

محمد القثمي

ASL_Research_Paper_v8.docx

 Midocean University

Document Details

Submission ID

trn:oid::2945:400484767

Submission Date

Jul 21, 2026, 12:50 AM GMT+3

Download Date

Jul 21, 2026, 2:17 AM GMT+3

File Name

ASL_Research_Paper_v8.docx

File Size

4.4 MB

31 Pages

11,411 Words

66,948 Characters

16% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

-  **130 Not Cited or Quoted 13%**
Matches with neither in-text citation nor quotation marks
-  **28 Missing Quotations 2%**
Matches that are still very similar to source material
-  **2 Missing Citation 1%**
Matches that have quotation marks, but no in-text citation
-  **1 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 11%  Internet sources
- 12%  Publications
- 14%  Submitted works (Student Papers)

Integrity Flags

2 Integrity Flags for Review

-  **Replaced Characters**
30 suspect characters on 16 pages
Letters are swapped with similar characters from another alphabet.
-  **Hidden Text**
464 suspect characters on 11 pages
Text is altered to blend into the white background of the document.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 130 Not Cited or Quoted 13%**
Matches with neither in-text citation nor quotation marks
- 28 Missing Quotations 2%**
Matches that are still very similar to source material
- 2 Missing Citation 1%**
Matches that have quotation marks, but no in-text citation
- 1 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 11% Internet sources
- 12% Publications
- 14% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	www.mdpi.com	<1%
2	Submitted works	University of Sheffield on 2025-08-27	<1%
3	Submitted works	Islington College,Nepal on 2026-05-09	<1%
4	Submitted works	Al Hussein bin Talal University on 2024-01-08	<1%
5	Internet	arxiv.org	<1%
6	Internet	github.com	<1%
7	Internet	journal.binus.ac.id	<1%
8	Internet	ph.pollub.pl	<1%
9	Submitted works	Asia Pacific University College of Technology and Innovation (UCTI) on 2026-05-05	<1%
10	Submitted works	UCSI KL Campus on 2026-05-04	<1%

11	Submitted works	Institute of International Studies on 2023-08-22	<1%
12	Submitted works	NALSAR University of Law Hyderabad on 2026-04-06	<1%
13	Internet	deepai.org	<1%
14	Publication	Poonam Nandal, Tapas Kumar, Meeta Singh, Mamta Dahiya. "Progressive Compu...	<1%
15	Submitted works	National College of Ireland on 2025-04-11	<1%
16	Internet	impdiagnostics.com	<1%
17	Publication	"Innovative Computing and Communications", Springer Science and Business Me...	<1%
18	Publication	Shivani Singh, Shanti Chandra, Divyanshu Awasthi. "Enhanced american sign lang...	<1%
19	Submitted works	Miami Dade College on 2025-06-08	<1%
20	Internet	journals.abuad.edu.ng	<1%
21	Submitted works	Delhi Technological University on 2026-04-22	<1%
22	Internet	sci-conf.com.ua	<1%
23	Submitted works	Faculty of Computer Science & Information Technology on 2026-07-06	<1%
24	Publication	Arash Nasr Esfahani, Amirreza Jalili, Hedieh Sajedi. "NeXtSwin-X: dual-branch cros...	<1%

25	Submitted works	Liverpool John Moores University on 2024-06-11	<1%
26	Submitted works	Westcliff University on 2026-04-27	<1%
27	Submitted works	Kingston University on 2025-01-09	<1%
28	Submitted works	Open Institute of Technology (OPIT) on 2026-02-22	<1%
29	Internet	www.ukm.my	<1%
30	Publication	Vi Duong-The, Quan Le-Hoang-Minh, Quang Bui-Minh, Thanh Pham-Phu, Minh Do...	<1%
31	Internet	imanagerpublications.com	<1%
32	Publication	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dharendra Kumar Shukla. "Artific...	<1%
33	Publication	Nisha, Sonu, Satya Narayan, Jyoti Gajrani. "Chapter 48 Lightweight Deep Learnin...	<1%
34	Publication	Ba-Duy Nguyen, Van-Dung Hoang, Hien D. Nguyen. "Enhancing Skin Lesion Classi...	<1%
35	Submitted works	Jawaharlal Nehru Technological University on 2025-07-30	<1%
36	Submitted works	Lebanese University on 2017-10-29	<1%
37	Submitted works	National Sun Yat-sen University on 2025-08-28	<1%
38	Submitted works	University of Lincoln on 2025-08-26	<1%

39	Submitted works	A'Sharqiyah University, Oman on 2023-03-02	<1%
40	Submitted works	Heriot-Watt University on 2025-03-27	<1%
41	Internet	conf.domnit.ru	<1%
42	Internet	papers.phmsociety.org	<1%
43	Internet	www.gyaanbodh.com	<1%
44	Internet	www.tup.com	<1%
45	Publication	Jaskaran Singh, Meenu Gupta, Rakesh Kumar. "Smart Technologies and Intelligen...	<1%
46	Submitted works	Al-Hussein Bin Talal University on 2026-06-14	<1%
47	Internet	ediss.uni-goettingen.de	<1%
48	Publication	Vidhi Bansal, Sandali Sinha, Rani Astya, Anil Kumar Sagar, Kalicharan Sahu. "A Hy...	<1%
49	Publication	"Data Mining and Information Security", Springer Science and Business Media LL...	<1%
50	Internet	studenttheses.uu.nl	<1%
51	Submitted works	Midocean University on 2026-05-02	<1%
52	Internet	asianentcarecentre.com	<1%

53	Internet	rvce.edu.in	<1%
54	Submitted works	Dublin Business School on 2025-08-28	<1%
55	Publication	Siddhartha Roy, Soumya Sen, Agostino Cortesi. "Intelligent Systems - Emerging Tr...	<1%
56	Internet	ebin.pub	<1%
57	Internet	www.nature.com	<1%
58	Submitted works	College of Public Health on 2026-03-09	<1%
59	Submitted works	Mindanao State University - IIT on 2026-07-14	<1%
60	Submitted works	National Institute of Technology Silchar on 2026-07-19	<1%
61	Submitted works	The Robert Gordon University on 2026-04-06	<1%
62	Submitted works	University of New York Tirana on 2026-07-19	<1%
63	Internet	pmc.ncbi.nlm.nih.gov	<1%
64	Internet	www.frontiersin.org	<1%
65	Publication	"Information Systems for Intelligent Systems", Springer Science and Business Me...	<1%
66	Submitted works	Faculty of Engineering, Chulalongkorn University on 2026-07-13	<1%

67	Publication	Fatma M. Najib. "A multi-lingual sign language recognition system using machine...	<1%
68	Publication	Ghulfam Hussain. "Prostate Cancer Detection in Bi-parametric MRI Using Deep Le...	<1%
69	Submitted works	Thesis 2026 on 2026-06-14	<1%
70	Submitted works	University of Glasgow on 2026-04-29	<1%
71	Internet	cdr.lib.unc.edu	<1%
72	Internet	edepot.wur.nl	<1%
73	Internet	www.sciedupress.com	<1%
74	Internet	www.scitepress.org	<1%
75	Internet	www.themoonlight.io	<1%
76	Publication	Alshagathrh, Fahad Muflih. "Advancing Non-Alcoholic Fatty Liver Disease Diagnos...	<1%
77	Publication	Hernández, Valeria Martín. "Building Labeled Satellite Imagery Datasets and Ben...	<1%
78	Submitted works	Islington College,Nepal on 2026-05-10	<1%
79	Submitted works	National College of Ireland on 2025-08-10	<1%
80	Submitted works	RDI Distance Learning on 2026-01-14	<1%

81	Submitted works	Shahid Shah on 2026-06-17	<1%
82	Submitted works	South Bank University on 2011-06-12	<1%
83	Submitted works	The African Institute for Mathematical Sciences on 2025-06-17	<1%
84	Submitted works	Thesis 2026 on 2026-02-17	<1%
85	Submitted works	University of East London on 2025-09-01	<1%
86	Submitted works	University of Glasgow on 2024-04-23	<1%
87	Submitted works	University of Greenwich on 2026-05-01	<1%
88	Submitted works	University of Southampton on 2024-05-03	<1%
89	Submitted works	University of Surrey on 2018-05-22	<1%
90	Publication	Yang, Haoming. "Incorporating Data Interaction in Deep Learning Architecture, T..."	<1%
91	Internet	actainformaticamalaysia.com	<1%
92	Internet	dirros.openscience.si	<1%
93	Internet	isprs-annals.copernicus.org	<1%
94	Internet	mdpi-res.com	<1%

95	Internet	oa.upm.es	<1%
96	Internet	studentrepo.iium.edu.my	<1%
97	Submitted works	universititeknologimara on 2026-07-20	<1%
98	Internet	www.atlantis-press.com	<1%
99	Internet	www.springerprofessional.de	<1%
100	Internet	www.techscience.com	<1%
101	Publication	Alsharif, Bader. "Advances in Real-Time American Sign Language Recognition Sys...	<1%
102	Submitted works	Beirut Arab University on 2020-06-07	<1%
103	Publication	Keerthi Kumar M, Parameshachari BD. "Pseudo-Temporal 3D CNN Fusion of Gradi...	<1%
104	Publication	Kumar, Dheeraj. "Automated Detection and Classification of Cholelithiasis from U...	<1%
105	Submitted works	Midocean University on 2025-09-05	<1%
106	Submitted works	Universidad Francisco de Vitoria on 2026-01-08	<1%
107	Publication	Wang, Sijia. "Towards Sustainable Machine Learning Systems.", Duke University	<1%
108	Submitted works	City University on 2024-12-09	<1%

109	Publication	Gonzalez, Sara. "From Morphology to Machine Learning and Genomics: Understa...	<1%
110	Publication	Jonas Rheiner, Daniel Kerger, Matthias Drüppel. "From pixels to letters: A high-ac...	<1%
111	Submitted works	Lebanese University on 2017-10-29	<1%
112	Submitted works	Technical University of Košice on 2026-06-03	<1%
113	Submitted works	University of Anbar on 2021-10-28	<1%
114	Submitted works	University of Diyala on 2017-10-09	<1%
115	Submitted works	University of Ulster on 2025-12-04	<1%
116	Publication	Wei Li, Jiahao Jiang, Yi Yao, Danian Liu, Yang Gao, Qi Li. "Hand character gesture r...	<1%

A Comparative Study of CNN, Transformer, and Hybrid Deep Learning Models for American Sign Language Fingerspelling Recognition

دراسة مقارنة لنماذج التعلم العميق القائمة على الشبكات الالتفافية والمحولات والنموذج
الهجين للتعرف على تهجئة الأصابع في لغة الإشارة الأمريكية

Anas Abdulkader Mohammed Alzain – Mohammed Omar Abdullah Alqathmi

ORCID ID: 0009-0001-9982-4864 - ORCID ID: 0009-0003-4905-0447

Artificial Intelligence – College of Informatics – Midocean University – Comoros

Email: anas11223@gmail.com – Email: mqathmi@hotmail.com

Dr. Hagar Saleh (Supervisor) – ORCID ID: XXXX-XXXX-XXXX-XXXX

College of Informatics, Midocean University, Comoros — Email:
XXXXXXXXXXXXXXXXXXXX

Abstract

Objectives: This study aims to compare six deep learning architectures — ResNet50, MobileNetV2, EfficientNetB0, a standard Vision Transformer, a hierarchical Swin Transformer, and a novel Hybrid MobileNetV2-Swin fusion model proposed by the authors — for recognizing American Sign Language (ASL) fingerspelling from static images, and to build a supporting word-formation and spelling-correction demonstration tool.

Study Location: College of Informatics, Midocean University; experiments were executed on cloud GPU infrastructure.

Methodology: A 36-class ASL fingerspelling dataset (2,515 images) was split 70/10/20 using a fixed random seed. Three CNNs were adapted via transfer learning, while the Vision Transformer, Swin Transformer, and the proposed Hybrid model were fully fine-tuned. All six models were trained under one unified protocol and evaluated using accuracy, weighted precision/recall/F1-score, macro F1-score, and inference time on the same GPU.

Results: The Hybrid model achieved the highest accuracy (99.80%), narrowly ahead of the Swin Transformer (99.60%) and the Vision Transformer (98.81%), and well ahead of the three CNNs (89.9%–92.5%). Unexpectedly, the Swin Transformer was the fastest model overall. The digit '0' and the letter 'o' were the dominant, near-universal source of classification error.

Recommendations: Deploy the Swin Transformer where both speed and accuracy matter, the Hybrid model where maximum accuracy is required, and extend the system to dynamic-gesture recognition and video-conferencing integration.

Keywords: American Sign Language, Deep Learning, Vision Transformer, Swin Transformer, Hybrid Model.

المخلص

الأهداف: تهدف هذه الدراسة إلى مقارنة ست معماريات من معماريات التعلم العميق ResNet50، و MobileNetV2، و EfficientNetB0، ومحول رؤية قياسي، ومحول Swin الهرمي، ونموذج هجين جديد يدمج MobileNetV2 مع Swin اقترحه الباحثان للتعرف على حروف وأرقام لغة الإشارة الأمريكية (ASL) بنظام التهجنة الإصبعية من صور ثابتة، وبناء أداة مساعدة لتكوين الكلمات وتصحيحها إملائياً.

مكان الدراسة: كلية المعلوماتية، جامعة ميدأوشن؛ نُفذت التجارب على بنية تحتية سحابية لمعالجة الرسوميات. المنهجية: استُخدمت مجموعة بيانات من 36 فئة لحروف وأرقام لغة الإشارة الأمريكية (2,515 صورة)، قُسمت بنسبة 20/10/70 ببذرة عشوائية ثابتة. تم تكييف ثلاثة نماذج شبكات عصبية التلافيفية عبر التعلم بالنقل، بينما جرى الضبط الدقيق الكامل لمحول الرؤية ومحول Swin والنموذج الهجين المقترح. لُزيت جميع النماذج الستة وفق بروتوكول موحد، وقيمت باستخدام الدقة، والدقة الموزونة والاستدعاء ودرجة F1، ومتوسط F1، وزمن الاستدلال، على نفس معالج الرسوميات.

النتائج: حقق النموذج الهجين أعلى دقة (99.80%)، بفارق ضئيل عن محول Swin (99.60%) ومحول الرؤية (98.81%)، وبفارق كبير عن النماذج الثلاثة الالتفافية (89.9%-92.5%). وبشكل غير متوقع، كان محول Swin الأسرع بين النماذج الستة جميعها. وشكل الرقم '0' والحرف 'o' المصدر السائد والشبه الوحيد للخطأ عبر كل النماذج. التوصيات: يُوصى باستخدام محول Swin في التطبيقات التي تتطلب السرعة والدقة معاً، والنموذج الهجين عند الحاجة لأقصى دقة ممكنة، مع توسيع النظام مستقبلاً ليشمل الإيماءات الديناميكية والتكامل مع منصات المؤتمرات المرئية. الكلمات المفتاحية: لغة الإشارة الأمريكية، التعلم العميق، محول الرؤية، محول Swin، النموذج الهجين.

1. Introduction

Sign language is the main form of communication for Deaf and hard-of-hearing persons, but most hearing people have little knowledge of it, this way forming a communication gap that impeded education, employment, and social interaction, in general. Based on the World Health Organization, about 1.5 billion people worldwide have hearing loss, out of which around 430 million people have a level of hearing loss disabling them to such an extent that intervention is needed; this number is expected to increase to around 2.5 billion - that's nearly one in four people - by the year 2050 (World Health Organization, 2021).

With Artificial Intelligence (AI), In particular Computer Vision (CV), the doors for automatic sign-language recognition have widened. With Convolutional Neural Networks, it was the driving force for the past ten years of gradual increase in gesture-recognition accuracy (LeCun et al. 2015), and lately, Vision Transformers - the kinds of hierarchical ones like the Swin Transformer, and the ones that combine the convolutions with the attention features - brought it even further, allowing models to take in not only the detailed local features but also the long-range context in an image at the same time (Dosovitskiy et al. 2021; Liu et al. 2021). American Sign Language (ASL) fingerspelling, in which a distinct hand shape represents each letter and digit, is one of the foundational building blocks of the language. Automatic recognition of fingerspelling gestures is an important first step toward more comprehensive sign-language translation systems. Earlier automatic recognition systems relied on invasive, sensor-based hardware; more recent, camera-based approaches simplified data acquisition considerably, though lighting, background clutter, and hand-pose variation remain persistent challenges (Khan et al., 2020). Recognizing an isolated letter, however, is only the first step toward a genuinely useful communication aid — a practical system must also assemble recognized letters into words and correct the errors that inevitably accumulate across a multi-letter

sequence, a capability rarely addressed together with the classification task itself in the existing literature.

This study is motivated by the goal of building and rigorously evaluating a reliable, end-to-end ASL fingerspelling recognition system, in order to help close the communication gap between the Deaf and hard-of-hearing community and the hearing population. Specifically, six deep learning architectures are compared on the task of recognizing ASL fingerspelling from static images: three well-established convolutional networks (ResNet50, MobileNetV2, EfficientNetB0), a standard Vision Transformer, a hierarchical Swin Transformer, and a novel Hybrid MobileNetV2-Swin fusion model proposed in this study. To illustrate what the best-performing models make possible in practice, a supporting demonstration tool was also built that assembles a sequence of recognized letters into a word and applies automatic English spelling correction. The remainder of this paper is organized as follows: Section 2 presents the research problem; Section 3 discusses the importance of the study; Section 4 lists the study objectives; Section 5 reviews the theoretical background; Section 6 reviews previous studies; Section 7 details the proposed methodology; Section 8 presents the experimental results; and the paper closes with a discussion, conclusion, and recommendations.

1.1 Research Motivation and Contribution

Our work was motivated by three intertwined needs. For one thing, many sign language recognition papers either present results of one model or one model family which makes it hard to see whether the performance gain came from a better architecture or the dataset split or even from the experimental setup. Besides that, static fingerspelling systems need to handle classes that are visual duplicates while maintaining a response speed fast enough for interactive use.

Finally, the usability of a system is improved when it can be seamlessly combined with a user interface that can generate a text of a sequence of recognized characters. These were the motivations for the present comparative and application-oriented setup.

We highlight the four main contributions of this research: (1) an executable six-mode comparison that includes three convolutional networks, two transformer types, and a brand new hybrid fusion model; (2) a parallel branch MobileNetv2-Swin that combines different local and global feature representations; (3) a common set of 36 classes, with image split of 70/10/20, and consistent evaluation through the use of accuracy, weighted and macro-averaged metrics, inference latency, parameter count, and model size; (4) a category-level error examination that reveals the misunderstanding between digit 0 and letter o; and finally (5) a demonstration through the medium of Gradio which can support recognition of single image, building whole word from set of images, selection of model, showing confidence, and fixing spelling of the English output.

1.2 Organization of Paper

The rest chapters can be found in Section 2 (defining the research problem), Section 3 (justifying the significance of the work) and Section 4 (listing the goals). Section 5 gives us with the foundational system. Section 6 gives an account of past work and locates the present study within the research gap. Section 7 is devoted to the description of the data preparation models, the training setting, the algorithm we developed, and the application sequence. Section 8 reports and evaluates the results of the experiments. Section 9 is where we address the

limitations of our work and discuss how our findings relate to previous research. Section 10 concludes with our practical suggestions and ideas for further research.

2. Research Problem

Despite substantial progress in deep-learning-based sign-language recognition, the existing literature shares several important limitations. First, the majority of prior studies compare only conventional CNN architectures against one another (e.g., AlexNet, VGG, ResNet), and even the smaller, more recent body of work that evaluates Vision Transformers rarely also evaluates a purpose-built hybrid CNN-Transformer fusion model under the same experimental conditions. Second, few studies report inference time, model size, and parameter count alongside accuracy, despite their direct relevance to real-time, resource-constrained assistive applications; where inference time is reported, it is rarely re-examined for the counter-intuitive possibility that a well-optimized transformer might outperform a CNN on speed as well as accuracy. Third, per-class error analysis — identifying which specific letters or digits are systematically confused by a given model, and why — is rarely reported in detail. Fourth, essentially no study pairs a classification benchmark with a working demonstration of multi-letter word assembly and automatic spelling correction, leaving a gap between a system that can classify a single letter and one that can help a Deaf or hard-of-hearing user communicate a whole word. The problem addressed by this study, therefore, is the absence of a rigorous, unified experimental comparison spanning conventional CNNs, a standard Vision Transformer, a hierarchical Swin Transformer, and a novel hybrid fusion model, evaluated identically and paired with a working word-formation and spelling-correction demonstration.

3. Importance of the Study

From the academic point of view this paper offers to the research field a single comprehensive, re-creatable, six-fold benchmark that is able to compare three CNN variants, a regular Vision Transformer, a hierarchical Swin Transformer, and an innovative hybrid fusion architecture, all in parallel under the same experimental setup - such a broad comparative study is not really typical for the available literature. Besides that, it also offers an updated analysis of the correctness and speed trade-off, which was able to contradict the commonly-held beliefs that transformer models necessarily are slower than CNNs, and a class-wise error breakdown revealing the particular visual errors that each model tends to confuse.

In reality, the work can help the Deaf and the hard-of-hearing community: through an automated ASL finger spelling recognition, coupled with the spell check and grammar features, this system makes communicating an education, a service, or a meeting at a distance feasible. The research gives clear direction for professionals and academics choosing the right model architecture for one specific setting - be it highest accuracy, shortest time to first result, or smallest model size - instead of assuming that CNNs or transformers are always one way or the other, as the results of the work prove it isn't a fixed case.

Bigger picture wise, the paper is connected with the emerging trend of using hybrid CNN-Attention Transformer models not only for sign language but also other visual discrimination tasks like medical diagnosis, where the same technique of combining lightweight convolution with Hierarchical Vision Transformer - self-attention has already led to some tangible enhancements. Showing that such a model design also applies to ASL finger spelling data that is a couple of orders of magnitude smaller than what one usually uses to train Vision Transformers from zero, proves that even though data is scarce hybrid fusion remains a good

solution strategy. Without a doubt this is a typical situation assistive-technology researchers face when developing for a language or a community that is not yet mainstream but still a large number of people have it as a means of communication.

4. Study Objectives

- To prepare a 36-class ASL fingerspelling image dataset with a fixed, reproducible train/validation/test split.
- To design a preprocessing and data-augmentation pipeline appropriate to each model family.
- To train three pretrained CNNs (ResNet50, MobileNetV2, EfficientNetB0) via transfer learning, a standard Vision Transformer, a hierarchical Swin Transformer, and a novel Hybrid MobileNetV2-Swin fusion model.
- To evaluate all six models under one controlled comparative protocol using accuracy, weighted precision/recall/F1-score, macro F1-score, inference time, model size, and parameter count.
- To conduct a detailed per-class error analysis identifying which letters and digits challenge each model most, and why.
- To design and demonstrate a supporting multi-letter word-formation and automatic spelling-correction pipeline within an interactive multi-model web application.
- To identify the most efficient model overall and propose practical recommendations for deployment, including integration into real-time communication platforms.

5. Theoretical Literature of the Study

Artificial Intelligence (AI) is an area of computer science that deals with the development of systems that can do activities that require human intelligence, like learning, perception, and decision-making (Russell & Norvig, 2021). Rather than executing through programmed instructions, AI systems detect statistical patterns from data, which gives them the possibility to change given the situations. Computer Vision (CV), considered one of the most practical applications of AI, is a technology that allows computers to comprehend visual information by first extracting from a pixel-level information, then transforming into representations that facilitate tasks like pattern recognition and object classification (Szeliski, 2022), e. g. recognizing hand gestures and sign language which this manuscript centers on.

Machine learning, with one of the main branches being Deep Learning (DL), builds on the usage of multi-layered neural networks for hierarchical feature learning from raw input, eventually turning it into increasingly higher-levelling, task-relevant representations (LeCun et al. 2015). The power in automatically doing the hierarchical feature extraction distinguishes deep learning from other, previous machine learning forms, the ones where a designer manually crafted the features. That very feature has been a major enabling tool for recognizing sign language directly from a video without further pre-processing, like Li et al. (2021).

Convolutional Neural Networks (CNNs), which are deep learning setups In particular designed for processing data arranged in grids, are best known for their capability to extract relevant features from digital images, the most common being that of the RGB pixel format (Khan et al. 2020). The CNN model is typically composed of convolutional layers. These filters extract the different features (like line point texture, etc.) from the given image and create feature maps. Pooling layers, a type of spatial dimensions reducer with the added benefit of increasing

robustness against small translations in the position of the object are also included in a typical CNN architecture. In addition, fully connected layers that integrate the features extracted into a final representation, typically a probability distribution of the correct output labels via Softmax, are also found in CNNs. To reduce a problem of overfitting, a Dropout layer - which randomly deactivates the selected proportion of the neural network during training - is used most commonly (Srivastava et al. 2014). In this article a model trained on the data has been used with a technique called transfer learning - instead of starting from scratch the model has been adapted to a new task - hand signs classification in a 36 classes ASL dataset by training only the classification part of the model. We have tested with three different CNN models, ResNet50 (He et al. 2016), MobileNetV2 (Sandler et al. 2018), and EfficientNetB0 (Tan & Le, 2019). In each case the layers responsible for learning features are not being retrained (they are frozen), only a layer, responsible for the final classification of inputs, is a new one and it is trained for the ASL task having 36 classes.

Vision Transformers (ViT), in effect, borrow the idea of the self-attention mechanism, a building block introduced for language processing, and apply it directly to blocks of images. That is, a picture is divided into equal-sized patches that are then embedded linearly and a Transformer encoder equipped with multi-head self-attention mechanism applies to the series of such a patch (Dosovitskiy et al. 2021). Because self-attention relates every patch to every other patch in a single operation, a Vision Transformer can capture long-range contextual relationships from its very first layer, though at the cost of requiring substantially more parameters and training data, since it lacks the built-in spatial inductive bias that convolutional layers provide by design.

The Swin Transformer is a hierarchical variant of the Vision Transformer that reintroduces a CNN-like inductive bias into self-attention (Liu et al., 2021). It processes an image through four successive stages, progressively merging patches to reduce spatial resolution while increasing feature depth, alternating between window-based multi-head self-attention (W-MSA), computed within small local windows for efficiency, and shifted-window multi-head self-attention (SW-MSA), which shifts window boundaries between blocks to let information flow across them. This design combines the computational efficiency of local, convolution-like processing with a transformer's ability to model long-range dependencies. Evidence from outside the sign-language domain supports the value of this design: a hybrid model that used the Swin Transformer as a feature extractor for monkeypox skin-lesion classification achieved 95.6% accuracy, clearly outperforming conventional CNN baselines on the same datasets (Saleh et al., 2024) — a result that directly inspired both the inclusion of the Swin Transformer in this comparison and the Hybrid MobileNetV2-Swin architecture proposed here.

A recent and fast-growing line of work combines convolutional and self-attention-based feature extraction within a single model, on the hypothesis that the two representations capture complementary information. This study contributes a Hybrid MobileNetV2-Swin architecture in this spirit: a lightweight MobileNetV2 branch and a Swin Transformer branch process the same input image in parallel, and their pooled feature vectors are concatenated into a single fused representation before classification. Similar dual-branch fusion strategies have shown strong results elsewhere in the sign-language literature (Aly & Fathi, 2025; Ma et al., 2022; Zhidebayeva et al., 2024).

5.1 Evaluation Criteria

Evaluating a multi-class image classification system such as the one proposed in this study requires a well-defined set of quantitative criteria. Accuracy — the proportion of correct predictions out of all predictions — is the most widely reported metric, but it can be misleading

when class distributions are imbalanced, since a model may appear accurate overall while performing poorly on minority classes. For this reason, this study reports weighted precision, recall, and F1-score alongside accuracy and macro-averaged F1-score, together with inference time, model size, and parameter count, formally defined in Section 7.6. Reporting both weighted and macro-averaged F1-scores is particularly important in an assistive-technology context: a model that performs well on average but poorly on a handful of rarely used letters could still fail a user who relies specifically on those letters, so the macro-averaged score, which weights all 36 classes equally regardless of how many test samples each contains, offers a complementary and arguably fairer picture of overall reliability.

6. Review of Previous Studies

Prior work on automatic sign-language recognition can be organized into four thematic groups: single-CNN approaches, hybrid and multi-stream approaches, detection-based approaches, and vision-transformer approaches.

6.1 CNN-Based Approaches

The majority of prior work on ASL recognition relies on a single CNN architecture. Zakariah et al. (2022) used transfer learning to recognize thirty-two Arabic Sign Language signs, fine-tuning a pretrained EfficientNetB4 model on 54,049 images from 40 signers and reaching a maximum accuracy of 95%. Yadav and Bohra (2021) built a CNN-based system translating ASL fingerspelling into text and speech in real time, reaching 95.8% accuracy on static fingerspelling. Sheth et al. (2023) proposed a CNN-based model that both recognizes and generates ASL signs, reaching 98.84% training accuracy with strong robustness to lighting and skin-tone variation. Kamruzzaman (2020) showed the same CNN-based approach generalizes to Arabic Sign Language recognition, reaching 90% accuracy. Marjusalinah et al. (2021) compared ResNet50 with DenseNet121 for ASL fingerspelling classification, with ResNet50 reaching a near-perfect 99.99% accuracy. Hurroo and Walizad (2020) trained a simple grayscale-input CNN, reaching 99% training and 98% testing accuracy. Sharma (2020) concluded, in a broad review, that deep-learning methods consistently outperform earlier hand-crafted-feature approaches, though complex backgrounds remain a persistent challenge. Collectively, this cluster confirms that CNNs, and ResNet50 in particular (Marjusalinah et al., 2021), are a strong, well-validated baseline — motivating the inclusion of ResNet50, MobileNetV2, and EfficientNetB0 as this study's three CNN baselines — though none of these studies benchmark their CNN against a Vision Transformer, a Swin Transformer, or a hybrid fusion model under matching conditions.

6.2 Hybrid and Multi-Stream Approaches

A second group of studies combines multiple feature sources or processing stages to improve robustness. Zhidebayeva et al. (2024) introduced a real-time fingerspelling recognition method using a two-stream deep CNN designed to capture spatio-temporal information. Ma et al. (2022) proposed a two-stream mixed CNN (TSM-CNN) fusing features from two consecutive frames to better handle the moving gestures required for the letters 'J' and 'Z', consistently outperforming a single-stream baseline across four CNN backbones. Bayrak et al. (2024) combined Complex Zernike Moments with a Complex-Valued Deep Neural Network, reaching 98.67% accuracy. Kouvakis et al. (2024) introduced an encoder-decoder approach for bandwidth-efficient transmission of ASL images, achieving approximately 99% reconstruction/recognition accuracy under noisy channel conditions. This cluster demonstrates that fusing multiple streams or representations of the same gesture consistently improves accuracy over a single-stream CNN — directly motivating the Hybrid MobileNetV2-Swin

architecture proposed here, in which the fused 'streams' are two structurally different feature extractors (a convolutional branch and a self-attention branch) rather than two temporal frames.

6.3 Detection-Based (YOLO) Approaches

A third group views the task of sign-language recognition as an object-detection task, and prioritizes real-time speed. Mujahid et al. (2021) employed YOLOv3 with a DarkNet-53 backbone which gave 97.68% mean average precision and also ran much faster than SSD and Faster R-CNN. When Bankar et al. (2022) found out that conventional CNN classifiers were getting worse under cluttered backgrounds, they used YOLOv5, which resulted in getting 88.4% accuracy against only 52.

98% for a standard CNN baseline. Aggarwal et al. (2023) designed ASLR-Net, a CNN model that's resistant to variations in lighting and background noise and achieved 96.9% accuracy under both seen and unseen conditions. Drishya (2023) pushed sign-language recognition into the realm of continuous video and was able to track hand movements in 3D very accurately. Though, such achievement came at the cost of greatly increased dataset requirements for the video. This group of studies makes the case that inference time and generalization to real-world conditions are, in a sense, as important as raw accuracy in a system that people can use - this study is a continuation of the research on inference time, model size, and parameter count in the same breath as accuracy for all the six models.

6.4 vision Transformer and Swin-Based Approaches

The last group, most closely related to sign-language recognition tasks, shifts away from convolutional neural networks (CNNs) towards attention-based models. Aly and Fathi (2025) implemented a model that combined a dual-path CONVNET with a Vision Transformer, for a total of 29 ASL gestures, and achieved an accuracy of about a near 100% on an 87,000-image dataset and a frame rate of 110 fps. This result shows very clearly that using a hybrid design of attention feature extraction and convolutional networks works extremely well for sign-language recognition - and also acts as a strong reason for including the Hybrid model in our proposal here. Alsharif et al. (2023) tested five types of deep learning models, like AlexNet, ConvNeXt, EfficientNet, and ResNet-150/50, as well as a Vision Transformer, and they found that the latter three deep CNN models and Vision Transformer performed much better at all times compared to early CNN designs. Kasapba et al. (2022) made Deep-ASL-Recognition, which is a specialized CNN and a specially prepared, highly varied ASL dataset, together giving over 99% accuracy. In fact, there was a study in 2024 which applied the Swin Transformer architecture across different sign-language datasets, and this led to findings where the hierarchical and shifted window setup was effective in many sign language benchmark tasks, for instance ("Applying Swin Architecture to Diverse Sign Language Sets", 2024). Dosovitskiy et al. (2021) pioneered the use of the self-attention mechanism without any convolutional layers applied directly to image patches in a pure Transformer network that matched or, under very large amounts of pretraining data, even exceeded the performance of the convolutional network. This architecture has been directly used in a way as one of the six models compared here.

Table (1): Comparison of representative previous studies with the present study.

Study	Model(s)	Best Result	Gap Addressed by This Study
Marjusalinah et al. (2021)	ResNet50 vs DenseNet121	Acc. = 99.99%	No transformer/hybrid benchmarked

Study	Model(s)	Best Result	Gap Addressed by This Study
Hurroo & Walizad (2020)	Grayscale CNN	Acc. = 98%	Static-only; no speed analysis
Ma et al. (2022)	Two-Stream Mixed CNN	Improved over single-stream	Fusion of frames, not architectures
Mujahid et al. (2021)	YOLOv3 (DarkNet-53)	mAP = 97.68%	No accuracy–speed trade-off vs transformers
Aly & Fathi (2025)	Hybrid CNN + ViT	Acc. = 99.97%	No Swin/CNN baselines under one protocol
Dosovitskiy et al. (2021)	Vision Transformer (ViT)	SOTA on ImageNet	Not benchmarked on ASL vs CNN/Swin/Hybrid
Liu et al. (2021)	Swin Transformer	SOTA on multiple benchmarks	Not benchmarked on ASL fingerspelling
This study (2026)	ResNet50, MobileNetV2, EfficientNetB0, ViT, Swin, Hybrid	Acc. = 99.80% (Hybrid)	Unified 6-model benchmark + speed + per-class analysis + word formation

7. Methodology

Figure 1 presents the end-to-end workflow of the proposed system, from raw image organization through preprocessing, model training and evaluation, to multi-letter word formation and automatic spelling correction; each stage is detailed in the subsections that follow.

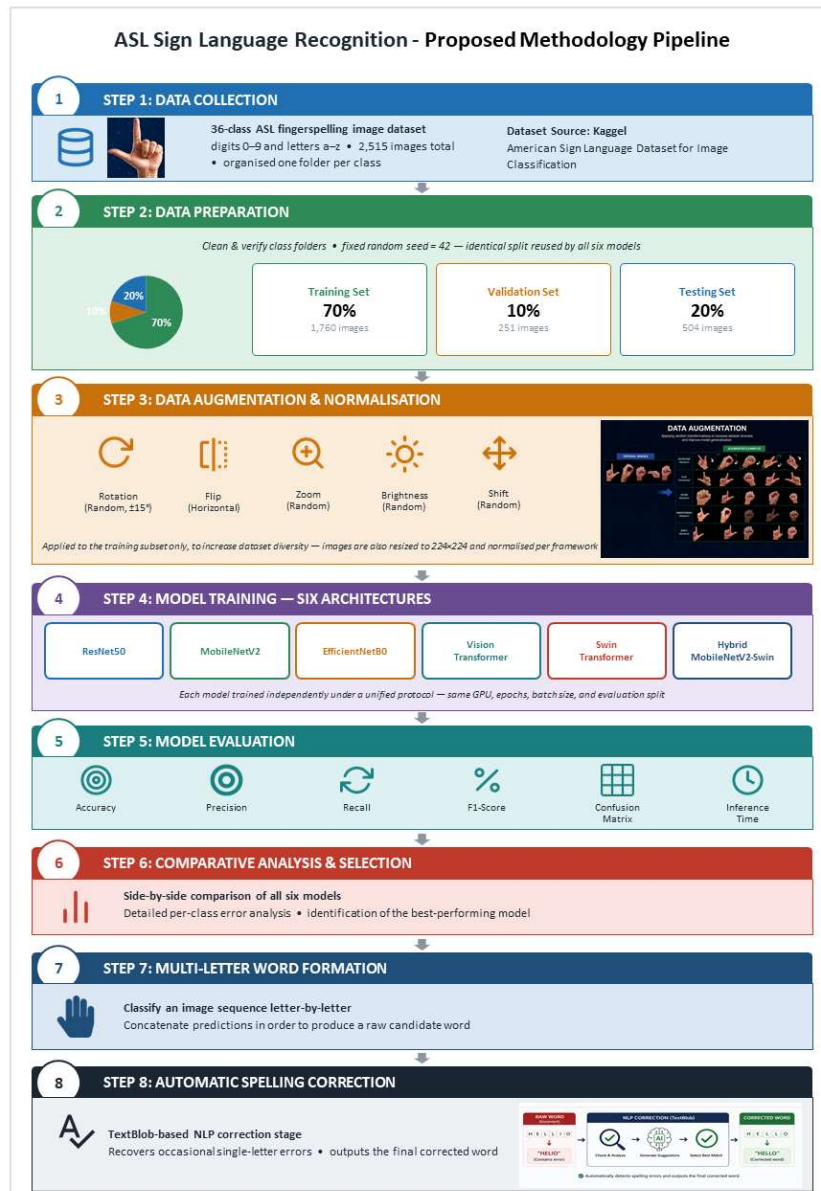


Figure (1): End-to-end workflow of the proposed ASL recognition and word-formation system.

7.1 Dataset Description

In this study, the researchers used an ASL finger-spelling dataset of images that have fingerspelling with 36 classes, that is, ten digits (0, 9) and twenty - six letters (a, z). There are a total of 2,515 images available with the class distribution broadly equal. The researchers first made a split of the dataset using a fixed random seed (42), into 70% training (1,760 images), 10% validation (251 images), and 20% testing (504 images); this same split was used by all six training pipelines resulting in every model being trained and evaluated on exactly the same set of images, i. e. 14 test images per class. The dataset has been test-balanced and is almost evenly distributed among the class folders, lowering the possibility that the overall accuracy measure is mainly the result of a small number of classes that have more samples than others.

All training class labels have been taken directly from the directory names of the images and stored in the alphanumeric order. The validation and test sets have never been augmented; augmenting was limited to the training dataset to prevent contaminating of the evaluation with the same pictures after the transformations. Due to the absence of signer identities, the split had to be done at the level of images. So the paper does not argue about signer Independent generalization from this data. Also, the entire ASL sign of letters J and Z has to be recognized based on motion paths; So, these letters are shown in the dataset as static hand poses only. So the results should, in fact, be interpreted as static class recognition rather than complete dynamic-sign recognition.

Table (2): Dataset composition and experimental split.

Subset	Images	Percentage	Use
Original dataset	2,515	100%	Source images across 36 classes
Training	1,760	70%	Model learning with augmentation
Validation	251	10%	Model selection and early stopping
Testing	504	20%	Final held-out evaluation (14 images/class)

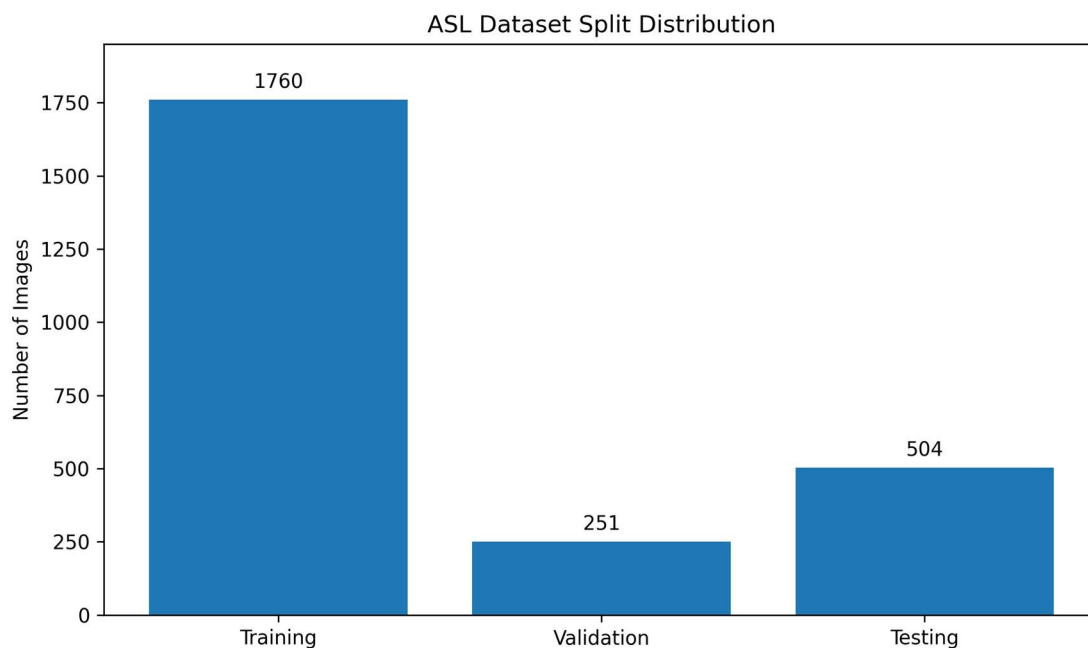


Figure (2): Distribution of images across the training, validation, and testing subsets.

7.2 Preprocessing and Data Augmentation

All images were resized to 224 x 224 pixels. For the three TensorFlow/Keras CNN models, each architecture's ImageNet-specific preprocessing function was combined with training-only augmentation comprising random rotation up to 20 degrees, horizontal and vertical shifts of 0.15, zoom of 0.15, shear of 0.10, and brightness adjustment between 0.7 and 1.3. Horizontal flipping was disabled because mirroring may alter the semantic interpretation of an ASL handshape. For the PyTorch-based Vision Transformer, Swin Transformer, and Hybrid MobileNetV2-Swin models, images were normalized using standard ImageNet statistics and the training subset was augmented using random rotation of +/-15 degrees and color jitter for brightness and contrast. Validation and test subsets were resized and normalized but were not augmented.

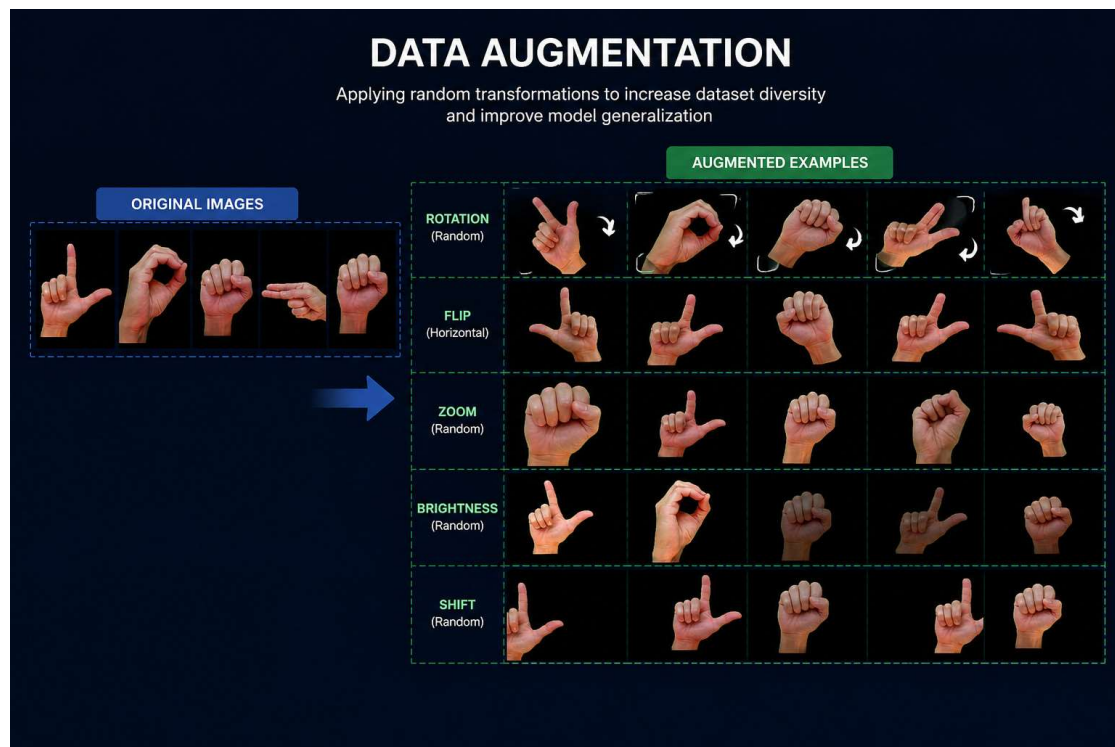


Figure (3): Illustrative data-augmentation transformations (rotation, horizontal flip, zoom, brightness, and shift) applied to the training subset only.

7.3 Model Architectures

Six architectures were compared: three pretrained CNNs adapted via transfer learning, a standard Vision Transformer, a hierarchical Swin Transformer, and a novel Hybrid MobileNetV2-Swin dual-branch fusion model. ResNet50 (He et al., 2016), a 50-layer CNN using residual connections, had its ImageNet-pretrained convolutional base (≈ 24.1 M parameters) frozen, with a new classification head trained for the 36-class task. MobileNetV2 (Sandler et al., 2018), designed for resource-constrained devices via depthwise-separable convolutions and inverted residual blocks, is the smallest of the six models (≈ 2.6 M parameters, 13.03 MB). EfficientNetB0 (Tan & Le, 2019) uses compound scaling to balance depth, width, and resolution (≈ 4.4 M parameters). The Vision Transformer (ViT-Base/16; Dosovitskiy et al., 2021) divides each image into 16×16 patches and applies multi-head self-attention across the full patch sequence (≈ 85.8 M parameters — the largest of the six); it was fully fine-tuned with all layers trainable. The Swin Transformer (Swin-Tiny; Liu et al., 2021) processes an image through four hierarchical stages alternating window-based (W-MSA) and shifted-window (SW-MSA) self-attention (≈ 27.5 M parameters); it too was fully fine-tuned.

7.3.1 ResNet50

ResNet50 uses residual skip connections to enable the optimization of a deep convolutional network without the degradation commonly observed when additional layers are stacked. In this study, the ImageNet-pretrained convolutional backbone was frozen and followed by global average pooling, a 256-unit ReLU layer, dropout of 0.5, and a 36-unit softmax classifier. The model served as a strong high-capacity CNN baseline.

7.3.2 MobileNetV2

MobileNetV2 is designed for computationally efficient inference through inverted residual blocks and depthwise separable convolutions. Its pretrained backbone was frozen and connected to the same 256-unit classification head used for the CNN baselines. With

approximately 2.60 million parameters and a 13.03 MB saved model, it provides the smallest deployment footprint in the comparison.

7.3.3 EfficientNetB0

EfficientNetB0 scales network depth, width, and input resolution in a coordinated manner. The pretrained feature extractor was frozen and connected to global average pooling, a dense layer, dropout, and a 36-class softmax output. It was included to evaluate whether compound scaling could offer a stronger accuracy-efficiency balance than the other CNN baselines.

7.3.4 Vision Transformer

First, the Vision Transformer chops up the image into equally sized squares, maps them onto corresponding tokens, and then explores the interactions through multi-head attention mechanism. This time, versus freezing the CNN backbones, the ViT model is retrained in total. As a result, they can discover global hand regions dependencies regardless of their spatial arrangement, but this method also results in the biggest model compared to the rest in the paper.

7.3.5 Swin Transformer

Feature maps and the shift-window attention technique are the two major components of the Swin Transformer. Within window areas, attention is calculated locally, and when a window is shifted, neighboring windows are used to exchange information. With this structure, the authors are able to cut down the expenses of global attention without affecting the multiscale visual structure. That means Swin is great to do handshape recognition at high detail.

7.3.6 Proposed Hybrid MobileNetV2-Swin

Our solution works like this: first of all, the two branches - parallel pretrained MobileNetV2 and pretrained Swin, are given a picture normalized. The outputs of pretrained MobileNetV2, the feature vector being 1,280-dimensional, are the representation of fine details which are the local textures, finger outlines, and compact spatial patterns. pretrained Swin-Tiny produces a feature vector 768-dimensional describing relationships on the wider spatial scale. In fact, one of these outputs is added to the other to get the final 2,048-dimensional vector which is used in a neural network for classification consisting of a 512-dimensional linear layer followed by the 36 output classes. This way of combination at the feature level preserves the unique character of each feature rather than simply calculating the average of probability scores of each class.

These operations given here can be considered a one-to-one representation of Hybrid Mobile Net Swin's forward passing logic. For an input x , the two frozen models extract feature vectors:

$$f_m = \text{MobileNetV2}(x), f_m \in \mathbb{R}^{1280} \quad (1)$$

$$f_s = \text{Swin}(x), f_s \in \mathbb{R}^{768} \quad (2)$$

The two feature vectors are concatenated along the feature dimension exactly as implemented with `torch.cat(..., dim=1)`:

$$f_h = f_m \oplus f_s, f_h \in \mathbb{R}^{2048} \quad (3)$$

The fused vector is then passed through the implemented 512-unit ReLU layer, dropout layer, and final 36-unit linear classifier:

$$h = \text{Dropout}(\text{ReLU}(W_1 f_h + b_1)) \quad (4)$$

$$z = W_2 h + b_2, z \in \mathbb{R}^{36} \quad (5)$$

During inference, the application converts the logits into class probabilities using SoftMax and returns the class with the highest probability:

$$p_c = \exp(z_c) / \sum_{i=1}^{36} \exp(z_i) \quad (6)$$

$$\hat{y} = \arg \max_c p_c \quad (7)$$

Here, $\oplus$ denotes vector concatenation, W_1 and W_2 are learned weight matrices, b_1 and b_2 are learned bias vectors, and $\hat{y}$ is the predicted ASL class.

The Hybrid MobileNetV2-Swin model proposed in this study is a dual-branch fusion architecture designed to combine the strengths of both preceding families. A MobileNetV2 backbone and a Swin Transformer backbone each process the same 224×224 image independently and in parallel, producing 1,280- and 768-dimensional feature vectors respectively; these are concatenated into a single 2,048-dimensional fused representation, passed through a classification head (fully connected $2,048 \rightarrow 512$, ReLU, Dropout $p=0.5$, fully connected $512 \rightarrow 36$). The resulting model (≈ 30.8 M parameters) is trained end-to-end, with both backbones fine-tuned jointly. This specific pairing was chosen for three reasons: (1) MobileNetV2's depthwise-separable convolutions capture local, fine-grained texture — precisely the detail distinguishing visually similar handshapes such as the digit '0' and the letter 'o' — while the Swin Transformer's shifted-window self-attention captures broader contextual relationships across the whole hand, so the fused representation carries both kinds of evidence; (2) both networks are individually lightweight relative to alternatives such as the standard Vision Transformer, so the combined model (30.81M parameters) remains smaller than a standalone ViT while gaining accuracy over either backbone alone; and (3) evidence from outside sign language — a Swin-based hybrid achieving a substantial accuracy gain over CNN baselines on a medical imaging task (Saleh et al., 2024) — suggested a similar hybrid could outperform CNN-only baselines on the fine-grained visual discrimination fingerspelling requires.

Proposed Hybrid MobileNetV2-Swin Architecture

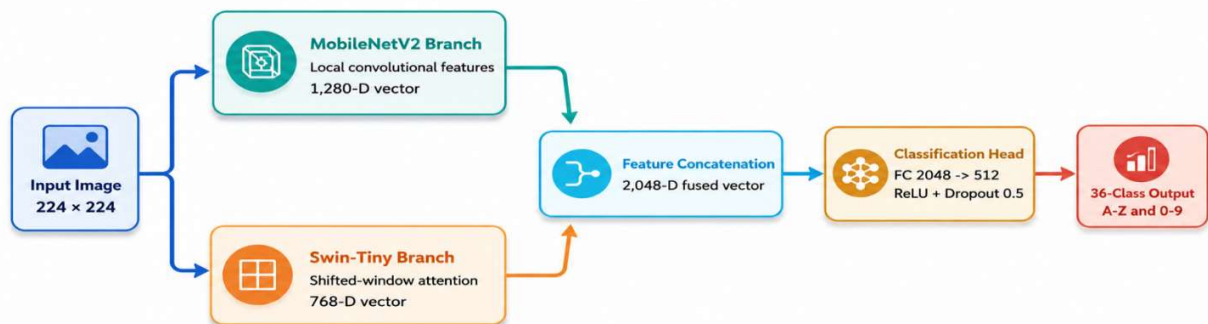


Figure (4): Dual-branch architecture of the proposed Hybrid MobileNetV2-Swin model.

Table (3): Summary of the six deep learning models compared in this study.

Model	Framework	Parameters	Fine-Tuning Strategy
ResNet50	TensorFlow/Keras	24.12 M	Frozen backbone + trainable head
MobileNetV2	TensorFlow/Keras	2.60 M	Frozen backbone + trainable head
EfficientNetB0	TensorFlow/Keras	4.39 M	Frozen backbone + trainable head
Vision Transformer (ViT-Base/16)	PyTorch (timm)	85.83 M	Full fine-tuning
Swin Transformer (Swin-Tiny)	PyTorch (timm)	27.55 M	Full fine-tuning
Hybrid MobileNetV2-Swin (proposed)	PyTorch (timm)	30.81 M	Full fine-tuning, both branches

7.4 Training Protocol

A controlled comparative protocol was adopted. All models used the same fixed train/validation/test split, 224 x 224 input size, batch size of 16, maximum of 30 epochs, initial learning rate of 0.0001, early-stopping patience of 7, ReduceLROnPlateau scheduling, and the same 504-image test set. Model-family-specific settings were retained where technically appropriate: the CNN baselines used Adam and categorical cross-entropy in TensorFlow/Keras, whereas the ViT, Swin, and Hybrid models used AdamW with weight decay of 0.01 and cross-entropy loss in PyTorch. The CNN backbones were frozen and only their newly attached classification heads were trained, while the transformer-based models were fine-tuned end-to-end. Consequently, the experiment is a controlled comparison under shared core conditions rather than a claim that every optimization and fine-tuning detail was identical.

The implemented training objective for all six models is multi-class cross-entropy. TensorFlow uses categorical_crossentropy for the CNN models, while PyTorch CrossEntropyLoss is used for ViT, Swin, and the Hybrid model. Both correspond to:

$$L_{CE} = -(1/N) \sum_{i=1}^N \sum_{c=1}^{36} y_{ic} \log(\hat{p}_{ic}) \quad (8)$$

where N is the number of samples, y_{ic} is the one-hot ground-truth indicator, and $\hat{p}_{ic}$ is the predicted probability for class c .

7.4.1 Experimental Environment

The experiments were executed in Google Colab using an NVIDIA A100-SXM4-40GB GPU. TensorFlow 2.20.0 was used for ResNet50, MobileNetV2, and EfficientNetB0. PyTorch 2.11.0 with CUDA 12.8 and TIMM 1.0.27 was used for ViT, Swin Transformer, and the Hybrid model. Python, NumPy, pandas, scikit-learn, Pillow, and Matplotlib supported preprocessing, metric calculation, result storage, and visualization. The application layer used Gradio 6.20.0 and TextBlob.

Inference time was measured on the complete 504-image test set after a warm-up pass. Each measurement was repeated three times, and the mean and standard deviation were stored. For the PyTorch models, GPU synchronization was applied before recording the end time. The resulting latency values describe the specific evaluation environment and should not be interpreted as universal hardware-independent speeds.

Table (4): Controlled training configuration and model-family-specific settings.

Parameter	Value
Image size	224 × 224
Batch size	16
Epochs	Up to 30 (early stopping, patience = 7)
Learning rate	0.0001
LR scheduler	ReduceLROnPlateau (factor = 0.3, patience = 4)
Optimizer	Adam (CNNs) / AdamW, weight decay = 0.01 (ViT, Swin, Hybrid)
Loss function	Categorical cross-entropy (CNNs) / Cross-entropy (PyTorch models)
Data split	70% train (1,760) / 10% validation (251) / 20% test (504)
Random seed	42 (fixed across all six models)
GPU	NVIDIA A100-SXM4-40GB (all six models, uniformly)

Parameter	Value
Fine-tuning strategy	Frozen CNN backbones; full end-to-end fine-tuning for ViT, Swin, and Hybrid

7.5 Proposed Algorithm

The overall procedure applied identically to each of the six models, from raw image input through training, evaluation, and the word-formation extension, can be summarized in the following steps:

- Step 1: Organize the raw image set into 36 class folders (digits 0–9, letters a–z).
- Step 2: Split the dataset into training (70%), validation (10%), and test (20%) subsets using a fixed random seed (42); reuse this identical split for every model.
- Step 3: Apply data augmentation to the training subset only (rotation, color jitter, or model-specific preprocessing).
- Step 4: Resize and normalize all images to 224×224 according to the statistics required by each model.
- Step 5: Initialize each model with pretrained weights; freeze the backbone and attach a new classification head for the three CNNs, or replace the classification head and fine-tune all layers for the Vision Transformer, Swin Transformer, and Hybrid model.
- Step 6: Train for up to 30 epochs, applying early stopping and learning-rate scheduling based on validation performance, and save the best-performing weights.
- Step 7: Evaluate the best model on the held-out test set, recording accuracy, weighted precision/recall/F1-score, macro F1-score, and total and per-image inference time.
- Step 8: For the word-formation extension, classify an ordered sequence of images letter by letter, concatenate the predictions into a raw word, and apply automatic spelling correction unless the word contains a digit.

Algorithm (1): Hybrid ASL recognition, word formation, and spelling correction.

Step	Operation
Input	Ordered ASL image sequence $X = \{x_1, \dots, x_n\}$; selected trained model M .
1	Resize each image to 224 x 224 and apply model-specific normalization.
2	For the Hybrid model, compute $fM(x_i)$ with MobileNetV2 and $fS(x_i)$ with Swin-Tiny.
3	Concatenate $z_i = [fM(x_i); fS(x_i)]$ and classify it with the fusion head.
4	Apply softmax and choose the class with maximum probability; retain the top five probabilities and confidence.
5	Append each predicted alphanumeric label in image order to construct the raw output string.
6	If the string contains alphabetic characters only, apply TextBlob spelling correction; otherwise preserve it unchanged.
Output	Predicted labels, confidence values, raw word, corrected word, and model information.

7.6 Word-Formation and Spelling-Correction Demonstration

Beyond single-image classification, a supporting natural-language-processing demonstration shows the system recognizing complete words. Two interaction modes are supported: sequential single-image word building, in which each predicted letter is appended to a running word string, and batch multi-image word formation, in which an ordered sequence of images is classified in one action and the predicted letters concatenated into a raw word. In both modes, the raw word is passed through a spelling-correction function wrapping the TextBlob library's statistical correction model, which identifies and substitutes the most probable intended English word (unless the raw word contains a digit, in which case it is returned unmodified). All six models and this pipeline were integrated into a single interactive Gradio web application with three tabs: single image/camera classification, multiple images to word, and side-by-side model comparison.

7.7 Evaluation Metrics

The six models were evaluated using accuracy, weighted precision/recall/F1-score, macro-averaged F1-score, total and per-image inference time, model size, and parameter count. Inference timing used one warm-up pass followed by three repeated full-test-set measurements on the same NVIDIA A100-SXM4-40GB GPU. Mean latency and its standard deviation were recorded; because three repetitions provide only a preliminary estimate and some models showed substantial timing variability, the speed results are interpreted as environment-specific rather than universal hardware-independent rankings.

The following equations correspond directly to the scikit-learn calculations used in the evaluation scripts. For class c , TP_c , FP_c , and FN_c denote true-positive, false-positive, and false-negative counts.

$$\text{Accuracy} = (\sum_c TP_c) / N \quad (9)$$

$$\text{Precision}_c = TP_c / (TP_c + FP_c) \quad (10)$$

$$\text{Recall}_c = TP_c / (TP_c + FN_c) \quad (11)$$

$$F1_c = 2(\text{Precision}_c \times \text{Recall}_c) / (\text{Precision}_c + \text{Recall}_c) \quad (12)$$

The code reports weighted precision, weighted recall, and weighted F1-score using the support n_c of each class:

$$\text{Weighted Metric} = \sum_c (n_c / N) \text{Metric}_c \quad (13)$$

It also saves the macro F1-score, which assigns equal weight to every class:

$$\text{Macro F1} = (1/36) \sum_{c=1}^{36} F1_c \quad (14)$$

Inference timing is also calculated directly in the scripts from three repeated full-test-set measurements:

$$\bar{T} = (1/R) \sum_{r=1}^R T_r, \quad R = 3 \quad (15)$$

$$T_{\text{image}} = \bar{T} / N_{\text{test}}, \quad N_{\text{test}} = 504 \quad (16)$$

$$\sigma_T = \sqrt{[(1/R) \sum_{r=1}^R (T_r - \bar{T})^2]} \quad (17)$$

These equations match the implemented NumPy mean and standard-deviation calculations and the division of average total time by the 504 test images.

8. Results

8.1 Overall Model Performance Comparison

Table 5 summarizes the performance of all six models on the identical 504-image test set. ResNet50 reached 92.46% test accuracy (weighted F1 92.42%); MobileNetV2, despite roughly a tenth of ResNet50's parameters, achieved a broadly comparable 91.87% (weighted F1 91.93%); EfficientNetB0 recorded the lowest accuracy of the six at 89.88% (weighted F1 89.51%). The Vision Transformer reached 98.81% accuracy (weighted F1 98.81%), substantially higher than any CNN. The Swin Transformer improved further to 99.60% accuracy (weighted F1 99.60%). The proposed Hybrid MobileNetV2-Swin model achieved the best overall performance: 99.80% accuracy and a 99.80% weighted F1-score, with only a single misclassified image across the entire 504-image test set.

Table (5): Performance comparison of the six models on the test set (504 images).

Model	Accuracy	Weighted Precision	Weighted Recall	Weighted F1	Macro F1
Hybrid MobileNetV2-Swin	99.80%	99.81%	99.80%	99.80%	99.80%
Swin Transformer	99.60%	99.60%	99.60%	99.60%	99.60%
Vision Transformer	98.81%	98.83%	98.81%	98.81%	98.81%
ResNet50	92.46%	93.36%	92.46%	92.42%	92.42%

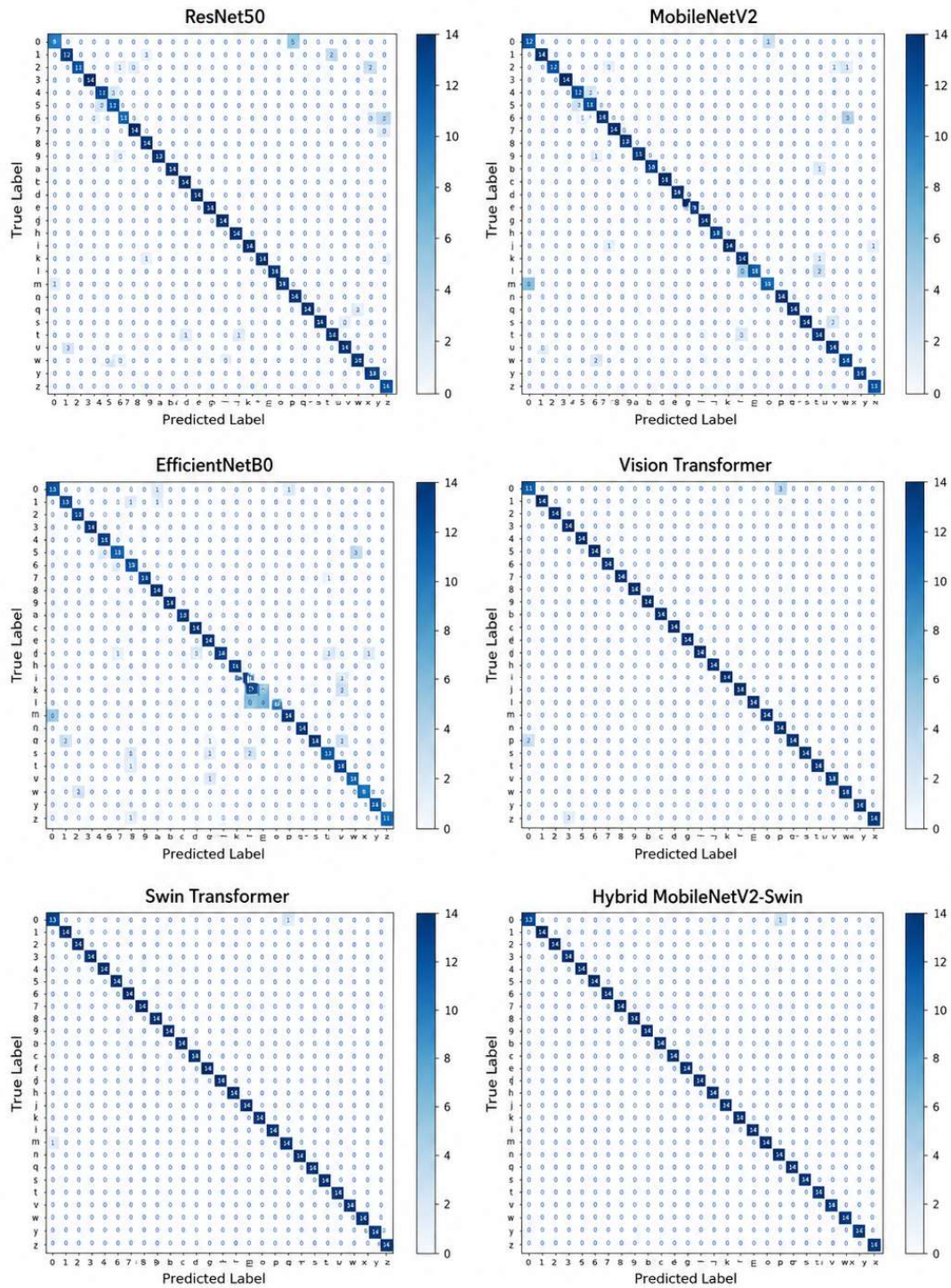
Model	Accuracy	Weighted Precision	Weighted Recall	Weighted F1	Macro F1
MobileNetV2	91.87%	92.86%	91.87%	91.93%	91.93%
EfficientNetB0	89.88%	91.42%	89.88%	89.51%	89.51%

The three transformer-based models (Hybrid, Swin, ViT) form a clearly separated top tier, each exceeding 98.8% accuracy, while the three CNN baselines form a second tier between 89.9% and 92.5% — a clear six-to-eight-percentage-point gap below the weakest transformer-based model.

8.2 Confusion Matrix and Per-Class Analysis

Per-class analysis (Table 6) shows that the digit '0' and the letter 'o' — whose ASL handshapes form a nearly identical closed circle — are the most commonly confused pair across every model, and essentially the only source of error in the three transformer-based models. The Hybrid model misclassified only one image overall (a true '0' predicted as 'o'); the Swin Transformer made two such errors; the Vision Transformer's confusion was bidirectional (three '0'→'o' and two 'o'→'0' errors, plus one isolated '6'→'w' error). The three CNN models showed markedly broader confusion beyond this pair: ResNet50 additionally confused the digit pair 4↔5 and showed low recall for the letter 'n' (0.5714); MobileNetV2's most severe confusion was directly between '0' and 'o' (six of fourteen true 'o' images misclassified); and EfficientNetB0 showed the broadest weakness, with the letters 'n' and 'v' each achieving a recall of only 0.4286.

Figure 5 presents the complete confusion matrices for all six models, enabling direct visual comparison of class-level error concentration and the progressively stronger diagonal dominance of the transformer-based approaches.



Confusion matrices of the evaluated ASL recognition models.

Figure (5): Confusion matrices of the six evaluated ASL recognition models.

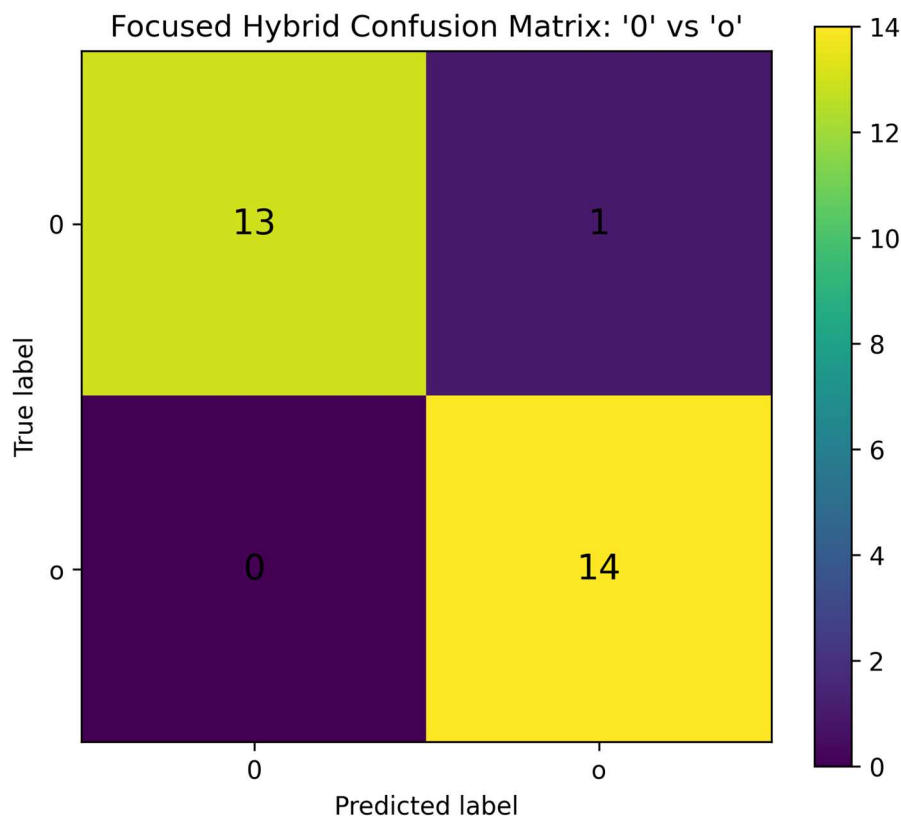


Figure (6): Focused confusion matrix for the Hybrid model showing the residual confusion between the digit '0' and the letter 'o'.

Table (6): Per-class F1-score for the digit '0' and letter 'o', and overall error spread, by model.

Model	F1 ('0')	F1 ('o')	Classes with F1 < 0.95	Worst-Performing Class(es)
Hybrid MobileNetV2-Swin	0.9630	0.9655	0	'0' (0.9630)
Swin Transformer	0.9286	0.9286	2	'0' and 'o' (tied)
Vision Transformer	0.8148	0.8276	2	'0' (0.8148)
ResNet50	0.7500	0.8125	15	'n' (0.7273)
MobileNetV2	0.7500	0.6667	17	'o' (0.6667)
EfficientNetB0	0.7879	0.6957	18	'n' and 'v' (tied)

8.3 Model Efficiency and Accuracy–Speed Trade-off

Table 7 reports model size, parameter count, and inference time. On the evaluation environment used in this study, the Swin Transformer produced the lowest mean total inference time (1.608 s; SD = 0.031 s) and per-image latency (3.191 ms). The Hybrid model required 2.513 s (SD = 1.122 s; 4.986 ms/image), while the Vision Transformer required 3.852 s and showed high timing variability (SD = 3.124 s; 7.642 ms/image). The three CNN models fell between 1.766 and 2.289 s. These results suggest that Swin was the fastest in this specific implementation and hardware environment; however, the relatively large standard deviations

for Hybrid and ViT mean that stronger deployment claims require more repetitions, batch-size-one testing, and measurement of model-forward time separately from data-loading overhead.

Table (7): Model size, parameter count, and inference-time comparison (sorted by total inference time, ascending).

Model	Parameters	Model Size (MB)	Total Inference Time (s)	Time per Image (ms)	Inference SD (s)
Swin Transformer	27.55 M	105.15	1.608	3.191	0.031
EfficientNetB0	4.39 M	20.11	1.766	3.504	0.105
MobileNetV2	2.60 M	13.03	1.814	3.6	0.081
ResNet50	24.12 M	96.72	2.289	4.542	0.503
Hybrid MobileNetV2-Swin	30.81 M	117.86	2.513	4.986	1.122
Vision Transformer	85.83 M	327.46	3.852	7.642	3.124

The accuracy-speed analysis indicates that the Swin Transformer offered the strongest balance in the tested environment: it was the second most accurate and had the lowest measured mean latency. The Hybrid model was the most accurate but not the fastest, making it preferable when maximum predictive performance is prioritized. MobileNetV2 remained the smallest model and is relevant to storage-constrained deployment. These conclusions are conditional on the current software stack, GPU, batch processing, and three-repeat timing procedure; they should not be generalized to mobile, CPU, or real-time camera deployment without additional benchmarking.

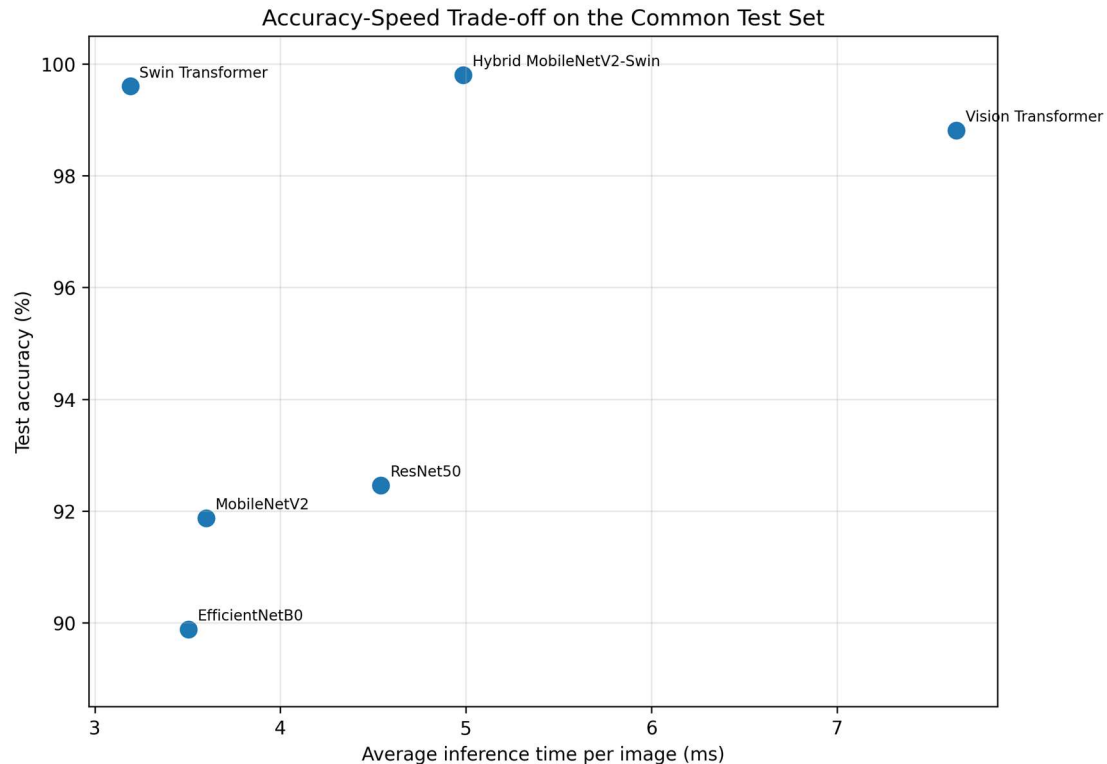


Figure (7): Accuracy-speed trade-off for the six evaluated models on the common test set.

8.4 Practical Application Demonstration

Finally, the Gradio demo that supports the project visually shows how effectively the three best models can be used for recognizing hand-sign language through finger spellings. Even if a letter is sometimes misread, as a rule, this will be corrected by the word the user is most probably going to mean, as long as the other letters form an unambiguous English word. The interactive Gradio interface allows either file upload or webcam capture as sources of input. Among all six models, users can pick their favorite one. The interface will also show top class probabilities that the model comes with predicted-character confidence levels, letter by letter. It features word creation in letter order and word creation with several images ordered in the correct way, then spelling correction as well as a saved model comparison table.

The figures 8 shows the panel where the user can choose the model and also see the performance summary of the interface and the tab for single-image inputs. The figure 9 gives examples captured during a live application session and includes both webcams based one-letter prediction and file upload based multi-letter word formation. The Gradio-based application helps the experimental models to a more interactive way. In the single-image mode, the user decides a model, uploads an image, or uses his webcam to capture a sign. Then the app handles the model's preprocessing, the user gets the five class probabilities, the predicted character, and the character's confidence level as the output.

If a user works with the multiple-image mode then he can, for getting back a correct word, go through each handshape and letter prediction. The individual letter predictions get displayed together in a gallery while the letters are glued together to make the raw word. Then, the correction of alpha-only words is made using TextBlob. A third tab shows the model's saved accuracy precision recall, F1-score latency model size and parameter counts. Although the interface is mainly for research it is quite accessible. Yet, it doesn't provide a number of the hand functionalities yet including hand detection, segmentation to separate hands, low-confidence rejection, or signer-independent calibration. These features might come up as new deployments.

(a) Model selection and performance-summary panel

ASL Multi-Model Recognition System
American Sign Language Classification, Word Formation, and Spelling Correction

Select one of the trained models, upload an ASL image or use the camera, and view the predicted letter with its confidence. You may also combine several predictions into a word and apply automatic English spelling correction.

Select Recognition Model:

Hybrid MobileNetV2-Swin
 ResNet50
 MobileNetV2
 EfficientNetB0
 Vision Transformer
 Swin Transformer
 ✓ Hybrid MobileNetV2-Swin

Application Status

Hybrid MobileNetV2-Swin selected. The model will load when prediction starts.

Test Accuracy	99.80%
Weighted Precision	99.81%
Weighted Recall	99.80%
Weighted F1-Score	99.80%
Test Loss	0.0112
Time per Image	4.986 ms
Parameters	30,810,782
Model Size	117.86 MB
Evaluation Device	NVIDIA A100-SXM4-40GB

Test Accuracy is the model's performance on the complete test dataset. Prediction Confidence is the confidence for the currently uploaded image.

(b) Single-image / camera input tab (idle state)

Single Image / Camera

Multiple Images to Word

Models Comparison

Upload an ASL Image or Use Camera

Click to Access Webcam

Top Five Predictions

Predicted Letter

Prediction Confidence

Predict Image

Word Formation

Raw Predicted Word

Corrected Word

Append Predicted Letter

Remove Last Letter

Correct Word

Clear Word

Figure (8): Screenshots of the Gradio-based ASL recognition and word-formation application: (a) the model-selection and performance-summary panel, and (b) the single-image/camera input tab in its idle state.

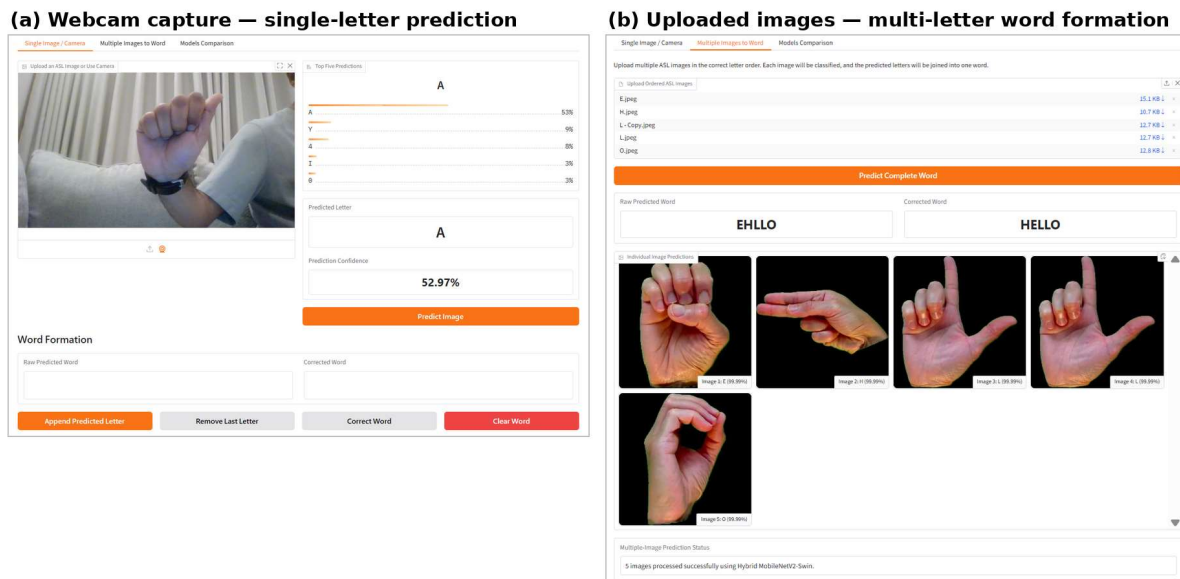


Figure (9): Representative prediction outputs from the deployed application: (a) a single-letter prediction obtained via live webcam capture, and (b) multi-letter word formation ('HELLO') obtained from an ordered sequence of uploaded images.

All the results mentioned above practically speak out for themselves and have implications that go beyond just accuracy ranking six models, to something deeper - the architecture choice for a real-world assistive system should depend upon the deployment context and not accuracy alone. If a kiosk application has the means of a capable GPU and no tight latency budget then the marginal accuracy benefit of the Hybrid model can reasonably be its top priority. But a live captioning service being part of a video call is much more suited to the Swin Transformer's near-top accuracy and lowest-latency combination among all six models. The Vision Transformer here is not in the top rank, and its position in the competition is quite clearly the slowest and the biggest. This is an independent warning finding because adding self-attention to ViT on its own (without the window-based efficiency mechanism used by Swin Transformer) doesn't guarantee the desirable level of accuracy relative to cost.

8.5 Model Performance Comparison and Interpretation

The Hybrid system obtained the top level of accuracy because two branches of its architecture capture different components of the hand gesture of one. MobileNetV2 branch is focused on localized edge, texture details, while the branch based on Swin focuses on relations between hand parts. That means, the final classifier is capable of dealing with the ambiguity of one of the individual representations of the hand gesture. The improvement of the Hybrid over the Swin is one correct test image and so it's insignificant, but such a result will Yes help to detect the error pattern.

Swin Transformer was a winner for speed-accuracy trade-off. Its shifted-window technique restricts most of the attention calculation to the vicinity of the window and because of this it avoids a global attention's fully quadratic computation cost. ViT was a very accurate model and at the same time the largest and slowest one, which fits very much with their global patch-attention structure and the large number of parameters. In some instances, CNNs were smaller than the Transformers but they were definitely much less precise. The interpretation of this finding calls for care: CNNs with ImageNet pre-trained weights were kept unchanged whereas the other models, transformer-based, were fully fine-tuned. Because of this, this experiment

compares different transfer-learning strategies, rather than model design only. An ablation study of the next stage can release topmost CNN layers, apply a reduced learning rate, and thereby figure out what is left of the performance difference if the depth of fine-tuning is the same.

8.6 Misclassification Between Digit 0 And Letter o

The Hybrid model's confusion matrix only has one leftover misclassification: a digit 0 was identified as a letter 'o'. Since both classes appear in the data as a closed rounded hand with different small variations in hand orientation, wrist position, the thumb location, hand image framing or the dataset acquisition method (capture), the identification is not straightforward. So, the hybrid error rate is defined entirely by this pair, since all other test images were correctly predicted. At the application level, such a substitution can alter a word or alphanumeric code. Spelling correction may repair an alphabetic word when o is produced in place of another letter, but it cannot reliably determine whether a visually ambiguous sign was intended as the number 0 or the letter o. A production system should therefore display confidence and alternatives, apply contextual language modeling, and request a repeated sign when the confidence margin between these two classes is small.

9. Conclusion and Discussion of Results

This study compared six deep learning architectures for American Sign Language fingerspelling recognition and found that transformer-based and hybrid attention-augmented architectures offer a genuine advantage over conventional CNNs on this task, even with a comparatively small dataset (2,515 images). The 99.80% accuracy achieved by the proposed Hybrid model is close to the 99.97% reported by the hybrid CNN–Vision Transformer model of Aly and Fathi (2025), and consistent with the strong results reported for the Swin Transformer architecture across other sign-language datasets, despite this study's considerably smaller dataset (2,515 images versus 87,000). The finding that fusing two feature extractors improves accuracy is also consistent with the two-stream CNN literature (Ma et al., 2022; Zhidebayeva et al., 2024), extending it to a new fusion axis — a CNN paired with a hierarchical transformer rather than two temporal frames or two CNN backbones. Conversely, the CNN accuracies obtained here (92.46%, 91.87%, 89.88%) fall short of the near-perfect accuracies reported by CNN-only studies trained on much larger datasets, such as the 99.99% accuracy reported for ResNet50 by Marjusalinah et al. (2021) on a dataset over thirty times larger — suggesting that the relatively modest dataset size constrained the CNN models' generalization more than it constrained the transformer-based models.

9.1 Comparison with Previous Studies

The findings are consistent with recent studies reporting strong performance from attention-based and hybrid architectures in sign-language and image-classification tasks. Earlier CNN-based ASL studies established that transfer learning can achieve high accuracy on controlled datasets, but many evaluated a single architecture or did not report latency and model size. The present work extends that line of research by placing CNN, ViT, Swin, and feature-fusion approaches in one reproducible comparison.

The result also parallels the methodological logic of recent hybrid studies, including Swin-PSO-SVM, in which a transformer representation is combined with an additional optimization or classification component. In the present study, however, the contribution is featuring fusion between a lightweight CNN and a hierarchical transformer, followed by end-to-end neural classification and deployment in an interactive word-formation application. Direct numerical

comparisons with other publications should be avoided because datasets, class counts, splits, augmentation policies, and signer conditions differ.

9.2 Limitations

The first limitation is dataset scale: 2,515 image may be enough for a controlled benchmark but not for broad real-world generalization. Next, the split is based on images not signer-independent so visual aspects related to a signer or recording setup may be found in all subsets. Then, the scope of the sign task has been limited to static isolated signs and does not account for continuous signing, co-articulation, facial expression, or body motion. Also, the static forms of J and Z are not able to fully represent the dynamic ASL movements of these signs. As a fifth point, CNN and transformer teams were following different fine-tuning methods and setups. Next reason, the inference duration was measured only three times in one cloud-GPU computing setting, Because of this resulting in large differences for some models. Also, the product was showcased for its functionality but was not subjected to a formal usability test with Deaf or hard-of-hearing people. Last but definitely not the least, the TextBlob correction is based on dictionaries and will at times wrongfully modify proper names, abbreviations, or terms specific to a field.

9.3 Future Work

The first thing that future studies should do is to gather signers-variety data set bigger in size and implement signer-independent cross-validation. A hand-detection and segmentation stage like MediaPipe Hands should be adopted for isolating the relevant area and making the system less sensitive to the background. Signs in action and long-lasting sequences should be dealt through video, 3D convolution, recurrent network, temporal transformer, or landmark-based sequence model.

Also, new researches could involve conducting fine-tuning as matched and running ablation studies, calibration of confidence, rejection of uncertain inputs, Grad-CAM for CNN branches, attention visualization for transformer branches, and running repeated latency experiments with single batch on CPU, mobile and edge devices. The formation of the final words should also be tested through error rate of character, accuracy of word and accuracy of correction after which an evaluation of accessibility through users' perspectives should follow.

Vision Transformer's significant, consistent lead over CNNs, Swin Transformer, and the Hybrid model can probably be traced to a real architectural feature difference self-attention: using self-attention a network can learn how to give content-dependent importance to different hand regions, which seems Mostly useful in a very fine discrimination setting like recognizing the difference between the numbers zero and the letter 'o'. Also, a better standalone Swin Transformer result was furthered by the Hybrid model which can suggest very well that the MobileNetV2 features which localize are quite different from the global Swin features resulting from self-attention of the SwinTransformer so their combination is not redundant but rather complementary. This study includes many shortcomings, for example, static images only were dealt with and continuous signing or word-level recognition were not tackled. Images from similar acquisition conditions may therefore occur across training and test subsets, potentially inflating generalization estimates. Fourth, the spelling-correction stage used TextBlob's general-purpose English dictionary and was not systematically evaluated on a dedicated noisy-word benchmark. Finally, no advanced hyperparameter search was performed, and the CNN/transformer fine-tuning strategies differed as described above.

A further point is that the observed gap between the CNN and transformer-based groups cannot be attributed to architecture alone. The three CNN baselines used frozen ImageNet-pretrained backbones, so only their newly added classification heads learned ASL-specific features, whereas the ViT, Swin, and Hybrid models were fully fine-tuned end-to-end. This asymmetry likely contributed to the broader CNN confusion patterns and may have amplified the apparent transformer advantage. Accordingly, the current findings should be interpreted as a comparison of the implemented training pipelines, not as a definitive isolation of architectural effects. A follow-up ablation should partially and fully unfreeze each CNN backbone, use a lower fine-tuning learning rate, and compare frozen versus end-to-end CNN results under the same split.

In summary, this study provides a reproducible six-model comparison using a common dataset split, input resolution, training budget, evaluation set, and reporting framework. It compares three frozen-backbone CNN pipelines with fully fine-tuned ViT, Swin, and Hybrid pipelines, presents an environment-specific accuracy-speed analysis, and demonstrates multi-letter word formation with automatic spelling correction. The methodological weaknesses mentioned above provide the clear roadmap for follow-up studies with a strong focus on signer-independent assessment, fully fine-tuning CNN, recognition of dynamic J/Z, and more accurate latency test.

10. Recommendations

Several practical recommendations are based on these results: In learning situations, the Swin Transformer - which delivers among the six models the highest accuracy and the fastest inference time - could be embedded to an interactive device which teaches the ASL alphabet, giving right after the sign a hint, a light to guide to know if the sign was correct.

In public services, the proposed Hybrid model, as the most accurate standalone system, could be used for self-service kiosks in government or healthcare facilities where accuracy is prioritized over inference speed. In a mobile or embedded environment, MobileNetV2 would still be the better compromise as for its much smaller file size and low computational cost, even when the accuracy loss is a little over 6 percent compared to those of the transformer models.

A great application is the video-conferencing apps like Zoom, Microsoft Teams, and Google Meet. Next research is recommended to 1) collect the dataset that is larger, more diverse and has signer labels, report the results of evaluation based on signer-independent;

2) redo the tests of CNN by freezing and unfreezing the backbone partially and fully so that the effect of architecture could be dissociated from that of fine-tuning depth; 3) measure inference time at least twenty times, then give the mean value, the median and the standard deviation. At the same time, check that if a batch is of size 1 the model-forward time is separated from that of data loading;

4) expand the program to include not just static signs but also the recognition of dynamic J/Z as well as signing continuously; 5) assess correction based on spelling with word accuracy and character error rate for a special noisy-sequence benchmark;

6) study quantization, pruning and knowledge distillation; 7) add the visualizations which help explain AI like Grad-CAM and attention maps in particular with the case of residual 0/o confusion.

10.1 Deployment Priorities A great GPU-powered system would pick Hybrid as it is the model with the highest precision at the cost of speed.

For usage on cloud or desktop, when the time to get a result is very important, the Swin Transformer brings the best combination between getting a correct result quickly. This model is one of the best candidates on this. It is known to deliver high accuracy and very low inference time among all the six models. This can be Mostly helpful for developing an app which teaches the users the letters in the American Sign Language using the Swin Transformer model. Such an app could give users real-time feedback, indicating whether a sign made by the user is correct or otherwise. This can be a great tool for learners of the language. In other scenarios, like public services which would be the most accurate but at the expense of speed. Hybrid could be used for instance when it comes to self-service kiosks at hospitals or government buildings. The Swin Transformer seems a strong contender among accuracy-latency tradeoffs as per the current experiments. As the size of a model is very important in memory-limited situations or for mobile use, the MobileNetV2 is the only choice as it calls for very few computing resources and is extremely compact. But besides Truth is its accuracy is not as good as that of transformer-based models - the difference is approximately six-eight percent, it still offers certain advantages in particular for mobile users.

Deploying MobileNetV2 after further fine-tuning together with confidence-based rejection is definitely recommended. The real version of any of those technologies should be built keeping in view privacy-preserving processing local on device, camera input consent by user being explicitly sought, doing accessibility test with the audience, the target community being aware of the limitations via display at least that part with low confidence level in the form as well as the alternative predictions. The prototype of the system should never take the place of interpreting by a professional interpreter in situations where making a mistake in communication might cause a lawsuit, health, or safety-related issue.

Data Availability

The dataset of ASL fingerspelling, model's weights trained, and evaluation scripts that were used to get the results of the study can be obtained from the corresponding author, if requested following the reasonable grounds.

Conflict of Interest

All authors state that there are no conflicts of interest on their behalf.

Ethical Approval

Not applicable. During this study, neither human nor animal subjects were involved; instead, the dataset consisted of images of handshapes for static ASL fingerspelling.

References

- 26 Aggarwal, J., Gupta, L., Gupta, M., Nitin, & Girdhar, P. (2023). Real-time sign language recognition using CNNs. *Advances and Applications in Mathematical Sciences*, 22(11), 2199–2211.
- 2 Alsharif, B., Altaher, A. S., Altaher, A., Ilyas, M., & Alalwany, E. (2023). Deep learning technology to recognize American Sign Language alphabet. *Sensors*, 23(18), 7970. <https://doi.org/10.3390/s23187970>
- Aly, M., & Fathi, I. S. (2025). Recognizing American Sign Language gestures efficiently and accurately using a hybrid transformer model. *Scientific Reports*, 15, 20253. <https://doi.org/10.1038/s41598-025-06344-8>
- 1 Applying Swin architecture to diverse sign language datasets. (2024). *Electronics*, 13(8), 1509. <https://doi.org/10.3390/electronics13081509>
- 22 Bankar, S., Kadam, T., Korhale, V., & Kulkarni, A. A. (2022). Real time sign language recognition using deep learning. *International Research Journal of Engineering and Technology*, 9(4).
- 21 Bayrak, S., Nabiyeve, V., & Atalar, C. (2024). American Sign Language recognition model using complex Zernike moments and complex-valued deep neural networks. *IEEE Access*, 12, 193001–193014.
- 3 Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/2010.11929>
- 48 Drishya. (2023). Sign language recognition. *International Research Journal of Modernization in Engineering Technology and Science*, 5(4).
- 10 He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- 31 Hurroo, M., & Walizad, M. E. (2020). Sign language recognition system using convolutional neural network and computer vision. *International Journal of Engineering Research & Technology*, 9(12).
- 29 Kamruzzaman, M. M. (2020). Arabic Sign Language recognition and generating Arabic speech using convolutional neural network. *Wireless Communications and Mobile Computing*, 2020, 3685614.
- 2 Kasapbaşı, A., Elbushra, A. E. A., Al-Hardanee, O., & Yilmaz, A. (2022). DeepASLR: A CNN based human computer interface for American Sign Language recognition for hearing-impaired individuals. *Computer Methods and Programs in Biomedicine Update*, 2, 100048. <https://doi.org/10.1016/j.cmpbup.2021.100048>
- 19 Khan, A., Sohail, A., Zahoor, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8), 5455–5516.
- 7 Kouvakis, V., Trevlakakis, S. E., & Boulogeorgos, A. A. A. (2024). Semantic communications for image-based sign language transmission. *IEEE Open Journal of the Communications Society*, 5, 1083–1100.
- 42 LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- 86 Li, Y., Zhang, K., & Li, K. (2021). Deep learning for image classification: A survey. *Neurocomputing*, 449, 20–36.

- 6 Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030. <https://arxiv.org/abs/2103.14030>
- 18 Ma, Y., Xu, T., & Kim, K. (2022). Two-stream mixed convolutional neural network for American Sign Language recognition. *Sensors*, 22(16), 5959. <https://doi.org/10.3390/s22165959>
- 7 Marjusalinah, A. D., Samsuryadi, S., & Buchari, M. A. (2021). Classification of finger spelling American Sign Language using convolutional neural network. *Computer Engineering and Applications*, 10(2), 91–102.
- 13 Mujahid, A., Awan, M. J., Yasin, A., Mohammed, M. A., Damaševičius, R., Maskeliūnas, R., & Abdulkareem, K. H. (2021). Real-time hand gesture recognition based on deep learning YOLOv3 model. *Applied Sciences*, 11(9), 4164.
- 50 Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- 20 Saleh, H., Mostafa, S., Ismail, A. H., Srivastava, S., & Alsamhi, S. H. (2024). Swin-PSO-SVM: A novel hybrid model for monkeypox early detection. *IEEE Access*, 12, 187367–187385. <https://doi.org/10.1109/ACCESS.2024.3513818>
- 8 Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>
- 33 Sharma, S. (2020). Vision-based sign language recognition system: A comprehensive review. *Proceedings of the Fifth International Conference on Inventive Computation Technologies (ICICT)*.
- 15 Sheth, P., Dedhia, R., Chheda, A., & Sawant, V. (2023). American Sign Language recognition and generation: A CNN-based approach. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(4), 241–250.
- 16 Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.
- 12 Szeliski, R. (2022). *Computer Vision: Algorithms and Applications* (2nd ed.). Springer.
- 46 Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 36th International Conference on Machine Learning*, 6105–6114. PMLR. <https://arxiv.org/abs/1905.11946>
- 43 World Health Organization. (2021). World report on hearing. WHO. <https://www.who.int/publications/i/item/9789240020481>
- 110 Yadav, B. K., & Bohra, H. (2021). Sign Language to Text and Speech Conversion. *International Journal of Advanced Research, Ideas and Innovations in Technology*, 7(3), V7I3-1560.
- 53 Zakariah, M., Alotaibi, Y. A., Koundal, D., Guo, Y., & Mamun Elahi, M. (2022). Sign language recognition for Arabic alphabets using transfer learning technique. *Computational Intelligence and Neuroscience*, 2022, 4567989.
- 11 Zhidebayeva, A., et al. (2024). Real-time sign language fingerspelling recognition system using 2D deep CNN with two-stream feature extraction approach. *International Journal of Advanced Computer Science and Applications*, 15(9).
- 5