International Journal of Science, Strategic Management and Technology
Volume **02** Issue **11** November-2026 | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Real-time American Sign Language Alphabet Recognition Using Knowledge Distillation-Based Lightweight Deep Learning: Stage I – ResNet50 Teacher Model

Mrs. Anuja Phapale
Assistant Professor (Project Guide), B.Tech Information Technology
Department of Information Technology
AISSMS Institute of Information Technology
Pune, Maharashtra

Atharv Jagdale
Student, B.Tech Information Technology
Department of Information Technology
AISSMS Institute of Information Technology
Pune, Maharashtra

Vivaan Kukreja
Student, B.Tech Information Technology
Department of Information Technology
AISSMS Institute of Information Technology
Pune, Maharashtra

Aditya Morey
Student, B.Tech Information Technology
Department of Information Technology
AISSMS Institute of Information Technology
Pune, Maharashtra

https://doi.org/10.55041/ijsmt.v2i11.038

Cite this Article: Jagdale, A., Kukreja, V. & Morey, A. (2026). Real-time American Sign Language Alphabet Recognition Using Knowledge Distillation-Based Lightweight Deep Learning: Stage I – ResNet50 Teacher Model. International Journal of Science, Strategic Management and Technology, *02*(11), 1-9. https://doi.org/10.55041/ijsmt.v2i11.038

License: This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

ABSTRACT

Real-time recognition of the American Sign Language (ASL) finger-spelling alphabet can support communication for people with hearing and speech impairments. This project follows a teacher–student plan in which a lightweight model will later be distilled from a ResNet50 teacher; this paper reports the completed first stage, the ResNet50 teacher, on the 29-class Kaggle ASL Alphabet dataset (87,000 images). The dataset was prepared with duplicate-aware preprocessing, in which duplicate groups cover 84,550 images, and was divided by group into training, validation and internal test splits to reduce near-duplicate leakage; the 28 official external test images were kept separate.

The teacher was trained for 10 epochs on 224×224 images. Its best checkpoint (epoch 10) reached a validation loss of 0.00048 and a validation accuracy of 99.9923%, with a training accuracy of 99.9113% at epoch 10. On the 13,050-image internal test split the teacher reached 99.9847% accuracy and a macro F1-score of 99.9847%, with 13,048 correct and 2 incorrect predictions, both from class del. A separate checkpoint (teacher_best_aug.pth) evaluated on 8,700 test images reached 94.4253% accuracy and is reported separately. These are internal-split results and not a measure of real-world accuracy. The external test evaluation, the MobileNetV2 student, Knowledge Distillation, real-time validation and robustness testing are planned future work.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

Keywords: ASL Alphabet Recognition; ResNet50 Teacher Model; Duplicate-Aware Dataset Splitting; Internal Test Evaluation; Confusion Matrix Analysis; Knowledge Distillation.

1. INTRODUCTION

Sign languages convey meaning through hand shapes, movement and facial expression. In American Sign Language (ASL), finger-spelling represents each English letter by a hand shape and is used for names, technical terms and words without an established sign. Most letters are static, whereas J and Z are articulated with motion. Automatic recognition of finger-spelled letters turns these hand shapes into text and can reduce the dependence of people with hearing or speech impairments on human interpreters in everyday communication; the work is therefore linked to SDG 10 (Reduced Inequalities), although a research model by itself does not reduce inequality.

Deep learning has made static ASL alphabet classification highly accurate on benchmark data: studies on the 87,000-image, 29-class ASL Alphabet dataset report accuracies from about 97% to 99.98% [2]–[6]. For practical use, however, accuracy alone is not enough. A system that processes a webcam stream on everyday hardware must also be small and fast. Virag and Katona [1] report accurate real-time recognition with a RetinaNet detector on a ResNet-50 backbone, with an average accuracy above 93% in a study with 30 participants, but the backbone is comparatively heavy and computational cost is left for further optimisation.

Knowledge Distillation (KD) [14] trains a small student network to reproduce the softened output probabilities of a larger teacher in addition to learning from the true labels. KD has produced compact models for sign languages other than ASL, for example a 2.7 M-parameter model for 20 classes of German and Chinese sign language [13]. This project follows a teacher–student plan in which a ResNet50 teacher [16] will later guide a lightweight MobileNetV2 student [15] for the 29-class ASL alphabet. This paper reports only the first stage, the ResNet50 teacher; the student and KD stages are described in Future Research (Section 12).

2. STATEMENT OF THE STUDY

From the review of literature, the following gaps are identified within the reviewed studies: (G1) accurate real-time ASL alphabet recognition has been shown with a heavy ResNet-50-based backbone [1], and accuracy on clean static data is close to saturated [2]–[6]; (G2) KD has produced compact sign-language models [13], but its use for the ASL alphabet is not addressed in the reviewed alphabet studies [2]–[7]; (G3) accuracy and computational efficiency are rarely reported together for a compact ASL alphabet model; and (G4) overall accuracy hides confusion between similar letters such as A and M [4], so per-class results are needed. These gaps do not claim that KD has never been applied to sign-language recognition elsewhere. This paper contributes to G4 for the ResNet50 teacher, the required first step towards G1–G3, by preparing the data with duplicate-aware splitting, training the teacher and analysing its per-class results; G1–G3 are left to future research.

Problem statement. Let $x \in \mathbb{R}^{3\times224\times224}$ be a preprocessed hand image and $y \in \{0,\ldots,28\}$ its class. Let f_t denote the ResNet50 teacher, which produces 29 logits. The problem is to train f_t with the true labels and the cross-entropy loss so that it classifies x into one of the 29 classes with high accuracy on duplicate-aware validation and internal test splits, and so that it can later serve as the frozen teacher for the distillation of a lightweight student.

3. NEED OF THE STUDY

Accuracy on clean static ASL alphabet data is already close to saturated in the reviewed studies [2]–[6]. A practical system that processes a webcam stream on everyday hardware must also be small and fast, and the reviewed ASL alphabet studies do not address KD [2]–[7]. A teacher whose data preparation, training and evaluation are documented is the required first step of a teacher–student plan, because the student can only be as reliable as the teacher that guides it.

Overall accuracy can hide confusion between visually similar letters such as A and M [4], so per-class analysis is needed. Near-duplicate frames in the dataset can also raise internal accuracy, so duplicate-aware splitting and an honest account of the limits of internal-split testing are needed before any result is quoted. This study is needed to provide a reproducible teacher stage on which the later student and KD stages can be built.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

4. SCOPE OF THE STUDY

1. **Dataset and Model Scope:** The study uses the public Kaggle ASL Alphabet dataset [18], which contains static single-hand images of 29 classes. Only the ResNet50 teacher is covered. No novelty is claimed for ResNet50, KD or the dataset themselves.

2. **Technical Scope:** The study covers dataset preparation, data preprocessing, teacher training, validation, internal test evaluation and confusion matrix analysis. The MobileNetV2 student, Knowledge Distillation, the external test evaluation, real-time validation and robustness testing are not covered and are listed in Future Research.

3. **Limitations:** The teacher results come from a single run and from the internal split of one dataset, so they must not be read as real-world ASL recognition accuracy. The validation split was also used to select the checkpoint, so the validation accuracy is not an independent estimate. Near-duplicate frames remain a leakage risk despite group-aware splitting, and the random seed of the split is not reported. The dataset has static single-hand images with undocumented capture conditions, so clean-split accuracy may not transfer to other conditions such as webcam use. The official external test set has only 28 images and has not yet been evaluated. A separately evaluated checkpoint (teacher_best_aug.pth) reached 94.4253% on a different 8,700-image test set, so results must be quoted with the checkpoint and test set used. The optimiser, learning rate, batch size and initialisation of the teacher run are not reported here.

4. **Out of Scope:** The dynamic letters J and Z, two-hand gestures, words and sentences and other sign languages are out of scope, and no study with sign-language users has been carried out. This work is a research stage and not a replacement for a human interpreter.

5. REVIEW OF LITERATURE

The studies on ASL alphabet and sign-language recognition and the techniques on which the proposed method rests are summarised in Table 1. All figures are as reported by the cited authors on their own datasets and protocols; they are not results of this project and must not be compared with each other or with the results of this paper.

A. Image-based ASL alphabet classifiers

Alsharif et al. [2] fine-tuned five pretrained networks on the ASL Alphabet dataset with five-fold cross-validation; ResNet-50 gave the best accuracy (99.98%) and the Vision Transformer the lowest (88.59%). This supports ResNet50 as a strong teacher for this data, but the models were neither compressed nor distilled and real-time use was left as future work. Smaller CNNs are also reported: Pangaribuan et al. [3] apply a three-convolution-layer CNN to the ASL alphabet and report 97% test accuracy without live-video or varied-lighting tests, and Abdullah et al. [4] use a compact CNN on 32×32 images with 99.77% test accuracy, while noting systematic confusion of the visually similar letters A and M. Al-Ghanim et al. [5] report 99.97% on a 26-class fingerspelling set and Bala et al. [7] 100% on Sign Language MNIST without J and Z. Aly and Fathi [6] propose a hybrid Transformer-CNN that reaches 99.97% on the ASL Alphabet dataset at 110 FPS and 5.0 GFLOPs, without knowledge distillation.

B. Landmark-based and detector-based approaches

Shin et al. [8] classify distance and angle features of MediaPipe hand-pose landmarks with SVM and LightGBM; Saini et al. [9] combine 21 landmarks with a Random Forest on a small self-collected set; Bhadouria et al. [10] feed landmarks to an LSTM for a three-word task; and Bhuiyan et al. [11] use YOLOv8 for phrase-level gestures. These pipelines avoid heavy image models but depend on landmark extraction and, in several cases, on very small datasets. Najib [12] pools eleven sign-language datasets and reports that pooling heterogeneous data can reduce accuracy.

C. Knowledge Distillation for compact sign-language models

Lamaakal et al. [13] propose TinyMSLR, a ConvNeXt-Tiny and Swin Transformer model with an adaptive fusion gate, compressed into a 2.7 M-parameter student by dual-teacher KD. On a shared 20-class subset of RWTH-PHOENIX-Weather 2014T (German Sign Language) and Chinese Sign Language, it reports 99.01% validation and 98.94% test accuracy, a 7.2 MB model and 13.5 ms latency on an edge GPU. This is the main evidence that KD gives compact sign-language models, but it addresses other sign languages and uses a more complex dual-teacher design than the single-teacher setting planned as future work. The techniques on which the wider project rests are ResNet [16], KD [14], MobileNetV2 [15] and the MediaPipe framework [17]; only ResNet is used in this paper.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

Table 1
Comparison of the Reviewed Studies (Results as Reported by Their Authors)

Ref.	Method / task	Data (as reported)	Reported result	Limitation relevant here
[1]†	RetinaNet, ResNet-50 backbone; real-time ASL alphabet	In-the-wild data; 30-participant study	Average accuracy above 93%	Heavy backbone; computational cost left for optimisation
[2]	Transfer learning, five backbones, 5-fold CV	ASL Alphabet, 87,000 images, 29 classes	ResNet-50 99.98%; EfficientNet 99.95%; ConvNeXt 99.51%; AlexNet 99.50%; ViT 88.59%	No KD or compression; real-time left as future work
[3]†	Three-convolution-layer CNN for ASL alphabet	ASL Alphabet, 87,000 images, 29 classes	Training 93.67%; validation 96.85%; test 97%	Static only; no live-video or lighting tests; no efficiency analysis
[4]†	Compact CNN on 32×32 images	ASL Alphabet, 87,000 images	Validation 95.48%; test 99.77%; confuses A and M	Raw-pixel features; errors on live images
[5]†	Three-conv-layer CNN, 64×64, augmentation	Kaggle fingerspelling, 26,331 images, 26 classes	Test 99.97%	Static images only; no real-time test
[6]	Hybrid Transformer-CNN	ASL Alphabet dataset	99.97%; 110 FPS; 5.0 GFLOPs	No knowledge distillation reported
[7]†	Eight-layer CNN with augmentation	Sign Language MNIST, 24 classes (no J, Z)	100% (proposed CNN)	Grayscale 28×28; excludes J and Z
[8]	MediaPipe hand-pose features, SVM / LightGBM	ASL Alphabet, Massey, Finger Spelling A	Reported to outperform earlier studies	Depends on landmark extraction; no CNN or KD
[9]†	MediaPipe landmarks, Random Forest	Self-collected, about 100 images per class, 33 classes	Random Forest 100%	Very small dataset; static only
[10]†	MediaPipe landmarks, LSTM; NLP text-to-sign	Self-recorded videos	100% on a 3-word task	Tiny vocabulary; scalability not evaluated
[11]†	YOLOv8 / YOLOv5; CNN with MediaPipe	Self-created, 22 phrase-level classes	YOLOv8 precision 95.6%; CNN 94.94%	Phrase level; small dataset
[12]†	Multilingual model, ML.NET classifier selection	11 Kaggle sign-language datasets	Combined test 92.33%; separate average 92.54%	Pooling degraded accuracy; static gestures
[13]	ConvNeXt-Tiny + Swin, adaptive fusion, dual-teacher KD	RWTH-PHOENIX-Weather 2014T (DGS) + CSL, 20 classes	Validation 99.01%; test 98.94%; 2.7 M parameters; 7.2 MB; 13.5 ms (edge GPU)	Other sign languages; not the ASL alphabet

† Reported details for these entries are taken from the group's Stage-I literature-survey table and have not been checked against the full texts.

6. OBJECTIVES OF THE STUDY

1. To develop a reproducible, duplicate-aware dataset preparation pipeline (exact and near-duplicate detection, group-aware splitting and external-test isolation) that limits data leakage between the training, validation and test sets.
2. To train a 29-class ResNet50 teacher for 10 epochs and record its training and validation results.
3. To evaluate the teacher on the 13,050-image internal test split using accuracy, macro precision, recall, F1-score and a confusion matrix analysis.
4. To state the limits of internal-split testing honestly and to report separately the evaluation of a second checkpoint (teacher_best_aug.pth).

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

7. RESEARCH METHODOLOGY

Component	Detail
Research approach and design	An experimental, offline pipeline: duplicate-aware data preparation, ResNet50 teacher training, checkpoint selection on the validation split and evaluation on the internal test split (Fig 1). It is the first stage of a teacher–student plan.
Dataset	The public Kaggle ASL Alphabet dataset [18] is used as it is; the images were not collected by the authors. It contains 87,000 RGB images of 200×200 pixels in 29 classes (A–Z, del, nothing, space) with 3,000 images per class. Labels are encoded as A = 0, …, Z = 25, del = 26, nothing = 27, space = 28. The dataset also has an official external test folder of 28 images (without a del image). The signers, camera, background and lighting of the dataset are not documented in our records and are not described here.
Data preprocessing	Every image is resized from 200×200 to 224×224 (bilinear interpolation, torchvision default), converted to a tensor with values in 0–1 and normalised with the ImageNet mean (0.485, 0.456, 0.406) and standard deviation (0.229, 0.224, 0.225). The repository's preprocessing module also provides training-only augmentation; whether it was used for the teacher run is not reported here.
Duplicate detection and splitting	The repository pipeline inspects and validates the images, detects exact duplicates with SHA-256 and near-duplicates with perceptual difference hashing (dHash), builds anchor-based duplicate groups and splits the data by group, so that duplicate groups stay inside one split. Exact copies of the 28 official external test images were excluded before grouping, leaving 86,972 images (Table 2). Duplicate groups cover 84,550 of the images. Manifests and reports are generated reproducibly, and automated tests cover the preprocessing. Two limitations are stated by the repository: near-duplicate external images may still exist in the internal data, and anchor-based grouping cannot guarantee that every visually similar neighbouring frame falls in the same split. The random seed of the group-aware split is not reported here. An earlier Stage-I split was stratified 80/10/10 with seed 42 (69,600 / 8,700 / 8,700 images, 2,400 / 300 / 300 per class); it did not control for duplicates.
Teacher architecture	The teacher is ResNet50 [16], a 50-layer deep residual network whose skip connections let gradients pass easily through the network. It takes 224×224 RGB images, and its final fully connected layer has 29 outputs, one for each class. The softmax of these outputs gives the class probabilities, and the class with the highest probability is the prediction.
Training procedure	The teacher was trained with the cross-entropy loss on the training split for 10 epochs with 224×224 input images on Google Colab. The validation split was used to select the best checkpoint, and the internal test split was not used for training or checkpoint selection. The best checkpoint (epoch 10, the last epoch of the run) was saved and loaded back successfully. The initialisation (for example, ImageNet weights), optimiser, learning rate and batch size of this run are not reported in this paper.
Environment	Google Colab workflow. The accelerator model and runtime details are not reported because they are not recorded consistently in the training logs. The software stack is Python, PyTorch/torchvision, NumPy, scikit-learn and Matplotlib, with Git/GitHub; the repository requires Python 3.11, and exact library versions are not recorded in this paper.
Evaluation protocol and metrics	The teacher was evaluated on the 13,050-image internal test split, which was not used for training or checkpoint selection. The split is class-balanced (450 images per class), so macro averages are reported together with accuracy. Per-class precision, recall and F1-score, macro and weighted averages, and a 29×29 confusion matrix are computed (Equations 1–3). Parameters, model size and CPU latency are reported only for the separately evaluated checkpoint (Table 6).

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

For every class c the following are computed:

$$\text{Accuracy} = N_{correct} / N \quad (1)$$

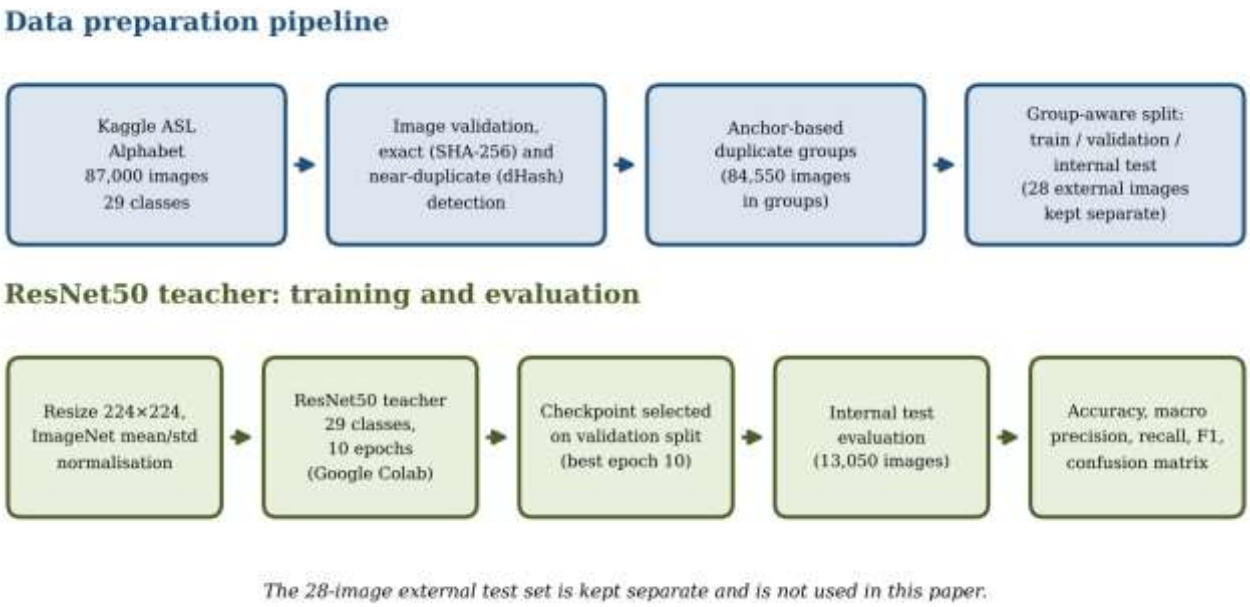
$$P_c = TP_c / (TP_c + FP_c), \quad R_c = TP_c / (TP_c + FN_c) \quad (2)$$

$$F1_c = 2\, P_c\, R_c / (P_c + R_c) \quad (3)$$

The macro precision, recall and F1-score are the unweighted means of the per-class values, and the weighted F1-score weights each class by its number of test images.

The system is an offline pipeline (Fig 1). The data preparation stage validates the images, detects exact and near-duplicates, builds duplicate groups and splits the data by group into training, validation and internal test sets, with the 28 official external test images kept separate. The preprocessing stage resizes and normalises the images. The teacher is then trained on the training split, its checkpoint is selected on the validation split, and it is evaluated on the internal test split. The pipeline produces the trained ResNet50 teacher, which is the starting point of the planned student and KD stages. The software is organised in modules with one function each, covering dataset preparation and splitting, preprocessing and evaluation.

Fig 1
Data preparation and ResNet50 teacher training and evaluation pipeline



The 28-image external test set is kept separate and is not used in this paper.

8. ANALYSIS OF DATA

This section reports the results of the ResNet50 teacher: the dataset splits (Table 2), training and validation results (Table 3), internal test results (Table 4), the confusion matrix analysis (Table 5) and a separate checkpoint evaluation (Table 6). They are reported separately because they come from different splits or checkpoints and must not be mixed.

Table 2
Dataset Splits

Split	Images	Share	Method
Training	60,872	70%	Duplicate-aware group split (repository pipeline)
Validation	13,050	15%	Duplicate-aware group split
Internal test	13,050	15%	Duplicate-aware group split
Total	86,972	100%	87,000 minus 28 exact copies of external test images
External test	28	–	Kept separate; smoke test only, not a benchmark

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

Split	Images	Share	Method
Stage-I split (earlier)	69,600 / 8,700 / 8,700	80/10/10	Stratified, seed 42; no duplicate control

Source: Project repository pipeline

Both splits are reported for transparency. The teacher was trained on the training split and its checkpoint was selected on the validation split; the internal test split was used for its test evaluation. The external test set is kept separate and is listed in Future Research.

Table 3
ResNet50 Teacher: Training and Validation Results (Validation Split, Not Test Results)

Item	Value
Model	ResNet50 (teacher), 29 output classes
Dataset	29-class ASL Alphabet dataset
Input image size	224 × 224
Training epochs	10
Best saved epoch	10
Best validation loss	0.00048092
Best validation accuracy	99.9923%
Training accuracy at epoch 10	99.9113%
Checkpoint	Saved and loaded successfully; size approximately 270 MB
Training platform	Google Colab

Source: Training log of the Google Colab run

The 29-class ResNet50 teacher was trained on all classes, and its best checkpoint was saved and can be loaded again, so it can serve as the teacher in the planned distillation. The validation accuracy of 99.9923% is a validation result. The validation split was also used to select the checkpoint, so the value is not an independent estimate of performance on unseen data, and near-duplicate frames in the dataset may raise it further. The validation accuracy is slightly higher than the training accuracy at epoch 10; the logs used here do not explain this difference, so no reason is claimed. The saved checkpoint is about 270 MB; this size is not compared with the 90.22 MB of the separately evaluated checkpoint in Table 6, because the two files come from different checkpoints and the content of the 270 MB file (for example, whether it holds optimiser state) is not described here.

Table 4
ResNet50 Teacher: Internal Test Results (13,050 Images, 29 Classes)

Metric	Value
Test images	13,050
Number of classes	29
Test accuracy	99.984674%
Macro precision	99.984708%
Macro recall	99.984674%
Macro F1-score	99.984666%
Weighted F1-score	99.984666%

Source: Metrics file of the internal test evaluation

The teacher was evaluated on the 13,050-image internal test split, which contains 450 images for each of the 29 classes and is separate from the training and validation images. These are internal test results; they are not validation results and not external test results.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

Table 5
Teacher Confusion Matrix Summary (Internal Test Split)

Class	Test images	Correct	Misclassified
28 classes (all except del)	450 each (12,600)	450 each	0
del	450	448	2 (1 predicted as C, 1 predicted as Q)
Total	13,050	13,048	2

Source: Confusion matrix of the internal test evaluation

The confusion matrix contains 13,048 correct and 2 incorrect predictions out of 13,050 images, so the accuracy is 13,048/13,050 = 99.984674%, which agrees with Table 4. All 28 classes other than del were classified correctly, 450 out of 450 each. Both errors came from class del: 448 of its 450 images were correct, one image was predicted as C and one as Q. With only two errors, no pattern can be established, and no reason for the errors is claimed here; a larger or different test set would be needed to show whether the confusion of del is systematic.

These values must not be read as real-world ASL recognition accuracy. Duplicate-aware preprocessing was used: duplicate groups cover 84,550 images, the split is group-aware to reduce near-duplicate leakage, and the 28 official external test images were kept out of the internal data. Near-duplicate leakage cannot be excluded completely, and the test images come from the same dataset as the training images. The official external test set has only 28 images, and the external evaluation of the teacher is planned as future work, so external validation is still required.

Table 6
Separate Checkpoint Evaluation (teacher_best_aug.pth, 8,700 Test Images; Not Merged with Table 4)

Metric	Value
Test images	8,700
Test accuracy	94.4253%
Macro precision	94.6457%
Macro recall	94.4253%
Macro F1-score	94.4439%
Parameters	23,567,453 (23.57 M)
Model size	90.22 MB
CPU latency (batch size 1)	1543.86 ms/image

Source: Separate evaluation of teacher_best_aug.pth

A separate evaluation was carried out on a different checkpoint, teacher_best_aug.pth, with 8,700 test images. This checkpoint and test set differ from those of Tables 3 and 4; the results are not merged with them, and the two checkpoints are not claimed to be identical. The accuracy of 94.4253% is lower than the 99.9847% of Table 4, but the checkpoints and test sets differ, so the difference is not attributed to any cause here, and results should always be quoted together with the checkpoint and test set used. The CPU latency of 1543.86 ms/image was measured with batch size 1; its measurement conditions are not recorded in this paper.

9. FINDINGS OF THE STUDY

1. **Teacher training:** The 29-class ResNet50 teacher was trained for 10 epochs; the best epoch was epoch 10, with a validation loss of 0.00048092, a validation accuracy of 99.9923% and a training accuracy of 99.9113%. The checkpoint (about 270 MB) was saved and loaded back successfully. These are validation results and not an independent estimate.

2. **Internal test performance:** On the 13,050-image internal test split the teacher reached 99.984674% accuracy, with macro precision of 99.984708%, macro recall of 99.984674% and macro F1-score of 99.984666%.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

3. **Errors in class del:** Of 13,050 test images, 13,048 were correct and 2 were incorrect. All 28 other classes were classified 450 out of 450; both errors came from class del (one predicted as C, one as Q). With only two errors, no pattern is established and no cause is claimed.

4. **Separate checkpoint:** The checkpoint teacher_best_aug.pth reached 94.4253% accuracy on 8,700 test images, with 23.57 M parameters, a model size of 90.22 MB and a CPU latency of 1543.86 ms/image at batch size 1. It is reported separately, and the difference from the internal test result is not attributed to any cause.

5. **Limits of the evidence:** The results come from a single run on the internal split of one dataset. Near-duplicate leakage cannot be excluded completely, and the 28-image external test set has not yet been evaluated, so the results are not a measure of real-world accuracy.

10. SUGGESTIONS AND RECOMMENDATIONS

1. **Report the full training configuration:** The initialisation, optimiser, learning rate, batch size, split seed, accelerator and exact library versions should be recorded and reported in later stages so that the teacher run can be reproduced.

2. **Evaluate beyond the internal split:** The teacher should be evaluated on the official external test set and on a larger or different test set, including webcam images, before any claim about real-world accuracy is made.

3. **Quote results with their checkpoint and test set:** Every accuracy value should be quoted together with the checkpoint and test split it came from, as the two checkpoints in this paper give different results on different test sets.

4. **Keep duplicate-aware splitting and per-class reporting:** Group-aware splitting should be kept in the later stages, and per-class results should be reported alongside overall accuracy.

5. **Use the teacher for the student and KD stages:** The trained ResNet50 teacher can serve as the frozen teacher for the MobileNetV2 student, and the teacher and student should then be compared on the same data for accuracy, parameter count, model size and latency, with the measurement conditions recorded.

11. CONCLUSION

This paper presented the completed first stage of a teacher–student project on ASL alphabet recognition: dataset preparation, preprocessing, and the training and evaluation of a 29-class ResNet50 teacher. The Kaggle ASL Alphabet dataset was prepared with duplicate-aware preprocessing and divided by group into training, validation and internal test splits, with the 28 official external test images kept separate. The teacher was trained on Google Colab for 10 epochs. Its best checkpoint (epoch 10) reached a validation loss of 0.00048 and a validation accuracy of 99.9923%, with a training accuracy of 99.9113% at epoch 10; these are training and validation results. On the 13,050-image internal test split the teacher reached 99.9847% accuracy and a macro F1-score of 99.9847%, with 13,048 correct and two incorrect predictions, both from class del. This is an internal-split result and not a measure of real-world accuracy. A separately evaluated checkpoint (teacher_best_aug.pth) reached 94.4253% on 8,700 test images and is reported separately. The MobileNetV2 student, Knowledge Distillation and the other pending work are described in Future Research.

12. FUTURE RESEARCH

The following work is planned and has not been carried out; none of it is claimed as a result of this paper:

1. Evaluation of the trained ResNet50 teacher on the official 28-image external test dataset.
2. Development and training of the MobileNetV2 student model.
3. Implementation of Knowledge Distillation using the trained ResNet50 teacher.
4. Comparison of the teacher and student performance on the same data.
5. Comparison of the model size, parameter count and inference efficiency of the future models.
6. Real-time webcam-based validation and performance testing.
7. Robustness testing under different lighting conditions, hand orientations and backgrounds.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

8. Further optimisation of the models for lightweight deployment.

References

1. N. Virag and J. Katona, "AI-powered real-time hand sign recognition: enhancing communication for the hearing- and speech-impaired community," Universal Access in the Information Society, vol. 25, art. no. 73, 2026, doi: 10.1007/s10209-026-01316-9.

2. B. Alsharif, A. S. Altaher, A. Altaher, M. Ilyas and E. Alalwany, "Deep learning technology to recognize American Sign Language alphabet," Sensors, vol. 23, no. 18, art. 7970, 2023, doi: 10.3390/s23187970.

3. J. J. Pangaribuan, R. Romindo, A. R. Mitra, O. P. Barus and S. Prashanth, "American Sign Language alphabet recognition using convolutional neural network," in Proc. 2nd Int. Conf. Technology Innovation and Its Applications (ICTIIA), 2024, pp. 1–6, doi: 10.1109/ICTIIA61827.2024.10761663.

4. A. Abdullah, N. Ali, R. H. Ali, Z. U. Abideen, A. Z. Ijaz and A. Bais, "American Sign Language character recognition using convolutional neural networks," in Proc. IEEE Canadian Conf. Electrical and Computer Engineering (CCECE), Regina, SK, Canada, 2023, pp. 165–169, doi: 10.1109/CCECE58730.2023.10288799.

5. F. J. Al-Ghanim, S. S. Baawi and N. R. Hamza, "Developing convolutional neural networks for recognition of American Sign Language," Muthanna Journal of Pure Science (MJPS), Jun. 2025.

6. M. Aly and I. S. Fathi, "Recognizing American Sign Language gestures efficiently and accurately using a hybrid transformer model," Scientific Reports, vol. 15, 2025, doi: 10.1038/s41598-025-06344-8.

7. D. Bala, M. A. Hossain, M. A. Islam, M. Mynuddin, M. S. Hossain and M. I. Abdullah, "Effective recognition system of American Sign Language alphabets using machine learning classifiers, ANN and CNN," in Proc. IEEE Conf. Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), 2022, doi: 10.1109/IATMSI56455.2022.10119336.

8. J. Shin, A. Matsuoka, M. A. M. Hasan and A. Y. Srizon, "American Sign Language alphabet recognition by extracting feature from hand pose estimation," Sensors, vol. 21, no. 17, art. 5856, 2021, doi: 10.3390/s21175856.

9. Saini et al., "Real-time sign language recognition and translation using MediaPipe and random forests for inclusive communication," IEEE CICTN, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10932602

10. Bhadouria et al., "LSTM-based recognition of sign language," ACM IC3, 2024.

11. H. J. Bhuiyan, M. F. Mozumder, M. R. I. Khan, M. S. Ahmed and N. Z. Nahim, "Enhancing bidirectional sign language communication: integrating YOLOv8 and NLP for real-time gesture recognition & translation," arXiv:2411.13597, 2024.

12. F. M. Najib, "A multi-lingual sign language recognition system using machine learning," Multimedia Tools and Applications, vol. 84, no. 24, pp. 27987–28011, 2025, doi: 10.1007/s11042-024-20165-3.

13. I. Lamaakal, C. Yahyati, Y. Maleh, K. El Makkaoui and I. Ouahbi, "An explainable hybrid CNN–transformer model for sign language recognition on edge devices using adaptive fusion and knowledge distillation," Scientific Reports, vol. 16, art. 7143, 2026, doi: 10.1038/s41598-026-38478-8.

14. G. Hinton, O. Vinyals and J. Dean, "Distilling the knowledge in a neural network," arXiv:1503.02531, 2015.

15. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

16. K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

17. C. Lugaresi et al., "MediaPipe: a framework for building perception pipelines," arXiv:1906.08172, 2019.

18. "ASL Alphabet," image dataset, Kaggle (user: grassknoted), 87,000 training images, 29 classes. [Online]. Available: https://www.kaggle.com/datasets/grassknoted/asl-alphabet

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.