International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I10P126

Volume 74 Issue 10, 362-377, October 2026
© 2026 Seventh Sense Research Group®

Original Article

Distribution-Aware Dynamic Temperature Knowledge Distillation for Efficient Indian Sign Language Recognition on Edge Devices

Seema Bhalgaonkar[1], Mousami Munot[2], Alwin Anuse[3]

[1]*Department of Electronics & Telecommunication Engineering, PES's Modern College of Engineering, Savitribai Phule Pune University, Pune, Research Scholar at SCTR's Pune Institute of Computer Technology, Savitribai Phule Pune University, Pune, India.*
[2]*Department of Electronics and Telecommunication Engineering, SCTR's Pune Institute of Computer Technology, Savitribai Phule Pune University, Pune, India.*
[3]*TV Engineering (recording) at Film and Television Institute of India (FTII), Pune, India.*

[1]*Corresponding Author: seema.bhalgaonkar@moderncoe.edu.in*

Received: 25 May 2026 Revised: 10 September 2026 Accepted: 18 September 2026 Published: 27 October 2026

Abstract *- Sign language recognition serves as a critical communication aid for the deaf-mute community, enabling seamless interaction with the hearing population. While deep learning models have demonstrated remarkable recognition accuracy, their large model size makes deployment on resource-constrained edge devices challenging. Knowledge distillation is highly effective in compressing heavy models; however, the classic knowledge distillation approach relies on a static temperature setting, which limits the student model's ability to acquire rich inter-class knowledge from the teacher. This study validates the effectiveness of the Distribution-Aware Dynamic Temperature Knowledge Distillation framework for Indian Sign Language gesture recognition. MobileNetV2 is employed as the teacher model, and two student configurations - MobileNetV2 α=0.5 and α=0.35, representing moderate and aggressive compression levels respectively, are evaluated across multiple performance parameters. Experimental results demonstrate a consistent accuracy improvement of approximately 6% (α=0.5) and 5% (α=0.35) over classic knowledge distillation across both student configurations, with model size reduced from 33.93MB to 5.76MB and 4.64MB, respectively. The quantized student models achieve a maximum compression ratio of 12.85×, retaining 95.92% recognition accuracy with an inference time of 3.63ms, enabling real-time gesture recognition on Raspberry Pi. These results demonstrate the effectiveness of DAD-DTKD for sign language recognition and its potential for compressed model deployment on edge devices.*

Keywords *- Edge AI, Dynamic temperature setting, Indian Sign Language, Model compression, Knowledge distillation.*

1. Introduction

The World Health Organization (WHO) estimates that approximately 20% of the global population, representing more than 1.5 billion individuals, suffers from some degree of hearing loss, and 430 million among them have disabling hearing loss [1]. The World Federation of the Deaf (WFD) estimates that more than 70 million deaf individuals belong to signing Deaf communities. In India, the 2011 Census reported approximately 5 million individuals had hearing impairment. Sign language is the principal mode of communication for the majority of deaf-mute individuals [2].

The traditional approach involves employing a sign language interpreter, which is neither efficient nor cost-effective due to the scarcity of interpreters and their unavailability round-the-clock [2]. Additionally, limited availability in rural areas and privacy and confidentiality concerns in legal and health-related matters pose a major hindrance. Hence, there is a need for a real-time, automated sign language translation system that can assist deaf individuals and promote social and digital inclusion of the community in the modern era.

Various approaches have been proposed over the years to develop an automated system for the recognition of sign language. Earlier approaches relied on sensor-based hardware solutions [3-5], followed by Machine Learning (ML) - based approaches that worked on handcrafted features [6, 7]. Sensor-based approaches often require a controlled environment and expensive hardware, which limits their practical deployment. Handcrafted-feature-based ML approaches depend on manually designed features and exhibit limited robustness and generalization under variations in user gestures and environmental conditions.

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

The computer vision era then enabled camera-based gesture recognition with the help of Visual Pattern Recognition Networks (VPRNs). VPRNs are Deep Neural Network (DNN)-based models designed to automatically learn and recognize visual patterns from image data and thus reduce the dependence on handcrafted features.

Their application to Sign Language Recognition (SLR) has resulted in significant improvements in recognition performance and hence has been widely adopted by researchers [8-10]. Various researchers have successfully demonstrated the effectiveness of VPRNs for SLR [11, 14]. VPRN architectures based on VGG [8], and ResNet [9], and MobileNet [15, 16] have demonstrated high SLR performance with reported accuracies exceeding 90% on image-based classification tasks [17, 18].

SLR systems are typically intended for deployment at the user end, where available edge devices often face constraints on available memory, computational power, and power resources. However, VPRNs used for SLR are inherently large in size and require considerable computational resources, memory and power for their execution. These limitations make it difficult to deploy VPRNs on edge devices having limited resources and necessitate compression of these models to reduce their size and enable efficient edge deployment [19-21].

Several model compression techniques, such as Pruning [22, 23], Quantization [24], Low Rank Approximation (LRA) [25], and Knowledge Distillation (KD) [21], have been widely reported in the literature for lowering DNN model size and computational demands [26-30]. Among these approaches, KD has emerged as an effective model compression technique. It is based on a teacher–student framework, in which knowledge from a larger teacher model is transferred to a smaller student model [26]. This enables the student model to learn from the teacher while maintaining a lower model complexity.

However, the classical teacher-student KD framework applies a single, fixed temperature to all input samples, although different sign samples may have different levels of recognition difficulty [25]. This fixed, static temperature may not provide the most suitable knowledge transfer for every sample. This motivates the use of a dynamic temperature that can adapt to the characteristics of individual samples [25]. Recently, researchers have started exploring dynamic temperature strategies for KD.

One such approach is Dynamic Temperature Knowledge Distillation (DTKD) [27], which adjusts the temperature based on the highest output score among the possible sign classes predicted by the teacher model. This allows the student model to better learn from teacher outputs that are adjusted according to the characteristics of each sample [27].

However, the maximum output score does not consider the scores of the other sign classes, which may result in incomplete knowledge being transferred to the student. This may limit the student's ability to learn the relationships among classes with similar output scores and consequently affect its recognition performance.

To address this limitation, this research proposes Distribution-Aware Delta-based Dynamic Temperature Setting Knowledge Distillation (DAD-DTKD), which determines the temperature for each sample based on the teacher's entire output distribution.

By considering the complete output distribution, DAD-DTKD enables the student model to receive more comprehensive knowledge from the teacher and aims to improve the recognition performance of the compact student model while maintaining a smaller model size.

The paper is organized into seven sections. Section 2 discusses the related work. Section 3 introduces the DAD-DTKD framework, while Section 4 describes its application to Indian Sign Language Recognition (ISLR). Section 5 presents the experimental setup, and Section 6 discusses the obtained results. Finally, Section 7 concludes the paper.

2. Related Work and Research Gap

This section presents a review of recent reported studies on SLR, ISLR, and model compression methods applied to VPRNs used for ISLR.

2.1. Sign Language Recognition Approaches

SLR is a system used to recognize sign language from hand gestures, hand shapes, and body movements of deaf individuals and convert it into text or speech so that it can be understood by hearing individuals. SLR thus plays a crucial role in reducing this gap and enhancing communication for the deaf-mute community.

The growing interest of researchers in SLR has led to the development of various approaches, which have gradually evolved over the years from traditional sensor-based methods to recent vision-based deep learning systems.

Recent review studies have also summarized this evolution, highlighting the development of different SLR approaches, datasets, and recognition methods [3]. Figure 1 illustrates the trend in SLR publications across different methodological categories from 2015 to 2025.

The results indicate a gradual shift from sensor-based and conventional ML approaches toward DNN-based approaches, with a notable increase in DNN-based publications during this period. (Source: Google Scholar keyword-based search, Aug 2026)

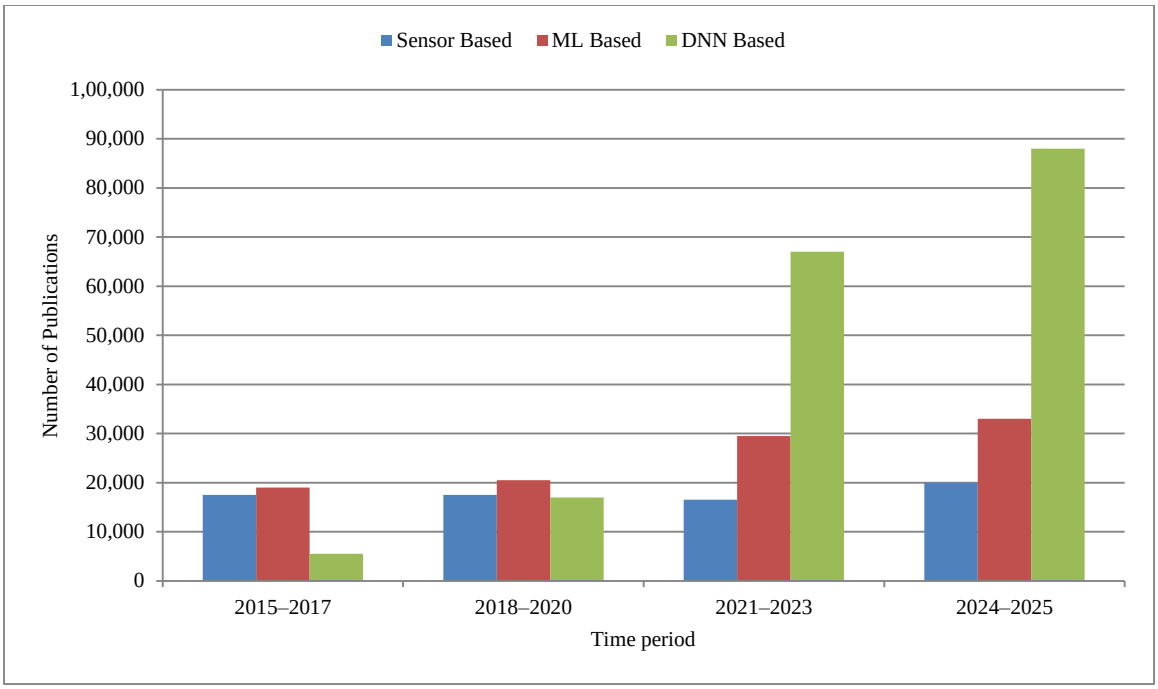

Fig. 1 Approximate publication trend in sign language recognition research by methodology (2020-2025)

Early hardware-based approaches relied primarily on data gloves [4], accelerometers, ElectroMyoGraphy (EMG) sensors [3], and Inertial Measurement Unit (IMU)-based wearable devices [22]. These techniques demonstrated promising recognition accuracy above 90% under controlled laboratory conditions. However, they were impractical due to the high cost of specialised hardware, user discomfort and inconvenience, poor scalability, and strict environmental requirements.

Subsequently, with the emergence of ML, researchers shifted toward vision-based approaches using handcrafted features. Features describing hand shape, texture, and motion were extracted and classified using algorithms such as K-Nearest Neighbours (KNN), Support Vector Machines (SVM) [5], and Random Forest classifiers [23]. These approaches, applied to features such as HOG and SIFT, achieved recognition accuracies in the range of 80% to 96%. Recent work has also combined SIFT with Bag-of-Visual-Words (BoVW) and SVM for sign language alphabet recognition, further demonstrating the use of handcrafted visual features with conventional ML classifiers [7]. However, these methods remained heavily dependent on manual feature engineering, controlled lighting, and hand orientation and hence were limited by their generalization across diverse real-world settings. Recent developments in computer vision and deep learning have facilitated the adoption of camera-based approaches for SLR without the need for wearable hardware [11]. Prominent CNN architectures, including VGG [8], ResNet [9], InceptionV3 [10], and the MobileNet family [15-17], have demonstrated outstanding ISL recognition performance, achieving recognition accuracies between 90% and 99.69% [18]. Recent studies have further explored CNN-based architectures such as EfficientNet for ISLR, demonstrating the continued development of deep learning-based approaches for improving recognition performance [12]. CNN-based approaches have also been extended to address practical challenges in ISLR, such as occlusion, by combining CNN features with gesture information to improve recognition under challenging conditions [18]. This high recognition accuracy is attributed to the automated feature learning capability of CNNs. This reduces the dependence on handcrafted feature extraction. Despite the substantial improvements in recognition accuracy and user convenience, the deployment of these DNN models on end-user devices remains difficult due to their resource requirements.

Many high-performing models require substantial computational resources and memory, and consume considerable power [20]. For instance, VGG-16 contains approximately 138 million parameters, while ResNet-50 contains approximately 25 million parameters [21, 22]. Such large parameter counts demand significant memory and computational power, making direct deployment on resource-constrained devices difficult [20]. For the deaf-mute community specifically, the end-use assistive device must be compact, energy-efficient to support extended battery life, and capable of delivering rapid inference to facilitate real-time gesture-to-text and gesture-to-audio conversion. To address this challenge, model compression techniques have become essential for developing efficient SLR systems suitable for edge deployment. Some recent studies have also explored lightweight architectures such as the MobileNet family [15-17], which are specifically designed for mobile and embedded

applications and are therefore suitable for SLR applications. These architectures were developed to alleviate the computational load of conventional CNNs and have relatively lower parameter counts and model sizes. For ISL recognition, MobileNetV2 combined with transfer learning has also been explored for real-time recognition, demonstrating high recognition accuracy with low computational and memory requirements [14]. However, lightweight CNN architectures may still contain redundant parameters when adapted to a particular task, and their real-world deployment on resource-limited edge devices such as Raspberry Pi remains challenging, as it often results in thermal throttling and high power consumption during inference [19]. Therefore, further compression of these models can help improve their suitability for resource-constrained edge deployment.

2.2. Model Compression for Edge Devices

Various model compression techniques reported in the literature include pruning, quantization, LRA and KD [19, 20]. Pruning reduces model complexity by eliminating redundant and less important parameters from the network [22]. Pruning methods are broadly classified into two categories: structured and unstructured pruning. In unstructured pruning, weights are eliminated individually from different locations in the network, and thus help to achieve high sparsity with minimal impact on accuracy. Structured pruning takes a coarser approach by eliminating complete filters, channels, or layers rather than individual weights, resulting in a systematic reduction in model complexity and more practical hardware acceleration. [22, 23].

A commonly explored unstructured pruning strategy is magnitude-based pruning, in which individual weights are ranked according to their L1 or L2 magnitude, and weights with lower magnitudes are removed [22]. Pruning has been extensively studied across domains, typically achieving parameter reduction of 50–90% with accuracy retention in the range of 88–97% depending on the pruning ratio and fine-tuning strategy applied [22, 23]. However, this technique is highly sensitive to the selection of parameters for removal, and aggressive pruning without careful fine-tuning leads to accuracy degradation. Quantization reduces the memory footprint and computational requirements of a model by approximating the inherent 32-bit Floating-Point (FP32) representation of model parameters using lower bit-width formats such as INT8 or Float16 [24]. INT8 quantization can provide substantial memory reduction and inference speedup on compatible hardware while maintaining comparable recognition accuracy, typically achieving 2–4× memory reduction and up to 3.25× inference speedup with minimal accuracy loss [24]. TensorFlow Lite, an inference framework tailored for edge and mobile deployment, supports quantization in multiple formats - dynamic range, INT8, and Float16 quantization, with Float16 being widely adopted for edge deployment due to its balance between compression and accuracy preservation [24]. TFLite allows Float16 quantization where weights are stored as 16-bit floats, which is the approach adopted in the present study. While quantization provides significant compression benefits, aggressive low-bit quantization leads to accuracy degradation. Hence, careful selection of the quantization method is required for fine-grained recognition tasks such as ISL, where subtle inter-class differences are critical [13].

LRA works by decomposing higher-order weight tensors into lower-order representations, thereby simplifying the computational complexity of matrix operations within deep neural networks [25]. While LRA has shown promising compression results, it involves computationally intensive decomposition procedures and requires careful selection of rank hyperparameters, making it less straightforward to apply compared to other compression methods. KD follows a teacher–student learning framework, where the smaller student model learns from the knowledge of a larger, well-trained teacher model [26]. The student model is trained with both ground truth hard labels and the soft label distributions distilled from the teacher, enabling it to perform comparably to the teacher despite being significantly smaller in size. KD has demonstrated consistent accuracy retention with model size reductions of 4–10× across various computer vision tasks [19], making it an effective compression technique for maintaining high recognition accuracy at reduced model complexity. KD has also been investigated for model compression on resource-constrained edge devices, where a smaller student model learns knowledge from a larger teacher model to reduce model complexity while maintaining performance [21]. Unlike pruning and quantization, KD does not merely simplify an existing model but actively transfers rich inter-class relationships captured in the teacher's soft output distributions to the student, resulting in superior generalization [26]. This characteristic makes KD particularly suitable for SLR applications where preserving subtle discriminative features between visually similar gesture classes is essential. Although KD is a very effective compression technique, the distillation parameters that play a very important role in the distillation process are not yet fully explored by the research community. One such critical distillation parameter -temperature (T) is used to control the smoothening of the teacher's logit outputs for each input sample and is typically set to a fixed value during the entire distillation process. However, in practical scenarios, samples can vary in difficulty, with some being easy to recognize and others being more difficult. The level of smoothness applied to each sample should be varied accordingly. Such a Dynamic Temperature Setting (DTS) enables a student model to better capture the logit distribution among the correct and incorrect classes and thereby helping the student understand the input samples more effectively. Very few research efforts have been initiated in this direction, and several studies [27, 31-33] have demonstrated encouraging results. Wei et al. [27] proposed Dynamic Temperature KD (DTKD), which sets the

temperature dynamically based on the sharpness of the output distribution, and applies it to both teacher and student simultaneously. Another study by Islam et al. [31] has presented a dynamic temperature scheduler that adjusts T based on the cross-entropy loss gap between the teacher and student models. However, the temperature is adjusted at the epoch level and not at the sample level. This limits its adaptability to individual sample difficulty. Adaptive multi-teacher KD presented by [32] uses learnable dynamic temperature, and it works across multiple teachers simultaneously. Liu et al. [33] investigated class-specific temperature settings, applying different T values for target and non-target classes separately.

All the above methods set T based on training dynamics or sharpness and do not consider the overall shape of the teacher's logit distribution, which is important for conveying to the student why other competing classes are not correct. Additionally, limited work has explored the use of DTS for edge deployment, particularly for assistive technology applications such as ISL recognition. Table 1 summarizes the existing reported research in SLR and KD. As evident from the table, the majority of the reported studies address different aspects of SLR, model compression, KD, dynamic temperature setting, and edge deployment. However, the combination of these aspects, particularly the use of distribution-aware DTS for knowledge transfer in compressed SLR models, remains insufficiently explored.

2.3. Research Gaps

Based on the above review, the identified research gap between the existing work and the proposed work can be summarized as follows.

- VPRNs have achieved high recognition accuracy in SLR, but their large size and high computational, memory, and power requirements make deployment on resource-constrained edge devices difficult. Therefore, greater model compression is needed while maintaining recognition accuracy.
- Pruning, quantization, LRA, and KD have been widely used for model compression. However, higher compression can lead to a loss in recognition accuracy. KD is particularly useful because a smaller student model can learn knowledge from a larger teacher model. However, classical KD uses a fixed temperature for all samples, which may not be suitable for samples with different output distributions and level of recognition difficulty.
- Dynamic temperature approaches such as DTKD have been introduced to overcome the limitation of fixed-temperature KD. However, approaches based mainly on the maximum teacher output score do not consider the complete output distribution. Therefore, a more effective temperature-setting approach that considers the characteristics of the teacher's output distribution requires further investigation.
- Although KD and lightweight models have been explored for SLR, limited work has investigated distribution-aware dynamic temperature setting for compressed VPRNs in ISLR. In addition, the combined use of KD and quantization for ISLR on edge devices has received limited attention. Therefore, this work investigates whether distribution-aware dynamic temperature setting can improve knowledge transfer to smaller student models while reducing model size and maintaining recognition accuracy.

Based on the identified research gaps and the objectives of the proposed approach, this study aims to investigate the following Research Questions (RQ)

RQ1: Can DAD-DTKD effectively compress MobileNetV2 for ISL gesture recognition while preserving recognition accuracy close to the teacher baseline model?

RQ2: How do two student configurations, MobileNetV2 α=0.5 and α=0.35, perform under DAD-DTKD guided training in terms of accuracy, model size, and inference efficiency?

RQ3: Is the DAD-DTKD compressed student model compatible with Float16 quantization for real-time deployment on Raspberry Pi edge hardware?

Table 1. Comparative summary of existing sign language recognition and knowledge distillation studies with respect to key research dimensions addressed in the present study

Study	SLR	ISL	KD	Dynamic T	Distribution aware	Edge deploy	Quantization	Assistive output
Katoch et al. 2022 [6]	✓	✓	✗	✗	✗	✗	✗	✗
Wali et al. 2023 [3]	✓	✗	✗	✗	✗	✗	✗	✗
Tao et al. (2024) [7]	✓	✗	✗	✗	✗	✗	✗	✗
Singla et al. 2024 [11]	✓	✓	✗	✗	✗	✗	✗	✗
Vidhyalakshmi et al. 2024 [12]	✓	✓	✗	✗	✗	✗	✗	✗
Das et al. 2024 [13]	✓	✓	✗	✗	✗	✗	✗	✗
Sharma et al. 2023 [14]	✓	✓	✗	✗	✗	✓	✗	✗

Lamaakal et al. TinyMSLR 2026 [34]	✓	✗	✓	✗	✗	✓	✗	✗
Lin et al. 2026 [21]	✗	✗	✓	✗	✗	✓	✗	✗
Lucian et al. 2025 [34]	✗	✗	✓	✗	✗	✗	✗	✗
Wei et al. DTKD 2024 [27]	✗	✗	✓	✓	✗	✗	✗	✗
Shunzhi Yang et al., 2025 [34]	✗	✗	✓	✓	✗	✓	✗	✗
Lin et al. ATMKD 2024 [32]	✗	✗	✓	✓	✗	✗	✗	✗
Liu et al. 2025 [33]	✗	✗	✓	✓	✗	✗	✗	✗
DAD-DTKD (Present work)	✓	✓	✓	✓	✓	✓	✓	✓

3. Methodology

This section presents the application of DAD-DTKD to ISL gesture recognition. The main objective is to develop a lightweight compressed student model that can recognize ISL gestures in real-time while retaining the accuracy at par with the full-scale teacher model.

3.1. DAD-DTKD Brief Overview

The classic KD proposed by Hinton [29] distills the knowledge to the student model using fixed temperature T. The soft label probability for each class is computed as given in Equation (1).

$$P_i = \frac{e^{\frac{z_i}{T}}}{\sum_j e^{\frac{z_j}{T}}} \quad (1)$$

Where,

P_i = probability of class *i*

Z_i = logit corresponding to class *i*

The T value is usually set empirically between the range 1 to 10. This does not reflect the variability in difficulty levels of input samples. Hence, the student model receives limited knowledge about other incorrect classes, which may affect the student model's performance. To address this, it is necessary to accommodate the input image difficulty levels during the distillation process. The DAD-DTKD approach dynamically adjusts the temperature based on the standard deviation of logits, as shown in Equation (2), and measures the spread of logit values across the classes, as shown below.

$$\delta = StdDev(z_t - z_s) \quad (2)$$

Where,

z_t: Teacher logits.

z_s: Student logits.

δ: measures the variability of disagreement across all classes.

A lower standard deviation indicates higher similarity between teacher and student logit distributions. It means the student is performing well, and accordingly, a lower T value can be set. In contrast, a higher standard deviation indicates greater disagreement between the teacher and student logits, indicating that the student requires stronger guidance from the teacher, so T can be set to a higher value accordingly. This T is set dynamically as given by Equation (3).

$$T = T_{min} + (T_{max} - T_{min})\,\sigma(\delta) \quad (3)$$

Where,

T_{min} and T_{max} define the allowable range.

$\sigma(\cdot)$is the sigmoid function.

Temperature adapts according to teacher–student disagreement.

The dynamic temperature $T^*$is then incorporated into the distillation loss as given in Equation (5), replacing the fixed $T$of classic KD.

$$L_{KD} = T^2 kL\left(p_t^{(T)} \parallel p_s^{(T)}\right) \quad (4)$$

$$L_{total} = \alpha k_{CE} + \beta L_{kD} \quad (5)$$

Where:

- L_{CE}: cross-entropy loss with true labels.
- L_{KD}: distillation loss.
- α, β: weighting coefficients.

This distribution-aware dynamic temperature setting results in richer and more precise distillation of knowledge to the smaller student model.

3.2. ISL Dataset

The effectiveness of the DAD-DTKD framework is demonstrated on the publicly available ISLRTC-referred ISL dataset [30]. The dataset consists of 36 classes, 26 classes representing the English alphabet A–Z, and 10 classes for

numeric digits 0–9. A total of 36,000 images are included in the dataset, with 1,000 images for each class. An image has a resolution of 250×250 pixels, captured under different backgrounds and under varying lighting conditions, thus providing diverse training data. This dataset was chosen as it contains complete ISL fingerspelling, including all alphabets and digits, and is acknowledged by ISLRTC - an autonomous centre for ISL established under the Ministry of Social Justice and Empowerment, Government of India. The Government of India and provides sufficient samples per class for reliable KD training. Despite its significance, very limited research has been reported on model compression applied to this dataset. Train/validation/test split.

An 80:20 split was used to separate the dataset into training and validation sets. Training robustness was further improved through data augmentation, including randomly flipping images horizontally and applying small rotational variations of up to ±10°, and random zoom within ±10% are applied during training. The experimentation is conducted using the TensorFlow Keras preprocessing pipeline.

3.3. Model Selection

The study employs a widely used MobileNetV2 architecture for both the teacher and the student model. MobileNet family models are lightweight and specially designed for edge devices. This architecture relies on depth-wise separable convolutions as its core building block. It further incorporates inverted residual blocks and linear bottlenecks, which enable high image classification performance, with significantly fewer parameters compared to standard architectures such as VGG and ResNet [11].

This makes MobileNetV2 a preferred choice for real-time ISL gesture recognition on resource-constrained edge hardware. For experimentation, one teacher model full capacity MobileNet V2 and two smaller student models with distinct compression levels - 42 % and 51 % less parameter size (α=0.5and 0.35) are employed. The teacher and student share the same architectural design, which differs only in channel width and thus enables a clean and fair comparison of model performance and avoids architectural mismatch as knowledge transfer is more direct.

3.4. Teacher - MobileNetV2 (α=1.0)

A pre-trained full-capacity MobileNetV2 with width multiplier α = 1.0 is employed as the teacher model. The full capacity configuration is deliberately selected as it possesses the ability to learn rich and discriminative feature representations from the ISL gesture classes, thereby serving as a reliable source of soft targets during distillation. The teacher model is pre-trained on ImageNet and subsequently fine-tuned on the ISLRTC-referred ISL dataset, achieving a recognition accuracy of 98.69% and thus serves as a strong and well-validated knowledge source. The teacher model weights are frozen during the distillation process, ensuring that only the student model parameters are updated through DAD-DTKD guided training.

3.5. Student - MobileNetV2 (α=0.5)

Two student models, which differ in the width of the channels as α=0.5 and α=0.35, are chosen for a thorough analysis of the DAD-DTKD approach. The width multiplier α defines the number of channels in each convolutional block of MobileNetV2 [10, 11]. The method provides a simple and reproducible means of compression by reducing the complexity of the network architecture. These width multipliers are standard values commonly reported in the literature for MobileNetV2-based compression studies. Here, α is the width control parameter that is used to reduce the number of channels in each layer by half in the MobileNetV2 [11].

The first configuration with α=0.5 corresponds to the moderate compression case when each layer of the student architecture has half the number of channels of the teacher model. Such a setup ensures that the student model has enough capacity to perform gesture recognition correctly. However, at the same time, it allows substantial reductions in memory usage and computation costs. The second configuration with α=0.35 refers to the aggressive compression setting, which leads to a more compact model architecture. This makes it well-suited for deployment in resource-limited environments such as edge devices. This setup is intentionally chosen to verify the ability of DAD-DTKD to maintain accuracy under the most stringent architecture constraints.

3.6. Why DAD-DTKD Suits ISL

The ISL dataset comprises visually similar alphabet pairs like M/N, U/V, R/W, where subtle differences in hand configuration are the only distinguishing features. Such similar gestures produce similar logit distributions, making accurate knowledge transfer to the student particularly challenging. A fixed temperature setting fails to adapt to such critical class pairs and treats all samples equally, regardless of their inter-class similarity. Hence, it may produce wrong classification of visually ambiguous gestures by the student model. The dynamic temperature setting of DAD-DTKD, on the other hand, adjusts T value based on the standard deviation of the teacher's logit distribution, accommodating the varying difficulty levels across gesture classes. For visually similar gestures, the teacher model produces a relatively flat logit distribution reflecting lower confidence. In such a scenario, the DAD-DTKD assigns a higher T value, which provides greater smoothing and reveals more information about the relationships among competing classes. For visually distinct gestures, the teacher model produces sharp logit distribution, reflecting high confidence, and hence DAD-DTKD assigns a lower T value to retain the stronger class distinctions in the teacher output. This distribution-aware dynamic temperature mechanism makes DAD-DTKD particularly well suited for fine-grained

recognition tasks such as ISL, where preserving inter-class distinctions is critical for reliable gesture classification. Both the DAD-DTKD trained student models, although compressed, are further optimized for edge deployment through TFLite quantization. Since Raspberry Pi has limited memory and processing capability, quantization is applied further to reduce model size beyond what is achieved through knowledge distillation alone. TensorFlow Lite is a lightweight inference framework designed for edge and mobile deployment is employed to convert these trained student models to an optimized TFLite format.

In this process, the 32-bit floating point weight representations are converted to 16-bit floating point representations and help to achieve approximately 2× reduction in model size with minimal impact on recognition accuracy. Float16 quantization is specifically chosen instead of more aggressive INT8 quantization to preserve the subtle inter-class distinctions critical for reliable ISL gesture recognition. Both student models - MobileNetV2 α=0.5 and α=0.35 - are converted to TFLite Float16 format and evaluated on Raspberry Pi for real-time deployment. The overall pipeline of the proposed approach is shown in Figure 2. Both the teacher and student models receive the preprocessed and augmented ISLRTC ISL data as input. The pre-trained, fixed teacher MobileNetV2 model with α=1.0 predicts the soft logits. The DAD-DTKD calculates the dynamic temperature T based on the standard deviation of the logit values. The student model is updated using distillation loss. Both trained student models are subsequently converted to TFLite Float16 format and deployed on a Raspberry Pi for real-time ISL gesture recognition.

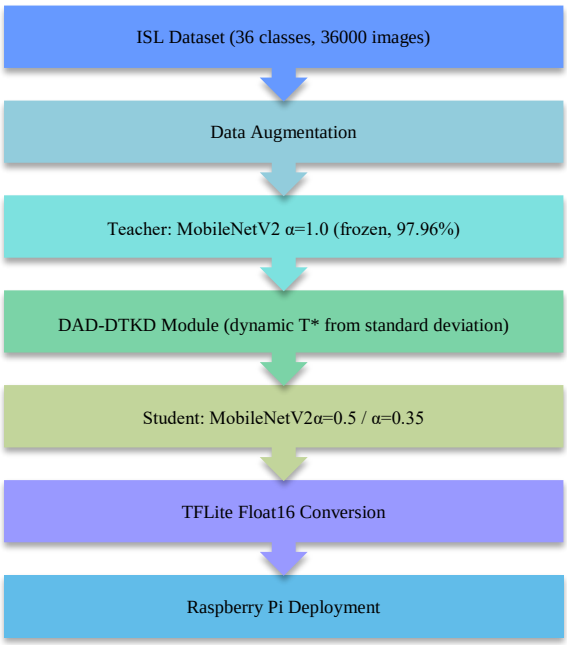

Fig. 2 Proposed DAD-DTKD pipeline for ISL gesture recognition-from knowledge distillation training to edge deployment

4. Experimental Setup

4.1. Experimental Environment

All experiments are conducted on a Linux workstation with an AMD Ryzen 5 7600 processor and 32GB RAM, with GPU-accelerated training using NVIDIA GeForce RTX 4070 with 12GB VRAM. The TensorFlow version used is 2.20.0, and the Python version is 3.10. TensorFlow Lite framework is used for quantization and for edge inference. For the deployment, Raspberry Pi 4 Model B with 8GB RAM and ARM Cortex-A72 quad-core processor clocked at 1.8GHz is used. The Raspberry Pi OS is 32-bit, and the camera module is V2, having 8MP with no IR and is suitable for varying lighting conditions. TensorFlow Lite for edge inference.

4.2. Training Configuration

The training configuration for the teacher and student models is as follows

4.2.1. Teacher Model:

- Architecture: MobileNetV2 (α=1.0), pre-trained on ImageNet and fine-tuned on the ISL dataset
- Epochs: 250 with Adam optimizer
- Initial learning rate: 1×10^{-3}, fine-tuning learning rate: 1×10^{-5}
- Input image size: 96×96 pixels, batch size: 16

4.2.2. DAD-DTKD Student Models:

- Architecture: MobileNetV2 (α=0.5 and α=0.35)
- Epochs: 150 with Adam optimizer, learning rate: 1×10^{-3}
- Reference temperature: T_ref = 5.0, temperature range: 1.0 – 5.0
- Loss weighting coefficient: α = 0.5 (equal weighting of CE and KD loss)
- Batch size: 16

4.3. Evaluation Metrics

The following evaluation metrics are used for the comprehensive evaluation of the proposed DAD-DTKD ISL recognition system.

4.3.1. Accuracy

Accuracy is used as the primary metric for gesture recognition performance in this study. As given by Equation (6), it is the ratio of correctly classified gesture samples to total validation samples [6, 18].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

4.3.2. Precision

Precision is used to measure how reliable positive predictions are, and as given in Equation (7), it is calculated as the number of true positive predictions divided by the total positive predictions per class [6, 18, 26].

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

4.3.3. Recall

Measures how well the model detects each gesture class and is calculated as the number of true positive predictions divided by the total number of actual positive instances, as shown in Equation (8) [6, 18].

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

4.3.4. F1-Score

F1-score is computed as the harmonic mean of precision and recall, and it is the most comprehensive single metric for multi-class recognition given by Equation (9) [6, 18].

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

4.3.5. Inference Time

A critical metric for real-time edge deployment is inference time and is calculated using Equation (10).

It is the average time in milliseconds per image during validation, and it is measured on both the training machine and the Raspberry Pi [6, 18].

$$Inference\ Time = \frac{T_{total}}{N} \times 1000ms \quad (10)$$

4.3.6. Frames Per Second (FPS)

FPS is derived from inference time, and it indicates the real-time capability of the deployed model, given by Equation (11) [6, 18].

$$FPS = \frac{1000}{T_{inf}} \quad (11)$$

4.3.7. Model Size

Model size is the size of the saved model file in megabytes and can be calculated for both Keras and TFLite formats. It directly reflects the memory footprint on the edge device.

4.3.8. Compression Ratio (CR)

CR, as given by Equation (12), is the ratio of teacher model size to student model size and quantifies the degree of compression achieved by DAD-DTKD [6, 18].

$$CR = \frac{P_{teacher}}{P_{student}} \quad (12)$$

5. Results and Discussion

This section presents the results of all teacher-student model configurations. The results are discussed with respect to three research questions: RQ1, RQ2, and RQ3. Table 2 summarizes all performance metrics, and Figures 3-5 provide a visual comparison of the achieved results.

Table 2. Comparative performance evaluation of baseline, vanilla KD, DAD-DTKD, and quantized models in terms of accuracy, compression, and inference efficiency

#	Model	α	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Inference Time (ms)	FPS	Model Size (MB)	CR
1	MobileNetV2 (baseline)	1.0	98.69	98.72	98.69	98.70	57.67	17.34	33.93	1×
2	MobileNetV2 (vanilla KD)	0.5	91.96	91.96	91.96	91.96	39.69	25.19	5.76	5.89×
3	MobileNetV2 (vanilla KD)	0.35	91.92	91.92	91.92	91.92	8.73	114.53	4.64	7.31×
4	DAD-DTKD	0.5	97.96	97.96	97.96	97.96	39.69	25.19	5.76	5.89×
5	DAD-DTKD	0.35	96.92	96.92	96.92	96.92	8.73	114.53	4.64	7.31×
6	Quantized DAD-DTKD	0.5	96.96	96.96	96.96	96.96	4.69	213.19	3.76	9.02×
7	Quantized DAD-DTKD	0.35	95.92	95.92	95.92	95.92	3.63	275.53	2.64	12.8×

5.1. Accuracy Analysis

The recognition performance of the teacher and student model configurations is evaluated in terms of accuracy, precision, recall, and F1-score. The MobileNetV2 teacher model with α=1.0 achieves 98.69% and thus establishes a strong reference knowledge source. The recognition accuracy of all model configurations is evaluated and compared, as summarized in Table 2 and illustrated in Figure 3. The F1 score for the teacher model is 98.69%, precision 98.72% and recall 98.69%. This indicates the teacher model is performing well and confirms the teacher model is a well-trained knowledge source for transferring knowledge to the compact student models.

Before applying DAD-DTKD to student models, the classic KD approach is used, and two student models are trained with this knowledge; these two models are Vanilla KD with α=0.5 and α=0.35. These models achieved an accuracy

of 91.96% and 91.92%, resulting in around 6 % drop in accuracy as compared to the teacher model. This accuracy drop indicates that transferring knowledge to compact student models using a fixed temperature does not fully preserve the teacher's recognition performance.

The DAD-DTKD distilled student models with α=0.5 achieved 97.96% accuracy, which is 6 % higher than the vanilla KD student model, whereas the DAD-DTKD student with α=0.35 achieved 96.92% accuracy, which is improved by 5 % over vanilla KD.

This shows consistent improvement across both configurations. DAD-DTKD determines the temperature according to the teacher's output distribution for each sample. This allows the knowledge transfer to adapt to the characteristics of individual samples. Furthermore, by considering the complete output distribution rather than only the highest output score, DAD-DTKD provides the student model with more information about the relative scores of different sign classes, rather than relying only on the highest output score.

This relative score information, often referred to as dark knowledge, indicates how the teacher relates different classes for a given input. For example, two visually similar ISL signs may receive similar output scores from the teacher, even when only one is the correct class. This information helps the student learn the relationships among similar sign classes. Since the student model has fewer parameters and a lower learning capacity than the teacher, such additional information provides useful guidance during training. The student can therefore learn the teacher's class relationships more effectively and retain higher recognition performance despite its smaller model size.

Direct comparison with existing state-of-the-art methods is limited because DTKD and other dynamic temperature methods have mainly been evaluated on general image classification datasets rather than SLR or ISL tasks. On CIFAR-100 and ImageNet, DTKD reports accuracy improvements over vanilla KD ranging from approximately 0.76% to 2.83%, depending on the teacher–student model configuration.

In comparison, DAD-DTKD achieves an improvement of approximately 6.00 percentage points for α = 0.5 and 5.00 percentage points for α = 0.35 over vanilla KD on the ISL dataset. Although these results are obtained on different datasets and tasks and therefore do not represent a direct benchmark comparison, the larger improvement observed with DAD-DTKD suggests that considering the complete teacher output distribution can be beneficial for ISL gesture recognition, particularly when different sign classes have similar visual characteristics.

These results demonstrate that DAD-DTKD consistently improves the recognition accuracy of both student configurations while maintaining the same student model size as the corresponding vanilla KD models. Thus, DAD-DTKD effectively preserves recognition performance under model compression and positively addresses RQ1.

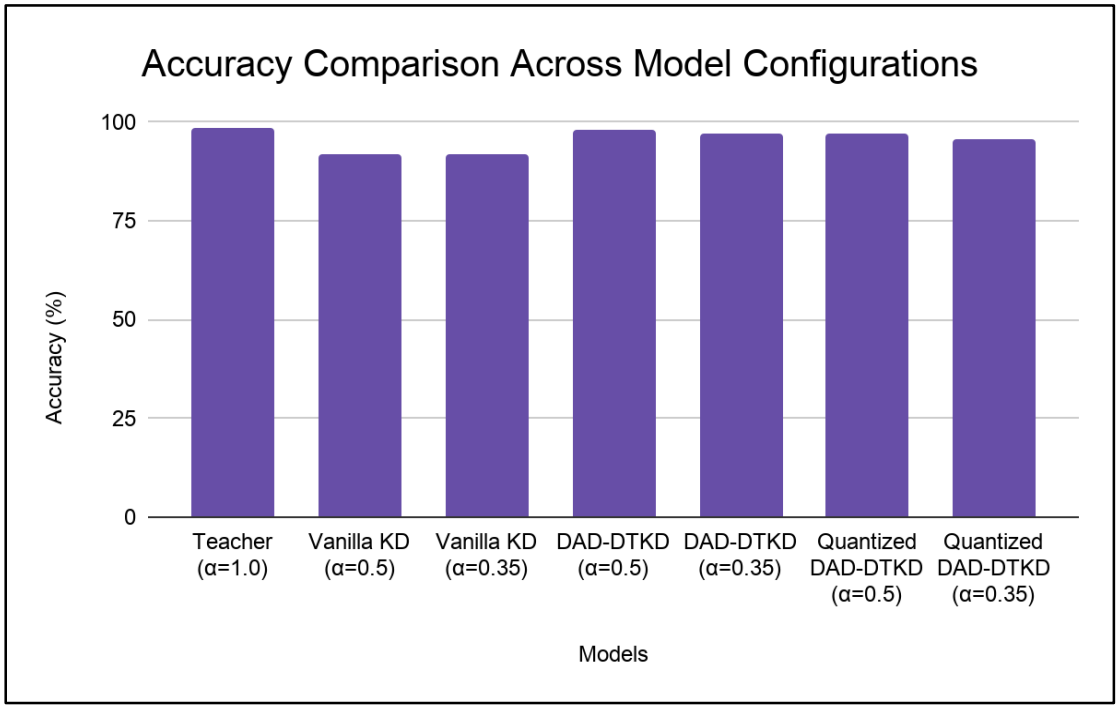

Fig. 3 Accuracy comparison across teacher, vanilla KD, DAD-DTKD, and quantized DAD-DTKD configurations (α=0.5 and α=0.35)

5.2. Model Compression Analysis

The model size and Compression Ratio (CR) are analyzed across the different model configurations and are summarized in Table 2 and illustrated in Figure 4. The original teacher model, MobileNetV2, has a size of 33.93MB.

The DAD-DTKD distilled student models with α=0.5 have a size of 5.76MB, resulting in a 5.89× reduction, and the student with α=0.35 has 4.64MB, which is a 7.31× reduction. This shows a significant reduction achieved while maintaining high recognition accuracy of 97.96% and 96.92%, respectively.

Reducing the width multiplier α decreases the number of channels in the MobileNetV2 layers, which reduces the number of parameters and the overall model size. Such a reduction can limit the model's capacity to learn useful discriminative features and may result in lower recognition accuracy.

However, DAD-DTKD provides the student model with detailed inter-class relational knowledge from the teacher. This additional knowledge guides the student during learning and reduces its dependence on independently learning these class relationships. Therefore, the compact student model can maintain high recognition performance despite having fewer parameters.

The model with α=0.5 is moderately compressed with 5.89× reduction in model size, whereas the student model with α=0.35 is aggressively compressed with 7.31× reduction in model size and is suitable for highly constrained devices. Both configurations maintain acceptable accuracy and address RQ2.

The systematic evaluation of two student configurations addresses RQ2, demonstrating that DAD-DTKD performs effectively across both moderate and aggressive compression scenarios.

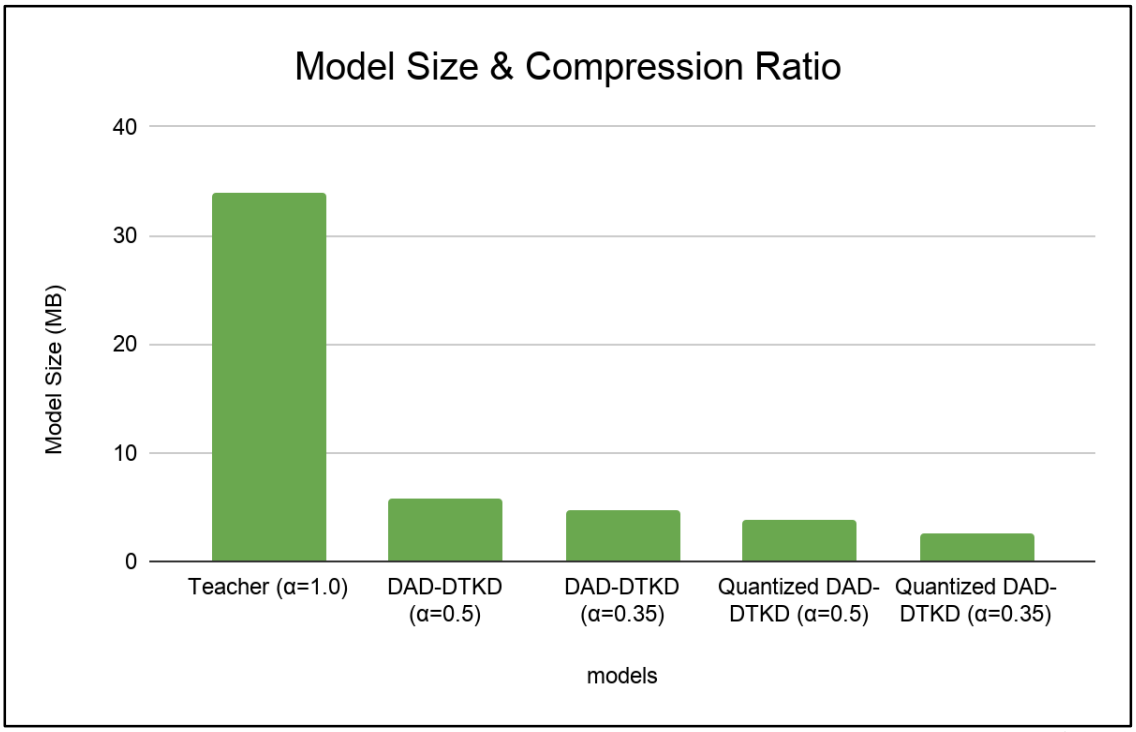

Fig. 4 Model size and Compression Ratio (CR) comparison across teacher, DAD-DTKD, and quantized DAD-DTKD configurations

5.3. Inference Efficiency Analysis

Inference time and FPS are also evaluated to address the efficiency dimension of the proposed DAD-DTKD. The baseline model α=1.0 has an inference time of 57.67ms and 17.34 FPS. This shows the model is large and comparatively slower during inference, further motivating the need for compression.

The DAD-DTKD student with α=0.5 has an inference time of 39.69ms and 25.19 FPS, so the model is both faster as well as 5.89× smaller and DAD-DTKD α=0.35 has 8.73ms and 114.53 FPS. The improvement in inference efficiency is mainly due to the reduced number of channels and parameters in the compact student models, which reduces the computational operations required during inference.

Fewer computations result directly into lower inference time and higher FPS. The α=0.35 DAD-DTKD student demonstrates that aggressive compression does not necessarily compromise inference speed, addressing RQ2.

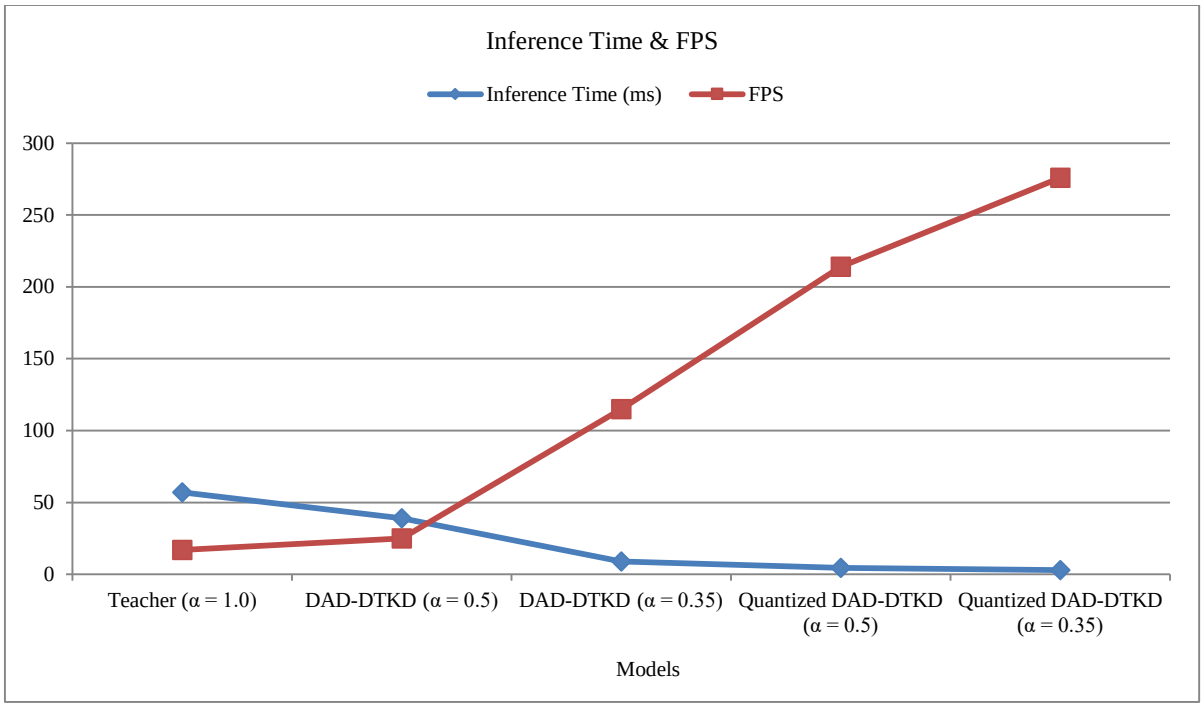

Fig. 5 Inference time and FPS comparison across teacher, DAD-DTKD, and quantized DAD-DTKD configurations deployed on Raspberry Pi

5.4. Effect of Quantization

To address RQ3, both DAD-DTKD student models are converted to TFLite Float16 format and evaluated for performance after quantization. The quantized student model with α=0.5 has achieved 96.96%, with only 1.0% drop from DAD-DTKD accuracy of 97.96%. The quantized student model with α=0.35 achieves an accuracy of 95.92% with only 1.0 % drop from the DAD-DTKD accuracy of 96.92%.

This minimal accuracy degradation shows that the proposed DAD-DTKD models are compatible with Float16 quantization. Float16 quantization reduces the numerical precision of the model weights from 32-bit to 16-bit, while retaining the learned weights of the model. Therefore, the learned knowledge of the student model is largely preserved, resulting in only a small reduction in accuracy despite the additional model compression.

As shown in Figures 3 and 4, the application of quantization has provided additional model size reduction, with the quantized student model with α=0.5 resulting in a 3.76MB model size and with a CR of 9.02×. The Quantized α=0.35 student model has an even smaller size of just 2.64MB, resulting in 12.85× CR. Thus, the combined DAD-DTKD + quantization achieves the highest compression ratio among the evaluated configurations.

As illustrated in Figure 5, the inference time of both models when deployed on Raspberry Pi is 4.69ms and 3.63ms, respectively, which is faster than their respective unquantized inference times, 39.69ms (α=0.5) and 8.73ms (α=0.35). This shows the real-time deployment capability of the proposed models. These results confirm that DAD-DTKD compressed models are fully compatible with Float16 quantization, affirmatively addressing RQ3.

5.5. Ablation Study

An ablation study is conducted to validate the contribution of the distribution-aware dynamic temperature setting in DAD-DTKD. The same student architectures are used in both cases, with only the temperature strategy changed from fixed T in vanilla KD versus dynamic temperature in DAD-DTKD.

As shown in Table 2, DAD-DTKD achieves 97.96% accuracy for α=0.5 compared to 91.96% with vanilla KD, resulting in an improvement of 6.0 %. Similarly, for α = 0.35, DAD-DTKD achieves 96.92% compared to 91.92% for vanilla KD, giving an improvement of 5.0 %. The improvement is observed for both student configurations while the student architecture and model size remain unchanged. This indicates that the dynamic temperature strategy contributes to the improved recognition performance of DAD-DTKD.

Figure 6 illustrates the relationship between teacher confidence and dynamically computed temperature T. A negative relationship is observed, where T decreases from approximately 1.65 to 1.1 as teacher confidence increases from 0.5 to 0.9. This enables the temperature to adapt to the teacher's output distribution for individual samples.

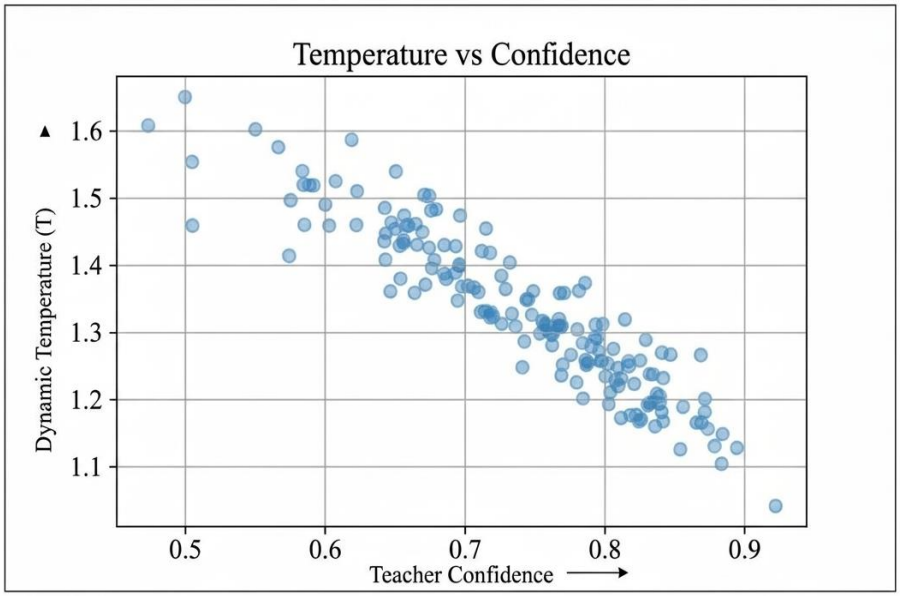

Fig. 6 Relationship between teacher model confidence and DAD-DTKD computed dynamic temperature T during ISL distillation training. Higher teacher confidence corresponds to lower T, confirming the distribution-aware temperature adaptation mechanism

Figure 7 shows the distribution of T values across all ISL training samples. The T values range from 1.1 to 1.65, with the majority concentrated between 1.2 and 1.45, forming an approximately normal distribution centered around 1.3. This indicates that the proposed method assigns a range of temperature values across the training samples rather than using a single fixed temperature, and demonstrates DTS across the 36 ISL gesture classes.

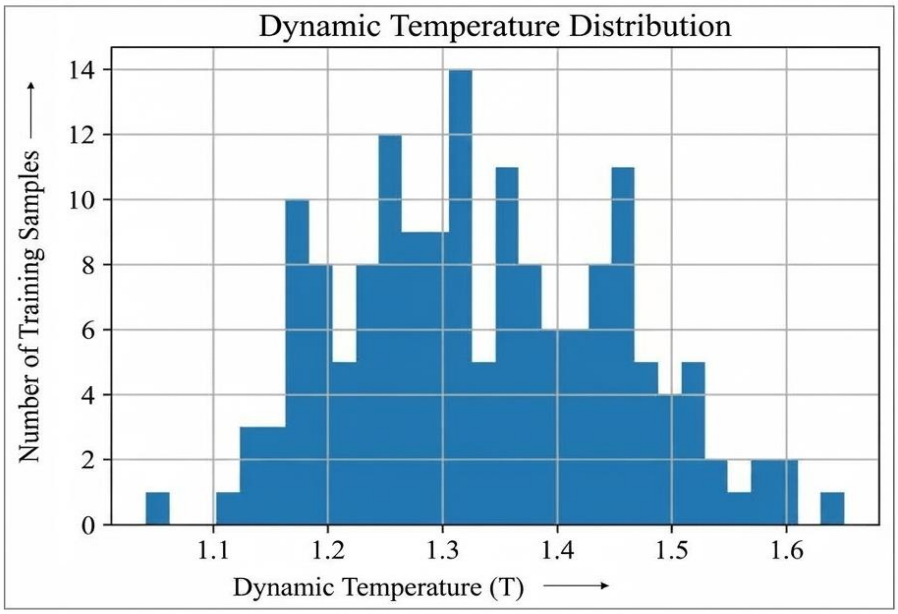

Fig. 7 Distribution of dynamically computed temperature values T across the ISL training samples. The approximately normal distribution centered around T*≈1.3 indicates stable and consistent temperature setting by DAD-DTKD across 36 ISL gesture classes

Figure 8 presents the scatter plot of T against training loss. No strong correlation is observed between T and loss values - confirming that T is determined purely by the teacher logit distribution and not by training loss. This distinguishes DAD-DTKD from loss-based dynamic temperature methods such as DTS and supports the distribution-aware nature of the proposed temperature setting. Together, the quantitative comparison and three visualizations validate that DAD-DTKD adapts temperature as theoretically described, supporting distribution-aware dynamic temperature as an effective mechanism for knowledge distillation in ISL gesture recognition.

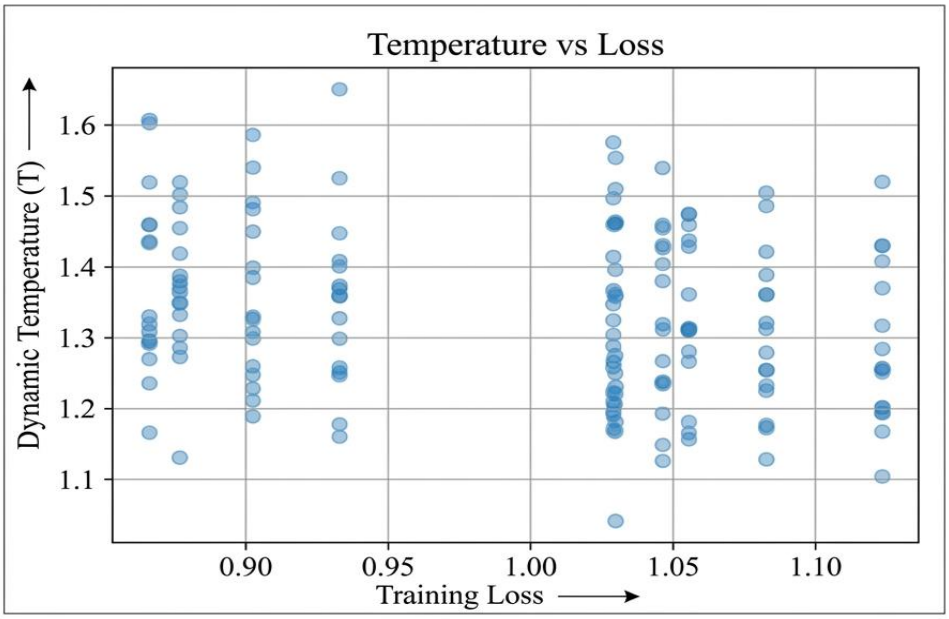

Fig. 8 Scatter plot of dynamic temperature T against training loss during DAD-DTKD distillation

6. Conclusion

Sign language recognition is critical for the deaf-mute community, for whom VPRN-based SLR systems serve as an effective assistive technology. However, the large model size of VPRN-based SLR systems hinders their deployment on resource-constrained edge devices typically used at the user end. This study successfully applies the DAD-DTKD framework to the ISL dataset, evaluating two student configurations- MobileNetV2 α=0.5 and α=0.35, deployed on Raspberry Pi with TFLite Float16 quantization.

The proposed DAD-DTKD effectively compresses MobileNetV2 while preserving recognition accuracy, achieving a consistent 5.00–6.00 % improvement over vanilla KD across both student configurations, affirmatively addressing RQ1. The moderately compressed student α=0.5 achieves 5.89× compression with 97.96% accuracy, while the aggressively compressed student α=0.35 achieves 7.31× compression with 96.92% accuracy, thereby systematically addressing RQ2. Both configurations demonstrate high recognition performance despite undergoing significant architectural compression. Both student models are further compatible with Float16 quantization, achieving a maximum compression ratio of 12.85× with 95.92% accuracy and 3.63ms inference time on Raspberry Pi, confirming real-time deployment capability and addressing RQ3.

The experimental results validate the generalization of DAD-DTKD to the ISL domain, establishing knowledge distillation-based compression as a viable and effective strategy for assistive technology edge deployment. Motivated by these results, future work will explore the application of DAD-DTKD to other regional sign languages and investigate the combination of KD with complementary compression techniques such as pruning and INT8 quantization for further efficiency gains.

Conflicts of Interest

The author(s) declare that no conflict of interest exists in relation to the publication of this paper.

Funding Statement

Authors state that no financial assistance is required in the proposed research and publication.

Acknowledgments

The authors are grateful to the research center for the invaluable support.

References

[1] World Health Organization, Deafness and Hearing Loss, *WHO Fact Sheet*, 2026. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[2] Rina Damdoo, and Praveen Kumar, "An Integrative Survey on Indian Sign Language Recognition and Translation," *IET Image Processing*, vol. 19, no. 1, pp. 1-29, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[3] Aamir Wali et al., "Recent Progress in Sign Language Recognition: A Review," *Machine Vision and Applications*, vol. 34, no. 6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Karly Kudrinko et al., "Wearable Sensor-based Sign Language Recognition: A Comprehensive Review," *IEEE Reviews in Biomedical Engineering, vol. 14, pp. 82-97, 2021*. [CrossRef] [Google Scholar] [Publisher Link]
[5] Ang Ji et al., "Dataglove for Sign Language Recognition of People with Hearing and Speech Impairment Via Wearable Inertial Sensors," *Sensors*, vol. 23, no. 15, pp. 1-20, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[6] Shagun Katoch, Varsha Singh, and Uma Shanker Tiwary, "Indian Sign Language Recognition System using SURF with SVM and CNN," *Array*, vol. 14, pp. 1-9, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[7] Tangfei Tao et al., "Sign Language Recognition: A Comprehensive Review of Traditional and Deep Learning Approaches, Datasets, and Challenges," *IEEE Access, vol. 12, pp. 75034-75060, 2024*. [CrossRef] [Google Scholar] [Publisher Link]
[8] Karen Simonyan, and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv*, pp. 1-14, 2015. [CrossRef] [Google Scholar] [Publisher Link]
[9] Kaiming He et al., "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016*. [Google Scholar] [Publisher Link]
[10] Christian Szegedy et al., "Going Deeper with Convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9, 2015. [Google Scholar] [Publisher Link]
[11] Venus Singla, Seema Bawa, and Jasmeet Singh, "Enhancing Indian Sign Language Recognition through Data Augmentation and Visual Transformer," *Neural Computing and Applications*, vol. 36, no. 24, pp. 15103-15116, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[12] M.K. Vidhyalakshmi et al., "Indian Sign Language Recognition using Transfer Learning with Efficient Net," *Smart Science*, vol. 12, no. 2, pp. 269-280, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[13] Soumen Das, Saroj Kr. Biswas, and Biswajit Purkayastha, "Occlusion Robust Sign Language Recognition System for Indian Sign Language Using CNN and Pose Features," *Multimedia Tools and Applications*, vol. 83, no. 36, pp. 84141-84160, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[14] Sakshi Sharma, and Sukhwinder Singh, "ISL Recognition System using Integrated Mobile-Net and Transfer Learning Method," *Expert Systems with Applications*, vol. 221, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[15] Andrew G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv*, pp. 1-9, 2017. [CrossRef] [Google Scholar] [Publisher Link]
[16] Mark Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 4510-4520, 2018*. [CrossRef] [Google Scholar] [Publisher Link]
[17] Andrew Howard et al., "Searching for MobileNetV3," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1314-1324, 2019. [Google Scholar] [Publisher Link]
[18] Shilpa Ingoley, and Jagdish Bakal, "Efficient Models based on Deep Learning Technique for Indian Sign Language," *Procedia Computer Science*, vol. 258, pp. 1980-1992, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[19] Defu Liu et al., "A Survey of Model Compression Techniques: Past, Present, and Future," *Frontiers in Robotics and AI*, vol. 12, pp. 1-16, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[20] Ali Alqahtani, Xianghua Xie, and Mark W. Jones, "Literature Review of Deep Network Compression" *Informatics*, vol. 8, no. 4, pp. 1-12, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[21] Kequan Lin, Zhenjie Lin, and Shiran Wang, "Model Compression for IoT Edge Devices based on Knowledge Distillation," *International Journal of Pattern Recognition and Artificial Intelligence*, pp. 1-21, 2026. [CrossRef] [Google Scholar] [Publisher Link]
[22] Yang He, and Lingao Xiao, "Structured Pruning for Deep Convolutional Neural Networks: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 5, pp. 2900-2919, 2024*. [CrossRef] [Google Scholar] [Publisher Link]
[23] Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi, "A Survey on Deep Neural Network Pruning: Taxonomy, Comparison, Analysis, and Recommendations," *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 12, pp. 10558-10578, 2024*. [CrossRef] [Google Scholar] [Publisher Link]
[24] Amir Gholami et al., *A Survey of Quantization Methods for Efficient Neural Network Inference*, 1st ed., Low-Power Computer Vision, Chapman and Hall/CRC, pp. 1-33, 2022. [Google Scholar] [Publisher Link]
[25] Kalyan Cherukuri, and Aarav Lala, "Low-Rank Matrix Approximation for Neural Network Compression," *arXiv preprint arXiv*, pp. 1-5, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[26] Jianping Gou et al., "Knowledge Distillation: A Survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789-1819, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[27] Yukang Wei, and Yu Bai, "Dynamic Temperature Knowledge Distillation," *arXiv preprint arXiv*, pp. 1-15, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[28] Zinah Raad Saeed et al., "A Systematic Review on Systems-based Sensory Gloves for Sign Language Pattern Recognition: An Update from 2017 to 2022," *IEEE Access, vol. 10, pp. 123358-123377, 2022*. [CrossRef] [Google Scholar] [Publisher Link]
[29] Deepti Aggarwal et al., "Sign Language Prediction using Machine Learning Techniques: A Review," *2023 Second International Conference on Electronics and Renewable Systems, Tuticorin, India, pp. 1296-1300, 2023*. [CrossRef] [Google Scholar] [Publisher Link]

[30] Sibgat Ul Islam et al., "Dynamic Temperature Scheduler for Knowledge Distillation," *2025 IEEE International Conference on Big Data (BigData)*, Macau, China, pp. 1412-1417, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[31] Yu-e Lin et al., "ATMKD: Adaptive Temperature Guided Multi-Teacher Knowledge Distillation," *Multimedia Systems*, vol. 30, no. 5, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[32] Wenjie Liu et al., "Be your Own Doctor: Temperature Scaling Self-Knowledge Distillation for Medical Image Classification," *Neurocomputing*, vol. 638, pp. 1-12, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[33] Ismail Lamaakal et al., "An Explainable Hybrid CNN-Transformer Model for Sign Language Recognition on Edge Devices using Adaptive Fusion and Knowledge Distillation," *Scientific Reports*, vol. 16, no. 1, pp. 1-23, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[34] Shunzhi Yang et al., "Adaptive Temperature Distillation Method for Mining Hard Samples' Knowledge," *Neurocomputing*, vol. 636, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[35] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the Knowledge in a Neural Network," *arXiv preprint*, pp. 1-9, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[36] Atharva Dumbre, Indian Sign Language (ISLRTC Referred), *Kaggle Dataset*, 2023. [Online]. Available: https://www.kaggle.com/datasets/atharvadumbre/indian-sign-language-islrtc-referred