electronics

MDPI

Article

Multi-Stream Isolated Sign Language Recognition Based on Finger Features Derived from Pose Data

Ali Akdag [1,*] and Omer Kaan Baykan [2]

[1] Department of Computer Engineering, Taşlıçiftlik Campus, Tokat Gaziosmanpaşa University, 60250 Tokat, Türkiye
[2] Department of Computer Engineering, Konya Technical University, 42250 Konya, Türkiye; okbaykan@ktun.edu.tr
* Correspondence: ali.akdag@gop.edu.tr

Abstract: This study introduces an innovative multichannel approach that focuses on the features and configurations of fingers in isolated sign language recognition. The foundation of this approach is based on three different types of data, derived from finger pose data obtained using MediaPipe and processed in separate channels. Using these multichannel data, we trained the proposed MultiChannel-MobileNetV2 model to provide a detailed analysis of finger movements. In our study, we first subject the features extracted from all trained models to dimensionality reduction using Principal Component Analysis. Subsequently, we combine these processed features for classification using a Support Vector Machine. Furthermore, our proposed method includes processing body and facial information using MobileNetV2. Our final proposed sign language recognition method has achieved remarkable accuracy rates of 97.15%, 95.13%, 99.78%, and 95.37% on the BosphorusSign22k-general, BosphorusSign22k, LSA64, and GSL datasets, respectively. These results underscore the generalizability and adaptability of the proposed method, proving its competitive edge over existing studies in the literature.

Keywords: sign language recognition; deep learning; feature fusion

Citation: Akdag, A.; Baykan, O.K. Multi-Stream Isolated Sign Language Recognition Based on Finger Features Derived from Pose Data. *Electronics* **2024**, *13*, 1591. https://doi.org/10.3390/electronics13081591

Academic Editors: Rudrasis Chakraborty and Indrasis Chakraborty

Received: 20 January 2024
Revised: 7 April 2024
Accepted: 15 April 2024
Published: 22 April 2024

1. Introduction

Sign language, composed of visual and kinesthetic elements, is the primary means of communication for approximately 70 million deaf individuals worldwide [1]. This language combines manual and non-manual features such as hand movements, facial expressions, and body posture, offering a rich form of expression [2]. Learning and using this language plays a critical role in the social integration of deaf individuals; however, a lack of knowledge about sign language in the general population and a scarcity of sign language interpreters create communication barriers. In this context, computer vision and machine learning technologies have the potential to overcome these barriers by automatically translating sign language into text or audio formats.

Sign language recognition (SLR) systems are being developed as part of this technological effort, aiming to serve as an effective bridge between the deaf and those who do not know sign language. SLR systems generally fall into two main categories: sensor-based and image-based systems [3]. Sensor-based systems [4–6] require specialized equipment like electronic gloves and directly detect hand and arm movements through this equipment. Using cameras and advanced image processing technologies, image-based systems [7,8] visually detect the user's hand, face, and body movements. The capabilities of these systems can be evaluated in two main categories: static and dynamic. Static SLR typically identifies unchanging hand signs and gestures, while dynamic SLR uses more advanced techniques to capture the fluid nature of sign language, recognizing words (isolated) or sentences (continuous) [9].

In SLR, the detailed analysis of finger movements and positions could play a pivotal role [10]. While SLR often involves holistic hand movements and facial expressions, the nuanced interpretation provided by finger gestures may also be crucial for accurate communication. To our knowledge, no studies in our literature review specifically focus on fingers in isolated SLRs. To fill this potential gap, we designed our study to explore the significance of finger movements in enhancing the accuracy and reliability of isolated SLR systems. Our approach aims to address specific challenges, such as the overlapping of fingers and seeks to improve interactions with deaf individuals, potentially reducing the reliance on sign language interpreters.

The main contributions of this research are as follows:

- Advanced Finger-Based Features: We create three distinct feature sets related to fingers, which include FINGER, obtained by visualizing finger poses in separate channels; MERGED-FINGER, further detailed by cropping and merging finger regions; and FD-FINGER, capturing temporal changes through the frame difference method. These three feature sets allow for a detailed analysis of finger movements and configurations in the SLR process, enabling systems to better understand subtle nuances.
- MultiChannel-MobileNetV2 Model: We developed an innovative model based on the MobileNetV2 architecture, capable of efficiently processing multichannel finger-based features. This model significantly enhances accuracy and precision in the field of SLR.
- Non-Manual Features: Our extended approach utilizes visuals of the body and face derived from pose data and visuals created by applying the frame difference method to these images. We train the MobileNetV2 model on these visuals to extract features, enhancing SLR accuracy and emphasizing the importance of non-manual expressions.
- Multi-Stream SLR Method: In this methodology, we reduce the dimensions of features related to fingers, bodies, and faces extracted from the trained MultiChannel-MobileNetV2 and MobileNetV2 models through Principal Component Analysis (PCA). Subsequently, we fuse these features and classify them with Support Vector Machine (SVM) to create a more comprehensive SLR system. This integrated approach significantly innovates in the field by maximizing accuracy and comprehensiveness in the SLR process.

The rest of this paper is structured as follows: Section 2 discusses Related Works, emphasizing advancements in sign language recognition. Section 3, Materials and Methods, details our approach, including developing the MultiChannel-MobileNetV2 model. Section 4, Experimental Studies, presents our findings and the evaluation of our models. Lastly, Section 5, Conclusions, summarizes our contributions and suggests directions for future research.

2. Related Works

Early studies in the field of SLR relied on handcrafted features and traditional machine learning techniques. These traditional methods typically included processes such as feature extraction, dimensionality reduction, and the use of simple classification models. These processes aimed to extract features from static images or simple video clips of the relevant signs. SLR was performed using various traditional classification algorithms based on features extracted from sign language letters, numbers, or words using feature extraction techniques such as Hough Transform [11], Contour Analysis [12], Local Binary Pattern [13,14], Gabor Filter [15], HOG [16,17], SIFT [18,19], SURF [20], optical flow [21], Dynamic Time Warping [22], Hidden Markov Models [23,24], and temporal accumulative features [25]. Methods based on handcrafted features and traditional machine learning techniques, while effective for a predefined and limited number of signs, typically struggle to recognize new or rare signs due to their limited flexibility and generalization ability. Additionally, detailed data analysis is required to extract powerful handcrafted features that can represent the data, which is a task that requires domain-specific expertise.

Deep learning has surpassed the limitations of traditional machine learning techniques by providing a high level of automation and learning capacity. Convolutional Neural Net-

works (CNNs), in particular, have revolutionized the field of image classification [26]. Two-dimensional CNNs have been frequently used for digit and letter signs, which usually do not contain temporal information and consist of static images. Damaneh et al. [7] proposed a hybrid method using CNN, Gabor Filter, and ORB feature descriptor for recognizing static hand gestures, achieving accuracies of 99.92%, 99.8%, and 99.80% on the Massey [27], American Sign Language (ASL) alphabet [28], and ASL datasets [29], respectively. Das et al. [30] used a hybrid method combining CNN and Random Forest classifiers to accurately predict numerals and characters in Bangla Sign Language with accuracies of 97.33% and 91.67%, respectively. Aldhahri et al. [31] developed a CNN architecture that identified letters in Arabic Sign Language with 94.46% accuracy. Ma et al. [32] used the Two-Stream Mixed (TSM) method to improve the correlation of feature expression between two consecutive time frames in datasets containing dynamic gestures like the letters J and Z. Their proposed TSM-ResNet50 model achieved 97.57% accuracy on the ASL alphabet dataset. Alsharif et al. [33] obtained accuracies of 99.50%, 99.51%, 99.95%, 99.98%, and 88.59% for recognizing the ASL alphabet using AlexNet, ConvNeXt, EfficientNet, ResNet-50, and Vision Transformer models, respectively.

While 2D-CNNs have achieved significant success in areas such as image classification and SLR, they have some limitations in capturing temporal features. This can pose a problem for applications where understanding changes over time, such as sign language words, is important. Therefore, models more suitable for capturing temporal information, such as Recurrent Neural Networks (RNNs) [34], Long Short-Term Memory Networks (LSTM) [35], and Transformers [36], have been used. Venugopalan et al. [8] successfully classified 13 words from Indian Sign Language, which are prevalently used by deaf farmers, with a 76.21% accuracy using their proposed GoogleNet-BiLSTM architecture. In a separate study, Masood et al. [37] employed Convolutional Neural Networks (CNNs) for extracting spatial features and Recurrent Neural Networks (RNNs) for temporal features, thereby accurately classifying 46 words from the Argentine Sign Language dataset [38] with a precision of 95.2%. Furthermore, Shin et al. [39] developed an SLR model leveraging CNNs and Transformer architectures, which accomplished an 89% accuracy rate on a dataset of 77 Korean Sign Language words.

Other structures that can capture both spatial and temporal information include 3D Convolutional Neural Networks (3D-CNNs) [40] and (2+1)D Convolutional Neural Networks, (2+1)D-CNNs [41]. Additionally, 3D-CNNs are an extension of traditional 2D-CNNs and can perform convolution operations in both spatial (space) and temporal (time) dimensions. This allows 3D-CNNs to directly process datasets that contain temporal information, such as SLR. On the other hand, (2+1)D-CNNs initially extract spatial features using 2D convolutions and then perform temporal analysis on these spatial features using 1D convolutions. Neto et al. [42] have achieved recognition accuracy of 93.9% on the LSA64 dataset [38], which consists of 64 words from Argentine Sign Language, using their proposed 3D-CNN model. In a separate study, Wang et al. [43] proposed a (2+1)D convolution-based (2+1)D-SLR network on the same dataset, achieving a performance of 98.7%. Özdemir et al. [44] obtained 88.53% accuracy using Improved Dense Trajectories and 78.85% accuracy using the MC3-18 model [41], which combines both 2D and 3D convolutions, in their study on the BosphorusSign22k [44,45] dataset consisting of 744 sign words. Sincan et al. [46] proposed a hybrid method in which RGB-MHI features are analyzed with Resnet-50 in addition to the 3D-CNN model to classify sign language words, which achieved 93.53% and 94.83% performance on AUTSL [47] and BosphorusSign22k datasets, respectively. Adaloglou et al. [48] employed the GoogLeNet + TConvs framework [49], the 3D-ResNet architecture [50], and the I3D model [51] in their study on Greek Sign Language (GSL), which consists of 310 signs. These models, in signer-independent evaluations, achieved accuracies of 86.03%, 86.23%, and 89.74%, respectively.

In recent years, methods have significantly increased that employ pose data as input data in SLR systems. These methods detect key points of the human body, particularly hands and fingers, as well as facial expressions and body posture, and use these data

as the basis for SLR. Samaan et al. [52] trained GRU, LSTM, and Bi-LSTM models using pose data obtained with MediPipe [53] and achieved over 99% success for ten sign words. Podder et al. [54] achieved 87.69% accuracy with their segmentation-based MobileNetV2-LSTM-SelfMLP model for Arabic SLR. Graph neural networks [55], which utilize these pose data, have also achieved significant success in understanding sign language words. Selvaraj et al. [56] implemented LSTM, Transformer, Spatial–Temporal Graph Convolutional Network (ST-GCN), and Structure Learning Graph Convolutional Network (SL-GCN) models for isolated SLR across diverse datasets. Their findings for the GSL dataset [48] indicated accuracies of 86.6%, 89.5%, 93.5%, and 95.4% for each respective model.

The task of SLR also inherently required the use of multiple streaming methods. This is because a robust SLR system requires a detailed analysis of all the components that make up the sign. In this context, Gökçe et al. [57] used hand, face, and body images separately as inputs for MC3-18 [41] and integrated the outputs of these networks at the score level. The developed model showed an accuracy rate of 94.94% on the BosphorusSign22k dataset. Gündüz and Polat [58] used the Inception3D (I3D) model to extract features from hand, face, full-body, and optical flow images and combined these with features derived from pose data processed through an LSTM network. These composite data were then inputted into a two-layer neural network. In their research using the general subset of the BosphorusSign22k dataset consisting of 174 signs, they achieved an 89.3% accuracy rate.

In the field of isolated SLR using computer vision, a review of current literature reveals that while there is a focus on the comprehensive data comprising a sign or on the general movements of the hand, finer details such as finger movements and configurations are often overlooked. Yet, in sign language, the position and movement of fingers are crucial in determining meaning. Thus, a thorough analysis of finger data can enhance accuracy and provide a more detailed interpretation in SLR. Furthermore, challenges like overlapping fingers and data gaps in certain positions make accurate sign recognition more complex. The multichannel approach we propose in this study, processing finger pose images in separate channels, presents an innovative solution to address these issues of overlap and data deficiency. In addition to finger pose features, an SLR system is proposed that takes into account full-body pose and face pose images and their temporal information.

3. Materials and Methods

The SLR method proposed in this paper aims to classify isolated sign language words using features specifically based on finger data. A detailed analysis of the fingers was made possible with the pose data obtained using the MediaPipe library. We created FINGER data by visualizing each finger pose in a separate channel. Furthermore, MERGED-FINGER data, which provide a closer look at the spatial features of the fingers, were created by cropping and merging finger regions from the FINGER data. In addition, we developed FD-FINGER data to capture temporal information by applying the frame difference method to the FINGER data. In response to these features, we propose the MultiChannel-MobileNetV2 model, which processes each channel separately. The proposed method applies PCA to the features obtained from these models, followed by their combination and classification using SVM. The proposed method, based on finger data, is extended and made more robust by including features of full-body and face data obtained from MobileNetV2 models, each trained separately. The general diagram of the proposed method is shown in Figure 1.

This section elucidates the key steps supporting our SLR process: the datasets utilized, data preprocessing techniques, the frame difference method, details of our proposed MultiChannel-MobileNetV2 model, min–max normalization, and dimensionality reduction via PCA, and, finally, the SVM algorithm used for classification. Each segment will reveal how our methodology has been comprehensively and effectively designed.

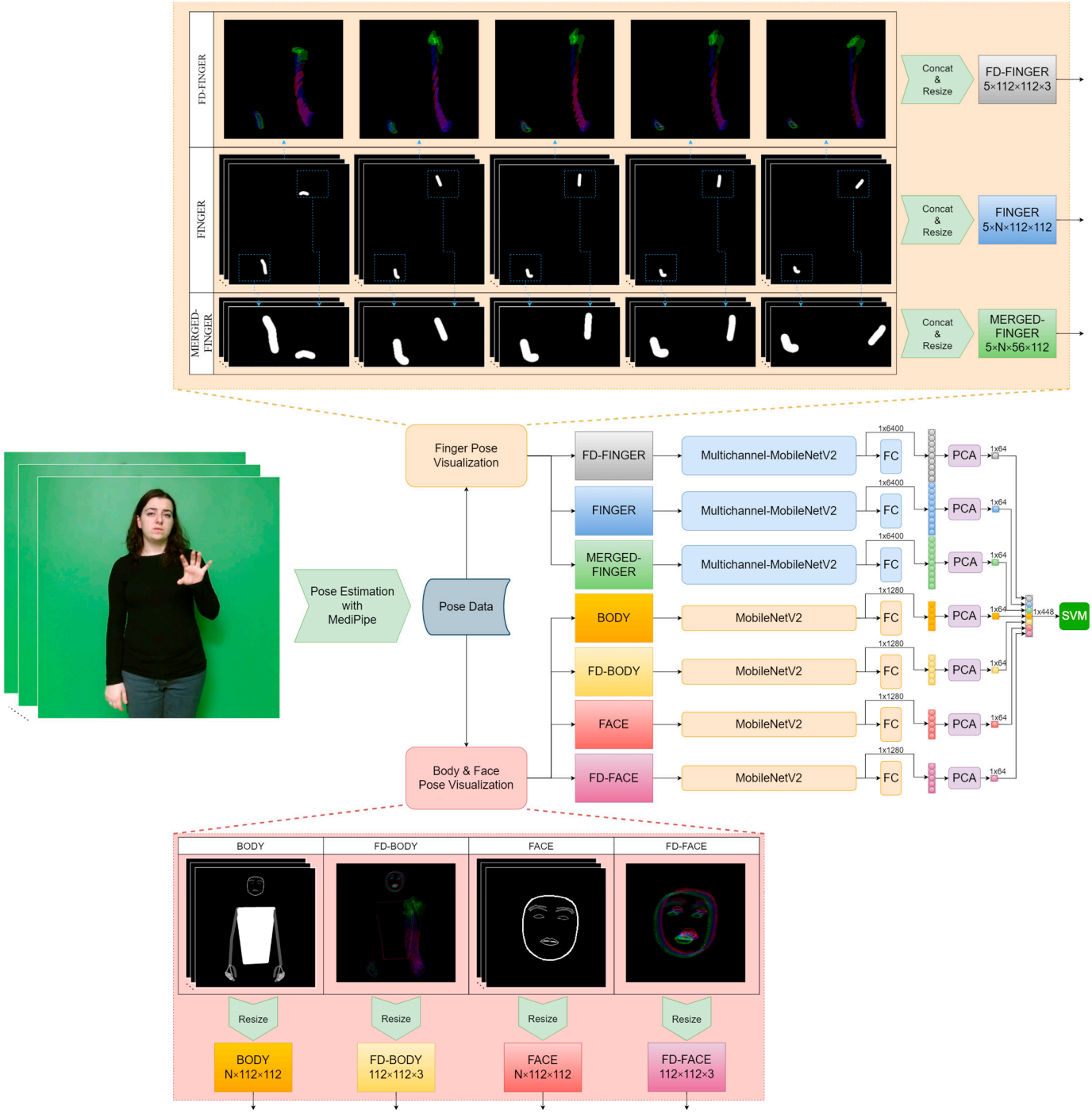

Figure 1. Diagram of the proposed method (N: frame size) (frame is from a video from the BosphorusSign22k dataset).

3.1. Datasets

BosphorusSign22k is an isolated Turkish Sign Language dataset created by Özdemir et al. [44,45]. Prepared by six signers, comprising four females and two males, this dataset comprises 22,542 videos across 744 different sign language word classes. The BosphorusSign22k dataset is divided into three main thematic areas: general (174 class), health (428 class), and finance (163 class). In our experimental study, the "general" subset of the BosphorusSign22k dataset was utilized. The BosphorusSign22k-general subset contains 5788 videos and 174 different sign language words. Among these data, 949 videos belonging to one signer have been specifically set aside for testing purposes. After

the experimental studies, the proposed method is further evaluated using the entire BosphorusSign22k dataset.

We also perform an evaluation using the LSA-64 dataset to examine the performance of our proposed method in different languages and on datasets with various numbers of classes. The LSA-64 dataset, created by Ronchetti et al. [38], consists of 64 sign words from Argentine Sign Language performed by ten signers, including three females and seven males. In the dataset, each signer repeated each word five times, resulting in a total of 3200 video images. The dataset creators did not specifically divide it into training and test sets. Therefore, in our study, based on the literature, the dataset was randomly divided into training and test sets in an 80:20 ratio, with one out of ten signers separated for testing and the rest for training data, and in another arrangement, the fifth and tenth signers were separated for testing, resulting in three different configurations.

Another dataset that we evaluated in our work is the Greek Sign Language (GSL) dataset [48], which consists of 310 signs. This dataset is generated by seven different signers, comprising one female and six males. The total number of videos is 40,826. All the data generated by one of the signers are divided into 2331 validation data and 3500 test data.

Using these diverse datasets, we aim to validate our proposed method thoroughly, emphasizing its reliability, flexibility, and scalability. This effort seeks to demonstrate our method's broad suitability for SLR and interpretation tasks. A summary of all datasets used in our study is given in Table 1.

Table 1. Datasets used in this study.

Dataset	Signer Size	Sample Size	Class Size
BosphorusSign22k-general [44,45]	6	5788	174
BosphorusSign22k [44,45]	6	22,542	744
LSA64 [38]	10	3200	64
GSL [48]	7	40,826	310

3.2. Data Preprocessing

In our study, we used MediaPipe to obtain finger pose images. MediaPipe, an open-source library developed by Google, is based on deep learning and computer vision techniques [53]. Its "Holistic" module provides an integrated solution for monitoring movements of the face, hands, and body, offering comprehensive analysis capabilities [59]. The Holistic module detects body, face, and hand movements using a total of 543 key points: 33 for the body, 468 for the face, and 21 for one hand.

In the preliminary process of obtaining finger pose images, landmark points corresponding to each hand were first acquired from each video frame using the MediaPipe Holistic solution. However, in some cases, such as when the hand covers the other hand or when the movement is fast, the landmark points of the hand may not be detected. For this reason, the linear interpolation method was applied to complete the missing landmark data. For example, for n missing data points between x_0 and x_1, the value of the i-th data point is obtained using Equation (1):

$$x_i = x_0 + i\frac{(x_1 - x_0)}{(n+1)}, \quad (1)$$

In this equation, x_i represents the estimated value of the i-th missing data point. The variable i is an index indicating the position of the currently estimated missing data point relative to the starting point x_0, with i = 1 being the first missing point immediately after x_0 and $i = n$ being the last one just before x_1.

These landmark points were then used to depict the image of each finger in a separate channel. These images show the positions, movements, and interrelations of the fingers in detail. This detailed visualization made it easier to examine finger movements and positions, bringing out the necessary fine details for accurate SLR. An illustration of the

acquisition of FINGER images from an image containing a hand is shown in Figure 2. In this figure, the pose landmark points obtained with MediaPipe Holistic are first shown representatively. Subsequently, visuals illustrating a detailed rendering of each finger in white on a black background, using these pose landmark points, are provided.

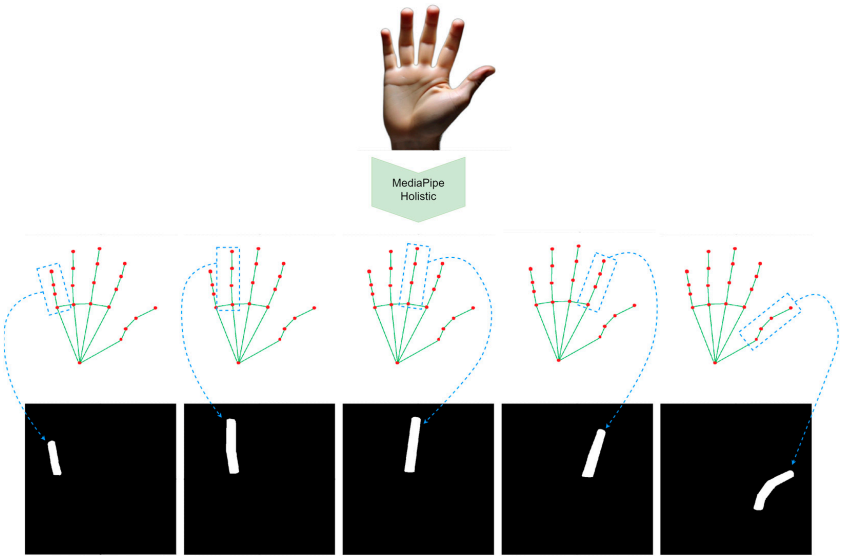

Figure 2. Obtaining FINGER data from a hand image.

Following the creation of the frames containing the finger pose images, the regions containing the images of the fingers were then cropped and merged. These MERGED-FINGER images provided a closer and more detailed analysis of the FINGER, facilitating a clearer resolution of finger movements with complex and subtle details in sign language. A visualization of the generation of FINGER and MERGED-FINGER images from a sign language video frame is shown in Figure 3.

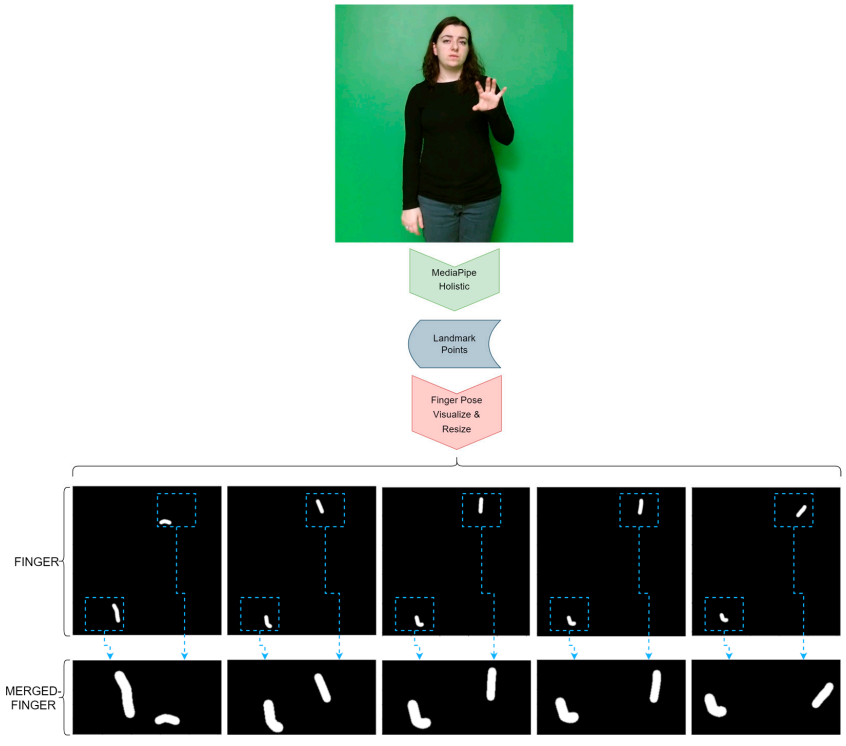

Figure 3. Generation of FINGER and MERGED-FINGER data from a sign language video frame (frame is from a video from the BosphorusSign22k dataset).

Our strategy of depicting finger pose images in separate channels enabled us to track the movement and position of each finger independently, even in situations where fingers were very close to or overlapped with each other. At the end of this process, the obtained FINGER images and MERGED-FINGER images were resized to 112 × 112 and 56 × 112, respectively, to train our MultiChannel-MobileNetV2 models.

A robust SLR system requires the combined evaluation of manual and non-manual features. For this reason, in addition to the finger pose data, full-body pose and face pose images were generated. Hand pose images were also created to examine the effect of representing each finger pose data point in a separate channel. These data were then resized to 112 × 112. These data, generated from an example frame, are shown in Figure 4.

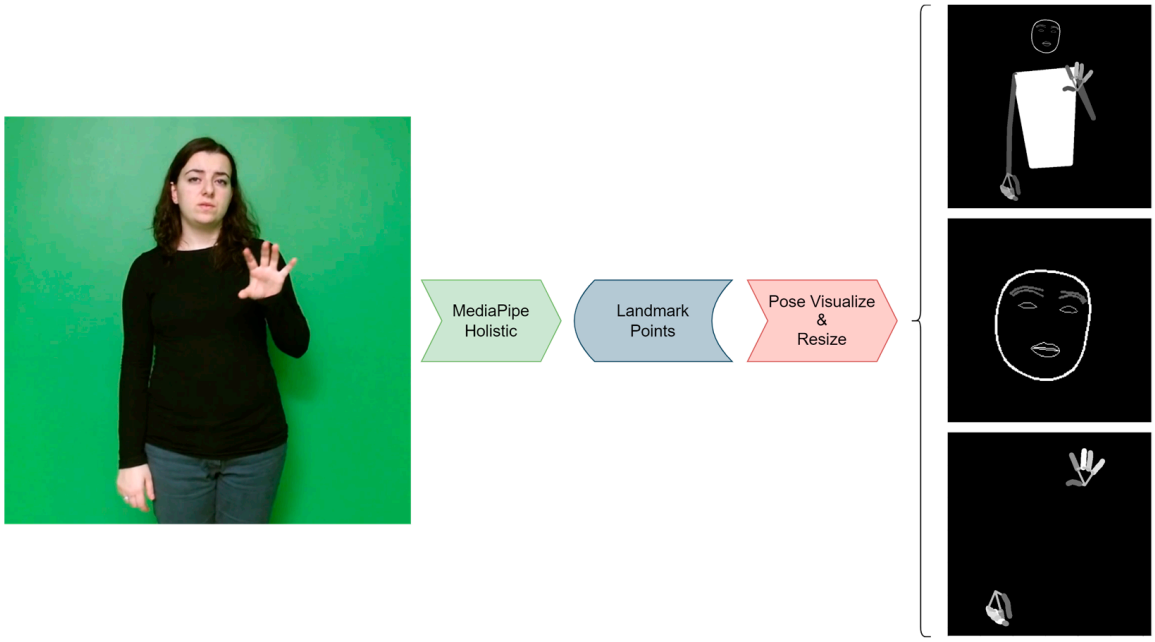

Figure 4. Generation of BODY, FACE, and HANDS data from a sign language video frame (frame is from a video from the BosphorusSign22k dataset).

3.3. Obtaining Temporal Information with Frame Difference Method

The frame difference method is a widely used motion detection method in video analysis [60–62]. This method was used in our study to extract temporal information from videos obtained from pose data. This method is ideal for studying motion-oriented communication forms because it effectively visualizes the intensity and temporal distribution of motion [61].

In applying the frame difference method, the absolute value of the pixel difference between consecutive frames is first calculated. The formulation is shown in Equation (2):

$$D_t(x, y) = |F_t(x, y) - F_{t-1}(x, y)|, \quad (2)$$

where $F_t(x, y)$ represents the pixel value of the video stream at the (x, y) coordinate at time t, and $F_{t-1}(x, y)$ is the pixel value at the (x, y) coordinate at the previous time $t - 1$. The resultant $D_t(x, y)$ represents the frame difference at time t, which is considered an indicator of motion. The motion map is then obtained by summing the frame differences across the entire video. As an extra here, instead of summing the frame differences over the whole video, we divide the video into three equal parts and obtain separate motion maps for each part so that the motion map we obtain for each part represents a channel of the image

we will create. For each part, the motion map $T_k(x, y)$ (where k is the channel number) is calculated as the sum of all frame differences in that segment (Equation (3)):

$$T_k(x, y) = \sum_{t=start}^{end} D_t(x, y), \quad (3)$$

where start and end are the start and end frame numbers for each channel. The motion map of each channel, $T_k(x, y)$, is then normalized and scaled from 0 to 255. This process makes the data suitable for visual analysis. Normalization is performed using Equation (4):

$$T_{norm,k}(x, y) = 255 \times \frac{T_k(x, y) - \min(T_k)}{\max(T_k) - \min(T_k)}, \quad (4)$$

This method was applied for each channel in the FINGER data, and frame difference-FINGER (FD-FINGER) data of size $5 \times 112 \times 112 \times 3$ were obtained from a single FINGER data point. Similarly, the frame difference method was applied to the BODY and FACE data to generate $112 \times 112 \times 3$-sized frame difference-BODY (FD-BODY) and frame difference-FACE (FD-FACE) datasets. These datasets, FD-FINGER, FD-BODY, and FD-FACE, encompass the extracted temporal dynamics from finger, body, and face movements, respectively, providing a comprehensive understanding of sign language gestures over time. A visualization of how FD-BODY data are derived from BODY data is presented in Figure 5. This illustration demonstrates that a video sample containing N frames is segmented into three equal parts. The frame difference method is then individually applied to each segment. Subsequently, the outcomes of these segments are amalgamated to create a composite RGB visual representation.

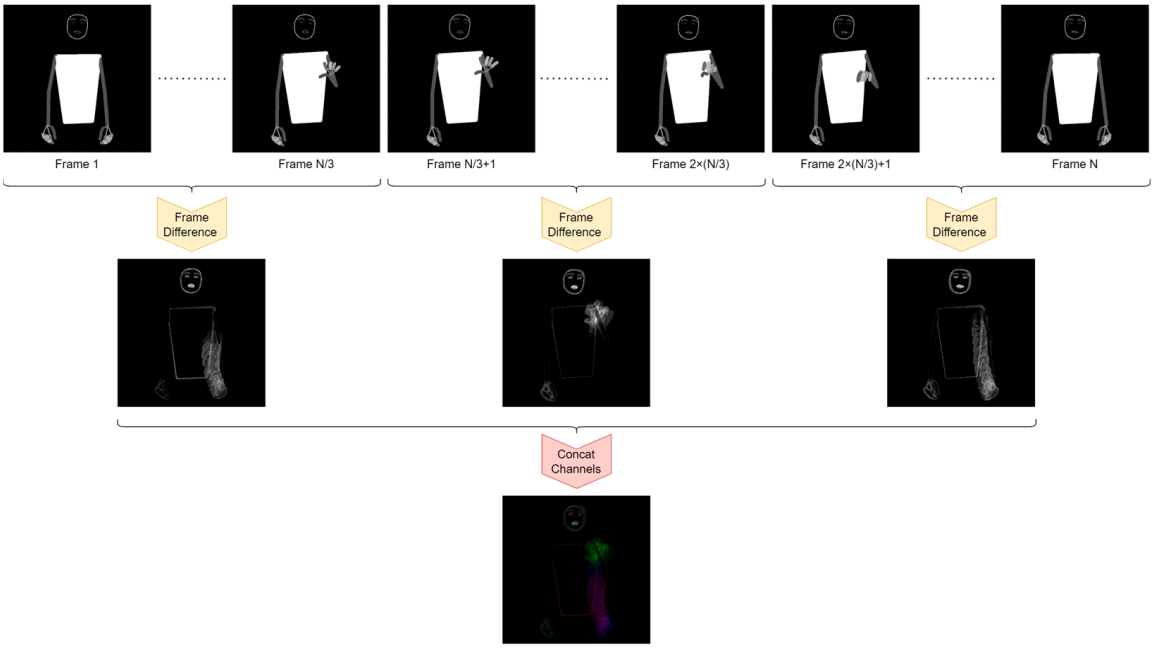

Figure 5. Generation of FD-BODY data from BODY data (N: frame size).

This representation of temporal information is crucial for training our SLR model, as it allows for understanding the dynamics of gestures over time, not just their static snapshots. Figure 6 showcases examples of the data obtained through this method, displaying FINGER, BODY, and FACE data at the top, with their corresponding frame difference versions, FD-FINGER, FD-BODY, and FD-FACE, illustrated at the bottom.

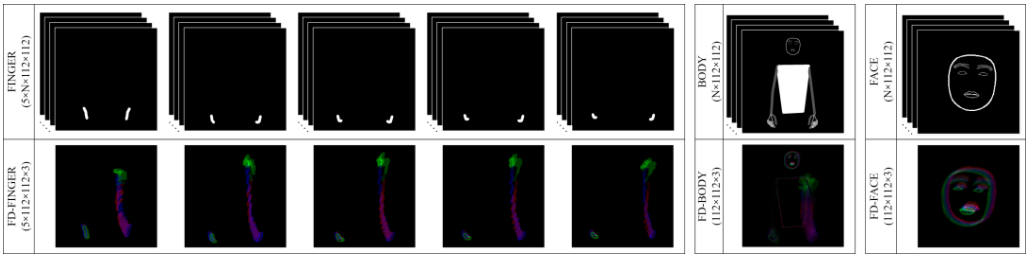

Figure 6. FD-FINGER, FD-BODY, and FD-FACE data consisting of FINGER, BODY, and FACE data (N: frame size).

3.4. MultiChannel-MobileNetV2

Our study proposes MultiChannel-MobileNetV2, an innovative multichannel adaptation of the MobileNetV2 architecture, which holds significant importance in the field of deep learning. MobileNetV2, developed by Sandler et al. [63], offers a lightweight and high-performance structure suitable for mobile and embedded systems, featuring characteristics such as depthwise-separable convolutions, inverted residual structures, and linear bottlenecks. The MultiChannel-MobileNetV2 is built upon this foundational MobileNetV2 architecture, combining five separate pre-trained MobileNetV2 networks to process different data channels individually. The use of pre-trained models accelerates the training process of our model and enhances its capability to recognize features trained on extensive datasets. The proposed MultiChannel-MobileNetV2 architecture, its component MobileNetV2 architecture, and the Inverted Residual Block structure, which is also a component of it, are shown in Figure 7.

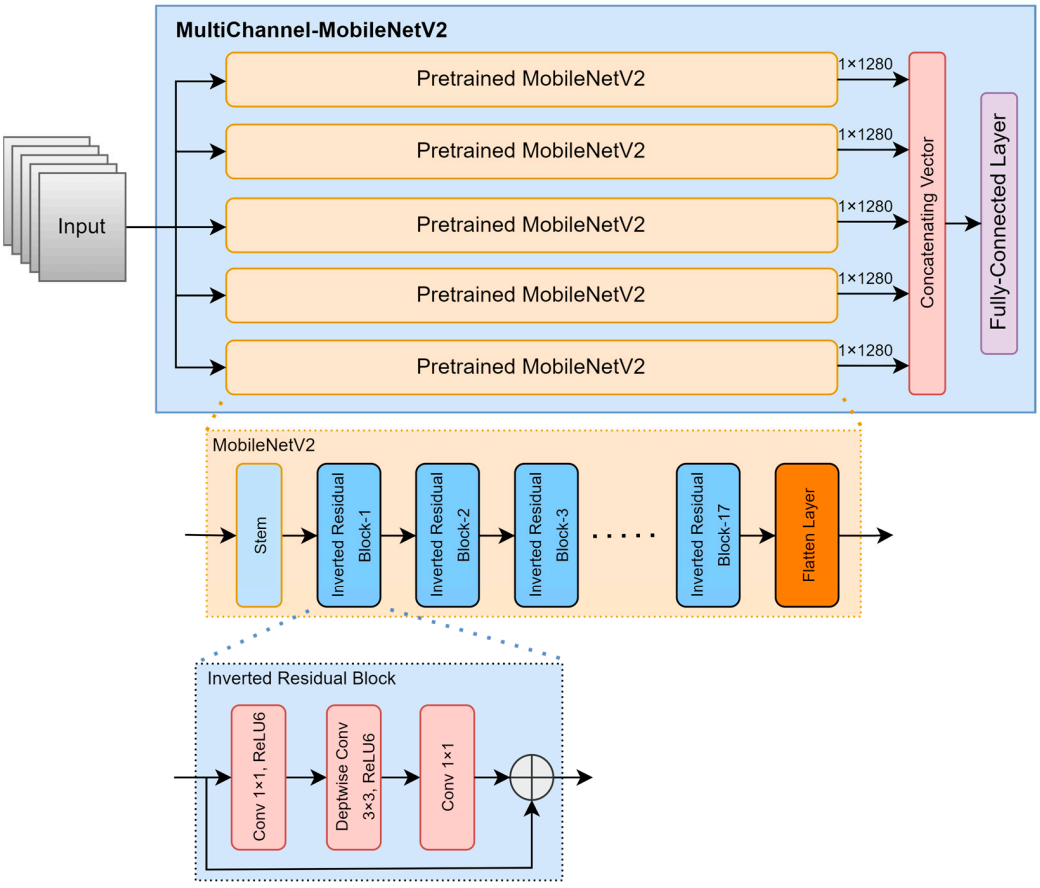

Figure 7. Proposed MultiChannel-MobileNetV2 architecture and its subcomponents.

This multichannel approach allows for separate and detailed analysis of data in each channel. Each sub-model is configured to process a different channel through customizations made at the initial layer of the model. This configuration, involving the alteration of the input dimension of the first convolutional layer, enables each model to accept 32-channel inputs for FINGER and MERGED-FINGER data and 3-channel inputs for FD-FINGER data, thus facilitating the processing of different types of data. From the flattening layer before the Fully Connected layer of MobileNetV2, 1280 features are obtained. In our model, the outputs from the flattening layers of these five sub-networks are combined to form a single output vector ($5 \times 1280 = 6400$). This concatenated vector is then processed by a Fully Connected layer to produce the final output of the model; this output is dimensioned according to the specified number of classes and can be used for a specific classification task.

In our study, MobileNetV2 models were also trained using BODY, FD-BODY, FACE, FD-FACE, and HANDS data. For the models that utilized BODY, FACE, and HANDS data, the convolution process in the first layer was modified to have an input size of 32, as these data were utilized by randomly selecting 32 frames during training. No additional modifications were made to the models for FD-BODY and FD-FACE data. Moreover, the output layers of all the deep learning models used in the study were restructured according to the number of classes in the datasets employed for training.

3.5. Min–Max Normalization, Dimension Reduction with PCA

Each MultiChannel-MobileNetV2 model, when trained on FINGER, MERGED-FINGER, and FD-FINGER data, produces a feature vector comprising 6400 elements. When combining these features for classification, the total feature size triples due to the combination of data from all three sources. This large feature set not only increases the computational burden in the classification and analysis processes but can also have a negative impact on the training and generalization ability of the model, often referred to as the curse of dimensionality [64]. The use of PCA is an ideal method to overcome these challenges and meaningfully reduce the size of the dataset while retaining its key features [65]. Principal Component Analysis (PCA) functions as an effective tool in statistical methodology, specially designed to handle the intricacies of data with high dimensions. The technique operates by converting existing features into a distinct group of variables known as principal components. These components are unique in their orthogonality and their capacity to capture the greatest degree of variance in the data. Typically, the foremost principal components are the most informative, encapsulating the bulk of vital information. This enables a significant reduction in the data's dimensionality, all while effectively preserving their most critical attributes [66].

Before using the PCA method, we applied min–max normalization to the feature datasets. This normalization process is a method that scales each feature in the dataset within a certain range. Usually, this range is [0, 1]. The formula for min–max normalization is expressed in Equation (5).

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}, \tag{5}$$

In this equation, X_{norm} is the post-normalization value, X is the original value, and X_{min} and X_{max} represent the minimum and maximum values of the feature, respectively. This process ensures that all features are on the same scale, contributing to a more effective application of PCA.

In the application of PCA, the covariance matrix of the dataset is first calculated. The covariance matrix expresses the variances and correlations between data features. First, the centered data matrix $X_{centered}$ is created by subtracting the mean value from each feature of the data matrix. Then, the covariance matrix of this centered data matrix is calculated (Equation (6)):

$$\Sigma = \frac{1}{n-1} X_{centered}^{T} X_{centered}, \tag{6}$$

In this formula, n represents the number of observations and Σ is the covariance matrix. The eigenvalues and eigenvectors of the covariance matrix indicate in which directions the dataset contains the most variance. The eigenvalues and eigenvectors are found by solving the following eigenvalue problem for the covariance matrix (Equation (7)):

$$\Sigma v = \lambda v, \tag{7}$$

In solving this equation, v is an eigenvector and λ is the eigenvalue corresponding to this eigenvector. Eigenvectors with large eigenvalues represent the highest variance in the dataset and are selected as principal components. These two steps form the basis of PCA and provide the ability to retain important information while effectively reducing the size of the dataset. The ranking and selection of the eigenvectors allow the identification of the components that best represent the unique structure of the dataset. In our experimental studies, we compared the accuracy rates obtained by applying PCA with different numbers of principal components. Thus, the optimal number of principal components to maximize the model performance was determined.

After the feature dimension reduction using PCA, the reduced feature sets are fused to form a unified feature vector for each data sample. This comprehensive feature vector encapsulates the essential characteristics of sign language gestures, capturing both spatial and temporal dynamics effectively. The unified feature vector is then used as input for the SVM for the final classification step.

3.6. Support Vector Machine (SVM)

In this study, Support Vector Machine (SVM) is used for feature classification. SVM is a supervised learning algorithm that determines the optimal hyperplane that separates classes in the feature space [67]. The basic principle of SVM is to find the hyperplane that separates data points into different classes and has the widest margin between these classes. The basic formula for the hyperplane is given in Equation (8):

$$w \cdot x + b = 0, \tag{8}$$

In these equations, w is the normal vector of the hyperplane, x is the feature vectors, and b is the bias.

During training, SVM uses a kernel function that can move the data into a higher-dimensional vector space for cases where the feature set is not linearly separable. In our study, the Radial Basis Function (RBF) kernel is particularly preferred. The RBF kernel is a function that smooths the Euclidean distance between two points of the feature vectors with an exponential function [68], expressed as in Equation (9):

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \tag{9}$$

In this formula, x_i and x_j are the feature vectors of the two data points and γ is the parameter that determines the degree of propagation of the kernel. The RBF kernel is characterized by its ability to handle non-linear relationships between class labels and features.

4. Experimental Studies

This study was carried out on a system with an Intel Core i5-8400 processor, 16 GB RAM, and 12 GB GeForce GTX 1080 Ti GPU, mainly using PyTorch, PyTorch-Lightning, and scikit-learn libraries. The experimental studies were carried out using the BosphorusSign22k-general dataset [44,45], and then the final proposed method was evaluated with the other three datasets. The parameters set for the deep learning models trained in the study are given in Table 2.

Table 2. Parameters used in training deep learning models.

Parameter	Value
Learning rate	0.001 (reduced by a factor of 0.1 every 15 epochs)
Optimization algorithm	Stochastic Gradient Descent
Momentum	0.9
Mini-batch size	8
Number of epochs	35
Data augmentation technique	Random rotation between −10 and +10 degrees
Frame selection for video data	Random selection of 32 frames

Table 2 details the key parameters used in our model's training, selected to optimize performance for our SLR study. The initial learning rate is set at 0.001, reducing by a factor of 0.1 every 15 epochs to allow for precise adjustments during training. We employ Stochastic Gradient Descent (SGD) as our optimization algorithm, facilitating efficient and effective training with a momentum set at 0.9 to minimize fluctuations and achieve faster convergence. A mini-batch size of 8 is chosen, optimizing memory usage and improving generalization, and the model is trained over 35 epochs to ensure thorough learning and performance saturation. Data augmentation with random rotation (between −10 and +10 degrees) strengthens the model's adaptation to different orientations. Lastly, selecting 32 random frames from video data increases training diversity by incorporating information from different timestamps.

4.1. Performance Evaluation Metrics

In this study, we utilize a range of essential metrics to evaluate our developed model's effectiveness. Below is an outline of these key metrics:

True Positives (TPs): Instances in which the model accurately identifies an example as belonging to a specific category.

False Positives (FPs): Occurrences where the model wrongly labels an example as part of a category it is not.

False Negatives (FNs): Cases where the model fails to recognize actual examples of a category, incorrectly labeling them as not belonging.

True Negatives (TNs): Instances where the model rightly concludes that an example is not part of a certain category.

To assess the model's overall effectiveness, we compute its accuracy, reflecting the proportion of correct predictions against the total number of predictions. Additionally, we examine precision and recall to gain insights into the model's classification abilities. Precision measures the fraction of correctly identified positive instances among all instances classified as positive for a category. Recall assesses the model's ability to correctly identify true positive instances. We also employ the F1-Score, the harmonic mean of precision and recall, providing a balance between these metrics. The formulas for these metrics are as follows:

$$\text{Accuracy} = \frac{\text{Total Correct Predictions}}{\text{Total Predictions}}, \tag{10}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{11}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{12}$$

$$\text{F1} - \text{Score} = \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

4.2. Classification Results of MultiChannel-MobileNetV2 Models

In the initial phase of this study, MultiChannel-MobileNetV2 models were trained using the obtained FINGER, MERGED-FINGER, and FD-FINGER data. Additionally, a

conventional MobileNetV2 model was trained using HANDS data to evaluate the advantages of visualizing finger data in separate channels. The training and testing accuracy graphs of these models are presented in Figure 8. The performance metrics of the models' test analyses are listed in Table 3.

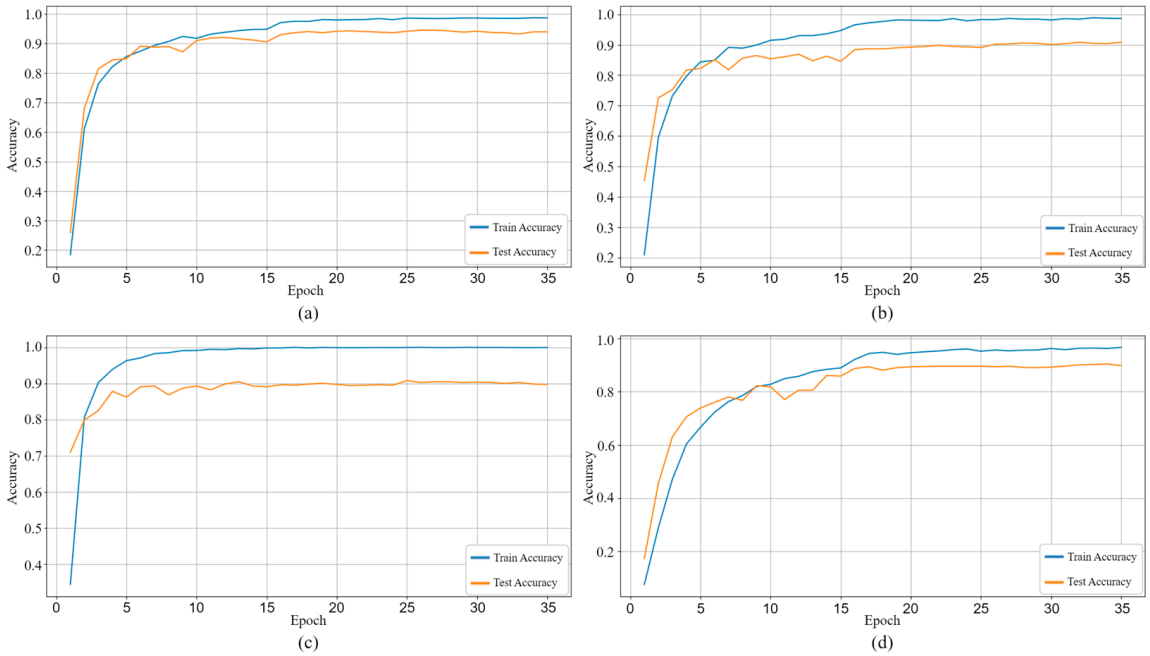

Figure 8. Training and test plots of the models (for BosphorusSign22k-general dataset), (**a**) FINGER-DATA, (**b**) MERGED-FINGER, (**c**) FD-FINGER, (**d**) HANDS (with MobileNetV2 model).

Table 3. Performance metrics of MultiChannel-MobileNetV2 and MobileNetV2 models trained on finger and hand data types (for BosphorusSign22k-general dataset).

Model	Data	Precision	Recall	F1-Score	Accuracy
MultiChannel-MobileNetV2	FINGER	93.94	94.49	93.40	94.52
MultiChannel-MobileNetV2	MERGED-FINGER	90.00	91.49	89.14	90.83
MultiChannel-MobileNetV2	FD-FINGER	90.26	91.67	89.73	90.83
MobileNetV2	HANDS	89.70	91.18	88.90	90.41

The MultiChannel-MobileNetV2 model has demonstrated remarkable success when trained with FINGER data. This model exhibited a precision of 93.94% and a recall of 94.49%, not only indicating accurate classifications but also showcasing a low number of missed true positives. This balance, when combined with an F1-Score of 93.40%, signifies the model's consistency and reliability. Moreover, an accuracy rate of 94.52% highlights the model's overall correctness, underscoring the critical importance of detailed examination of finger movements in SLR.

The model trained with MERGED-FINGER data, obtained by cropping regions containing fingers in FINGER data, achieved slightly lower values with 90.00% precision, 91.49% recall, 89.14% F1-Score, and 90.83% accuracy compared to the FINGER data. This can be attributed to the loss of complete positional information of fingers within the full image in MERGED-FINGER. However, a high accuracy rate of 90.83% indicates that this approach offers a valuable perspective for SLR, suggesting that a closer examination of fingers still provides significant information despite some loss of details.

The model trained with FD-FINGER data, which captures temporal changes in movement from FINGER data using the frame difference method, demonstrated robust values with 90.26% precision, 91.67% recall, 89.73% F1-Score, and 90.83% accuracy, successfully

capturing the dynamics of motion. Together with these results, the FD-FINGER data demonstrate the critical role of temporal information in SLR by effectively capturing the dynamics of finger gestures.

Lastly, the traditional MobileNetV2 model trained with HANDS data showed a slightly lower performance with an accuracy of 90.41% compared to other models. This outcome indicates that a detailed analysis of fingers provides more information and is more effective in SLR compared to general hand pose imagery.

Figure 9 illustrates a common challenge faced in SLR by visualizing situations where fingers overlap. This overlapping of fingers can make it difficult to recognize signs, especially when fingers cover each other accurately. This issue is also relevant for HANDS data derived through pose information, where finger overlap can obstruct a clear view. However, our FINGER data, achieved by visualizing each finger position in separate channels, present an effective solution to this overlapping issue. Additionally, the MERGED-FINGER data not only provide a closer examination of finger imagery but also distinctly separate the fingers of both hands, enabling a detailed analysis. These visual insights and the results from the corresponding data underscore the effectiveness of our proposed method in tackling challenges like finger overlap in SLR. The detailed analysis facilitated by the MultiChannel-MobileNetV2 model, which processes each finger in separate channels, significantly improves the accuracy and efficiency of SLR systems.

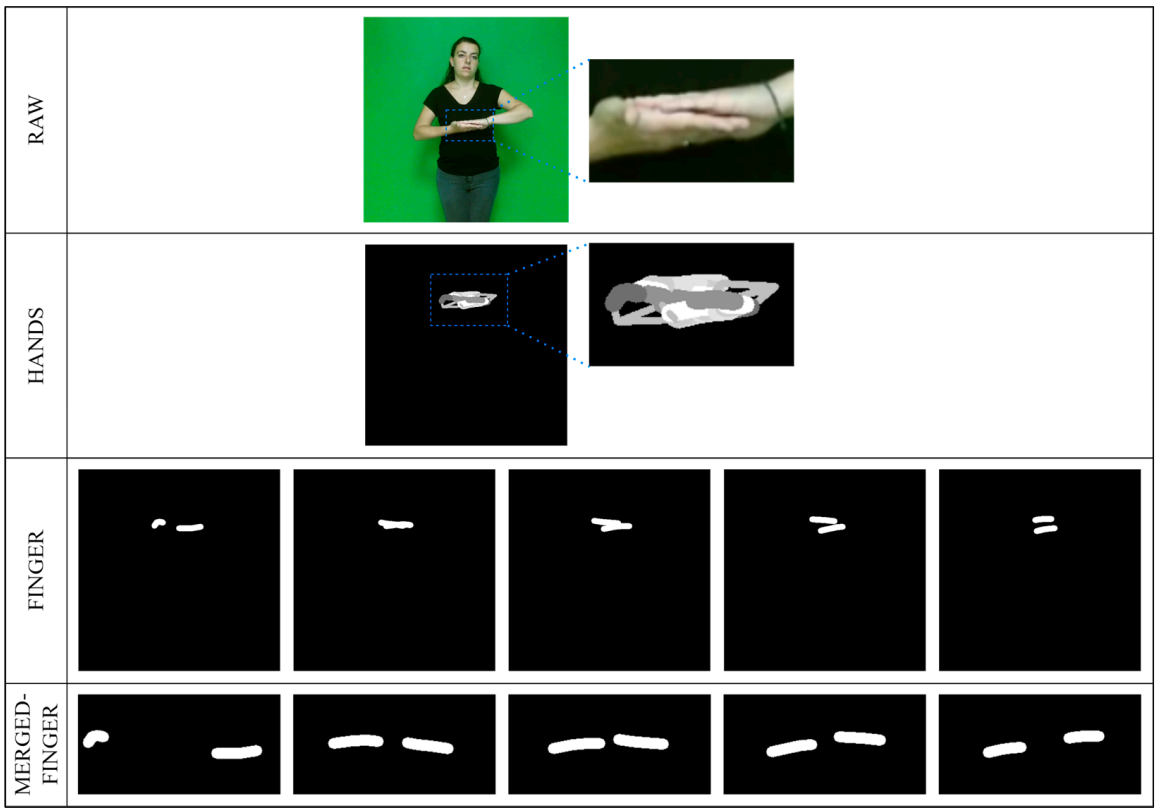

Figure 9. Demonstration of the problem of overlapping fingers (frame is from a video from the BosphorusSign22k dataset).

4.3. Fusion and Classification of Finger-Based Features

In this section, we aim to investigate the fusion of features obtained through trained MultiChannel-MobileNetV2 models and the impact of this combination on the overall accuracy of the model. Each MultiChannel-MobileNetV2 model produces a 6400-dimensional feature vector for each data point through its flattening layer. These vectors offer a rich representation reflecting the complex structure of the data. When we combine the feature vectors obtained from the FINGER, MERGED-FINGER, and FD-FINGER datasets,

a total feature vector of 19,200 dimensions per data point is obtained. However, this high dimensionality can pose challenges, particularly in terms of computational load and data management.

To address this issue, we plan to reduce the dimensionality of the feature vectors using the PCA method. PCA helps eliminate features that are redundant or contain little information while preserving the most significant aspects of the data. This characteristic of PCA not only shortens the training duration of the model but also enhances its overall performance and interpretability. For this purpose, we apply min–max normalization followed by PCA to the feature sets obtained from each model. The reduced feature sets post-PCA corresponding to each model are then combined and classified using the SVM classifier. Additionally, we repeat this application with different numbers of principal components in PCA to determine the optimal number of components. We also perform a similar classification process without applying PCA to observe its impact more distinctly. The results of the experimental study conducted with different principal component numbers in PCA are illustrated in the graph in Figure 10.

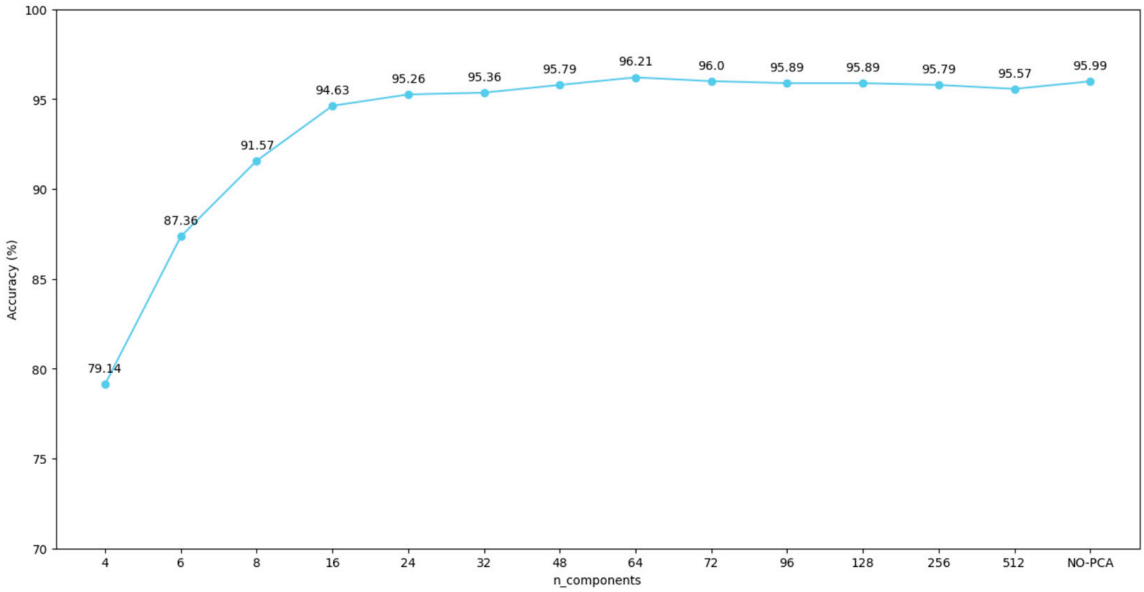

Figure 10. Accuracy values of the proposed method for FINGER-FEATURES for different numbers of principal components (for BosphorusSign22k-general dataset).

According to the data in the graph, the accuracy rate starting with four principal components (79.14%) significantly increased as the number of principal components rose. Notably, when using 16, 24, and 32 principal components, there were marked increases in accuracy rates (94.63%, 95.26%, and 95.36%, respectively). This indicates that PCA, while preserving the fundamental features of the dataset, filters out unnecessary information, thereby focusing on the model. A key observation is that the highest accuracy rate was achieved with 64 principal components (96.21%). However, beyond this point, despite an increase in the number of principal components, a decrease in accuracy rates was observed. For instance, the rates obtained with 72, 96, and 128 principal components were 96.00%, 95.89%, and 95.89%, respectively. This suggests that beyond a certain point, additional components do not contribute to, or may even detract from, the overall performance of the model. Another conclusion drawn from this analysis is that the performance achieved without using PCA (an accuracy rate of 95.99%) was lower than some results obtained with PCA. This demonstrates that using the entire feature set does not automatically increase accuracy. However, it is evident that when the correct number of principal components is selected, PCA can optimize the performance of the model and achieve higher accuracy rates. In our experiments, this optimal number of principal components was identified as 64, which will be utilized in future experimental studies.

FINGER, MERGED-FINGER, and FD-FINGER data reflect finger movements from different perspectives, and understanding the impact of each on the model's outcomes is crucial for enhancing the precision and efficiency of SLR systems. Following the achievement of 96.21% accuracy using the PCA method with 64 principal components, a fusion process of all possible combinations of feature sets was conducted to determine which sets most significantly improved model performance and which were less effective. The results of this analysis are presented in Table 4.

Table 4. Fusion of the finger features in different combinations and classifications (for BosphorusSign22k-general dataset).

FINGER	MERGED-FINGER	FD-FINGER	Precision	Recall	F1-Score	Accuracy
✓	✓		94.94	96.20	94.59	95.57
✓		✓	93.78	94.78	93.35	94.31
	✓	✓	94.98	95.89	94.67	95.57
✓	✓	✓	95.63	96.66	95.29	96.21

Note: ✓denotes data type included.

The results presented in Table 4 detail the impact of different feature combinations on the performance of SLR systems. Notably, the highest performance is achieved when all features are combined, with an accuracy of 96.21%. This underscores that integrating different features significantly enhances the model's precision and effectiveness. When examining the performance of individual feature sets, it is observed that the standalone use of the FINGER data yields the highest performance, with an accuracy of 94.52%. However, these individual performances are lower than the accuracy rates achieved when features are combined. For instance, the accuracy rate obtained by combining FINGER and MERGED-FINGER data (95.57%) is higher than when each data type is used independently. This illustrates how combining each type of data enhances the overall performance of the model, surpassing the information provided by each dataset individually.

4.4. Classification Performance of BODY and FACE Features

A robust SLR system should encompass a detailed analysis of not only the fingers but also all components that constitute the sign. Therefore, in the continuation of our study, we integrate body and face features into the system developed with finger features. To achieve this, we first use the BODY data, which are full-body pose images, and the FACE data, which are face pose images, obtained during the data preprocessing stage. In addition to these, we also incorporate FD-BODY and FD-FACE data, obtained by applying the frame difference method, into the system. All these datasets have been used in training separate MobileNetV2 models for each. The training and testing accuracy graphs obtained as a result of this training are presented in Figure 11, and the performance metric results are shown in Table 5.

Table 5. Performance metrics of MobileNetV2 models trained on body and face data types (for BosphorusSign22k-general dataset).

Model	Data	Precision	Recall	F1-Score	Accuracy
MobileNetV2	BODY	79.69	82.81	78.44	80.08
MobileNetV2	FD-BODY	84.40	86.33	83.23	85.14
MobileNetV2	FACE	16.32	16.91	13.04	15.70
MobileNetV2	FD-FACE	15.11	15.82	12.49	15.48

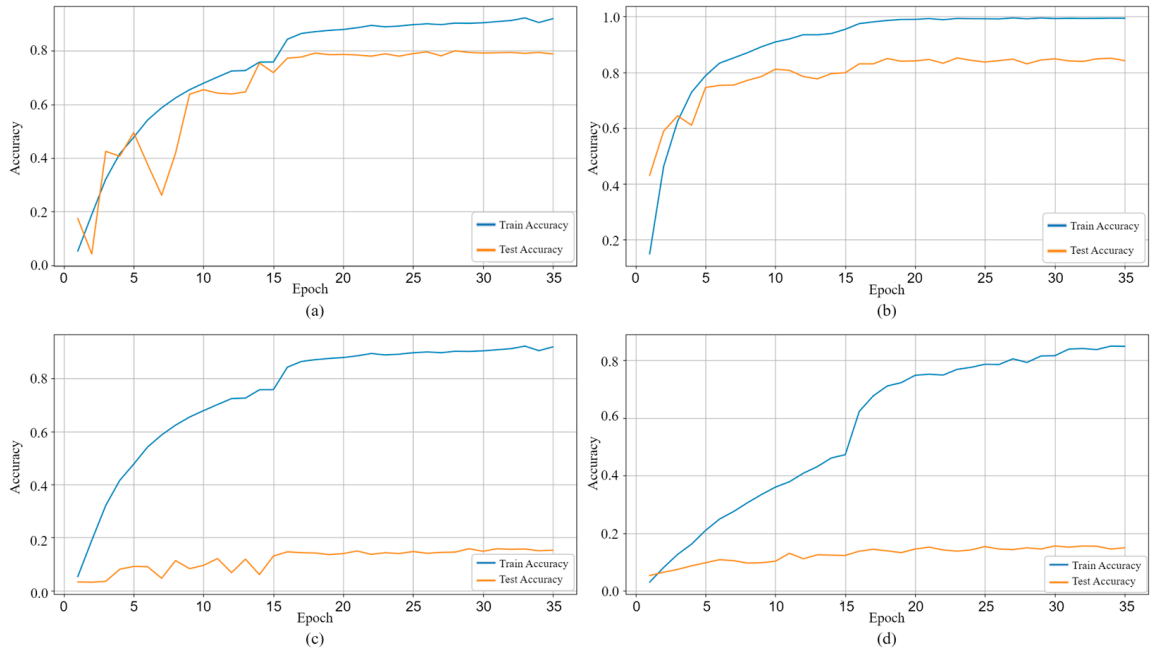

Figure 11. Training and test plots of the models (for BosphorusSign22k-general dataset): (**a**) BODY, (**b**) FD-BODY, (**c**) FACE, (**d**) FD-FACE.

Table 5 displays the performance of the MobileNetV2 model on BODY, FD-BODY, FACE, and FD-FACE data. According to the results, the BODY dataset was recognized by the model with an accuracy rate of 80.08%. The FD-BODY dataset, after the application of the frame difference method, exhibited a higher performance with an accuracy rate of 85.14%. These outcomes indicate that features related to the body are important components in SLR and that the models can effectively process these data. Additionally, the FD-BODY data obtained through the frame difference method demonstrate an enhanced ability to capture the dynamics and changes in movement, thereby improving recognition capabilities. On the other hand, the performance of the FACE and FD-FACE datasets remained quite low (accuracy rates of 15.70% and 15.48%, respectively). However, these low scores suggest that facial features serve more as complementary elements to the meaning of the sign rather than representing the sign independently. When combined with other features, they are expected to enhance the overall accuracy.

4.5. Fusion and Classification of All Features

At this stage of the study, features derived from BODY and FD-BODY data, as well as FACE and FD-FACE data, from the trained models were reduced using the PCA method (n_components = 64). These reduced features were then classified using SVM. Subsequently, all features related to fingers, body, and face were combined in all possible combinations and reclassified. At this stage, we label the combined data processed with PCA from the models as follows: FINGER-FEATURES (from FINGER, MERGED-FINGER, and FD-FINGER data), BODY-FEATURES (from BODY and FD-BODY data), and FACE-FEATURES (from FACE and FD-FACE data). The performance results obtained from classifying these different combinations using SVM are presented in Table 6.

According to the results in Table 6, the combination of BODY and FD-BODY features to create BODY-FEATURES (87.88% accuracy) and the combination of FACE and FD-FACE features to create FACE-FEATURES (17.07% accuracy) are noteworthy. The results obtained from both BODY-FEATURES and FACE-FEATURES demonstrate that combining spatial features with temporal features represented through the frame difference method enhances accuracy.

Table 6. Fusion of all features in different combinations and classifications (for BosphorusSign22k-general dataset).

FINGER-FEATURES	BODY-FEATURES	FACE-FEATURES	Precision	Recall	F1-Score	Accuracy
✓			95.63	96.66	95.29	96.21
	✓		87.38	90.66	87.03	87.88
		✓	16.73	20.34	14.78	17.07
✓	✓		95.80	96.77	95.54	96.31
✓		✓	96.21	96.97	95.92	96.73
	✓	✓	87.82	90.60	87.32	88.51
✓	✓	✓	96.65	97.29	96.45	97.15

Note: ✓denotes data type included.

The standalone use of the FINGER-FEATURES set exhibits a notably high performance, with an accuracy of 96.21%. However, it has been observed that the combination of FINGER-FEATURES and FACE-FEATURES achieves a higher accuracy rate (96.73%) compared to the combination of FINGER-FEATURES and BODY-FEATURES. This suggests that, although facial expressions alone may be insufficient for representing signs, they have the ability to convey emotional tones and subtle nuances in sign language. Integrating these features enhances the sensitivity and detailed understanding of SLR systems.

The highest recognition accuracy of 97.15% was achieved by combining all feature sets (FINGER-FEATURES, BODY-FEATURES, and FACE-FEATURES). This result proves how essential the integration of different features is for implementing a robust SLR system.

4.6. Time and Cost Analysis of the Proposed Method

In this section, time and cost analyses of the proposed SLR methods (FINGER-FEATURES and ALL-FEATURES) have been conducted. The analyses were carried out using the BosphorusSign22k-general test dataset. This test set encompasses a total of 949 videos, with the total duration of the videos being 2623 s and the number of frames being 78,684. This indicates that the average duration of a video is approximately 2.76 s, and the average number of frames per video is 82.

The Real-Time Factor (RTF), a critical criterion for evaluating real-time performance, has been used. RTF is calculated by dividing the total processing time by the actual duration of the sign language video. A lower RTF indicates a system capable of processing videos required for uninterrupted communication in real time or faster. Upon analyzing the test dataset, it is determined that the FINGER-FEATURES and ALL-FEATURES approaches have average RTF values of 2.19 and 2.47, respectively. These values, especially for the ALL-FEATURES approach, indicate that the data processing time is approximately 2.47 times greater than the data duration; this suggests that the method may require optimization for real-time processes. The RTF value of the FINGER-FEATURES approach is lower than that of ALL-FEATURES, indicating that it is slightly more advantageous in terms of real-time performance. However, considering the requirements for real-time operation, ideally, the RTF value should be close to 1 or lower. In this context, it is clear that both approaches need improvements in their real-time performances.

In the next phase of our research, to identify potential areas for improvement, a comprehensive temporal analysis was conducted for each of the SLR processes (pose extraction, pose visualization, and classification). This analysis was carried out across our entire test dataset, and the results were averaged. Thus, the results can be considered to have been obtained from a video of an average length of 2.76 s (the average duration per video). Additionally, a maximum memory cost analysis was also conducted using a sample equal to the average number of frames in our test set, which was 82 frames. Table 7 presents the detailed results of this analysis.

Table 7. Time and cost analysis results of the proposed methods.

Method	Pose Extraction (ms)	Pose Visualization (ms)	Classification Process (ms)	All Processes (ms)	Maximum Memory Usage (MB)
FINGER-FEATURES	4228	1744	63 (GPU)/ 274 (CPU)	6035 (GPU)/6246 (CPU)	1958
ALL-FEATURES	4237	2482	91 (GPU)/ 499 (CPU)	6810 (GPU)/7218 (CPU)	2334

The detailed temporal analysis demonstrates that the feature extraction performance during the classification process of the proposed MultiChannel-MobileNetV2 model, used for finger-based features, is quite reasonable. Particularly, executing this phase on the GPU significantly reduces the duration of classification operations. This underlines the efficiency and practical applicability of our proposed model. However, the pose extraction and visualization stages account for a significant portion of the total processing time, and the costs associated with these stages play a pivotal role in the overall efficiency of the methods. This situation highlights the need for improvements and identifies opportunities for optimization that could potentially enhance performance.

The ALL-FEATURES approach, compared to FINGER-FEATURES, offers a broader set of information that includes data on fingers as well as bodies and faces. Thanks to this additional information, recognition accuracy significantly increases; however, this situation leads to an increase in processing time and memory usage. Nonetheless, the accuracy improvement achieved through the provided comprehensive information set justifies the higher resource utilization of this method.

The obtained maximum memory usage values can be considered reasonable for high-performance computers. However, in environments with limited memory capacity, such as mobile devices, managing these amounts can pose a challenge and necessitate the development of more effective memory management strategies.

These analyses have thoroughly evaluated the performance of the proposed SLR methods in terms of time and cost. The results indicate that both methods have their own advantages; however, they also reveal the need for careful optimization for real-time applications and environments with limited resources, thereby laying a solid foundation for future work.

4.7. Comparison with Other Studies

In this section, we go beyond the initial experimental work on the 174-class BosphorusSign22k-general dataset [44,45] and examine the performance of our proposed method on larger and more diverse datasets. In this context, the 744-class BosphorusSign22k [44,45], 64-class LSA64 [38], and 310-class GSL datasets [48] are used to evaluate the generalizability and cross-linguistic compatibility of our method.

Our comparisons with other studies focus on two main methods: the first one is based on the FINGER-FEATURES set, which is derived from finger data (FINGER, MERGED-FINGER and FD-FINGER); the second one is based on the ALL-FEATURES set, which integrates features derived from body (BODY, FD-BODY) and face (FACE, FD-FACE) data in addition to FINGER-FEATURES. The performance comparison of these two approaches with the existing literature is detailed in Tables 8–11.

Table 8. Comparison with results in the literature (BosphorusSign22k-general dataset).

References	Method	Input Data	Focus Areas	Acc. (%)
Kindiroglu et al. (2019) [25]	CNN	TAF	Full	81.58
Gündüz and Polat, (2021) [58]	Inception 3D, LSTM-RNN	RGB, pose, optical flow	Body, hand, face	89.35
Proposed method (FINGER-FEATURES)	MultiChannel-MobileNetV2	Pose	Finger	96.21
Proposed method (ALL-FEATURES)	MultiChannel-MobileNetV2, MobileNetV2	Pose	Finger, body, face	97.15

Table 8 compares the results of the studies performed on the BosphorusSign22k-general dataset. The method proposed by Gündüz and Polat (2021) [58], based on Inception 3D and LSTM-RNN models integrating body, hand, face, optical flow, and pose data, achieved an accuracy of 89.35% using these complex and multifaceted data. This method demonstrated the potential of integrating various models and data types in the field of SLR. In contrast, our proposed FINGER-FEATURES approach emphasizes the critical role of finger gestures in sign language understanding and achieves an accuracy of 96.21% by focusing only on these features. This result is a strong indication that in-depth analysis of finger features can significantly improve the accuracy of SLR. In particular, our proposed MultiChannel-MobileNetV2 model processes FINGER, MERGED-FINGER, and FD-FINGER features in separate channels, enabling detailed evaluation of each finger feature and thus achieving an accuracy rate that exceeds the performance of previous works in the literature. In addition, compared to the Inception 3D and LSTM-RNN models, which have the capacity to capture spatial and temporal information, the integration of features obtained by using a relatively simpler model such as MobileNetV2 on pose data (e.g., BODY) and features produced by the frame difference method applied to these data (e.g., FD-BODY) yielded a remarkable accuracy rate of 87.88%, indicating that the frame difference method is highly effective in detecting temporal features. The 97.15% accuracy rate achieved with our ALL-FEATURES approach using all features makes a significant contribution by further improving the accuracy of SLR recognition.

Table 9. Comparison with results in the literature (BosphorusSign22k dataset).

References	Method	Input Data	Focus Areas	Acc. (%)
Özdemir et al., 2020 [44]	MC3-18	RGB	Full	78.85
Özdemir et al., 2020 [44]	IDT	RGB	Full	88.53
Gökçe et al., 2020 [57]	MC3-18, score-level fusion	RGB	Body, hand, face	94.94
Sincan and Keles, 2021 [46]	ResNet-50, I3D	RGB, MHI	Full	94.83
Kındıroglu et al., 2023 [69]	Aligned temporal accumulative features, temporal transformer networks, MC3-18	RGB, pose	Full	94.90
Özdemir et al., 2023 [70]	ST-GCN, MC-LSTM	RGB, pose	Body, hand, face	92.58
Proposed method (FINGER-FEATURES)	MultiChannel-MobileNetV2	Pose	Finger	92.79
Proposed method (ALL-FEATURES)	MultiChannel-MobileNetV2, MobileNetV2	Pose	Finger, body, face	95.13

Table 9 shows the studies using the 744-class BosphorusSign22k dataset. Among the past highest performances, Gökçe et al. (2020) [57] and Kındıroglu et al. (2023) [69] utilized the MC3-18 model, a powerful model capable of processing spatial and temporal

information, while Sincan and Keles (2021) [46] benefited from the I3D model. In our proposed method, the relatively simpler MobileNetV2 and MultiChannel-MobileNetV2 models are used, and temporal information is obtained from video data using the frame difference method. The highest recognition accuracy of 94.94% obtained by Gökçe et al. (2020) [57] with the method including body, hand, and face components was exceeded with a value of 95.13% in our method including body, face, and finger data. In addition, the approach we developed based only on finger features achieved a competitive value with a recognition accuracy of 92.79%, surpassing many studies.

Table 10. Comparison with results in the literature (LSA64 dataset).

	References	Method	Input Data	Focus Areas	Acc. (%)
Ex-1	Ronchetti et al. (2016) [71]	BoW + SubCls	RGB	Hand	97.00
	Konstantinidis et al. (2018b) [72]	LSTM	Pose	Body, hands	98.09
	Konstantinidis et al. (2018a) [73]	VGG-16, LSTM	RGB, pose, optical flow	Body, hand, face	99.84
	Zhang and Li (2019) [74]	MEMP network	RGB	Full	99.063
	Imran and Raman (2020) [75]	CNN, kernel-based extreme learning machine	MHI, dynamic image, RGBMI	Full	97.81
	Marais et al. (2022) [76]	Pruned VGG	RGB	Full	95.50
	Bohacek and Hruz (2022) [77]	Transformer	Pose	Body, hand	100
	Alyami et al. (2023) [78]	Transformer	Pose	Hand, face	98.25
	Proposed method (FINGER-FEATURES)	MultiChannel-MobileNetV2	Pose	Finger	99.09
	Proposed method (ALL-FEATURES)	MultiChannel-MobileNetV2, MobileNetV2	Pose	Finger, body, face	99.78
Ex-2	Marais et al. (2022) [79]	InceptionV3-GRU	RGB	Full	74.22
	Proposed method (FINGER-FEATURES)	MultiChannel-MobileNetV2	Pose	Finger	95.93
	Proposed method (ALL-FEATURES)	MultiChannel-MobileNetV2, MobileNetV2	Pose	Finger, body, face	97.96
Ex-3	Ronchetti et al. (2016) [71]	BoW + SubCls	RGB	Hand	91.70
	Rodríguez and Martínez (2018) [80]	Cumulative SD-VLAD with SVM	RGB	Full	85.00
	Alyami et al. (2023) [78]	Transformer	Pose	Hand, face	91.09
	Proposed method (FINGER-FEATURES)	MultiChannel-MobileNetV2	Pose	Finger	97.59
	Proposed method (ALL-FEATURES)	MultiChannel-MobileNetV2, MobileNetV2	Pose	Finger, body, face	98.93

Table 10 presents the studies conducted on the LSA64 dataset containing 64 sign classes. In this evaluation, the results for different training and test partitions of the LSA64 dataset are analyzed. In experiment 1, the dataset was randomly split into 80% training and 20% testing, and a 5-fold cross-validation method was applied to this configuration. This allowed for a more reliable assessment of the overall performance and robustness of our model. The average accuracy of 99.78% obtained in experiment 1 is a competitive result, very close to the most recent work in the literature. However, it is important to note that the results from the experiment 1 set are completely randomized in terms of the distribution of training and test data, and the randomness may vary across all the studies compared.

Table 11. Comparison with results in the literature (GSL dataset).

References	Method	Input Data	Focus Areas	Acc. (%)
Adaloglou et al. (2020) [48]	GoogLeNet + TConvs	RGB	Full	86.03
Adaloglou et al. (2020) [48]	3D-ResNet + BLSTM	RGB	Full	86.23
Adaloglou et al. (2020) [48]	I3D + BLSTM	RGB	Full	89.74
Selvaraj et al. (2021) [56]	LSTM	Pose	Full	86.60
Selvaraj et al. (2021) [56]	Transformer	Pose	Full	89.50
Selvaraj et al. (2021) [56]	ST-GCN	Pose	Full	93.50
Selvaraj et al. (2021) [56]	SL-GCN	Pose	Full	95.40
Fang et al. (2023) [81]	Adversarial multi-task learning	RGB, pose	Full	91.49
Proposed method (FINGER-FEATURES)	MultiChannel-MobileNetV2	Pose	Finger	94.05
Proposed method (ALL-FEATURES)	MultiChannel-MobileNetV2, MobileNetV2	Pose	Finger, body, face	95.37

Experiments 2 and 3 involve special training and test splits for signer-independent evaluation in our work on the LSA64 dataset. In experiment 2, the fifth and tenth of the ten signers in the dataset are specifically allocated as the test set. Experiment 3 represents an approach where one of the ten signers is designated as the test set, and the remaining nine are used as the training set. This process was repeated to evaluate the generalization ability of the model against all signers, with each signer being allocated as a test set in turn. The values in the table represent average accuracies. For both experiments, the results obtained with our FINGER-FEATURES and ALL-FEATURES approaches outperformed all the work in the literature. For experiment 2, accuracy rates of 95.93% and 97.96% were obtained with FINGER-FEATURES and ALL-FEATURES, respectively. These results demonstrate the superior performance and adaptability of our model. For experiment 3, 97.59% and 98.93% accuracy rates were obtained with FINGER-FEATURES and ALL-FEATURES, respectively, proving that the generalization capability of our model and its capacity to recognize sign languages produced by different signers are quite high.

Finally, in order to verify the applicability of our proposed method to different sign languages, we evaluated it on the GSL dataset, which contains 310 sign words. The results of this evaluation are presented in Table 11. In the study by Adaloglou et al. (2020) [48], methods such as GoogLeNet + TConvs [49] (86.03%), 3D-ResNet + BLSTM [50] (86.23%), and I3D + BLSTM [82] (89.74%) that can evaluate both spatial and temporal information were used on this dataset and achieved the mentioned accuracy rates. In contrast, our proposed method achieved a remarkable recognition accuracy of 94.05%, with the model based only on finger features. This result surpasses many studies in the literature and shows how effective it is to process the data obtained by visualizing finger positions in different channels with our MultiChannel-MobileNetV2 model. The recognition accuracy of 95.37% obtained with the ALL-FEATURES model closely approaches the highest reported accuracy rate in the literature of 95.40%, found in the work by Selvaraj et al. (2021) [56], and surpasses all other studies. This achievement demonstrates that our proposed system can provide robust and reliable recognition performance in the field of SLR by effectively utilizing both spatial and temporal information.

In summary, we evaluated our proposed method on various datasets with 64, 174, 310, and 744 sign classes. When examining the data from Tables 8–11, it is evident that our methods provide superior and competitive results compared to existing methods in the literature. Particularly, our multichannel approach using only finger-related features (FINGER-FEATURES) has achieved notable performance. The integration of features related to the face (FACE-FEATURES) and body (BODY-FEATURES) further enhanced this performance. The high recognition rates obtained from all datasets demonstrate that our proposed methodologies are reliable and robust across different datasets.

5. Conclusions

In this study, a multi-stream approach to the SLR process has been developed. Focusing particularly on finger movements, we have developed three detailed feature sets, FINGER, MERGED-FINGER, and FD-FINGER, where the data for each finger is visualized in a separate channel. This approach has facilitated a better understanding of the intricate details of sign language. Our developed MultiChannel-MobileNetV2 model has significantly improved accuracy and precision in the SLR field by effectively processing these multichannel features. Moreover, by expanding our methodology to include body and facial data, we have emphasized the importance of recognizing non-manual expressions, further enhancing the accuracy of the SLR process. This integrated approach, following the dimensionality reduction of features via PCA and their fusion and classification through SVM, presents an innovative SLR system that maximizes the comprehensiveness and accuracy of the SLR process.

Our study has made significant advancements in the field of SLR, yet it has some limitations. Firstly, we addressed the diversity of sign language by using datasets of different class sizes and languages. However, the focus of these datasets on specific languages limits the generalizability of our model across different sign languages. Considering the cultural and geographical diversity of sign language, our model requires additional datasets encompassing a wider range of languages to achieve more comprehensive access. Moreover, the chosen datasets might not be extensive enough to cover all gestures and expressions of sign language, which could limit the universal applicability of our model. Therefore, integrating datasets containing various languages and dialects in future studies is crucial for the model to appeal to a broader and more diverse user group.

While our study meticulously describes finger-based features, it has not directly examined the potential effects of individual differences, such as hand sizes and finger lengths, on the performance of our model. Sign language includes a rich array of gestures and expressions that can be influenced by physical differences among individuals. Therefore, future research should more thoroughly investigate the impact of various hand sizes and finger lengths on the accuracy and general applicability of our model. This could enhance the suitability of our model for a broader user base and strengthen the universality of SLR systems.

Additionally, evaluations of our system's real-time processing capability have shown that the RTF values are far from the ideal, being close to or less than 1. This situation underscores that our system has not yet reached the performance levels required for real-time applications and highlights the need for optimization, especially on devices with limited resources. Our future work will focus on improving the real-time performance of our model and developing more efficient memory management strategies, thereby expanding the applicability of SLR systems to a broader usage area.

In summary, although this study has taken innovative steps in the field of SLR technology and achieved significant successes, there are areas that future research needs to focus on. Enhancing the generalizability of the model across different sign languages and dialects, improving sensitivity to individual physical variations, and optimizing real-time processing capabilities are at the forefront of these areas. Improvements in these directions will enable SLR systems to reach a broader user base and significantly expand the communication possibilities of sign language users within society.

Author Contributions: Conceptualization, A.A. and O.K.B.; methodology, A.A. and O.K.B.; software, A.A. and O.K.B.; validation, A.A. and O.K.B.; formal analysis, A.A. and O.K.B.; investigation, A.A. and O.K.B.; writing—original draft preparation, A.A. and O.K.B.; writing—review and editing, A.A. and O.K.B.; visualization, A.A.; supervision, O.K.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: The identifiable human images employed in this study are sourced from a publicly accessible dataset. The dataset provider has supplied the necessary written informed consents, obtained from the individuals depicted, for the use of these images.

Data Availability Statement: The BosphorusSign22k-general and BosphorusSign22k datasets [44,45] used in this study can be obtained from the dataset creators upon reasonable request. The dataset creators can be contacted for access at the following link: https://ogulcanozdemir.github.io/bosphorussign22k/ (accessed on 14 November 2023). The LSA64 dataset [38] is available at this link: https://facundoq.github.io/datasets/lsa64/ (accessed on 21 December 2023). The GSL dataset [48] can be obtained from this link: https://vcl.iti.gr/dataset/gsl/ (accessed on 15 January 2024).

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

References

1. International Day of Sign Languages. Available online: https://www.un.org/en/observances/sign-languages-day (accessed on 10 January 2024).
2. Wadhawan, A.; Kumar, P. Sign Language Recognition Systems: A Decade Systematic Literature Review. *Arch. Comput. Methods Eng.* **2021**, *28*, 785–813. [CrossRef]
3. Nimisha, K.P.; Jacob, A. A Brief Review of the Recent Trends in Sign Language Recognition. In Proceedings of the 2020 IEEE International Conference on Communication and Signal Processing, ICCSP 2020, Chennai, India, 28–30 July 2020.
4. Kanwal, K.; Abdullah, S.; Ahmed, Y.B.; Saher, Y.; Jafri, A.R. Assistive Glove for Pakistani Sign Language Translation Pakistani Sign Language Translator. In Proceedings of the 17th IEEE International Multi Topic Conference: Collaborative and Sustainable Development of Technologies, IEEE INMIC 2014—Proceedings, Karachi, Pakistan, 8–10 December 2014.
5. Praveen, N.; Karanth, N.; Megha, M.S. Sign Language Interpreter Using a Smart Glove. In Proceedings of the 2014 International Conference on Advances in Electronics, Computers and Communications, ICAECC 2014, Bangalore, India, 10–11 October 2014.
6. Sadek, M.I.; Mikhael, M.N.; Mansour, H.A. A New Approach for Designing a Smart Glove for Arabic Sign Language Recognition System Based on the Statistical Analysis of the Sign Language. In Proceedings of the National Radio Science Conference, NRSC, Proceedings, Alexandria, Egypt, 13–16 March 2017.
7. 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]
8. Venugopalan, A.; Reghunadhan, R. Applying Deep Neural Networks for the Automatic Recognition of Sign Language Words: A Communication Aid to Deaf Agriculturists. *Expert Syst. Appl.* **2021**, *185*, 115601. [CrossRef]
9. Sarhan, N.; Frintrop, S. Unraveling a Decade: A Comprehensive Survey on Isolated Sign Language Recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 3210–3219.
10. Miozzo, M.; Peressotti, F. How the Hand Has Shaped Sign Languages. *Sci. Rep.* **2022**, *12*, 11980. [CrossRef] [PubMed]
11. Munib, Q.; Habeeb, M.; Takruri, B.; Al-Malik, H.A. American Sign Language (ASL) Recognition Based on Hough Transform and Neural Networks. *Expert Syst. Appl.* **2007**, *32*, 24–37. [CrossRef]
12. Kishore, P.V.V.; Prasad, M.V.D.; Kumar, D.A.; Sastry, A.S.C.S. Optical Flow Hand Tracking and Active Contour Hand Shape Features for Continuous Sign Language Recognition with Artificial Neural Networks. In Proceedings of the Proceedings—6th International Advanced Computing Conference, IACC 2016, Bhimavaram, India, 27–28 February 2016.
13. Hrúz, M.; Trojanová, J.; Železný, M. Local Binary Pattern Based Features for Sign Language Recognition. *Pattern Recognit. Image Anal.* **2012**, *22*, 519–526. [CrossRef]
14. Aly, S.; Mohammed, S. Arabic Sign Language Recognition Using Spatio-Temporal Local Binary Patterns and Support Vector Machine. In *Communications in Computer and Information Science*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 488.
15. Uddin, A.; Chowdhury, S.A. Hand Sign Language Recognition for Bangla Alphabet Using Support Vector Machine. In Proceedings of the 2016 International Conference on Innovations in Science, Engineering and Technology, ICISET 2016, Dhaka, Bangladesh, 28–29 October 2016.
16. Ben Jmaa, A.; Mahdi, W.; Ben Jemaa, Y.; Ben Hamadou, A. Arabic Sign Language Recognition Based on HOG Descriptor. In Proceedings of the Eighth International Conference on Graphic and Image Processing (ICGIP 2016), Tokyo, Japan, 29–31 October 2016; Volume 10225.
17. Mahmud, I.; Tabassum, T.; Uddin, M.P.; Ali, E.; Nitu, A.M.; Afjal, M.I. Efficient Noise Reduction and HOG Feature Extraction for Sign Language Recognition. In Proceedings of the 2018 International Conference on Advancement in Electrical and Electronic Engineering, ICAEEE 2018, Gazipur, Bangladesh, 22–24 November 2018.
18. Yasir, F.; Prasad, P.W.C.; Alsadoon, A.; Elchouemi, A. SIFT Based Approach on Bangla Sign Language Recognition. In Proceedings of the 2015 IEEE 8th International Workshop on Computational Intelligence and Applications, IWCIA 2015—Proceedings, Hiroshima, Japan, 6–7 November 2015.
19. Tharwat, A.; Gaber, T.; Hassanien, A.E.; Shahin, M.K.; Refaat, B. Sift-Based Arabic Sign Language Recognition System. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 334. [CrossRef]
20. Yang, Q.; Peng, J.Y. Chinese Sign Language Recognition Method Based on Depth Image Information and SURF-BoW. *Moshi Shibie Yu Rengong Zhineng/Pattern Recognit. Artif. Intell.* **2014**, *27*, 741–749.

21. Lim, K.M.; Tan, A.W.C.; Tan, S.C. Block-Based Histogram of Optical Flow for Isolated Sign Language Recognition. *J. Vis. Commun. Image Represent.* **2016**, *40*, 538–545. [CrossRef]
22. Jangyodsuk, P.; Conly, C.; Athitsos, V. Sign Language Recognition Using Dynamic Time Warping and Hand Shape Distance Based on Histogram of Oriented Gradient Features. In *Proceedings of the Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*; Association for Computing Machinery: New York, NY, USA, 2014.
23. Fagiani, M.; Principi, E.; Squartini, S.; Piazza, F. Signer Independent Isolated Italian Sign Recognition Based on Hidden Markov Models. *Pattern Anal. Appl.* **2015**, *18*, 385–402. [CrossRef]
24. Yang, W.; Tao, J.; Xi, C.; Ye, Z. Sign Language Recognition System Based on Weighted Hidden Markov Model. In Proceedings of the Proceedings—2015 8th International Symposium on Computational Intelligence and Design, ISCID 2015, Hangzhou, China, 12–13 December 2015; Volume 2.
25. Kindiroglu, A.A.; Ozdemir, O.; Akarun, L. Temporal Accumulative Features for Sign Language Recognition. In Proceedings of the Proceedings—2019 International Conference on Computer Vision Workshop, ICCVW 2019, Seoul, Republic of Korea, 27–28 October 2019.
26. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems*; Pereira, F., Burges, C.J., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25.
27. 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. In *Research Letters in the Information and Mathematical Sciences*; Massey University: Palmerston North, New Zealand, 2011; Volume 15.
28. 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]
29. Rahim, M.A.; Shin, J.; Yun, K.S. Hand Gesture-Based Sign Alphabet Recognition and Sentence Interpretation Using a Convolutional Neural Network. *Ann. Emerg. Technol. Comput.* **2020**, *4*, 20–27. [CrossRef]
30. Das, S.; Imtiaz, M.S.; Neom, N.H.; Siddique, N.; Wang, H. A Hybrid Approach for Bangla Sign Language Recognition Using Deep Transfer Learning Model with Random Forest Classifier. *Expert Syst. Appl.* **2023**, *213*, 118914. [CrossRef]
31. Aldhahri, E.; Aljuhani, R.; Alfaidi, A.; Alshehri, B.; Alwadei, H.; Aljojo, N.; Alshutayri, A.; Almazroi, A. Arabic Sign Language Recognition Using Convolutional Neural Network and MobileNet. *Arab. J. Sci. Eng.* **2023**, *48*, 2147–2154. [CrossRef]
32. Ma, Y.; Xu, T.; Kim, K. Two-Stream Mixed Convolutional Neural Network for American Sign Language Recognition. *Sensors* **2022**, *22*, 5959. [CrossRef]
33. Alsharif, B.; Altaher, A.S.; Altaher, A.; Ilyas, M.; Alalwany, E. Deep Learning Technology to Recognize American Sign Language Alphabet. *Sensors* **2023**, *23*, 7970. [CrossRef] [PubMed]
34. Elman, J.L. Finding Structure in Time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
35. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
36. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *2017*.
37. Masood, S.; Srivastava, A.; Thuwal, H.C.; Ahmad, M. Real-Time Sign Language Gesture (Word) Recognition from Video Sequences Using CNN and RNN. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 695.
38. Ronchetti, F.; Quiroga, F.; Lanzarini, L. LSA64: An Argentinian Sign Language Dataset. In Proceedings of the XXII Congreso Argentino de Ciencias de la Computación (CACIC), XIII Workshop on Databases and Data Mining (WBDMD), San Luis, Argentina, 3–7 October 2016; pp. 794–803, Red de Universidades con Carreras en Informática (RedUNCI).
39. Shin, J.; Musa Miah, A.S.; Hasan, M.A.M.; Hirooka, K.; Suzuki, K.; Lee, H.S.; Jang, S.W. Korean Sign Language Recognition Using Transformer-Based Deep Neural Network. *Appl. Sci.* **2023**, *13*, 3029. [CrossRef]
40. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231. [CrossRef]
41. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; Lecun, Y.; Paluri, M. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
42. Neto, G.M.R.; Junior, G.B.; de Almeida, J.D.S.; de Paiva, A.C. *Sign Language Recognition Based on 3D Convolutional Neural Networks*; Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2018; Volume 10882.
43. Wang, F.; Du, Y.; Wang, G.; Zeng, Z.; Zhao, L. (2+1)D-SLR: An Efficient Network for Video Sign Language Recognition. *Neural Comput. Appl.* **2021**, *34*, 2413–2423. [CrossRef]
44. Özdemir, O.; Kindiroglu, A.A.; Camgöz, N.C.; Akarun, L. BosphorusSign22k Sign Language Recognition Dataset. *arXiv* **2020**, arXiv:2004.01283.
45. Camgoz, N.C.; Kindiroglu, A.A.; Karabüklü, S.; Kelepir, M.; Sumru Ozsoy, A.; Akarun, L. BosphorusSign: A Turkish Sign Language Recognition Corpus in Health and Finance Domains. In Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016, Portorož, Slovenia, 23–28 May 2016.
46. Sincan, O.M.; Keles, H.Y. Using Motion History Images with 3D Convolutional Networks in Isolated Language Recognition. *arXiv* **2021**, arXiv:2110.12396.

47. Sincan, O.M.; Keles, H.Y. AUTSL: A Large Scale Multi-Modal Turkish Sign Language Dataset and Baseline Methods. *IEEE Access* **2020**, *8*, 181340–181355. [CrossRef]
48. Adaloglou, N.; Chatzis, T.; Papastratis, I.; Stergioulas, A.; Papadopoulos, G.T.; Zacharopoulou, V.; Xydopoulos, G.J.; Atzakas, K.; Daras, P. A Comprehensive Study on Sign Language Recognition Methods. *arXiv* **2020**, arXiv:2007.12530.
49. Cui, R.; Liu, H.; Zhang, C. A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training. *IEEE Trans. Multimed.* **2019**, *21*, 1880–1891. [CrossRef]
50. Pu, J.; Zhou, W.; Li, H. Iterative Alignment Network for Continuous Sign Language Recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; Volume 2019.
51. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; Volume 2016.
52. Samaan, G.H.; Wadie, A.R.; Attia, A.K.; Asaad, A.M.; Kamel, A.E.; Slim, S.O.; Abdallah, M.S.; Cho, Y.I. MediaPipe's Landmarks with RNN for Dynamic Sign Language Recognition. *Electronics* **2022**, *11*, 3228. [CrossRef]
53. Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E.; Hays, M.; Zhang, F.; Chang, C.-L.; Yong, M.G.; Lee, J.; et al. MediaPipe: A Framework for Building Perception Pipelines. *arXiv* **2019**, arXiv:1906.08172.
54. Podder, K.K.; Ezeddin, M.; Chowdhury, M.E.H.; Sumon, M.S.I.; Tahir, A.M.; Ayari, M.A.; Dutta, P.; Khandakar, A.; Mahbub, Z.B.; Kadir, M.A. Signer-Independent Arabic Sign Language Recognition System Using Deep Learning Model. *Sensors* **2023**, *23*, 7156. [CrossRef] [PubMed]
55. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph Neural Networks: A Review of Methods and Applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]
56. Selvaraj, P.; NC, G.; Kumar, P.; Khapra, M.M. OpenHands: Making Sign Language Recognition Accessible with Pose-Based Models across Languages. *arXiv* **2021**, arXiv:2110.05877.
57. Gökçe, Ç.; Özdemir, O.; Kındıroğlu, A.A.; Akarun, L. *Score-Level Multi Cue Fusion for Sign Language Recognition*; Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2020; Volume 12536.
58. Gündüz, C.; Polat, H. Turkish Sign Language Recognition Based on Multistream Data Fusion. *Turk. J. Electr. Eng. Comput. Sci.* **2021**, *29*, 1171–1186. [CrossRef]
59. Grishchenko, I.; Bazarevsky, V. MediaPipe Holistic—Simultaneous Face, Hand and Pose Prediction, on Device. Available online: https://ai.googleblog.com/2020/12/mediapipe-holistic-simultaneous-face.html (accessed on 11 January 2022).
60. Zhan, C.; Duan, X.; Xu, S.; Song, Z.; Luo, M. An Improved Moving Object Detection Algorithm Based on Frame Difference and Edge Detection. In Proceedings of the 4th International Conference on Image and Graphics, ICIG 2007, Xiamen, China, 5–7 July 2019.
61. Husein, A.M.; Calvin; Halim, D.; Leo, R.; William. Motion Detect Application with Frame Difference Method on a Surveillance Camera. *J. Phys. Conf. Ser.* **2019**, *1230*, 012017. [CrossRef]
62. Singla, N. Motion Detection Based on Frame Difference Method. *Int. J. Inf. Comput. Technol.* **2014**, *4*, 1559–1565.
63. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
64. Altman, N.; Krzywinski, M. The Curse(s) of Dimensionality This-Month. *Nat. Methods* **2018**, *15*, 399–400. [CrossRef]
65. Aremu, O.O.; Hyland-Wood, D.; McAree, P.R. A Machine Learning Approach to Circumventing the Curse of Dimensionality in Discontinuous Time Series Machine Data. *Reliab. Eng. Syst. Saf.* **2020**, *195*, 106706. [CrossRef]
66. Ringnér, M. What Is Principal Component Analysis? *Nat. Biotechnol.* **2008**, *26*, 303–304. [CrossRef] [PubMed]
67. Andrew, A.M. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. *Kybernetes* **2001**, *30*, 103–115. [CrossRef]
68. Amari, S.; Wu, S. Improving Support Vector Machine Classifiers by Modifying Kernel Functions. *Neural Netw.* **1999**, *12*, 783–789. [CrossRef] [PubMed]
69. Kındıroglu, A.A.; Özdemir, O.; Akarun, L. Aligning Accumulative Representations for Sign Language Recognition. *Mach. Vis. Appl.* **2023**, *34*, 12. [CrossRef]
70. Özdemir, O.; Baytaş, İ.M.; Akarun, L. Multi-Cue Temporal Modeling for Skeleton-Based Sign Language Recognition. *Front. Neurosci.* **2023**, *17*, 1148191. [CrossRef] [PubMed]
71. Ronchetti, F.; Quiroga, F.; Estrebou, C.; Lanzarini, L.; Rosete, A. *Sign Languague Recognition without Frame-Sequencing Constraints: A Proof of Concept on the Argentinian Sign Language*; Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2016; Volume 10022.
72. Konstantinidis, D.; Dimitropoulos, K.; Daras, P. Sign Language Recognition Based on Hand and Body Skeletal Data. In Proceedings of the 3DTV-Conference, Silja Serenade, Baltic Sea, 3–5 June 2018; Volume 2018.
73. Konstantinidis, D.; Dimitropoulos, K.; Daras, P. A Deep Learning Approach for Analyzing Video and Skeletal Features in Sign Language Recognition. In Proceedings of the IST 2018—IEEE International Conference on Imaging Systems and Techniques, Proceedings, Krakow, Poland, 16–18 October 2018.
74. Zhang, X.; Li, X. Dynamic Gesture Recognition Based on MEMP Network. *Future Internet* **2019**, *11*, 91. [CrossRef]

75. Imran, J.; Raman, B. Deep Motion Templates and Extreme Learning Machine for Sign Language Recognition. *Vis. Comput.* **2020**, *36*, 1233–1246. [CrossRef]
76. Marais, M.; Brown, D.; Connan, J.; Boby, A. An Evaluation of Hand-Based Algorithms for Sign Language Recognition. In Proceedings of the 2022 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), Durban, South Africa, 4–5 August 2022; pp. 1–6.
77. Bohacek, M.; Hruz, M. Sign Pose-Based Transformer for Word-Level Sign Language Recognition. In Proceedings of the Proceedings—2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, WACVW 2022, Waikoloa, HI, USA, 4–8 January 2022.
78. Alyami, S.; Luqman, H.; Hammoudeh, M. Isolated Arabic Sign Language Recognition Using A Transformer-Based Model and Landmark Keypoints. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2023**, *23*, 1–19. [CrossRef]
79. Marais, M.; Brown, D.; Connan, J.; Boby, A.; Kuhlane, L. Investigating Signer-Independent Sign Language Recognition on the LSA64 Dataset. In Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2022, George, South Africa, 28–31 August 2022.
80. Rodríguez, J.; Martínez, F. Towards On-Line Sign Language Recognition Using Cumulative SD-VLAD Descriptors. In *Communications in Computer and Information Science*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 885.
81. Fang, Y.; Xiao, Z.; Cai, S.; Ni, L. Adversarial Multi-Task Deep Learning for Signer-Independent Feature Representation. *Appl. Intell.* **2023**, *53*, 4380–4392. [CrossRef]
82. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In Proceedings of the Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017.