

Deep Learning Based Hand Gesture Recognition for Assistive Communication

Arlangki Sungoh¹, Abdul Samad K A², Ahmed Nazim Nisham³, Abdul Rahman Arshad⁴, Asst. Prof. Shruthi⁵

^{1,2,3,4}Student, Department of Computer Science and Engineering, Shree Devi Institute of Technology, Mangaluru, Karnataka, India

⁵Guide, Department of Computer Science and Engineering, Shree Devi Institute of Technology, Mangaluru, Karnataka, India

Abstract—People who are deaf or have a speech impairment rely on sign language, yet most members of the public cannot read it, and the gap becomes serious in emergencies where a message must be understood at once. This paper presents ASL2VOICE, a real-time system that watches a person's hand signs through an ordinary camera and turns them into on-screen text and spoken words. The system uses an SSD MobileNet V2 detector trained with the TensorFlow Object Detection API on ten emergency signs (Ambulance, Call, Danger, Emergency, Fire, Help, Medicine, Police, Stop and Water) plus a No Sign class for frames that hold no gesture. A FastAPI server receives frames from a Next.js web interface, runs detection and returns the sign together with a confidence score. Predictions below an adjustable threshold are rejected; accepted ones are displayed and read aloud with text-to-speech. The interface also lets the user choose a camera, switch voice output on or off and review recently detected signs. The paper describes the design, dataset, training procedure and evaluation plan of the system.

Index Terms—Assistive Communication, Deep Learning, Emergency Signs, Hand Gesture Recognition, Object Detection, SSD MobileNet V2, Text-to-Speech.

I. INTRODUCTION

Sign language is the main way many deaf and speech-impaired people communicate, but very few people outside that community can understand it. The difficulty is greatest in urgent situations such as an accident or a medical problem, where a person may need to ask for help, call an ambulance or warn others of danger and cannot wait for an interpreter. Computer vision offers a practical route to closing this gap. A camera can watch a person's hands, a trained

model can recognise which sign is being made, and the result can be shown as text or spoken aloud. Earlier work has explored this idea with convolutional networks and hand-landmark methods [1]– [5], and lightweight detectors such as SSD [6] with a MobileNetV2 backbone [7] are designed to run fast enough for live use.

This project builds ASL2VOICE, a real-time web application that recognises a fixed vocabulary of emergency-related signs and converts each recognised sign into text and speech. The user needs only a camera and a browser. The remainder of the paper covers related work, the problem, the methodology, the dataset, the model, the evaluation plan and the system architecture.

II. RELATED WORK

Several recent studies apply deep learning to sign recognition. Naidu et al. [1] combine RGB images with hand landmarks and report accuracy above 94%, at the price of a more complex architecture and heavier computation. Aruna et al. [2] add a documentation module to a real-time sign-to-text system, which improves usability, although accuracy still depends on the training data and the dataset is hard to scale.

Convolutional networks are the most common choice. Jaidhan et al. [3] show that a simple CNN pipeline can exceed 90% accuracy on static gestures but copes poorly with moving ones. Manoj Kumar et al. [4] report over 95% accuracy for a gesture-to-text CNN and note its sensitivity to lighting and to variation between signers. Diviya and Sanjana [5] find that careful image preprocessing improves results under

real conditions, although the approach needs large datasets and considerable computation.

Object detectors offer a different route, because they find the hand and classify the sign in one pass. The Single Shot MultiBox Detector [6] does this with a single network, and MobileNetV2 [7] supplies an efficient backbone with modest computing needs. The TensorFlow Object Detection API [8] provides tools for training and exporting such models.

Landmark tracking such as MediaPipe Hands [9] is a further alternative that reduces the hand to key points before classification.

Most of these systems stop at recognising a gesture. The present work joins recognition to a complete usable pipeline: a web interface, confidence-based filtering of doubtful predictions and spoken output, applied to signs chosen for emergency use.

III. PROBLEM STATEMENT

People with hearing and speech disabilities depend mainly on sign language. Existing gesture recognition solutions are often limited, slow or not accurate enough for practical use, and many assistive tools are hard to use, not portable or need special training.

Recognition itself is difficult because the appearance of a hand changes with its shape, orientation, the lighting and the background.

A wrongly recognised sign also costs more in an emergency than in casual use, so a practical system should be able to refuse a prediction it is unsure about instead of speaking a wrong word.

To address these problems, the developed system combines real-time detection, a confidence check, text display and speech output in one browser-based application.

IV. PROPOSED METHODOLOGY

Data collection.

Each sign is recorded as short video clips, grouped by sign and by recording session. Frames are extracted from the clips and each frame is labelled with a box around the hand and the name of the sign. A separate No Sign class holds frames where the hand is at rest or absent.

Preprocessing.

Frames are resized to the input size of the detector and normalised. The hand region is isolated by the detector's bounding box, so no separate background-removal step is required.

The work is organised into the stages shown in Fig. 1.

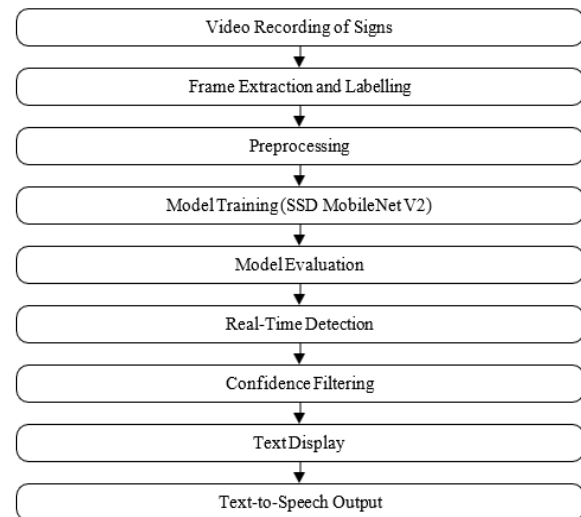


Fig. 1. System Workflow

Feature learning.

Hand shape, finger position and orientation are learned by the network from the labelled frames instead of being extracted with hand-written rules.

Training and evaluation.

The labelled data are divided into a training set and a test set, and the model is trained and scored as described in Sections VI and VII.

Deployment.

The trained model is served by the back end. Live frames are detected, filtered by confidence, displayed as text and spoken (Sections VIII and IX).

V. DATASET AND SIGN VOCABULARY

The model vocabulary has ten emergency signs and one background class, listed in Table I. The clips were recorded for this project and are stored in one folder per class, with a sub-folder for each recording session.

The dataset is small and is meant to grow as more sessions are recorded.

TABLE I SIGN CLASSES USED IN THE MODEL

No.	Class	No.	Class
1	AMBULANCE	7	MEDICINE
2	CALL	8	POLICE
3	DANGER	9	STOP
4	EMERGENCY	10	WATER
5	FIRE	11	NO SIGN
6	HELP		

Recording each sign in several sessions, with changes in hand position, distance from the camera and lighting, exposes the model to the variation described in Section III.

VI. DETECTION MODEL AND TRAINING

The recogniser is an SSD MobileNet V2 detector. SSD predicts box locations and class scores for a fixed set of default boxes in a single forward pass [6], and the MobileNetV2 backbone uses inverted residual blocks to keep the computation low [7]. The model is trained with the TensorFlow Object Detection API [8] on the labelled frames.

For every frame the model outputs a set of boxes, each with a sign label and a confidence score between 0 and 1. The training and test sets are separated by recording session rather than by individual frame. Frames from one clip look almost identical, so a frame-level split would leak training images into the test set and make the scores look better than they are.

VII. MODEL EVALUATION

The detector is scored on the held-out test set. Precision is the share of detections that are correct, and recall is the share of real signs that are detected. F1-score combines the two through their harmonic mean. Mean average precision at an intersection-over-union of 0.5 (mAP@0.5) measures detection quality over all classes together, and the inference time per frame shows whether the system is fast enough for live use.

These values are computed for each sign as well as over all signs, since one poorly recognised sign can be hidden by a good average. They also guide the choice of confidence threshold. A high threshold rejects more wrong predictions but also more correct ones, and a low threshold does the opposite. For emergency signs

a wrongly spoken word is costlier than a repeated attempt, so the threshold is set to favour precision. Because the threshold is adjustable in the interface, it can be tuned to the lighting and camera in use.

VIII. CONFIDENCE FILTERING AND SPEECH OUTPUT

Every prediction passes a confidence check before it is used, as shown in Fig. 2. A sign whose score reaches the threshold is accepted; otherwise, the prediction is rejected and nothing is shown or spoken. The No Sign class lets the model decide that a frame holds no gesture, and such frames are not spoken.

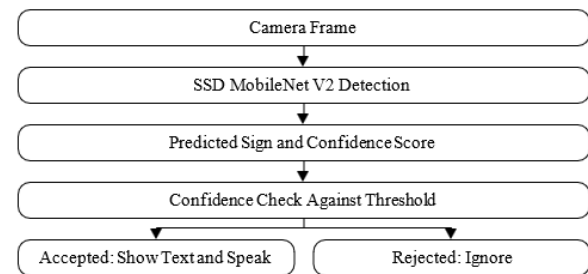


Fig. 2. Confidence Filtering Flow

Accepted signs are displayed as text with their confidence, added to the list of recent signs and, when voice output is switched on, spoken through the Windows speech synthesis interface. The user can start and stop detection, select the camera and toggle voice output.

IX. IMPLEMENTATION ARCHITECTURE

ASL2VOICE is a web application with a Next.js and React front end written in TypeScript and a Python back end built with FastAPI.

The browser captures frames from the selected camera and sends each one to the /predict endpoint. The back end decodes the image with OpenCV and NumPy, runs the TensorFlow SSD MobileNet V2 model and returns the sign with its confidence score, after which the confidence check is applied and the result is shown.

Keeping the model in the back end leaves the browser responsible only for the camera, the display and the speech. Fig. 3 traces one prediction. The interface supports dark and light themes and adapts to different screen sizes.

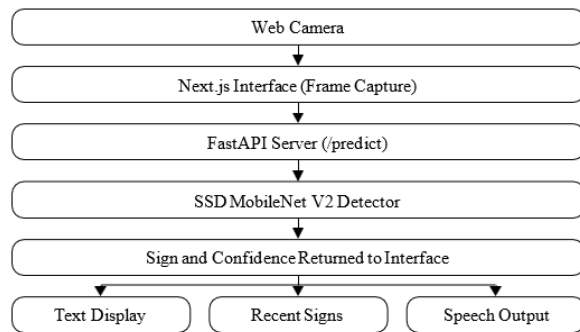


Fig. 3. Prediction Flow Through the System

X. RESULTS AND DISCUSSION

The two parts of the system behave differently and are judged differently. The detector is judged by the measures in Section VII, computed on sessions that were kept out of training. The interface is judged by whether the whole chain works from camera capture to spoken output.

Signs whose hand shapes look alike are the most likely to be confused, and per-sign precision and recall are needed to show which pairs these are. Changes in lighting and background are the other main source of error, which is why each sign is recorded in more than one session. The confidence threshold links the two parts: raising it makes spoken output more trustworthy but asks the user to repeat a sign more often. The evaluation in this study concentrates on the design of the pipeline and the way it is scored. A detailed numerical comparison across many signers, cameras and lighting conditions needs a larger dataset and is identified as future work in Section XI.

XI. LIMITATIONS AND FUTURE SCOPE

This is an academic project with a small fixed vocabulary recorded by a limited number of signers, so results may not carry over to other people, cameras or lighting. The system recognises signs one frame at a time and does not yet handle moving gestures or full sentences. Speech output uses the Windows speech interface, so spoken output currently depends on a Windows machine.

Future work can add more signs, moving gestures, sentence building from consecutive signs, other speech engines and a mobile version, and can test the system with more signers.

XII. CONCLUSION

We have presented ASL2VOICE, a real-time system that detects emergency sign gestures with an SSD MobileNet V2 model, rejects doubtful predictions with an adjustable confidence threshold and presents accepted signs as text and speech in a web interface. By joining detection, filtering and voice output in one application, the system aims to make urgent communication easier between sign-language users and people who do not know sign language. Completing the evaluation on a larger and more varied set of signers is the main step that remains.

ACKNOWLEDGMENT

We thank our guide, Asst. Prof. Shruthi, for her guidance and support during this work.

REFERENCES

- [1] G. Naidu, R. Alhate, A. Bhatjoshi, and A. Chaudhari, "Sign language recognition using CV & ML," 2026.
- [2] R. Aruna, B. L. Reddy, and M. S. Manikanta, "Sign language recognition & documentation using DL," 2026.
- [3] B. J. Jaidhan, P. V. Babu, S. Ketha, and D. V. Kumar, "Deep learning for inclusive communication (CNN-based)," 2024.
- [4] V. S. Manoj Kumar, V. Golla, M. V. Prasad, and R. Reenadevi, "Gesture-to-text CNN model for sign translation," 2024.
- [5] M. Diviya and N. Sanjana, "CNN-based sign language translation system," 2025.
- [6] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Proc. European Conf. Computer Vision (ECCV)*, 2016, pp. 21–37.
- [7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.
- [8] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [9] F. Zhang *et al.*, "MediaPipe Hands: On-device real-time hand tracking," arXiv:2006.10214, 2020.