

محمد القثمي

A Comparative Study of Deep Learning Models for American Sign Language Fingerspelling Recognition %28Thesis%29.docx

 Midocean University

Document Details

Submission ID

trn:oid:::2945:402290206

Submission Date

Jul 27, 2026, 9:05 PM GMT+3

Download Date

Aug 11, 2026, 12:55 PM GMT+3

File Name

A Comparative Study of Deep Learning Models for American Sign Language Fingerspelling Reco....docx

File Size

4.8 MB

57 Pages

14,416 Words

85,147 Characters

13% Overall Similarity

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

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

-  **173 Not Cited or Quoted 11%**
Matches with neither in-text citation nor quotation marks
-  **27 Missing Quotations 2%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 6%  Internet sources
- 7%  Publications
- 11%  Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

-  **Hidden Text**
434 suspect characters on 9 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

- 173 Not Cited or Quoted 11%**
Matches with neither in-text citation nor quotation marks
- 27 Missing Quotations 2%**
Matches that are still very similar to source material
- 0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 6% Internet sources
- 7% Publications
- 11% 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
Technical University of Košice on 2026-06-03 <1%
- 3** Submitted works
University of Hertfordshire on 2023-09-18 <1%
- 4** Internet
ebin.pub <1%
- 5** Submitted works
University of Lincoln on 2025-08-26 <1%
- 6** Internet
mdpi-res.com <1%
- 7** Internet
thesis.unipd.it <1%
- 8** Submitted works
Midocean University on 2025-09-05 <1%
- 9** Publication
Poonam Nandal, Tapas Kumar, Meeta Singh, Mamta Dahiya. "Progressive Compu... <1%
- 10** Submitted works
ICTS on 2026-06-21 <1%

11	Publication	El houssine Amraouy, Ali Yahyaouy, Sanaa Faquir, Hicham Chaoui, Hamid Gualou...	<1%
12	Publication	Ghulfam Hussain. "Prostate Cancer Detection in Bi-parametric MRI Using Deep Le...	<1%
13	Internet	dspace.daffodilvarsity.edu.bd:8080	<1%
14	Internet	www.pmu.edu.sa	<1%
15	Submitted works	Abo Akademi University on 2026-05-24	<1%
16	Submitted works	RDI Distance Learning on 2026-01-14	<1%
17	Submitted works	2U Simmons University-MSW on 2025-09-21	<1%
18	Submitted works	Aston University on 2026-01-29	<1%
19	Submitted works	Green University Of Bangladesh on 2026-01-17	<1%
20	Publication	B. G. Nagaraja, S. Kannadhasan. "Information and Communication Systems", CRC...	<1%
21	Submitted works	College of Public Health on 2026-03-16	<1%
22	Submitted works	Mar Athanasius College of Engineering on 2025-10-15	<1%
23	Publication	Lafrance, Samantha. "A Multi-Scale Fusion CNN-ViT Hybrid Architecture for Semic...	<1%
24	Submitted works	University of Engineering and Technology ORIC on 2026-06-24	<1%

25	Submitted works	Faculty of Computer Science & Information Technology on 2026-07-06	<1%
26	Submitted works	British University in Egypt on 2026-06-03	<1%
27	Submitted works	Coventry University on 2025-12-04	<1%
28	Submitted works	University of Economics, Prague on 2026-05-13	<1%
29	Internet	www.informatica.si	<1%
30	Submitted works	Brunel University on 2026-05-01	<1%
31	Publication	Hassan, Alla Ahmad. "Automatic COVID-19 Detection Using Machine Learning Tec...	<1%
32	Submitted works	Universiti Teknologi MARA on 2026-07-19	<1%
33	Submitted works	University of New York Tirana on 2026-07-19	<1%
34	Internet	koreascience.kr	<1%
35	Internet	123dok.net	<1%
36	Internet	cgspace.cgiar.org	<1%
37	Internet	pmc.ncbi.nlm.nih.gov	<1%
38	Internet	repository.upi.edu	<1%

39	Internet	studentsrepo.um.edu.my	<1%
40	Internet	www.scpe.org	<1%
41	Internet	lib.iium.edu.my	<1%
42	Publication	Mohammad Qatawnah, Khaled Hamad, Suhaila Abuowaida, Nawaf Alshdaifat, M...	<1%
43	Submitted works	Technological University Dublin on 2025-09-01	<1%
44	Submitted works	UCSI KL Campus on 2026-05-04	<1%
45	Internet	file.techscience.com	<1%
46	Submitted works	Midlands State University on 2025-05-28	<1%
47	Internet	arxiv.org	<1%
48	Internet	dspace.univ-tlemcen.dz	<1%
49	Internet	oulurepo.oulu.fi	<1%
50	Internet	vtechworks.lib.vt.edu	<1%
51	Publication	Alsharif, Bader. "Advances in Real-Time American Sign Language Recognition Sys...	<1%
52	Publication	Kumar, Dheeraj. "Automated Detection and Classification of Cholelithiasis from U...	<1%

53	Submitted works	National Institute of Business Management Sri Lanka on 2025-04-25	<1%
54	Submitted works	Universidade do Porto on 2023-02-05	<1%
55	Submitted works	University of Ulster on 2026-05-01	<1%
56	Submitted works	Australian National University on 2025-10-24	<1%
57	Submitted works	KoE on 2026-06-12	<1%
58	Submitted works	University of Hertfordshire on 2023-09-18	<1%
59	Submitted works	University of Wales Institute, Cardiff on 2018-01-25	<1%
60	Publication	V. Sekhar, P. Ajay Kumar Reddy, K. Logesh, T. Raghunadha Reddy, Dara Raju. "Ne...	<1%
61	Internet	github.com	<1%
62	Internet	smartech.gatech.edu	<1%
63	Internet	www.atlantis-press.com	<1%
64	Internet	www.coolpo.io	<1%
65	Internet	www.preprints.org	<1%
66	Publication	Alam, Mohammad Mahbub. "Towards Precise Localization in Object Detection: St...	<1%

67	Publication	Bhabani Prasanna Pattanaik, Dharendra Kumar Shukla, Sambit Satpathy, Kamala...	<1%
68	Submitted works	Napier University on 2025-08-20	<1%
69	Publication	Rupali H. Shende, Nuzhat F. Shaikh. "Detection and Classification of American Sig...	<1%
70	Internet	palashroy975.github.io	<1%
71	Internet	www.americaspg.com	<1%
72	Internet	www.epj-conferences.org	<1%
73	Publication	"Information Systems for Intelligent Systems", Springer Science and Business Me...	<1%
74	Submitted works	Birzeit University Main Library on 2026-05-30	<1%
75	Submitted works	Copenhagen Business School on 2026-05-15	<1%
76	Submitted works	Far Eastern University on 2022-07-04	<1%
77	Submitted works	National University of Ireland, Galway on 2026-03-29	<1%
78	Submitted works	Palestine Technical University Kadoorie on 2026-05-17	<1%
79	Submitted works	University of Huddersfield on 2025-04-23	<1%
80	Submitted works	WHU - Otto Beisheim School of Management on 2026-06-04	<1%

81	Internet	encyclopedia.pub	<1%
82	Internet	eprints.usm.my	<1%
83	Internet	fdocuments.us	<1%
84	Submitted works	universiteknologimara on 2026-07-15	<1%
85	Internet	unsworks.unsw.edu.au	<1%
86	Internet	upcommons.upc.edu	<1%
87	Internet	uvadoc.uva.es	<1%
88	Submitted works	AUT University on 2024-06-05	<1%
89	Publication	Aishagathrh, Fahad Muflih. "Advancing Non-Alcoholic Fatty Liver Disease Diagnos...	<1%
90	Publication	Asali, Ehsan. "Spatiotemporal Multimodal Representation Learning for Activity U...	<1%
91	Publication	Chakravarthy, Kishor. "DART: Domain Adaptive Robust Transfer Learning for Obje...	<1%
92	Submitted works	Federation University on 2026-06-08	<1%
93	Submitted works	Higher Education Commission Pakistan on 2012-04-04	<1%
94	Submitted works	Imam Abdulrahman Bin Faisal University on 2026-06-03	<1%

95	Submitted works	King Abdullah University of Science and Technology (KAUST) on 2026-05-12	<1%
96	Publication	Kosaraju, Naren Kumar. "Classification of Solidago Species Using CNN Architectur...	<1%
97	Submitted works	La Trobe University on 2011-05-23	<1%
98	Submitted works	Liverpool John Moores University on 2023-09-05	<1%
99	Submitted works	Midocean University on 2026-05-02	<1%
100	Publication	Paola Zuccolotto, Marica Manisera, Marco Sandri, Dean Oliver, Sergio Scariolo. "A...	<1%
101	Submitted works	Rhodes University on 2019-12-01	<1%
102	Publication	Stein, Kyle Richard. "Integrating Transformers for Cyber Defense Under Unseen a...	<1%
103	Submitted works	Teesside University (Blackboard Ultra) on 2026-05-01	<1%
104	Submitted works	Texas A&M University, College Station on 2026-03-17	<1%
105	Submitted works	The Hashemite University on 2026-06-24	<1%
106	Submitted works	Thesis 2026 on 2026-01-16	<1%
107	Submitted works	Universiti Teknologi Malaysia on 2026-07-16	<1%
108	Submitted works	University of Makati on 2023-05-21	<1%

109	Submitted works	University of Sheffield on 2025-08-27	<1%
110	Submitted works	University of South Florida on 2026-07-14	<1%
111	Publication	Valentin Bercaru, Nirvana Popescu. "Exploring Sign Language Dataset Augmenta...	<1%
112	Submitted works	Vrije Universiteit Amsterdam on 2026-07-06	<1%
113	Publication	Wang, Sijia. "Towards Sustainable Machine Learning Systems.", Duke University	<1%
114	Publication	Yasir Altaf, Abdul Wahid. "Leveraging ensemble methods with pretrained CNNs f...	<1%
115	Submitted works	York St John University on 2025-01-28	<1%
116	Internet	dokumen.pub	<1%
117	Internet	kluedo.ub.rptu.de	<1%
118	Internet	mobt3ath.com	<1%
119	Internet	ojs.unimal.ac.id	<1%
120	Internet	ri-ng.uaq.mx	<1%
121	Internet	www.iris.unina.it	<1%
122	Internet	www.nature.com	<1%

123	Internet	www.ukm.my	<1%
124	Submitted works	Anna University on 2024-03-13	<1%
125	Submitted works	Asia Pacific University College of Technology and Innovation (UCTI) on 2023-08-06	<1%
126	Publication	Keerthi Kumar M, Parameshachari BD. "Pseudo-Temporal 3D CNN Fusion of Gradi...	<1%
127	Submitted works	National Economics University on 2026-03-30	<1%
128	Publication	Paolo Ferro, Harinadh Vemanaboina, Chander Prakash. "Computational Techniqu...	<1%
129	Submitted works	Somerville House on 2026-07-24	<1%
130	Submitted works	The University of Manchester on 2022-05-06	<1%
131	Submitted works	Ton Duc Thang University on 2025-02-16	<1%
132	Submitted works	Universidad Francisco de Vitoria on 2026-01-08	<1%
133	Submitted works	University of Hertfordshire on 2024-11-11	<1%
134	Publication	A. Punitha, S. Syedakbar, S. Jeyasudha. "Intelligent and Sustainable Systems: AI, G...	<1%
135	Publication	Alnajjar, Majd. "Artificial Intelligence Meets Medulloblastoma: Breaking Barriers i...	<1%
136	Publication	Harvinder Singh, Priyanka Kaushal, Sarabjeet Kaur. "Advanced Computing and AI...	<1%

137	Publication	Jaskaran Singh, Meenu Gupta, Rakesh Kumar. "Smart Technologies and Intelligen...	<1%
138	Submitted works	Liverpool John Moores University on 2023-09-13	<1%
139	Submitted works	Liverpool John Moores University on 2026-05-05	<1%
140	Submitted works	National College of Ireland on 2025-04-11	<1%
141	Publication	Parikh, Yogita Maheshkumar. "Heart Chamber Segmentation From Cardiac MRI.",...	<1%
142	Submitted works	The Maldives National University on 2022-05-30	<1%
143	Submitted works	University of East London on 2025-09-09	<1%
144	Submitted works	University of Ulster on 2026-04-15	<1%

Midocean University
College of Informatics

**A Comparative Study of Deep Learning Models for American
Sign Language Fingerspelling Recognition**

Prepared By

Anas Abdulkader Mohammed Alzain ID No.: 453001724

Mohammed Omar Abdullah Alqathmi ID No.: 253001437

Submitted in fulfillment of the requirements for the degree of
Master of Science in Artificial Intelligence

At
College of Informatics
Midocean University

Supervisor: Dr. Hagar Saleh
College of Informatics, Midocean University, Comoros

ACKNOWLEDGEMENTS

Our deepest thanks go to our supervisor, Dr. Hagar Saleh. Her guidance, encouragement, and sound advice were essential to every stage of preparing and carrying out this Master's thesis in Artificial Intelligence.

The scientific rigor of this work owes much to her broad experience and the constructive criticism through which she shaped its direction. Throughout this journey, her enthusiasm for developing our academic and research skills was a genuine source of motivation.

Without her wise direction, this thesis simply would not have reached completion. We are sincerely grateful for her patience, continual encouragement, and the time and trust she generously invested in us.

We also thank the College of Informatics at Midocean University for the academic environment and resources that made this research possible, and our families, whose unwavering support carried us through our studies.

ABSTRACT

This study developed the hybrid model based on Swin Transformer and MobileNetV2 for enhance recognizing American Sign Language (ASL) fingerspelling. The hybrid model was compared with three pre-trained CNN models: ResNet50, MobileNetV2, EfficientNetB0 and Transformer models (Swin and ViT), for recognizing American Sign Language (ASL) fingerspelling from static images, and 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, and 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. Training images were augmented with rotation, shifting, zoom, shear, and brightness changes for the CNN models, and with rotation and color jitter for the transformer-based models, while horizontal flipping was avoided because it can reverse the meaning of a handshape; validation and test images were left unaugmented. The proposed Hybrid model fuses a lightweight MobileNetV2 branch, which captures fine local texture and edge detail, with a Swin Transformer branch, which captures broader relationships across the whole hand, concatenating the two feature vectors before classification so that the final decision draws on both local and global evidence, which is what gives it an edge over either backbone alone. 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, and 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.

LIST OF ABBREVIATIONS

Abbreviation	Meaning
AI	Artificial Intelligence
ANN	Artificial Neural Network
ASL	American Sign Language
CNN	Convolutional Neural Network
CV	Computer Vision
D&M	Deaf and Mute
DL	Deep Learning
F1	F1-score (harmonic mean of Precision and Recall)
FN	False Negative
FP	False Positive
GPU	Graphics Processing Unit
ML	Machine Learning
NLP	Natural Language Processing
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue (colour channels)
SLR	Sign Language Recognition
SW-MSA	Shifted-Window Multi-head Self-Attention
TN	True Negative
TP	True Positive
ViT	Vision Transformer
W-MSA	Window-based Multi-head Self-Attention
WHO	World Health Organization
XAI	Explainable Artificial Intelligence

LIST OF FIGURES

Figure 1.1: Global Hearing Loss Statistics (WHO, 2021)	7
Figure 2.1: Artificial Intelligence (AI) — Relationship to Machine Learning and Deep Learning	12
Figure 2.2: Computer Vision (CV)	13
Figure 2.3: Deep Learning (DL)	13
Figure 2.4: How a CNN Processes an Image.....	Error! Bookmark not defined.
Figure 2.5: Convolutional Layers	Error! Bookmark not defined.
Figure 2.6: Pooling Layers.....	Error! Bookmark not defined.
Figure 2.7: Fully Connected Layers.....	15
.....	9
Figure 3.1: Proposed Methodology Pipeline (Eight-Stage View, with Word-Formation and Spelling-Correction)	27
Figure 3.2: Data Augmentation Examples (Rotation, Flip, Zoom, Brightness, and Shift)	29
Figure 3.3: Hybrid MobileNetV2-Swin Dual-Branch Architecture	31
Figure 4.1: Test Accuracy Comparison of the Six Models	37
Figure 4.2: Weighted Precision, Recall, and F1-Score Comparison of the Six Models.....	38
Figure 4.3: Confusion Matrices of the Six Models (36 Classes, Test Set)	40
Figure 4.4: Model Size Comparison of the Six Models.....	43
Figure 4.5: Model Parameter Count Comparison of the Six Models	44
Figure 4.6: Total Inference Time Comparison of the Six Models on the Test Set.....	44
Figure 4.7: Average Inference Time per Image of the Six Models	45
Figure 4.8: Accuracy–Speed Trade-off of the Six Models	45
Figure 4.9: Gradio Application — Single Image / Camera Tab	47
Figure 4.10: Gradio Application — Multiple Images to Word Tab	48
Figure 4.11: Gradio Application — Models Comparison Tab	48

LIST OF TABLES

16	Table 2.1: Comparison of Related Studies on Sign Language Recognition.....	23
16	Table 3.1: Number of Images per Class Used in the Dataset (Summary)	28
	Table 3.2: Summary of the Six Deep Learning Models Used	31
13	Table 4.1: Training Configuration	35
	Table 4.2: Performance Comparison of the Six Models on the Test Set.....	37
	Table 4.3: Per-Class F1-Score for the Digit '0' and Letter 'o' Across the Six Models	38
	Table 4.4: Model Size, Parameter Count, and Inference-Time Comparison.....	43

14

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	2
ABSTRACT.....	3
LIST OF ABBREVIATIONS.....	4
LIST OF FIGURES	5
LIST OF TABLES.....	6
TABLE OF CONTENTS.....	7
Introduction.....	9
1.1 Overview.....	9
1.2 Problem Definition.....	10
1.3 Thesis Objectives	11
1.4 Thesis Contribution.....	12
1.5 Thesis Organization	13
Background and Related Work.....	14
2.1 Background.....	Error! Bookmark not defined.
2.1.1 Artificial Intelligence.....	Error! Bookmark not defined.
2.1.2 Computer Vision.....	15
2.1.3 Deep Learning.....	15
2.1.4 Convolutional Neural Networks	15
2.1.5 Vision Transformers	17
2.1.6 The Swin Transformer	18
2.1.7 Hybrid CNN-Transformer Fusion Architectures	Error! Bookmark not defined.
2.1.8 Evaluation Criteria	19
2.2 Related Work	19
2.2.1 CNN-Based Approaches.....	Error! Bookmark not defined.
2.2.2 Hybrid and Multi-Stream Approaches.....	19
2.2.3 Detection-Based (YOLO) Approaches.....	21
2.2.4 Vision Transformer and Swin-Based Approaches.....	22
2.3 Summary and Research Gap.....	25
Proposed Approach.....	27
3.1 Introduction.....	27
3.2 Proposed Methodology Pipeline.....	27
3.3 Dataset Description.....	29
3.4 Data Preprocessing and Augmentation.....	29

83

36

38

129

127

26

21

	3.5 Model Architectures.....	30
	3.6 Word Formation and Automatic Spelling Correction (Demonstration)	32
	3.7 Proposed Algorithm.....	34
	3.8 Evaluation Metrics	35
41	Experimental Results and Discussion.....	36
	4.1 Introduction.....	36
	4.2 Experimental Setup.....	36
	4.3 Comparative Results	37
93	4.4 Per-Class Error Analysis.....	39
	4.5 Discussion.....	41
	4.5.1 Architectural Advantage of the Transformer-Based Models.....	41
	4.5.2 A Corrected Accuracy-Speed Trade-off.....	42
	4.5.3 Comparison with Related Work.....	42
	4.6 Limitations of the Study.....	43
	4.7 Time Analysis	43
	4.8 Word-Formation and Spelling-Correction Application	47
7	Conclusion and Future Work.....	51
	5.1 Summary of the Study	51
	5.2 Key Findings and Contributions	51
97	5.3 Recommendations for Practical Application	52
	5.4 Future Work.....	53
	References.....	55

CHAPTER 1 Introduction

1.1 Overview

Artificial intelligence (AI) now sits at the core of modern technological progress, and one of its most tangible applications is Computer Vision (CV) — the field that lets machines perceive, interpret, and reason about visual data such as images, video, and hand gestures. Convolutional Neural Networks (CNNs) drove a decade of steady gains in the accuracy and reliability of gesture-recognition systems, and more recently Vision Transformers — together with hierarchical variants such as the Swin Transformer and hybrid designs that fuse convolutional with attention-based features — have pushed this further, capturing fine local detail and long-range context within an image at the same time.

The World Health Organization estimates that over 1.5 billion people worldwide live with some degree of hearing loss, with about 430 million experiencing disabling hearing loss that calls for intervention; by 2050, that figure is expected to climb to nearly 2.5 billion — roughly one person in four [31], as shown in Figure 1.1. For the Deaf and Mute (D&M) community, sign languages such as American Sign Language (ASL) are the primary channel of communication, yet most hearing people cannot understand them. The result is a real communication barrier that touches education, employment, and social participation.

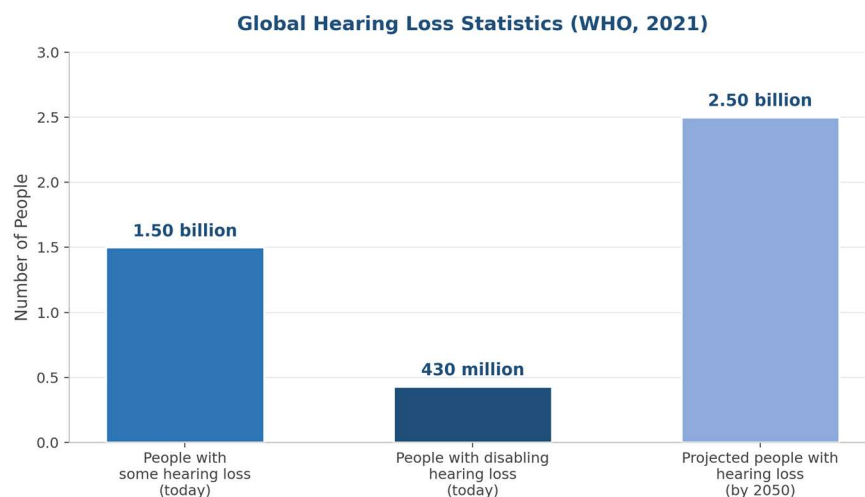


Figure 1.1: Global Hearing Loss Statistics (WHO, 2021)

Automatic sign-language recognition began with bulky, often invasive hardware, such as sensor-embedded gloves, which made everyday use impractical. Systems built instead around ordinary cameras and computer-vision algorithms simplified data collection considerably, though lighting, cluttered backgrounds, and variation in hand pose remain persistent challenges. Deep learning, and CNNs specifically, addressed much of this by extracting robust visual features directly from raw

images. Yet recognizing a single letter is only half the problem: a system that is actually useful for communication also needs to assemble recognized letters into words and correct the errors that inevitably build up across a multi-letter sequence — a capability the existing literature rarely tackles alongside the classification task itself.

This thesis is driven by the aim of building and thoroughly evaluating a deep-learning-based, end-to-end American Sign Language recognition system that can help close the communication gap between the Deaf and Mute community and hearing people. Concretely, six deep learning architectures are compared on the task of recognizing ASL fingerspelling — individual letters and digits — from static images: three established convolutional networks, a standard Vision Transformer, a hierarchical Swin Transformer, and a Hybrid CNN-Transformer fusion model introduced in this work. The aim is to find the architecture that is both the most accurate and the most practical to deploy. To show what the strongest of these models enables in practice, a supporting demonstration tool was also built around automatic multi-letter word formation and automatic English spelling correction, delivered through an interactive multi-model web application; this tool supports the thesis's central experimental comparison without being that contribution itself.

1.2 Problem Definition

For Deaf and Mute individuals, sign language is central to daily life. Like other sign languages, American Sign Language has its own grammar and visual vocabulary, largely unknown to hearing people, and this unfamiliarity creates a communication barrier that reaches into health care, education, employment, and other essential areas of life. Fingerspelling — spelling out words using a distinct hand shape for each letter and digit — is one of ASL's foundational building blocks, and teaching a system to recognize fingerspelling gestures automatically is an important step toward fuller sign-language translation.

The earliest motion-recognition devices depended on hardware sensors that worked but were often invasive, costly, and impractical day to day. Moving to camera-based, vision-only methods removed that hardware burden, but traded it for sensitivity to lighting, background, and camera angle. Deep learning, and CNNs above all, have gone a long way toward solving feature extraction from raw images, yet most studies in this area still share the same few gaps. Most compare only conventional CNN architectures against each other — AlexNet, VGG, ResNet, and similar — and even the smaller, newer body of work on vision transformers rarely tests a purpose-built hybrid CNN-transformer model alongside them under matching conditions. Inference time, model size, and parameter count are also rarely reported next to accuracy, despite mattering directly for real-time, resource-constrained assistive use. And almost no study pairs recognition accuracy with a downstream word-formation and spelling-correction step, even though a classifier that only

recognizes single letters is of limited use to someone who wants to communicate full words and sentences.

The problem this thesis takes on, then, is twofold: first, no rigorous, unified experimental comparison currently evaluates a broad range of modern architectures — pretrained CNNs (ResNet50, MobileNetV2, EfficientNetB0), a standard Vision Transformer, a hierarchical Swin Transformer, and a novel Hybrid MobileNetV2-Swin fusion model — on the same ASL fingerspelling recognition task, dataset, preprocessing, and evaluation protocol; second, published work pairing such a comparison with a working demonstration of multi-letter word assembly and spelling correction is essentially absent. Both a comparison of this kind and an end-to-end demonstration are needed to give researchers and developers evidence-based guidance when building practical ASL recognition systems.

1.3 Thesis Objectives

The overarching goal of this thesis is to design, build, and rigorously evaluate a comparative deep-learning framework for recognizing American Sign Language fingerspelling from static images, then extend it with practical word-formation and spelling-correction abilities, so as to identify the model best suited — in both accuracy and practicality — to an assistive ASL communication system. This goal breaks down into the following specific objectives:

- Assemble and organize an ASL fingerspelling image dataset spanning all 36 classes (digits 0–9, letters a–z), including data cleaning and a single, reproducible 70/10/20 train–validation–test split (fixed random seed) shared by all six models.
- Build a preprocessing and augmentation pipeline that standardizes image dimensions and enriches the diversity of the training data, so as to strengthen how well each model generalizes.
- Train, on the prepared dataset, three pretrained CNNs via transfer learning (ResNet50, MobileNetV2, EfficientNetB0), a standard Vision Transformer, a hierarchical Swin Transformer, and a Hybrid MobileNetV2-Swin dual-branch fusion model designed for this thesis.
- Evaluate all six models under one shared protocol — accuracy, weighted precision/recall/F1-score, macro F1-score, inference time, model size, and parameter count — to keep the comparison fair and reproducible.
- Carry out a detailed per-class error analysis to pin down which letters and digits give each model the most trouble, and explain why.

- As a supporting rather than a primary objective, design and build an automatic word-formation pipeline that assembles a sequence of individually recognized letters into a word, paired with an NLP stage that applies automatic English spelling correction.
- As a further supporting objective, integrate all six trained models together with the word-formation/spelling-correction pipeline into one interactive, multi-model Gradio application, so the six models' results can be viewed and compared side by side.
- Identify which model is the most efficient overall, and put forward practical recommendations for deploying an ASL recognition system, including possible integration into real-time communication platforms.

1.4 Thesis Contribution

This work contributes the following to the field of automatic sign-language recognition:

- A unified, reproducible benchmark that pits three widely used pretrained CNNs, a standard Vision Transformer, a hierarchical Swin Transformer, and a new Hybrid MobileNetV2-Swin fusion model against each other on the same ASL fingerspelling task, dataset, and evaluation protocol — a six-way, cross-paradigm comparison still rare in the literature.
- A new Hybrid MobileNetV2-Swin architecture, fusing a 1,280-dimensional MobileNetV2 feature vector with a 768-dimensional Swin Transformer feature vector into one 2,048-dimensional representation before classification. It achieves the highest accuracy of the six models (99.80%), showing that pairing convolutional and self-attention feature extraction pays off on this task even with a fairly modest dataset.
- Evidence that the Hybrid model, the Swin Transformer, and the Vision Transformer clearly outperform the three CNN baselines here, together with a counter-intuitive discovery: the standalone Swin Transformer, not a CNN, is the fastest of all six models on both total and per-image inference time, while the larger Vision Transformer is the slowest — a finding that revises the usual assumption that transformer models are necessarily slower.
- A fine-grained, per-class error analysis applied identically to all six models, showing that the digit '0' and the letter 'o' — near-identical closed-circle handshapes in ASL — are the most consistent source of error across every architecture, and essentially the only source of error in the two top-performing models; the three CNN baselines, by contrast, show wider and deeper confusion, including among the letters m/n/o and u/v/w and the digit pair 4/5.
- Alongside these experimental contributions, a supporting multi-letter word-formation pipeline paired with a TextBlob-based automatic spelling-correction stage, showing what the accuracy of the top models makes possible in practice, delivered within a single Gradio

application with dedicated tabs for single-image classification, multi-image word formation, and side-by-side model comparison.

- Concrete recommendations for extending this work into real-time and assistive-technology settings, including a proposed design for folding the recognition and spelling-correction pipeline into video-conferencing and communication platforms in support of the Deaf and Mute community.

1.5 Thesis Organization

The rest of the thesis unfolds across four chapters. Chapter 2 lays out the theoretical background needed to follow this work — artificial intelligence, computer vision, deep learning, convolutional neural networks, Vision Transformers, the Swin Transformer, and hybrid CNN-Transformer architectures — then reviews related work on sign-language recognition and closes by summarizing the research gap this thesis addresses. Chapter 3 sets out the proposed approach: the overall methodology pipeline, the dataset, preprocessing and augmentation, the architecture of each of the six compared models (including the Hybrid MobileNetV2-Swin model), the word-formation and spelling-correction pipeline, the proposed algorithm, and the evaluation metrics. Chapter 4 presents and discusses the experimental setup and results — training configuration, comparative performance, a detailed per-class error analysis, a discussion of the findings against related work, the study's limitations, a dedicated time analysis, and a demonstration of the word-formation and spelling-correction application. Chapter 5 closes the thesis with a summary of the key findings and contributions, practical recommendations, and directions for future work.

CHAPTER 2

Background and Related Work

This chapter introduces the theoretical basis required for grasping the system proposed, plus a survey of literature related to automatic sign-language recognition. The background presented is a rather short but self-contained overview that traces the path from artificial intelligence to computer vision, then on to deep learning, convolutional neural networks, Vision Transformers, the Swin Transformer and hybrid CNN-Converter Architectures.

2.1 Background

2.2.1 Artificial Intelligence

Artificial Intelligence (AI) is a field of computer science that focuses on creating systems that do human-like tasks, like learning, seeing things, making decisions, and so forth [1]. AI systems find patterns in data statistically, instead of being instructed by rules set out in advance.

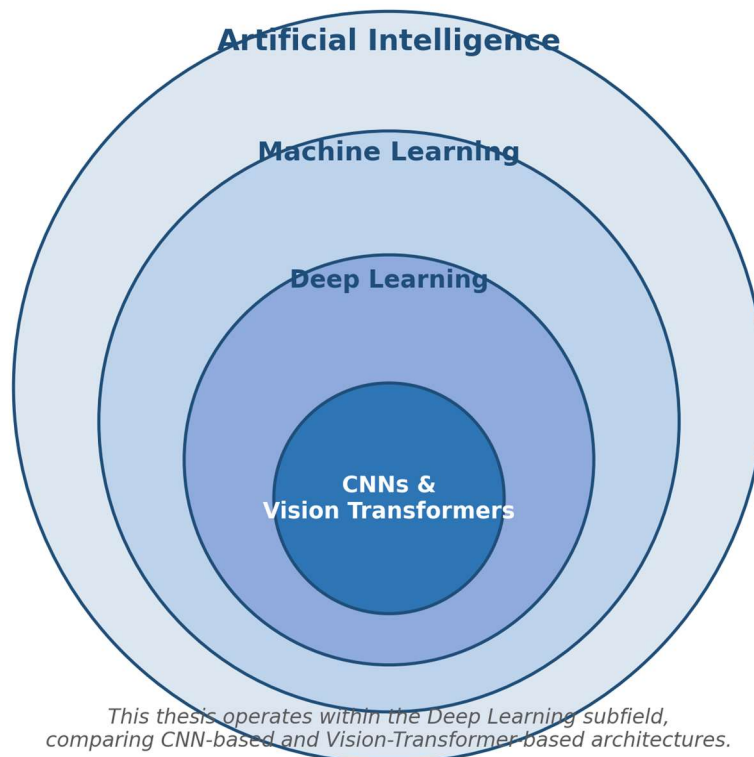


Figure 2.1: Artificial Intelligence (AI) — Relationship to Machine Learning and Deep Learning

2.1.2 Computer Vision

One area of AI that is being most widely practically applied is computer vision (CV) technology. With this capability, machines can process image and video, extracting insights from raw pixels to identify objects and patterns for recognition and classification tasks [2]. The technology also allows for detection of hand gestures and recognition of sign language which is the focus of the thesis described here.

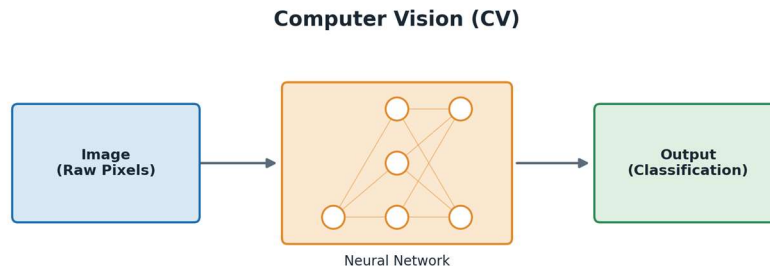


Figure 2.2: Computer Vision (CV)

2.1.3 Deep Learning

Deep Learning (DL), a subfield of machine learning, uses multi-layered networks to learn features hierarchically, gradually turning raw input into representations that grow more abstract and task-relevant with depth [3]. This automatic, layered feature extraction sets deep learning apart from earlier machine-learning methods that depended on hand-crafted features, and it is exactly this capability that has powered recent advances in recognizing sign language directly from raw images [4].

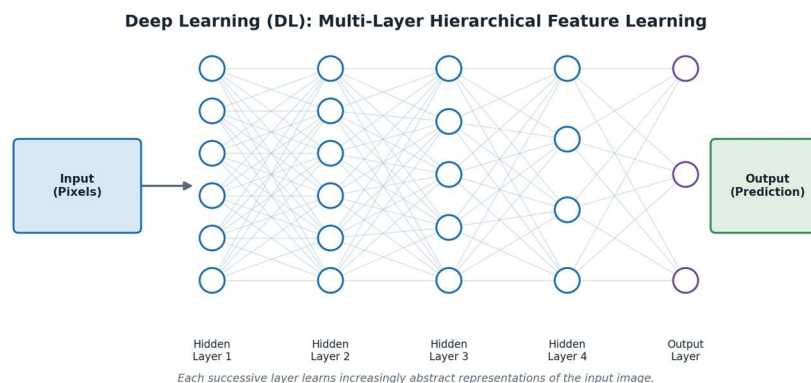


Figure 2.3: Deep Learning (DL)

2.1.4 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are deep learning architectures built for grid-structured data, most notably two-dimensional images [5]. A typical CNN passes an input image through a series of specialized layers, shown in Figure 2.4.

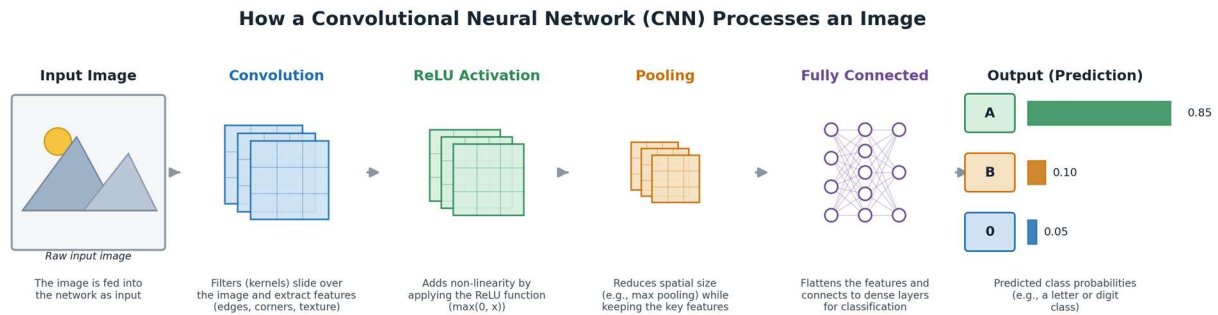


Figure 2.4: How a CNN Processes an Image

Small, learnable filters slide across the image in the convolutional layers, producing feature maps that pick out visual patterns such as edges, corners, and texture (Figure 2.5). A non-linear activation function — typically the Rectified Linear Unit (ReLU) — follows, speeding up training and easing the vanishing-gradient problem.

Convolutional Layers: Filters Sliding over the Input

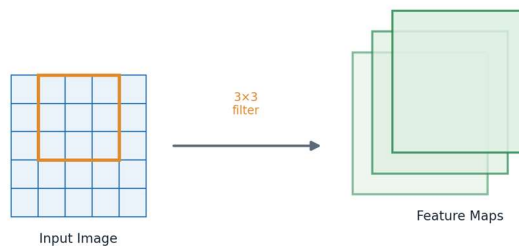


Figure 2.5: Convolutional Layers

The feature maps are then downsampled by pooling layers, usually max pooling (Figure 2.6), which both cuts computational cost and makes the network less sensitive to small shifts in where the hand sits within the image.

Pooling Layers: Reducing Spatial Dimensions (Max Pooling)



Figure 2.6: Pooling Layers

One or more fully connected layers then combine these features into a final classification (Figure 2.7), typically through a Softmax function that converts the network's output into a probability distribution over the target classes. Dropout — randomly deactivating a fraction of neurons during training — is often applied before this final layer to curb overfitting and improve generalization [6].

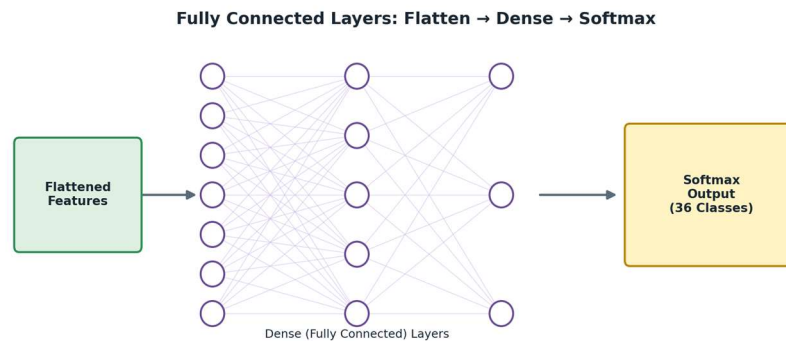


Figure 2.7: Fully Connected Layers

Instead of training a CNN from scratch, this thesis relies on transfer learning: starting from a model already trained on a large, general-purpose dataset such as ImageNet. For the three CNN baselines used here — ResNet50 [7], MobileNetV2 [8], and EfficientNetB0 [9] — the pretrained convolutional layers stay frozen, and only a new classification head, sized for the 36-class ASL fingerspelling task, is added and trained. This cuts down sharply on the amount of task-specific data needed for strong performance, which matters given the comparatively modest size of the ASL dataset used in this study.

2.1.5 Vision Transformers

Vision Transformers (ViT) take the self-attention mechanism originally built for natural language processing and apply it directly to sequences of image patches. An input image is cut into fixed-size, non-overlapping patches — 16×16 pixels in the standard ViT-Base configuration used here — each patch is linearly embedded, and a standard Transformer encoder then runs multi-head self-attention across the full sequence of patch embeddings [10]. Because self-attention relates every patch to every other patch in one step, a Vision Transformer can pick up long-range relationships between distant parts of an image from its very first layer, offering a wider field of view than a convolutional filter's local receptive field typically allows. The trade-off is that it needs substantially more parameters and training data to reach its potential, since it lacks the built-in spatial inductive bias — locality and translation invariance — that convolutional layers get by design.

2.1.6 The Swin Transformer

The Swin Transformer is a hierarchical variation of the Vision Transformer which adds a CNN-like bias back to the self-attention mechanism [11]. It processes information in four consecutive levels, gradually combining patches to reduce the spatial resolution while increasing depth of features, resembling how a CNN works, which was an inspiration to the researchers. Each stage consists of a combination of window-based multi-head self-attention (W-MSA) which is only calculated within small local windows for faster processing, and shifted-window multi-head self-attention (SW-MSA) which moves the boundaries between blocks enabling exchange of information. Because of this, it is able to perform both very precise, local operations similar to the convolutional ones on one hand but, it is still a good mechanism for the modeling of the long-range relationships. The thesis hypothesizes that this is one of the main reasons why such a model is well suited to recognizing subtle hand-shape differences among nearly indistinguishable sign language letters and digits in real-time and, at the same time, is able to remain computationally light. In support the study shows that the Swin Transformer was both the most accurate of the three transformer-based models relative to its size and also the unexpected fastest among all six models compared (Section 4.7).

More external sign-language evidence supports this further: a hybrid model combining Swin Transformer as a feature extractor for identification of monkeypox skin-lesion achieved 95.6% accuracy, Much outperforming CNNs like DenseNet121 VGG16 ResNet18, and AlexNet on the same tasks [12]. This finding prompted not only to include the Swin Transformer in the comparison but also the concept of Hybrid MobileNetV2-Swin architecture here.

2.1.7 Hybrid CNN-Transformer Fusion Architectures

Currently there is an increasing trend towards the integration of convolutional and self-attention feature extraction in a single model with the general belief that their functionalities can be complements to each other, the convolution features detect local texture and detail, but the self-attention features grasp global context and spatial relationships in an over the entire picture way. This thesis contributes a Hybrid MobileNetV2-Swin architecture in that spirit, fully described in Section 3.5: a lightweight MobileNetV2 branch and a Swin Transformer branch process the same image in parallel, their pooled feature vectors (1,280 and 768 dimensions) are concatenated into a single 2,048-dimensional fused representation, and a two-layer classification head with ReLU and Dropout produces the final 36-class prediction. Similar dual-branch fusion strategies have proven effective elsewhere in the sign-language literature — a hybrid dual-path CNN combined with a Vision Transformer reached 99.97% accuracy on a large ASL alphabet dataset [13], and two-stream CNNs that fuse features from multiple inputs have consistently beaten single-stream

baselines on ASL fingerspelling tasks [14], [15]. What sets the Hybrid MobileNetV2-Swin model proposed here apart from these prior approaches is that it pairs a CNN branch with a hierarchical, rather than standard, Vision Transformer branch, and that it is evaluated for the first time directly alongside five other architectures spanning both convolutional and transformer-based paradigms under one identical protocol.

2.1.8 Evaluation Criteria

A multi-class system like the one proposed here needs a clearly defined set of quantitative criteria to evaluate it properly. Accuracy — correct predictions as a share of all predictions — is the most commonly reported metric, but it can mislead when class distributions are uneven, since a model can look accurate overall while doing poorly on minority classes. For that reason, this thesis reports weighted precision, recall, and F1-score alongside accuracy and macro-averaged F1-score, together with inference time, model size, and parameter count; these are formally defined in Section 3.8.

2.2 Related Work

This section highlights some recent research on deep-learning-based automatic sign-language recognition, arranged in four categories: single-CNN approaches; hybrid and multi-stream approaches; detection-based (YOLO) approaches, and vision-transformer approaches. A brief paragraph in each group summarizes how the work reported in each fit into this thesis's design choices. More details on these choices are given in Section 2. 3.

2.2.1. A Summary of Some Single-CNN-Based ASL Recognition Work

Earlier ASL recognition work mostly uses some form of only one CNN architecture.

Zakariah et al. [16] recognized thirty-two signs of the Arabic Sign Language Dataset using transfer learning. They took a already pre-trained EfficientNetB4 model to fine-tune it on 54,049 RGB images from the videos recorded by 40 signers; they got a maximum accuracy of 95% at some significant computational cost.

Yadav and Bohra [17] designed a CNN-based system to automatically translate ASL fingerspelling into text and speech in real time. To do so, a separate image preprocessing stage that comes before the classification helped this system to achieve the 95. 8% accuracy. Still, the system was limited to static fingerspelling.

Sheth et al. [18] introduced a system which, apart from recognizing, can also reproduce ASL signs with the help of its CNN. It was trained on 25,000 images covering 26 letters and common words that make up a large fraction of the alphabet. The model got a very high training accuracy of 98. 84% and was resistant to changing lighting and skintone. Still, it was resource-hungry to do such computations.

Kamruzzaman [19] has demonstrated that the same CNN approach works perfectly for the recognition of Arabic Sign Language as well as the generation of speech, Because of this reaching a 90% accuracy level.

Marjusalinah, Samsuryadi, and Buchari [20] performed a comparison test between ResNet50 head-to-head with a public Kaggle ASL Alphabet dataset of ASL fingerspelling at the same time on DenseNet121 where ResNet50 was slightly better reaching a near-perfect 99.99%.

Hurroo and Walizad [21] built an input CNN with only one layer and trained it on 25,110 images that cover all the letters (26) and numbers (0, 9) to achieve 99.0% training and 98.0% testing accuracy which apart from the model training and test set accuracy that were very close, implied very limited overfitting unlike earlier approaches where the difference was much bigger despite using very similar inputs). The problem with Hurroo/Walizad's model is that it is limited only to static gestures.

Broad overview of deep-learning ASL works that Sharma [22] did has pointed to In reality deep learning techniques consistently outperform earlier human hand-crafted features not only for ASL but also for other sign-language datasets such as the Indian Sign Language but the presence of the complex background was an ongoing issue.

Relevance of the thesis: In short, all these works provide a strong empirical support that CNN networks in general and the ResNet50 [17] in particular are well-established baselines for the recognition of fingerspelling of ASL and for this reason, this thesis includes the three CNN baselines namely ResNet50, MobileNetV2, and EfficientNetB0. \par Yet, these works did not compare their CNN with Vision Transformer or a hybrid CNN-Transformer model, the results of the evaluations of Vision Transformer, Swin Transformer, and hybrid model are reported separately in this thesis by comparing them all in a single experiment.

2.2.2 Hybrid and Multi-Stream Approaches

A second group of studies improves robustness by combining multiple feature sources or processing stages.

Zhidebayeva et al. [14] introduced a real-time fingerspelling recognition method built around a two-dimensional deep CNN with two-stream feature extraction, designed to capture spatio-temporal information; accuracy climbed further as more training data was added.

Ma, Xu, and Kim [15] proposed a two-stream mixed CNN (TSM-CNN) that fuses features from two consecutive frames, specifically to handle the moving gestures needed for the letters 'J' and 'Z'; across four different CNN backbones, fusing two visual streams consistently beat a single-stream baseline.

Bayrak, Nabiyeve, and Atalar [23] took a mathematically different route, pairing Complex Zernike Moments with a Complex-Valued Deep Neural Network and reaching 98.67% accuracy.

Kouvakis, Trevlakakis, and Boulogeorgos [24] introduced an encoder–decoder approach for transmitting ASL images semantically and bandwidth-efficiently, achieving roughly 99% reconstruction/recognition accuracy even under noisy channel conditions.

Relation to this thesis: these studies show that fusing multiple streams or representations of the same gesture reliably beats a single-stream CNN, which is the direct motivation behind this thesis's Hybrid MobileNetV2-Swin architecture. Here, though, the two “streams” being fused are not two temporal frames but two structurally different feature extractors — a convolutional branch and a self-attention branch — processing the same static image in parallel. The result that this Hybrid model achieves the highest accuracy of all six compared (Section 4.3) extends this literature's general finding along a new fusion axis: architecture type, rather than temporal frame or mathematical representation.

2.2.3 Detection-Based (YOLO) Approaches

A third group of studies treats sign-language recognition as an object-detection problem rather than pure classification, with real-time speed as the priority.

Mujahid et al. [25] applied YOLOv3 with a DarkNet-53 backbone to a custom dataset of 2,000 images, reaching 97.68% mean average precision while running well ahead of SSD and Faster R-CNN in speed.

Bankar et al. [26] turned to YOLOv5 after finding that conventional CNN classifiers degraded sharply against cluttered backgrounds; YOLOv5 reached 88.4% accuracy against just 52.98% for a standard CNN baseline on the same task, though its precision (76.6%) was comparatively lower.

Aggarwal et al. [27] proposed ASLR-Net, a CNN deliberately hardened against lighting changes, viewing-angle shifts, and background clutter through extensive augmentation; it reached 96.9% accuracy that held up even under previously unseen conditions.

Drishya [28] pushed the problem further, into continuous video rather than isolated static images, demonstrating accurate hand-movement tracking across live sequences — at the cost of needing a substantially larger and more diverse video dataset.

Relation to this thesis: this cluster makes clear that inference time and robustness to real-world conditions matter just as much as raw accuracy for a deployable system. That is exactly why this thesis reports inference time, model size, and parameter count alongside accuracy for all six models (Section 4.7), and why dynamic-gesture and video-based recognition are discussed explicitly as future work in Chapter 5, rather than chasing only incremental accuracy gains on static images.

2.2.4 Vision Transformer and Swin-Based Approaches

The last group of studies, and the one most directly relevant here, moves past CNNs toward attention-based architectures.

Aly and Fathi [13] combined a dual-path CNN with a Vision Transformer to recognize 29 ASL gestures, training on the 87,000-image ASL Alphabet dataset and reaching a near-perfect 99.97% accuracy at 110 frames per second — strong evidence that combining convolutional and attention-based feature extraction pays off for this task, and a direct precedent for the Hybrid MobileNetV2-Swin model proposed in this thesis.

Alsharif et al. [29] benchmarked five deep learning models — AlexNet, ConvNeXt, EfficientNet, ResNet-50, and a Vision Transformer — on the same large ASL alphabet dataset, and found that deeper, more modern architectures, the Vision Transformer included, consistently outperformed earlier CNN designs.

Kasapbaşı et al. [30] developed DeepASLR, a custom CNN paired with a newly built, deliberately varied ASL dataset, reaching 99.38% accuracy and underscoring how much dataset diversity matters for robust performance.

A 2024 study that applied the Swin Transformer architecture across several sign-language datasets [32] confirmed that the same hierarchical, shifted-window design used in this thesis generalizes well across different sign-language recognition benchmarks.

At a more foundational level, Dosovitskiy et al. [10] first showed that a pure Transformer, applied directly to sequences of image patches with no convolutional layers at all, can match or beat CNN performance on large-scale image classification given enough pretraining data — this is the architecture adopted directly as one of the six models compared in this thesis.

Relation to this thesis: this is the smallest but fastest-growing cluster of studies, and the one this thesis most directly builds on. Where [13] and [29] either combine a transformer with a CNN as one fixed design, or include a transformer as just one of several models without a dedicated look at inference time and model size, this thesis benchmarks a standalone Vision Transformer, a standalone hierarchical Swin Transformer, and a new Hybrid MobileNetV2-Swin model against three separately optimized CNN baselines under one identical protocol, complete with a dedicated per-class error analysis and an accuracy–inference-time trade-off. This extends the evidence from [13], [29], and [32] into a more tightly controlled, six-way comparison, and revises the usual trade-off narrative with the finding that the standalone Swin Transformer — not a CNN — is actually the fastest model overall.

Table 2.1 pulls together the studies reviewed above — the models used, their main strengths and limitations, and the datasets and results reported — alongside this thesis for direct comparison.

Table 2.1: Comparison of Related Studies on Sign Language Recognition

Ref.	Year	Model(s)	Strength	Limitation	Best Result	Accuracy
[10]	2021	Vision Transformer (ViT)	Pure self-attention, no convolutions needed	Requires large pretraining data	State-of-the-art on ImageNet	N/A
[11]	2021	Swin Transformer	Hierarchical, efficient shifted-window attention	More complex than standard CNNs	State-of-the-art on multiple vision benchmarks	N/A
[12]	2024	Swin + PSO-SVM (hybrid)	Outperforms CNN baselines on lesion classification	Domain outside sign language	Acc. = 95.6%	95.6%
[13]	2025	Hybrid CNN + ViT	Near-perfect accuracy	Static images only	Acc. = 99.97%	99.97%
[14]	2024	2D Deep CNN, two-stream	Captures spatio-temporal cues	Needs large training data	High accuracy	N/A
[15]	2022	Two-Stream Mixed CNN	Handles moving letters (J, Z)	Needs paired consecutive frames	Improved over single-stream	N/A
[16]	2022	EfficientNetB4	Handles complex images well	High computational cost	Acc. = 95%	95%
[17]	2021	CNN + preprocessing filter	Improved accuracy via preprocessing	Static fingerspelling only	Acc. = 95.8%	95.8%
[18]	2023	CNN (recognition + generation)	Robust to lighting/skin tone	Hardware-constrained	Acc. = 98.84%	98.84%
[19]	2020	CNN (Arabic SL)	High accuracy on a different SL	High training cost	Acc. = 90%	90%
[20]	2021	ResNet50 vs DenseNet121	Near-perfect accuracy (ResNet50)	Excluded letters 'J', 'Z'	Acc. = 99.99%	99.99%
[21]	2020	CNN (grayscale input)	Small train-test gap	Static gestures only	Acc. = 98%	98%
[22]	2020	Survey (CNN, HMM)	Detailed taxonomy of methods	Complex backgrounds remain hard	—	N/A
[23]	2024	Complex-Valued DNN	Exceptional accuracy via complex features	Requires specialised math framework	Acc. = 98.67%	98.67%

Ref.	Year	Model(s)	Strength	Limitation	Best Result	Accuracy
[24]	2024	Semantic encoder–decoder	High-quality reconstruction	High computational overhead	Acc. $\approx$ 99%	$\approx$ 99%
[25]	2021	YOLOv3 (DarkNet-53)	High speed and accuracy	Needs high-performance GPU	mAP = 97.68%	N/A
[26]	2022	YOLOv5	Robust to cluttered backgrounds	Lower precision (76.6%)	Acc. = 88.4%	88.4%
[27]	2023	ASLR-Net (CNN)	Robust to unseen conditions	Static gestures only	Acc. = 96.9%	96.9%
[28]	2023	CNN (video-based)	Handles continuous motion	Needs large video dataset	Reliable tracking	N/A
[29]	2023	AlexNet, ConvNeXt, EfficientNet, ResNet-50, ViT	Broad, modern comparison	—	Competitive across models	N/A
[30]	2022	DeepASLR (custom CNN)	New diverse dataset	Custom, non-public dataset	Acc. = 99.38%	99.38%
[32]	2024	Swin Transformer	Generalises across SL datasets	—	Consistently strong results	N/A
This thesis	2026	ResNet50, MobileNetV2, EfficientNetB0, ViT, Swin, Hybrid MobileNetV2-Swin	Unified 6-model benchmark (3 CNN + ViT + Swin + novel Hybrid) with inference-time, model-size, per-class analysis, and integrated spelling correction / word formation	Moderate dataset size (2,515 images)	Acc. = 99.80% (Hybrid)	99.80%

2.3 Summary and Research Gap

The literature surveyed above shows that deep learning, and CNNs especially, has driven real progress in automatic ASL recognition, with accuracies often above 95–99% under favorable dataset conditions. Four gaps, however, keep showing up across this body of work. First, most studies compare CNN architectures only against each other, and only a small, recent subset [10], [11], [13], [29], [32] evaluate vision transformers on this or a related task — usually without directly benchmarking a standard Vision Transformer, a hierarchical Swin Transformer, and a purpose-built hybrid CNN-Transformer model against multiple CNN baselines under matching conditions. Second, inference time, model size, and parameter count are rarely reported alongside accuracy despite mattering directly for real-time, assistive deployment; and where inference time is reported, it is rarely re-examined for the counter-intuitive possibility that a well-optimized transformer could beat a CNN on speed as well as accuracy. Third, per-class error analysis —

pinpointing which letters or digits a given model systematically confuses, and why — is seldom reported in detail, even though it would directly inform future data collection and model refinement. Fourth, virtually no study pairs a classification benchmark with a working demonstration of multi-letter word assembly and automatic spelling correction, leaving a gap between being able to classify a single letter and being able to help a Deaf user actually communicate a word.

This thesis takes on all four gaps directly. It benchmarks three widely used pretrained CNNs (ResNet50, MobileNetV2, EfficientNetB0) against a standard Vision Transformer, a hierarchical Swin Transformer, and a Hybrid MobileNetV2-Swin fusion model on the same ASL fingerspelling task, dataset, and evaluation protocol; it reports inference time, model size, and parameter count alongside accuracy, precision, recall, and F1-score, revising the conventional accuracy-speed trade-off accordingly; it carries out a detailed per-class error analysis identifying and explaining exactly which classes challenge each architecture most; and it designs, builds, and demonstrates a multi-letter word-formation and automatic spelling-correction pipeline within an interactive multi-model web application. Chapter 3 lays out the full approach used to meet these objectives.

23

40

70

CHAPTER 3

Proposed Approach

3.1 Introduction

80

This chapter sets out the approach used to compare the six deep learning models introduced in Chapter 1, starting with the proposed methodology pipeline. It then covers the dataset, the preprocessing and augmentation pipeline, the architecture of each of the six models (including the Hybrid MobileNetV2-Swin model), the multi-letter word-formation and spelling-correction demonstration, the proposed algorithm as pseudocode, and finally the evaluation metrics used for the comparison in Chapter 4.

3.2 Proposed Methodology Pipeline

144

The system as a whole was built around six independently trained models — ResNet50, MobileNetV2, EfficientNetB0, the Vision Transformer, the Swin Transformer, and the proposed Hybrid MobileNetV2-Swin model — as Figure 3.1 shows. A single, unified data-preparation pipeline feeds all six models, and their outputs are then brought together for comparative evaluation, model selection, and the downstream word-formation and spelling-correction demonstration. The same eight stages apply consistently across all six models:

5

- Stage 1 — Data Collection: A fingerspelling image dataset covering all 36 classes (digits 0–9, letters a–z) was gathered and organized into one folder per class.
- Stage 2 — Data Preparation: Non-image files were removed and class folders verified to clean the raw dataset, which was then split, using a fixed random seed (42), into training (70%), validation (10%), and testing (20%) subsets — the same split reused identically across all six models — yielding 1,760 training, 251 validation, and 504 testing images out of 2,515 total.
- Stage 3 — Data Augmentation: Random rotation, color jitter (brightness/contrast adjustment), and framework-appropriate normalization were applied to the training subset only, to add data diversity and curb overfitting.
- Stage 4 — Model Training: Each of the six models — ResNet50, MobileNetV2, EfficientNetB0, Vision Transformer, Swin Transformer, and the Hybrid MobileNetV2-Swin — was trained independently under one unified protocol, fully described in Section 4.2.
- Stage 5 — Model Evaluation: Each trained model was tested on the same held-out 504-image subset, with accuracy, weighted precision/recall/F1-score, macro F1-score, and inference time all measured on the same NVIDIA A100-SXM4-40GB GPU.

53

25

- Stage 6 — Comparative Analysis and Selection: All six sets of results, including a detailed per-class error analysis, were compared side by side to identify the model best suited on both accuracy and practicality, as reported in Chapter 4.
- Stage 7 — Multi-Letter Word Formation (demonstration): The selected model classifies a sequence of images (or single-image predictions) letter by letter, and the predicted letters are joined in order into a raw candidate word.
- Stage 8 — Automatic Spelling Correction (demonstration): A TextBlob-based NLP stage applies automatic English spelling correction to the raw candidate word, producing a final corrected word that can absorb occasional single-letter misclassifications.

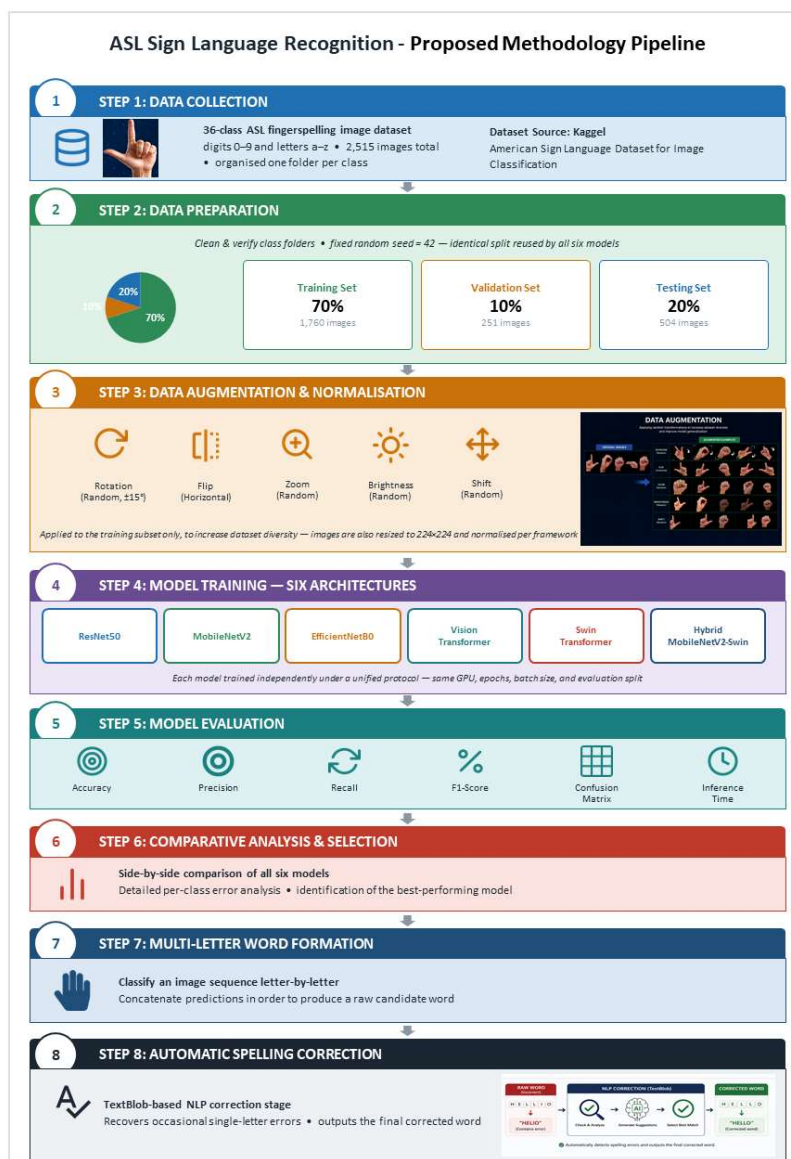


Figure 3.1: Proposed Methodology Pipeline (Eight-Stage View, with Word-Formation and Spelling-Correction)

3.3 Dataset Description

This study's dataset is an ASL fingerspelling image collection covering all 36 classes — the ten digits (0–9) and the twenty-six letters (a–z) — organized in the usual one-folder-per-class layout, and totaling 2,515 images broadly balanced across classes. It was split just once, with a fixed random seed (42), into 70% training (1,760 images), 10% validation (251 images), and 20% testing (504 images). That single split was stored and loaded directly by all six training notebooks, so every model trained and was evaluated on exactly the same images, with exactly 14 test images per class ($36 \times 14 = 504$) — ruling out any risk of an unfair comparison stemming from different data splits.

Table 3.1: Number of Images per Class Used in the Dataset (Summary)

Split	Images	Percentage	Images per Class
Training	1,760	$\approx 70\%$	≈ 48.9
Validation	251	$\approx 10\%$	≈ 7.0
Testing	504	$\approx 20\%$	14 (exact)
Total	2,515	100%	≈ 69.9

3.4 Data Preprocessing and Augmentation

Every image was resized to 224×224 pixels to meet all six models' input requirements. The three PyTorch-based transformer models — Vision Transformer, Swin Transformer, and Hybrid MobileNetV2-Swin — had their pixel values normalized against standard ImageNet statistics (mean = [0.485, 0.456, 0.406], standard deviation = [0.229, 0.224, 0.225]), and their training subset was augmented with random rotation ($\pm 15^\circ$) and color jitter (brightness/contrast adjustment of 0.2). The three TensorFlow/Keras CNN models each used their own dedicated preprocessing function (resnet50_preprocess, mobilenetv2_preprocess, efficientnetb0_preprocess), so that each backbone received input normalized exactly as it was during its original ImageNet pretraining. Validation and test subsets were left unaugmented — resized and normalized only — so the reported performance reflects how each model behaves on realistic, unmodified images. Figure 3.2 illustrates this augmentation pipeline: starting from an original hand-gesture image, random rotation, horizontal flipping, zoom, brightness adjustment, and positional shift are each applied to generate additional training variants, enlarging the effective diversity of the training subset without collecting any new images.

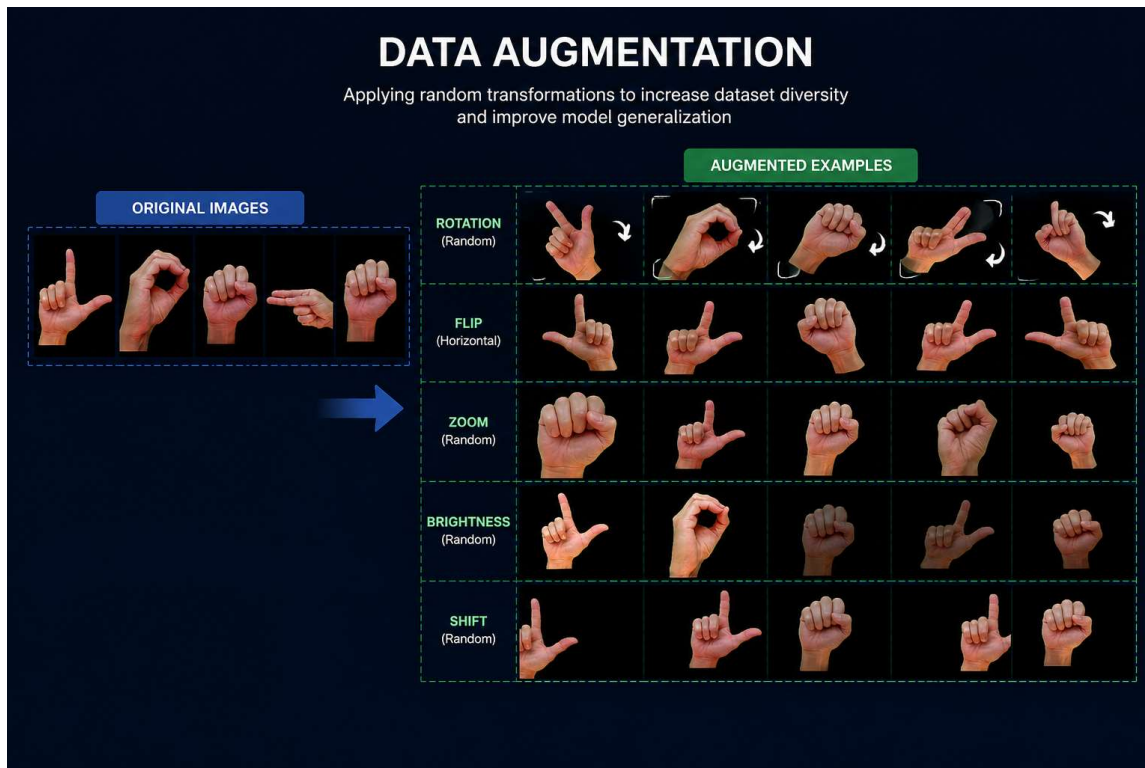


Figure 3.2: Data Augmentation Examples (Rotation, Flip, Zoom, Brightness, and Shift)

3.5 Model Architectures

This study compares six architectures: 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, a 50-layer CNN, uses residual (skip) connections that let a layer's input bypass intermediate layers and add directly onto the output — a design that solves the vanishing-gradient problem that would otherwise limit training in very deep networks [7]. Here, its ImageNet-pretrained convolutional base (about 24.1 million parameters) was frozen while a new classification head was trained for 36-class ASL fingerspelling recognition. MobileNetV2, built for resource-constrained devices, uses depthwise-separable convolutions and inverted residual blocks to sharply cut the computation needed relative to standard convolutions without giving up much accuracy [8]; at roughly 2.6 million parameters it is by far the smallest of the six models (13.03 MB), which shows directly in it being among the smallest and fastest models evaluated (Section 4.7). EfficientNetB0 uses compound scaling — a single coefficient that balances network depth, width, and input resolution — to aim for the best trade-off between accuracy and size [9]; the variant used here has roughly 4.4 million parameters. The Vision Transformer (ViT-Base/16, vit_base_patch16_224) splits each 224×224 image into a sequence of 16×16 patches, linearly embeds each one, and runs a stack of standard Transformer encoder blocks with multi-head self-attention across the whole patch sequence [10]; at about 85.8 million parameters — by far the largest of the six models — and with no built-in convolutional inductive

1 bias, the pretrained model was fully fine-tuned, all layers trainable, with its classification head swapped out for the 36-class ASL task. The Swin Transformer (Swin-Tiny, swin_tiny_patch4_window7_224) is a hierarchical vision transformer that moves an image through four successive stages, progressively merging patches to build increasingly abstract features in a way that echoes a CNN's hierarchy; within each stage it alternates between window-based multi-head self-attention (W-MSA) and shifted-window multi-head self-attention (SW-MSA), letting it capture both fine local detail and broader context across the hand and image [11]. The pretrained swin-tiny variant used here (about 27.5 million parameters) was likewise fully fine-tuned, with its classification head replaced for the 36-class task.

22 The Hybrid MobileNetV2-Swin model proposed in this thesis, finally, is a dual-branch fusion architecture built specifically to combine the strengths of the two model families above. 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 one 2,048-dimensional fused representation, then passed through a classification head made up of a fully connected layer (2,048 → 512), a ReLU activation, a Dropout layer (p = 0.5), and a final fully connected layer (512 → 36). The resulting model, at roughly 30.8 million parameters, is trained end to end — every layer in both backbones is fine-tuned jointly — so the classification head can learn how best to combine MobileNetV2's local convolutional features with the Swin Transformer's global self-attention features. As Chapter 4 reports, this Hybrid model reaches the highest test accuracy of all six (99.80%).

78 Pairing MobileNetV2 specifically with the Swin Transformer, rather than any other convolutional or transformer backbone, was a deliberate choice, made for three reasons. First, the two networks complement rather than duplicate each other: MobileNetV2's depthwise-separable convolutions are especially good at picking up local, fine-grained texture and edges — exactly the kind of detail that separates visually similar ASL handshapes like the digit '0' from the letter 'o' — while the Swin Transformer's shifted-window self-attention captures broader relationships across the whole hand, so the fused representation carries both kinds of evidence instead of doubling up on one. Second, both networks are comparatively lightweight on their own (2.60 million and 27.55 million parameters, respectively) next to alternatives such as the standard Vision Transformer (85.83 million parameters); fusing them still produces a combined model of 30.81 million parameters — smaller than a standalone ViT, yet more accurate than either backbone alone — a better accuracy-to-size trade-off than pairing two larger networks would have given. Third, evidence from outside sign language pointed the same way: a hybrid model using the Swin Transformer as a feature extractor achieved a substantial accuracy gain over conventional CNN baselines on a medical image classification task [12], suggesting a Swin-based hybrid might likewise outperform CNN-only baselines on the fine-grained visual discrimination fingerspelling requires. Chapter 4 bears

this reasoning out empirically: the Hybrid model reaches the highest accuracy of all six compared (99.80%), with only one residual misclassification across the full 504-image test set — recovering one of the two errors the standalone Swin Transformer still made on its own.

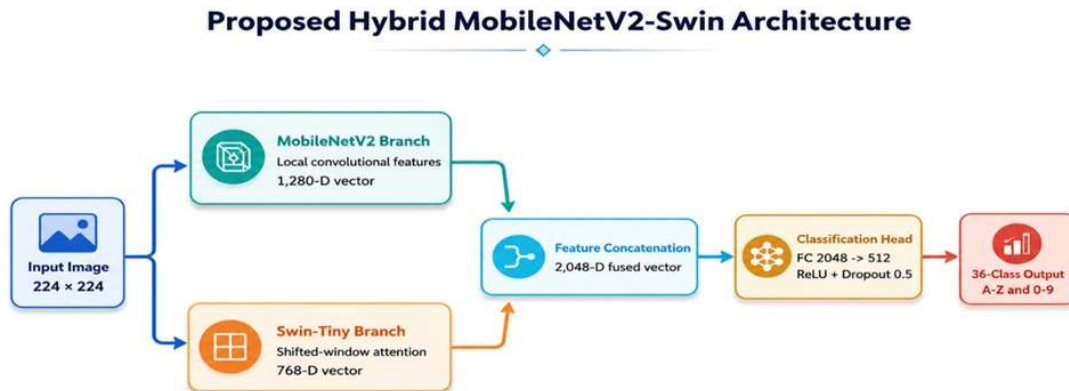


Figure 3.3: Hybrid MobileNetV2-Swin Dual-Branch Architecture

Table 3.2 pulls together the framework, parameter count, fine-tuning strategy, and input size of the six architectures described above.

Table 3.2: Summary of the Six Deep Learning Models Used

Model	Framework	Parameters	Fine-Tuning Strategy	Input Size
ResNet50	TensorFlow / Keras	24.12 M	Frozen ImageNet backbone + trainable head	224 × 224
MobileNetV2	TensorFlow / Keras	2.60 M	Frozen ImageNet backbone + trainable head	224 × 224
EfficientNetB0	TensorFlow / Keras	4.39 M	Frozen ImageNet backbone + trainable head	224 × 224
Vision Transformer (ViT-Base/16)	PyTorch (timm)	85.83 M	Full fine-tuning, pretrained	224 × 224
Swin Transformer (Swin-Tiny)	PyTorch (timm)	27.55 M	Full fine-tuning, pretrained	224 × 224
Hybrid MobileNetV2-Swin (proposed)	PyTorch (timm)	30.81 M	Full fine-tuning, both branches pretrained	224 × 224

3.6 Word Formation and Automatic Spelling Correction (Demonstration)

Besides classifying individual images, this thesis includes a demo NLP application that shows the ability to recognize and form complete words. It illustrates the real-world implications of the theoretical gaps found in Section 2. 3 and should not be viewed as the main contribution of the paper. There are two ways to interact with the program. Both of them have been implemented

inGradio multi-model app that was discussed in Section 4. 8: Sequential building of a word via single images: the program predicts one image at a time, the predicted letter gets joined with the word under construction (`append_predicted_letter`). When it feels the predicted letter was wrong (`remove_last_letter`) or even a completely wrong one (`clear_current_word`), the user has an option to go back or reset the state altogether.

Creation of a word through images in a batch: the user uploads an ordered sequence of pictures in one go; a function called `predict_multiple_images` classifies each one with the help of the model previously trained, then the predicted letters in the order of uploading form the word and are displayed together with each picture prediction and the confidence score.

When the raw word is put together, in both scenarios it is handed over to a general `correct_spelling` function which is wrapped around the TextBlob's spelling-correction model. In case a digit was part of the word then it is simply kept as is because digit sequences do not make proper English words and because of this not a thing to be corrected otherwise it is turned to lowercase letters, fed to `TextBlob(word).correct()` which makes use of TextBlob's statistical model to identify the most likely desired English word, substitute the misclassified characters, and outputs the correct upper-case result. Practically, this means, for example, a wrong prediction for a letter can sometimes still be right in the word the user thought of because of the remaining correctly recognized letters pointing clearly to a valid word.

```
def correct_spelling(raw_word):
    raw_word = (raw_word or "").strip()
    if not raw_word:
        return ""
    if any(character.isdigit() for character in raw_word):
        return raw_word.upper() # do not "correct" digit sequences
    try:
        return str(TextBlob(raw_word.lower()).correct()).upper()
    except Exception:
        return raw_word.upper()

def predict_multiple_images(uploaded_files, model_name, progress=gr.Progress()):
    predicted_letters = []
    for image_index, uploaded_file in enumerate(uploaded_files, start=1):
        image = Image.open(get_uploaded_file_path(uploaded_file)).convert("RGB")
        probabilities, model_class_names = predict_probabilities(image, model_name)
        predicted_index = int(np.argmax(probabilities))
        predicted_letters.append(model_class_names[predicted_index].upper())
    raw_word = "".join(predicted_letters)
    corrected_word = correct_spelling(raw_word)
    return raw_word, corrected_word, gallery_results, status_message
```

This correction stage works as follows: the raw, potentially misspelled word assembled from individual letter predictions is spell-checked, TextBlob analyzes the surrounding context to generate candidate suggestions, and the best-matching English word is selected as the final corrected output.

3.7 Proposed Algorithm

Algorithm 3.1 spells out formally the end-to-end pipeline applied to each of the six models, running from raw image input through training and evaluation to the word-formation and spelling-correction stage.

Algorithm 3.1: Unified ASL Recognition, Comparison, and Word-Formation Pipeline

Input: Image set $X = \{x_1, x_2, \dots, x_n\}$ with labels $Y = \{y_1, y_2, \dots, y_n\}$
 Model M in $\{\text{ResNet50, MobileNetV2, EfficientNetB0, ViT, Swin, Hybrid MobileNetV2-Swin}\}$
 Output: Predicted labels $Y_{\text{hat_test}}$, evaluation metrics, model size/inference time, and (optionally) a spell-corrected word from a sequence of images

```

1: Set global random seed (42) for Python, NumPy, and TensorFlow/PyTorch;
   verify GPU availability before proceeding
2: Load  $X$  from its 36 class folders (digits 0-9, letters a-z) and verify
   that all 36 expected classes are present
3: Split  $X$  once into  $X_{\text{train}}$  (70%),  $X_{\text{val}}$  (10%),  $X_{\text{test}}$  (20%) using the
   fixed random seed (42); reuse this identical split for every model  $M$ 
4: Build data loaders for  $M$ : resize images to 224x224 and apply  $M$ -specific
   normalisation; augment  $X_{\text{train}}$  only (rotation, shift/zoom/shear/
   brightness for the three CNNs; rotation + colour jitter for ViT, Swin,
   and Hybrid); leave  $X_{\text{val}}$  and  $X_{\text{test}}$  unaugmented
5: Initialise  $M$  with pretrained weights (ImageNet for the CNNs and the
   ViT/Swin backbones)
6: If  $M$  is a CNN (ResNet50, MobileNetV2, EfficientNetB0): freeze its
   backbone and attach a new head (GAP -> Dense(256) -> Dropout(0.5) ->
   Dense(36))
   Else (ViT, Swin, Hybrid): replace the classification head and
   fine-tune all layers end to end
7: Compile  $M$  with Adam (CNNs) or AdamW (ViT, Swin, Hybrid), learning rate
   = 0.0001, and cross-entropy loss
8: for epoch = 1 to 30 do
9:   Train  $M$  on  $X_{\text{train}}$  for one epoch
10:  Evaluate  $M$  on  $X_{\text{val}}$ ; save weights if validation accuracy improves
11:  Step the ReduceLROnPlateau scheduler on validation loss (factor
    0.3, patience 4)
12:  if validation loss has not improved for 7 consecutive epochs then
13:    stop training early (Early Stopping)
14:  end if
15: end for
16: Load the best-performing saved weights of  $M$ 
17: Predict  $Y_{\text{hat\_test}} = M(X_{\text{test}})$ ; record total and per-image inference
   time (3 repeats, 1 warm-up)
18: Compute Accuracy, weighted Precision/Recall/F1, macro F1 from
   ( $Y_{\text{test}}$ ,  $Y_{\text{hat\_test}}$ ); record model size and parameter count
19: Save the trained model, class names, confusion matrix, training
   curves, and metrics (JSON) for  $M$ 
20: # Word-formation and spelling-correction extension (demonstration only):
21: for each image  $x_i$  in an ordered user-supplied sequence do
22:   predicted_letter_i = argmax( $M(x_i)$ )
23: end for
24: raw_word = concatenate(predicted_letter_1, predicted_letter_2, ...,
   predicted_letter_k)
25: if raw_word contains a digit then corrected_word = raw_word
26: else corrected_word = TextBlob(raw_word.lower()).correct()
27: return  $Y_{\text{hat\_test}}$ , Accuracy, Precision, Recall, F1, Inference Time,
   Model Size, raw_word, corrected_word
  
```

3.8 Evaluation Metrics

The six trained models were compared using the classification metrics below — where TP, TN, FP, and FN stand for true positives, true negatives, false positives, and false negatives — together with total and per-image inference time, model size, and parameter count:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad \text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{F1-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

Weighted precision, recall, and F1-score average each class's metric in proportion to its number of test samples (support), while the macro F1-score weights all 36 classes equally regardless of support — giving a complementary view of performance across rare and common classes alike. Inference time was measured as the average total time needed to process all 504 test images, repeated three times after a single warm-up pass (to exclude one-off GPU/CUDA initialization overhead), along with the corresponding average time per image; model size is the size in megabytes of the saved model file on disk, and parameter count is the total number of trainable and non-trainable parameters in the model.

CHAPTER 4

Experimental Results and Discussion

4.1 Introduction

This chapter presents and discusses the results of applying the six deep learning models from Chapter 3 to the ASL fingerspelling recognition task. It opens with the experimental setup and training configuration, then moves through the comparative results across all six models, a detailed per-class error analysis, a discussion of the findings against the related work reviewed in Chapter 2, the study's limitations, a dedicated inference-time analysis, and a demonstration of the word-formation and spelling-correction application.

4.2 Experimental Setup

All six experiments ran on the same NVIDIA A100-SXM4-40GB GPU, so timing comparisons between models are directly meaningful (Table 4.2 confirms an identical 504-image test set and GPU across all six metrics files). The Vision Transformer, Swin Transformer, and Hybrid MobileNetV2-Swin model were implemented in PyTorch using the timm library, while the three CNN models were built in TensorFlow/Keras. Evaluation metrics were computed with scikit-learn, and a unified Gradio interface was built for interactive testing of all six models: upload an image or a sequence of images, and view the predicted class, confidence score, and — for sequences — the assembled, spelling-corrected word.

All six models were also trained under the same core configuration, summarized in Table 4.1, on the same NVIDIA A100-SXM4-40GB GPU.

Table 4.1: Training Configuration

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, min_lr = 1e-6) — PyTorch models
Optimizer	Adam (CNNs) / AdamW, weight decay = 0.01 (ViT, Swin, Hybrid)
Loss Function	Categorical / Cross-Entropy Loss
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)

4.3 Comparative Results

This section reports what happened when each of the six models was applied to the prepared dataset, and compares them directly against one another. ResNet50's validation accuracy peaked over the course of 30 epochs before its frozen-backbone classification head converged; on the held-out test set, the restored best model reached 92.46% accuracy (test loss 0.3774), a weighted precision of 93.36%, and a weighted F1-score of 92.42% — a solid but not top-tier CNN baseline. MobileNetV2, despite having roughly a tenth of ResNet50's parameters, achieved a broadly similar 91.87% test accuracy (test loss 0.3565, weighted F1-score 91.93%), showing that a lightweight, efficiency-focused architecture can stay competitive with a much larger CNN on this task. EfficientNetB0 came in lowest of the six at 89.88% test accuracy (test loss 0.5200, weighted F1-score 89.51%); Section 4.4 shows this stems mainly from a wider spread of per-class weaknesses — 18 of 36 classes below an F1-score of 0.95, including two classes, n and v, with a recall of just 0.4286 — rather than any single dominant confusion. The Vision Transformer converged to a substantially higher test accuracy than any CNN, reaching 98.81% with a test loss of 0.0427 and a weighted F1-score of 98.81%; per-class analysis (Section 4.4) shows its performance is essentially perfect apart from the digit '0' and the letter 'o', confirming that self-attention generalizes well to this task even without the convolutional inductive bias the other five models have, though at the cost of being the largest (85.8 million parameters, 327.46 MB) and slowest (Section 4.7) model in the comparison. The Swin Transformer improved on this further, reaching 99.60% test accuracy with a notably low test loss of 0.0189 and a weighted F1-score of 99.60%; only two classes — the digit '0' and the letter 'o', each with an F1-score of 0.9286 — fell short of a perfect score. Unexpectedly, and against the assumption that a more sophisticated hierarchical transformer would necessarily be slower than a plain CNN, the Swin Transformer was also the fastest of all six models on both total and per-image inference time (Section 4.7). Finally, the proposed Hybrid MobileNetV2-Swin model produced the best overall performance of the six: 99.80% test accuracy, the lowest test loss in the comparison (0.0112), and a weighted F1-score of 99.80%. Of the 504 test images, only a single one was misclassified across the entire test set — one true digit '0' predicted as the letter 'o' — confirming that fusing MobileNetV2's convolutional features with the Swin Transformer's self-attention features gives a small but measurable accuracy gain over either backbone alone.

Bringing all six results together, the proposed Hybrid MobileNetV2-Swin model leads on every metric, reaching 99.80% accuracy and a 99.80% weighted F1-score — just ahead of the Swin Transformer (99.60%) and the Vision Transformer (98.81%), and well clear of the three CNN baselines: ResNet50 (92.46%), MobileNetV2 (91.87%), and EfficientNetB0 (89.88%, the lowest of the six). The three transformer-based models (Hybrid, Swin, ViT) sit in a clearly separated top tier, each above 98.8% accuracy, while the three CNN baselines form a second tier between 89.9%

and 92.5% — a narrower spread among themselves than the top tier shows, but still a clear six-to-eight-point gap below even the weakest transformer-based model. Table 4.2 and Figures 4.1–4.2 present the full results.

Table 4.2: Performance Comparison of the Six Models on the Test Set

Rank	Model	Test Images	Test Loss	Accuracy	Weighted Precision	Weighted Recall	Weighted F1	Macro F1
1	Hybrid MobileNetV2-Swin	504	0.0112	99.80%	99.81%	99.80%	99.80%	99.80%
2	Swin Transformer	504	0.0189	99.60%	99.60%	99.60%	99.60%	99.60%
3	Vision Transformer	504	0.0427	98.81%	98.83%	98.81%	98.81%	98.81%
4	ResNet50	504	0.3774	92.46%	93.36%	92.46%	92.42%	92.42%
5	MobileNetV2	504	0.3565	91.87%	92.86%	91.87%	91.93%	91.93%
6	EfficientNetB0	504	0.5200	89.88%	91.42%	89.88%	89.51%	89.51%

All six models were tested on the same 504-image test set (14 images per class), on the same NVIDIA A100-SXM4-40GB GPU.

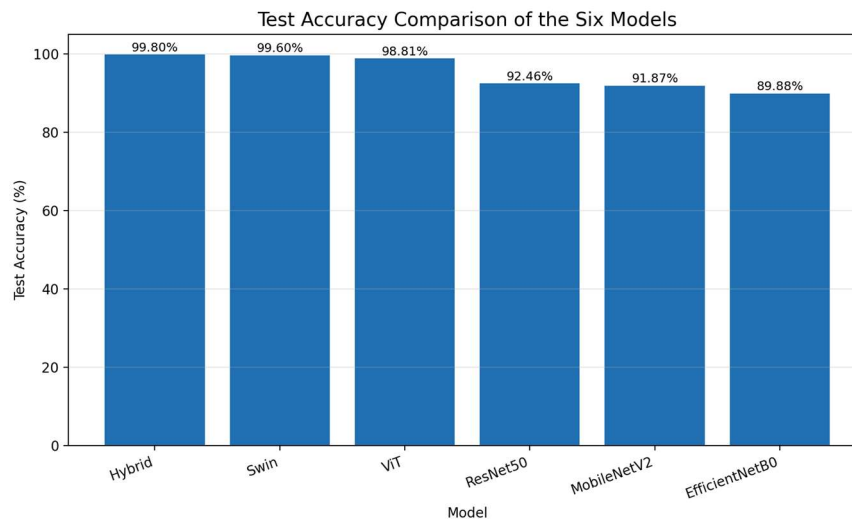


Figure 4.1: Test Accuracy Comparison of the Six Models

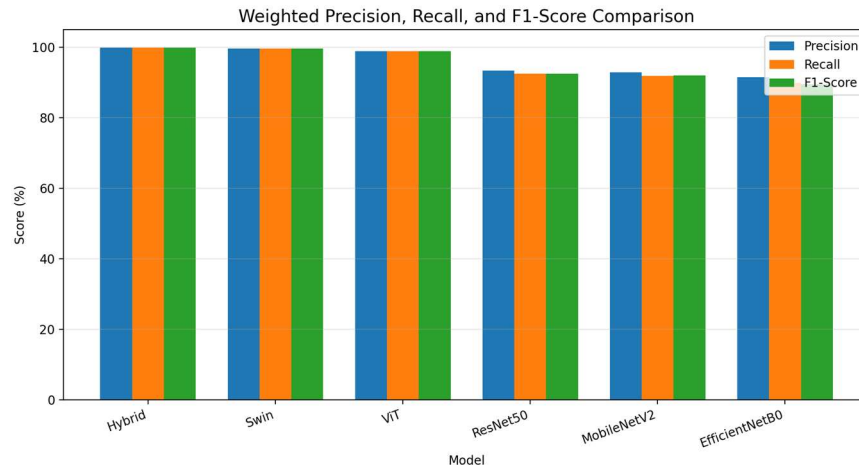


Figure 4.2: Weighted Precision, Recall, and F1-Score Comparison of the Six Models

4.4 Per-Class Error Analysis

A detailed classification report was generated for every model across all 36 classes. The single most consistent pattern across the whole comparison: the digit '0' and the letter 'o' — both closed-circle handshapes in ASL, and consequently hard to tell apart visually — are the pair most often confused by every one of the six models, and essentially the only source of error in the three transformer-based ones. Table 4.3 reports each model's F1-score on these two classes, together with how many of the 36 classes scored below an F1-score of 0.95, as a measure of how widely each model's errors are spread.

Table 4.3: 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, 0.9286)
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, 0.6000)

The Hybrid model misclassified only a single image across the entire 504-image test set: one true digit '0' predicted as the letter 'o'. The Swin Transformer made two errors, out of 504 — a '0' predicted as 'o', and an 'o' predicted as '0'. The Vision Transformer's confusion runs in both directions too: three true '0' images were predicted as 'o' and two true 'o' images as '0', plus one further isolated error (a digit '6' predicted as 'w'), so the '0'/'o' pair is not quite this model's only

source of error, though it remains by far the dominant one. The three CNN models, in contrast, show a noticeably wider spread of confusion beyond '0' and 'o'. ResNet50 also mixes up the digit pair 4 and 5, and shows low recall for the letter 'n' (0.5714) — six of fourteen true 'n' images were misclassified as 'm', not 'o'. MobileNetV2's most severe confusion sits squarely between '0' and 'o' (six of fourteen true 'o' images called '0'), alongside notably low precision for the letter 't' (0.6190). EfficientNetB0 shows the broadest weakness of the six: the letters 'n' and 'v' each achieve a recall of only 0.4286 (six of fourteen correct), with true 'n' images most often mistaken for 'm' and true 'v' images for '7', on top of the persistent '0'/'o' confusion. Altogether, this pattern suggests that self-attention, as used by the Vision Transformer, Swin Transformer, and Hybrid model, resolves visually near-identical handshapes across the whole class space substantially better than the CNN baselines do — not just on average — and that the residual '0'/'o' confusion even the best models show reflects a genuine, physically grounded visual ambiguity in the ASL alphabet itself, rather than a weakness specific to any one model.

Figure 4.3 shows the full 36×36 confusion matrix for every model on the complete 504-image test set (14 images per class), with all six models arranged together for direct comparison. Read alongside Table 4.3 and the discussion above, this figure makes plain exactly how each model's errors are spread: the three CNN baselines (ResNet50, MobileNetV2, EfficientNetB0) show a visibly wider scatter of small off-diagonal counts across several class pairs, while the three transformer-based models (Vision Transformer, Swin Transformer, Hybrid MobileNetV2-Swin) present an almost perfectly clean diagonal, with errors concentrated overwhelmingly in the single '0'/'o' cell.

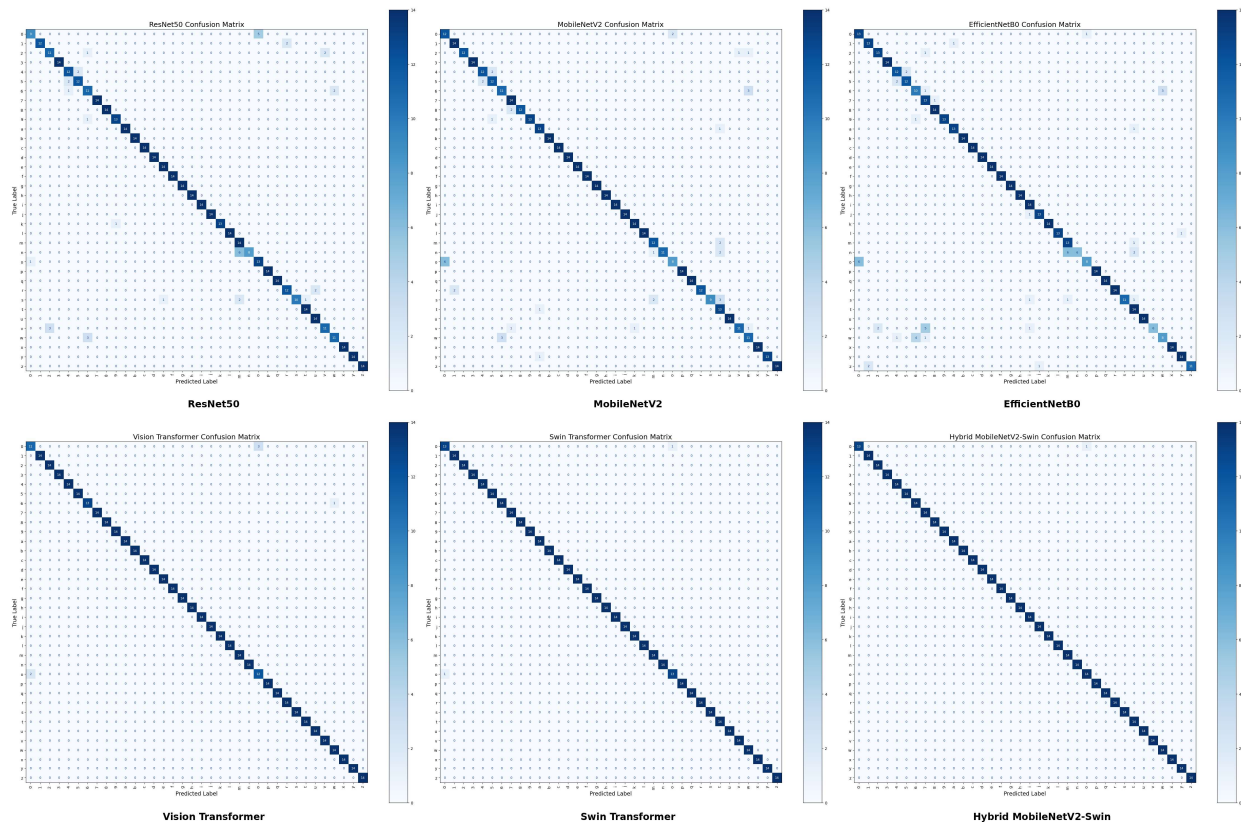


Figure 4.3: Confusion Matrices of the Six Models (36 Classes, Test Set)

4.5 Discussion

4.5.1 Architectural Advantage of the Transformer-Based Models

The steady edge the Vision Transformer, Swin Transformer, and Hybrid model hold over the three CNN baselines is best explained by a real architectural difference, not incidental factors. A convolutional filter has a fixed, local receptive field and applies the same learned weights uniformly across every position in an image — excellent for picking up local texture and edges, but limited when it comes to relating distant parts of the hand to one another. Self-attention, by contrast, lets a model weight the relationship between different regions of the hand adaptively, based on content, and the Swin Transformer's hierarchical, shifted-window design lets this information merge progressively across ever-larger regions of the image while keeping much of a CNN's computational efficiency. For a fine-grained task like ASL fingerspelling — where the difference between two classes, say the digit '0' and the letter 'o', may come down to the precise position and curvature of a few fingers — this ability to relate distant hand regions adaptively appears to offer a genuine, consistent advantage, reflected directly in the higher, more uniform per-class F1-scores reported in Section 4.4. The Hybrid model's further, if modest, improvement over the standalone Swin Transformer (99.80% versus 99.60% accuracy; one residual error versus two) supports the idea that MobileNetV2's local convolutional features and the Swin Transformer's

global self-attention features are genuinely complementary rather than redundant — combining them recovered one of the two errors the standalone Swin Transformer still made on its own.

4.5.2 A Corrected Accuracy-Speed Trade-off

One central and rather counter-intuitive finding here concerns speed. Transformer-based models, given their large parameter counts and quadratic self-attention cost, are commonly assumed to be necessarily slower than lightweight CNNs. That assumption doesn't hold up in this study's results (full figures in Section 4.7): the standalone Swin Transformer was actually the fastest of all six models, both in total inference time (1.608 s across the full 504-image test set) and per-image latency (3.191 ms) — faster even than the lightweight MobileNetV2 (1.814 s; 3.600 ms) and EfficientNetB0 (1.766 s; 3.504 ms). The Vision Transformer, on the other hand, ranked only third in accuracy yet was the slowest model overall (3.852 s; 7.642 ms per image), which fits with it also being by far the largest (85.8 million parameters; 327.46 MB): a standard Vision Transformer computes full self-attention across every patch, without the Swin Transformer's window-based restriction, so it misses out on that same architectural efficiency. The Hybrid model, running both a MobileNetV2 and a Swin Transformer branch on every image, lands closer to the middle of the pack on speed (2.513 s; 4.986 ms) — the price of its small extra accuracy gain. This finding revises the usual accuracy-speed trade-off story for this task: the right recommendation, at least for this six-model comparison, is not “pick a CNN if you need speed,” but rather “pick the transformer architecture whose design matches your speed needs,” since the Swin Transformer's hierarchical, window-restricted self-attention makes it both the most accurate model in its size class and, unexpectedly, the fastest model overall.

4.5.3 Comparison with Related Work

These results broadly agree with, and build on, the recent literature reviewed in Chapter 2. The 99.80% accuracy of the proposed Hybrid model comes close to the 99.97% reported by Aly and Fathi's hybrid CNN–Vision Transformer model [13], and lines up with the strong results reported for the Swin Transformer architecture on other sign-language datasets [32] — despite this study using a considerably smaller dataset (2,515 images versus 87,000 in [13]). This supports the conclusion that transformer-based and hybrid attention-augmented architectures offer a genuine advantage on this task even with comparatively limited data. The finding that fusing two feature extractors improves accuracy also matches the two-stream CNN literature [14], [15], extending it to a new fusion axis — a CNN paired with a hierarchical transformer, rather than two temporal frames or two CNN backbones. The CNN results here (92.46%, 91.87%, 89.88%), by contrast, fall noticeably short of the near-perfect accuracies reported by CNN-only studies trained on much larger datasets, such as the 99.99% ResNet50 reached in [20] on a dataset over thirty times larger. This comparison suggests the relatively modest dataset used in this study constrained the CNN

models' ability to generalize more than it constrained the transformer-based models, which appear to have made more effective use of their pretrained representations within the same data budget — consistent with the well-documented tendency of self-attention architectures to transfer strong pretrained representations efficiently even with limited fine-tuning data.

4.6 Limitations of the Study

- This study covers only static fingerspelling images of ASL letters and digits; it does not address dynamic gestures or full sentences, which would call for additional temporal modelling.
- The dataset itself is relatively small (2,515 images) and was captured mostly under fairly uniform lighting and backgrounds, which may limit how well these results generalize to messier real-world conditions — poor lighting, cluttered backgrounds, or a wider range of skin tones.
- The spelling-correction stage depends on TextBlob's general-purpose English dictionary and statistical model; it was not systematically tested against a benchmark of deliberately noisy letter sequences, and it is expected to be less effective on words outside standard English vocabulary — proper nouns, technical terms, or non-English words.
- No field test of the models or the word-formation/spelling-correction pipeline was carried out in a live, real-time deployment; inference-time evaluation was confined to the cloud GPU environment used for the experiments.
- No advanced hyperparameter tuning — grid search, evolutionary optimization, or similar — was carried out; fixed, uniform settings were used across all six models specifically to keep the comparison fair, which may mean each model's absolute best possible performance was not fully reached.

4.7 Time Analysis

Table 4.4 gathers model size, parameter count, and inference time for all six approaches; model size appears in Figure 4.4, parameter count in Figure 4.5, total inference time in Figure 4.6, per-image inference time in Figure 4.7, and the accuracy–speed trade-off graphically in Figure 4.8.

44

Table 4.4: Model Size, Parameter Count, and Inference-Time Comparison

Model	Parameters	Model Size (MB)	Total Inference Time (s)	Time per Image (ms)
Swin Transformer	27.55 M	105.15	1.608	3.191
EfficientNetB0	4.39 M	20.11	1.766	3.504
MobileNetV2	2.60 M	13.03	1.814	3.600
ResNet50	24.12 M	96.72	2.289	4.542
Hybrid MobileNetV2-Swin	30.81 M	117.86	2.513	4.986
Vision Transformer	85.83 M	327.46	3.852	7.642

Ordered by total inference time, lowest first. All timings were measured on the same NVIDIA A100-SXM4-40GB GPU across three repeated passes of the full 504-image test set, after a single warm-up pass.

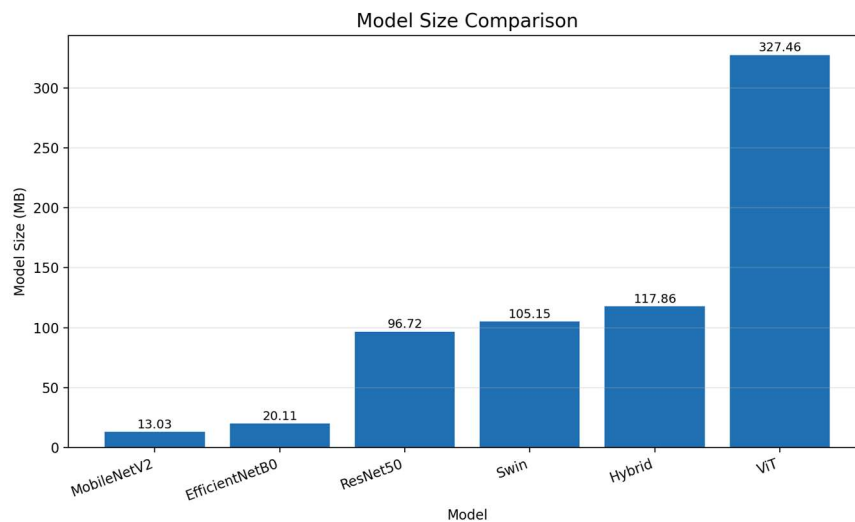


Figure 4.4: Model Size Comparison of the Six Models

130

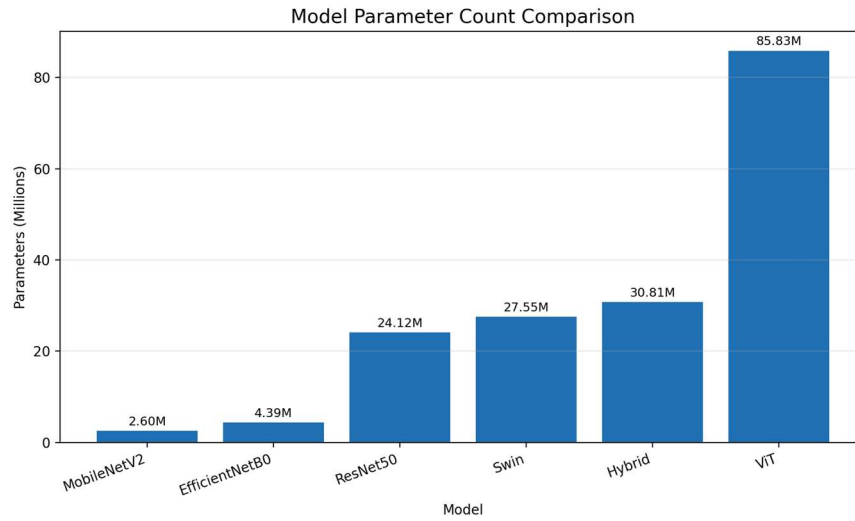


Figure 4.5: Model Parameter Count Comparison of the Six Models

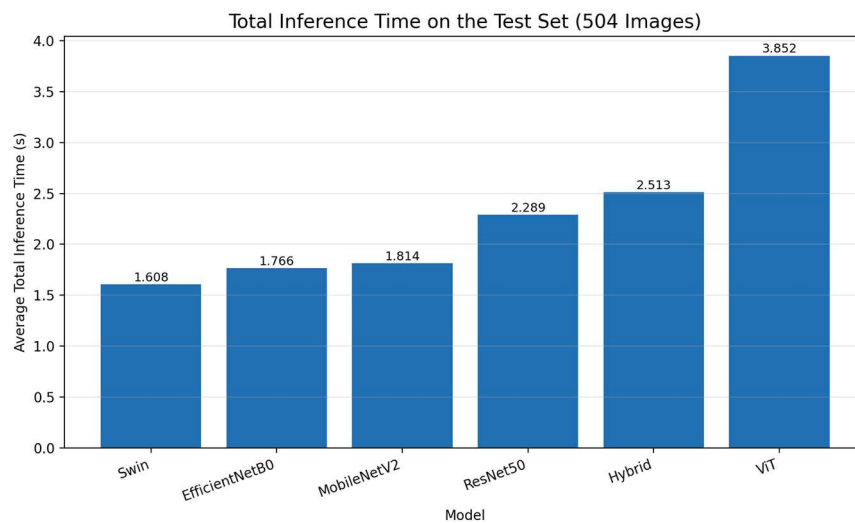


Figure 4.6: Total Inference Time Comparison of the Six Models on the Test Set

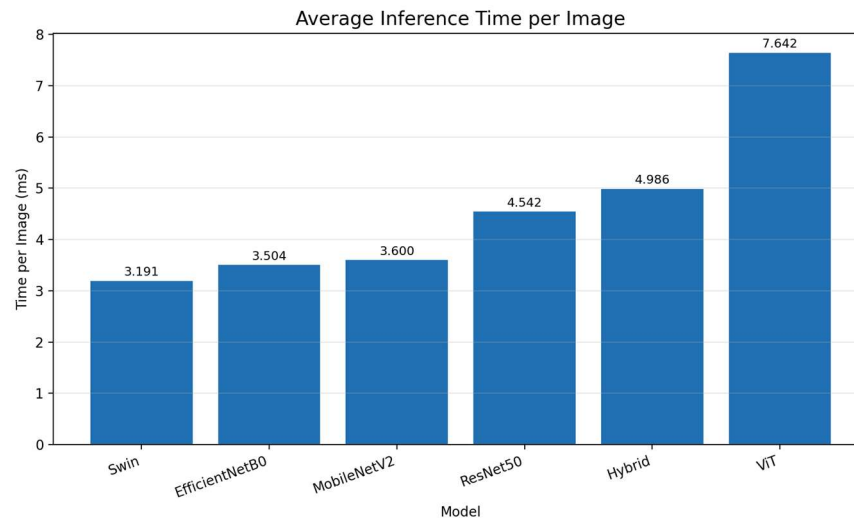


Figure 4.7: Average Inference Time per Image of the Six Models

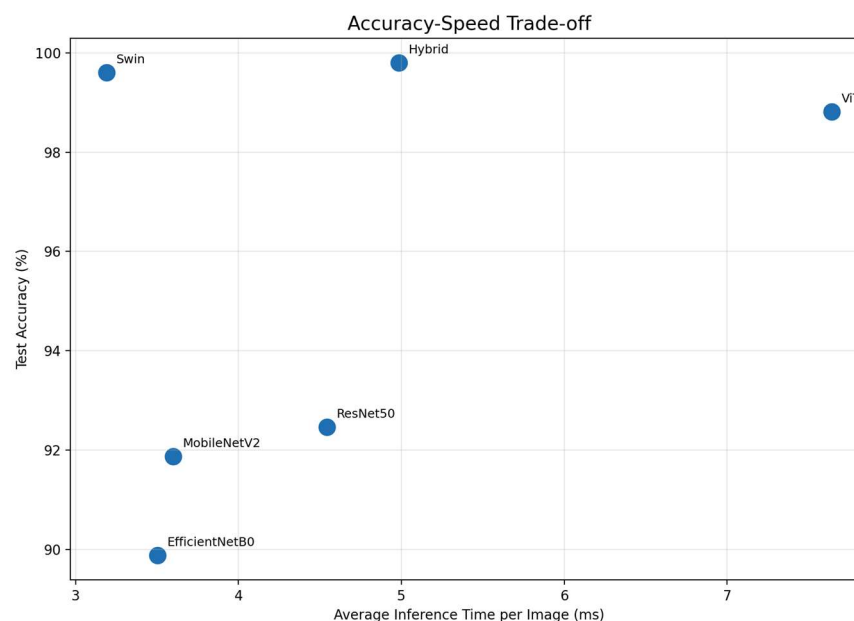


Figure 4.8: Accuracy-Speed Trade-off of the Six Models

Table 4.4 and Figure 4.8 make the trade-offs clear. The Swin Transformer sits in the most favourable spot of all six: fastest overall and second most accurate, arguably the single best all-round choice if only one model could be deployed. The proposed Hybrid model is the most accurate of the six but not the fastest, reflecting the cost of running two backbones per prediction; it is the model to choose when maximum accuracy matters most and a roughly 2.5-second total inference time (about 5 ms per image) is acceptable for the intended use. MobileNetV2 stays the smallest and among the fastest of the six, making it the sensible pick for tightly resource-constrained or on-device deployment where a six-to-eight-point accuracy gap relative to the transformer-based models is a trade-off worth accepting. The Vision Transformer, being

simultaneously the largest, the slowest, and only the third most accurate, is not recommended for practical deployment here unless its specific architectural properties — interoperability with other ViT-based pipelines, for example — are independently needed.

4.8 Word-Formation and Spelling-Correction Application

To show the practical value of the models compared above, all six were folded into a single interactive Gradio application (“ASL Multi-Model Recognition System”). A dynamic model loader keeps only the currently selected model in memory, clearing the previous one via `tf.keras.backend.clear_session()`, `gc.collect()`, and `torch.cuda.empty_cache()`, so that all six models — spanning two different deep learning frameworks — can live in one application without exhausting GPU or system memory. The application offers three tabs:

- **Single Image / Camera:** upload an image or capture one from a webcam, pick any of the six models from a dropdown, and get back the predicted letter, a confidence score, and the top five candidate classes; the predicted letter can be appended to a running word, and that word corrected, trimmed of its last letter, or cleared, all directly from the interface.
- **Multiple Images to Word:** upload an ordered sequence of ASL images in one action; the application classifies each with the selected model, joins the predicted letters into a raw word, applies automatic spelling correction, and shows both the raw and corrected word next to a gallery of the individual per-image predictions and confidence scores.
- **Models Comparison:** an automatically generated, always-current table — the same underlying data as Table 4.2 — showing accuracy, precision, recall, F1-score, test loss, per-image inference time, parameter count, and model size for all six models, built directly from the saved metrics JSON files produced during evaluation.

Single Image / Camera Multiple Images to Word Models Comparison

Upload an ASL Image or Use Camera

Top Five Predictions

Prediction	Confidence
A	53%
Y	9%
4	8%
I	3%
0	3%

Predicted Letter: A

Prediction Confidence: 52.97%

Predict Image

Word Formation

Raw Predicted Word:

Corrected Word:

Append Predicted Letter Remove Last Letter Correct Word Clear Word

Figure 4.9: Gradio Application — Single Image / Camera Tab

Figure 4.10 shows the Multiple Images to Word tab in action: five uploaded letter images (E, H, L, L, O) are classified independently, joined into the raw word "EHLLO", and corrected by the TextBlob stage into "HELLO" — a direct illustration of the single-letter error recovery discussed above.

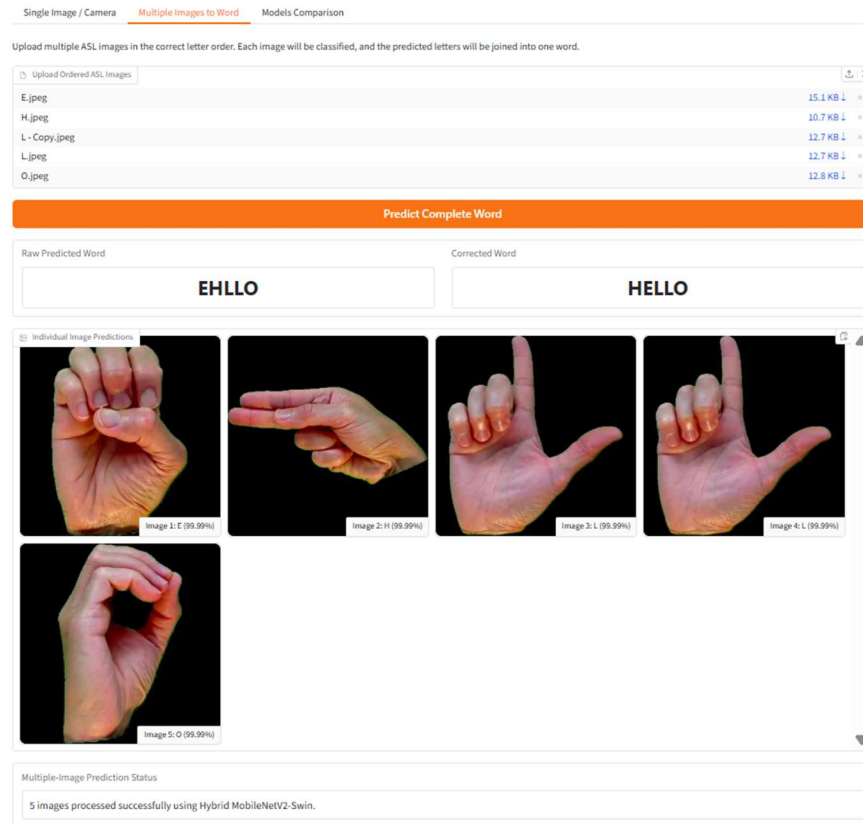


Figure 4.10: Gradio Application — Multiple Images to Word Tab

Figure 4.11 shows the Models Comparison tab, an always-current table built directly from the saved evaluation metrics, letting a user select any of the six models and immediately see its accuracy, precision, recall, F1-score, inference time, parameter count, and model size.

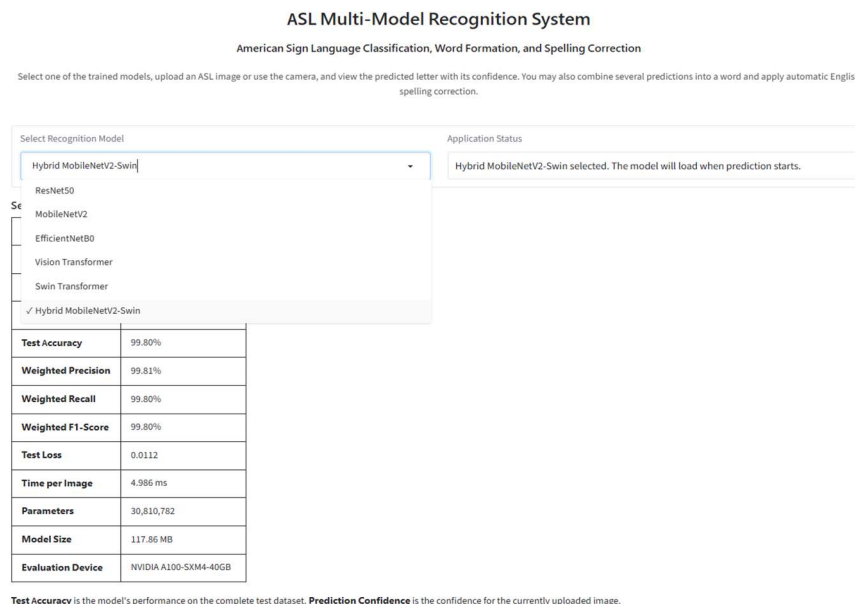


Figure 4.11: Gradio Application — Models Comparison Tab

In a working system, this application demonstrates that the recognition accuracy of the top three models (Section 4.4) is high enough to support a genuinely usable word-formation and spelling-correction workflow: an occasional single-letter misclassification — most often the digit '0' / letter 'o' pair discussed in Section 4.5 — can frequently still be recovered as the user's intended word by the TextBlob-based correction stage, so long as the remaining letters in the sequence unambiguously point to a valid English word. This directly answers the fourth research gap identified in Section 2.3: the absence, in prior literature, of a working demonstration that pairs a classification benchmark with multi-letter word assembly and automatic spelling correction.

7

CHAPTER 5

Conclusion and Future Work

5.1 Summary of the Study

This thesis set out to design, build, and rigorously evaluate a comparative deep-learning framework for recognizing American Sign Language fingerspelling, then extend it with practical word-formation and spelling-correction capabilities, aiming to identify the model best suited — on accuracy and practicality both — to an assistive recognition system. Six architectures — ResNet50, MobileNetV2, EfficientNetB0, a Vision Transformer, a Swin Transformer, and a Hybrid MobileNetV2-Swin fusion model designed for this thesis — were trained and evaluated under one shared protocol on a 36-class ASL dataset (digits 0–9, letters a–z, 2,515 images), using accuracy, weighted precision/recall/F1-score, macro F1-score, inference time, model size, and parameter count as the basis of comparison. The proposed Hybrid model came out on top overall, at 99.80% test accuracy and a 99.80% weighted F1-score — narrowly ahead of the Swin Transformer (99.60%) and the Vision Transformer (98.81%), and well ahead of the three CNN-based models. Beyond classification, this thesis also designed, built, and demonstrated a multi-letter word-formation pipeline combined with automatic English spelling correction, integrated into an interactive multi-model Gradio web application.

5.2 Key Findings and Contributions

- A new Hybrid MobileNetV2-Swin architecture — fusing a 1,280-dimensional convolutional feature vector with a 768-dimensional self-attention feature vector — achieves the highest accuracy of the six models compared (99.80%), showing that combining convolutional and transformer-based feature extraction is an effective strategy for ASL fingerspelling recognition, consistent with, and building on, recent hybrid-architecture findings elsewhere in the sign-language literature.
- The three transformer-based models (Hybrid, Swin, Vision Transformer) consistently and substantially beat the three pretrained CNN baselines, reaching 98.8%–99.8% accuracy against 89.9%–92.5% for the CNNs — a gap traceable to self-attention's ability to relate different regions of the hand adaptively, rather than applying fixed local filters uniformly, and visible in a much narrower, more consistent per-class error profile.
- Per-class analysis showed that the digit '0' and the letter 'o' — near-identical closed-circle handshapes in ASL — are the single most consistent source of error across all six models, and essentially the only source of error in the three transformer-based ones; the three CNN models additionally show broader confusion among the letters m/n/o, u/v/w, and the digit pair 4/5, meaning their errors are not confined to this one visually ambiguous pair.

- A corrected accuracy–speed trade-off revises the conventional wisdom for this task: the standalone Swin Transformer is not just highly accurate (99.60%) but in fact the fastest of all six models (1.608 s total inference time; 3.191 ms per image), while the considerably larger Vision Transformer, despite ranking only third in accuracy, is the slowest overall (3.852 s; 7.642 ms per image). This shows transformer-based models are not inherently slower than CNNs, and that architectural design — specifically the Swin Transformer's hierarchical, window-restricted self-attention — matters more for practical inference speed than raw parameter count.
- A working multi-letter word-formation and automatic English spelling-correction pipeline was designed, built, and integrated into an interactive multi-model web application, showing that the recognition accuracy of the top-performing models is enough to support genuinely usable word-level communication — directly closing the gap, identified in the reviewed literature, between isolated-letter classification and a practical communication aid. What was initially envisioned as future work has, within the scope of this thesis, been fully realized and demonstrated.
- These results, and their interpretation, align with and extend the recent literature on deep learning for sign-language recognition, offering a unified, reproducible six-way benchmark that directly compares three CNN paradigms, a standard Vision Transformer, a hierarchical Swin Transformer, and a new hybrid fusion architecture under identical experimental conditions — a comparison this broad remains uncommon in existing studies.

5.3 Recommendations for Practical Application

Building on the findings above, the models studied here can serve several practical applications directly benefiting the Deaf and Mute community. In education, the Swin Transformer — pairing high accuracy with the fastest inference time of the six — could power an interactive application for teaching the ASL alphabet, giving immediate feedback on whether a sign was performed correctly. In public services, the proposed Hybrid model, as the single most accurate architecture, could serve as the recognition layer in self-service kiosks at government or healthcare facilities, where maximum accuracy matters more than speed, enabling communication without a human interpreter having to be physically present. In mobile or embedded settings, MobileNetV2 remains the more practical choice given its considerably smaller footprint (13.03 MB) and low computational cost, even at a six-to-eight-point accuracy cost relative to the transformer-based models.

One particularly promising direction, now considerably de-risked by the word-formation and spelling-correction pipeline demonstrated in Section 4.8, is integrating the proposed system into video-conferencing platforms such as Zoom, Microsoft Teams, and Google Meet, given the

growing need for the Deaf and Mute community to take part fully in remote professional and educational meetings. In practice, this could take the shape of a plugin that continuously captures a Deaf participant's video feed, recognizes fingerspelled letters in real time using the Swin Transformer (for its favourable balance of accuracy and speed), assembles them into readable text with the language-processing layer already built in this thesis, and displays it as live captioning for other participants — while also converting other participants' speech into text for the Deaf participant, achieving two-way communication within a single meeting. Bringing this to full scale would still require tackling real-time video latency, continuous rather than isolated gesture recognition, and robust hand localization against a changing background, but the accuracy of the top three models in this study, together with the working word-formation and spelling-correction demonstration, provides a mature, substantially de-risked foundation for that future work.

5.4 Future Work

- Expand the dataset with a larger, more diverse set of images per class — varying lighting, background, and skin tone — to improve how well all six models generalize to real-world conditions, and to establish whether the CNN models' comparatively lower accuracy here is mainly a function of dataset size rather than architecture.
- Explore model compression and quantization, including knowledge distillation from the Hybrid or Swin Transformer models into a smaller student model, to further cut inference time and model size while keeping as much of the top models' accuracy edge as possible — particularly for the Vision Transformer and Hybrid models, whose larger sizes (327.46 MB and 117.86 MB respectively) could rule out some mobile deployment scenarios.
- Extend the system from static fingerspelling recognition toward dynamic gesture and continuous-sentence recognition, using architectures that capture spatial and temporal information jointly, building on the static-image foundation and the word-formation pipeline this thesis establishes.
- Systematically test the spelling-correction stage against a benchmark of deliberately noisy letter sequences, simulating realistic per-letter misclassification rates for each of the six models, and explore ASL-fingerspelling-aware language models as a more targeted alternative or complement to the general-purpose TextBlob model used here.
- Bring in Explainable AI (XAI) techniques, such as attention-map visualization, to build transparency and user trust in the model's predictions — particularly for the self-attention mechanisms in the Vision Transformer, Swin Transformer, and Hybrid models — and to dig further into the specific visual features behind the residual '0'/'o' confusion identified in Section 4.5.

- Build and evaluate the video-conferencing integration proposed in Section 5.3 as a working prototype, including a full real-time performance evaluation on representative hardware, building directly on the word-formation and spelling-correction pipeline already implemented and demonstrated in this thesis.

REFERENCES

- [1] Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- [2] Szeliski, R. (2022). *Computer Vision: Algorithms and Applications* (2nd ed.). Springer.
- [3] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- [4] Li, Y., Zhang, K., & Li, K. (2021). Deep learning for image classification: A survey. *Neurocomputing*, 449, 20–36.
- [5] 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.
- [6] 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.
- [7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778). <https://doi.org/10.1109/CVPR.2016.90>
- [8] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4510–4520). <https://doi.org/10.1109/CVPR.2018.00474>
- [9] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning* (pp. 6105–6114). PMLR. <https://arxiv.org/abs/1905.11946>
- [10] 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. In *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/2010.11929>
- [11] 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>
- [12] 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>

- [13] 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>
- [14] 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).
- [15] 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>
- [16] 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.
- [17] 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), V713-1560.
- [18] 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.
- [19] Kamruzzaman, M. M. (2020). Arabic Sign Language recognition and generating Arabic speech using convolutional neural network. *Wireless Communications and Mobile Computing*, 2020, 3685614.
- [20] 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.
- [21] 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).
- [22] Sharma, S. (2020). Vision-based sign language recognition system: A comprehensive review. In *Proceedings of the Fifth International Conference on Inventive Computation Technologies (ICICT)*.
- [23] 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.

- [24] Kouvakis, V., Trevlakis, 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.
- [25] 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.
- [26] 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).
- [27] 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.
- [28] Drishya (2023). Sign language recognition. *International Research Journal of Modernization in Engineering Technology and Science*, 5(4).
- [29] 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>
- [30] 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>
- [31] World Health Organization. (2021). *World report on hearing*. WHO. <https://www.who.int/publications/i/item/9789240020481>
- [32] Applying Swin architecture to diverse sign language datasets. (2024). *Electronics*, 13(8), 1509. <https://doi.org/10.3390/electronics13081509>