Innovative: International Multi-disciplinary Journal of Applied Technology
(ISSN 2995-486X) VOLUME 04 ISSUE 03 2026

A WEBCAM-BASED RECOGNITION SYSTEM FOR THE UZBEK FINGERSPELLING ALPHABET

Sevinch Jovlieva
Academy of Islamic Studies of Uzbekistan, Third-Year Student in Information Security Management

Abstract:
For individuals with hearing or speech impairments, sign language serves as the primary channel of natural connection with society. To date, work on the automated recognition of Uzbek sign language remains scarce, and an openly available dataset for this purpose is essentially nonexistent. This paper presents a system that recognizes the Uzbek fingerspelling alphabet using an ordinary webcam. Hand movements are tracked with the MediaPipe Hands tool; letters represented by static hand poses are classified using a Random Forest classifier, while letters expressed through motion are recognized using an LSTM neural network. Trained on a manually collected dataset of 1,523 static frames and 468 motion sequences, the system recognized static letters with 99.11% accuracy and dynamic letters with 65.96% accuracy. The results indicate that the recognition of stable hand poses can be achieved with high reliability, whereas the recognition of temporally evolving gestures remains a considerably more complex scientific and technical challenge.

Keywords: Uzbek Sign Language, Fingerspelling Alphabet, Mediapipe, Random Forest, LSTM, Webcam

Introduction

For individuals with speech or hearing impairments, sign language is key to full participation in society, access to education, and independence in daily life. Automated sign-language recognition systems are being actively developed worldwide: for a number of major languages, sufficiently large datasets and well-tested algorithms already exist [1]. For the Uzbek language, however, this line of research remains at an early stage. The reasons for this are clear: the absence of a standardized, openly available dataset, the lack of a shared methodological approach among researchers, and the absence of a single, widely accepted reference for how certain letters should be represented by hand [2].

Automated recognition tools serve as a bridge that narrows the gap between people who know sign language and those who do not, rather than replacing live human interaction [3]. The aim of this study is to build a system that reliably recognizes the Uzbek fingerspelling alphabet using ordinary, widely available equipment such as a webcam, and to present its strengths and weaknesses within a transparent, reproducible protocol [4].

In the Uzbek fingerspelling alphabet, some letters are represented by holding the hand in a fixed pose, while others are expressed through a distinct hand motion. This distinction directly shaped the choice of architecture: cases where a single frame suffices and cases that require accounting for the temporal dimension call for fundamentally different solutions — accordingly, the study adopts a two-component, two-stage recognition approach. The scientific novelty of this approach lies in three aspects [5]. First,

an independent dataset was assembled from scratch for the Uzbek fingerspelling alphabet, without relying on any pre-existing source. Second, a hybrid architecture was developed that unites static and dynamic letter recognition within a single coherent pipeline. Third, the resulting performance was evaluated under a clearly defined protocol on data that took no part whatsoever in training. The following sections present, in sequence, the technical details, empirical results, and analytical conclusions of this approach [6].

Materials and Method

The dataset was collected manually by the author, with a single participant, in front of an ordinary webcam. A total of 1,523 static frames and 468 motion sequences were collected, each sequence consisting of 30 frames. The letters C and W are not used in the Uzbek Latin alphabet. In practice, the fingerspelling alphabet comprises 24 Latin letters plus four special characters — O', G', Sh, and Ch — for a total of 28 symbols. Static evaluation was carried out on 1,115 samples covering the 24 Latin letters; the 408 static frames corresponding to the four special letters were excluded from static testing owing to a difference in labeling format, but were fully included in the dynamic component. Consequently, static evaluation was performed across 24 classes based on 1,115 samples, while dynamic evaluation was performed across 28 classes.

Hand tracking was performed using the MediaPipe Hands tool, which detects 21 landmark points of the hand in each frame and converts them into a 63-dimensional numerical vector. These vectors were used directly for model training and recognition.

Rather than using raw coordinates directly, the wrist point of the hand was shifted to the origin, after which hand size was scaled relative to the distance to the farthest point. Wrist-centering and scaling substantially improved holdout accuracy in a supplementary check: a model tested without normalization was limited to roughly 7% accuracy, whereas after normalization this figure approached 98%. This particular figure pertains to a separate, supplementary check conducted solely to illustrate the overall effect of normalization; the primary results reported in Sections 3 and 4 — 99.11% for the static model and 65.96% for the dynamic model — are the final figures obtained under the independent, full evaluation protocol.

For letters represented by static poses, a Random Forest classifier with 100 trees was chosen. The rationale for this choice is straightforward: the 63-dimensional coordinate vector from a single frame constitutes a static, time-independent feature space, and tree-based ensemble algorithms perform effectively on exactly this kind of bounded-dimensional yet nonlinearly separable spatial problem, while also keeping computational cost low.

For letters expressed through motion, an LSTM neural network consisting of three sequential layers — of 64, 128, and 64 neurons — was employed. Here the task is fundamentally different: rather than a single frame, the model must process an entire 30-frame trajectory, which is why a recurrent, memory-based architecture was chosen — a time-independent model such as Random Forest would be unable to capture the internal dynamics of a trajectory.

For practical use, the two models were combined into a single demo application: if a hand is tracked continuously for 30 frames and the LSTM network expresses sufficient confidence in its prediction, the result is accepted as a dynamic letter; otherwise, the system returns a prediction from the Random Forest classifier based on a single frame. This combination was designed purely to streamline the user experience; the accuracy figures reported below correspond to the independent, separately conducted evaluation of each model — the hybrid combination itself was not separately measured.

To obtain an unbiased assessment of the models' true performance, the dataset was split into two parts: 90% for training and the remaining 10% for testing, with the test data taking no part whatsoever in training. For the static model, a five-fold cross-validation was additionally performed, in which the dataset was split five times in different ways and accuracy was recomputed on each split. For the dynamic model, a Wilson confidence interval was calculated based on 47 test samples — a statistical method indicating the range within which the true accuracy is likely to lie for a small test set.

Results

The static model achieved a mean accuracy of 97.85% ± 1.28% under five-fold cross-validation. This figure confirms that performance was nearly consistent across different splits of the dataset, indicating that the Random Forest classifier's performance did not depend on a particular random split. On the final held-out test set, the result was even higher: 111 out of 112 samples were correctly classified, corresponding to an accuracy of 99.11%. The difference between these two figures — 97.85% under cross-validation and 99.11% on the final test set — is a statistically expected occurrence, since the final test set is relatively small, meaning that a single error has a proportionally larger effect on the overall percentage [7].

The single error consisted of confusing the letter B with the letter Z. The hand configurations for these two letters are geometrically close to one another [8]: the difference in finger placement and hand-bend angle between them is minimal, and this is only weakly reflected in the 63-dimensional vector produced by MediaPipe.

The dynamic model, the LSTM network, correctly identified 31 of 47 test samples — an accuracy of 65.96%. The Wilson confidence interval spans a relatively wide range, from 51.7% to 77.9%, which is directly attributable to the small size of the test set — only 47 samples: in such a small sample, a single additional correct or incorrect prediction shifts the final percentage considerably [9]. Nevertheless, the 65.96% figure is more than tenfold higher than the 3.57% expected under random guessing across 28 classes, which statistically confirms that the model learned genuine distinctions among the trajectories [10].

Table 1. Test results of the static and dynamic models.

Model	Number of classes	Evaluation method	Accuracy
Random Forest (static)	24	5-fold cross-validation	97.85% ± 1.28%
Random Forest (static)	24	Test set, 112 samples	99.11%
LSTM (dynamic)	28	Test set, 47 samples	65.96%
Random guessing	28	—	3.57%

The class-wise distribution is uneven: only 14 samples were collected for the letters B and J, while 22 samples were collected for the letter K — meaning the gap between the least- and most-represented classes is nearly twofold. In the static set, the 24 classes correspond to an average of 46 samples each, whereas in the dynamic set the 28 classes correspond to an average of only 17 samples each — this discrepancy forms the numerical basis for the accuracy gap analyzed in the following section [11].

Discussion

The 33-percentage-point gap between the Random Forest classifier's 99.11% result and the LSTM network's 65.96% result arises from the overlap of two factors: the computational complexity of the task and the volume of available training data.

Static letter recognition is a spatial classification problem [12]. The hand pose within a single frame does not change over time, so the 63-dimensional feature vector forms a stable, recurring pattern. A tree-based algorithm such as Random Forest can separate such spatial boundaries using a relatively small number of rules, and the 1,115 training samples — an average of 46 per class — provide a volume sufficient for this algorithm to form stable decision boundaries [13].

Dynamic letter recognition, by contrast, adds the temporal dimension: the model must understand an entire 30-frame sequence as a whole, learning the relationship among the onset, transition phase, and conclusion of a motion. This is a demanding task even for an LSTM architecture with a considerably larger number of parameters, since learning sequential dependencies generally requires more training samples than spatial classification does [14]. Here, the average of 17 samples per class is only about

one-third of the 46 available in the static set. As a result, the model manages to learn the general direction of a trajectory but cannot fully distinguish finer differences among motions — the 65.96% figure is the numerical expression of precisely this limitation [15].

As a supplementary test, a model trained with data augmentation achieved 97.87% on the test set. This figure was not included in the main results, since it was obtained under a broader and differently configured training protocol and cannot be directly compared with the primary findings. Nonetheless, this finding provides an important signal: it suggests that the low dynamic accuracy may be primarily attributable to the limited size of the training sample, and the 97.87% figure obtained through augmentation indirectly supports this interpretation.

All data were collected from a single participant. This leaves open the question of how well the system would generalize to other individuals' hand shapes, movement speed, and gestural style.

Reviewing prior work on Uzbek sign language, the MediaPipe-based dataset-construction approach presented by Jurayev in 2025, as well as studies published at the ICTACS conference in 2024 and in the European International Journal of Multidisciplinary Research and Management Studies, have largely focused on broader data collection or general classification tasks; studies that evaluate static and dynamic letters separately under a clearly defined protocol are essentially absent. In this sense, the present study offers a clearer and more reproducible two-component evaluation benchmark for the Uzbek fingerspelling alphabet.

Conclusion

This project demonstrates that the Uzbek fingerspelling alphabet can be recognized at a practical level using an ordinary webcam and open-source libraries, but that the two components of the task — detecting static poses and understanding dynamic sequences — differ fundamentally in their level of complexity. The main practical takeaway for developers and researchers is that, when designing sign-language recognition systems, adopting a two-component architecture tailored to the nature of the task, rather than a single universal model, proves worthwhile in practice: separating the spatial and temporal-sequence problems rather than loading them onto a single model turns out to be key to allocating resources appropriately.

For institutions in education and healthcare, this work indicates that building an open dataset for Uzbek sign language should now become an institutional priority rather than merely a technical one — it is precisely this shortage of resources that is currently setting the pace for further research.

Three directions are identified as priorities for future work: unifying the labeling format so that the static portion of the four special letters can be incorporated into static evaluation; collecting data from multiple participants to make the system robust to varying hand shapes and movement styles; and moving beyond individual letters toward recognition at the level of whole words and phrases. These directions will serve to progressively close the existing gap in computer-based recognition of Uzbek sign language.

References

[1] World Health Organization, “Deafness and hearing loss: Fact sheet,” WHO, 2021.

[2] President of the Republic of Uzbekistan, “Resolution No. PQ-2909 of April 21, 2017: On the rights of persons with disabilities,” 2017.

[3] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, S. Wei, and C. L. Chang, “MediaPipe Hands: On-device real-time hand tracking,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2020.

[4] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, “OpenPose: Realtime multi-person 2D pose estimation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, 2021.

[5] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[6] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] O. Koller, S. Zargaran, and H. Ney, “Deep Sign: Hybrid CNN-HMM for continuous sign language recognition,” in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2016.

[8] N. C. Camgoz *et al.*, “Neural sign language translation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018.

[9] D. B. Jurayev, “Technology for constructing a dataset for translating into Uzbek sign language based on MediaPipe,” *Int. J.: Theory Pract.*, vol. 8, no. 3, 2025.

[10] “A machine learning algorithm to assist people with speaking disabilities for sign language detection of people speaking Uzbek language,” in *Proc. ICTACS*, 2024.

[11] O. A. Kayumov, N. R. Kayumova, and D. F. Xodjabekova, “Uzbek sign language classifier based on machine learning,” *Eur. Int. J. Multidisciplinary Res. Manage. Stud.*, vol. 4, no. 5, pp. 269–280, 2024, doi: 10.55640/eijmrms-04-05-43.

[12] O. Koller, N. C. Camgoz, H. Ney, and R. Bowden, “Weakly supervised learning of sign language,” *Found. Trends Comput. Graph. Vis.*, vol. 13, no. 3, pp. 1–118, 2020.

[13] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, “Sign language transformers: Joint end-to-end sign language recognition and translation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 10023–10033.

[14] O. Koller, O. Zargaran, H. Ney, and R. Bowden, “Deep sign: Hybrid CNN-HMM for continuous sign language recognition,” in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2016.

[15] A. M. Sincan and H. Y. Keles, “AUTSL: A large scale multi-modal Turkish sign language dataset and baseline methods,” *IEEE Access*, vol. 8, pp. 181340–181355, 2020.