Journal of Engineering Science *Vol. XXXIII, no. 2 (2026), pp. 7 - 19*
Fascicle *Industrial Engineering* *ISSN 2587-3474*
Topic *Applied Engineering Sciences and Management* *eISSN 2587-3482*

https://doi.org/10.52326/jes.utm.2026.33(2).01
UDC 004.8:004.934:629.735.33

VOICE CONTROL SYSTEM FOR PILOTING DRONES USING ARTIFICIAL INTELLIGENCE TECHNIQUES

Gheorghe Bulai[1]*, ORCID: 0009-0002-1664-7856,
Viorel Bostan[1], ORCID: 0000-0002-2422-3538,
Lilia Rotaru[1], ORCID: 0000-0002-6578-418X,
Tudor Bumbu[2], ORCID: 0000-0001-5311-4464

[1]*Technical University of Moldova, 168 Stefan cel Mare blvd, MD2004,Chisinau, Republic of Moldova*
[2]*Moldova State University, 60 Alexei Mateevici Street, Chișinău, MD-2009, Republic of Moldova*
* Corresponding author: Gheorghe Bulai, *gheorghe.bulai@iis.utm.md*

Received: 05.21.2026
Accepted: 06.17.2026

Abstract. Voice control of drones is an innovation direction with high potential for many professional and social domains. This article analyzes the practical usefulness of a voice control system for drones based on an artificial intelligence model that runs locally, without an internet connection. The hybrid CNN+BiLSTM model is trained to recognize commands in Romanian and is deployed on the ESP-Drone platform with an ESP32-S2 microcontroller. Unlike commercial alternatives based on cloud infrastructure, the proposed solution runs entirely locally, eliminating latency and dependence on connectivity. The application domains with the greatest impact are examined, such as industrial inspections, search and rescue operations and precision agriculture, while also highlighting an often overlooked category of beneficiaries: people with motor disabilities, for whom voice control offers the possibility to pilot drones independently, without using their hands. The experimental results demonstrate a test accuracy of 99.4% and an end-to-end latency of 210–350 ms, confirmed under real flight conditions.

Keywords: *accessibility, voice control, human-machine interaction, BiLSTM, CNN, MFCC, neural networks, embedded systems, TinyML*.

Rezumat. Controlul vocal al dronelor este o direcție de inovare cu potențial ridicat pentru multe domenii profesionale și sociale. Acest articol analizează utilitatea practică a unui sistem de control vocal pentru drone bazat pe un model de inteligență artificială care rulează local, fără conexiune la internet. Modelul hibrid CNN+BiLSTM este antrenat să recunoască comenzi în limba română și este implementat pe platforma ESP-Drone cu un microcontroler ESP32-S2. Spre deosebire de alternativele comerciale bazate pe infrastructură cloud, soluția propusă rulează în întregime local, eliminând latența și dependența de conectivitate. Sunt examinate domeniile de aplicație cu cel mai mare impact, cum ar fi inspecțiile industriale, operațiunile de căutare și salvare și agricultura de precizie, evidențiind totodată o categorie de beneficiari adesea trecută cu vederea: persoanele cu dizabilități motorii, pentru care controlul vocal oferă posibilitatea de a pilota drone independent, fără a-și folosi mâinile.

Rezultatele experimentale demonstrează o precizie de testare de 99,4% și o latență end-to-end de 210–350 ms, confirmată în condiții reale de zbor.

Cuvinte cheie: *accesibilitate, control vocal, interacțiune om-mașină, BiLSTM, CNN, MFCC, rețele neuronale, sisteme integrate, TinyML.*

1. Introduction

In recent years, drones have seen accelerated market growth [1], evolving from devices intended for entertainment and multimedia applications to accessible platforms used in agriculture, technical inspections, logistics and emergency operations [2]. This rapid expansion exposes a fundamental limitation of the current control paradigm: the operator must use both hands for the remote controller, focusing their entire attention on maneuvering the device. In an industrial inspection scenario, an engineer who must simultaneously operate measuring equipment and pilot a drone faces an ergonomic constraint that is hard to solve with traditional solutions.

Voice control is a natural answer to this problem. The voice is the most intuitive human-machine interaction interface: it requires no prolonged training, leaves the hands free and can work under reduced visibility or operational stress. Recent advances in artificial intelligence, in particular the TinyML paradigm [3] (running machine learning models directly on embedded devices, without an internet connection), have made it possible to implement real-time voice recognition on affordable hardware [4].

The traditional remote controller with analog joysticks is a robust tool, but it implies an essential constraint: the operator cannot carry out any other manual activity while piloting the drone. Compared with the commercial DJI or Parrot solutions, which process voice commands in the cloud and do not support Romanian, the proposed system has an end-to-end latency of 210–350 ms, versus 500–1200 ms for cloud architectures [5], and runs entirely locally, including in areas without internet coverage. The system supports nine voice commands in Romanian (take off, land, forward, back, left, right, rotate left, rotate right and stop), enough coverage to perform all the fundamental movements of a quadcopter.

The usefulness of voice control appears in scenarios where the operator must be physically present, carry out parallel activities or act quickly. In industrial and critical-infrastructure inspections, the engineer can dictate observations and record data from a thermometer or a gas detector without interrupting piloting, thus removing a critical point of inefficiency. In search and rescue operations, a voice-controlled drone can be piloted by a rescuer who is simultaneously holding binoculars, a GPS or a safety rope; the local system performs the same regardless of telecommunication network coverage (GSM or internet), since communication with the drone takes place over a local WiFi network. In precision agriculture, a farmer can speak to the drone while manually examining plants or collecting soil samples. Voice control is also useful in tactical and security operations, as well as in cinematography, where the director can send simple commands (forward, rotate right, stop) directly to the drone.

An often overlooked application domain is accessibility for people with motor disabilities of the upper limbs. The traditional remote controller is practically inaccessible to people with paralysis or amputation of both upper limbs, and voice control removes this barrier: a person who can speak is able to pilot a drone with a system such as the one described in this paper. Concrete applications for this category of users include property surveillance, delivery of light objects inside the home, photography from inaccessible angles

and participation in recreational activities on an equal footing with people without disabilities. Because the voice-recognition neural network is trained on the individual user's voice, the system also adapts to the pronunciation particularities of people with speech difficulties. Local operation removes the dependence on the internet connection, ensuring the system's availability under any conditions.

The main aim of the paper is to highlight the concrete domains where such a solution produces a real impact, including for people with motor disabilities, and to present the methodology for building a local voice control system for drones.

2. Materials and methods

This section presents the methodology for building the voice control system: the overall two-component architecture, the audio signal acquisition and processing chain together with the mathematical feature-extraction model, the architecture of the neural classification model and the communication protocol with the drone. For each stage, the components involved, the corresponding mathematical relations and their role in the processing chain are presented.

The architecture of the analyzed system is structured around two main components, the ground station (the operator's laptop) and the drone, which communicate over WiFi, without dependence on external infrastructure [6]. This conceptual separation is illustrated in Figure 1. This separation keeps the heavy computation, audio capture and neural inference, on the laptop, while the drone only receives ready-to-use flight commands. As a result, the drone needs no additional onboard hardware beyond its standard ESP32-S2 microcontroller, which keeps weight and cost low. The two components exchange data exclusively over a local WiFi link, so the system stays fully operational in areas without internet or cellular coverage. Because all model logic lives on the ground station, the recognition model can be retrained or replaced without any change to the drone firmware.

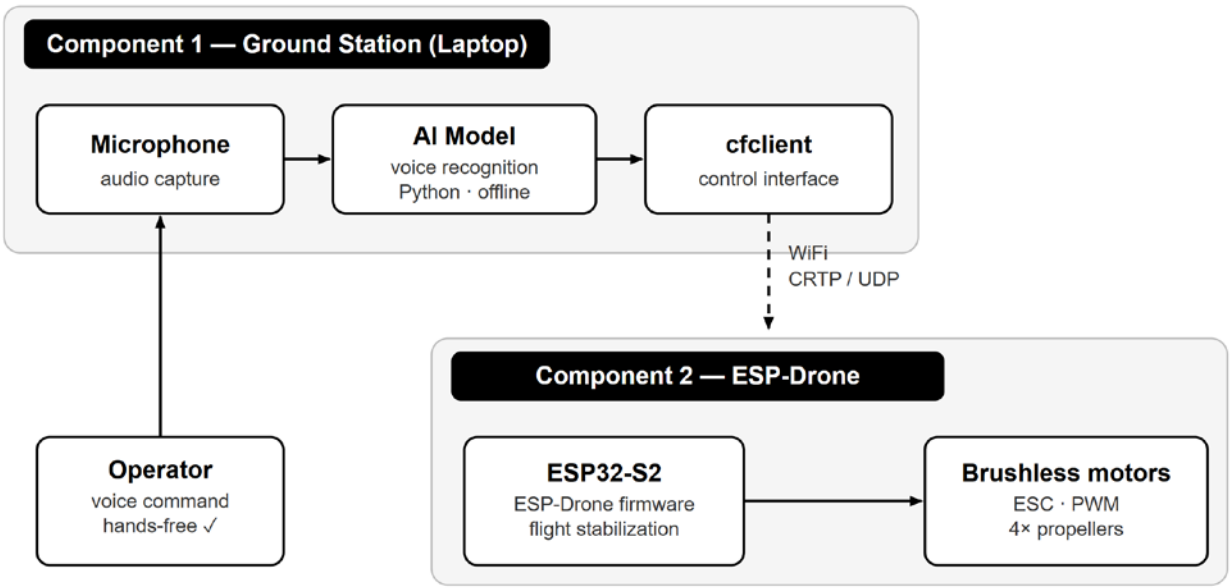

Figure 1. Conceptual architecture of the system: the ground station and the drone communicate over WiFi, without external infrastructure.

The ground station encompasses the entire voice-processing chain: capturing the audio signal through the microphone, preprocessing by extracting MFCC coefficients,

inference of the CNN+BiLSTM artificial intelligence model and transmitting the resulting command over WiFi to the drone. This approach makes use of the computing resources of an existing laptop and removes the need for additional dedicated hardware. The ESP-Drone is equipped with the ESP32-S2 microcontroller, which receives the commands and translates them into control signals for the brushless motors, through electronic speed controllers (ESC), running the open-source ESP-Drone firmware derived from the Crazyflie project [6].

2.1. Audio signal acquisition and processing

The voice signal is captured by the microphone at a sampling rate of 16 kHz, in 2-second analysis windows. The complete processing flow, from signal capture to command execution on the drone, is shown in Figure 2 as a sequence diagram. The microphone sends the signal to the AudioFeatureExtractor component, which extracts the MFCC coefficients; these are passed to the CommandRecognizer component, which runs the CNN+BiLSTM network inference and, if the confidence score exceeds the set threshold, sends the command to the DroneController; the latter translates it into flight parameters and transmits it to the drone through CRTP packets over WiFi.

Sampling at 16 kHz is a deliberate compromise: it preserves the frequency range that carries the information of spoken commands while keeping the amount of data per window low enough for real-time processing on a standard laptop. The fixed 2-second window sets the maximum length of a command and gives the network a consistent input size, which simplifies both feature extraction and inference. The confidence threshold acts as a rejection mechanism: when no command is spoken, or when background noise dominates, the score stays low and the DroneController receives nothing, so the drone holds its current state instead of reacting to spurious input. Because each stage passes a compact representation to the next, the microphone hands over raw audio, the AudioFeatureExtractor a small feature matrix and the CommandRecognizer a single label, the data volume shrinks along the chain, which keeps the end-to-end latency within the range required for responsive control.

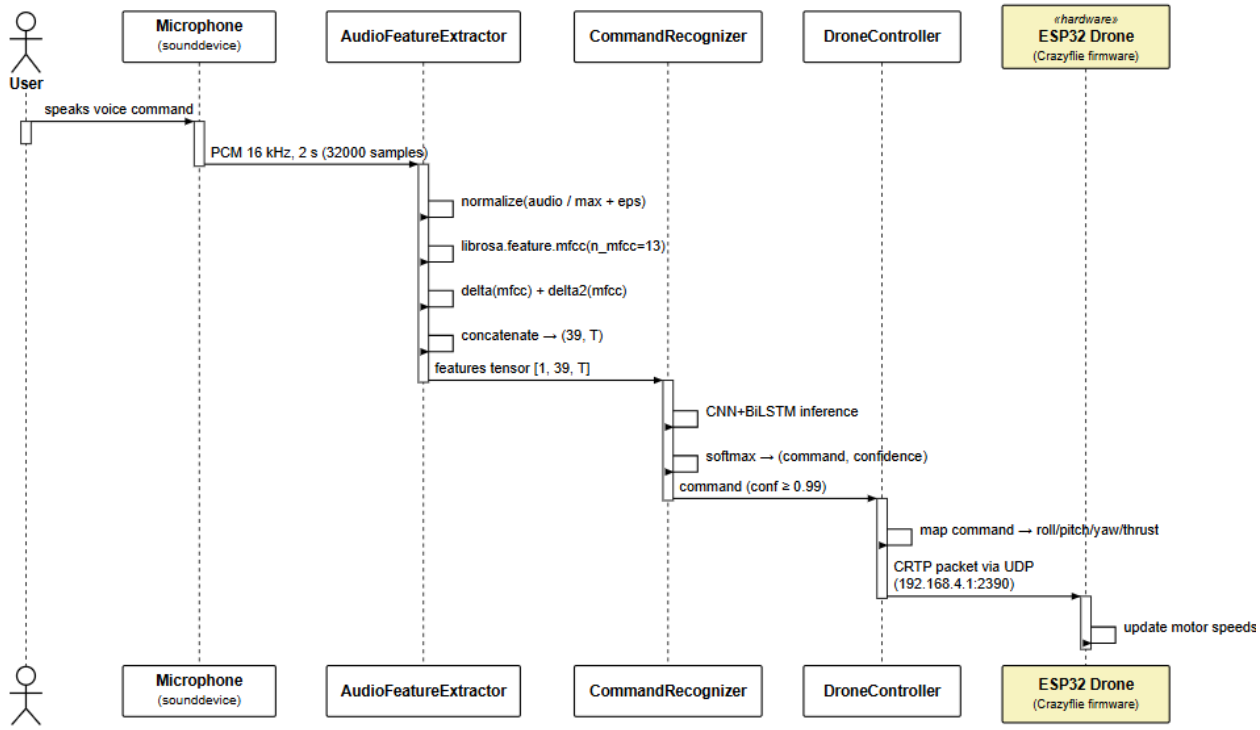

Figure 2. Sequence diagram of voice command processing.

The MFCC representation is preferred to the raw audio signal because it compresses the relevant spectral information of the human voice into a compact vector, significantly reducing the computational complexity of inference without loss of accuracy [7]. The coefficient extraction process, shown in Figure 3, goes through the following stages: pre-emphasis, framing, windowing, fast Fourier transform (FFT), Mel-scale filtering, log-energy computation and the discrete cosine transform (DCT).

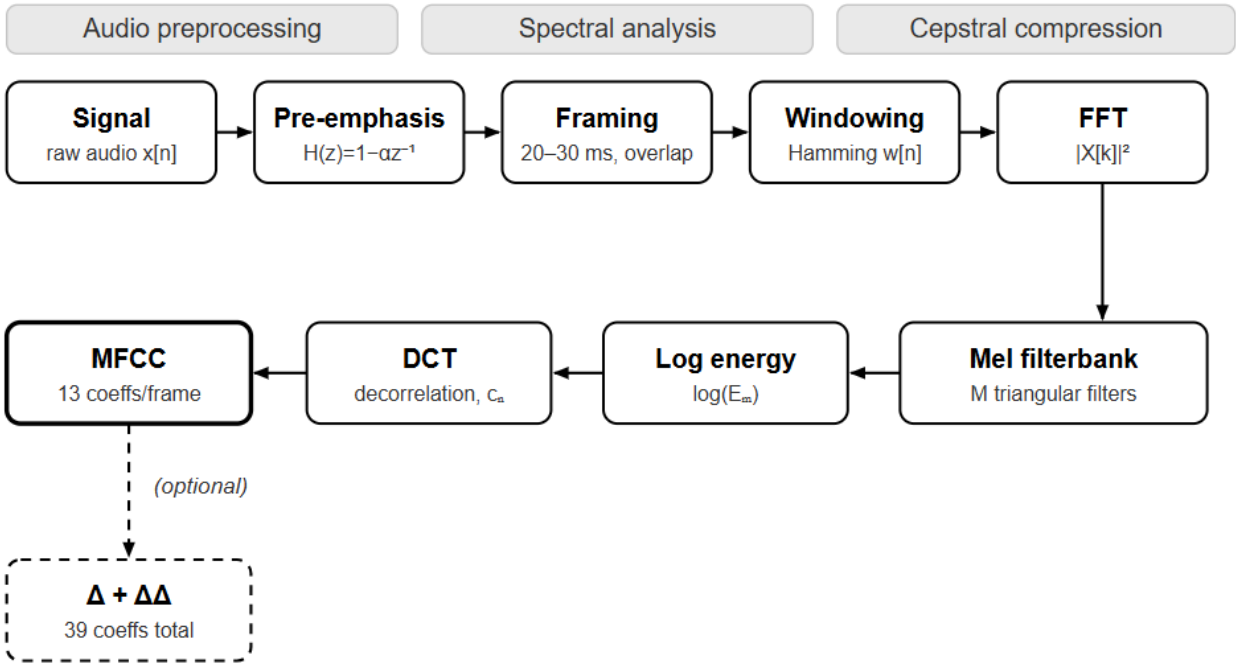

Figure 3. Mel-Frequency Cepstral Coefficients (MFCC) extraction pipeline.

The mathematical model of the extraction chain is described by relations (1)–(7), each stage playing a specific role in transforming the raw signal into a set of discriminative features. Pre-emphasis amplifies the high-frequency components of the signal $x[n]$, naturally attenuated in speech, balancing the spectrum and improving the signal-to-noise ratio for the upper formants; α ≈ 0.97 (Eq. (1)):

$$y[n] = x[n] - \alpha \cdot x[n-1] \tag{1}$$

The signal is then segmented into frames of 20–30 ms and weighted with a Hamming window $w[n]$, which reduces spectral leakage at the frame edges and ensures the continuity required for the Fourier transform; N is the number of samples in the frame (Eq. (2)):

$$w[n] = 0.54 - 0.46 \cdot cos\left(\frac{2\pi n}{N-1}\right) \tag{2}$$

For each frame, the power spectrum is computed through the discrete Fourier transform of the N samples, moving the signal from the time domain to the frequency domain, where voice information is more easily separable (Eq. (3)):

$$P[k] = \left| \sum_{n=0}^{N-1} x[n] \cdot w[n] \cdot e^{-j2\pi kn/N} \right|^2 \tag{3}$$

The linear frequencies f (in Hz) are converted to the perceptual Mel scale, which mimics the nonlinear frequency resolution of the human ear, finer at low frequencies, where the relevant voice information is concentrated (Eq. (4)):

$$m(f) = 2595 \cdot log_{10}\left(1 + \frac{f}{700}\right) \quad (4)$$

The energies resulting from the M triangular Mel filters are logarithmized, an operation that compresses the dynamic range, similar to human perception of sound intensity, and groups the energy into perceptually relevant bands; $H_m[k]$ is the response of filter m (Eq. (5)):

$$S_m = ln\left(\sum_{k=0}^{N-1} P[k] \cdot H_m[k]\right) \quad (5)$$

The discrete cosine transform decorrelates the log-energies and concentrates the information in the first coefficients, producing a compact feature vector; this yields the cepstral coefficients c_i, i = 0…12 (Eq. (6)):

$$c_i = \sum_{m=1}^{M} S_m \cdot cos\left(\frac{\pi \cdot i \cdot (m-0.5)}{M}\right) \quad (6)$$

To capture the temporal dynamics of speech (the speed and acceleration of spectral variation, essential for distinguishing commands), the delta coefficients Δc_t and, analogously, the accelerations $\Delta\Delta c_t$ are computed (Eq. (7)):

$$\Delta c_t = \frac{\sum_{n=1}^{K} n \cdot (c_{t+n} - c_{t-n})}{2 \cdot \sum_{n=1}^{K} n^2} \quad (7)$$

Each analysis window is thus transformed into a matrix of 39 coefficients (13 base MFCC, 13 delta and 13 delta-delta), which constitutes the input tensor of the neural network.

2.2. The CNN+BiLSTM neural model

Voice command recognition is treated as a multi-class classification problem: for a 2-second audio segment, the model assigns one of the nine labels corresponding to the commands, while recordings below the confidence threshold are rejected as background noise. The hybrid architecture was chosen because it combines the efficiency of convolutions in extracting local features from the spectral representation with the ability of LSTM networks to model long-term temporal dependencies [8]. The convolutional module consists of two successive Conv1d blocks (64 and 128 filters, kernel 5), each followed by ReLU activation, LayerNorm normalization and MaxPool1d subsampling, with Dropout (0.3). The output is passed to a recurrent module made of two stacked BiLSTM layers, with 256 hidden units per direction in the first layer (512 in total) and 128 in the second (256 in total), followed by temporal average pooling (global average pooling). The resulting vector passes through two fully connected layers with Dropout (0.3) for regularization; the last layer has nine outputs, one for each command.

The model was trained with the Adam optimizer (learning rate 0.001) and the CrossEntropyLoss objective. To improve robustness, the original set of about 1500 recordings was expanded through noise-based data augmentation [9], and an early stopping rule ended training when the validation accuracy stopped improving for ten consecutive epochs, which limited overfitting. Layer normalization was preferred over batch normalization because the small batch size and the variable length of the audio sequences make batch statistics unstable. The per-class behaviour of the trained model on the test set is summarized in the confusion matrix in Figure 4.

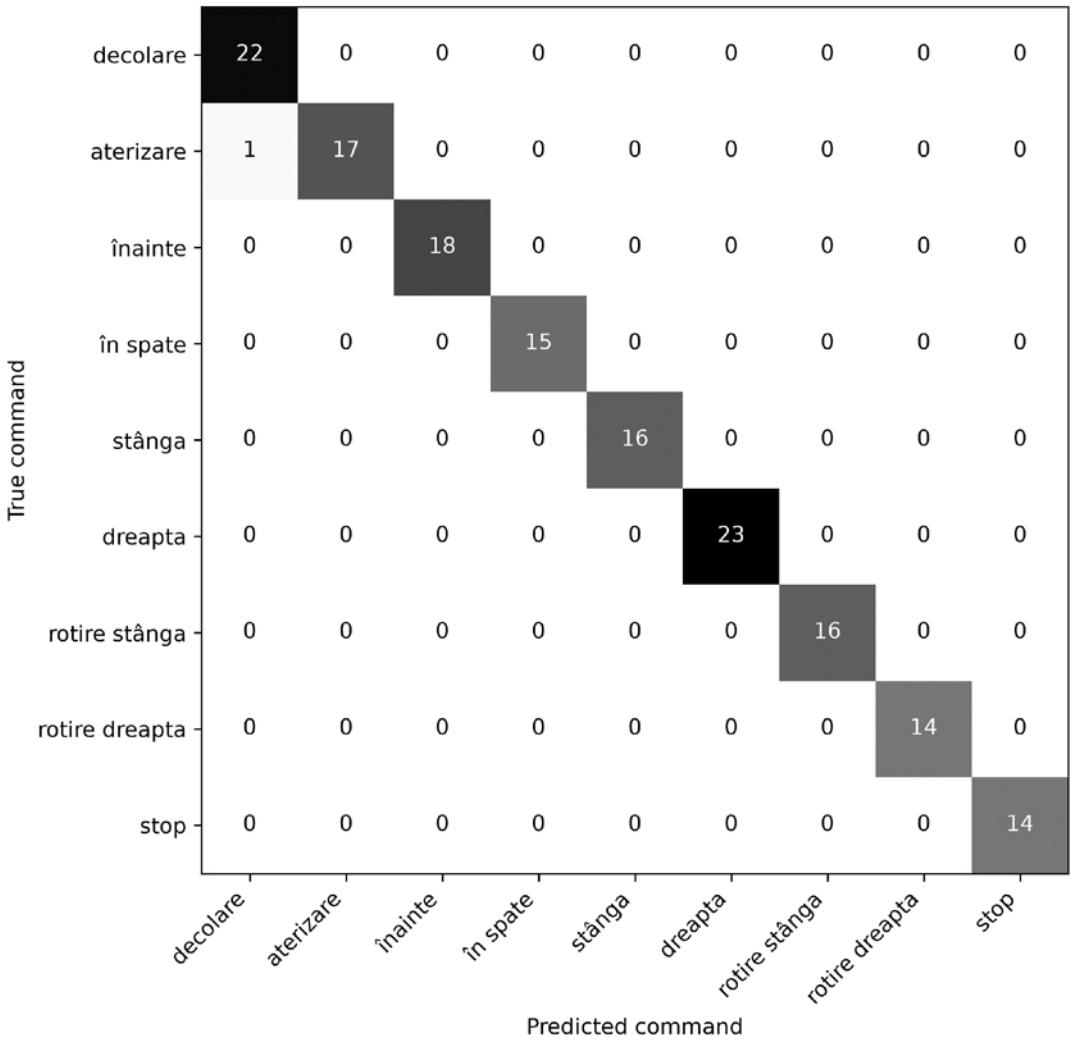

Figure 4. Confusion matrix of the CNN+BiLSTM model on the test set. Command labels are kept in Romanian, the language the model was trained on.

The matrix shows a near-diagonal structure: every command except one is recognized correctly, the single error being one "*aterizare*" (LAND) sample predicted as "*decolare*" (TAKE OFF), two short commands that begin with similar plosive sounds. No confusion appears between the compound commands and their short forms, for example "*rotire stânga*" (ROTATE LEFT) is never mistaken for "*stânga*" (LEFT), nor "*rotire dreapta*" (ROTATE RIGHT) for "*dreapta*" (RIGHT), which confirms that the stacked BiLSTM layers capture the temporal structure needed to separate them. The directional commands "*înainte*" (FORWARD) and "*în spate*" (BACK), which an operator might pronounce with overlapping vowels, are also kept apart without error. This balanced per-class behaviour, with no systematic weakness on any individual command, reflects the complementary roles of the convolutional stage, which extracts local spectral patterns, and the recurrent stage, which models how those patterns evolve in time. The complete layer-by-layer architecture that produces this behaviour is shown in Figure 5.

Figure 4 shows the full layer chain as implemented, from the sequence input of the 39 features to the softmax layer with nine outputs. The Deep Network Designer-style representation makes explicit the exact order of operations within each convolutional block (convolution, activation, normalization, subsampling, dropout) and how the two BiLSTM layers are chained before temporal aggregation.

Figure 5. CNN+BiLSTM model architecture.

This view was also used to check the consistency between the theoretical description and the actual model implementation.

2.3. Communication with the drone via the CRTP protocol

Communication between the laptop and the drone is carried out through the Crazyflie Real-Time Protocol (CRTP), a stateless real-time protocol that requires no handshake [10]. The 32-byte CRTP packets (1 header byte and 31 payload bytes) are transported through UDP datagrams to IP address 192.168.4.1, port 2390 of the drone, which acts as a WiFi access point. The measured communication latency is below 5 ms under normal conditions.

The recognized voice commands are translated into flight parameters (roll, pitch, yaw and thrust) and transmitted periodically to the drone. The take-off command initiates a progressive ramp-up of the thrust value over 2.4 seconds, avoiding abrupt movements, and a

3-second post-take-off immunity mechanism blocks accidental stop commands immediately after lift-off, preventing unintended landings caused by motor noise picked up by the microphone.

3. Case study

3.1. Software implementation and the state machine

The graphical interface, developed in Python with the CustomTkinter library, is structured into five functional sections: audio data collection, model training, validation, voice control and hardware configuration. The voice control section displays in real time the captured audio waveform, the recognized command and the associated confidence score, allowing the operator to monitor recognition quality. The code is organized according to clean architecture principles, with a clear separation of responsibilities between modules [11]. The complete implementation of the system is described in the authors' bachelor's thesis [12].

The transitions between the operating states of the drone controller (IDLE, CONNECTING, HOVER, MOVING, ROTATING and LANDING) are managed by a state machine, illustrated in Figure 6. From the IDLE state, starting voice control triggers the connection; a take-off command with high confidence brings the drone into the HOVER state, from which direction and rotation commands temporarily activate the MOVING and ROTATING states. Loss of the WiFi connection triggers an automatic transition to the IDLE state, with a controlled stop of the motors, ensuring predictable and safe behavior.

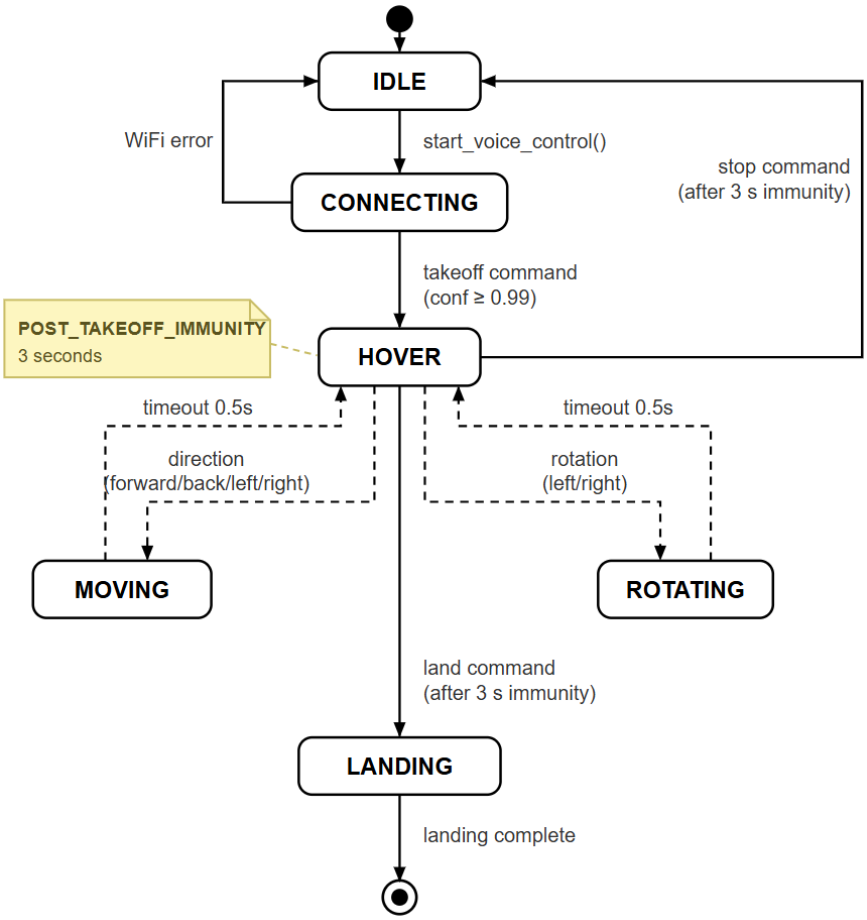

Figure 6. Drone controller state machine.

Organizing the controller as an explicit state machine guarantees that each command is interpreted only in the context where it is valid: a take-off command is ignored while the drone is already flying, and a direction command has no effect before take-off. After a movement, the MOVING and ROTATING states return to HOVER once the command ends, so the drone defaults to a stable hover rather than continuing an action indefinitely. The transition back to IDLE on connection loss is the central safety feature, since it prevents the drone from drifting under the last received command when communication drops. Because every transition is tied to a well-defined event, a voice command, a confidence threshold or a connection status, the behavior of the system remains deterministic and easy to predict, which is essential for safe operation near the operator.

This organization of the states guarantees that each voice command is interpreted only in the context in which it is valid, eliminating nondeterministic behaviors and increasing flight safety. For example, a stop command is accepted in HOVER, MOVING or ROTATING, but is ignored in IDLE or CONNECTING, where it has no meaning. The post-takeoff immunity window adds a further safety layer by blocking stop commands for three seconds after lift-off, preventing accidental landings caused by motor noise. Loss of the WiFi connection in any active state triggers a controlled transition back to IDLE, so the drone never remains in an undefined state when communication drops.

3.2. Experimental results

The system was evaluated under real flight conditions. The CNN+BiLSTM model reached an accuracy of 99.4% on the validation test set, and the measured end-to-end latency, from speaking the command to the drone's reaction, fell within the 210–350 ms range, compatible with real-time control. The latency of the CRTP communication layer remained below 5 ms. The main parameters and results of the system are summarized in Table 1.

Table 1

Parameters and performance of the voice control system

Parameter	Value
Recognized language	Romanian
Recognized voice commands	9
Sampling rate	16 kHz
Analysis window	2 s
MFCC coefficients per frame	39
Model architecture	CNN (2× Conv1d) + BiLSTM (512)
Dataset	~1500 samples
Test accuracy	99.4%
End-to-end latency	210 – 350 ms
CRTP communication latency	< 5 ms
Hardware platform cost	~40 USD

The values in Table 1 confirm that a low-cost platform (~40 USD) can support real-time local voice recognition, with an accuracy comparable to that of cloud-based commercial

solutions, but without dependence on connectivity. To document the in-flight behavior, Figure 7 shows a sequence captured during voice operation of the drone.

Figure 7. Drone airborne while executing the voice command "*înainte*" (*forward*).

The in-flight test confirms that the system works under real conditions: the "forward" command was recognized and sent to the drone with no perceptible delay, and the motion was controlled, with no abrupt attitude corrections. The post-take-off immunity window prevented accidental interpretation of a command during initial stabilization, which kept the flight stable throughout the sequence. Together with the values in Table 1, this shows that the system responds correctly to voice commands. Testing outdoors also exposed the system to conditions absent from the controlled recordings, such as wind noise and ambient park sounds, yet the command was still recognized on the first attempt, which suggests that the noise-based data augmentation helped the model generalize beyond the training environment. The operator kept both hands on the laptop throughout the maneuver, illustrating the hands-free interaction that motivates the whole approach: the drone was guided by voice alone, without any physical controller. The short, perceptible gap between speaking the command and the drone reacting stayed within the latency range reported in Table 1, confirming that local, offline processing is fast enough for responsive control. Taken together, the laboratory measurements and this field observation indicate that the system behaves consistently when it leaves the test environment.

4. Discussion

The results confirm the working hypothesis: a voice control system that runs entirely locally can achieve an accuracy and a latency compatible with real-time piloting, without depending on cloud infrastructure. The main advantages of the solution are fully local operation, the low cost of the platform, support for Romanian and adaptation to the individual user's voice, which makes it accessible even to people with speech difficulties.

The solution also has limitations. The vocabulary of nine commands covers the fundamental movements but restricts complex maneuvers; training on the individual voice increases accuracy but requires a data-collection stage for each new user; and the operating range is limited by the coverage of the local WiFi network. Compared with cloud architectures, the system reduces latency from 500–1200 ms to 210–350 ms and eliminates the dependence on the internet, at the cost of a more restricted vocabulary and a local training stage.

The identified development directions target continuous command recognition, vocabulary expansion and porting the TinyML model directly onto the drone, which would

reduce latency below 100 ms and completely eliminate the need for a ground station. Periodic retraining on the user's voice could further improve accuracy in noisy environments.

5. Conclusions

The local voice control system for drones analyzed in this paper demonstrates that hands-free aerial piloting is a functional technical reality, experimentally validated with an accuracy of 99.4% and a latency compatible with real-time control. The solution removes the need for manual interaction between the operator and the drone and instead offers the possibility of full guidance through voice commands, keeping an optimal balance between performance, cost and independence from cloud infrastructure.

The practical impact appears on two complementary levels. On the professional level, voice control increases operator efficiency in industrial inspections, mountain rescue, agricultural monitoring and security operations, domains in which free hands are an operational necessity. On the social level, the system opens access to drone technology for people with motor disabilities of the upper limbs, giving them autonomy and the capacity to act. The open-source nature of the entire platform ensures the reproducibility and extensibility of the solution, including in resource-limited environments.

These results were obtained on a low-cost platform built around an ESP32-S2 microcontroller, which shows that the approach does not depend on specialized or expensive hardware and can be reproduced with widely available components. Because the recognition model runs entirely on the ground station, the system can be extended with new commands or adapted to a new user's voice without any change to the drone itself, which keeps the path open for future improvements. The main directions that follow from this work are enlarging the command vocabulary, moving the model directly onto the drone to remove the need for a ground station, and validating the system across more speakers and noisier environments. Taken together, the findings confirm that local, offline voice control is a practical and accessible way to pilot drones, with concrete benefits for both professional operators and users with motor disabilities.

Research Data Availability Statement: No new public datasets were created. The audio recordings used for training were collected by the authors and are not shared due to privacy considerations regarding the voice samples.

Acknowledgments. This work was carried out within the “Research on intelligent methods and effective means for monitoring forest, perennial and aquatic areas with the application of drones”, 25.80012.5007.71SE. During the preparation of this manuscript, Gheorghe Bulai used Claude (Claude Opus 4.5, Anthropic) for translating the text from Romanian to English.

Contribution of authors.

Bulai Gheorghe: Conceptualization, Investigation, Methodology, Writing – original draft.
Viorel Bostan: Funding acquisition, Supervision, Project administration.
Rotaru Lilia: Investigation, Writing – editing.
Bumbu Tudor: Conceptualization, Supervision, Writing – review.

Conflicts of Interest. The authors declare no conflict of interest.

References

1. Pal, O.K., Shovon, M.S.H., Mridha, M.F. and Shin, J. (2024) In-depth review of AI-enabled unmanned aerial vehicles: Trends, vision, and challenges, Discover Artificial Intelligence, 4(1), 97. https://doi.org/10.1007/s44163-024-00209-1.
2. Valavanis, K.P. and Vachtsevanos, G.J. (eds.) (2015) Handbook of Unmanned Aerial Vehicles. Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-90-481-9707-1.
3. Tsoukas, V., Gkogkidis, A., Boumpa, E. and Kakarountas, A. (2024) A review on the emerging technology of TinyM', ACM Computing Surveys, 56(10), pp. 1–37. https://doi.org/10.1145/3661820.
4. Di Leo, S., De Cicco, L. and Mascolo, S. (2025) Real-Time Speech-to-Text on Edge: A Prototype System for Ultra-Low Latency Communication with AI-Powered NLP, Information, 16(8), 685. https://doi.org/10.3390/info16080685.
5. Picovoice (2025) Speech-to-Text Latency: How to Read Vendor Claims. Available at: https://picovoice.ai (Accessed: May 2026).
6. Espressif Systems (2024) ESP-Drone Documentation – Get Started. Available at: https://docs.espressif.com (Accessed: March 2026).
7. Lyons, R.G. (2011) Understanding Digital Signal Processing. 3rd edn. Upper Saddle River, NJ: Prentice Hall.
8. Hochreiter, S. and Schmidhuber, J. (1997) Long Short-Term Memory, Neural Computation, 9(8), pp. 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
9. De Simone, G., Greco, A., Rosa, F., Saggese, A. and Vento, M. (2025) Context-aware data augmentation for enhanced speech command recognition in industrial environments, Scientific Reports, 15(1), 17445. https://doi.org/10.1038/s41598-025-01886-3.
10. Bitcraze AB (2024) CRTP – Communication with the Crazyflie. Available at: https://www.bitcraze.io (Accessed: March 2026).
11. Martin, R.C., Grenning, J., Brown, S. and Henney, K. (2018) Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston, MA: Prentice Hall.
12. Bulai, G. (2026) Voice Control System for Piloting Drones. Bachelor's thesis. Technical University of Moldova, Chișinău.

Citation: Bulai, Gh., Bostan, V., Rotaru, L, Bumbu, T. (2026). Voice control system for piloting drones using artificial intelligence techniques. *Journal of Engineering Science*. 2026, 33 (2), pp. 7-19. https://doi.org/10.52326/jes.utm.2026.33(2).01.

Submission of manuscripts: jes@meridian.utm.md