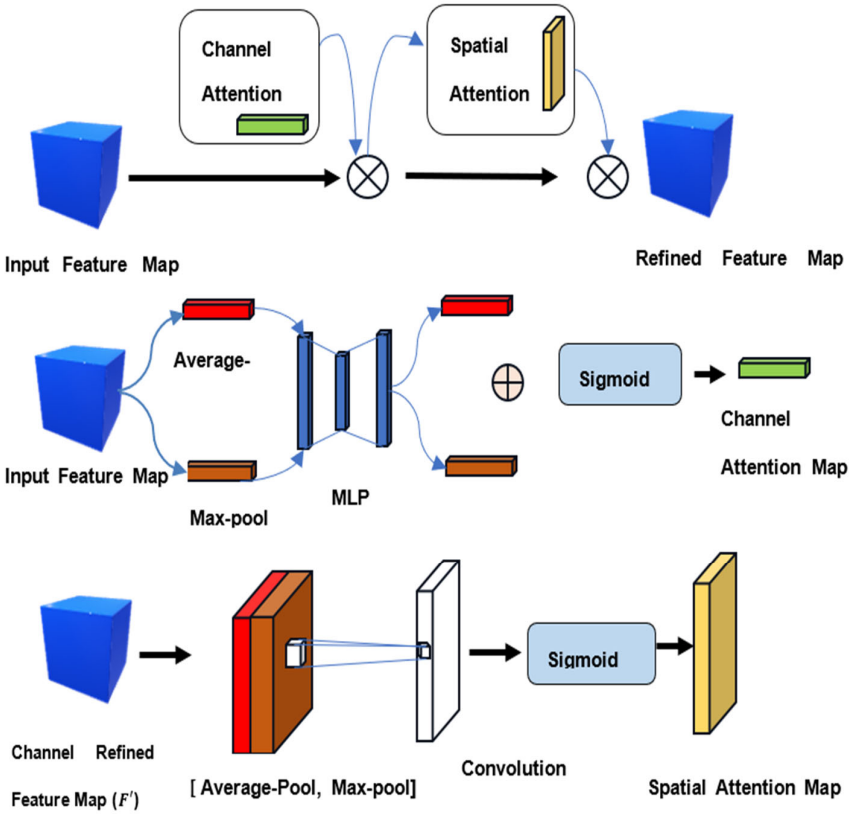

Figure 5. Architecture of CBAM mechanism.

The fused feature map is passed through a 7×7 kernel and sigmoid function to generate a spatial feature map with attention weight $M_S \in \mathbb{R}^{1 \times H \times w}$ and results in a more efficient feature representation by multiplying the spatial feature map with the refined input as given in Equations (6) and (7).

$$M_s(F) = f_{sigmoid}\left(Conv\left(\begin{bmatrix} GlobalAvgPool(F); \\ MaxPool(F) \end{bmatrix}\right)\right) \tag{6}$$

$$F'' = M_s(F') \otimes F' \tag{7}$$

where the symbol "⊗" represents elementwise multiplication, and F'' is the final refined feature map [26,27].

In this study, we employed a CBAM attention-based module that consists of three CNN convolution blocks to generate attention-based feature maps from the input dataset. Figure 6 demonstrates the flowchart of the CBAM-CNN model. The convolutional layer is the initial layer in each block, and each convolution layer generates features and gives a feature map. Then, we call the CBAM attention module to extract channel and spatial data. Each convolution block has batch normalization and max pooling layers. Also, we used equal padding and the rectified linear unit (ReLU) activation function within each block. We used 32, 64, and 128 filters in the CNN model's first, second, and third stages. Finally, the model results in a 1D feature vector after flattening the 3D feature map obtained after the last layer of max pooling. The flattened 1D vector that contains information on image-based features extracted from convolutional layers is expressed by Equation (8).

$$F_{CNN} = \{f_{CNN1}, f_{CNN2}, f_{CNN3}, \ldots\ldots f_{CNNm}\} \tag{8}$$

where m is the number of features. Then, the CNN features are fused with the HOG-based feature vector to obtain the training feature vector by Equation (9).

$$F_{concat} = F_{CNN} \oplus F_{HOG} \tag{9}$$

There are two fully connected dense layers in the network. The first layer, with 300 neurons, takes the combined feature vector as input and performs a linear transformation, and the second layer, with 150 neurons, further processes the previous layer's output. We have used the L2 regularization technique with hyperparameter λ = 0.0001 for each dense layer to minimize the network complexity and loss while training the model.

Then, a dropout layer is applied to the output with a 0.5 dropout value to prevent overfitting during the model's training. Finally, the model's output layer has 36 units and a Softmax activation function. It produces probabilities for each of the 36 possible classes. Algorithm 1 summarizes the overall working of the proposed network.

Algorithm 1: Proposed Algorithm for recognition of sign language gestures

Input: Image Dataset with size (128 × 128 × 3)
Output: Sign language Gestures
//HOG feature extraction F_{HOG}
Step 1: Convert each image from the training dataset into grayscale images.
Step 2: For each pixel of the image gradient is calculated (f_x, f_y).
Step 3: Divide input images into 8 × 8 cells, and the corresponding histogram of each cell is computed.
Step 4: Merge all histograms to obtain the HOG feature vector.
Output: HOG feature vector F_{HOG} of dimension 1 × 8100.
//Attention-based CNN feature extraction F_{CNN}.
Step 1: CNN layers are defined including convolution, normalization, CBAM, and max pooling.
Step 2: Flatten the last max pooling layer CNN output to 1-dimensional feature vector F_{CNN}.
Step 3: Concatenate the flattened CNN feature with HOG feature $F_{Concat} = [F_{CNN}, F_{HOG}]$.
Step 4: The fully connected layer takes the merged feature as input.
Step 5: The final output layer classifies 36 sign gestures with a Softmax activation function.

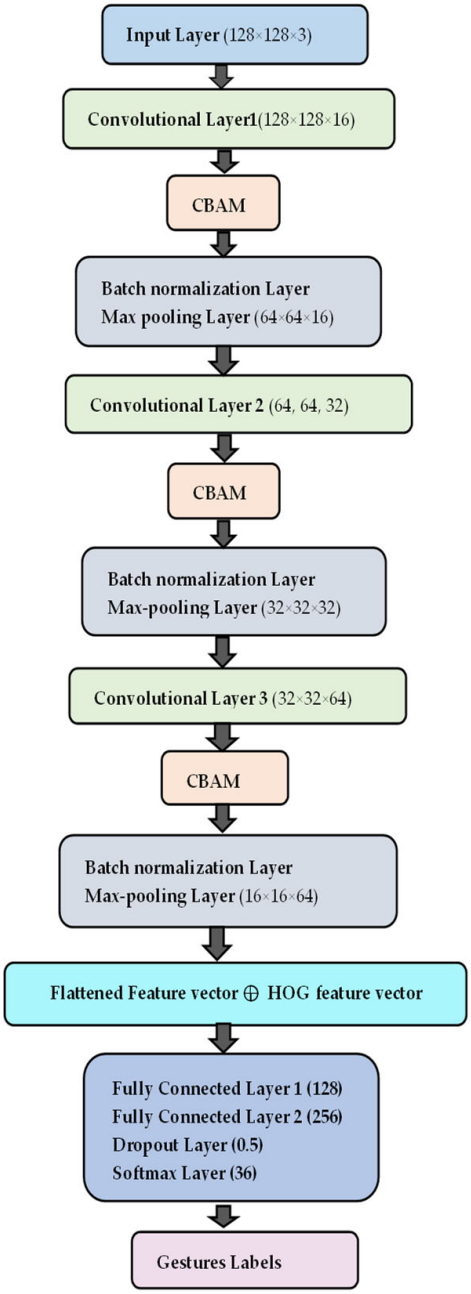

Figure 6. Flowchart of the proposed CBAM-CNN model.

4. Results and Discussion

We used the deep learning TensorFlow framework in the Python 3.7 programming platform with a Windows 10 Pro operating system, Intel(R) Core (TM) with i7 CPU@3.20 GHz, and Nvidia GTX 1060 with a 24G memory graphics card to implement the model architecture and perform the experiments on two datasets. We have used TensorFlow, OpenCV, Pandas, Numpy, and Matplotlib Python libraries for computational graphics, image processing, and mathematical processing of data and visualization of various parameters [5]. We used an Adam optimizer with Nesterov momentum called Nesterov-accelerated adaptive moment (Nadam) estimation [28] with an initial learning rate of 0.001 to improve high fluctuations in the loss and enhance the cost function's convergence process. We tuned the hyperparameters to achieve good training of the model.

The details of hyperparameters used while training the model to recognize 36 classes are shown in Table 1.

Table 1. Hyperparameter Details.

Parameters	Values
Input size	128×128
Initial Learning Rate	0.001
Optimizer	Nadam
Batch Size	32
Cost Function	Categorical Cross-entropy
Epoch	30
Beta 1	0.9
Beta 2	0.999
Epsilon	1×10^{-8}

We performed five-fold cross-validation to achieve unbiased results and avoid overfitting while training. This method divides the dataset into five equal folds, in which a fold is unseen data for testing. The remaining folds are for training. The number of samples for each class in the test set for each fold will vary due to randomness in the process of k-fold cross-validation. We assessed our proposed architecture's efficiency using accuracy, precision, recall, and F1-score for individual folds. Then, we average these metrics across all the folds to assess your model's performance. This averaging process helps smooth out the effects of sample size variation and gives a more reliable estimate of your model's performance.

This variation is due to the random partitioning of our dataset into folds. The precision is the ratio of the total count of precisely identified positive observations to the total number of values classified as positive. Recall is the fraction of accurate predictions to the overall correct classes. The F1-score is a widely used metric to measure a model's efficiency, also known as the balanced mean of the recall and precision metrics [29]. The mean accuracy of all folds is calculated for both datasets, as shown in Table 2. We computed the precision, recall, and F1-score using the following Equations (10)–(12).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11}$$

$$\text{F1} - \text{score} = 2* \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{12}$$

where "TP", "FP", and "FN" are true positives, false positives, and false negatives, respectively. Figures 7 and 8 represent the values of recall, precision, F1-score, and recognition accuracy of our proposed model on both datasets for fold two. Figure 9 shows the accuracy and loss graphs of both datasets for training and validation data.

Table 2. The average recognition rate for both datasets.

Folds	Dataset 1	Dataset 2
1	99.10	99.69
2	99.22	99.77
3	99.24	99.79
4	99.22	99.81
5	99.30	99.83
Average Accuracy (%)	99.22	99.77

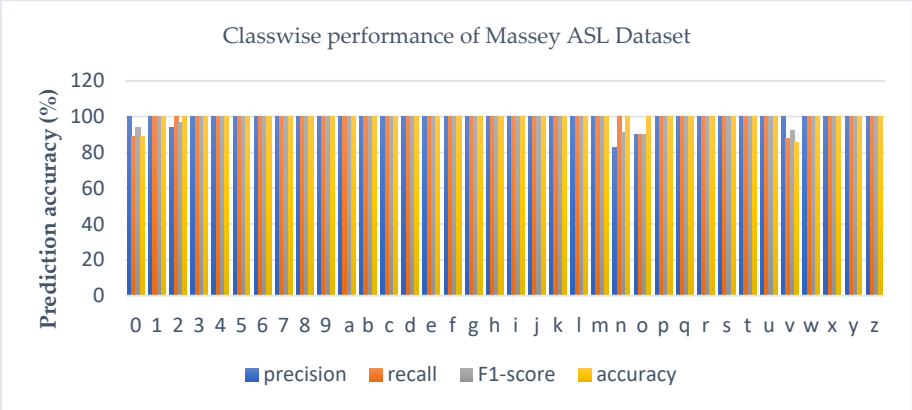

Figure 7. Chart of precision, recall, F1-score, and accuracy values for Massey ASL.

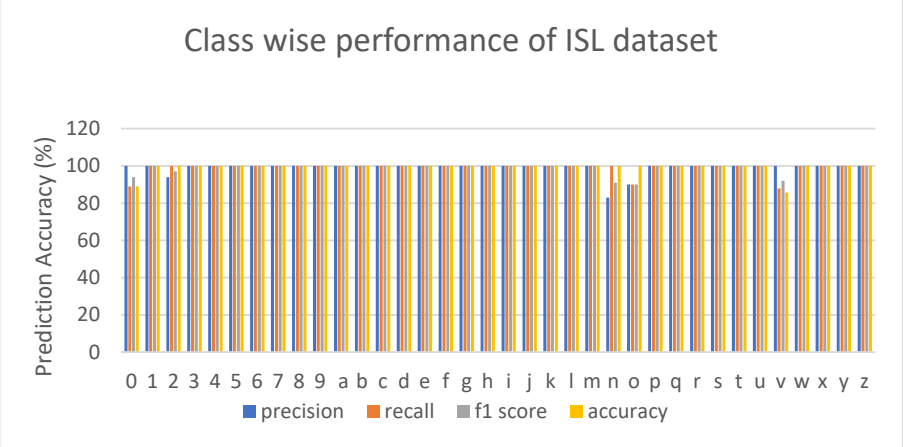

Figure 8. Chart of precision, recall, F1-score, and accuracy values for the ISL dataset.

It is observed from Figure 9 that the loss is reduced after each iteration and reaches a minimum value, and the accuracy metric is attaining the maximum value. In the ASL dataset, the accuracy and loss stabilized after the 25th epoch. In the same way, both metrics stabilize after the 10th epoch for the ISL database. We used categorical cross entropy as a cost function to analyze loss metrics for multiclass classification. Furthermore, we presented the confusion matrix used to predict correct sign classes and give misclassified samples. It provides accuracy of each sign gesture in terms of percentage. The resulting confusion matrix of both datasets is presented in Figures 10 and 11, respectively.

The confusion matrix signifies the actual and the predicted classes in the x and y axes, respectively. The diagonal values in the confusion matrices show the actual recall values of all the signs, and the consistent rows signify the confused classes. Some classes in the ASL dataset have similar gesture signs, such as "0" is misclassified with 10.5% as "O" due to similar hand gestures with little variations in the position of hands, and "V" is misclassified with 14.3% as "2". The reason behind this misclassification is the similarity in sign gestures with only a change in the position of the thumb in signs "2" and "V". As shown in the confusion matrix of the ISL dataset in Figure 9, the class label "8" obtained a minimum recognition rate of 92.3% due to misclassification with 2.6% as "4" and 5.1% as "9" due to similarity in sign gesture with a small change in the number of fingers in case of "9" sign gesture and little change in both finger and thumb position in case of "4" sign. The numeral class "2" was misclassified with "3" by 4.7% and recognized with 95% accuracy as it had similar hand gestures but differences in thumb placement with finger and finger orientation. Overall, our model predicted the class labels accurately for both datasets. The

diagonal values in the confusion matrices show the actual recall values of all the signs, and the consistent rows signify the confused classes.

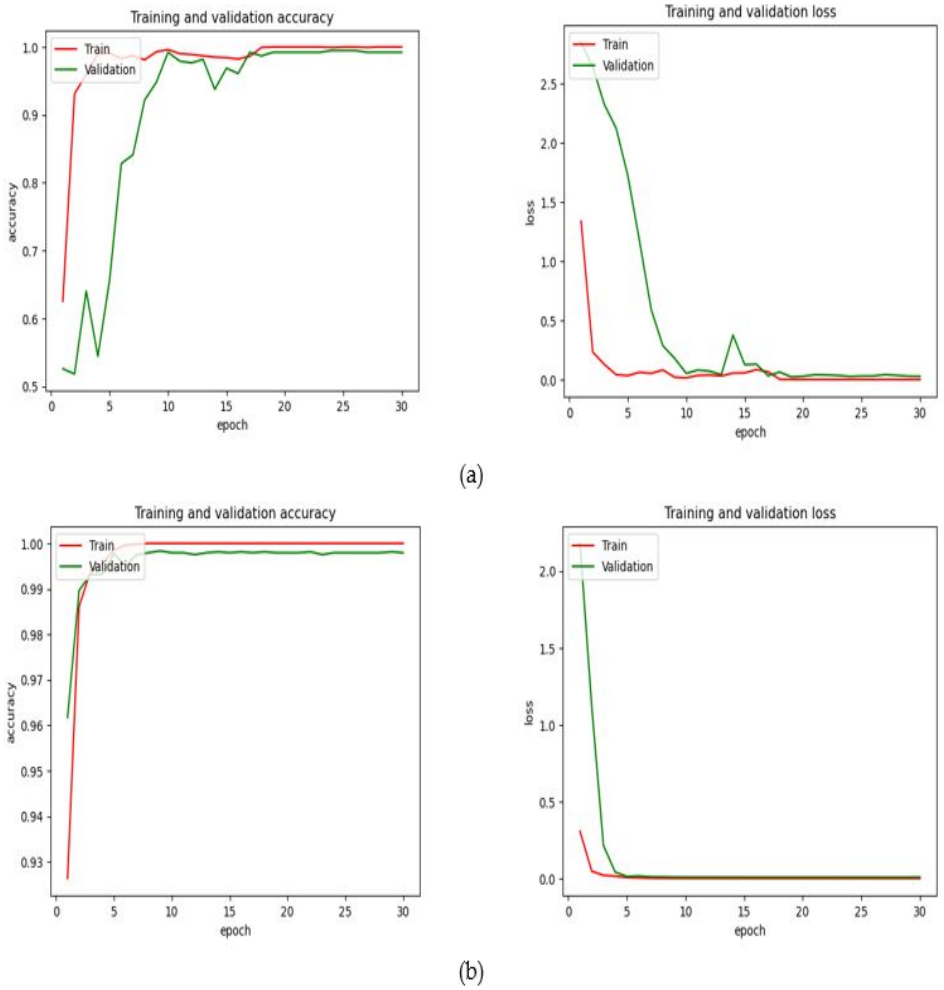

Figure 9. Training and validation accuracy and loss curves for both datasets: (**a**) Massey ASL, (**b**) ISL dataset.

4.1. Comparison with Previous Studies

We compared the obtained results with other related models implemented based on accuracy, model parameters, training epochs, inference time, and time complexity. Table 3 shows that our proposed model has 2.3 million parameters, less than the existing networks. Significantly, the acquired recognition rate of the Massey ASL dataset is 99.22%, and the accuracy of the ISL dataset is 99.79%, which is higher than the previous studies with fewer training epochs and an excellent average inference time on testing images.

4.2. Time Complexity and Order of the Proposed Method

We can determine the efficiency of the proposed algorithm using time complexity. The complexity is computed for each operation involved in the algorithm using Big-Oh notation. This technique measures the maximum amount of time taken by the algorithm. In HOG feature extraction, the calculations are performed on each cell and block for each image. It involves various stages of operations having the complexity of O (p^2), where p is the number of pixels of an image. The algorithm consists of computing gradients and orientations for each pixel, which requires iterating over all pixels in the image. For CBAM-based CNN, the time complexity is O $\left(\sum_{i=1}^{L} S_L^2 * K_L^2 * C_{L-1} * C_L\right)$, where S, K, and C are the dimension of the feature maps, convolutional kernel size, and channel count for

each convolutional layer L. The batch normalization layer, max pooling layer, and CBAM module layer have a time complexity of O (1) hand and have minor operations that do not depend on the size of the input feature. The flattening layer has an O (n ∗ m) complexity. The concatenation layer is a linear operation with an O (n) complexity. The fully connected layer has the complexity of $O\left(\sum_{i=1}^{f} F_f\right)$, where F_f is the number of neurons and f is the number of fully connected layers. The last classification layer has complexity O ($k * h$), where k is the total sign classes and h is the neuron count in the hidden layer [35]. So, we can express the overall complexity as

$$O\left(p^2\right) + O\left(\sum_{i=1}^{L} S_L^2 * K_L^2 * C_{L-1} * C_L + \sum_{i=1}^{f} F_f\right) + O(k * h) \quad (13)$$

Table 3. State-of-the-art comparison of Massey ASL and ISL dataset.

Data	Year	Method (Year)	Accuracy (%)	Training Parameters	Training Epochs	Inference Time (s)
Massey ASL	[30]	2-level ResNet-50 (2020)	99.03	25,636,712	200	0.025
	[31]	VGG16+Attention (2022)	98.02	138.35 M	30	-
	[32]	Canny + HOG+ KNN (2022)	97.6	-	-	0.39
	[33]	Spatial feature learning network (2023)	80.44	3.8 M	-	2.23
	Proposed	**HOG+ CBAM-based CNN**	99.22%	2.3 M	30	0.023 s
Static ISL	[34]	Vision Transformer based approach (2023)	99.29%	7 M	5	-
	[21]	Ensemble (ResNet50 +Attention) (2022)	98.20%	23.5 M	50	-
	Proposed	**HOG+ CBAM-based CNN**	99.79%	2.3 M	30	0.023 s

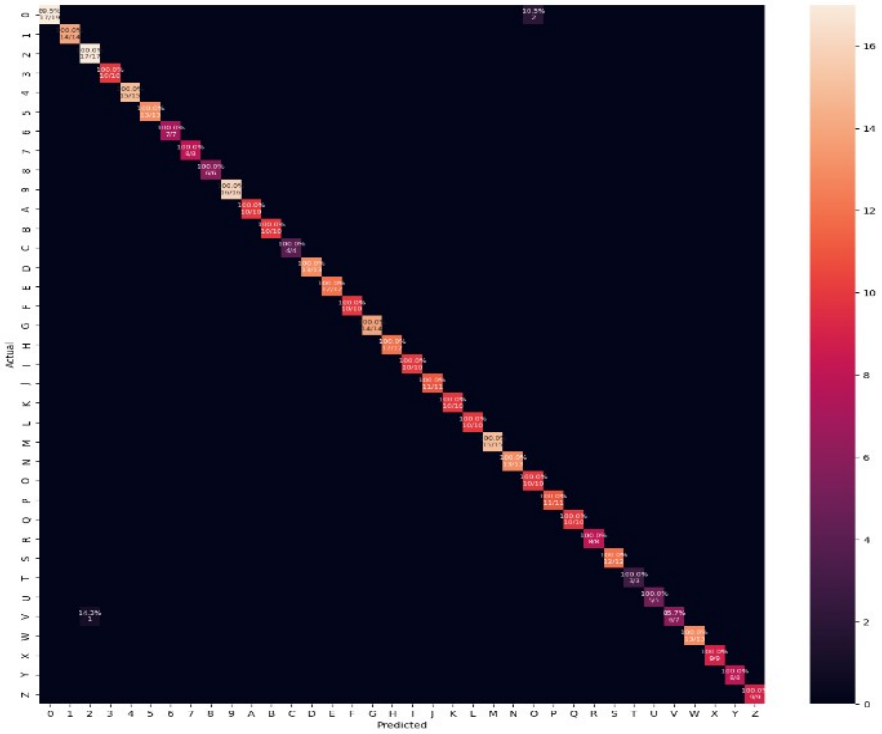

Figure 10. Confusion matrix for classification of Massey ASL database.

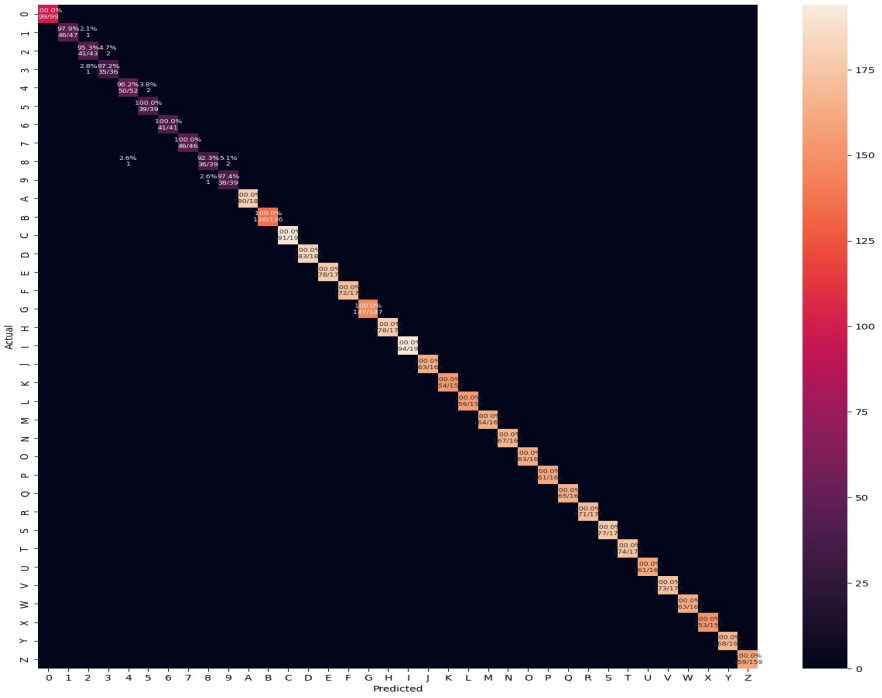

Figure 11. Confusion matrix for classification of ISL database.

We have discussed the comparison of time complexity with other algorithms. The time complexity of two-level ResNet50 [30] is $2 * O\left(\sum_{i=1}^{L} S_L^2 * K_L^2 * C_{L-1} * C_L + \sum_{i=1}^{f} F_f\right)$+ O(k ∗ h) since there are two branches of the CNN model for feature extraction for each layer of the model. In [31], the authors have used fused features of pre-trained VGG16 and attention-based VGG16.

The model has a computational complexity of $2{*}O\left(\sum_{i=1}^{L} S_L^2 * K_L^2 * C_{L-1} * C_L + \sum_{i=1}^{f} F_f\right)$ +O(k ∗ h). The study [32] involves canny edge detection, HOG feature extraction, and KNN for classification, which has an overall time complexity of O (m ∗ n log m ∗ n) + O (n^2) + O(d ∗ F), where m ∗ n is the image size, d, and F is the number of data points and features, respectively. In [34], authors have used a vision transformer encoder with a time complexity of O ($8 * s^2 * d$) + O ($8 * s * d^2$), where s is the sequence length and d is the depth.

It is observed from the analysis that deep learning models used in [30,31] have more time complexity, training epochs, and model parameters as compared to the proposed model. The author used handcrafted features with KNN as a classifier, which is less computationally expensive than the proposed algorithm. Still, the inference time of prediction is longer, and our methodology has a higher accuracy rate. The spatial feature learning network in the paper [33] has implemented a lightweight network by merging features from two convolutional layers with multiscale filters. This model has 3.8 million trainable parameters and an average inference time of 2.23 s, which is comparatively more than the proposed model. The vision transformer in [34] used eight layers of the encoder with four heads with seven million parameters, which is computationally expensive. The authors in [21] have used ensemble network architecture employing ResNet50 with an attention module with more training parameters and epochs.

The proposed method performs superiorly to the other existing methods on all benchmark datasets due to the following reasons: (i) The model combines the handcrafted features from the HOG feature descriptor with the CBAM attention-based CNN features that gather meaningful information from two different streams for better representation of gesture images. (ii) The model acquired good recognition accuracy with fewer model parameters and built an appropriate balance between accuracy and efficiency. (iii) The

framework recognized all the sign gestures accurately with two misclassifications in both datasets, which is comparatively less compared to other existing methods.

5. Conclusions

This work aimed to implement a framework combining CNN with the CBAM module and traditional HOG features to recognize static sign language gestures efficiently. HOG features provide extensive details about hand shape and orientation. The CBAM module employed spatial- and channel-based attention mechanisms within each convolutional layer of the CNN to extract critical and relevant information about images. In our experiment, we analyzed our methodology on two datasets that achieved accuracies of 99.22% and 99.79%, respectively. The model outperformed the state-of-the-art techniques with the least possible parameters and is computationally efficient. We can extend this work to recognize dynamic hand gestures against complex backgrounds for real-time applications and effective recognition of sign language gestures.

Author Contributions: D.K. implemented the methodology, experimental setting, and visualization and wrote the manuscript. R.S.A. supervised, reviewed, and edited the paper. All authors have read and agreed to the published version of the manuscript.

Funding: There is no external funding granted for this research work.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: 1. The Massey ASL dataset is available at https://www.massey.ac.nz/~albarcza/gesture_dataset2012.html (accessed on 31 January 2011). 2. The ISL dataset is available at https://github.com/DeepKothadiya/Static_ISL (accessed on 19 October 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Das, S.; Biswas, S.K.; Purkayastha, B. Automated Indian sign language recognition system by fusing deep and handcrafted features. *Multimed. Tools Appl.* **2023**, *82*, 16905–16927. [CrossRef]
2. Damaneh, M.M.; Mohanna, F.; Jafari, P. Static hand gesture recognition in sign language based on convolutional neural network with feature extraction method using ORB descriptor and Gabor filter. *Expert Syst. Appl.* **2023**, *211*, 118559. [CrossRef]
3. de Castro, G.Z.; Guerra, R.R.; Guimarães, F.G. Automatic translation of sign language with multi-stream 3D CNN and generation of artificial depth maps. *Expert Syst. Appl.* **2023**, *215*, 119394. [CrossRef]
4. Nandi, U.; Ghorai, A.; Singh, M.M.; Changdar, C.; Bhakta, S.; Kumar Pal, R. Indian sign language alphabet recognition system using CNN with diffGrad optimizer and stochastic pooling. *Multimed. Tools Appl.* **2023**, *82*, 9627–9648. [CrossRef]
5. Miah, A.S.M.; Hasan, A.M.; Shin, J.; Okuyama, Y.; Tomioka, Y. Multistage spatial attention-based neural network for hand gesture recognition. *Computers* **2023**, *12*, 13. [CrossRef]
6. Marin, G.; Dominio, F.; Zanuttigh, P. Hand gesture recognition with leap motion and kinect devices. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 1565–1569.
7. Lahiani, H.; Neji, M. Hand gesture recognition method based on hog-lbp features for mobile devices. *Procedia Comput. Sci.* **2018**, *126*, 254–263. [CrossRef]
8. Parvathy, P.; Subramaniam, K.; Prasanna Venkatesan, G.K.D.; Karthikaikumar, P.; Varghese, J.; Jayasankar, T. Development of hand gesture recognition system using machine learning. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 6793–6800. [CrossRef]
9. Sharma, S.; Singhm, S. Vision-based hand gesture recognition using deep learning for the interpretation of sign language. *Expert Syst. Appl.* **2021**, *182*, 115–657. [CrossRef]
10. Xu, J.; Wang, H.; Zhang, J.; Cai, L. Robust hand gesture recognition based on RGB-D Data for natural human–computer interaction. *IEEE Access* **2022**, *10*, 54549–54562. [CrossRef]
11. Masood, S.; Thuwal, H.C.; Srivastava, A. American sign language character recognition using convolution neural network. In *Smart Computing and Informatics: Proceedings of the First International Conference on SCI 2016*; Springer: Singapore, 2018; Volume 2, pp. 403–412.
12. Sruthi, C.J.; Lijiya, A. Signet: A deep learning based indian sign language recognition system. In Proceedings of the 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 4–6 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 596–600.
13. Ma, Y.; Xu, T.; Kim, K. Two-Stream Mixed Convolutional Neural Network for American Sign Language Recognition. *Sensors* **2022**, *22*, 5959. [CrossRef]

14. Eid, A.; Schwenker, F. Visual Static Hand Gesture Recognition Using Convolutional Neural Network. *Algorithms* **2023**, *16*, 361. [CrossRef]
15. Suneetha, M.; Prasad, M.V.D.; Kishore, P.V.V. Multi-view motion modelled deep attention networks (M2DA-Net) for video-based sign language recognition. *J. Vis. Commun. Image Represent.* **2021**, *78*, 103161.
16. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3156–3164.
17. Park, J.; Woo, S.; Lee, J.Y.; Kweon, I.S. Bam: Bottleneck attention module. *arXiv* **2018**, arXiv:1807.06514.
18. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective kernel networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 510–519.
19. Zhang, L.; Tian, Q.; Ruan, Q.; Shi, Z. A simple and effective static gesture recognition method based on attention mechanism. *J. Vis. Commun. Image Represent.* **2023**, *92*, 103783. [CrossRef]
20. Barczak, A.L.C.; Reyes, N.H.; Abastillas, M.; Piccio, A.; Susnjak, T. A new 2D static hand gesture colour image dataset for ASL gestures. *Res. Lett. Inf. Math. Sci.* **2011**, *15*, 12–20.
21. Kothadiya, D.R.; Bhatt, C.M.; Rehman, A.; Alamri, F.S.; Saba, T. SignExplainer: An Explainable AI-Enabled Framework for Sign Language Recognition with Ensemble Learning. *IEEE Access* **2023**, *11*, 47410–47419. [CrossRef]
22. Sharma, S.; Singh, S. ISL recognition system using integrated mobile-net and transfer learning method. *Expert Syst. Appl.* **2023**, *221*, 119772. [CrossRef]
23. Choudhury, A.; Rana, H.S.; Bhowmik, T. Handwritten bengali numeral recognition using hog based feature extraction algorithm. In Proceedings of the 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 22–23 February 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 687–690.
24. Sharma, A.; Mittal, A.; Singh, S.; Awatramani, V. Hand gesture recognition using image processing and feature extraction techniques. *Procedia Comput. Sci.* **2020**, *173*, 181–190. [CrossRef]
25. Arun, C.; Gopikakumari, R. Optimisation of both classifier and fusion based feature set for static American sign language recognition. *IET Image Process.* **2020**, *14*, 2101–2109.
26. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–19.
27. Wang, Y.; Zhang, Z.; Feng, L.; Ma, Y.; Du, Q. A new attention-based CNN approach for crop mapping using time series Sentinel-2 images. *Comput. Electron. Agric.* **2021**, *184*, 106090. [CrossRef]
28. Tato, A.; Nkambou, R. Improving adam optimizer. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
29. Katoch, S.; Singh, V.; Tiwary, U.S. Indian Sign Language recognition system using SURF with SVM and CNN. *Array* **2022**, *14*, 100141. [CrossRef]
30. Rathi, P.; Kuwar Gupta, R.; Agarwal, S.; Shukla, A. Sign language recognition using resnet50 deep neural network architecture. In Proceedings of the 5th International Conference on Next Generation Computing Technologies, Dehradun, India, 20–21 December 2019; SSRN: Parkville, Australia, 2020.
31. Barbhuiya, A.A.; Karsh, R.K.; Jain, R. Gesture recognition from RGB images using convolutional neural network-attention based system. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e7230. [CrossRef]
32. Adeyanju, I.A.; Bello, O.O.; Azeez, M.A. Development of an american sign language recognition system using canny edge and histogram of oriented gradient. *Niger. J. Technol. Dev.* **2022**, *19*, 195–205. [CrossRef]
33. Bhaumik, G.; Govil, M.C. SpAtNet: A spatial feature attention network for hand gesture recognition. *Multimed. Tools Appl.* **2023**. [CrossRef]
34. Kothadiya, D.R.; Bhatt, C.M.; Saba, T.; Rehman, A.; Bahaj, S.A. SIGNFORMER: DeepVision Transformer for Sign Language Recognition. *IEEE Access* **2023**, *11*, 4730–4739. [CrossRef]
35. Umar, S.S.I.; Iro, Z.S.; Zandam, A.Y.; Shitu, S.S. Accelerated Histogram of Oriented Gradients for Human Detection. Ph.D. Thesis, Universiti Teknologi Malaysia, Johor Bahru, Malaysia, 2016.