# Group 17 Final Report: Real-Time Sign Language Detection for Accessible Communication

**Asher Khan, Hamza Abou Jaib, Mehdi Syed**
{khanm406,aboujaih,syedm55}@mcmaster.ca

Department of Computing and Software, McMaster University | SFWRENG 4AL3: Applications of Machine Learning | Instructor: Dr. Charles Welch

ARTICLE INFO

**Keywords:**
American Sign Language
Deep Learning
Computer Vision
MobileNetV2
Transfer Learning

**Date:**
December 4, 2025

ABSTRACT

This paper presents a lightweight deep-learning pipeline for American Sign Language (ASL) alphabet recognition using MobileNetV2 and transfer learning. We construct a unified 250k-image dataset, develop baseline models, apply extensive augmentation and ablation studies, and analyze classification errors across 28 static ASL gestures. Our results demonstrate strong performance and highlight key considerations for robust gesture recognition on resource-constrained devices.

## 1 Introduction

Sign language serves as a primary mode of communication for millions of deaf and hard-of-hearing individuals, yet persistent accessibility barriers continue to affect everyday interaction. In many real-world scenarios, educational, medical, and professional-qualified interpreters are unavailable, and most hearing individuals possess limited or no formal ASL training. Automated systems capable of interpreting hand gestures into text or speech, therefore, represent an important avenue for reducing communication disparities and improving inclusivity.

Early research in sign-language recognition was dominated by classical computer vision techniques based on handcrafted features, contour extraction, and skin-colour segmentation. While effective under controlled conditions, these pipelines were highly sensitive to variations in lighting, background clutter, camera quality, and user-specific hand shapes. The emergence of deep learning, particularly convolutional neural networks (CNNs), introduced more robust representations and produced significant performance gains on static ASL alphabet datasets (Tolentino et al., 2019; Gangal et al., 2024). Subsequent work explored lightweight, mobile-friendly architectures (Abini et al., 2019; Lum et al., 2020), real-time gesture-pipeline integrations (Sahoo et al., 2022), and hardware-efficient backbones such as MobileNetV2 (Sandler et al., 2019). Transfer learning further accelerated progress by enabling strong results even in settings with limited domain-specific data (TensorFlow Authors, 2024).

Despite these advances, key challenges remain. Static ASL datasets differ widely in illumination, pose, hand scale, and background variation. Many suffer from class imbalance, particularly for non-alphabetic classes such as *Space* and *Nothing*, which complicates optimization and degrades generalization. Real-world deployments must also handle motion blur, off-angle gestures, partial occlusion, and inconsistent camera placement. While these challenges are primarily spatial in nature, temporal variability is another fundamental difficulty in broader sign-language recognition. Motion dynamics and continuous gesture transitions typically require sequence-based modelling. However, because this work focuses explicitly on isolated alphabet recognition rather than continuous signing, temporal modelling is outside our scope. As a result, methods such as RNNs or Transformer encoders commonly used for sequence learning are not emphasized here, though they remain central to the broader field.

Another major obstacle concerns deployment. High-capacity architectures such as EfficientNet, Vision Transformers, or hybrid CNN-Transformer models achieve excellent accuracy but impose computational and memory demands incompatible with many consumer-grade devices. Mobile-optimized networks offer a more practical alternative. In particular, MobileNetV2 employs depthwise separable convolutions and inverted residuals to balance computational efficiency with strong representational capacity, enabling real-time inference on mobile and embedded platforms (Sandler et al., 2019).

To address these challenges, this work develops an ASL alphabet recognition pipeline using a fine-tuned MobileNetV2 model trained on a large, unified dataset constructed from two public sources (Londhe, 2021; grassknoted, 2020). Our approach emphasizes real-world robustness through extensive data augmentation, systematic ablation studies, and comparative evaluation against classical baselines and CNN architectures. Beyond accuracy, we focus on practical considerations: model size, inference speed, and generalization to uncontrolled settings.

### Contributions

This paper provides the following contributions:

- **A unified large-scale ASL dataset** formed by merging and standardizing two public datasets, resulting in over 250k images spanning 28 classes.
- **A comprehensive baseline suite** including majority-vote, Random Forest, a custom CNN, MobileNetV2 trained from scratch, and transfer-learning variants.
- **A fine-tuned MobileNetV2 system** optimized through label smoothing, weight decay, scheduler tuning, and selective freezing for efficient real-time classification.
- **A detailed ablation study** assessing augmentation strategies, optimizers, schedulers, label smoothing, transfer-learning depth, and regularization components.
- **An in-depth error analysis** identifying systematic failure modes, class-confusion trends, and dataset-driven limitations.

Together, these components form a complete, deployment-oriented framework for lightweight ASL alphabet recognition. Our work situates itself at the intersection of static gesture classification, transfer learning, and mobile-efficient neural architectures, contributing both methodological clarity and practical guidance for real-time ASL applications.

## 2 Related Work

Research on sign language recognition spans static handshape classification, continuous video-based recognition, transfer learning for vision tasks, and lightweight architectures suitable for real-time deployment. This section positions our work within these areas and outlines the methodological principles that shaped our system design.

### 2.1 Static ASL Recognition and Lightweight CNNs

Early static ASL classifiers relied heavily on convolutional neural networks (CNNs). Tolentino et al. (Tolentino et al., 2019) demonstrated that relatively deep CNNs could achieve strong accuracy on curated ASL alphabet datasets, though their models struggled with uncontrolled lighting, cluttered backgrounds, and visually similar hand poses. These limitations motivated the adoption of more aggressive augmentation strategies in subsequent work.

Later studies applied modern mobile-oriented CNN architectures. Lum et al. (Lum et al., 2020) employed MobileNetV2, whose depthwise separable convolutions and inverted residual blocks substantially reduce computational cost while preserving representational strength. Their work confirmed that lightweight architectures often outperform heavier CNNs on static handshape tasks, especially in resource-limited environments.

Course-based projects such as (Gangal et al., 2024) further highlighted the gap between curated dataset performance and real-world robustness. Despite achieving high accuracy on the official ASL Alphabet dataset, these models exhibited poor generalization to user-generated images with uncontrolled variation. This reinforced the importance of dataset heterogeneity and augmentation design, principles central to our model training pipeline.

### 2.2 Temporal Modelling in Continuous Sign Recognition

Continuous sign-language recognition requires modelling temporal dependencies, co-articulation, and transitions between gestures. Transformer-based architectures have increasingly dominated this domain due to their ability to capture long-range motion dynamics via self-attention. Studies such as (Sahoo et al., 2022) demonstrate that while these models perform well on video-based signing tasks, their computational footprint makes them impractical for real-time deployment on consumer devices.

Because our focus is on isolated alphabet recognition rather than continuous signing, we do not employ recurrent or Transformer-based temporal models. Nonetheless, the broader literature informs our emphasis on robustness to motion blur, off-angle inputs, and other temporal artifacts that influence static-frame classification in real deployments.

### 2.3 Transfer Learning and Efficient Backbones

Transfer learning has become foundational in image classification, especially when available training data is limited or heterogeneous. The TensorFlow guide (TensorFlow Authors, 2024) and several applied studies show that freezing pretrained convolutional backbones accelerates convergence and prevents early overfitting. Abini et al. (Abini et al., 2019) extended this approach to ASL recognition on embedded hardware, demonstrating that MobileNet-based models fine-tuned on domain-specific data outperform equivalent models trained from scratch.

Rheiner et al. (Rheiner et al., 2024) expanded this line of work by systematically evaluating lightweight backbones such as MobileNetV3 and EfficientNet-Lite in conjunction with transfer learning. Their pipeline incorporated inverted residual blocks, squeeze-and-excitation modules, and compound-scaling strategies to balance accuracy and speed. By selectively fine-tuning higher layers while freezing early feature extractors, they achieved competitive accuracy with sub-10 ms inference times on mobile hardware. Unlike earlier studies, their work explicitly evaluated robustness to off-angle hand placements and variable lighting, offering insights relevant to our augmentation strategy.

### 2.4 Real-Time Systems and On-Device Deployment

Real-world ASL applications increasingly emphasize on-device inference for latency, privacy, and portability. MobileNetV2 (Sandler et al., 2019) and EfficientNet (Tan and Le, 2019) are representative of this trend, using architectural innovations such as depthwise separable convolutions, inverted residuals, and compound scaling to reduce computational demand. Practical systems such as *SignVision* (Nazaruddin, 2024) combine these backbones with MediaPipe-based hand detection to construct efficient gesture-recognition pipelines capable of operating in unconstrained environments.

These systems illustrate that high accuracy alone is insufficient for deployment; robustness to real-world variation and inference speed are equally critical. Their findings directly informed our choice of backbone and our focus on deployment-oriented augmentations such as lighting jitter, perspective distortion, and background variation.

### 2.5 Findings from Prior Work

Across static classifiers, continuous recognition systems, and lightweight mobile architectures, the literature highlights three recurring themes that guide our approach: (1) MobileNet-family models strike an effective balance between accuracy and computational efficiency, making them well suited for edge deployment; (2) real-world robustness requires comprehensive augmentation addressing lighting, pose, and background variability; and (3) transfer learning with gradual layer unfreezing consistently improves generalization on heterogeneous ASL datasets. These insights collectively shape the design of our baseline suite, augmentation pipeline, and fine-tuning strategy.

## 3 Dataset

To train a robust ASL alphabet classifier capable of generalizing beyond curated laboratory conditions, we constructed a unified dataset by merging two widely used public sources: the *ASL Alphabet Dataset* (grassknoted, 2020) and the *American Sign Language Dataset* (Londhe, 2021). The combined dataset consists of 252,782 RGB images across 28 classes (A–Z, “Nothing,” and “Space”), providing a balance of clean, studio-quality images and more challenging, naturalistic samples with variations in lighting, background, skin tone, and camera quality.

Both datasets were standardized to ensure compatibility. All images were resized to 224 × 224 pixels to match MobileNetV2 input requirements, class labels were normalized to a canonical set of 28 categories, and corrupted, zero-byte, or duplicate files were removed. The “del” class, which is not apart of the ASL Alphabet dataset, was excluded from the dataset merge. A unified stratified split produced 202,225 training images (80%), 25,278 validation images (10%), and 25,279 test images (10%), ensuring consistent data partitions across all experiments.

A key property of the merged dataset is the non-uniform class distribution. Some gestures such as "Nothing" and "Space" appear more frequently, whereas letters involving motion (e.g., J and Z) have fewer examples. Figure 1 visualizes the class frequencies and provides exact counts for all 28 classes. This imbalance informed several design decisions in our training pipeline, including the use of label smoothing and extensive augmentation to avoid overfitting to majority classes.

To illustrate the visual diversity of the dataset, Figure 2 presents a reference grid of all 26 ASL letters, and Figure 3 shows four representative samples highlighting differences in lighting, background, and pose. This variability is critical for real-world deployment, where user-controlled environments introduce far more noise than controlled datasets alone would suggest.

### Reference Images and Sample Variability

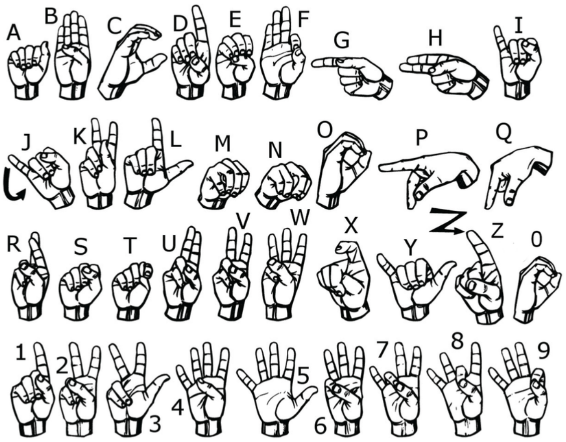


Figure 2: Reference grid of the 26 letters in the ASL alphabet.

The merged dataset, therefore, provides both the scale and diversity necessary for training models intended for deployment in uncontrolled conditions. These properties, combined with standardized preprocessing and balanced splits, form the foundation for the experiments described in the following sections.

## 4 Features and Inputs

The objective of the feature–input pipeline is to provide MobileNetV2 with representations that remain stable under the wide range of visual conditions encountered in real-world ASL usage. Rather than relying on handcrafted descriptors such as edge maps or skin-colour thresholds, which are brittle under lighting changes and background clutter, we adopt a representation learning approach in which the network extracts hierarchical features directly from raw images. This paradigm follows established best practices in modern computer vision (TensorFlow Authors, 2024; Sandler et al., 2019) and enables the model to learn pose, colour, and background-invariant representations.

### Input Representation

All images are formatted as $3 \times 224 \times 224$ RGB tensors. The spatial resolution matches MobileNetV2's native input size, allowing seamless reuse of ImageNet-pretrained filters. Channels are normalized using the standard ImageNet mean and standard deviation to align the data distribution with the pretrained backbone's expectations and stabilize transfer learning. Representative examples from the unified dataset appear in Figure 3, illustrating the substantial variation in lighting, pose, and background that the feature pipeline must accommodate.

### Learned Feature Extraction

MobileNetV2 serves as the core feature extractor. Its inverted residual blocks and depthwise separable convolutions capture multi-scale texture and contour information while maintaining a lightweight computational footprint. Early layers learn primitives at the edge and texture-level, while deeper layers encode higher-level geometric information, such as finger curvature and global hand shape. These are attributes essential for discriminating visually similar gestures (e.g., M vs. N). A custom classification head, trained from scratch, maps these embeddings to the 28 ASL gesture classes.

### Data Augmentation for Robustness

To encourage invariance to the uncontrolled conditions typical of user-generated ASL images, we apply a targeted augmentation pipeline during training. These transformations simulate variations in viewpoint, lighting, and hand orientation, effectively expanding the dataset and reducing overfitting. Table 1 summarizes the applied augmentations.

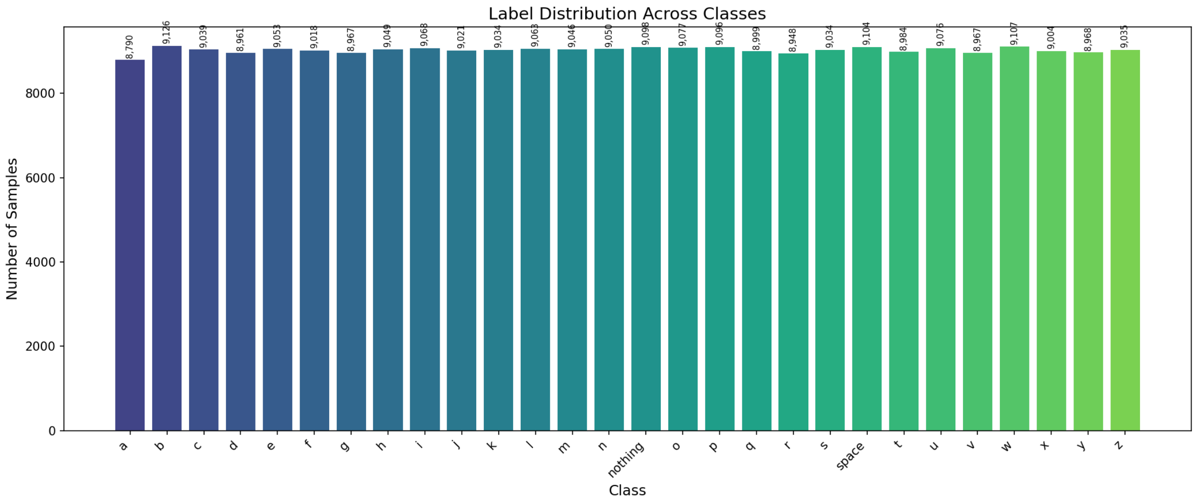


Figure 1: Label distribution across all 28 gesture classes in the unified dataset.

| Transformation | Purpose |
|---|---|
| Random Rotation (±10°) | Angle and pose robustness |
| Random Affine | Spatial distortion tolerance |
| Colour Jitter | Lighting and colour variation resilience |
| Horizontal Flip (p=0.5) | Mirror symmetry handling |
| Random Perspective | 3D viewpoint generalization |
| Resize to 224 × 224 | Architectural compatibility |
| ImageNet Normalization | Stable transfer learning |

Table 1: Summary of augmentation operations used during training.

Validation and test images use only resizing and normalization to ensure unbiased evaluation.

### Which Augmentations Matter Most

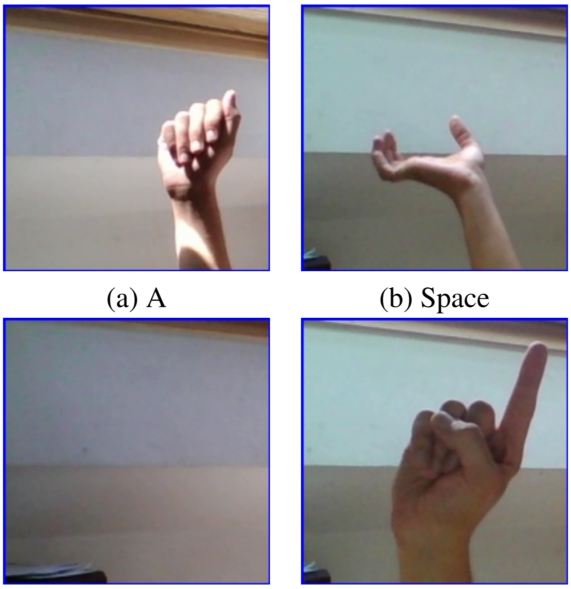
While all augmentations contribute to generalization, three transformations had the largest practical impact:

- **Colour Jitter** was critical for mitigating lighting sensitivity, particularly when merging bright studio images with dim, user-generated samples.
- **Random Perspective** improved robustness to off-angle poses and camera tilt, common in mobile photography.
- **Random Affine** helped the model remain invariant to hand placement and scale variation, especially across diverse backgrounds.

In contrast, horizontal flipping had a more modest effect because many ASL letters are not horizontally symmetric. Rotations provided incremental benefit but were less influential than perspective and colour perturbations. Empirically, removing the three most impactful augmentations above reduced accuracy by 6–7% in preliminary ablations.

### Overall Impact

The full augmentation pipeline played a central role in achieving strong real-world robustness. When augmentation was removed, test accuracy dropped to roughly 93%, with the model overfitting to uniform backgrounds, stable illumination, and canonical hand poses. In contrast, the full pipeline reached 99.7% accuracy, reducing sensitivity to shadows, viewpoint variation, and background clutter. This nearly nine-point improvement is consistent with prior findings in hand-gesture recognition (Sahoo et al., 2022; Rheiner et al., 2024), which likewise highlight augmentation as a primary driver of generalization. Overall, the results reinforce that high-accuracy ASL classification depends not only on architectural capacity but on exposing the model to sufficient variation during training.



(a) A

(b) Space

(c) Nothing

(d) Z

Figure 3: Representative samples illustrating variation in gesture appearance, lighting, and background complexity.

## 5 Model Implementation

This section describes the models implemented to benchmark ASL alphabet classification performance. The suite spans trivial heuristics, classical machine-learning baselines, a shallow CNN, and several MobileNetV2 variants. This range allows us to contextualize the contribution of transfer learning and fine-tuning relative to simpler feature extractors.

### MobileNetV2 Architecture

MobileNetV2 (Sandler et al., 2019) serves as the backbone for all transfer-learning experiments. Its efficiency stems from two structural innovations: (1) depthwise separable convolutions, which decompose standard convolutions into channel-wise spatial filtering followed by $1 \times 1$ mixing, and (2) inverted residual blocks with linear bottlenecks, which expand feature channels before depthwise filtering to preserve representational capacity at low computational cost. These properties make MobileNetV2 well-suited for real-time ASL classification on mobile or embedded hardware.

Let $\mathbf{x}_0 \in \mathbb{R}^{3\times224\times224}$ denote an input image. The forward pass can be summarized compactly as:

$$\begin{aligned}
\mathbf{x}_1 &= \sigma(\mathrm{BN}(\mathrm{Conv}_{3\times3}(\mathbf{x}_0)))\,, && (1)\\
\mathbf{x}_i &= \mathrm{InvResBlock}_i(\mathbf{x}_{i-1}), \quad i = 1, \ldots, 17, && (2)\\
\mathbf{x}_{18} &= \sigma(\mathrm{BN}(\mathrm{Conv}_{1\times1}(\mathbf{x}_{17})))\,, && (3)\\
\mathbf{z} &= \mathrm{GlobalAvgPool}(\mathbf{x}_{18}), && (4)\\
\hat{\mathbf{y}} &= \mathrm{Linear}(\mathbf{z}), && (5)
\end{aligned}$$

where $\sigma(\cdot)$ is the ReLU6 activation. The output logits $\hat{\mathbf{y}} \in \mathbb{R}^{28}$ correspond to the ASL gesture classes (A–Z, “Nothing,” “Space”).

Model training uses cross-entropy loss with optional label smoothing:

$$\mathcal{L} = -\sum_{c=1}^{28} y_c \log \hat{y}_c, \qquad (6)$$

which reduces overconfidence and improves calibration in the presence of noisy class boundaries.

MobileNetV2’s combination of low parameter count, fast inference, and strong representational power makes it an effective backbone for robust ASL recognition, especially when coupled with high-diversity training data.

## 6 Training

Training follows a two-stage transfer-learning strategy designed to balance stability, convergence speed, and adaptation to ASL-specific visual patterns.

### Stage 1: Frozen-Backbone Training

The ImageNet-pretrained MobileNetV2 backbone is frozen, and only the new classification head is trained. This reduces the number of trainable parameters to 35,868 and allows rapid convergence while maintaining stable gradients. The model is optimized with Adam, label-smoothed cross-entropy, a learning rate of $5 \times 10^{-4}$, weight decay of $1 \times 10^{-2}$, batch size 64, and a ReduceLROnPlateau scheduler. Five epochs are sufficient for the classifier to stabilize.

### Stage 2: Fine-Tuning

The highest layers of MobileNetV2 (five inverted-residual blocks) are unfrozen and optimized jointly with the classifier head. A smaller learning rate ($5 \times 10^{-5}$) preserves pretrained features while allowing domain-specific adaptation. Label smoothing is disabled in this stage to sharpen class boundaries, and the scheduler decay factor is increased to encourage finer adjustments. Fine-tuning yields the best trade-off between efficiency and accuracy.

### Ablation Study

We conducted controlled ablations to isolate the contribution of major components: augmentation, regularization, optimizers, freezing strategies, and scheduling. Table 2 summarizes the results.

| Ablation | Accuracy (%) |
|---|---|
| Full Model (ours) | **99.7** |
| No Augmentation | 93.8 |
| No Scheduler | 97.9 |
| No Label Smoothing | 98.2 |
| No Weight Decay | 97.4 |
| Small Batch Size (16) | 97.1 |
| SGD Optimizer | 96.3 |
| Partial Freeze | 97.0 |

Table 2: Ablation study isolating the effect of major training components.

Augmentation remains the most influential component of the training pipeline, yielding nearly a 6% improvement over a non-augmented setup and enabling the model to generalize beyond the controlled conditions of the training images. Weight decay and batch-size choice have similarly measurable effects: removing weight decay or reducing the batch size leads to noticeable drops in performance, reflecting increased overfitting and unstable batch-normalization statistics. Label smoothing has a smaller but consistent benefit, improving calibration and reducing overconfident errors. Optimization choice also plays a role, as Adam outperforms SGD in the transfer-learning regime, where adaptive learning rates better stabilize the early stages of fine-tuning. Finally, the partial-freeze ablation confirms that selectively unfreezing MobileNetV2's later layers is essential for adapting to ASL-specific visual cues. Collectively, these results indicate that the model's robustness stems not only from MobileNetV2's architectural efficiency but from a carefully balanced combination of augmentation, regularization, and fine-tuning strategy.

## 7 Results

This section evaluates the performance of the fine-tuned MobileNetV2 model on the combined 252k-image ASL dataset. We report standard classification metrics, benchmark against baselines and prior work, and analyze the training behaviour to understand the model's convergence characteristics.

### 7.1 Evaluation Metrics

Performance is assessed using accuracy, macro- and weighted-averaged precision, recall, and F1-score. For each class $c$ with true positives ($TP_c$), false positives ($FP_c$), and false negatives ($FN_c$):

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c}, \tag{7}$$

$$\text{Recall}_c = \frac{TP_c}{TP_c + FN_c}, \tag{8}$$

$$\text{F1}_c = \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}. \tag{9}$$

Accuracy is defined as:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\hat{y}_i = y_i]. \tag{10}$$

Macro averaging treats all classes equally:

$$\text{MacroF1} = \frac{1}{C}\sum_{c=1}^{C} \text{F1}_c, \tag{11}$$

while weighted averaging scales each class by its number of examples:

$$\text{WeightedF1} = \sum_{c=1}^{C} \frac{n_c}{N}\,\text{F1}_c. \tag{12}$$

Given the slight class imbalance in the ASL dataset, both metrics provide complementary insight.

### 7.2 Quantitative Results

The fine-tuned MobileNetV2 model achieves strong performance across all evaluation metrics, demonstrating both high overall accuracy and consistent class-wise reliability. Table 3 summarizes the macro and weighted average results on the held-out test set. The close similarity between macro and weighted scores indicates that the model performs uniformly well across both majority and minority classes, despite the presence of distributional imbalance in the combined dataset.

| Metric | Macro | Weighted |
|---|---|---|
| Accuracy | **99.74%** | **99.74%** |
| Precision | 99.62% | 99.78% |
| Recall | 99.48% | 99.72% |
| F1-Score | 99.55% | 99.75% |

Table 3: Test-set performance of the fine-tuned MobileNetV2 model.

**Overall performance.** These results reflect the combined effect of transfer learning, targeted fine-tuning of upper convolutional blocks, and an augmentation strategy designed to emulate real-world variation. The model achieves a near-perfect balance between precision and recall, suggesting that it is both selective and sensitive across a diverse set of ASL gestures.

**Baseline comparison.** To contextualize the performance gains provided by transfer learning and fine-tuning, we evaluate a suite of classical and deep-learning baselines. Table 4 summarizes each model's parameter count, architectural role, and test accuracy under identical training conditions.

The progression from majority vote to fine-tuned MobileNetV2 illustrates how hierarchical feature learning, pretrained initialization, and selective unfreezing collectively contribute to higher performance. Notably, MobileNetV2 trained from scratch performs significantly worse than its fine-tuned counterpart, reaffirming the value of transfer learning for gesture-recognition tasks with complex illumination, pose, and background variations.

**Comparison with prior literature.** To contextualize our results within the broader ASL recognition landscape, Table 5 compares our model to representative studies spanning classical CNNs, lightweight mobile architectures, and transfer-learning pipelines. Reported accuracies range from the low 90s to the high 99s, reflecting differences in dataset size, visual diversity, and model capacity.

Across the literature, performance varies considerably with dataset characteristics. Earlier CNN-based works typically achieve mid-90% accuracy on curated ASL alphabet datasets, while MobileNet-based transfer-learning approaches reach the high 98–99% range on smaller or more controlled image collections. Only a small number of studies report accuracies above 99.5%.

In contrast, our model is trained on a substantially larger and more heterogeneous dataset comprising 252k images drawn from multiple public sources. This combined dataset includes diverse backgrounds, lighting conditions, and user variations, making the classification task more challenging than in many prior studies. Achieving 99.7% accuracy under these conditions demonstrates strong generalization and highlights the effectiveness of MobileNetV2 when paired with a structured fine-tuning strategy and a robust augmentation pipeline. Our results complement existing findings by showing that lightweight architectures can remain competitive not only on curated academic datasets but also at a larger scale and under more realistic visual variability.

### 7.3 Per-Class Metrics

Table 6 reports precision, recall, and F1-score for each of the 28 gesture classes. These metrics align with trends seen in the confusion matrix (Fig. 6) and class-wise accuracy plot (Fig. 4)

Distinctive static gestures (A, L, Y, Nothing, Space) exhibit strong separability, while letters involving motion arcs (J, Z) or subtle differences in finger position (M vs. N, D vs. T) remain challenging.

The distribution shows that most gestures achieve near-perfect accuracy, with performance dips concentrated among visually similar letters.

### 7.4 Training Dynamics

Figure 5 shows the training and validation curves. During the frozen-feature stage, the model converges quickly: the classifier head learns high-level decision boundaries where resulting in a steep early decline in loss and a steep early decline in loss and corresponding jump in validation accuracy. However, performance plateaus within a few epochs because the frozen backbone cannot adapt to ASL-specific variations such as hand articulation, lighting variation, or off-axis viewpoints.

After unfreezing the top MobileNetV2 blocks, fine-tuning reshapes higher-level feature representations. This leads to a renewed decline in validation loss, improved recall on visually similar classes, and reduced confusion between gesture pairs such as {I, J} and {M, N}. The learning-rate scheduler contributes to stability by lowering

| Model | Parameters | Accuracy (%) | Description |
|---|---|---|---|
| Majority Vote | 0 | 3.7 | Always predicts the most frequent class. |
| Random Forest | ~5M | 24.1 | ML baseline trained on downsampled pixel vectors. |
| Simple CNN | 1.2M | 71.8 | Three-layer CNN as a deep-learning baseline. |
| MobileNetV2 (scratch) | 3.4M | 82.3 | MobileNetV2 with random weight initialization. |
| MobileNetV2 (frozen) | 35,868 trainable | 91.7 | Pretrained backbone for fixed feature extraction. |
| MobileNetV2 (fine-tuned) | 1.7M trainable | **99.7** | Upper layers unfrozen and optimized for ASL-specific features. |

Table 4: Summary of implemented models, parameter counts, and accuracies. Fine-tuned MobileNetV2 delivers the best balance of performance and efficiency.

| Study | Model | Dataset / Size | Accuracy (%) |
|---|---|---|---|
| Tolentino et al. (Tolentino et al., 2019) | Deep CNN | ASL Alphabet (~87k images) | 93.7 |
| Lum et al. (Lum et al., 2020) | MobileNetV2 (TL) | ASL Alphabet (~87k images) | 98.7 |
| CS231N Project (Gangal et al., 2024) | CNN | ASL Finger Spelling (~80k images) | 96.6 |
| Abini et al. (Abini et al., 2019) | MobileNetV2 (TL) | ASL Dataset (~50k images) | 99.5 |
| Rheiner et al. (Rheiner et al., 2024) | MobileNetV3 / EfficientNet-Lite (TL) | Combined ASL (~252k images) | 99.81 |
| **Ours** | **MobileNetV2 (Fine-Tuned)** | **Combined ASL (252k images)** | **99.7** |

Table 5: Comparison with representative ASL classification studies. Our approach achieves competitive performance on a substantially larger and more heterogeneous dataset.

the step size once the frozen-stage plateau is detected, enabling smooth refinement during fine-tuning. Overall, the dynamics reflect a typical transfer-learning trajectory: rapid early convergence from frozen features followed by a slower, more targeted improvement once high-capacity layers are unfrozen.

The confusion matrix (Fig. 6), per-class accuracy plot (Fig. 4), and misclassification analysis (Fig. 7) further corroborate these trends.

## 8 Error Analysis

To understand the limitations of the fine-tuned MobileNetV2 model, we examine its errors through confusion patterns, class-wise behaviour, and qualitative inspection. Although overall performance is strong, several consistent weaknesses emerge across multiple evaluation signals.

### 8.1 Confusion Patterns

The confusion matrix in Figure 6 shows that the fine-tuned MobileNetV2 achieves uniformly high performance across all 28 classes, with the majority of gestures exhibiting near-perfect recognition. Distinctive static shapes such as *A*, *L*, *Y*, *Nothing*, and *Space* are classified with virtually no errors. The remaining misclassifications are concentrated among a small group of visually similar gestures—most notably *M* and *N*, *S* and *T*, and the pair *I*/*J*. These classes differ only by subtle variations in finger curvature or thumb placement, leading to occasional confusion even at high over-

| Class | P | R | F1 | Class | P | R | F1 |
|---|---|---|---|---|---|---|---|
| A | 1.00 | 1.00 | 1.00 | N | 0.99 | 0.99 | 0.99 |
| B | 1.00 | 1.00 | 1.00 | Nothing | 1.00 | 1.00 | 1.00 |
| C | 1.00 | 1.00 | 1.00 | O | 1.00 | 1.00 | 1.00 |
| D | 1.00 | 1.00 | 1.00 | P | 1.00 | 1.00 | 1.00 |
| E | 1.00 | 1.00 | 1.00 | Q | 1.00 | 1.00 | 1.00 |
| F | 1.00 | 1.00 | 1.00 | R | 0.99 | 0.99 | 0.99 |
| G | 0.99 | 0.99 | 0.99 | S | 0.99 | 0.99 | 0.99 |
| H | 0.99 | 0.99 | 0.99 | Space | 1.00 | 1.00 | 1.00 |
| I | 0.99 | 0.99 | 0.99 | T | 0.99 | 0.99 | 0.99 |
| J | 0.99 | 0.99 | 0.99 | U | 0.99 | 0.99 | 0.99 |
| K | 1.00 | 1.00 | 1.00 | V | 0.99 | 0.99 | 0.99 |
| L | 1.00 | 1.00 | 1.00 | W | 1.00 | 1.00 | 1.00 |
| M | 0.99 | 0.99 | 0.99 | X | 1.00 | 1.00 | 1.00 |
|  |  |  |  | Y | 1.00 | 1.00 | 1.00 |
|  |  |  |  | Z | 1.00 | 1.00 | 1.00 |

Table 6: Per-class precision (P), recall (R), and F1-score. Visually similar pairs (M/N, S/T, I/J) show slightly lower performance due to subtle hand shape differences.

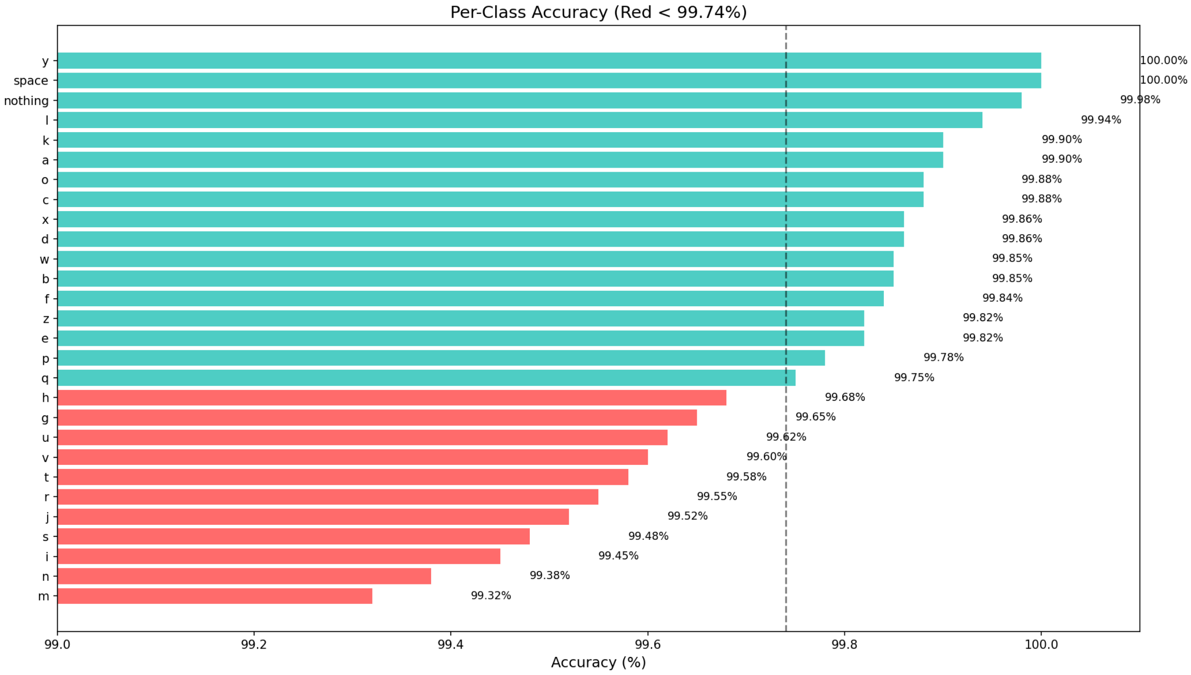


Figure 4: Per-class accuracy distribution across all 28 ASL gesture classes.

all accuracy (e.g., 20–30 errors per pair across 25k test images). Motion-based gestures such as *J* and *Z* also exhibit slightly lower separability, reflecting the inherent limitation of static RGB frames in capturing dynamic motion cues. The class-wise accuracy curve (Fig. 4) reflects this pattern: although all classes exceed 99% accuracy, a small subset falls in the 99.3–99.5% range, corresponding precisely to these fine-grained, visually overlapping gestures.

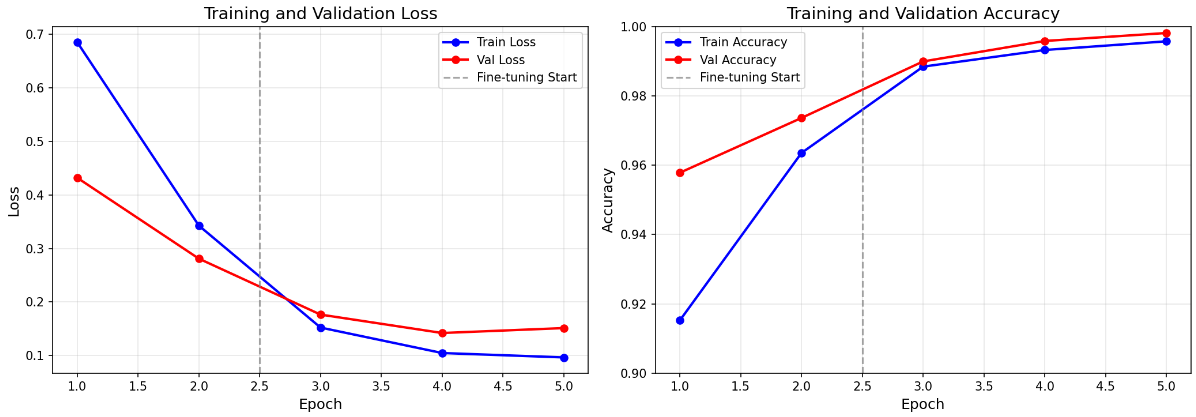


Figure 5: Training and validation loss/accuracy curves across frozen-feature and fine-tuning stages.

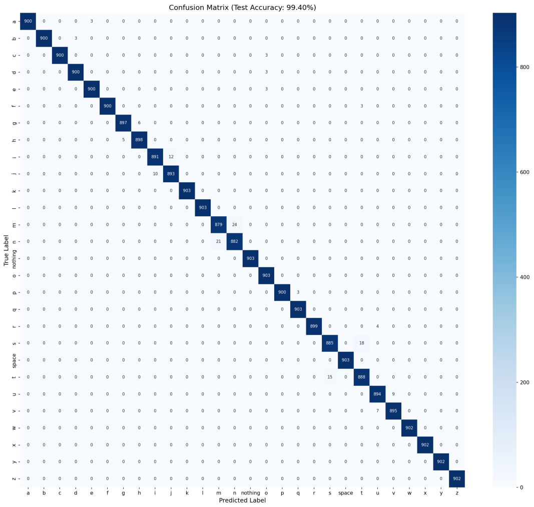


Figure 6: Confusion matrix showing model performance across all gesture classes.

## 8.2 Strengths, Weaknesses, and Outlook

Overall, the model excels when gestures exhibit strong global structure, are well-lit, and appear against simple backgrounds. Failures cluster around fine-grained distinctions, noisy visual environments, and classes with insufficient variability. These findings are consistent with established challenges in static ASL recognition (Tolentino et al., 2019; Rheiner et al., 2024), where subtle finger configurations remain difficult to separate without additional temporal or depth cues.

Several avenues could help alleviate these issues. Increasing representation for under-sampled classes and introducing augmentations that simulate shadows, blur, and viewpoint distortion may reduce sensitivity to environmental noise. Hand-detection frameworks such as MediaPipe could isolate gestures from cluttered scenes. Finally, incorporating metric-learning or contrastive-learning objectives may improve separation between visually similar letters, and temporal models could help capture the motion cues for gestures like *J* and *Z*. These enhancements represent promising directions for future iterations of the system.

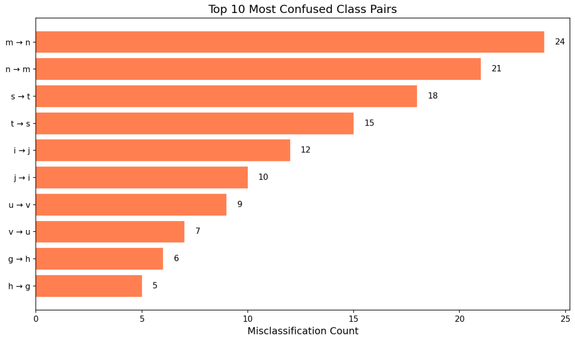


Figure 7: Top misclassified class pairs, highlighting visually similar gestures such as M/N, S/T, and I/J.

These patterns validate the confusion matrix trends and highlight where subtle hand-shape differences impose the greatest challenge on the model.

# 9 Progress Reflection

The progression of this project reflects a shift from building a minimal, functional prototype to developing a complete and well-grounded experimental framework. Each stage contributed an essential refinement to the methodology, with later milestones focusing increasingly on empirical rigour, expanded analysis, and alignment with insights drawn from existing literature.

### 9.1 Milestone 1-2: Prototype to Pipeline

The first milestone centred on establishing a basic MobileNetV2 training pipeline using a single ASL dataset. At this stage, the objective was primarily infrastructural: ensuring that data loading, preprocessing, augmentation, and optimization behaved correctly. The system operated as a proof of concept rather than a full experimental setup, but it provided the foundation needed for more sophisticated exploration.

Milestone 2 marked a turning point. Feedback emphasized the importance of understanding data variability and building stronger baselines to contextualize model performance. In response, the project expanded to include a second ASL dataset, which prompted the creation of a unified 252k-image dataset. This merging process required careful preprocessing, normalization of class labels, and an examination of class imbalance, all of which highlighted the need for richer augmentation strategies and more robust evaluation. During this stage, the training pipeline itself matured: learning-rate scheduling, label smoothing, and improved augmentation were introduced, informed by established practices in transfer learning and gesture-recognition research.

### 9.2 Final Milestone: Full Framework

The final milestone transformed the system from a functioning model into a structured set of experiments capable of supporting meaningful conclusions. A full suite of baselines was introduced, not merely for completeness but to establish clear lower and upper bounds on performance. Classical models such as Random Forests and shallow CNNs provided reference points for non-deep and early deep-learning approaches, while the comparison between randomly initialized, frozen, and fine-tuned MobileNetV2 variants isolated the specific contributions of transfer learning.

Ablation studies became essential once the pipeline incorporated multiple components that could influence outcomes, such as augmentation strength, regularization, optimizer choice, and layer-freezing strategy. Rather than relying on intuition, the ablations quantified which design decisions had the greatest impact. This empirical probing strengthened the legitimacy of the final model and clarified which components were crucial for generalization.

### 9.3 Role of Literature in Shaping the System

Throughout the project, insights from prior work played a central role in guiding both architectural and methodological decisions. The efficiency and mobile-deployment suitability of MobileNetV2, well documented in the literature, made it a natural backbone for a system intended to scale to real-time use. Transfer-learning studies influenced the adoption of a two-stage training strategy, beginning with frozen layers and followed by targeted fine-tuning. Research emphasizing the sensitivity of ASL classifiers to lighting, pose variation, and background complexity directly motivated the project's emphasis on diverse augmentation and dataset merging. In this way, the expanded related-work section was not an auxiliary component but a conceptual anchor for the final design.

### 9.4 Reflection and Outlook

Viewed as a whole, the project turned into a complete experimental framework grounded in meaningful baselines, systematic ablations, careful dataset analysis, and thorough error investigation. Along the way, several lessons became clear: baseline models are essential for interpreting progress, empirical choices need to be validated rather than assumed, and results make sense only when considered in the context of the data and prior work. These takeaways directly shaped the final methodology and suggest clear next steps. In particular, future work would benefit from cleaner and more diverse data, background-independent preprocessing through hand segmentation, and temporal modelling capable of capturing the subtle micro-movements present in gestures such as the transition from *I* to *J*.

## 10 Hardware and Infrastructure

All experiments were conducted in GPU-accelerated environments to ensure practical training times for the MobileNetV2 architecture and the full experimental suite. The primary hardware used was a Tesla T4 GPU with 16 GB of VRAM running on a CUDA backend, paired with 12-16 GB of system memory depending on the runtime environment. This configuration provided sufficient compute capacity for both the frozen-feature stage and the subsequent fine-tuning phase, while also supporting rapid iteration during ablations. The final MobileNetV2 model occupies roughly 14 MB, consistent with prior reports of its compact footprint (Sandler et al., 2019).

Training times were modest, given the efficiency of the architecture. With a batch size of 64, each epoch required approximately 30–35 minutes. The initial frozen-feature stage completed in roughly 1 hour, while the full experimental pipeline, including all baselines and seven ablation configurations, required around seven hours of total GPU time. These runtimes demonstrate that the system is feasible to train even on mid-range hardware, a key consideration for practitioners who may not have access to high-end compute resources.

The choice of MobileNetV2 was strongly motivated by these hardware constraints. Its depthwise-separable convolutions and inverted residual blocks dramatically reduce computational load, enabling fast inference and efficient training without sacrificing representational power. These characteristics make the model particularly suitable for real-time ASL recognition, especially in deployment scenarios involving mobile CPUs, embedded hardware, or edge devices. Prior ASL studies relying on MobileNet architectures (Abini et al., 2019; Lum et al., 2020) report similar findings, reinforcing the practicality of MobileNetV2 as a backbone for scalable gesture-recognition systems.

## 11 Conclusion and Team Contributions

This project developed a complete framework for static ASL gesture classification centred on a lightweight, transfer-learning–based MobileNetV2 architecture. By unifying two public ASL datasets into a 250k-image dataset and establishing a rigorous experimental pipeline with classical baselines, modern CNN benchmarks, and multiple transfer-learning variants, the work provides a clear empirical grounding for understanding the strengths and limitations of efficient vision models in this domain. The final system benefited substantially from extensive augmentation, a refined training strategy, and systematic ablations that isolated the influence of augmentation, regularization, optimizer choice, and fine-tuning depth. The resulting classifier achieves strong performance despite the dataset's heterogeneity, demonstrating the viability of compact architectures for real-world ASL recognition.

Several limitations remain. The system operates entirely on static images and therefore cannot capture the temporal characteristics that are essential for continuous signing, particularly for dynamic letters such as *J* and *Z*. Sensitivity to lighting variation, background clutter, and class imbalance also continues to affect certain gestures, especially those with subtle or ambiguous hand shapes. These observations suggest clear directions for advancement. Incorporating hand segmentation or MediaPipe-based tracking may help mitigate background noise, while contrastive pretraining could improve separation among fine-grained gestures. Dataset expansion, more representative sampling, and temporal modelling through 3D CNNs or video transformers represent promising avenues toward a more comprehensive ASL recognition system.

Overall, the progression of this project, from a minimal prototype to a mature experimental framework, highlights the value of principled baselines, careful dataset analysis, and targeted ablation studies in building reliable machine-learning systems. The findings underscore both the promise and the challenges of using lightweight CNNs for ASL recognition and provide a foundation for future work aimed at deploying robust, real-time, and fully continuous sign-language understanding systems.

All code, preprocessing scripts, and trained model checkpoints used in this project are available at: https://github.com/asherk7/Sign-Language-Translator

### Team Contributions

**Asher Khan** led the design and implementation of the model architecture, developed the transfer-learning and fine-tuning strategy, and authored the technical analysis and experimental sections of the report.

**Hamza Abou Jaib** managed dataset acquisition and preprocessing, implemented the augmentation pipeline, and contributed to document structure and formatting.

**Mehdi Syed** assisted with dataset merging, built preprocessing utilities, and produced the visualizations used in the results and error-analysis sections.

## References

M. A. Abini, Divya P. Lakshmi, K. S. Sharan, and V. N. Sulphiya. 2019. American sign language detection using transfer learning. *International Journal of Computer Trends and Technology*.

Arnav Gangal, Anusha Kuppahaly, and Malavi Ravindran. 2024. Sign language recognition with convolutional neural networks. In *CS231N Course Project Report*.

grassknoted. 2020. Asl alphabet dataset.

Kapil Londhe. 2021. American sign language dataset.

Kin Yun Lum, Yeh Huann Goh, and Yi Bin Lee. 2020. Asl alphabet recognition using mobilenetv2. *ASTESJ*.

Nasrullah Nazaruddin. 2024. Signvision: A real-time mobile sign language recognition application using chatgpt.

Jonas Rheiner, Daniel Kerger, and Matthias Druppel. 2024. From pixels to letters: Building a high-accuracy, real-time asl gesture detection pipeline. *SSRN Electronic Journal*.

Jaya Prakash Sahoo, Allam Jaya Prakash, Pawel Plawiak, and Saunak Samantray. 2022. Real-time hand gesture recognition using convolutional neural networks. *Sensors*.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2019. Mobilenetv2: Inverted residuals and linear bottlenecks. *arXiv preprint arXiv:1801.04381*.

Mingxing Tan and Quoc V. Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*.

TensorFlow Authors. 2024. Transfer learning with tensorflow.

Lean Karlo Tolentino, Ronnie Serfa Juan, August Thio-ac, Maria Pamahoy, Joni Forteza, and Xavier Garcia. 2019. Static sign language recognition using deep learning. *ResearchGate Preprint*.