Multimodal Technologies and Interaction

MDPI

Brief Report

Real-Time Norwegian Sign Language Recognition Using MediaPipe and LSTM

Md. Zia Uddin [1], Costas Boletsis [1,*] and Pål Rudshavn [2]

[1] SINTEF Digital, 0373 Oslo, Norway; zia.uddin@sintef.no
[2] Statped, 7088 Heimdal, Norway; pal.rudshavn@statped.no
* Correspondence: konstantinos.boletsis@sintef.no

Abstract: The application of machine learning models for sign language recognition (SLR) is a well-researched topic. However, many existing SLR systems focus on widely used sign languages, e.g., American Sign Language, leaving other underrepresented sign languages such as Norwegian Sign Language (NSL) relatively underexplored. This work presents a preliminary system for recognizing NSL gestures, focusing on numbers 0 to 10. Mediapipe is used for feature extraction and Long Short-Term Memory (LSTM) networks for temporal modeling. This system achieves a testing accuracy of 95%, aligning with existing benchmarks and demonstrating its robustness to variations in signing styles, orientations, and speeds. While challenges such as data imbalance and misclassification of similar gestures (e.g., Signs 3 and 8) were observed, the results underscore the potential of our proposed approach. Future iterations of the system will prioritize expanding the dataset by including additional gestures and environmental variations as well as integrating additional modalities.

Keywords: artificial intelligence; long short-term memory; machine learning; Norwegian sign language; sign language recognition

Received: 6 December 2024
Revised: 25 February 2025
Accepted: 26 February 2025
Published: 3 March 2025

Citation: Uddin, M.Z.; Boletsis, C.; Rudshavn, P. Real-Time Norwegian Sign Language Recognition Using MediaPipe and LSTM. *Multimodal Technol. Interact.* **2025**, *9*, 23. https://doi.org/10.3390/mti9030023

1. Introduction

According to the World Health Organization, approximately 5% of the global population (430 million people) experience hearing loss, and this number is expected to increase in the coming years [1,2]. Of this population, 70 million people are completely deaf [2,3]. For deaf and hard-of-hearing individuals, communicating with the broader population can be challenging due to the limited number of people who are proficient in sign language. This communication barrier often leads to significant difficulties in daily life [2,4].

One potential and highly promising solution is the use of machine learning (ML) models to create systems capable of translating sign language into text or speech in real time. These systems can help to bridge the communication gap for the deaf and hard-of-hearing and to facilitate their better integration into society, and have led to several real-life applications [2,4]:

- Sign language-enabled virtual assistants, wearable devices for real-time translation, and mobile apps can provide dynamic translation between signers and non-signers, greatly improving day-to-day communication for the deaf community [5].
- Artificial intelligence (AI)-based sign language interpreters embedded into television broadcasts, websites, and social media platforms have the potential to provide accessible content in sign language [6].

- Sign language educational apps and tools that leverage AI can guide learners in mastering gestures, facial expressions, and other components of sign language, enhancing their learning experience and proficiency [7].

The application of ML models for sign language recognition (SLR) is a well-researched topic [2]. Recent works have combined Mediapipe for spatial feature extraction with Long Short-Term Memory (LSTM) networks for temporal modeling to recognize dynamic sign language gestures [8]. The integration of Mediapipe and LSTM in SLR systems addresses key challenges in lightweight and multimodal gesture recognition [8,9]. Mediapipe efficiently extracts spatial features without the computational burden of deep convolutional models, while LSTM processes the temporal dynamics of gestures, making this combination particularly effective for real-time applications on low-resource devices [8,9]. However, many existing SLR systems focus on widely used sign languages such as American Sign Language (ASL), while other underrepresented sign languages such as Norwegian Sign Language (NSL) remain underexplored [2,8].

In this work, we address this gap by applying Mediapipe and LSTM networks to video data as the main part of a broader methodology for implementing a lightweight and multimodal real-time recognition system for NSL.

The work presented in this brief-report article is at the feasibility stage, with the first preliminary step involving an investigation into the use of MediaPipe and LSTM networks with a limited vocabulary. This includes testing on a dataset of NSL educational videos provided by Statped (Statped's sign language webpage: https://www.statped.no/tegnsprak/, accessed on 5 December 2024) specifically for recognizing numerical gestures (numbers 0 through 10) in NSL and translating them into text. Focusing on numerical gestures as a first step allows the feasibility and performance of the proposed approach to be evaluated with minimal resources before expanding it to the broader complexity of NSL, while also addressing challenges such as gesture overlap and multimodal integration.

The ultimate intention of this work is to further contribute to the "Mediapipe/LSTM-based SLR" field, adding a small-scale yet scalable case in the literature while providing NSL researchers and SLR developers with a lightweight and multimodal AI-based recognition methodology that can also be applied to other underrepresented sign languages.

The rest of this paper is organized as follows: Section 2 provides an overview of the related literature; Section 3 presents the proposed approach for real-time NSL recognition; Section 4 describes the results of the proposed method; Section 5 discusses these results; finally, the conclusion to the paper is provided in with Section 6.

2. Background

The development of SLR systems has seen significant progress, with various machine learning (ML) and deep learning (DL) models applied to tasks ranging from static image recognition to dynamic gesture classification. Modeling approaches such as support vector machine (SVM), K-nearest neighbor (KNN), convolutional neural networks (CNN), and long short-term memory (LSTM) networks have been widely employed in this domain.

SVM and KNN were prominent in early SLR systems, especially for static gestures. SVM approaches are known for their robustness in handling high-dimensional data, and have shown success in classifying hand gestures in controlled environments [10,11]. Similarly, the simplicity and ability of KNN-based models to adapt to small datasets has led to their application in ASL recognition, achieving classification accuracy exceeding 90% [12,13]. However, these models struggle with scalability for large datasets and dynamic gestures, making them less suitable for real-time applications [14,15].

The CNN architecture has become the dominant one for gesture recognition due to its ability to extract hierarchical features from image data. CNNs are particularly effective at

handling complex image inputs, as demonstrated in ASL recognition systems, achieving near-perfect accuracy using large datasets and advanced preprocessing techniques [16–18]. However, CNNs are computationally intensive and require substantial hardware resources, which can hinder real-time applications in resource-constrained environments [19,20].

LSTM networks are widely used for sequential data processing, making them suitable for dynamic gesture recognition. By capturing temporal dependencies, LSTM networks can model the sequential nature of sign language gestures more effectively than models focused solely on static inputs [21–23]. However, LSTM-based systems alone often lack spatial feature extraction capabilities, which requires complementary models or preprocessing steps to enhance their performance [24–26]. This dependence increases computational overhead, challenging the suitability of LSTM for lightweight real-time applications.

In the conclusion of Dewanto et al.'s scoping review of SLR research [8], it was stated that "The integration of deep learning LSTM networks with the MediaPipe framework presents a powerful approach for sign language recognition. . . achieving high accuracy rates and demonstrating versatility across different sign languages." Mediapipe (Mediapipe by Google AI: https://github.com/google-ai-edge/mediapipe, accessed on 5 December 2024) is a lightweight real-time framework that uses machine learning to detect and track key landmarks such as hand and facial positions in video data. In the last three years, it has become a valid choice in SLR research for feature extraction in combination with LSTM networks for temporal modeling [8]. For example, Sundar and Bagyammal [9] utilized Mediapipe and LSTM to recognize ASL alphabets, achieving high accuracy by leveraging Mediapipe's real-time landmark detection and LSTM's temporal capabilities. Similarly, Khartheesvar et al. [27] developed a system for automatic Indian Sign Language recognition using Mediapipe Holistic and LSTM, demonstrating the efficiency of this combination in capturing multimodal features and sequential patterns. Rao et al. [28] applied this approach for real-time recognition of dynamic gestures, validating the suitability of Mediapipe and LSTM for lightweight applications. Nguyen et al. [29] also explored the integration of MediaPipe with LSTM layers for real-time sign language recognition, demonstrating the adaptability of this approach to diverse challenges and data types. In [30], Farhan and Madi developed an ASL recognition system that effectively utilized MediaPipe and LSTM to achieve high accuracy in real-time dynamic gesture recognition scenarios. Other studies have extended the same approach to regional sign languages such as Thai Sign Language; Jintanachaiwat et al. [31] used Mediapipe and LSTM to translate gestures into text in real time, achieving practical usability on resource-constrained devices.

Despite these advancements, NSL recognition poses unique challenges: (i) NSL gestures often include overlapping hand configurations, which are difficult to disambiguate without incorporating complementary non-manual signals such as facial expressions and mouth movements. (ii) Unlike ASL, NSL lacks large publicly available datasets, making it difficult to train robust models [2,32]. Svendsen and Kadry [2] addressed this gap by creating a custom dataset of 24,300 images of 27 NSL alphabet signs, which they used to evaluate the performance of the SVM, KNN, and CNN models. Their study found that CNN outperformed SVM and KNN for NSL recognition, particularly in handling diverse hand shapes and orientations. Their research has highlighted the need for more extensive studies on NSL recognition, particularly with respect to real-time applications and generalization across diverse users and environments.

To the best of our knowledge, this is the only SLR-related work on NSL to this day. The aim of the present work is to build upon the work of Svendsen and Kadry by focusing on video-based multimodal recognition of NSL, specifically through the application of Mediapipe and LSTM networks.

3. Method

The goal of this preliminary work is to achieve automatic recognition of NSL gestures for numbers ranging from 0 to 10. To accomplish this, a two-stage approach was implemented, combining feature extraction using Mediapipe with classification through LSTM networks. The video dataset was provided by Statped and features signing by NSL educators. These videos were originally recorded for a public NSL educational platform, ensuring adherence to ethical and privacy considerations. The dataset was curated to include diversity in signer characteristics, such as age, gender, and signing styles. A total of five participants performed gestures corresponding to numbers 0 through 10, with each gesture recorded multiple times under consistent lighting and framing conditions.

The dataset consists of 1059 short video samples, with the distribution of gestures as follows: Sign 0 has 101 videos, Sign 1 has 102 videos, and Sign 2 has 103 videos; Signs 3, 4, and 10 each have 100 videos, while Sign 5 has 102 videos. Signs 6 and 7 include 94 and 98 videos, respectively. Signs 8 and 9 have the smallest representation, with 79 and 80 videos. Despite slight variations in counts across categories, this dataset offers a diverse and representative collection of NSL gestures.

To prepare the videos for analysis, they were segmented into individual frames while maintaining a consistent frame rate to preserve the temporal dynamics of the gestures. From each short video, 20 key frames were selected for feature extraction and classification. Each key frame was determined by identifying the shortest distance between the mouth markers and the dominant hand markers, as the primary gesture for these 11 signs was captured in such frames. This approach ensured the uniformity and comparability of the data regardless of variations in video length or frame rate.

Figure 1 illustrates randomly selected gesture images with annotated hand and mouth markers, providing an overview of the dataset distribution and its key features.

Figure 1. Randomly selected key frames from the NSL gesture video dataset, showing annotated markers for hand and mouth positions across gestures representing numbers 0 through 10. These key frames highlight the spatial dynamics of the gestures used for feature extraction and classification.

Mediapipe was used to detect key landmarks on the active/dominant hand and around the mouth, as the dominant hand is essential for numerical gestures in NSL and the mouth provides complementary information for distinguishing similar hand shapes with varying lip movements (Figure 2). The hand detection module identified 21 key points, capturing the structures of the hand, finger joints, and palm, while the face detection module identified landmarks around the mouth. The (x, y, z) coordinates of these landmarks were normalized relative to the frame dimensions in order to account for variability in camera positioning and signer distance.

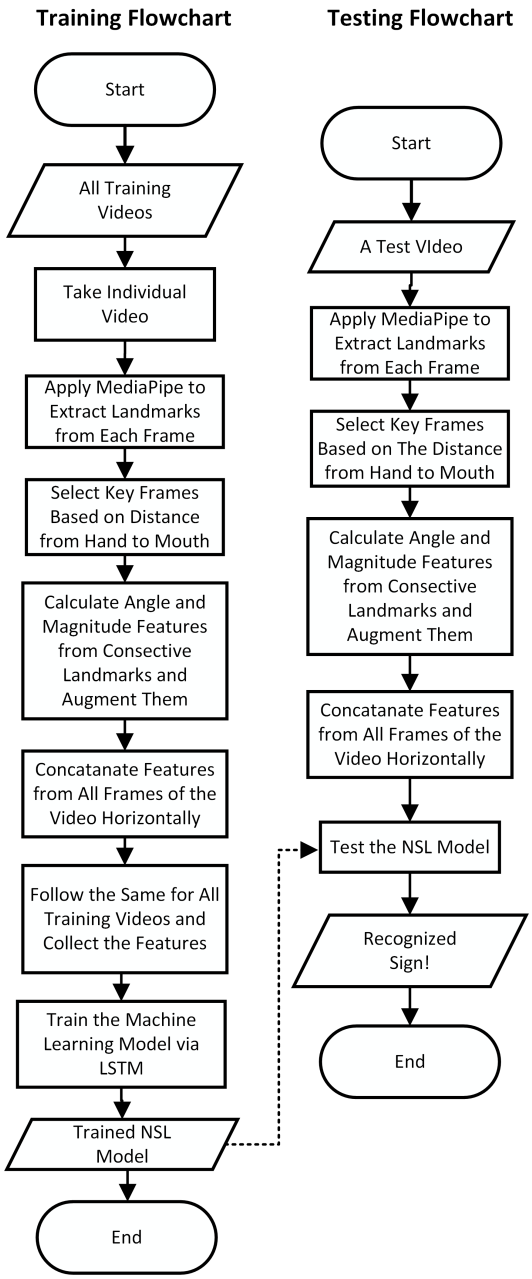

Figure 2. Flowcharts illustrating the training and testing pipeline for NSL gesture recognition. The process includes key frame selection and feature extraction via MediaPipe followed by gesture classification using LSTM networks.

Using Mediapipe and OpenCV, angle and magnitude features were extracted from the detected hand and lip landmarks to analyze their movements in videos. Mediapipe's Face Mesh and Hands models were initialized to detect landmarks corresponding to predefined

regions of interest. Angles between consecutive landmarks were calculated to represent orientation, while magnitudes captured spatial distances between points, as described in previous work [29].

These geometric features were extracted sequentially and combined for both hand and lip landmarks, providing a robust representation of spatial and directional information. This feature extraction method can effectively capture the dynamics of hand and lip movements, making it well suited for gesture recognition tasks.

The features were standardized to ensure uniform scaling and improve model stability and performance, following commonly accepted preprocessing practices in machine learning (cf. [33]). This transformation adjusted the feature values to have a mean of zero and a standard deviation of one, mitigating the influence of differing feature scales.

After standardization, the hand and mouth features were concatenated into a single vector for each frame, creating a comprehensive representation of the gestures. Twenty key frames were considered from each video in which the hand and face were nearby in order to represent the features. Thus, the feature size from each video became 20×102, where 102 represents 51 angles and the same amount of magnitude features. The feature processing time for each key frame was 44.92 ms on average. This step was critical for effectively distinguishing between similar gestures (e.g., Signs 3 and 8) by leveraging both spatial and directional information from the standardized data.

After completing the preprocessing stage, the standardized and concatenated feature vectors served as input to the LSTM-based classification model. The LSTM architecture used in this study consisted of two layers, each with 128 units, enabling the model to capture both short-term and long-term dependencies in the gesture sequences. Dropout layers were placed between the LSTM layers to prevent overfitting by randomly disabling a fraction of the neurons during training. The output layer was a fully connected dense layer with softmax activation, producing probabilities for each of the 11 gesture classes (numbers 0 through 10). This setup effectively handled the multiclass classification task.

To train the LSTM model, the dataset was divided into three subsets: 70% for training, 10% of the training for validation, and 30% for testing. The split was performed randomly, with a fixed random seed used to ensure reproducibility across experiments. This ensured a representative distribution of data across both subsets and avoided bias from uneven splits. The training set was used to teach the model, while the testing set was reserved for evaluating its generalization performance on unseen data.

The training process aimed to minimize the cross-entropy loss function, and the Adam optimizer was used to update model weights. Data augmentation techniques were incorporated to improve the model's ability to generalize to unseen signing variations. These augmentations introduced variability by applying transformations such as random scaling, rotation, and slight shifts to the hand coordinate data. These transformations simulated real-world variations in signing styles and environmental conditions, improving the model's robustness in diverse scenarios.

The model was trained for 200 epochs, allowing it to learn the underlying patterns in the data without overfitting. A batch size of 32 was chosen to balance computational efficiency and stable parameter updates. The learning rate was set to 0.001 to ensure stable convergence without overshooting the optimal solution. During training, metrics such as training accuracy and loss were monitored to evaluate the model's learning progress. This allowed us to identify potential issues such as overfitting or vanishing gradients in real time and make adjustments to the training process as necessary.

The final LSTM model consisted of three layers, as illustrated in Figure 3. The first layer was an LSTM with 20 units, producing a 3D output shape of (None, 20, 20) and containing 9840 trainable parameters. The second LSTM layer also had 20 units, producing

a 2D tensor output of shape (None, 20) with 3280 trainable parameters. The final dense layer contained 11 output units corresponding to the gesture classes, with 231 trainable parameters. The total number of trainable parameters in the model was 13,351, with no non-trainable parameters.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 20, 20)            9840
_________________________________________________________________
lstm_1 (LSTM)                (None, 20)                3280
_________________________________________________________________
dense (Dense)                (None, 11)                231
=================================================================
Total params: 13,351
Trainable params: 13,351
Non-trainable params: 0
_________________________________________________________________
```

Figure 3. Architecture of the final LSTM model used for NSL gesture recognition. The model comprises two LSTM layers, each with 20 units, followed by a dense layer with 11 units corresponding to the gesture classes. The total number of trainable parameters is 13,351, with no non-trainable parameters.

For more explanation, the LSTM model architecture with different gates is shown in Figure 4. There, the input data with a structural shape of (20, 102), representing 20 time steps and 102 features per step, enter the input layer and are passed to the first LSTM layer (LSTM 1) consisting of 20 units. This layer is configured with return sequences, ensuring that the entire sequence of hidden states is forwarded to the next layer. Within LSTM 1, the data flow through three critical gates: the input gate, which regulates how much new information enters the cell state; the forget gate, which decides what past information to discard; and the output gate, which controls the portion of the cell state that influences the hidden state passed forward. The processed sequence then feeds into the second LSTM layer (LSTM 2) with another 20 units. Unlike the first layer, this LSTM outputs only the final hidden state, as there are no return sequences. Similar to the first LSTM layer, data in LSTM 2 pass through this layer's input, forget, and output gates, refining the temporal features extracted from the sequence. Finally, the output from LSTM 2 is directed into a fully connected dense layer comprising 11 units with a softmax activation function. This layer converts the processed features into a probability distribution over 11 output classes, yielding the final classification result. This sequential data flow is guided by the gating mechanisms, allowing the network to effectively model long-term dependencies and complex temporal relationships in the input data.

Figure 5 illustrates the training loss and accuracy trends across epochs. The top plot shows a sharp decrease in training loss during the initial epochs followed by a plateau, indicating effective learning and convergence. The bottom plot shows that the training accuracy rapidly increases during early epochs and stabilizes near 100%, reflecting the model's ability to correctly predict the training data. These trends confirm that the model successfully minimized the loss function and improved accuracy without signs of overfitting or instability.

Evaluation was performed on the testing subset, which had not been seen during training. Accuracy was the primary evaluation metric, complemented by precision, recall, and F1-score to assess the model's ability to minimize false positives and false negatives. A confusion matrix was generated to analyze classification performance across all gesture classes, providing insights into specific gestures prone to misclassification.

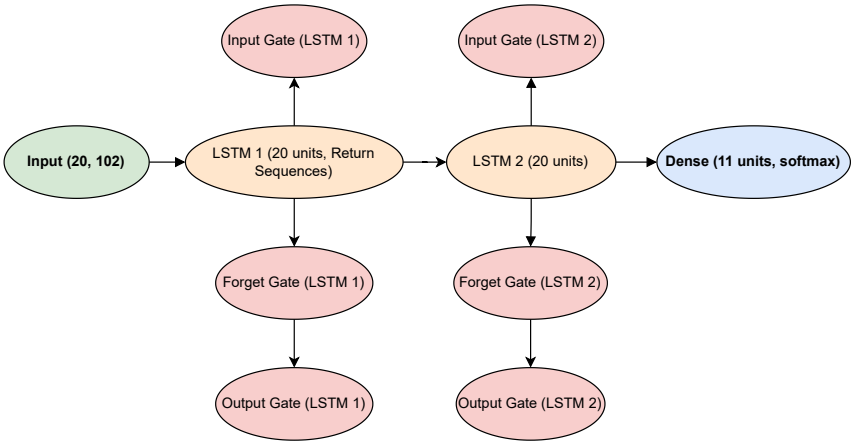

Figure 4. LSTM model architecture with different gates.

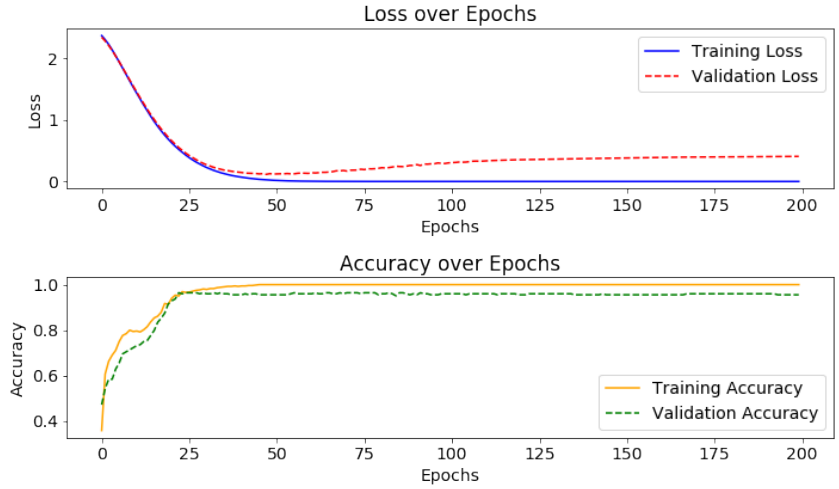

Figure 5. Training and validation loss and accuracy trends across epochs for the LSTM model. The top plot shows a sharp decrease in training loss during the initial epochs followed by a plateau, indicating convergence. The bottom plot shows the training accuracy increasing rapidly before stabilizing near 100%, demonstrating the model's effective learning of the training data.

4. Results

The experimental evaluation of the proposed system involved assessing its performance on both the training and testing datasets. The model exhibited stable convergence during training, with the cross-entropy loss function consistently decreasing over the first 20 epochs. By the end of training the model achieved a training accuracy of 100%, indicating that it had successfully learned the underlying patterns in the training data. This rapid convergence and high training accuracy suggests that combining Mediapipe features with the LSTM architecture is well suited for the task of NSL gesture recognition. The addition of dropout layers proved effective in mitigating overfitting, ensuring that the model's performance did not degrade when evaluated on unseen data.

The model achieved an accuracy of 95% on the testing dataset, demonstrating its robustness in recognizing gestures that it had not encountered during training. Precision and recall metrics were consistently above 90% for most gesture classes, reflecting the model's ability to correctly identify gestures while minimizing false positives and false negatives. The high F1-scores further corroborate the model's effectiveness, indicating a balance between precision and recall across all gesture classes. Gestures with distinct hand shapes, such as 0, 4, 5, 6, 7, and 9, were recognized with perfect accuracy, and their unique

features were easily distinguishable by the model. These results highlight the strength of the system in handling gestures with clear spatial differences.

The performance of a gesture or sign language recognition model can be evaluated using a confusion matrix in both sample-based and normalized forms. The confusion matrix in Figure 6 highlights the model's classification accuracy and misclassifications across the different gesture classes.

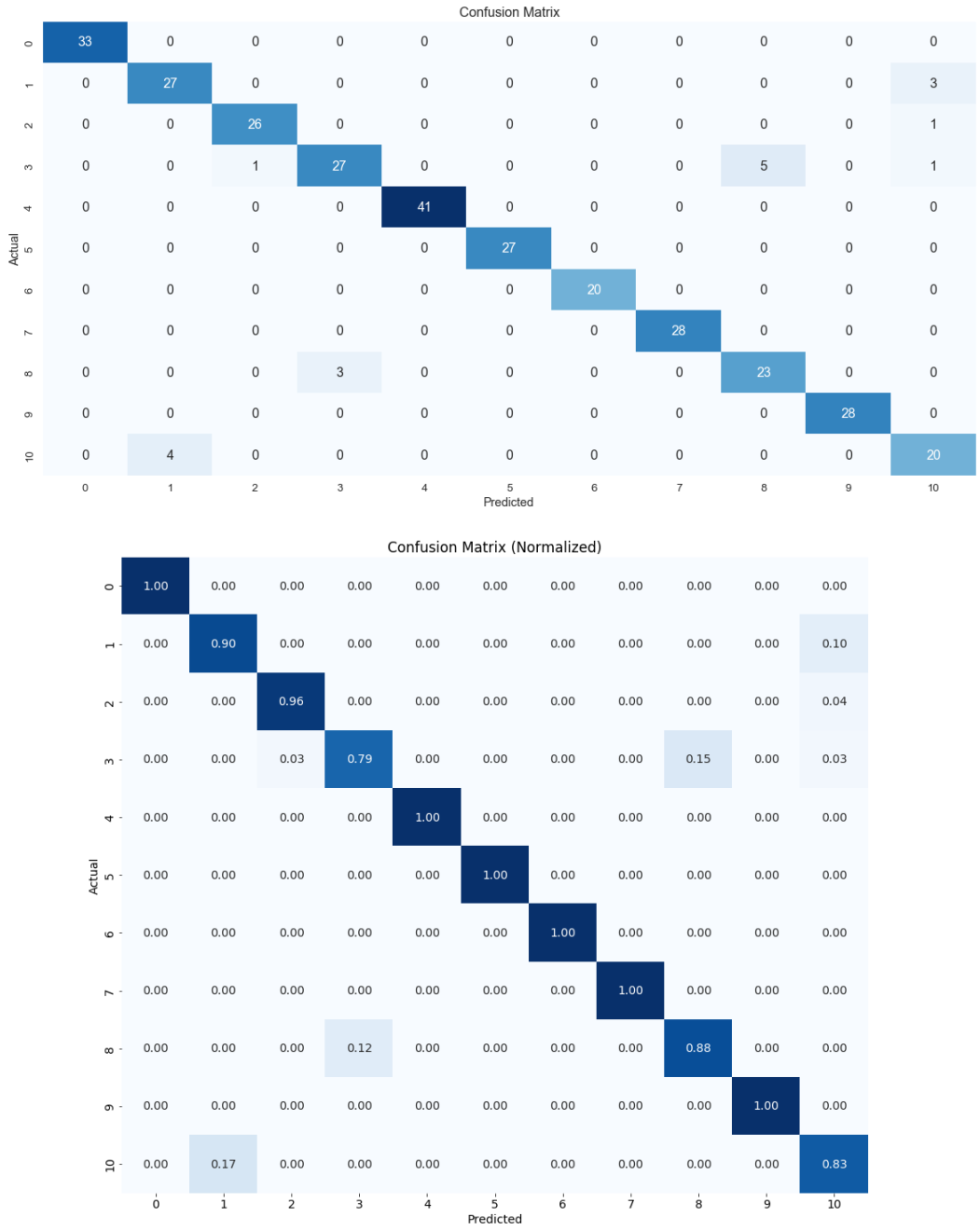

Figure 6. Sample-based and normalized confusion matrices illustrating the performance of the proposed gesture recognition model for NSL gestures (numbers 0 to 10). Correct classifications are shown along the diagonal, while off-diagonal values represent misclassifications. These matrices provide insights into the strengths and weaknesses of the model across all gesture classes.

The confusion matrix represents the performance of our classification model designed to recognize 11 signs in NSL. Each row corresponds to the actual signs, while each column represents the predicted signs. Correct classifications are reflected along the diagonal,

while misclassifications are represented by the non-diagonal values. A detailed analysis of the model's correct and incorrect classifications provides insight into the its strengths and weaknesses. The model performed perfectly for several signs. Sign 0 was correctly classified in all 33 instances without any misclassification; similarly, Signs 4, 5, 6, 7, and 9 all achieved perfect classification, with 36, 27, 22, 29, and 31 accurately recognized instances, respectively. These results indicate that the model has strong ability to distinguish these signs from others, likely due to the clear separability of their features.

For Signs 1 and 2, the model correctly classified most of the instances, but misclassified a few instances as Sign 10. Sign 3 had 27 correct classifications, but showed confusion with Signs 2, 8, and 10, with one, five, and one instances being misclassified, respectively. Sign 8 had 23 correct classifications, but three samples were misclassified as Sign 3. Furthermore, Sign 10 had 20 correct classifications, but four samples were misclassified as Sign 1. This indicates potential similarity in the gestures or features of these signs, leading to difficulty in distinguishing them.

This bidirectional misclassification suggests that these signs share common patterns that challenge the model's differentiation capabilities. Overall, the model demonstrated strong classification performance, achieving perfect or near-perfect recognition for the majority of signs. However, there are clear areas for improvement. While signs such as 0, 4, 5, 6, 7, and 9 show flawless performance, challenges remain in distinguishing between others, such as Signs 3 and 8. By focusing on these areas and implementing targeted improvements, the model's accuracy and robustness for NSL recognition can be further enhanced.

In addition to accuracy metrics, we evaluated the system's robustness to variations in signing styles and environmental conditions. The data augmentation techniques applied during training proved beneficial, enabling the model to generalize well to gestures performed under different conditions. For instance, the model handled variations in hand orientation, slight changes in camera angles, and differences in signing speed with minimal impact on accuracy. These findings suggested that the system is well suited for practical applications where such variations are common. Table 1 summarizes the model's performance across all classes in terms of precision, recall, F1-score, and support. In addition to LSTM, we also tested gated recurrent units (GRUs) and simple recurrent neural networks (RNNs), which achieved mean accuracy of 94% and 93%, respectively. RNN, LSTM, and GRU models are all designed to handle sequential data by maintaining information about previous inputs through internal memory; however, they differ in how they manage this memory and address issues such as the vanishing gradient problem. RNNs are the simplest form, with each neuron passing information to the next time step; however, they struggle with long-term dependencies due to vanishing gradients during training. In general, LSTMs excel at learning complex patterns compared to GRUs, while RNNs are usually most suitable for simpler tasks.

We tested several other algorithms for NSL. We utilized the scikit-learn (sklearn) library, a widely used open-source machine learning framework in Python, to implement and evaluate various classic classification algorithms on the NSL dataset. Known for its simplicity and efficiency, scikit-learn offers a wide range of machine learning algorithms along with tools for model selection, preprocessing, and evaluation. Using scikit-learn, we tested models such as random forest, logistic regression, support vector machine (SVM), K-nearest neighbors (KNN), and decision tree along with and ensemble methods such as AdaBoost, ExtraTrees, bagging, gradient boosting machine (GBM), XGBoost, and CatBoost. Additionally, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), naive Bayes, and a simple multilayer perceptron (MLP) were applied. The features were flattened before applying the algorithms to each video, i.e., 20×102 became 2040.

Table 1. Classification report for NSL gesture recognition, summarizing precision, recall, F1-score, and support for each class (0 to 10). The macro-average and weighted-average metrics indicate strong overall performance across all classes.

Sign	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	28
1	0.91	0.91	0.91	33
2	0.97	0.97	0.97	29
3	0.75	0.92	0.83	26
4	1.00	1.00	1.00	35
5	0.95	1.00	0.98	20
6	1.00	1.00	1.00	25
7	1.00	1.00	1.00	30
8	0.96	0.76	0.85	33
9	1.00	1.00	1.00	28
10	0.90	0.90	0.90	31
Accuracy	0.95	0.95	0.95	-
Macro Avg	0.95	0.95	0.95	318
Weighted Avg	0.95	0.95	0.95	318

The random forest classifier achieved an accuracy of 88%, demonstrating reliable performance due to its ensemble approach. Similarly, the logistic regression classifier obtained an accuracy of 87%, showing its effectiveness in handling linearly separable data. The SVM with a linear kernel performed slightly better, with an accuracy of 91%. The KNN classifier achieved an accuracy of 87%, benefiting from its simplicity and ability to classify based on feature proximity. The decision tree classifier provided an accuracy of 83%, which was slightly lower due to its tendency to overfit the data. Among ensemble methods, the AdaBoost classifier achieved an accuracy of 90%, while the ExtraTrees and bagging classifiers reached accuracy scores of 92% and 90%, respectively, highlighting the strength of combining multiple base models. Boosting techniques demonstrated strong results, with the GBM and XGBoost classifiers reaching accuracy scores of 92% and 93%, respectively. The CatBoost classifier followed closely with an accuracy of 93%, emphasizing its ability to effectively handle structured datasets.

Furthermore, LDA and QDA achieved accuracy scores of 85% and 86%, respectively, showing reasonable performance for datasets with distinct class distributions. The naive Bayes classifier achieved an accuracy of 86%, demonstrating effectiveness despite the assumption of feature independence. The simple MLP neural network achieved an accuracy of 91%, showcasing the strength of neural networks even in a simple architecture. Overall, the LSTM network achieved the highest accuracy of 95%, outperforming all other classifiers due to its capability to capture temporal dependencies and sequential patterns within the data.

The integration of hand and mouth features also played a significant role in the system's success. By combining these complementary features, the model was able to distinguish between gestures that might otherwise be ambiguous when relying solely on hand movements. For example, the inclusion of mouth landmarks improved the recognition of gestures involving lip movements, such as certain numerical gestures in NSL. This multimodal approach enhances the system's overall accuracy and robustness, making it a valuable tool for NSL recognition. For each testing video, the process of feature extraction from all the key frames followed by application to the trained machine learning model was fast (950 ms) on a typical CPU-based personal computer, indicating the real-time applicability of the proposed system.

5. Discussion

The proposed system for recognizing NSL gestures demonstrated strong performance, achieving a testing accuracy of 95%. This success highlights the system's robustness to variations in signing styles, orientations, and speeds, which are common in real-world scenarios. The combination of Mediapipe for feature extraction and an LSTM network for temporal modeling proved highly effective, as Mediapipe's ability to extract precise hand and mouth landmarks provided a comprehensive representation of gestures. Simultaneously, the LSTM leveraged this multimodal information to capture the temporal dynamics critical for sequential gesture recognition. Notably, the inclusion of multimodal features, specifically both hand and mouth landmarks, was instrumental in distinguishing gestures with overlapping hand configurations but differing lip movements, such as Signs 1 and 10. Although this study focused on a limited-scale dataset of numbers 0 to 10, its promising results underscore the potential for real-world applications, particularly with further optimization for real-time deployment.

Despite its strengths, the proposed system faced challenges in handling gestures with overlapping features, such as Signs 3 and 8, which were occasionally misclassified. This suggests that the model struggled to differentiate gestures with similar hand shapes or spatial configurations. To address this limitation, additional contextual or spatial information could be incorporated into the system, such as arm movements, body posture, or environmental cues. Another limitation was the data imbalance across gesture classes, with fewer samples for gestures such as Signs 3 and 6. This imbalance likely contributed to the observed misclassifications, highlighting the importance of balanced datasets in training robust models. Addressing these challenges through targeted data augmentation or additional data collection for underrepresented classes could further improve model performance.

Compared to other studies employing Mediapipe and LSTM models for sign language recognition, the performance of the proposed system aligns with or surpasses existing benchmarks. For instance, a system for German Sign Language recognition utilizing Mediapipe for feature extraction and LSTM for temporal modeling achieved a validation accuracy of 96.55% [34]. Similarly, a hybrid CNN and bidirectional LSTM approach for Arabic Sign Language achieved an accuracy of 94.8%, demonstrating the effectiveness of combining spatial and temporal modeling techniques [35]. While the proposed system achieved a slightly lower accuracy of 95%, it should be noted that the dataset used here consisted of a limited vocabulary of 11 gestures, making direct comparisons less reliable.

The inclusion of multimodal features, specifically hand and mouth landmarks, is consistent with findings from previous studies where the integration of additional spatial features improved model accuracy. For example, a framework utilizing pose details and CNN-LSTM for multilingual sign recognition demonstrated that multimodal inputs could enhance classification performance, achieving an accuracy above 95% on benchmark datasets [36]. These results collectively highlight the efficacy of LSTM-based architectures in handling the temporal and spatial complexities of sign language gestures when paired with precise feature extraction mechanisms such as Mediapipe.

Future iterations of the proposed system should prioritize expanding the dataset by including additional gestures and environmental variations, such as different lighting conditions, camera angles, and signing speeds. Increasing the vocabulary of NSL gestures would enhance the system's scalability and applicability in real-world contexts. Furthermore, integrating additional modalities such as depth sensors or accelerometers could provide richer spatial and temporal data, addressing challenges related to gestures with overlapping features and improving the model's overall performance and adaptability. Additionally, future work will incorporate cross-validation to provide a more robust evalu-

ation of the model's performance and ensure its stability across diverse data splits. These enhancements align with trends in sign language recognition research, where multimodal systems have been found to outperform vision-only approaches in complex scenarios [37].

The ultimate goal of this work is to develop an educational application that facilitates real-time NSL recognition and evaluation. Such an application could allow users to input phrases that are translated into NSL using an avatar signer while simultaneously enabling them to practice NSL gestures and receive real-time feedback on their accuracy. This work represents an important first step towards addressing the lack of resources for NSL, which is currently underrepresented in sign language research. By laying the foundation for a robust dataset and exploring methods to enhance recognition performance, this study highlights the potential for future applications tailored to NSL. By combining real-time translation capabilities with educational tools, the proposed system has the potential to support NSL learners and promote accessibility for the deaf and hard-of-hearing communities. Future developments will focus on refining the system's performance to achieve efficient real-time deployment and seamless user experiences.

6. Conclusions

This study has presented a preliminary system for recognizing NSL gestures, focusing on the numbers 0 to 10. The proposed system uses Mediapipe for feature extraction and an LSTM network for temporal modeling. The system achieved a testing accuracy of 95%, demonstrating its robustness to variations in signing styles, orientations, and speeds. The integration of multimodal features, specifically hand and mouth landmarks, proved instrumental in distinguishing gestures with overlapping configurations. While challenges such as data imbalance and misclassification of similar gestures (e.g., Signs 3 and 8) were observed, our results underscore the potential of the proposed approach. This work marks an important first step toward addressing the underrepresentation of NSL in sign language research, and highlights the need for a robust video-based dataset to enable future applications. With further development, the proposed system can form the basis of an educational application for real-time NSL recognition, translation, and evaluation, promoting accessibility and support for NSL learners and the broader deaf and hard-of-hearing community.

Author Contributions: Conceptualization, M.Z.U. and C.B.; methodology, M.Z.U.; software, M.Z.U.; validation, M.Z.U.; investigation, C.B.; resources, P.R.; figures, M.Z.U.; data curation, M.Z.U.; writing—original draft preparation, C.B. and M.Z.U.; writing—review and editing, M.Z.U., C.B. and P.R.; visualization, M.Z.U. and C.B.; project administration, M.Z.U.; funding acquisition, M.Z.U., C.B. and P.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Stiftelsen Dam through the "KI-drevet Norsk Tegnspråkoversetter" project.

Institutional Review Board Statement: No new data collection took place for this work. Pre-existing videos of Statped employees/educators demonstrating NSL gestures, originally recorded for a public NSL educational platform, were used. (Sign language webpage: https://www.statped.no/tegnsprak/, accessed on 5 December 2024).

Informed Consent Statement: Statped confirms that it owns the copyright to the videos referenced in this research work. These videos feature Statped employees (educators of Norwegian Sign Language), who have provided consent to appear and are aware that their likeness is used by the company in various capacities, including educational, research, and public-facing projects.

Data Availability Statement: The datasets presented in this article are not readily available because they are part of an ongoing study. Requests to access the datasets should be directed to the authors.

Acknowledgments: We would like to thank Statped and Statped educators for providing the video material needed in this work, and the Norwegian Association of the Deaf (Norges Døveforbund) for their interest in and support of the project.

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ASL	American Sign Language
CNN	Convolutional Neural Network
DL	Deep Learning
KNN	K-Nearest Neighbor
LSTM	Long Short-Term Memory
ML	Machine Learning
NSL	Norwegian Sign Language
SL	Sign Language
SLR	Sign Language Recognition
SVM	Support Vector Machine

References

1. World Health Organization. Deafness and Hearing Loss. 2024. Available online: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss (accessed on 5 December 2024).
2. Svendsen, B.; Kadry, S. Comparative Analysis of Image Classification Models for Norwegian Sign Language Recognition. *Technologies* **2023**, *11*, 99. [CrossRef]
3. World Federation of the Deaf. Our Work. 2021. Available online: https://wfdeaf.org/our-work/ (accessed on 5 December 2024).
4. Pigou, L.; Dieleman, S.; Kindermans, P.J.; Schrauwen, B. Sign language recognition using convolutional neural networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 572–578. [CrossRef]
5. Ambar, R.; Salim, S.; Abd Wahab, M.H.; Jamil, M.M.A.; Phing, T.C. Development of a Wearable Sensor Glove for Real-Time Sign Language Translation. *Ann. Emerg. Technol. Comput.* **2023**, *7*, 25–38. [CrossRef]
6. Otoom, M.; Alzubaidi, M.A. Ambient intelligence framework for real-time speech-to-sign translation. *Assist. Technol.* **2018**, *30*, 119–132. [CrossRef] [PubMed]
7. Brega, J.; Rodello, I.; Dias, D.R.C.; Salvador, V.F.M.; Guimarães, M. A virtual reality environment to support chat rooms for hearing impaired and to teach Brazilian Sign Language (LIBRAS). In Proceedings of the 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), Doha, Qatar, 10–13 November 2014; pp. 433–440. [CrossRef]
8. Dewanto, F.M.; Santoso, H.A.; Shidik, G.F.; Purwanto. Scoping Review Of Sign Language Recognition: An Analysis of MediaPipe Framework and Deep Learning Integration. In Proceedings of the IEEE 2024 International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, Indonesia, 21–22 September 2024; pp. 451–458. [CrossRef]
9. Sundar, B.; Bagyammal, T. American sign language recognition for alphabets using MediaPipe and LSTM. *Procedia Comput. Sci.* **2022**, *215*, 642–651. [CrossRef]
10. Zhi, D.; Oliveira, T.E.D.; Prado da Fonseca, V.; Petriu, E. Teaching a Robot Sign Language using Vision-Based Hand Gesture Recognition. In Proceedings of the 2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), Ottawa, ON, Canada, 12–13 June 2018; pp. 1–6. [CrossRef]
11. Dardas, N.H.; Georganas, N. Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 3592–3607. [CrossRef]
12. Alsharif, B.; Alanazi, M.; Ilyas, M. Machine Learning Technology to Recognize American Sign Language Alphabet. In Proceedings of the 2023 IEEE 20th International Conference on Smart Communities: Improving Quality of Life Using AI, Robotics and IoT (HONET), Boca Raton, FL, USA, 4–6 December 2023; pp. 173–178. [CrossRef]
13. Guerrieri, B.T. Enhancing American Sign Language Classification by Leveraging Hand Landmark Extraction. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2. ACM, Portland, OR, USA, 20–23 March 2024. [CrossRef]
14. Ahmed, H.F.T.; Ahmad, H.; Narasingamurthi, K.; Harkat, H.; Phang, S.K. DF-WiSLR: Device-Free Wi-Fi-based Sign Language Recognition. *Pervasive Mob. Comput.* **2020**, *69*, 101289. [CrossRef]

15. Ji, A.; Wang, Y.; Miao, X.; Fan, T.; Ru, B.; Liu, L.; Nie, R.; Qiu, S. Dataglove for Sign Language Recognition of People with Hearing and Speech Impairment via Wearable Inertial Sensors. *Sensors* **2023**, *23*, 6693. [CrossRef] [PubMed]
16. Rahman, M.M.; Islam, M.S.; Rahman, M.H.; Sassi, R.; Rivolta, M.W.; Aktaruzzaman, M. A New Benchmark on American Sign Language Recognition using Convolutional Neural Network. In Proceedings of the 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 24–25 December 2019; pp. 1–6. [CrossRef]
17. Ma, Y.; Xu, T.; Han, S.; Kim, K. Ensemble Learning of Multiple Deep CNNs Using Accuracy-Based Weighted Voting for ASL Recognition. *Appl. Sci.* **2022**, *12*, 11766. [CrossRef]
18. Sharma, S.; Kumar, K. ASL-3DCNN: American Sign Language Recognition Technique Using 3-D Convolutional Neural Networks. *Multimed. Tools Appl.* **2021**, *80*, 26319–26331. [CrossRef]
19. Liu, X.; Cao, C.; Duan, S. A Low-Power Hardware Architecture for Real-Time CNN Computing. *Sensors* **2023**, *23*, 2045. [CrossRef]
20. Aydin, S.; Bilge, H.S. Optimal Hardware Implementation for End-to-End CNN-Based Classification. In Proceedings of the 2023 4th International Conference on Innovative Trends in Information Technology (ICITIIT), Kottayam, India, 11–12 February 2023; pp. 1–6. [CrossRef]
21. Liu, T.; Zhou, W.g.; Li, H. Sign Language Recognition with Long Short-Term Memory. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 2871–2875. [CrossRef]
22. Guo, D.; Zhou, W.g.; Li, H.; Wang, M. Hierarchical LSTM for Sign Language Translation. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI), New Orleans, LA, USA, 2–7 February 2018; pp. 6845–6852. [CrossRef]
23. Abdullahi, S.B.; Chamnongthai, K. American Sign Language Words Recognition Using Spatio-Temporal Prosodic and Angle Features: A Sequential Learning Approach. *IEEE Access* **2022**, *10*, 15911–15923. [CrossRef]
24. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
25. Mercanoglu, O.; Tur, A.O.; Keles, H. Isolated Sign Language Recognition with Multi-scale Features using LSTM. In Proceedings of the IEEE 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 24–26 April 2019; pp. 1–4. [CrossRef]
26. Huang, J.; Chaijaruwanich, J.; Chouvatut, V. Video-based Sign Language Recognition with R(2+1)D and LSTM Networks. In Proceedings of the IEEE 2024 16th International Conference on Knowledge and Smart Technology (KST), Krabi, Thailand, 28 February–2 March 2024; pp. 214–219. [CrossRef]
27. Khartheesvar, G.; Kumar, M.; Yadav, A. Automatic Indian sign language recognition using MediaPipe holistic and LSTM network. *Multimed. Tools Appl.* **2024**, *83*, 58329–58348. [CrossRef]
28. Rao, G.; Sowmya, C.; Mamatha, D. Sign Language Recognition using LSTM and Media Pipe. In Proceedings of the 2023 IEEE International Conference on Advances in Signal Processing and Communication Systems (SPCOM), Shanghai, China, 25–28 September 2023; pp. 1–5. [CrossRef]
29. Nguyen, P.T.; Nguyen, T.H.; Hoang, N.X.N.; Phan, H.T.B.; Vu, H.S.H.; Huynh, H.N. Exploring MediaPipe Optimization Strategies for Real-Time Sign Language Recognition. *CTU J. Innov. Sustain. Dev.* **2023**, *15*, 142–152. [CrossRef]
30. Farhan, Y.; Madi, A. Real-Time Dynamic Sign Recognition Using MediaPipe. In Proceedings of the 2022 IEEE 3rd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), Fez, Morocco, 1–2 December 2022; pp. 1–7. [CrossRef]
31. Jintanachaiwat, W.; Jongsathitphaibul, K.; Pimsan, N.; Sojiphan, M.; Tayakee, A.; Junthep, T.; Siriborvornratanakul, T. Using LSTM to translate Thai sign language to text in real time. *Discov. Artif. Intell.* **2024**, *4*, 17. [CrossRef]
32. Norges Døveforbund. Norsk Tegnspråk. 2023. Available online: https://www.doveforbundet.no/tegnsprak/ (accessed on 5 December 2024).
33. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning (ICML-PMLR), Lille, France, 6–11 July 2015; pp. 37:448–37:456.
34. Irahal, F.N.; Youssef, R.B.; Meyer, D. A Real-time Approach for Recognizing German Sign Language. In Proceedings of the 2024 10th International Conference on Control, Decision and Information Technologies (CoDIT), Valetta, Malta, 1–4 July 2024; pp. 2443–2448. [CrossRef]
35. Sagheer, N.S.; Almasoudy, F.H.; Bashaa, M.H. Enhancing Arabic Sign Language Recognition using Deep Learning. *Int. J. Innov. Technol. Explor. Eng.* **2024**, *13*, 18–23. [CrossRef]
36. Natarajan, B.S.; Rajalakshmi, E.; Elakkiya, R.; Kotecha, K.; Abraham, A.; Gabralla, L.A.; Subramaniyaswamy, V. Development of an End-to-End Deep Learning Framework for Sign Language Recognition, Translation, and Video Generation. *IEEE Access* **2022**, *10*, 104358–104374. [CrossRef]
37. Bird, J.J.; Ekárt, A.; Faria, D.R. British Sign Language Recognition via Late Fusion of Computer Vision and Leap Motion with Transfer Learning to American Sign Language. *Sensors* **2020**, *20*, 5151. [CrossRef] [PubMed]