Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.6, December 2025, pp. 3904~3918
e-ISSN: 2548-6861

Comparative Analysis of 1D CNN Architectures for Guitar Chord Recognition from Static Hand Landmarks

Rafi Abhista Naya [1]*, Evan Tanuwijaya [2]*
* Informatika, School of Information Technology, Universitas Ciputra
rafiabhista.naya@gmail.com [1], evan.tanuwijaya@ciputra.ac.id[2]

Article Info

Article history:

Received 2025-09-24
Revised 2025-12-04
Accepted 2025-12-10

Keyword:

Guitar Chord Recognition,
Hand Landmarks,
1D CNN,
Mediapipe,
Computer Vision,
Music Technology.

ABSTRACT

Vision-based guitar chord recognition offers a promising alternative to traditional audio-driven methods, particularly for silent practice, classroom environments, and interactive learning applications. While existing research predominantly relies on full-frame image analysis using 2D convolutional networks, the use of structured hand landmarks remains underexplored despite their advantages in robustness and computational efficiency. This study presents a comprehensive comparative analysis of three one-dimensional convolutional neural network architectures—CNN-1D, ResNet-1D, and Inception-1D—for classifying seven guitar chord types using 63-dimensional static hand-landmark vectors extracted via MediaPipe Hands. The methodology encompasses extensive dataset preprocessing, targeted landmark augmentation, Bayesian hyperparameter optimization, and stratified 5-fold cross-validation. Results show that CNN-1D achieves the highest mean accuracy (97.61%), outperforming both ResNet-1D and Inception-1D, with statistical tests confirming significant improvements over ResNet-1D. Robustness experiments further demonstrate that CNN-1D maintains superior resilience under Gaussian noise, landmark occlusion, and geometric scaling. Additionally, CNN-1D provides the fastest inference and most stable computational performance, making it highly suitable for real-time or mobile deployment. These findings highlight that, for structured and low-dimensional landmark data, simpler convolutional architectures outperform deeper or multi-branch designs, offering an efficient and reliable solution for vision-based guitar chord recognition.

This is an open access article under the CC–BY-SA license.

I. INTRODUCTION

The ability to recognize guitar chords from hand pose information addresses several practical applications in modern music education and technology[1]. Silent practice applications represent a significant use case, where musicians require feedback on their chord formations without generating audio output[1]. Traditional electric guitar practice through amplifiers or acoustic instruments can be disruptive in apartment settings, shared living spaces, or late-night practice sessions[1]. Vision-based chord recognition systems enable musicians to receive real-time feedback on their fingering patterns through headphones or visual displays, eliminating noise concerns while maintaining effective practice sessions[1]. Classroom environments present another compelling application domain. Music educators increasingly integrate technology to enhance learning experiences, particularly in scenarios where multiple students practice simultaneously[1]. Vision-based chord recognition systems can provide individualized feedback to students without audio interference from neighboring learners[1]. Research on augmented reality piano tutoring systems demonstrates the potential for visual feedback mechanisms in music education, showing improved engagement and accelerated learning for beginner students[1].

The convergence of computer vision and music technology has opened promising avenues for enhancing musical learning and practice experiences[1]. Guitar chord recognition, traditionally reliant on audio analysis, presents unique opportunities for vision-based approaches that can address

limitations inherent in acoustic methods. While audio-based chord recognition has achieved considerable maturity over the past two decades, with sophisticated systems employing pitch class profiles, hidden Markov models, and deep learning architectures, vision-based approaches remain in their infancy[2], [3]. Existing audio chord recognition systems demonstrate high accuracy on clean recordings but struggle with polyphonic music, background noise, and real-time performance constraints[2], [3]. These limitations motivate the exploration of complementary visual approaches.

Current vision-based guitar chord recognition research primarily focuses on full image analysis using traditional 2D convolutional neural networks applied to raw camera footage[4], [5]. These approaches achieve promising results but require substantial computational resources and are sensitive to lighting conditions, camera angles, and background variations[4], [5]. Studies report accuracies ranging from 83% to 97% on controlled datasets, but performance degrades significantly in real-world conditions[5].

Landmark-only approaches remain significantly underexplored in the guitar chord recognition domain. While MediaPipe Hands provides robust hand landmark detection with 21 anatomically-defined keypoints, few studies investigate how effectively these structured representations can be leveraged for chord classification [6]. This represents a critical gap, as landmark-based approaches offer several advantages: (1) reduced computational requirements, (2) invariance to lighting and background conditions, and (3) direct encoding of the anatomical relationships essential for chord formation[6].

Furthermore, there exists uncertainty regarding which 1D CNN architectures perform optimally on structured landmark inputs. While 1D CNNs have demonstrated effectiveness on time-series data, sensor readings, and sequential patterns, their application to spatially-ordered anatomical landmarks requires careful consideration of architectural choices[5], [7], [8]. Different 1D CNN families—including traditional architectures, residual networks, and attention-based models—may exhibit varying performance characteristics when processing hand landmark sequences.

Early vision-based guitar chord recognition systems focused on fretboard analysis and finger position detection using traditional computer vision techniques. Mitjans Coma developed a deep learning approach using 2-stack Hourglass networks for chord detection, achieving 97% accuracy on a controlled dataset of 205 images[5]. However, this approach required clear visibility of the entire fretboard and was sensitive to camera positioning and lighting conditions.

Recent research has shifted toward hand-centric approaches that focus on finger configurations rather than fretboard analysis. Studies by Ooaku et al. introduced deep learning methods for recognizing finger patterns in guitar playing videos, demonstrating the feasibility of chord recognition through hand pose analysis[7]. Similarly, research projects documented in IEEE conferences explore convolutional neural network architectures for real-time guitar chord identification, reporting accuracies between 83% and 89% on diverse datasets[8].

The development of robust hand landmark detection systems has been revolutionized by MediaPipe Hands, which provides real-time detection of 21 hand keypoints with high accuracy across diverse conditions. These landmarks encode finger joint positions, palm center, and wrist location, providing a structured representation of hand configuration that is invariant to scale, translation, and background variations[6].

Hand pose-based action recognition has gained significant attention in computer vision research. Doosti et al.[9] proposed HOPE-Net, a graph-based approach for joint hand-object pose estimation using graph convolutional networks. Their work demonstrates that graph-based architectures can effectively model the relationships between hand landmarks for complex recognition tasks[9]. However, graph neural networks require specialized implementations and may be computationally intensive for real-time applications.

Alternative approaches using recurrent neural networks and attention mechanisms have also been explored. These methods treat hand landmarks as temporal sequences, leveraging LSTM or transformer architectures to model the relationships between keypoints. While effective for dynamic gesture recognition, their applicability to static chord recognition requires careful consideration of the spatial rather than temporal relationships between landmarks[10].

This research investigates the application of one-dimensional convolutional neural networks (1D CNNs) to guitar chord recognition using static hand landmarks, contributing to the emerging field of vision-based musical instrument analysis. This comparative analysis also aims to provide comprehensive insights into the trade-offs between architectural complexity, computational requirements, and recognition accuracy, ultimately guiding practitioners toward optimal solutions for vision-based guitar chord recognition systems.

II. METHODS

This chapter presents a comprehensive methodology for guitar chord recognition using hand landmark detection and deep learning architectures. The proposed approach transforms the traditional computer vision problem of chord recognition from guitar images into a structured classification task based on hand pose estimation. The methodology encompasses three principal phases: exploratory analysis and dataset preprocessing, hand landmark extraction with data augmentation, and comparative evaluation of neural network architectures optimized for temporal landmark sequences. The experimental protocol followed a systematic approach as shown in Figure 1.

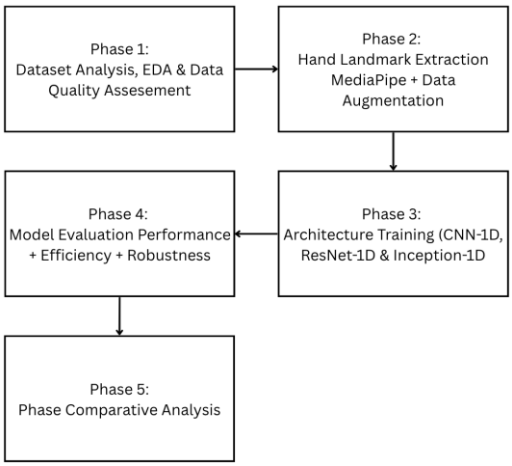


Figure 1. Experimental design pipeline illustrating the five research phases: (1) dataset analysis and preprocessing, (2) hand landmark extraction and augmentation, (3) model training and optimization, (4) evaluation and cross-architecture comparison, and (5) result analysis and recommendations.

A. Dataset Preparation and Exploratory Data Analysis

1) *Dataset Characteristics:* This study utilized the Bilkent CS-464 Guitar Chord Dataset, sourced from Roboflow Universe, containing approximately 3,784 annotated images distributed across seven chord classes (A, B, C, D, E, F, G). The dataset is formatted using the standard YOLO structure with bounding-box annotations for the fretting hand. However, the dataset does not provide metadata regarding the data collection process, including the number of subjects, recording conditions, or camera specifications[11]. To address this gap, a qualitative visual inspection was performed. Based on this inspection, the fretting hand appears to belong to a single individual across the majority of images, indicating limited subject diversity. The dataset also exhibits non-uniform lighting conditions, as some samples appear brightly illuminated while others are captured under low-light or shadowed environments. While there is some variation in hand pose and fretboard positioning, most images appear to be recorded from similar camera angles and distances, suggesting constrained capture settings. These characteristics imply that the dataset may not fully represent the diversity found in real-world guitar-playing scenarios. Prior to model training, the dataset will be rebalanced and re-split, and stratified k-fold cross-validation will be employed to ensure fair class representation and more reliable performance estimation despite the dataset's inherent limitations.

2) *Exploratory Data Analysis Framework:* A systematic exploratory data analysis (EDA) framework was implemented to assess dataset quality and identify potential preprocessing requirements. The analysis encompassed two primary dimensions:

- Image Property Analysis: Comprehensive examination of image characteristics including dimensional analysis (width, height, aspect ratios), file size distributions, and format consistency. Statistical summaries were computed to identify outliers and assess uniformity across the dataset.
- Class Distribution Analysis: Quantitative assessment of chord class representation across dataset splits. This analysis included calculation of class frequencies, identification of class imbalances, and evaluation of distribution consistency across training, testing, and validation partitions.

B. Hand Landmark Extraction Pipeline

1) *MediaPipe Integration:* The landmark extraction process utilized Google's MediaPipe Hands framework, which employs a two-stage approach combining palm detection and hand landmark regression. MediaPipe was configured with the following parameters:

- Static image mode: True (optimized for individual image processing)
- Maximum hands detected: 1 (single-hand chord recognition)
- Detection confidence threshold: 0.7
- Tracking confidence threshold: 0.7

2) *Feature Extraction Process:* Each processed image yielded 21 hand landmarks, with each landmark represented by three-dimensional coordinates (x, y, z) normalized to the image dimensions.

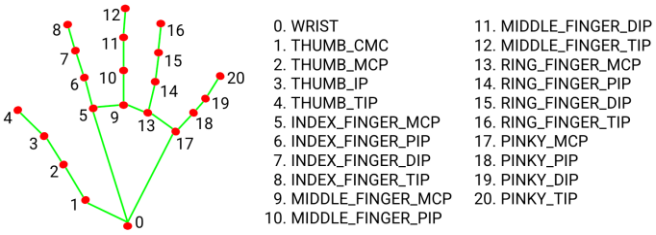


Figure 2. MediaPipe hand landmark model consisting of 21 annotated key points, each representing anatomical joints from the wrist to fingertips. These landmarks are used as structured input features for chord recognition.

This resulted in a 63-dimensional feature vector per sample. The extraction process included:

- Image Preprocessing: Conversion from BGR to RGB color space as required by MediaPipe
- Hand Detection: Application of MediaPipe's palm detection model
- Landmark Regression: Extraction of 21 anatomical hand landmarks
- Confidence Validation: Retention of samples exceeding the specified confidence threshold
- Feature Vector Formation: Flattening of landmark coordinates into a 63-dimensional array

MediaPipe Hands, while robust, remains sensitive to variations in image quality. Differences in brightness, contrast, sharpness, and exposure influence the palm detection and landmark regression stages, often leading to noisy, shifted, or missing landmarks. Low-light or blurred

images can cause the model to misidentify finger boundaries, while overexposed regions may reduce the precision of joint localization. Although landmark coordinates are normalized, extreme variation in hand scale or camera distance can still degrade depth estimation and reduce landmark stability. Because the dataset includes images with inconsistent lighting and visual quality, these conditions directly affect the reliability of the resulting 63-dimensional feature vectors. To account for this, the study evaluates the model under additional noise, missing-landmark, and scale-variation scenarios to assess robustness against real-world input imperfections.

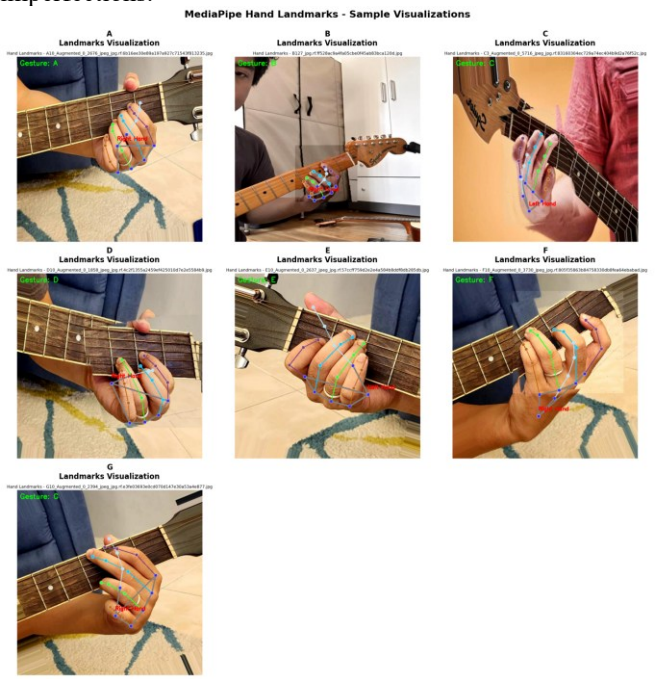


Figure 3. Example visualization of extracted hand landmarks on guitar chord gestures. Each landmark corresponds to a joint or fingertip, providing structured spatial representation for feature extraction and subsequent classification.

3) *Data Augmentation Strategy:* To address class imbalances identified during EDA, a sophisticated augmentation strategy was implemented specifically designed for hand landmark data. The augmentation process incorporated four transformation types:

- Additive Noise: Gaussian noise with standard deviation proportional to image dimensions (±3 pixels normalized) applied independently to x, y, and z coordinates.
- Geometric Scaling: Uniform scaling factors ranging from 0.9 to 1.1, applied relative to the hand's geometric center to preserve anatomical proportions.
- Rotational Transformation: Random rotations within ±15 degrees around the hand center, maintaining relative finger positions while introducing natural hand pose variations.
- Spatial Translation: Random displacement within ±10 pixels (normalized) to simulate variations in hand positioning within the frame.

All augmented landmarks were constrained to remain within the normalized coordinate space [0,1] to maintain physical validity[12], [13].

4) *Class Balancing Protocol:* The class balancing protocol employed oversampling of minority classes through augmentation. The target sample count was set to match the most represented class, with synthetic samples generated using the augmentation transformations described above. This approach ensured equal representation across all seven chord classes while maintaining the original data distribution as the foundation.

C. Neural Network Architecture Design

1) *Architecture Selection Rationale:* Three distinct neural network architectures were selected to evaluate different approaches to temporal landmark sequence classification:

- One-Dimensional Convolutional Neural Network (CNN-1D): Designed to capture local patterns in landmark sequences through convolutional operations along the feature dimension, enabling detection of finger position relationships[14], [15], [16].

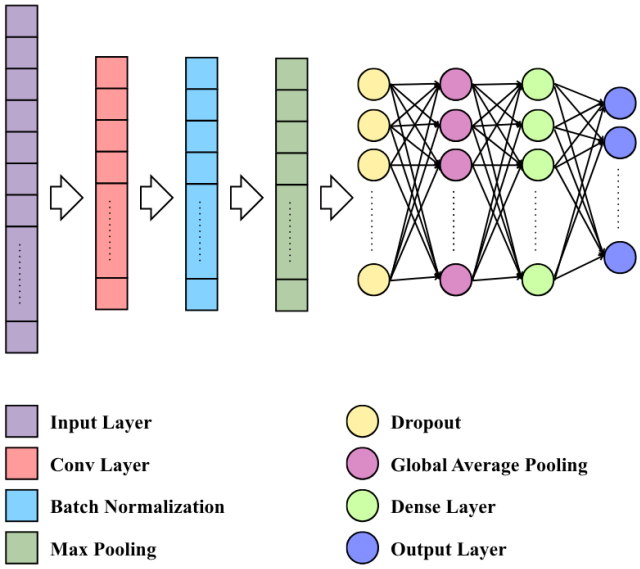


Figure 4. Linear flow showing the basic convolutional approach with progressive filtering and pooling.

The CNN-1D architecture was designed to capture local spatial dependencies in landmark vectors[14][15][16]. A one-dimensional convolution operation computes feature maps as:

$$y_i = \sum_{j=1}^{k} w_j \cdot x_{i+j} + b$$

where x is the input sequence, w represents kernel weights of size k, and b is a bias term, which then allows the detection of finger-position relationships across the 63-dimensional landmark vector as a temporal sequence, applying convolutions along the landmark sequence dimension[14], [15], [16]. Max pooling operations reduced dimensionality while preserving salient features.

- One-Dimensional Residual Network (ResNet-1D): Incorporated skip connections to enable deeper network training while mitigating gradient vanishing problems, allowing for hierarchical feature learning from hand structures[17].

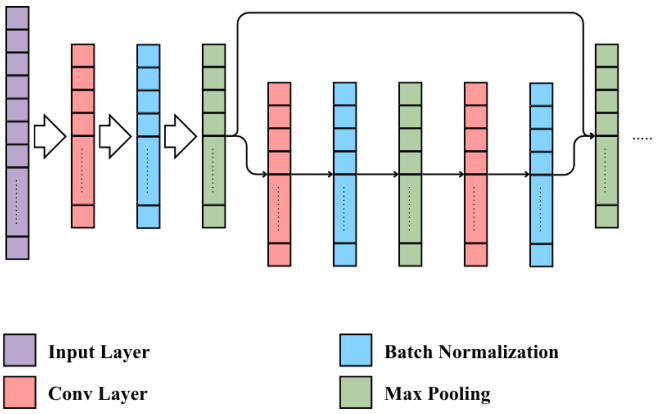


Figure 5. ResNet-1D architecture with residual block structure and skip connections

Each residual block was composed of two convolutional layers with batch normalization, and the output was defined as:

$$y = F(x, \{W_i\}) + x,$$

where $F(x, \{W_i\})$ denotes the residual mapping parameterized by weights W_i, and the skip connection directly passes the input x. When input and output dimensions differed, a 1×1 convolution was used for alignment[17]. This design facilitated hierarchical feature extraction across multiple network depths[17].

- One-Dimensional Inception Network (Inception-1D): The Inception-1D architecture was implemented to capture multi-scale feature extraction through parallel convolutional operations with different kernel sizes, enabling simultaneous analysis of fine details and broader gestural patterns[18], [19], [20]. Each Inception block contained four parallel paths: 1×1 convolution, 1×1 followed by 3×3 convolution, 1×1 followed by 5×5 convolution, and max pooling followed by 1×1 convolution[18], [19], [20]. Outputs were concatenated to form multi-scale representations[18], [19], [20]. Outputs were concatenated to form a rich multi-scale representation:

$$y = Concat(f_{1\times1}(x), f_{3\times3}(x), f_{5\times5}(x), f_{pool}(x))$$

This design allowed simultaneous modeling of fine-grained and broad landmark patterns[18], [19], [20].

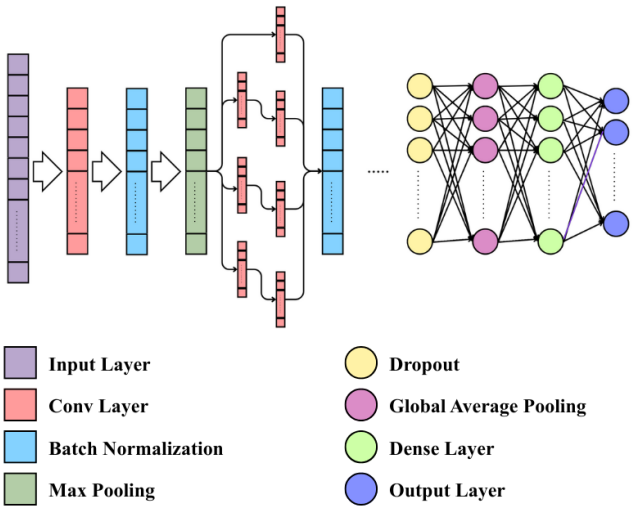


Figure 6. Inception-1D with multi-branch inception block and concatenating four parallel paths.

2) *Hyperparameter Optimization:* Hyperparameter tuning was performed using Bayesian optimization implemented through the Optuna framework. The search procedure utilized the Tree-structured Parzen Estimator (TPE) sampler, with 50 optimization trials conducted for each architecture[21], [22], [23]. TPE models the distribution of hyperparameters by separating previous trials into two groups—those yielding higher objective values and those with lower ones—and then samples new candidate values from the estimated probability distributions[21], [22], [23]. This allows the search to efficiently focus on promising regions of the hyperparameter space rather than exploring randomly[21], [22], [23].

Key hyperparameters included:

- Network depth (number of layers)
- Hidden unit dimensions
- Dropout rates
- Learning rates
- Activation function
- Kernel sizes (for convolutional architectures)
- Filter numbers and progression patterns

Each hyperparameter trial was evaluated using Stratified 3-Fold Cross Validation on the combined training–validation dataset. Stratification ensured that all chord classes were proportionally represented in every split, following best practices for imbalanced classification. [24] For each trial, the model was trained on two folds and validated on the

remaining fold, and this process was repeated three times. The average validation accuracy across the three folds served as the optimization objective. Using cross-validation during tuning provided a more stable and reliable estimate of model performance compared to a single validation split. [25][26]

3) *Training Protocol:* Models were trained using the Adam optimizer with early stopping based on validation accuracy[27], [28]. Training employed a maximum of 100 epochs with early termination after 15 epochs without improvement[28]. Learning rate reduction was applied when validation loss plateaued, with a reduction factor of 0.5 and patience of 8 epochs[29].

4) *Stratified K-Fold Cross Validation:* Model training was conducted using a Stratified 5-Fold Cross Validation protocol to ensure a robust and unbiased evaluation. The combined training–validation dataset was divided into five equally sized folds while maintaining the proportional distribution of all chord classes. In each iteration, four folds were used for training and the remaining fold served as the validation set, allowing every sample to be used once for validation and four times for training[30], [31].

Each model was trained using Adam optimizer accompanied by Early Stopping and ReduceLROnPlateau from scratch in every fold using the hyperparameters obtained from the optimization stage, and performance was assessed on both the validation fold and the held-out test set, which remained entirely unseen during hyperparameter tuning and model development. This process was repeated across all five folds, and the resulting metrics—accuracy, precision, recall, F1-score, per-class performance, confusion matrices, training time, and parameter counts—were aggregated by computing their mean and standard deviation[32]. This cross-validated training protocol provided a comprehensive and statistically reliable estimation of each model's generalization ability and stability across multiple training conditions[33].

D. Evaluation Methodology

1) *Performance Metrics:* The evaluation of the models was carried out across several dimensions. Classification performance was measured using accuracy, precision, recall, and F1-score, reported in macro averages across all chord classes. Computational efficiency was also assessed by recording training time, per-sample inference time, parameter count, and storage requirements to determine deployment feasibility[34]. In addition, robustness was evaluated through multiple tests:

- Noise robustness was evaluated by introducing additive Gaussian noise to the test dataset. Specifically, zero-mean Gaussian noise with varying standard deviations (σ = 0.01, 0.05, 0.10, 0.20) was applied to each feature dimension[31]. This perturbation simulates sensor-level noise or measurement inaccuracies frequently encountered in vision-based landmark detection systems. The formulation can be expressed as:

$$X' = X + N(0, \sigma^2)$$

where X denotes the original feature vector, and σ represents the noise level. By systematically increasing the noise variance, the evaluation quantifies each model's ability to preserve classification accuracy under degraded input quality[34].

- To simulate landmark occlusion or detection failure, we performed random landmark zeroing at varying proportions. For each trial, a predefined fraction of hand landmarks (10%, 20%, 30%, and 40% of the 21 total landmarks) was selected at random, and the corresponding *x, y, z* coordinates were set to zero[34]. Formally, for a missing ratio *r*,

$$X'_{i,j} = \begin{cases} 0 & \text{if landmark } i \text{ is selected as missing} \\ X_{i,j} & \text{otherwise} \end{cases}$$

where i indexes the landmarks and j denotes the coordinate dimension. This approach reflects scenarios in which hand tracking systems fail to detect certain landmarks due to self-occlusion, poor lighting, or camera angle variation.

- Scale robustness was examined by applying uniform scaling transformations to all landmark coordinates. The test dataset was rescaled using multiplicative factors of 0.8, 0.9, 1.1, and 1.2, simulating variability in hand size or changes in the camera's distance from the hand[31]. The transformation is defined as:

$$X' = \alpha \cdot X$$

where α represents the scale factor. This procedure evaluates each architecture's ability to generalize across different spatial scales of hand representations.

2) *Cross-Architecture Comparison Framework:* A comprehensive comparisonframework was established to evaluate trade-offs between accuracy, computational efficiency, model complexity, and robustness. This framework included:

- Performance-Complexity Analysis: Evaluation of accuracy relative to model parameters and storage requirements.
- Speed-Accuracy Trade-offs: Assessment of inference time relative to classification performance.
- Robustness Profiling: Comparative analysis of degradation patterns under different perturbation types.
- Deployment Suitability: Classification of architectures based on application requirements (real-time processing, mobile deployment, high-accuracy applications).

3) *Statistical Validation:* All experiments employed fixed random seeds (seed = 42) to ensure reproducibility. For

each architecture, performance was evaluated using stratified 5-fold cross-validation, producing five accuracy values per model. These fold-wise results were then used to assess whether the observed performance differences between models were statistically significant[36].

Pairwise comparisons between architectures were performed using two established statistical tests: the paired t-test, which evaluates differences under the assumption of normally distributed performance gaps, and the Wilcoxon signed-rank test, a non-parametric alternative that does not rely on normality assumptions. Both tests were applied to the accuracy values across the same cross-validation folds, enabling fair paired comparisons. Confidence intervals were computed for key metrics, and significance levels were reported to support the robustness of the findings[37], [38].

E. Computational Environment

1) *Computer Specifications:* All experiments were conducted on a workstation equipped with an AMD Ryzen 5 5600 6-Core CPU, 16 GB RAM, and an NVIDIA GeForce RTX 3050 GPU with 6 GB VRAM. The models were implemented in Python 3.8.20 using the TensorFlow 2.10.1 deep learning framework and executed on a Windows 11 operating system. These specifications are reported to ensure reproducibility and to provide context for the reported training and inference times.

III. RESULT AND DISCUSSION

This study evaluated three distinct 1D CNN architectures for guitar chord recognition from static hand landmarks: CNN-1D, ResNet-1D, and Inception-1D networks. The experiments were conducted using a dataset containing 7 guitar chord classes (A, B, C, D, E, F, G) with 2,338 training samples and 259 test samples. Each sample consisted of 63-dimensional feature vectors representing 21 hand landmarks with 3 coordinates each. The evaluation employed standard classification metrics including accuracy, precision, recall, and F1-score, alongside computational efficiency measures such as training time, inference time, and model size.

A. Model Specifications

TABLE I
MODEL SPECIFICATIONS COMPARISON

Model	# of params	Model Size (MB)
CNN-1D	3,517,191	40.331
ResNet-1D	2,208,647	25.571
Inception-1D	2,779,111	32.122

Table I summarizes the structural characteristics of the three evaluated architectures, highlighting the number of trainable parameters and the resulting model sizes. These specifications provide a foundational understanding of each architecture's computational footprint prior to analyzing their performance on guitar chord recognition.

The CNN-1D model contains the largest number of parameters at 3.52 million, resulting in the largest model size (40.33 MB). This reflects the model's relatively deep yet straightforward sequential design, where each convolutional layer contributes linearly to parameter growth. Although this capacity may allow the model to learn complex spatial patterns from hand landmarks, it also implies heavier memory usage and potentially longer training times[39], [40], [41]. In contrast, the ResNet-1D architecture has the smallest parameter count (2.21 million) and the most compact storage footprint (25.57 MB). The reduction in parameters is primarily due to the use of residual connections that enable deeper feature extraction without requiring proportionally larger convolutional blocks. This efficiency indicates that ResNet-1D may achieve competitive representational power while maintaining higher computational efficiency, which is an important consideration for real-time or mobile applications of chord recognition[17], [42], [43]. The Inception-1D model sits between the two extremes, with 2.78 million parameters and a 32.12 MB model size. Its mixed-scale convolutional pathways introduce moderate architectural complexity while avoiding excessively large parameter growth. The model's multi-branch structure is advantageous for capturing features at multiple receptive field sizes, which is relevant given the variability in hand shapes and finger span across different chords[44], [45]. However, this comes at the cost of a slightly larger model than ResNet-1D.

Overall, the comparison highlights a trade-off between architectural complexity, parameter efficiency, and memory usage. ResNet-1D offers the most compact design, CNN-1D provides the highest capacity at the cost of size, and Inception-1D balances multi-scale feature extraction with moderate resource requirements. These differences provide important context for interpreting their training behavior, generalization performance, and suitability for deployment in resource-constrained environments.

B. Model Performance

Prior to model training, a comprehensive hyperparameter optimization procedure was performed for each architecture in order to identify configuration settings that maximized validation performance. The resulting best-fit hyperparameters provide insights into how each architecture responds to variations in model depth, filter size, and regularization strategies when applied to static hand-landmark inputs.

For the baseline CNN-1D model, the optimization process selected a relatively deep configuration with four convolutional layers and a high initial filter count of 128, indicating that the model benefited from increased representational capacity. A kernel size of 5 and pool size of

2 suggest that moderately wide receptive fields were optimal for capturing spatial patterns across the landmark sequence. A dropout rate of 0.2 and a dense layer of 64 units balanced regularization and downstream feature aggregation. The selected learning rate ($\approx 3.24\times10^{-4}$) reflects a relatively conservative update step, which is typical for models with higher parameter counts to ensure stable convergence[46], [47].

The ResNet-1D architecture yielded its best performance with six residual blocks, demonstrating that introducing additional depth enhanced feature extraction while residual connections mitigated degradation problems. The model favored a lower base filter size (64) compared to the CNN baseline, consistent with the efficiency of residual learning in extracting hierarchical features using fewer filters. A kernel size of 7 indicates that the model benefited from broader temporal/spatial coverage per convolution. Similar to the CNN model, a dropout rate of 0.2 and a dense layer with 128 units were optimal for regularization and final classification. The learning rate ($\approx 7.54\times10^{-4}$) was notably higher than that of CNN-1D, suggesting that the residual structure allowed the model to learn effectively with more aggressive gradient updates.

The Inception-1D model achieved its best configuration with five inception blocks and a comparatively large base filter size of 192, highlighting the architecture's reliance on multi-scale feature extraction with substantial channel capacity. Unlike the fixed kernel sizes of the previous models, the Inception design leverages parallel convolutions, which likely benefited from higher filter dimensionality. A dropout rate of 0.2 and a dense layer with 256 units reflect the model's need for additional capacity in its final layers to integrate multi-scale representations. The optimal learning rate ($\approx 2.42\times10^{-3}$) was the highest among the three models, indicating that the parallel-branch structure facilitated stable optimization even with larger update steps.

These differences in optimal hyperparameters across the three architectures highlight their distinct representational characteristics. CNN-1D benefited from depth and moderate learning rates, ResNet-1D leveraged residual connections to maintain performance with fewer filters and higher learning rates, and Inception-1D required substantial channel capacity and tolerated the most aggressive learning schedule. These optimized configurations set the foundation for the subsequent comparative evaluation of model performance.

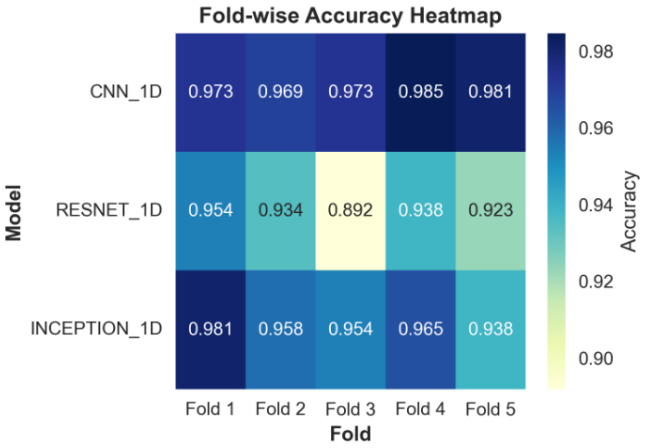


Figure 7. Fold-wise accuracy heatmap showing the performance of CNN-1D, ResNet-1D, and Inception-1D across five cross-validation folds.

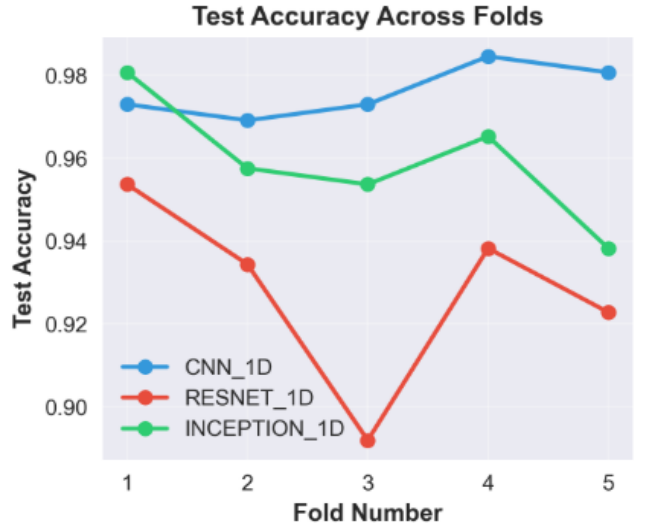


Figure 8. Test accuracy trends across folds for the three 1D architectures, illustrating consistency differences between the models.

The superior performance of CNN-1D over more complex architectures provides empirical evidence that architectural complexity does not necessarily enhance classification performance for temporal hand landmark sequences. This finding aligns with recent research on time-series classification demonstrating that simpler, task-aligned architectures often outperform more sophisticated variants[48], [49], [50].

Figure 7 and 8 presents the aggregated confusion matrices for the three evaluated architectures—1D CNN, ResNet-1D,

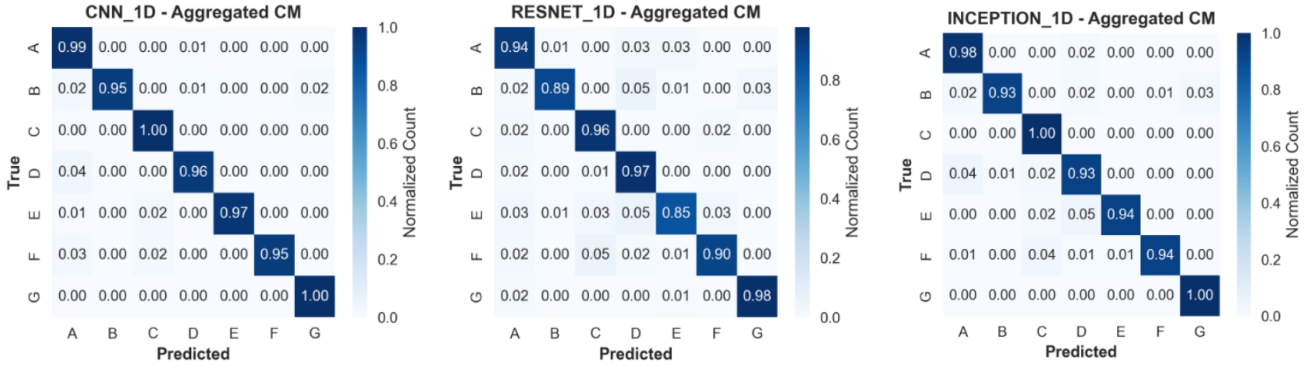


Figure 9. Aggregated confusion matrices for the three evaluated architectures (1D CNN, ResNet 1D, and Inception 1D) across all five cross-validation folds. Each matrix is normalized by true class count to highlight inter-class confusion patterns. The CNN-1D model demonstrates the highest concentration along the diagonal, indicating superior chord discriminability from static hand-pose landmarks. ResNet-1D exhibits more distributed misclassification, particularly among structurally similar chords, while Inception-1D provides competitive performance with improved stability across classes.

and Inception-1D—across all five folds. Overall, the results demonstrate that all models are capable of learning discriminative representations from static hand-pose landmarks, consistent with findings in prior gesture-recognition literature where hand-pose features offer strong separability across classes [51]. However, notable differences emerge in the degree of inter-class confusion and the consistency of predictions.

The CNN-1D architecture exhibits the highest diagonal concentration, with most chord classes achieving normalized accuracies of 0.95–1.00, indicating robust class separability. Minimal misclassification occurs primarily between neighboring or visually similar hand configurations—for example, occasional confusion between chords D and F or E and F, which share overlapping finger-position patterns. This aligns with prior observations that simple CNNs often perform exceptionally well when the feature space is structured and low-dimensional, such as landmark-based inputs [39]. The low off-diagonal values suggest strong generalization and stable feature extraction without overfitting.

The ResNet-1D model presents a more distributed error pattern, with noticeable drops in accuracy for certain classes (e.g., chord E: 0.85, F: 0.90) and higher off-diagonal activation. Although residual connections are known to enhance deeper models' ability to capture complex temporal patterns [17], the increased model capacity may lead to overfitting when input representations are already highly discriminative. The broader confusion, particularly among E–F–G, suggests that ResNet-1D failed to consistently leverage the subtle differences in landmark structure and may have learned redundant or noisy features.

In contrast, Inception-1D achieves performance close to CNN-1D, with diagonal values often exceeding 0.94, and minimal cross-class confusion. Its multi-scale convolutional filters likely enhance sensitivity to subtle geometric variations in finger positioning—consistent with prior work showing that Inception-style architectures excel in tasks where both global and local patterns matter [45], [52]. Notably, the Inception-1D model reduces the misclassification seen in ResNet-1D and maintains stability across classes with highly similar visual patterns, such as E, F, and G.

Across the three architectures, the patterns in misclassification provide meaningful insight into chord-specific difficulty. Chords with greater intra-class landmark variability (e.g., D, E) show slightly lower precision across all models, matching previous findings that hand-pose representations can exhibit natural variation even for the same gesture or chord [51], [7]. Meanwhile, highly rigid or distinctive chords (e.g., A, C, G) consistently yield near-perfect classification.

Overall, these confusion matrices highlight that while advanced architectures like ResNet-1D and Inception-1D introduce capabilities for hierarchical or multi-scale feature extraction, the simpler CNN-1D architecture remains the most stable and accurate for this particular task. This reinforces existing evidence that model complexity does not guarantee superior performance, particularly when the input representation is low-dimensional and structurally consistent [53].

He et al. (2015) introduced residual learning to address the vanishing gradient problem in extremely deep networks, enabling training of networks with hundreds of layers[17]. However, the ResNet-1D architecture employed relatively shallow depth, suggesting skip connections confer limited benefits when the vanishing gradient problem is less pronounced. The 4.79 percentage point accuracy deficit of ResNet-1D (92.82%) relative to CNN-1D (97.61%) indicates that the architectural complexity of skip connections imposes computational overhead without corresponding benefits for this low-dimensional temporal classification task.

Szegedy et al. (2014) proposed the Inception architecture for efficient multi-scale feature extraction through parallel convolutional pathways[44]. While Inception-1D demonstrated competitive performance (95.91% accuracy), it underperformed CNN-1D by 1.70 percentage points. The multi-scale feature benefits of Inception modules appear less pronounced for one-dimensional temporal sequences compared to spatial features in image classification. The higher variance in Inception-1D performance (1.39%

standard deviation versus 0.57% for CNN-1D) further suggests less stable adaptation across different data distributions.

C. Statistical Significance Analysis

To determine whether the performance differences between the three architectures were statistically meaningful, pairwise comparisons were conducted using both the paired t-test and the Wilcoxon signed-rank test on the 5-fold accuracy values for each model. These tests evaluate whether two models differ consistently across the same cross-validation splits, with the paired t-test assuming normally distributed differences and the Wilcoxon test providing a non-parametric alternative.

TABLE II
PER-FOLD MODEL ACCURACY

Fold	CNN-1D	ResNet-1D	Inception-1D
1	0.97297	0.95367	0.98069
2	0.96911	0.93436	0.95753
3	0.97297	0.89189	0.95367
4	0.98456	0.93822	0.96525
5	0.98069	0.92278	0.93822
Mean	0.97606	0.92819	0.95907
Std	0.00634	0.02310	0.01559

Both statistical tests indicate that CNN-1D significantly outperforms ResNet-1D.The paired t-test yielded $t = 4.57$, $p = 0.0103$, and the Wilcoxon test produced $W = 0.0$, $p = 0.0625$. While the Wilcoxon p-value is slightly above the conventional 0.05 threshold, the strong t-test significance suggests a consistent advantage for CNN-1D across folds. These results collectively indicate that CNN-1D provides statistically higher accuracy than ResNet-1D.

For CNN-1D compared to Inception-1D, no statistically significant difference was observed. The paired t-test produced $t = 2.11$, $p = 0.1028$, and the Wilcoxon test yielded $W = 1.0$, $p = 0.1250$. Both p-values are well above 0.05, indicating that the performance gap between the two models is not statistically reliable. Although CNN-1D shows slightly higher mean accuracy, the variation across folds prevents strong conclusions about superiority.

Inception-1D outperformed ResNet-1D with statistical significance. The paired t-test resulted in $t = -3.86$, $p = 0.0182$, and the Wilcoxon test yielded $W = 0.0$, $p = 0.0625$. Similar to the CNN-1D vs. ResNet-1D comparison, the t-test indicates a significant difference, while the Wilcoxon test approaches but does not cross the 0.05 threshold. Taken together, these tests support the conclusion that Inception-1D achieves higher accuracy than ResNet-1D.

These findings confirm that while both CNN-1D and Inception-1D outperform ResNet-1D, the difference between CNN-1D and Inception-1D is not strong enough to be considered statistically significant within the 5-fold evaluation framework. This suggests that the two leading models perform comparably in terms of classification accuracy.

D. Model Robustness Evaluation

To further assess the reliability of the three architectures, this study evaluates their performance under controlled perturbations frequently encountered in real-world hand-pose data acquisition. These perturbations include Gaussian noise, missing landmarks, and geometric scaling variations, reflecting common sources of error in landmark-based gesture recognition systems [54], [55], [56], [57], [58]. The robustness analysis reveals distinct behavioral patterns across architectures, offering deeper insight into the generalizability and stability of learned representations.

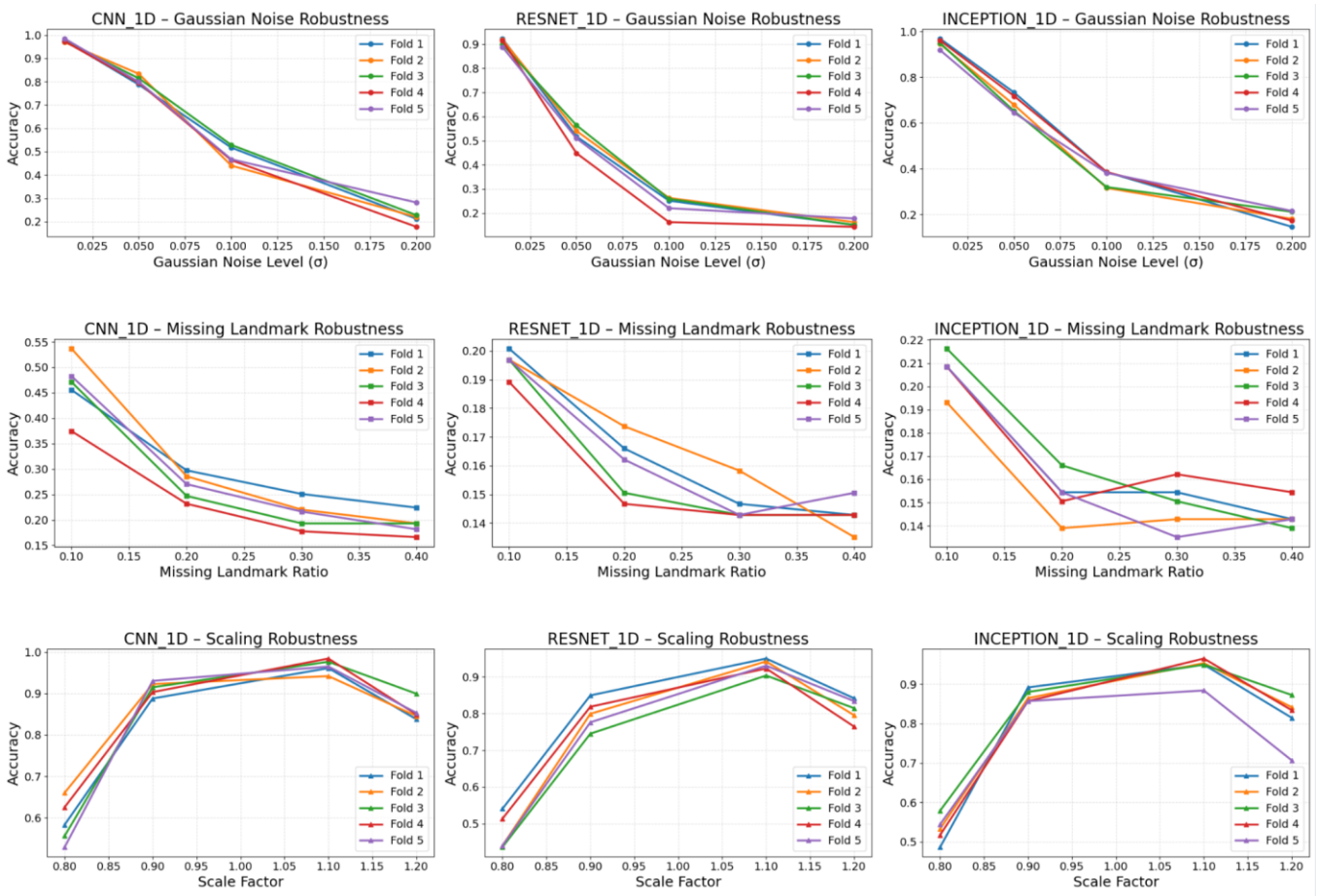
Figure 10 illustrates the accuracy decline as increasing levels of Gaussian noise (σ) are applied to the input landmarks. Across all models, performance deteriorates monotonically with higher noise amplitudes, consistent with prior findings that hand-pose features are highly sensitive to positional jitter [59], [60].

The CNN-1D model demonstrates the strongest resilience, maintaining accuracies above 0.80 for mild noise ($\sigma \leq 0.05$) and retaining comparatively higher performance even at $\sigma = 0.10$. This behavior aligns with established observations that shallow CNNs tend to learn more localized, noise-tolerant filters when the feature domain is structurally simple [61].

In contrast, the ResNet-1D architecture shows the steepest degradation, particularly beyond $\sigma = 0.05$, where accuracy drops sharply across all folds. Residual networks are known to be sensitive to noisy signals when trained on small or clean datasets due to their deeper architecture and reliance on skip connections that propagate perturbations [62].

Inception-1D performs better than ResNet-1D but slightly below CNN-1D. Its multi-scale convolutional kernels appear to mitigate noise effects, consistent with prior work showing that Inception modules improve robustness by capturing features at different receptive field scales [63], [64], [65]. However, the model still exhibits substantial degradation once σ exceeds 0.10, indicating limited resistance to high-magnitude perturbations.

Figure 10. Robustness evaluation of CNN-1D, ResNet-1D, and Inception-1D across all five folds under three perturbation conditions: Gaussian noise, missing landmarks, and scaling transformations. The first row shows the accuracy degradation as increasing levels of Gaussian noise are added to the input features. The second row presents model performance as different proportions of hand landmarks are randomly removed. The third row illustrates the impact



of scaling variations on accuracy. Together, these plots highlight each model's stability and sensitivity to input distortions, with notable differences in how the architectures handle noise, partial data loss, and geometric scaling.

Missing landmarks simulate occlusion and sensor dropout—frequent issues in hand-tracking systems due to self-occlusion, low lighting, or tracking failures [66], [67]. As shown in Figure 10, all architectures experience notable performance reductions even at modest missing ratios (10–20%).

The CNN-1D architecture again demonstrates superior robustness, with accuracies remaining around 0.30 at 20% missing landmarks. Although the values are relatively low overall, the model shows consistent decline patterns with limited fold variance. This suggests that CNN-1D learns stable feature dependencies where the absence of partial information does not immediately disrupt classification—a property previously observed in low-dimensional CNN gesture-recognition models [68], [69].

The ResNet-1D model, however, consistently performs the worst, dropping to ≈0.14–0.16 by 20% missing ratio. This outcome mirrors prior evidence that deeper residual models strongly depend on complete spatial information and are disproportionately affected when critical structural cues are removed [17].

The Inception-1D architecture performs slightly better than ResNet-1D, showing marginally higher accuracy and better fold consistency. Still, it exhibits instability at higher missing ratios, likely because multi-branch convolution paths struggle when a significant portion of the structured landmark space is absent. Prior studies have similarly reported sensitivity of Inception-like architectures to structured dropout in low-channel inputs [44], [63].

Furthermore, a controlled scaling experiment was conducted in which the hand-landmark inputs were uniformly scaled across five scale factors: 0.80, 0.90, 1.00, 1.10, and 1.20. The evaluation was performed independently for all five folds, enabling a consistent fold-wise comparison of robustness trends across models. The resulting accuracy curves for each architecture are shown in Figure 10.

Across all models, a consistent pattern emerged: performance was lowest at extreme down-scaling (0.80), improved substantially near 0.90, peaked around 1.10, and declined again slightly at the largest scale (1.20). This behavior reflects the sensitivity of landmark-based chord classification to distortions in inter-joint distances, especially

when scaling reduces the relative spatial resolution of finger positions.

The CNN-1D model demonstrated a pronounced sensitivity at the 0.80 scale, with accuracy dropping across all folds. However, performance quickly stabilized at 0.90 and peaked near 1.10, where all folds achieved accuracies above 0.95. At 1.20, the model exhibited moderate degradation, though most folds remained above 0.90. Overall, CNN-1D showed strong robustness for moderate scaling but reduced stability under aggressive down-scaling.

ResNet-1D displayed similar scaling behavior but with consistently lower performance at the 0.80 level compared to CNN-1D. Notably, its improvement from 0.80 → 0.90 was steeper, indicating higher sensitivity to lost spatial detail. The model reached its peak around 1.10 but showed a sharper decline at 1.20 relative to CNN-1D. This suggests that while ResNet-1D effectively captures scale-invariant features at moderate ranges, it is less stable under extreme scaling perturbations.

Inception-1D demonstrated the most variability across folds, particularly at higher scales. The model followed the general trend of accuracy improvement from 0.80 to 1.10, achieving competitive performance near the peak. However, at scale 1.20, its robustness dropped more severely—especially in Fold 5, which fell below 0.75. This fold-specific instability indicates that Inception-style multi-receptive-field processing may be more sensitive to geometric distortions when the input departs significantly from the natural scale observed during training.

Overall, all three architectures performed reliably within the 0.90–1.10 range, indicating that moderate deviations in hand-landmark scale do not significantly affect recognition accuracy. However, extreme down-scaling (0.80) and over-scaling (1.20) consistently reduced performance across models. Among the three, CNN-1D exhibited the most stable performance across folds, whereas Inception-1D showed the highest fold-to-fold variability, particularly at larger scales. These findings highlight the importance of scale normalization in landmark-based guitar chord recognition systems and suggest that incorporating scale-augmentation during training may further enhance robustness.

E. Model Computational Efficiency

The three architectures exhibited distinct efficiency profiles relevant for real-world deployment scenarios.

1D CNN required the shortest training time averaging 45.19 ± 7.19 seconds per fold, representing approximately 54% computational savings compared to ResNet 1D (96.42 ± 21.47 seconds) and 35% savings relative to Inception 1D (69.87 ± 5.38 seconds). The low variance in 1D CNN training times (coefficient of variation = 15.9%) indicates stable convergence behavior. ResNet 1D exhibited the highest variance (22.3%), suggesting less predictable optimization dynamics.

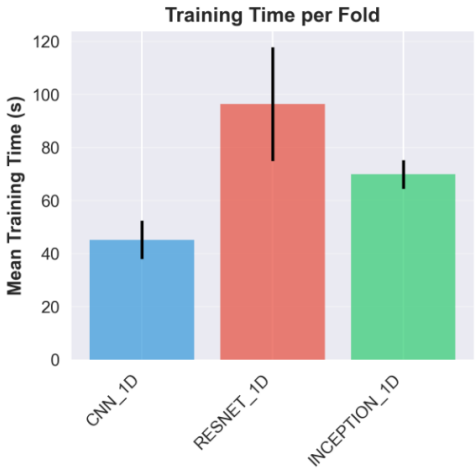


Figure 11. Mean training time per fold for the three 1D architectures, showing that CNN-1D trains the fastest while ResNet-1D requires the longest training time.

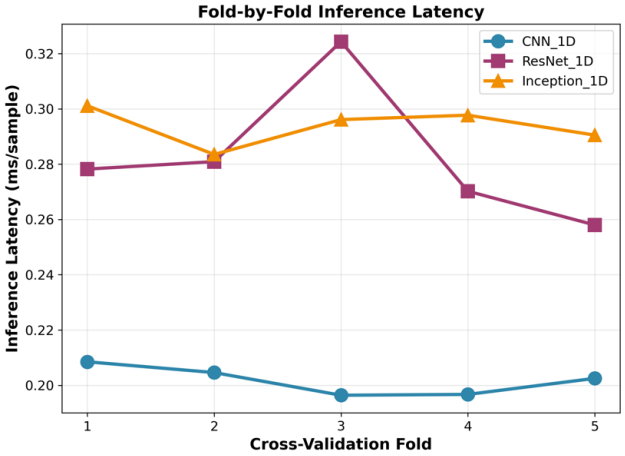


Figure 12. Fold-by-fold inference latency comparison, indicating that CNN-1D consistently achieves the lowest latency, with ResNet-1D and Inception-1D exhibiting higher and more variable inference times.

From a deployment efficiency perspective, ResNet 1D achieved the smallest model footprint at 25.57 MB (36.6% smaller than 1D CNN), yet failed to translate this parameter efficiency into performance advantages. Inference latency measurements revealed 1D CNN's superior speed at 0.202 ± 0.005 ms per sample, approximately 29% faster than ResNet 1D (0.282 ± 0.022 ms) and 31% faster than Inception 1D (0.294 ± 0.006 ms). The deterministic inference latency of 1D CNN (coefficient of variation = 2.27%) contrasts with ResNet 1D's variable performance (7.95%), indicating more consistent real-time behavior essential for interactive systems.

IV. CONCLUSION

This study presented a comprehensive comparative analysis of three one-dimensional convolutional neural network architectures—CNN-1D, ResNet-1D, and Inception-1D—for the task of guitar chord recognition using static hand-landmark coordinates. Across hyperparameter optimization,

classification performance, statistical testing, robustness evaluation, and computational efficiency metrics, several key findings emerged that clarify the suitability of each architecture for this low-dimensional, structured classification problem.

First, the results demonstrate that CNN-1D consistently delivers the highest overall performance, achieving a mean accuracy of 97.61%, outperforming both ResNet-1D and Inception-1D. Despite having the largest parameter count among the models, CNN-1D exhibited strong generalization, minimal inter-class confusion, and the most stable predictions across folds. Statistical analyses further confirmed that CNN-1D significantly outperforms ResNet-1D, while its advantage over Inception-1D is not statistically significant, indicating that the two models perform comparably in accuracy but differ substantially in stability and computational characteristics. ResNet-1D, although parameter-efficient and compact in memory footprint, consistently produced the weakest performance across all evaluation dimensions. The architecture struggled with subtle inter-class variations in landmark geometry and showed limited robustness to noise, occlusion, and geometric scaling. These findings suggest that the advantages of residual learning—typically observed in deep or high-dimensional feature spaces—do not translate effectively to static landmark-based chord recognition. Inception-1D offered competitive performance and moderate robustness, benefiting from multi-scale feature extraction. However, it displayed higher fold-to-fold variability and pronounced instability under certain perturbations, particularly at larger scaling factors and higher missing-landmark ratios. This variability indicates that the model's multi-branch structure may be overly complex for the constrained geometry of hand-pose landmarks.

Robustness experiments reinforced these conclusions: CNN-1D consistently demonstrated superior resilience across Gaussian noise, missing data, and scale distortion. While all models performed adequately within moderate scaling ranges (0.90–1.10), CNN-1D maintained the most stable accuracy trends and lowest variance across folds. Such robustness is essential for real-world applications where hand-tracking systems frequently encounter sensor noise, partial occlusion, or geometric inconsistencies.

Finally, computational efficiency analysis highlighted the practicality of CNN-1D for deployment. It exhibited the fastest training times, the lowest inference latency, and the most stable runtime behavior, outperforming both alternative architectures despite its larger parameter count. These attributes make CNN-1D particularly well-suited for interactive or real-time chord recognition systems, including those intended for mobile or resource-constrained environments.

However, while this study provides a detailed comparative analysis of three 1D convolutional architectures for static hand-landmark–based guitar chord recognition, several promising extensions could further advance the field and enhance the practical applicability of the system.

Future research should expand the evaluation beyond deep 1D CNN architectures. Traditional machine learning models—such as Support Vector Machines (SVM), Random Forests, k-Nearest Neighbors (k-NN), and Gradient Boosting—have historically performed strongly on structured, low-dimensional data such as landmark coordinates. Comparing the proposed deep-learning models against these classical approaches would help determine whether the additional complexity of CNN-based architectures provides meaningful advantages or if simpler methods yield comparable performance with greater interpretability and lower computational cost.

Additionally, exploring 2D convolutional or hybrid architectures may offer new performance perspectives. Landmark coordinates can be projected into structured representations such as joint heatmaps, distance matrices, or skeletal adjacency images, enabling the application of 2D CNNs, Vision Transformers (ViT), or attention-based networks. These representations may capture spatial relationships more effectively than sequential 1D encoding. Likewise, Graph Neural Networks (GNNs), which operate directly on skeletal graphs, could offer powerful alternatives for feature extraction from anatomical structures.

Furthermore, although this study focuses on static hand poses, guitar chord transitions inherently involve temporal dynamics. Future research could integrate temporal modeling using architectures such as LSTMs, GRUs, Temporal Convolutional Networks (TCNs), or Transformer-based sequence encoders. Combining static and dynamic data may improve recognition accuracy for ambiguous chords or transitions characterized by subtle movement patterns.

Given the strong performance and computational efficiency of the CNN-1D model, future work should explore practical deployment pathways. Integration into a mobile or tablet-based guitar learning application represents a promising direction, utilizing frameworks such as TensorFlow Lite, Core ML, or ONNX Runtime to achieve real-time inference. Such a system could use smartphone or tablet cameras to perform hand tracking (e.g., via MediaPipe), process landmarks through the trained model, and provide immediate chord recognition or corrective feedback to users.

Beyond mobile deployment, incorporating the system into IoT or embedded devices—such as smart guitar tuners, external camera modules, or wearable sensors—could enable hands-free chord monitoring during practice sessions. Low-latency inference and lightweight model footprints make the proposed architectures suitable for edge computing environments.

Future implementations may also incorporate interactive features such as feedback analytics, chord correctness scoring, error localization (e.g., detecting incorrect finger placements), or adaptive learning pathways. Combining recognition models with visual overlays or augmented reality guidance could substantially enhance user experience and educational value.

Overall, this study concludes that simple, well-designed convolutional architectures are highly effective for static hand-landmark classification, and increased architectural complexity does not necessarily yield performance benefits for this domain. The findings emphasize the importance of model–task alignment: when input representations are low-dimensional, structured, and semantically coherent, straightforward CNN architectures can outperform more advanced deep learning models.

REFERENCES

[1] Wilson, K., & Pfeiffer, P. E. (2023). Feedback in augmented and virtual reality piano tutoring systems: a mini review. Frontiers in Virtual Reality, 4, 1207397.

[2] Chen, R., Shen, W., Srinivasamurthy, A., & Chordia, P. (2012, October). Chord Recognition Using Duration-explicit Hidden Markov Models. In ISMIR (pp. 445-450).

[3] Rao, Z., & Feng, C. (2023). Automatic Identification of Chords in Noisy Music Using Temporal Correlation Support Vector Machine. IAENG International Journal of Computer Science, 50(2).

[4] Birkeland, S., Fjeldvik, L. J., Noori, N., Yeduri, S. R., & Cenkeramaddi, L. R. (2024). Thermal video-based hand gestures recognition using lightweight cnn. Journal of Ambient Intelligence and Humanized Computing, 15(12), 3849-3860.

[5] Mitjans Coma, A. (2020). Visual recognition of guitar chords using neural networks.

[6] Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C. L., & Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214.

[7] Ooaku, T., Linh, T. D., Arai, M., Maekawa, T., & Mizutani, K. (2018, November). Guitar chord recognition based on finger patterns with deep learning. In Proceedings of the 4th International Conference on Communication and Information Processing (pp. 54-57).

[8] *Özbaltan, N. (2024). Real-time chord identification application: Enabling lifelong music education through seamless integration of audio processing and machine learning. Online Journal of Music Sciences, 9(2), 405-414.*

[9] Doosti, B., Naha, S., Mirbagheri, M., & Crandall, D. J. (2020). Hope-net: A graph-based model for hand-object pose estimation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6608-6617).

[10] Biswas, S., Nandy, A., Naskar, A. K., & Saw, R. (2023, November). MediaPipe with LSTM architecture for real-time hand gesture recognization. In International Conference on Computer Vision and Image Processing (pp. 422-431). Cham: Springer Nature Switzerland.

[11] bilkent, "Bilkent Cs464 Dataset," Roboflow, 2024. https://universe.roboflow.com/bilkent/bilkent-cs464 (accessed Sep. 22, 2025).

[12] Kay, C., Mahowald, M., & Hernandez, C. PalmPilot: Drone Control using Live Hand Signal Detection.

[13] Gordienko, Y., Gordienko, N., Taran, V., Rojbi, A., Telenyk, S., & Stirenko, S. (2025). Effect of natural and synthetic noise data augmentation on physical action classification by brain–computer interface and deep learning. Frontiers in Neuroinformatics, 19, 1521805.

[14] Chen, Y., Shi, J., Hu, J., Shen, C., Huang, W., & Zhu, Z. (2025). Simulation data driven time–frequency fusion 1D convolutional neural network with multiscale attention for bearing fault diagnosis. Measurement Science and Technology, 36(3), 035109.

[15] Chen, J., Geng, X., Yao, F., Liao, X., Zhang, Y., & Wang, Y. (2024). Single-cycle pulse signal recognition based on one-dimensional deep convolutional neural network. Electronics, 13(3), 511.

[16] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. Mechanical systems and signal processing, 151, 107398.

[17] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[18] Li, Z., Wang, H., Han, Q., Liu, J., Hou, M., Chen, G., ... & Weng, T. (2022). Convolutional neural network with multiscale fusion and attention mechanism for skin diseases assisted diagnosis. Computational Intelligence and Neuroscience, 2022(1), 8390997.

[19] Yan, T., Chen, G., Zhang, H., Wang, G., Yan, Z., Li, Y., ... & Wang, B. (2024). Convolutional neural network with parallel convolution scale attention module and ResCBAM for breast histology image classification. Heliyon, 10(10).

[20] Al-qaness, M. A., Dahou, A., Trouba, N. T., Abd Elaziz, M., & Helmi, A. M. (2024). TCN-inception: temporal convolutional network and inception modules for sensor-based human activity recognition. Future Generation Computer Systems, 160, 375-388.

[21] Vaiyapuri, T. (2025). An Optuna-Based Metaheuristic Optimization Framework for Biomedical Image Analysis. Engineering, Technology & Applied Science Research, 15(4), 24382-24389.

[22] Tao, S., Peng, P., Li, Y., Sun, H., Li, Q., & Wang, H. (2024). Supervised contrastive representation learning with tree-structured parzen estimator Bayesian optimization for imbalanced tabular data. Expert Systems with Applications, 237, 121294.

[23] Hanifi, S., Cammarono, A., & Zare-Behtash, H. (2024). Advanced hyperparameter optimization of deep learning models for wind power prediction. Renewable Energy, 221, 119700.

[24] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," Proc. 14th Int. Joint Conf. Artificial Intelligence (IJCAI), 1995, pp. 1137–1143, doi:10.1145/3730436.3730498.

[25] D. Krstajic, L. Buturovic, D. E. Leahy, and P. Thomas, "Cross-validation pitfalls when selecting and assessing regression and classification models," J. Cheminformatics, vol. 6, no. 1, p. 10, 2014, doi:10.1186/1758-2946-6-10.

[26] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," Stat. Surveys, vol. 4, pp. 40–79, 2010, doi:10.1214/09-SS054.

[27] Kingma, D. P. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[28] Anam, M. K., Defit, S., Haviluddin, H., Efrizoni, L., & Firdaus, M. B. (2024). Early stopping on CNN-LSTM development to improve classification performance. Journal of Applied Data Sciences, 5(3), 1175-1188.

[29] You, K., Long, M., Wang, J., & Jordan, M. I. (2019). How does learning rate decay help modern neural networks?. arXiv preprint arXiv:1908.01878.

[30] Bengio, Y., & Grandvalet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation. Journal of machine learning research, 5(Sep), 1089-1105.

[31] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," BMC Bioinformatics, vol. 7, no. 1, p. 91, Feb. 2006, doi: 10.1186/1471-2105-7-91.

[32] S. Bates, T. Hastie, and R. Tibshirani, "Cross-validation: what does it estimate and how well does it do it?," J. Am. Stat. Assoc., vol. 119, no. 546, pp. 1434–1445, Oct. 2023, doi: 10.1080/01621459.2023.2197686.

[33] D. Wilimitis, L. Foschini, M. Rajkomar, and A. Beam, "Practical considerations and applied examples of cross-validation in machine learning," JMIR AI, vol. 2, no. 1, p. e49023, Dec. 2023, doi: 10.2196/49023.

[34] Łańcucki, A., Staniszewski, K., Nawrot, P., & Ponti, E. M. (2025). Inference-Time Hyper-Scaling with KV Cache Compression. arXiv preprint arXiv:2506.05345.

[35] Zhang, Y., & Notni, G. (2025). 3D geometric features based real-time American sign language recognition using PointNet and MLP with MediaPipe hand skeleton detection. Measurement: Sensors, 101697.

[36] Muñoz, J., & Young, C. (2018). We ran 9 billion regressions: Eliminating false positives through computational model robustness. Sociological Methodology, 48(1), 1-33.

[37] Bayle, P., Bayle, A., Janson, L., & Mackey, L. (2020). Cross-validation confidence intervals for test error. Advances in Neural Information Processing Systems, 33, 16339-16350.
[38] Bender, R., & Lange, S. (2001). Adjusting for multiple testing—when and how?. Journal of clinical epidemiology, 54(4), 343-349.
[39] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.
[40] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[41] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.
[42] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
[43] Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.
[44] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
[45] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).
[46] You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., ... & Hsieh, C. J. (2019). Large batch optimization for deep learning: Training bert in 76 minutes. arXiv preprint arXiv:1904.00962.
[47] Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv:1803.09820.
[48] Wang, Z., Yan, W., & Oates, T. (2017, May). Time series classification from scratch with deep neural networks: A strong baseline. In 2017 International joint conference on neural networks (IJCNN) (pp. 1578-1585). IEEE.
[49] Bagnall, A., Lines, J., Bostrom, A., Large, J., & Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data mining and knowledge discovery, 31(3), 606-660.
[50] Tavenard, R., Faouzi, J., Vandewiele, G., Divo, F., Androz, G., Holtz, C., ... & Woods, E. (2020). Tslearn, a machine learning toolkit for time series data. Journal of machine learning research, 21(118), 1-6.
[51] Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., & Kautz, J. (2016). Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4207-4215).
[52] Kiranyaz, S., Ince, T., Abdeljaber, O., Avci, O., & Gabbouj, M. (2019, May). 1-D convolutional neural networks for signal processing applications. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 8360-8364). IEEE.
[53] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Adam, H. (2019). Searching for mobilenetv3. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 1314-1324).
[54] Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., & Theobalt, C. (2017). Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In Proceedings of the IEEE international conference on computer vision (pp. 1154-1163).
[55] Núnez, J. (2022). Comparison of Spatio-Temporal Hand Pose Denoising Models (Doctoral dissertation, PhD thesis, Universitat DE Barcelona, 2022. 3).
[56] Guo, W., Qiao, Z., Sun, Y., Xu, Y., & Xiong, H. (2025, October). Revisiting Noise Resilience Strategies in Gesture Recognition: Short-Term Enhancement in sEMG Analysis. In Forty-second International Conference on Machine Learning.
[57] Caughlin, K., Duran-Sierra, E., Cheng, S., Cuenca, R., Ahmed, B., Ji, J., ... & Busso, C. (2022). Aligning small datasets using domain adversarial learning: Applications in automated in vivo oral cancer diagnosis. IEEE journal of biomedical and health informatics, 27(1), 457-468.
[58] Schwabe, D., Becker, K., Seyferth, M., Klaß, A., & Schaeffter, T. (2024). The METRIC-framework for assessing data quality for trustworthy AI in medicine: a systematic review. NPJ digital medicine, 7(1), 203.
[59] Sridhar, S., Mueller, F., Oulasvirta, A., & Theobalt, C. (2015). Fast and robust hand tracking using detection-guided optimization. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3213-3221).
[60] Karimi, D., Dou, H., Warfield, S. K., & Gholipour, A. (2020). Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. Medical image analysis, 65, 101759.
[61] Meir, Y., Tevet, O., Tzach, Y., Hodassman, S., Gross, R. D., & Kanter, I. (2023). Efficient shallow learning as an alternative to deep learning. Scientific Reports, 13(1), 5423.
[62] Roy, P., Ghosh, S., Bhattacharya, S., & Pal, U. (2018). Effects of degradations on deep neural network architectures. arXiv preprint arXiv:1807.10108.
[63] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1).
[64] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). Pmlr.
[65] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).
[66] Simon, T., Joo, H., Matthews, I., & Sheikh, Y. (2017). Hand keypoint detection in single images using multiview bootstrapping. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1145-1153).
[67] K. Tokunaga, R. Ozaki, Y. Kamihoriuchi, T. Kawasetsu and K. Hosoda, "Hand Tracking System Utilizing Learning Based on Vision Sensing and Ionic Gel Sensor Glove," 2025 IEEE/SICE International Symposium on System Integration (SII), Munich, Germany, 2025, pp. 696-701, doi: 10.1109/SII59315.2025.10871040.
[68] Khan, H., Wang, X., & Liu, H. (2022). Handling missing data through deep convolutional neural network. Information Sciences, 595, 278-293.
[69] Kim, S., Kim, H., Yun, E., Lee, H., Lee, J., & Lee, J. (2023, July). Probabilistic imputation for time-series classification with missing data. In International Conference on Machine Learning (pp. 16654-16667). PMLR.