applied sciences

MDPI

Article

A Four-Stage Mahalanobis-Distance-Based Method for Hand Posture Recognition

Dawid Warchoł *,† **and Tomasz Kapuściński** †

Department of Computer and Control Engineering, Faculty of Electrical and Computer Engineering, Rzeszów University of Technology, W. Pola 2, 35-959 Rzeszów, Poland; tomekkap@kia.prz.edu.pl
* Correspondence: dawwar@prz.edu.pl; Tel.: +48-17-865-1614
† These authors contributed equally to this work.

Abstract: Automatic recognition of hand postures is an important research topic with many applications, e.g., communication support for deaf people. In this paper, we present a novel four-stage, Mahalanobis-distance-based method for hand posture recognition using skeletal data. The proposed method is based on a two-stage classification algorithm with two additional stages related to joint preprocessing (normalization) and a rule-based system, specific to hand shapes that the algorithm is meant to classify. The method achieves superior effectiveness on two benchmark datasets, the first of which was created by us for the purpose of this work, while the second is a well-known and publicly available dataset. The method's recognition rate measured by leave-one-subject-out cross-validation tests is 94.69% on the first dataset and 97.44% on the second. Experiments, including comparison with other state-of-the-art methods and ablation studies related to classification accuracy and time, confirm the effectiveness of our approach.

Keywords: hand posture recognition; static gesture recognition; Polish finger alphabet; American finger alphabet; multistage classification; Mahalanobis distance

Citation: Warchoł, D.; Kapuściński, T. A Four-Stage Mahalanobis-Distance-Based Method for Hand Posture Recognition. *Appl. Sci.* **2023**, *13*, 12347. https://doi.org/10.3390/app132212347

Academic Editor: Antonio Fernández-Caballero

Received: 12 October 2023
Revised: 7 November 2023
Accepted: 13 November 2023
Published: 15 November 2023

1. Introduction

Automatic recognition of static hand gestures, also referred to as hand postures or hand shapes, is an important and long-developed research topic. Its primary application is to provide technological support for people with hearing loss and deafness [1]. Recognition algorithms can also be applied in real-time automotive interfaces [2], gaming (e.g., virtual reality video games) [3], smart home automation [4], etc.

In recent years, algorithms for the automatic recognition of objects in color images have mainly been based on deep learning techniques. The goal of this paper is to prove that effective hand posture recognition can be performed by an algorithm that does not require a large amount of training data or a time-consuming training process, and does not use hand-crafted or automatically generated features. We propose an approach in which deep learning is used only at the stage of hand detection, e.g., by the MediaPipe library, and the recognition is based only on hand joint coordinates. The means to achieve this goal is a two-stage classification algorithm with two additional stages related to joint preprocessing (normalization) and a rule-based system, specific to hand shapes that the algorithm is meant to classify. The additional advantage of our method is the ability to perform well and run in real time even with a low-resolution, low-quality laptop camera.

The contributions of this paper are as follows.

1. A novel four-stage hand posture recognition method based on posture-specific rules and Mahalanobis distance;
2. A novel lightweight dataset consisting of hand skeletal data recorded from 12 people;
3. Comparative evaluation of the proposed algorithm with related approaches on a popular, publicly available dataset.

The rest of the paper is arranged as follows. Related works are presented in Section 2. The proposed hand shape recognition algorithm is presented in Section 3. Section 4 discusses the benchmark datasets and the experiments carried out on them. Section 5 concludes the paper.

2. Related Work

Recently, deep networks have been developed to obtain articulated hand models called skeletons [5,6]. These networks use color images from normal cameras and are an attractive alternative to hardware solutions requiring specialized RGB-D sensors [7,8]. Approaches using the potential of these networks to recognize static hand configurations have appeared in the literature.

Once the skeleton data is determined, various classification methods are used. There are known solutions using k-nearest neighbors classifier (KNN) [9], support vector machine (SVM) [10–12], rule-based classifier [13], artificial neural networks (ANN) [14–17], random forest (RF) [12,18], gradient boosting (GB) [11,12], and various models based on deep learning [15,18–30].

The authors of [19] used an artificially generated skeleton data to tune up a deep network trained on real samples. The authors of [15] showed that it is reasonable to train a deep network on hybrid data containing color images and skeletons.

One of the challenges when recognizing finger alphabets is the high similarity of the shapes considered. To solve this problem, a hierarchical approach, in which groups of classes with similar hand shapes are identified first, has been proposed [16].

There are also descriptions of useful applications in the literature, e.g., the Sign-to-911 system, which is an emergency call service for sign language users [31], a doctor–patient dialogue system [27], or a solution supporting medical consultation [29]. Lightweight algorithms are being developed that could be run on mobile devices with limited computing resources [10].

Works are also described in which skeletal hand models are used to build educational games for learning the finger alphabet or sign language [20,32–34].

Table 1 briefly characterizes a selection of recent works on static gesture recognition based on algorithmically determined skeletal information.

Table 1. Recent works on static gesture recognition based on algorithmically determined skeletons.

Work	Domain	Classifier	# Classes	l-o-s-o	Accuracy [%]
[14]	Thai Finger Alphabet	ANN	30	no	84.57
[19]	Irish Finger Alphabet	CNN	23	no	71.00
[10]	Japanese Finger Alphabet	SVM	24	no	100.00
[18]	Arabic Finger Alphabet + control signs	RF	30	no	99.70
[11]	American Finger Alphabet	GB	26	no	99.39
[20]	Static gestures from Greek Sign Language	ResNet2+1D	36	yes	96.20
[35]	American Finger Alphabet + numbers and words	SVM	51	no	98.98
[16]	American Finger Alphabet + control signs	ANN	29	no	94.07
[36]	American Finger Alphabet + static gestures	ANN	38	no	94.29
[13]	Japanese Finger Alphabet	rule set	46	no	52.83
[22]	Static gestures from Indian Sign Language	CNN	15	no	98.90
[12]	Kazakh Finger Alphabet	RF, SVM, GB	31	no	98.80

For some of the works described, the table shows only that part of the results concerning static gestures. In some cases, there is no information on the number of users making gestures. In addition, the authors rarely use leave-one-subject-out validation, which verifies

the reliability of the method for gestures shown by people who are not present in the training dataset.

Solutions based on deep learning dominate. However, a recent review article [37] points out the lack of comprehensive, representative, and annotated datasets, especially for languages other than American Sign Language (ASL). Therefore, training deep models from scratch is tedious and expensive. With this in mind, it is worth turning to solutions that use the potential of trained networks generating skeleton data, require only fine-tuning using a limited training set, and use domain knowledge. The approach proposed in this work fits into this group. The proposed multistage method uses a small training set and a priori knowledge about the recognized shapes acquired by analysis of the one-stage classification confusion matrix and the considered shapes comparison.

3. Proposed Method

Our method is based on the Mahalanobis distance, which is a measure of the dissimilarity between two points in multidimensional space [38]. It takes into account the correlations between variables, making it especially useful in tackling the problems of multivariate data classification or clustering.

Given two points x and y, and covariance matrix Cov, the Mahalanobis distance d is defined as:

$$d(x,y) = \sqrt{(x-y)^T Cov^{-1}(x-y)} \quad (1)$$

A four-stage hierarchical classification method is proposed. Stage I is the preprocessing of skeletal data. During stage II, a preliminary classification is performed, some classes are rejected as less likely, and the remaining classes are selected for further processing. In stage III, a set of rules is applied to reject some joints for letters identified as problematic (challenging). The idea is to focus on the most distinctive part of the input data when dealing with frequently misclassified shapes. The final classification is carried out in stage IV, where only the remaining classes and joints are considered. The general scheme of the algorithm is presented in Figure 1. The input of the algorithm consists of an unknown sample, i.e., a posture to be classified, and the training samples, i.e., postures based on which the classification is performed. Stage III requires training, which is performed on a limited and disjoint dataset prior to recognition. During this process, inverse covariance matrices and mean representations are constructed. Thus, training and testing are not simultaneous. The training dataset can also be used to determine the rules used in stage III. Running stages I, II, and III on the training data can help identify difficult hand shapes. The following is a detailed description of the individual stages.

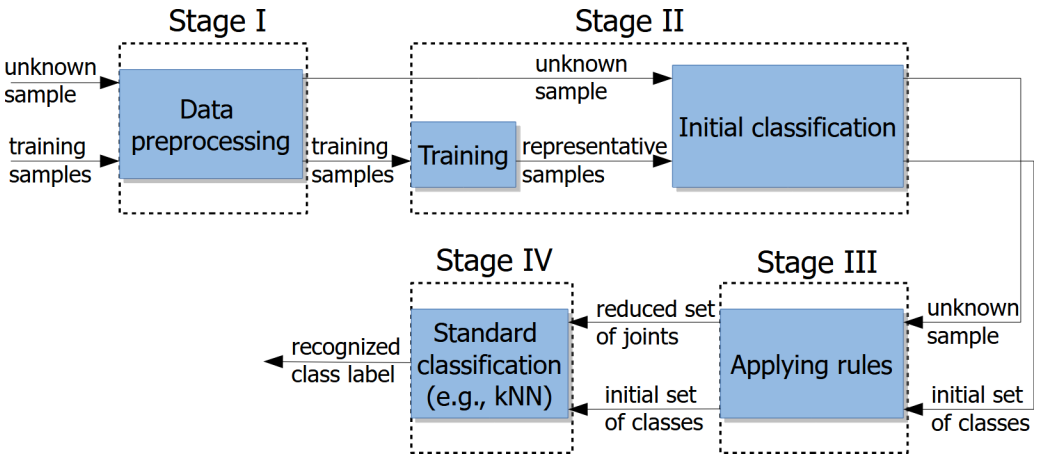


Figure 1. General scheme of the recognition algorithm.

3.1. Stage I

We consider a hand skeleton as a set of N joints (knuckles) where each joint P_i is described by three coordinates: P_i^x , P_i^y , P_i^z calculated based on a color image using the MediaPipe library. The skeletons are in a clockwise coordinate system in which the horizontal X axis is directed to the left, the vertical Y axis is facing up, and the Z axis coincides with the optical axis of the camera and is turned towards the observed objects. All 21 joints and their descriptions are presented in Figure 2.

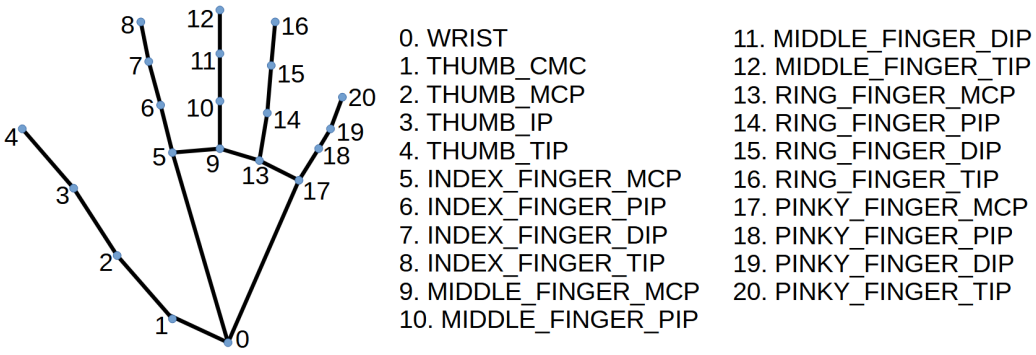


Figure 2. Hand skeleton joints generated by the MediaPipe library.

The first stage of the proposed recognition method involves initial processing of hand skeletons to make the recognition invariant to hand size, location, and orientation around the Z axis perpendicular to the camera lens. The size invariance is achieved by dividing all coordinates of each joint by the sum of distances between joints 0 and 9, and between joints 9 and 10. We decided to include not only the segment representing palm size (0–9) but also the segment representing the longest finger length (9–10). Our motivation to normalize hand size according to both segments was the observation that the fingers-to-palm ratio varies between people and the differences can be significant [39]. The decision to exclude the segments between joints 10–11 and 11–12 was made upon the observation that these joints are often imprecisely calculated because they are occluded by other hand parts in some gestures.

The invariance of the hand location is achieved by translating the whole skeleton so that joint 0 is always at (0, 0, 0) m coordinates. The final step of data preprocessing involves making the method invariant to the orientation of the hand around the Z axis by rotating the whole skeleton so that the segment between joints 0 and 9 is vertical. It is important to verify whether among the recognized gestures there exist any whose shape is identical, and the only difference is their orientation. If such gestures exist, the normalization of hand orientation should be omitted.

3.2. Stage II

Stage II requires training using a set of M hand posture classes. The training procedure is described in Algorithm 1.

After the inverse covariance matrices Cov_i and the mean representatives $Mean_i^x$, $Mean_i^y$, $Mean_i^z$ for all classes and joints are calculated in the training procedure, the initial classification of an unknown sample U can be performed as in Algorithm 2.

Algorithm 1: Training procedure of the stage II of the proposed method.

```
foreach posture class C_j, 1 ≤ j ≤ M do
    foreach joint P_i, 1 ≤ i ≤ N do
        Calculate covariance matrix Cov_i for all coordinates based on samples from
         a training dataset
        Cov'_i ← inverse of Cov_i
        Mean_i^x ← mean x coordinate of all training samples
        Mean_i^y ← mean y coordinate of all training samples
        Mean_i^z ← mean z coordinate of all training samples
    end
end
```

Algorithm 2: Classification procedure of the stage II of the proposed method.

```
IniC = Ø
foreach posture class C_j, 1 ≤ j ≤ M do
    mismatches = 0
    foreach joint P_i, 1 ≤ i ≤ N do
        Calculate Mahalanobis distance d of the unknown sample U and
         representative sample Mean_i^x, Mean_i^y, Mean_i^z based on inv. covariance
         matrices Cov'_i
        if d > ε then
            mismatches ← mismatches + 1
        end
    end
    if mismatches ≤ AllowableMisNum then
        Add C_j to set IniC
    end
end
if IniC is empty then
    No class is recognized and the algorithm ends
else if IniC has one element then
    The element of IniC is the final recognized class of the algorithm
else
    Go to stage III
end
```

where $AllowableMisNum$ is a number of joints whose Mahalanobis distance to other samples is allowed to be greater than ε. If there are more such joints, the class of the representative sample is not added to the set of initially recognized classes $IniC$, which is the result of the second stage of the algorithm. Otherwise, the class is added to $IniC$. $AllowableMisNum$ and ε are parameters of the method. It should be noted that if $IniC$ is empty or has only one element, then stages III and IV are omitted. In the first scenario, no class is recognized by the method; in the second case, the first and only element of $IniC$ is the final recognized class.

3.3. Stage III

In this optional stage, a user should provide rules specific to the shapes that the algorithm is supposed to recognize. In each rule, the decision about removing particular joints is made based on the conditions under which the coexistence of particular posture classes in $IniC$ is verified.

We found that our main dataset with all 16 static gestures of the Polish finger alphabet (letters: A, B, C, E, I, L, M, N, O, P, R, S, T, U, W, Y) requires only three rules to significantly improve the performance of the algorithm. The rules are as follows:

- If *IniC* contains classes O and S or *IniC* contains classes S and T, then remove all joints except THUMB_IP, THUMB_TIP, and INDEX_FINGER_TIP;
- If *IniC* contains classes M and E, then remove all joints except PINKY_TIP;
- If *IniC* contains classes L and C, then remove all joints except INDEX_FINGER_TIP.

The best way to develop rules is to perform an initial validation with only stages I, II, and IV, generate a confusion matrix, and find classes that have been confused with each other most often. Then, key hand joints, crucial to distinguish between those classes have to be chosen, and all joints except these should be removed from the next classification stage.

3.4. Stage IV

The final stage of the recognition algorithm is a typical classification method that recognizes the hand shape based on a set of classes reduced in stage II and a set of joints reduced in stage III. Any classifier can be used, although it is strongly recommended that it does not require a training process or that the training is very fast, since a model cannot be trained before the end of stage II (it requires a reduced set of classes). In our case, the common k-nearest neighbor (kNN) classifier was applied with the Euclidean metric and k (number of neighbors) set to 1. These values of the kNN parameters led to the most accurate classification in all experiments with each dataset.

4. Experiments and Discussion

4.1. Datasets

To validate our approach, we used two datasets. The first is a novel SPAS dataset (**S**keletons of **P**olish finger **A**lphabet **S**tatic gestures). It consists of 16 classes of static hand gestures corresponding to letters of the Polish finger alphabet (PFA): A, B, C, E, I, L, M, N, O, P, R, S, T, U, W, and Y. The hand shapes are shown in Figure 3. The remaining letters were not included, since they involve motion. Each gesture was shown five times by 12 subjects (9 male and 3 female) for a total of 960 gestures. The dataset was recorded with a simple laptop camera providing color images with a resolution of 640 × 480 pixels. The recording program was running in a continuous stream and each frame was processed by MediaPipe to generate hand skeletons, from which those corresponding to PFA letters were saved to separate files. The SPAS dataset is very lightweight (624 KB) since it contains only skeletal data without color images. It can be considered challenging because it includes the letters O, S, and T, whose PFA representations have very similar hand shapes. It is also worth noting that the resolution of images, based on which the skeletal joints were calculated, is relatively low compared to modern high-quality cameras. Therefore, the SPAS dataset is suitable for verifying whether the recognition method performs well even with older devices and does not require large input data, which is obviously related to faster processing time and lower memory consumption. The SPAS dataset, along with Python scripts for its processing, can be downloaded from our website [40].

The second dataset is the Massey University (MU) ASL digits dataset [41], which is referred to in our article as MU-ASL-digits. It consists of the 10 classes of static hand gestures corresponding to digits of the American finger alphabet. The hand shapes are shown in Figure 3. Each gesture was shown 5 to 25 times by five subjects for a total of 700 gestures stored as color images with various resolutions (they are cropped to contain only hands).

MediaPipe hand detection performs well only with images provided to it in a stream, since it is supported by contextual information from previous frames. If the method does not see a context, skeletons are often calculated with a low detection confidence level, which results in imprecise joint matching. In extreme cases, all joints are placed in completely wrong positions. Unfortunately, all popular static hand gesture datasets consist of separate, unrelated images. We only managed to generate skeletons with a high confidence level from MU-ASL-digit among several tested datasets. This was achieved by tricking the MediaPipe detector by showing it other gestures of the same class before detecting each hand. This artificial context was sufficient to generate high-confidence skeletons. However, the trick was unsuccessful with the Massey University ASL alphabet dataset and with

several other popular datasets. Therefore, we decided to compare our method with the other approaches using only the MU-ASL-digits dataset. It is worth noting that this issue does not mean that the method has serious practical limitations. It only makes it difficult (or impossible) to validate on datasets with images unrelated to each other. The issue does not affect the method implemented as a program recognizing gestures in real time.

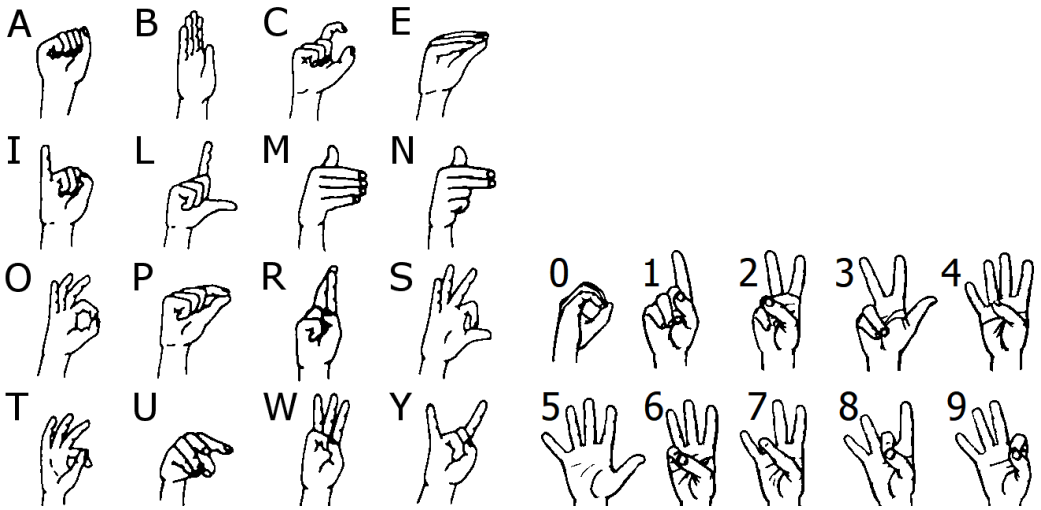


Figure 3. Hand shapes from the datasets used in our experiments: (**a**) Polish finger alphabet letters, (**b**) American finger alphabet digits.

The following rules were applied in stage III of the algorithm for MU-ASL-digit dataset:

- If *IniC* contains classes 4 and 6, then remove all joints except PINKY_TIP;
- If *IniC* contains classes 5 and 6, then remove all joints except THUMB_TIP and PINKY_TIP;
- If *IniC* contains classes 5 and 8, then remove all joints except THUMB_TIP and MIDDLE_FINGER_TIP;
- If *IniC* contains classes 4 and 9, then remove all joints except INDEX_FINGER_TIP;
- If *IniC* contains classes 2 and 7, then remove all joints except INDEX_FINGER_TIP, MIDDLE_FINGER_TIP, and PINKY_TIP.

The rules applied for the SPAS dataset are the same as those presented as an example in Section 3.

Figure 4 presents two postures from MU-ASL-digits with the MediaPipe skeletons.

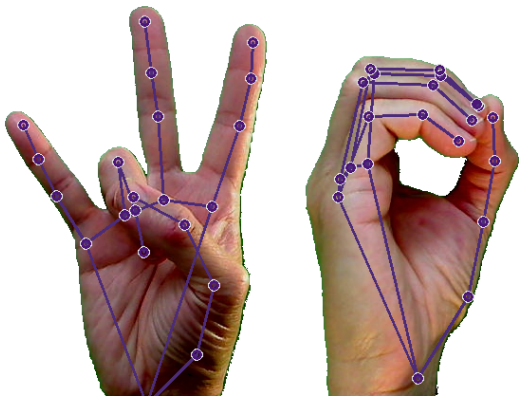


Figure 4. Postures of digits 7 and 0 from the MU-ASL-digits dataset with drawn skeletal joints obtained from MediaPipe.

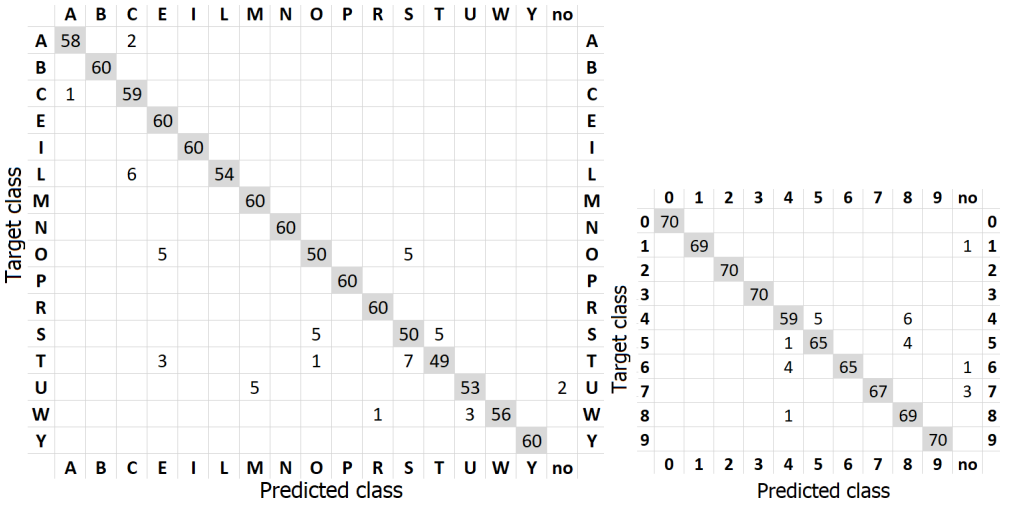
4.2. Validation Protocol

Our approach was verified using leave-one-subject-out (LOSO) k-fold cross-validation, where k was the number of subjects (12 for SPAS and 5 for MU-ASL-digits). We decided to use this validation protocol because it simulates real-life cases where the people whose gestures are being recognized usually do not participate in the creation of a training dataset. All methods compared with our approach were also verified using LOSO k-fold cross validation.

4.3. Results

In our experiments, we had to set the parameters of our method, $AllowableMisNum$ and ε, separately for SPAS and MU-ASL-digits to achieve the best performance. The value of $AllowableMisNum$ was set to 1 for SPAS and 0 for MU-ASL-digits. ε was set to 9 for SPAS and 8.6 for MU-ASL-digits. The impact of parameter values on classification accuracy is discussed in Section 4.5. Our method achieved a recognition rate of 94.69% with a standard deviation of 3.54%.

The confusion matrices are presented in Figure 5. For the SPAS dataset, the most frequently confused letter pairs are S-T (twelve times), O-S (ten times), and C-L (six times). All of the hand shapes corresponding to these letters are visually similar. In the case of MU-ASL-digits, the most frequent misclassifications are digits 4–8 (seven times) and 4–5 (six times). Interestingly, three times, the digit 7 was classified as 'no digit'. Such mistakes when a particular gesture is classified as 'no gesture' almost always result from the incorrectly detected skeleton by MediaPipe.



(**a**) (**b**)

Figure 5. Confusion matrices of the proposed method on the benchmark datasets: (**a**) SPAS and (**b**) MU-ASL-digits. The column named 'no' represents the case where no gesture was recognized.

We compared our approach with other methods found in the literature on the MU-ASL-digits dataset. For a fair comparison, we ensured that all the methods were validated by the same protocol. The results are presented in Table 2. Our method achieved the best recognition rate, outperforming the second best method (DeReFNets) by 1.3 percentage points.

Table 2. Comparison of the proposed method with other existing approaches on MU-ASL-digits.

Method	Recognition Rate [%]
Non-Negative Matrix Factorization + Compressive Sensing [42]	87.8
AlexNet 'FC6' + PCA and SVM [43]	95
Set of Geometric Features + Fisher Vector and SVM [44]	95.3
DeReFNets [45]	96.14
Our method	97.44

4.4. Ablation Study

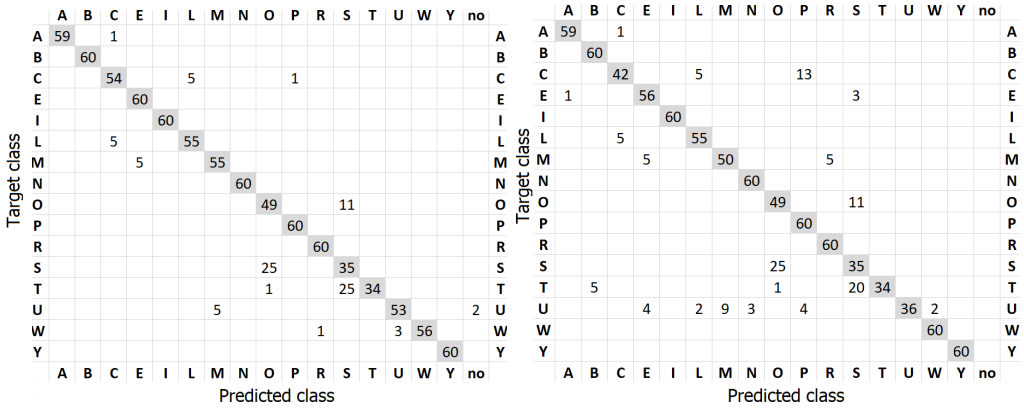
The classification accuracies of the whole method compared to versions with particular stages removed are presented in Table 3. Additionally, for the SPAS dataset, the confusion matrices for the reduced method are presented in Figure 6.

As one can see, stages II and III are important and their removal leads to lower effectiveness of the method on both datasets. However, the greatest deterioration of the results (about 20 percentage points) is visible in the case of MU-ASL-digits.

Based on the confusion matrix of the algorithm without stage III, we can see that the most frequently confused letter pairs are S-T and O-S. The rules applied in stage III are crucial to greatly reduce the number of such mistakes. Stage II, in turn, affects mainly the recognition of the letter U. It is probably due to the unusual hand shape of this letter, which sometimes makes it difficult for MediaPipe to detect all hand joints correctly. Even with some inaccurately detected joints, the algorithm is able to filter most of the remaining classes based on Mahalanobis distance to their representative samples.

Table 3. Recognition rates/stanard deviations [%] of the whole algorithm compared to versions with particular stages removed.

	SPAS	**MU-ASL-Digits**
whole algorithm (stages: I, II, III and IV)	94.69/3.54	97.44/3.07
stages: I, II, and IV	90.62/4.63	83.6/5.25
stages: I and IV	87.08/6.08	76.64/5.72



(**a**) (**b**)

Figure 6. Confusion matrices of the proposed method with particular stages removed on the SPAS dataset: (**a**) stages I, II, and IV and (**b**) stages I and IV.

The classification times of the whole method compared to versions with particular stages removed are presented in Table 4. The times were measured on a laptop with an Intel Core i5 8300H CPU. The average classification time of a gesture from the SPAS dataset is 4.2 ms with a standard deviation of 2.2 ms. The average time of a simple classification using only stage I and IV of the algorithm (data preprocessing and kNN) is 18.2 ms with a standard deviation of 4.6 ms. The whole algorithm is much faster because stage IV has the longest computation time and it can be greatly reduced by excluding a majority of classes in stage II and, in some cases, many skeletal joints in stage III. The gain from the reduction in stage IV computation time is greater than the combined time of stage II and stage III.

However, it should be mentioned that the presented times were measured without detecting a hand and creating skeletal data by MediaPipe. The average hand detection time in our experiments with the SPAS dataset was 50 ms with a standard deviation of 10 ms.

The learning time of our method on the entire SPAS dataset considered as a training set is 0.8 s. This means that learning is almost instantaneous for datasets of this size, which can be considered another advantage of our method.

Our method can run as a real-time video streaming application with more than 10 frames per second and high effectiveness, even if a person showing gestures is not present in the training set, which confirms the correctness of the validation tests performed.

Table 4. Average classification times/standard deviations [ms] of the whole algorithm compared to versions with particular stages removed.

	SPAS	**MU-ASL-Digits**
whole algorithm (stages: I, II, III, IV)	4.2/2.2	1.2/0.4
stages: I, II, IV	4.5/2.4	1.5/0.4
stages: I, IV	18.2/4.6	10.2/0.5

4.5. Impact of Parameters

The impact of the method parameters ε, *AllowableMisNum*, and k (the number of neighbors of the kNN classifier) on the classification accuracy with the SPAS and MU-ASL-digits datasets is presented in Figures 7–9. ε has little impact on accuracy within the range of [7.8–9.4] for both datasets. The impact of *AllowableMisNum* on the SPAS dataset is small, especially within the range of [0–3]. Interestingly, *AllowableMisNum* showed a great negative impact on MU-ASL-digits for any value different from 0. This may be related to the fact that there are more rules in stage III for MU-ASL-digits. The number of neighbors k shows an inverse correlation with the classification accuracy on both datasets. However, its impact is much greater in the case of SPAS.

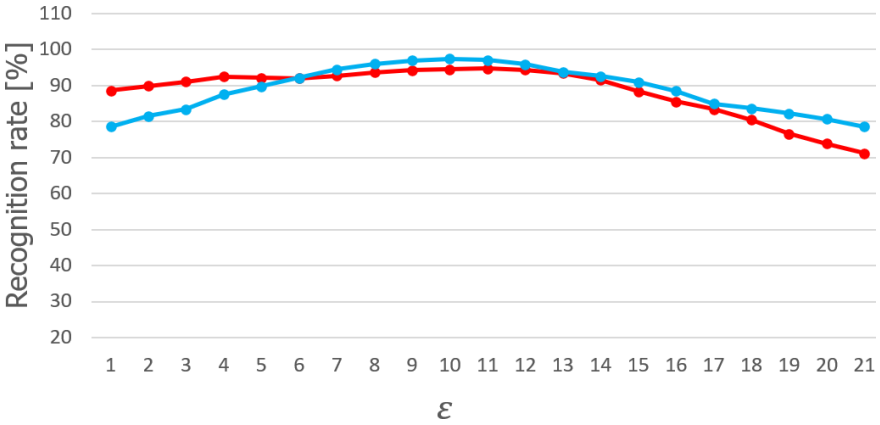


Figure 7. Impact of ε parameter on classification accuracy with SPAS (red curve) and MU-ASL-digits (blue curve) datasets.

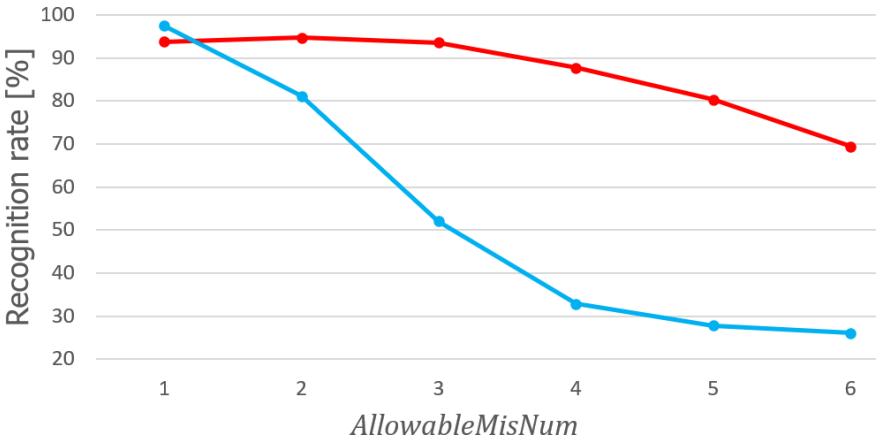


Figure 8. Impact of *AllowableMisNum* parameter on classification accuracy with SPAS (red curve) and MU-ASL-digits (blue curve) datasets.

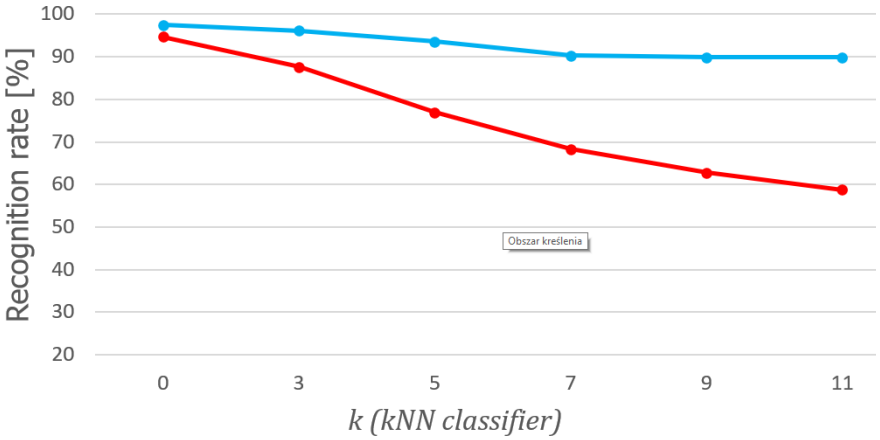


Figure 9. Impact of the number of neighbors k on classification accuracy with SPAS (red curve) and MU-ASL-digits (blue curve) datasets.

5. Conclusions

In this paper, we present a novel four-stage Mahalanobis-distance-based method for hand posture recognition using skeletal data. The proposed method achieves superior effectiveness on two benchmark datasets, the first of which was created by us for the purpose of this work, while the second is a well-known and publicly available dataset. The proposed method was validated using the challenging and practical LOSO k-fold cross-validation protocol. A comparison with other state-of-the-art methods and ablation studies related to classification accuracy and time confirm the usefulness of our approach. Our method uses deep learning only on the hand detection stage; however, it manages to outperform the other methods, some of which are entirely deep-learning-based.

In our experiments, hand skeletons were obtained using the MediaPipe library. This was chosen because the authors believe that it is currently the most reliable of the publicly available hand detection tools. The latest improvements in the field of the automatic detection and tracking of human body parts allow us to assume that in the near future such algorithms will be even more accurate, which in turn can probably improve the effectiveness of our method.

Future work may include extending our algorithm so that it can recognize gestures with motion and distinguish them from static gestures. This would enable the recognition of all letters/digits of finger alphabets.

Author Contributions: Conceptualization, D.W., T.K.; methodology, D.W.; software, D.W., T.K.; validation, D.W.; resources, D.W., T.K.; data curation, D.W.; writing—original draft preparation, D.W., T.K.; writing—review and editing, D.W., T.K.; visualization, D.W.; literature review, T.K. All authors have read and agreed to the published version of the manuscript.

Funding: This project is financed by the Minister of Education and Science of the Republic of Poland within the "Regional Initiative of Excellence" program for years 2019–2023. Project number: 027/RID/2018/19, amount granted: 11 999 900 PLN.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: In this study, publicly available datasets were analyzed. They can be found here: SPAS– http://vision.kia.prz.edu.pl (accessed on 13 November 2023), MU-ASL-digits—https://www.massey.ac.nz/~albarcza/gesture_dataset2012.html (accessed on 12 October 2023) The source codes of our methods can be found here: http://vision.kia.prz.edu.pl. (accessed on 13 November 2023)

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

References

1. Cheok, M.J.; Omar, Z.; Jaward, M.H. A review of hand gesture and sign language recognition techniques. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 131–153. [CrossRef]
2. Wang, Y.; Wang, D.; Fu, Y.; Yao, D.; Xie, L.; Zhou, M. Multi-Hand Gesture Recognition Using Automotive FMCW Radar Sensor. *Remote Sens.* **2022**, *14*, 2374. [CrossRef]
3. Khalaf, A.S.; Alharthi, S.A.; Dolgov, I.; Toups, Z.O. A Comparative Study of Hand Gesture Recognition Devices in the Context of Game Design. In Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces, New York, NY, USA, 10–13 November 2019 ; pp. 397–402.
4. Roberge, A.; Bouchard, B.; Maître, J.; Gaboury, S. Hand Gestures Identification for Fine-Grained Human Activity Recognition in Smart Homes. *Procedia Comput. Sci.* **2022**, *201*, 32–39. [CrossRef]
5. Cao, Z.; Hidalgo Martinez, G.; Simon, T.; Wei, S.; Sheikh, Y.A. OpenPose: Realtime Multi-Person 2D Pose Estimation using *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 172–186. . Part Affinity Fields. [CrossRef] [PubMed]
6. Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E.; Hays, M.; Zhang, F.; Chang, C.L.; Yong, M.; Lee, J.; et al. Mediapipe: A framework for perceiving and processing reality. In Proceedings of the Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 17 June 2019; Volume 2019.
7. Han, J.; Shao, L.; Xu, D.; Shotton, J. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Trans. Cybern.* **2013**, *43*, 1318–1334. [PubMed]
8. Guzsvinecz, T.; Szucs, V.; Sik-Lanyi, C. Suitability of the Kinect sensor and Leap Motion controller—A literature review. *Sensors* **2019**, *19*, 1072. [CrossRef]
9. Ansar, H.; Al Mudawi, N.; Alotaibi, S.S.; Alazeb, A.; Alabduallah, B.I.; Alonazi, M.; Park, J. Hand Gesture Recognition for Characters Understanding Using Convex Hull Landmarks and Geometric Features. *IEEE Access* **2023**, *11*, 82065–82078. [CrossRef]
10. Yasumuro, M.; Jin'no, K. Japanese fingerspelling identification by using MediaPipe. *Nonlinear Theory Its Appl. IEICE* **2022**, *13*, 288–293. [CrossRef]
11. Shin, J.; Matsuoka, A.; Hasan, M.A.M.; Srizon, A.Y. American sign language alphabet recognition by extracting feature from hand pose estimation. *Sensors* **2021**, *21*, 5856. [CrossRef]
12. Kenshimov, C.; Buribayev, Z.; Amirgaliyev, Y.; Ataniyazova, A.; Aitimov, A. Sign language dactyl recognition based on machine learning algorithms. *East.-Eur. J. Enterp. Technol.* **2021**, *4*, 112. [CrossRef]
13. Hagimoto, S.; Nitta, T.; Okada, R.; Nakanishi, T. A Dynamic Finger Character Recognition Method Using Landmark Behavior Rule Base. In Proceedings of the 2022 13th International Congress on Advanced Applied Informatics Winter (IIAI-AAI-Winter), Phuket, Thailand, 12–14 December 2022; pp. 189–195.
14. Sanalohit, J.; Katanyukul, T. Thai Finger Spelling Recognition: Investigating MediaPipe Hands Potentials. *arXiv* **2022**, arXiv:2201.03170.
15. John, J.; Sherif, B.V. Hand Landmark-Based Sign Language Recognition Using Deep Learning. In *Machine Learning and Autonomous Systems: Proceedings of ICMLAS 2021*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 147–157.
16. Ray, A.; Syed, S.; Poornima, S.; Pushpalatha, M. Sign language recognition using deep learning. *J. Pharm. Negat. Results* **2022**, *13*, 421–428. [CrossRef]
17. Rodríguez-Moreno, I.; Martínez-Otzeta, J.M.; Goienetxea, I.; Sierra, B. Sign language recognition by means of common spatial patterns: An analysis. *PLoS ONE* **2022**, *17*, e0276941. [CrossRef]
18. Hisham, E.; Saleh, S.N. ESMAANI: A Static and Dynamic Arabic Sign Language Recognition System Based on Machine and Deep Learning Models. In Proceedings of the 2022 5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA), Cairo, Egypt, 27–29 December 2022; pp. 1–6.
19. Fowley, F.; Ventresque, A. Sign Language Fingerspelling Recognition using Synthetic Data. In Proceedings of the AICS, Beijing, China, 29–31 July 2021; pp. 84–95.
20. Papadimitriou, K.; Potamianos, G.; Sapountzaki, G.; Goulas, T.; Efthimiou, E.; Fotinea, S.E.; Maragos, P. Greek sign language recognition for an education platform. *Univers. Access Inf. Soc.* **2023**, 1–18. [CrossRef]
21. Papadimitriou, K.; Potamianos, G. Sign Language Recognition via Deformable 3D Convolutions and Modulated Graph Convolutional Networks. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
22. Kushwaha, R.; Kaur, G.; Kumar, M. Hand Gesture Based Sign Language Recognition Using Deep Learning. In Proceedings of the 2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 26–28 May 2023; pp. 293–297.
23. Al-Hammadi, M.; Bencherif, M.A.; Alsulaiman, M.; Muhammad, G.; Mekhtiche, M.A.; Abdul, W.; Alohali, Y.A.; Alrayes, T.S.; Mathkour, H.; Faisal, M.; et al. Spatial attention-based 3d graph convolutional neural network for sign language recognition. *Sensors* **2022**, *22*, 4558. [CrossRef] [PubMed]
24. 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** . [CrossRef]
25. NC, G.; Ladi, M.; Negi, S.; Selvaraj, P.; Kumar, P.; Khapra, M. Addressing Resource Scarcity across Sign Languages with Multilingual Pretraining and Unified-Vocabulary Datasets. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 36202–36215.

26. Abdulhamied, R.M.; Nasr, M.M.; Abdulkader, S.N. Real-time recognition of American sign language using long-short term memory neural network and hand detection. *Indones. J. Electr. Eng. Comput. Sci.* **2023**, *30*, 545–556. [CrossRef]
27. Sheela, K.A.; Kumar, C.A.; Sandhya, J.; Ravindra, G. Indian Sign Language Translator. In Proceedings of the 2022 IEEE International Symposium on Smart Electronic Systems (iSES), Warangal, India, 18–22 December 2022; pp. 7–12.
28. Hanjar, S.; Rangannavar, V.; Pannagadhara, K.; Karthik, H.; Saravana, M. Vision Based Indian Sign Language Recognition Model. In *Perspectives in Communication, Embedded-systems and Signal-processing-PiCES*; Kashyap, N., Ed.; WorldServe Online: Stuttgart, Germany , 2022 ; pp. 63–67.
29. Xia, K.; Lu, W.; Fan, H.; Zhao, Q. A Sign Language Recognition System Applied to Deaf-Mute Medical Consultation. *Sensors* **2022**, *22*, 9107. [CrossRef]
30. Vijitkunsawat, W.; Racharak, T.; Nguyen, C.; Minh, N.L. Video-Based Sign Language Digit Recognition for the Thai Language: A New Dataset and Method Comparisons. In Proceedings of the Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2023, Lisbon, Portugal, 22–24 February 2023; Marsico, M.D., di Baja, G.S., Fred, A.L.N., Eds.; Scitepress: Setúbal, Portugal. 2023, pp. 775–782. [CrossRef]
31. Guo, Y.; Zhao, J.; Ding, B.; Tan, C.; Ling, W.; Tan, Z.; Miyaki, J.; Du, H.; Lu, S. Sign-to-911: Emergency Call Service for Sign Language Users with Assistive AR Glasses. In Proceedings of the 29th Annual International Conference on Mobile Computing and Networking, New York, NY, USA, 2–6 October 2023.
32. Tobias, J.L.; Di Mitri, D. Using Accessible Motion Capture in Educational Games for Sign language Learning. In Proceedings of the European Conference on Technology Enhanced Learning, Aveiro, Portugal, 4–8 September 2023; Springer: Cham, Switzerland, 2023; pp. 762–767.
33. Ivanska, L.; Korotyeyeva, T. Mobile real-time gesture detection application for sign language learning. In Proceedings of the 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 10–12 November 2022; pp. 511–514.
34. Kapuscinski, T. Handshape Recognition in an Educational Game for Finger Alphabet Practicing. In Proceedings of the International Conference on Intelligent Tutoring Systems, Bucharest, Romania, 29 June–1 July 2022; Springer: Cham, Switzerland, 2022; pp. 75–87.
35. Chandwani, L.; Khilari, J.; Gurjar, K.; Maragale, P.; Sonare, A.; Kakade, S.; Bhatt, A.; Kulkarni, R. Gesture based Sign Language Recognition system using Mediapipe. *Res. Sq.* **2023** . [CrossRef]
36. Cucurull, X.; Garrell, A. Continual Learning of Hand Gestures for Human-Robot Interaction. *arXiv* **2023**, arXiv:2304.06319.
37. Robert, E.J.; Duraisamy, H.J. A review on computational methods based automated sign language recognition system for hearing and speech impaired community. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7653. [CrossRef]
38. Mahalanobis, P.C. On the generalized distance in statistics. *Proc. Natl. Inst. Sci. Calcutta* **1936**, *2*, 49–55.
39. Galuska, L.; Garai, I.; Csiki, Z.; Varga, J.; Bodolay, E.; Bajnok, L. The clinical usefulness of the fingers-to-palm ratio in different hand microcirculatory abnormalities. *Nucl. Med. Commun.* **2000**, *21*, 659–663. . [CrossRef] [PubMed]
40. SPAS Dataset and Matlab scripts for the Recognition Method Presented in This Paper. Available online: http://vision.kia.prz.edu.pl (accessed on 12 July 2023).
41. Barczak, A.; Reyes, N.; Abastillas, M.; Piccio, A.; Susnjak, T. A New 2D Static Hand Gesture Colour Image Dataset for ASL Gestures. *Res. Lett. Inf. Math. Sci.* **2011**, *15*, 12–20.
42. Zhuang, H.; Yang, M.; Cui, Z.; Zheng, Q. A method for static hand gesture recognition based on non-negative matrix factorization and compressive sensing. *IAENG Int. J. Comput. Sci.* **2017**, *44*, 52–59.
43. Sahoo, J.P.; Ari, S.; Patra, S.K. Hand Gesture Recognition Using PCA Based Deep CNN Reduced Features and SVM Classifier. In Proceedings of the 2019 IEEE International Symposium on Smart Electronic Systems (iSES), Rourkela, India, 16–18 December 2019; pp. 221–224.
44. Fang, L.; Liang, N.; Kang, W.; Wang, Z.; Feng, D.D. Real-time hand posture recognition using hand geometric features and Fisher Vector. *Signal Process. Image Commun.* **2020**, *82*, 115729. [CrossRef]
45. Sahoo, J.P.; Sahoo, S.P.; Ari, S.; Patra, S.K. DeReFNet: Dual-stream Dense Residual Fusion Network for static hand gesture recognition. *Displays* **2023**, *77*, 102388. . [CrossRef]