ISSN 2007-9737

Enhancing Keypoint Selection for Hand Model in Recognition of Mexican Sign Language Alphabet

Jesús Javier Gortarez-Pelayo, Jesús Antonio Navarrete-López, Irvin Hussein Lopez-Nava*

Centro de Investigación Científica y de Educación Superior de Ensenada, Mexico

{jgortarez, jnavarrete, hussein}@cicese.edu.mx

Abstract. The foundation of learning any language lies in its alphabet, which serves as the basis for word formation and expression. This principle also applies to sign languages, where the manual alphabet is primarily used to spell proper nouns, technical terms, and words without established signs. Technological advancements now enable the tracking and interpretation of human movements—particularly hand gestures—facilitating the computational modeling of these alphabets. However, achieving efficient real-time performance remains a challenge, requiring the optimization of recognition models. This study improves the recognition of the Mexican Sign Language (LSM) alphabet by selecting the most critical hand landmarks, simplifying hand models while maintaining high performance (F1-score $>$ 0.94). The findings contribute to the development of lightweight and efficient systems for learning and deploying the LSM alphabet in real-time applications.

Keywords. Sign language recognition, Mexican sign language, LSM alphabet, hand landmarks, hand keypoints.

1 Introduction

Sign language recognition has emerged as a key area in computer vision, machine learning, and human-computer interaction, aiming to reduce communication barriers for Deaf communities. Mexican Sign Language (LSM), officially recognized in Mexico since 2005, is a full-fledged natural language with its own grammar, lexicon, and syntax [18]. One of its key components is the LSM manual alphabet, a set of 27 hand configurations representing the letters of the Spanish alphabet.

The LSM manual alphabet—commonly referred to as fingerspelling—is widely used to spell proper nouns, acronyms, technical terms, and lexical items without established signs. It also serves as a first point of contact for hearing individuals learning LSM, due to its visual correspondence with the written alphabet and its relatively straightforward memorization [4]. Combined with its standardized and compact articulation, these factors make it an attractive and widely adopted target in computational modeling and sign recognition systems.

Linguistic studies show that fingerspelling in LSM is structured around queiremas (handshapes), toponemas (articulation locations), and kinemas (movement components), with handshape as the key parameter for distinguishing letters [1]. These signs are typically articulated near the upper chest or shoulder, promoting clarity and visual accessibility. Most LSM alphabet signs are static with fixed handshapes, while a subset—including J, K, Ñ, Q, X, and Z—requires dynamic movements [19]. Static signs are especially relevant for computational recognition systems due to their standardized form and consistent spatial execution.

Recognition of LSM manual alphabet signs has been approached through various sensing modalities, each offering advantages for capturing handshape intricacies. Early efforts primarily used sensor-based systems, such as instrumented gloves, to obtain joint angles, finger flexion, and hand orientation data. While these systems provided high precision in controlled environments,

they were often intrusive, costly, and impractical for large-scale or real-time applications [15].

With advancements in computer vision, the research focus has increasingly shifted toward camera-based approaches. These vision-based systems rely on RGB or depth images to visually model hand gestures, offering a non-intrusive alternative. Within this paradigm, representations can vary across levels of abstraction—from raw pixel data capturing the full visual context, to mid-level descriptors such as segmented hand contours, or motion trajectories.

More recently, high-level representations based on anatomical keypoints or landmarks have gained prominence. These are typically extracted via pose estimation frameworks, which produce skeletal hand models. Landmark-based representations offer semantically rich and compact encodings of hand configurations, facilitating more robust and interpretable recognition pipelines. They also enhance generalization across varying backgrounds and lighting conditions.[10].

MediaPipe Hands [9] is one of the most widely used models for hand landmark extraction. This real-time framework by Google estimates 21 3D keypoints per hand from a single RGB image. The skeletal representation includes landmarks for each finger joint and key palm positions, providing a comprehensive encoding of hand posture. While extracting all keypoints yields detailed hand configurations, this granularity may be excessive for certain scenarios.

The human visual system often identifies objects and gestures using partial information, relying on salient features rather than processing every available detail. This phenomenon aligns with the Gestalt principle of Prägnanz, which states that perceptual systems favor simplified, coherent interpretations over exhaustive analysis. Similarly, in computational models, utilizing a full set of keypoints indiscriminately can introduce redundancy, increasing computational load without proportional improvements in recognition accuracy [7].

Recent studies in full-body pose estimation have demonstrated that dimensionality reduction strategies can preserve key discriminative features of human motion while enhancing model efficiency [2]. When applied to hand pose data, dimensionality reduction strategies yield compact and interpretable representations that enhance responsiveness in real-time, resource-constrained sign recognition systems without sacrificing sign recognition performance.

2 Related Work

Static fingerspelling recognition has been addressed using diverse sensing modalities, feature representations, and learning strategies. Within LSM, early systems have progressed from sensor- and image-based inputs to landmark-based approaches using pose estimation frameworks. For example, depth images and Haar-like features have been used for classifying a limited subset of signs [6], and geometric descriptors have been extracted from normalized hand volumes [16]. More recent studies have adopted RGB video pipelines combined with anatomical landmarks extracted via MediaPipe [12, 8, 17]. Beyond LSM, similar trends are seen in studies on American Sign Language (ASL) [21], Korean Sign Language (KSL) [20], Arabic Sign Language (ArSL) [13], and Assamese Sign Language [3], where keypoint vectors and convolutional models are frequently employed.

Existing studies vary widely in the selection of alphabet signs, dataset size, and capture conditions. For LSM, datasets have included 1,000 depth images per class for ten signs (A–E and digits 1–5) [6]; 21 static handshapes captured via depth sensing from 15 participants performed each sign once [16]; 91,326 labeled samples from 10 participants [12]; 200 grayscale images per class for 30 classes (21 alphabet signs and nine digits) [8]; and 8,100 samples from 10 participants [17]. Outside LSM, ASL datasets include 5,000 MediaPipe-processed frames [21], KSL datasets consist of 3,200 labeled images [20], and datasets for ArSL and Assamese Sign Language feature between 5,600 and 6,400 gesture images [3, 13].

The representation of hand configurations differs substantially across systems. Depth- and grayscale-based studies have relied on handcrafted features such as Haar-like templates [6], geometric descriptors [16], and histogram-based

methods [20]. More recent approaches leverage anatomical landmarks extracted by MediaPipe to create normalized vectors of hand keypoints [12, 17, 8, 21, 3, 13]. Some systems incorporate hybrid representations by combining handcrafted descriptors with CNN-derived visual features [20]. Dimensionality reduction is addressed explicitly through PCA [16, 20] or implicitly through neural bottlenecks and pooling operations [17, 3, 13].

Classification strategies vary across systems. Classical machine learning models, including AdaBoost [6] and SVM [16], are employed in early handcrafted-feature pipelines, achieving high accuracy. Landmark-based systems report high performance using classifiers such as kNN, Random Forest, Naive Bayes, and SVM [12], with F1-scores exceeding 0.98 for most classes. CNN-based architectures trained on normalized keypoints or matrix-structured landmark data achieve between 96% and 98% accuracy [17, 13], and dense feedforward networks yield comparable results [3]. Feature-level fusion of handcrafted and learned representations has achieved up to 99% accuracy in static sign recognition [20].

Despite promising results, current systems exhibit a notable limitation in their reliance on the full set of features and fail to provide an explicit optimization or rationale regarding the minimal information required to perform the task. Furthermore, evaluations are commonly restricted to internal cross-validation, offering limited insight into performance under information loss—a frequent problem in real-world scenarios. Specifically, landmark-based approaches typically depend on the complete set of 21 MediaPipe keypoints, despite the absence of studies assessing the relevance of individual landmarks or exploring reduced configurations. This dependence often leads to unnecessarily dense representations that are suboptimal for real-time or resource-constrained deployment. Consequently, the lack of attention to compact, task-specific feature sets represents a key gap that this work aims to address.

In light of these insights, this study investigates whether reduced landmark representations can maintain classification accuracy in recognizing static signs from the LSM manual alphabet. We systematically evaluate the contribution of each of the 21 hand keypoints extracted by MediaPipe to identify a minimal subset with high discriminative capacity. This addresses a gap in prior work, where full landmark sets are typically used without assessing their relevance. Using an iterative feature selection strategy, we derive a compact configuration that improves interpretability and computational efficiency.

3 Methods

This study aims to identify the most relevant subset of features for recognizing static signs from the LSM manual alphabet. The overall methodology, summarized in Figure 1, follows a four-stage workflow comprising dataset construction, feature extraction, feature selection, and classification.

The dataset consists of annotated images representing 21 static hand configurations from the LSM alphabet, excluding dynamic letters. For each image, 21 anatomical hand keypoints are extracted using the MediaPipe model, retaining only the two-dimensional (X, Y) coordinates. These values are then standardized using Z-score normalization to ensure comparability across samples and reduce variability due to scale or position. To identify the most informative keypoints, a feature selection strategy is applied using four ranking techniques. Their results are aggregated into a consensus score to establish a global relevance ordering. Model performance is then evaluated using incrementally larger subsets of top-ranked features and compared against a Principal Component Analysis (PCA) baseline. Classification is performed using a Support Vector Machine (SVM), with hyperparameters optimized via the Optuna framework. Evaluation is conducted through Leave-One-Group-Out cross-validation, where each group corresponds to an individual participant, ensuring signer-independent generalization.

3.1 Corpus

The dataset used in this study is the *Mexican Sign Language Alphabet (static signs)* corpus [12],

ISSN 2007-9737

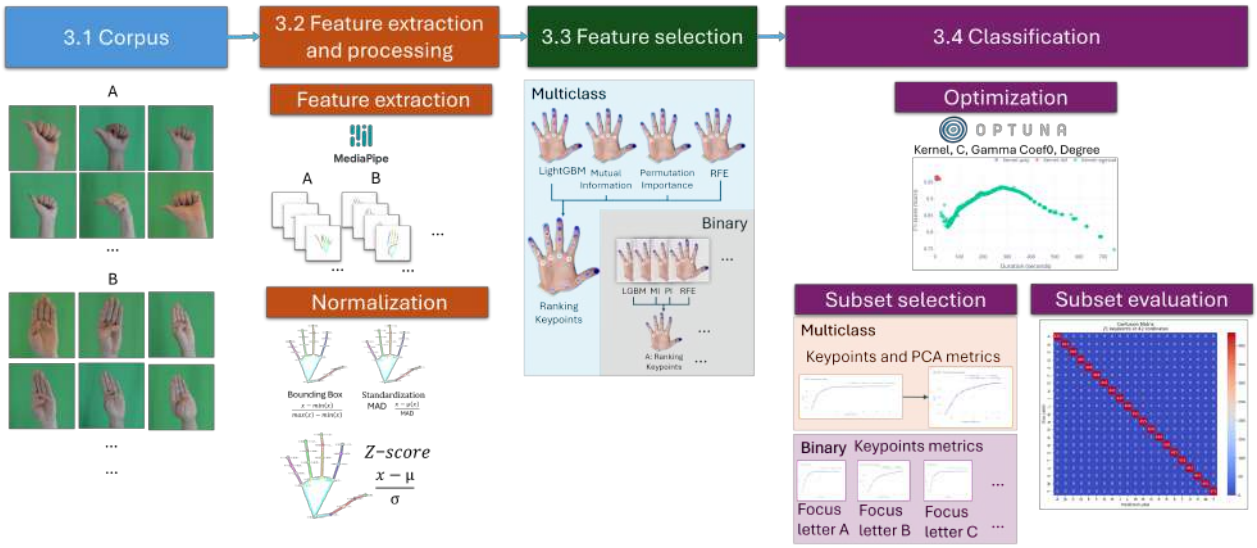


Fig. 1. Proposed overall methodology divided into four stages: (3.1) dataset preparation, (3.2) feature extraction and processing, (3.3) feature selection, and (3.4) LSM manual alphabet classification

which comprises 21 static handshapes corresponding to letters from the LSM manual alphabet. Images were captured using a uniform green background under controlled lighting conditions, each with a resolution of 360 × 360 pixels. Data were collected from 20 participants. This study exclusively uses the subset referred to as `Group A`, which includes samples with slight hand rotations across all three axes. This controlled variability accurately represents the real-world variability present within LSM without the orientation changing the meaning of the sign. Group A contains a total of 92,703 images. Of these, 83,264 images from 18 participants are provided for training, while 9,439 images from two held-out participants are reserved for testing.

3.2 Feature Extraction and Processing

Keypoint extraction was performed using the MediaPipe HandLandmarker model with default parameters. As previously noted, MediaPipe detects 21 3D hand landmarks per frame, providing normalized coordinates where X and Y are relative to the image width and height, and Z encodes depth with respect to the hand's palm plane.

Only the X and Y coordinates were retained in this study. Depth (Z) values were excluded due to their sensitivity to minor changes in camera distance and their limited contribution to the classification of static handshapes, where the two-dimensional spatial configuration of fingers is the most distinctive feature. MediaPipe failed to detect the hand in approximately 0.5% of the images. Figure 2 illustrates the 21 keypoints and their skeletal configuration for all alphabet signs. Each coordinate pair was systematically labeled using a predefined nomenclature, as shown in the figure. These features were further annotated with the target letter label and a subject identifier parsed from the image filename.

3.2.1 Normalization

This step is essential to ensure that numerical representations are comparable across samples, minimizing the influence of scale, position, or potential distortions. In this study, three commonly used normalization techniques were evaluated: Min-Max scaling, Median Absolute Deviation (MAD), and Z-score normalization. Preliminary experiments revealed that Z-score normalization yielded the most consistent results, as it centers

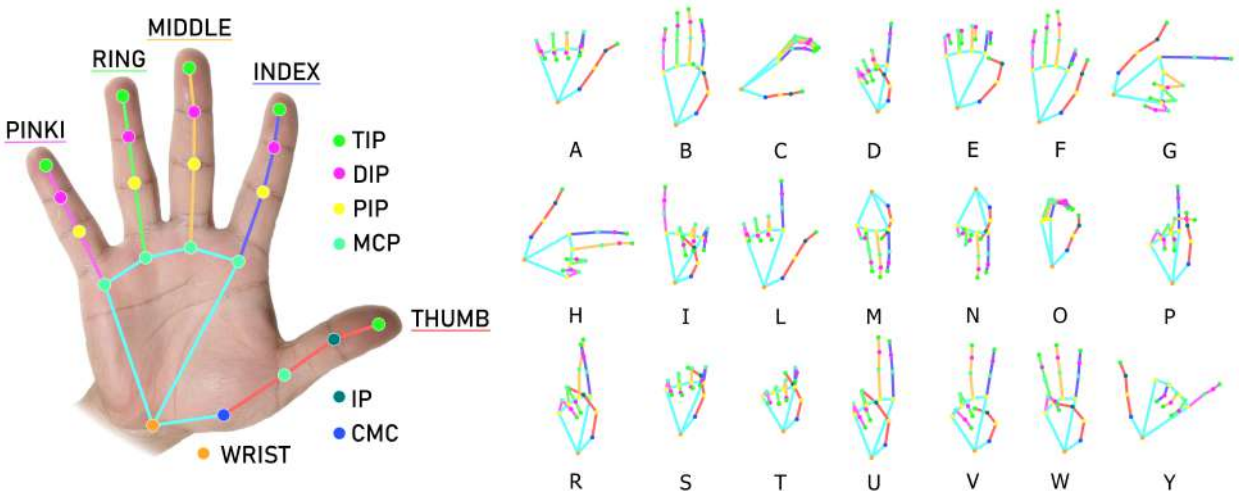


Fig. 2. Keypoints with nomenclature (left) and skeletal hand structure for 21 static LSM signs (right)

the data and scales it to a uniform dispersion. Let X and Y represent the vectors of 2D coordinates for the horizontal and vertical axes, respectively, extracted from a single sample's keypoint representation. Z-score normalization is applied independently to each vector as follows:

$$X^* = \frac{X - \mu(X)}{\sigma(X)}, \quad Y^* = \frac{Y - \mu(Y)}{\sigma(Y)}, \qquad (1)$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and standard deviation of the corresponding coordinate vector. The final normalized feature vector is obtained by interleaving the standardized components as $V^* = [X^*, Y^*] \in \mathbb{R}^{2n}$, where n is the number of keypoints.

This technique is particularly suitable when the relevant information lies in the relative configuration of the points rather than their absolute values. Moreover, it ensures that classifiers sensitive to input scales, such as SVM, are not biased by differences in magnitude across features.

3.3 Feature Selection

To identify the most informative keypoints for representing the LSM manual alphabet, a feature ranking process was conducted using both multiclass (all letters jointly) and binary (one-vs-rest) approaches, based on Z-score normalized landmark coordinates.

In the binary approach, labels are binarized (1 for the target letter, 0 for all others), while the multiclass strategy preserves the original categorical labels. Four complementary techniques were employed to estimate feature relevance, each grounded in a different evaluation principle:

- Mutual Information: Quantifies the statistical dependency between each feature and the target variable, capturing potential nonlinear associations.
- LightGBM-based Importance: Uses gradient-boosted decision trees to compute the contribution of each feature to predictive performance.
- Recursive Feature Elimination with Random Forests: Iteratively removes the least important features based on their impact on model performance.
- Permutation Importance: Measures the drop in model accuracy when a feature's values are randomly shuffled, estimating its marginal contribution.

To consolidate these perspectives—statistical dependence, tree-based modeling, recursive elimination, and perturbation—the resulting scores were individually normalized via Min-Max scaling, then averaged and rescaled to yield the final ranking.

Table 1. Hyperparameters explored for SVM model optimization

Hyperparameter	Explored values	Applicable to
Kernel type	RBF, polynomial, sigmoid	General
C (regularization)	1 to 100 (step 0.5)	General
γ	`scale`, `auto`	General
coef_0	0.0 to 3.0 (step 0.1)	Polynomial and sigmoid
Polynomial degree	2 to 6 (integer values)	Polynomial

3.4 Classification

Following the identification of the most relevant hand keypoints, the final stage evaluates the performance of an SVM classifier trained on feature subsets of varying sizes. This stage aims to validate whether compact representations maintain high classification performance and generalization across signers, particularly under realistic deployment conditions.

The choice of SVM was guided by a prior comparative study in which multiple classifiers were evaluated for LSM handshape recognition, with SVM achieving the highest performance across models [12]. The dataset used corresponds to the Z-score normalized data described previously. Hyperparameter tuning was performed using the Optuna framework, with the objective of maximizing the macro-averaged F1-score, which balances performance across all 21 classes.

The search explored three SVM kernels—Radial Basis Function (RBF), polynomial, and sigmoid—along with their respective hyperparameters: regularization coefficient C, kernel coefficient γ, polynomial degree (for *poly*), and additive coefficient *coef0*. Table 1 summarizes the parameter ranges used during optimization. To ensure reproducibility and comparability across trials, the random seed was fixed at 42, the maximum iteration count was set to 10,000, and cache size was limited to 2000 MB.

The data from all 18 training participants and the 2 test participants provided in the dataset repository were combined to perform a generalized evaluation. Model evaluation was performed using Leave-One-Group-Out cross-validation, where each group corresponds to a different participant. This strategy enables a signer-independent assessment and prevents overfitting to individual users, simulating deployment in real-world conditions.

After exhaustive evaluation, the polynomial kernel SVM with $C = 4.5$, *degree*$= 4$, *gamma*=*scale*, and *coef0*= 1.1 was selected as optimal. Although other models achieved slightly higher accuracy (within 0.02), this configuration offered a strong balance between performance, generalization, and computational efficiency—making it suitable for real-time use.

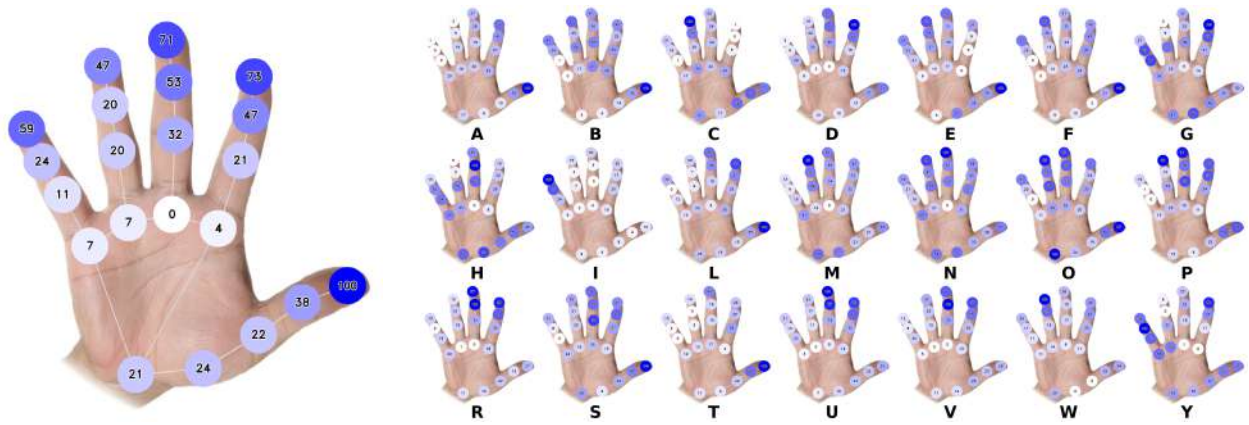
4 Results

To identify the most informative keypoint subsets, scores from each feature selection technique were first individually rescaled using Min-Max normalization, then averaged and normalized again to produce the final ranking. The aggregated results from the multiclass selection strategy are shown in Subfigure 3a. For the binary analysis, an independent experiment was conducted for each letter by binarizing the entire dataset with respect to the target letter and repeating the ranking process; the corresponding results are presented in Subfigure 3b.

As observed, the relevance of each keypoint depends on the specific handshape being signed. For example, the thumb plays a central role in letters A, B, and E; the index finger is crucial for letters D and G; and the pinky is most informative for letter I, which aligns with intuitive expectations. When analyzed globally, the most important keypoints are `THUMB_TIP`, `INDEX_TIP`, `MIDDLE_TIP`, and `PINKY_TIP` (see Figure 2).

Building upon the ranked keypoints, several SVM classification models were trained to comprehensively evaluate the performance of different keypoint subsets. Additionally, Principal Component Analysis (PCA) was applied as a dimensionality reduction technique to generate alternative feature sets, providing a comparative baseline to the feature selection approach.

The evaluation iterated through feature subsets, starting with one keypoint and progressively adding up to all 21. For each selected keypoint (e.g., `WRIST`), both corresponding coordinates (`WRIST_X` and `WRIST_Y`) were automatically included in the training set. In the PCA-based approach, models



(a) Multiclass **(b)** Binary

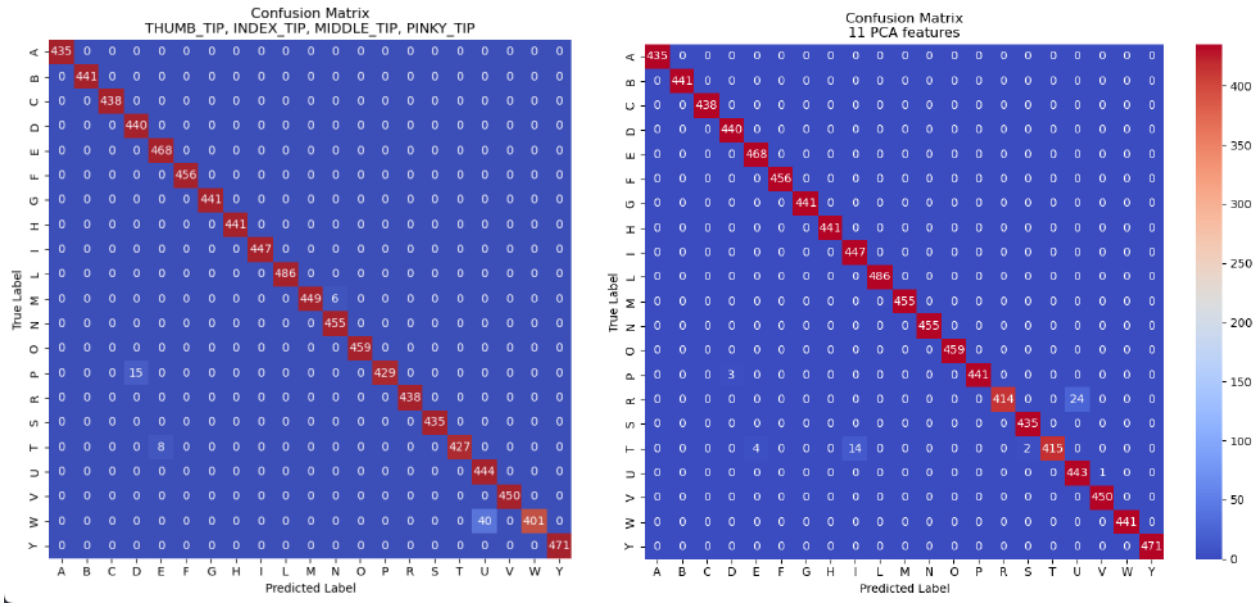
Fig. 3. Keypoints-based rankings: The values represent the final aggregated importance score, scaled from 0 (least important) to 100 (most important). This score is derived from the four methods detailed in Section 3.3 which aggregate the relevance of the individual keypoints: (a) all-signs, and (b) individually

Fig. 4. Macro F1-score comparison between the 21 keypoints and PCA components

were trained and evaluated on the respective transformed feature subsets using the same incremental strategy. Classification performance was assessed using the macro-averaged F1-score (harmonic mean of precision and recall), as shown in Figure 4.

As shown in the performance curve, classification improves as the number of features increases, with both approaches converging in F1-score from around 11 features onward. Since effectiveness is not the only relevant factor in this task, we also measured training and inference times, as well as accuracy for comparison with related work. Detailed results are presented in Table 2. Although it might seem intuitive that models with fewer features would train and infer faster, the opposite trend is observed here. This behavior can occur because models trained on extremely small feature sets may require more iterations to converge, as decision boundaries become less separable in low-dimensional spaces [5].

Furthermore, when the feature space is too limited, kernel-based methods often rely on more complex transformations to achieve separability, which can increase computational overhead. Consequently, our models with fewer than three



(a) Top 4 keypoints

(b) Top 11 PCA features

Fig. 5. Confusion matrices using compact feature subsets

features exhibit longer training and inference times, suggesting that excessive feature reduction may hinder efficient learning despite the smaller input dimensionality.

The keypoint-based models achieve near-maximum performance (F1-score of 0.95) with just 10 features, with results stabilizing beyond this point. In contrast, the PCA-based model improves gradually, matching the keypoint model's performance only after 17 components. In terms of computational efficiency, keypoint models are faster in both training and inference. Notably, the shortest inference time (0.316 seconds) is achieved with eight keypoint features—faster than any PCA setup. Specifically, for the 4-feature setting, the PCA model yields an F1-score two tenths lower while requiring five times more training time and ten times more inference time than its keypoint counterpart.

Finally, a detailed per-class analysis was conducted to evaluate recognition performance across all 21 static LSM alphabet letters. Figure 5 presents confusion matrices for the compact model using the top 4 keypoints and the PCA-based model with 11 principal components. As a reference, the best overall performance is achieved with more than 19 keypoints (see Table 2), where classification errors are minimal and primarily limited to slight confusion (fewer than three errors per class) between letters *U* and *V*.

When reducing the feature set to the top 4 keypoints (Figure 5a), overall performance remains high, but specific confusions emerge. The most prominent occurs with letter *W*, which is frequently misclassified as *U*. Additional misclassifications include letter *P* being predicted as *D*. Similarly, the PCA-based model using 11 components (Figure 5b) also yields high accuracy, though with a different error pattern. In this case, the most frequent confusion involves letter *R* being classified as *U*, and to a lesser extent, letter *T* misidentified as *I*.

Additional experiments assessed classification performance when treating individual coordinate axes separately. This approach yielded improved results using a subset of 11 coordinates—namely `THUMB_TIP_X`, `INDEX_TIP_Y`, `MIDDLE_TIP_Y`, `THUMB_TIP_Y`, `PINKY_TIP_Y`, `MIDDLE_DIP_Y`, `RING_TIP_Y`, `MIDDLE_PIP_Y`,

Table 2. Performance metrics for SVM classification using selected keypoints and PCA-based feature subsets. KP: keypoints, PCA: principal component analysis

Num features	Accuracy		F1-score		Training Time (s)		Inference Time (s)	
	KP	PCA	KP	PCA	KP	PCA	KP	PCA
1	0.410	0.204	0.363	0.139	63.094	128.928	17.601	21.938
2	0.811	0.444	0.799	0.402	10.323	50.745	2.999	17.697
3	0.928	0.652	0.924	0.624	3.474	21.965	0.766	11.079
4	0.944	0.754	0.937	0.734	2.537	13.522	0.534	5.160
5	0.949	0.805	0.942	0.789	2.321	8.869	0.418	2.986
6	0.950	0.847	0.942	0.832	2.225	6.728	0.396	1.975
7	0.950	0.886	0.943	0.873	2.244	5.147	0.365	1.355
8	0.953	0.914	0.947	0.905	2.199	3.836	0.316	0.927
9	0.955	0.940	0.948	0.936	2.241	3.023	0.323	0.578
10	0.956	0.940	0.950	0.936	2.275	2.885	0.324	0.522
11	0.958	0.946	0.955	0.942	2.361	2.707	0.346	0.448
12	0.958	0.948	0.956	0.943	2.429	2.637	0.355	0.426
13	0.960	0.948	0.957	0.943	2.511	2.609	0.366	0.426
14	0.960	0.947	0.957	0.942	2.616	2.617	0.406	0.396
15	0.959	0.951	0.956	0.943	2.693	2.639	0.418	0.401
16	0.959	0.950	0.956	0.940	2.389	2.657	0.347	0.373
17	0.959	0.951	0.947	0.947	2.524	2.579	0.370	0.356
18	0.959	0.952	0.956	0.950	2.556	2.615	0.374	0.360
19	0.960	0.954	0.957	0.950	2.670	2.655	0.390	0.357
20	0.960	0.956	0.957	0.954	2.774	2.610	0.395	0.341
21	0.960	0.958	0.957	0.955	2.899	2.633	0.409	0.345

`INDEX_DIP_Y`, `INDEX_TIP_X`, and `MIDDLE_TIP_X`. While this configuration surpassed the four-keypoint model in performance (F1 score macro = 0.944), it required a larger number of features, some of which extend beyond the compact subset, introducing additional complexity.

We observe that the results in the present work yield an F1-score of 0.94, which is slightly lower than that reported in previous studies [12, 14]. Regarding [12], this discrepancy is mainly due to the fact that the current study performs an exhaustive analysis on a subset of keypoints and optimizes the inference algorithm's hyperparameters, whereas [12] utilized default parameters. This indicates that our study achieves results comparable to the previous work while utilizing fewer features. On the other hand, concerning [14], the difference arises because their strategy is based on feature extraction from video sequences where MediaPipe employs a previous Region of Interest (ROI) as a reference, while the current work utilized the analysis of static signs as a starting point. Despite these variations, the synthesis of the three studies has allowed us to establish a minimum keypoint subset configuration, a minimum number of frames for continuous recognition, and the optimization of hyperparameters for the best performing recognition model. These findings are fundamental for the development of future real-time recognition tools.

5 Conclusions

This study investigated whether a compact subset of hand landmarks could support recognition of static signs from the LSM manual alphabet. Through systematic feature selection and comparison with PCA-based reduction, we showed that lightweight representations using few keypoints maintain high classification performance while significantly improving computational efficiency.

The confusion errors observed in the classification matrices reflect inherent gestural similarities across LSM manual alphabet signs. The most prominent confusions—U versus V, W versus U, and R versus U—are consistent with the signs' shared manual configurations, where differences

lie in the number of extended fingers or in subtle orientation shifts that require fine-grained positional detail. Likewise, confusion between P and D can be attributed to the proximity of the index finger and similar wrist orientation.

Although the proposed representation was developed for one-handed static signs in LSM, it may serve as a foundational framework for broader applications in sign language recognition. For example, alphabets in languages such as British Sign Language (BSL) involve both hands, introducing further spatial dependencies that could benefit from similar compact modeling. Dynamic signs—like the six LSM letters involving motion—require temporal analysis across frame sequences, where reducing spatial dimensionality can facilitate real-time performance. Furthermore, recognizing full ideograms or lexical signs often involves multimodal cues, such as facial expressions and upper-body posture, highlighting the need for scalable and efficient representations [11].

Acknowledgments

We thank the Ministry of Science, Humanities, Technology, and Innovation (SECIHTI) for the graduate grants 1351117 (JJGP) and 1294009 (JANL).

References

1. **Aldrete, M. C. (2008).** Gramática de la Lengua de Señas Mexicana. Ph.D. thesis, El Colegio de México.
2. **Arya, V., Maji, S. (2024).** Enhancing human pose estimation: A data-driven approach with mediapipe blazepose and feature engineering analysing. First International Conference on Pioneering Developments in Computer Science & Digital Technologies, IEEE, pp. 1–6.
3. **Bora, J., Dehingia, S., Boruah, A., Chetia, A. A., Gogoi, D. (2023).** Real-time assamese sign language recognition using mediapipe and deep learning. Procedia Computer Science, Vol. 218, pp. 1384–1393.
4. **Gortarez-Pelayo, J. J., Morfín-Chávez, R. F., Lopez-Nava, I. H. (2023).** DAKTILOS: An Interactive Platform for Teaching Mexican Sign Language (LSM). Int. Conference on Ubiquitous Computing and Ambient Intelligence, Springer, pp. 264–269.
5. **Guyon, I., Elisseeff, A. (2003).** An introduction to variable and feature selection. Journal of Machine Learning Research, Vol. 3, pp. 1157–1182.
6. **Jimenez, J., Martin, A., Uc, V., Espinosa, A. (2017).** Mexican sign language alphanumerical gestures recognition using 3d haar-like features. IEEE Latin America Transactions, Vol. 15, No. 10, pp. 2000–2005.
7. **Kirupakaran, A. M., Laskar, R. H. (2024).** Scale-adaptive gesture computing: detection, tracking and recognition in controlled complex environments. Machine Vision and Applications, Vol. 35, No. 4, pp. 75.
8. **López-Vázquez, F. J., Guerrero-Osuna, H. A., Nava-Pintor, J. A., Olvera-Olvera, C. A., Díaz-Flórez, G., Vega, L. F. L. (2024).** Development of a recognition system for the mexican sign language alphabet. 2024 IEEE International Autumn Meeting on Power, Electronics and Computing, IEEE, Vol. 8, pp. 1–7.
9. **Lugaresi, C., Tang, J., Nash, H., McGuire, C., Chang, Y., Yong, M., Lee, J., Grundmann, M. (2019).** Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172.
10. **Madhiarasan, M., Roy, P. P. (2022).** A comprehensive review of sign language recognition: Different types, modalities, and datasets. arXiv preprint arXiv:2204.03328.
11. **Martinez-Seis, B., Pichardo-Lagunas, O., Hernández-Morales, E., Rivera-Rodríguez, O., Miranda, S. (2025).** Automatic translation of sentences to mexican sign language: Rule-based machine translation and animation synthesis in avatar. Computación y Sistemas, Vol. 29, No. 1, pp. 145–155.

12. **Morfín-Chávez, R. F., Gortarez-Pelayo, J. J., Lopez-Nava, I. H. (2023).** Fingerspelling recognition in Mexican sign language (LSM) using machine learning. Mexican International Conference on Artificial Intelligence, Springer, pp. 110–120.

13. **Moustafa, A. M. A., Mohd Rahim, M. S., Bouallegue, B., Khattab, M. M., Soliman, A. M., Tharwat, G., Ahmed, A. M. (2023).** Integrated mediapipe with a CNN model for arabic sign language recognition. Journal of Electrical and Computer Engineering, Vol. 2023, No. 1, pp. 8870750.

14. **Navarrete-López, J. A., Gortarez-Pelayo, J. J., Lopez-Nava, I. H. (2025).** Dynamic strategy for recognizing the mexican sign language alphabet: Bridging static and dynamic signs. Pattern Recognition, Springer Nature Switzerland, Cham, pp. 113–122. DOI: 10.1007/978-3-031-96255-4_11.

15. **Rastgoo, R., Kiani, K., Escalera, S. (2021).** Sign language recognition: A deep survey. Expert Systems with Applications, Vol. 164, pp. 113794.

16. **Rios-Figueroa, H. V., Sánchez-García, A. J., Sosa-Jiménez, C. O., Solís-González-Cosío, A. L. (2022).** Use of spherical and cartesian features for learning and recognition of the static mexican sign language alphabet. Mathematics, Vol. 10, No. 16, pp. 2904.

17. **Sánchez-Vicinaiz, T. J., Camacho-Pérez, E., Castillo-Atoche, A. A., Cruz-Fernandez, M., García-Martínez, J. R., Rodríguez-Reséndiz, J. (2024).** Mediapipe frame and convolutional neural networks-based fingerspelling detection in mexican sign language. Technologies, Vol. 12, No. 8, pp. 124.

18. **Secretaría de Gobernación (2005).** Ley General para la Inclusión de las Personas con Discapacidad. Diario Oficial de la Federación, México.

19. **Serafín, M. E., González Pérez, R. (2011).** Manos con voz: Diccionario de Lengua de Señas Mexicana. Consejo Nacional para Prevenir la Discriminación, México.

20. **Shin, J., Miah, A. S. M., Akiba, Y., Hirooka, K., Hassan, N., Hwang, Y. S. (2024).** Korean sign language alphabet recognition through the integration of handcrafted and deep learning-based two-stream feature extraction approach. IEEE Access, Vol. 12, pp. 68303–68318.

21. **Sundar, B., Bagyammal, T. (2022).** American sign language recognition for alphabets using mediapipe and lstm. Procedia Computer Science, Vol. 215, pp. 642–651.

Article received on 27/08/2025; accepted on 02/12/2025.
**Corresponding author is Irvin Hussein Lopez-Nava.*