

NeuroVision AI: An Integrated Framework for Neurological Disease Screening Using MRI, Voice, and EEG Data

Dr. Avinash G M¹, Ashwini Hiregoudru², Paladugu Tejaswini³, Syeda NaaZ Z⁴, Namitha J⁵

¹Associate Professor Department of CS&E, SJMIT, Chitradurga, India.

²Seventh Semester Student, Department of CS&E, SJMIT, Chitradurga, India.

³Seventh Semester Student, Department of CS&E, SJMIT, Chitradurga, India.

⁴Seventh Semester Student, Department of CS&E, SJMIT, Chitradurga, India.

⁵Seventh Semester Student, Department of CS&E, SJMIT, Chitradurga, India.

Abstract: *Neurological conditions such as Alzheimer's disease, Parkinson's disease, and epilepsy require analysis of different forms of biomedical information. This paper presents NeuroVision AI, a Flask-based web application developed as an educational demonstration of AI-based screening for these three conditions. The system provides independent processing pipelines for MRI images, voice recordings, and EEG data within a common web interface. The Alzheimer's module uses a TensorFlow/Keras EfficientNetB0 model for MRI image prediction. The Parkinson's module extracts features from WAV audio using Librosa and applies a Random Forest classifier, while the epilepsy module extracts statistical and frequency-domain features from EEG data and uses an ExtraTrees classifier. The application supports user authentication, file validation, prediction, result presentation, and prediction-history storage using SQLite. The implemented system was tested for authentication, module workflows, supported input handling, integration, prediction history, and basic usability. The project evaluation does not report final numerical clinical performance metrics; therefore, the results presented focus on functional implementation and software integration rather than clinical accuracy. The proposed system demonstrates how separate AI and signal-processing pipelines can be integrated into a unified educational software platform.*

Keywords: Multimodal AI, Neurological Screening, Alzheimer's Disease, Parkinson's Disease, Epilepsy, MRI, Voice Biomarkers, EEG, EfficientNetB0, Random Forest, Extra-Trees, Flask, SQLite.

I. INTRODUCTION

Neurological disorders such as Alzheimer's disease, Parkinson's disease, and epilepsy affect the brain and nervous system and require appropriate medical assessment. Different conditions require different forms of biomedical information. **MRI images** can provide structural information for Alzheimer's screening, **voice recordings** can contain measurable characteristics associated with Parkinson's disease, and **EEG signals** can be analyzed for patterns related to epileptic activity. Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) provide computational methods for processing these data types.

NeuroVision AI is developed as a unified educational web application rather than as three unrelated programs. It combines an image-based deep-learning workflow, an audio feature-extraction and machine-learning workflow, and an EEG feature-extraction and machine-learning workflow through a common Flask interface. The shared application also provides authentication, file handling, result presentation, and prediction-history functionality.

The system is intentionally positioned as an educational and demonstration platform. It is not a medical diagnostic system, and its predictions should not be treated as a substitute for professional medical evaluation. Accordingly, this paper focuses on the implemented software architecture, processing pipelines, integration, and functional testing described in the project report.

II. LITERATURE SURVEY

The literature survey on AI-based neurological disease screening highlights the application of deep learning and machine learning techniques for analysing diverse biomedical data. Tan and Le (2019) introduced EfficientNet, a convolutional neural network architecture that improves model scaling and is relevant to MRI-based Alzheimer's disease screening. Research by Kabir, Siuly, and Zhang (2016) investigated epileptic seizure detection from EEG signals using Logistic Model Trees. Ntahobari et al. (2022) explored an enhanced Extra-Trees classifier for epileptic seizure prediction. Çelik and Akbal (2025) examined voice-based feature extraction and classification for distinguishing Parkinson's disease patients. McFee et al. (2015) presented Librosa, a Python library for audio and music signal analysis, supporting audio preprocessing and feature extraction. TensorFlow/Keras and Scikit-learn provide essential frameworks for implementing deep learning and traditional machine learning models. However, the reviewed approaches primarily focus on individual diseases or specific biomedical data modalities. Integrating MRI, voice, and EEG analysis into a unified application remains an important software integration objective. The proposed NeuroVision AI addresses this requirement by combining EfficientNetB0, Random Forest, and ExtraTrees classifiers within a Flask-based web application. SQLite supports prediction-history storage, while common services provide authentication, file validation, and result presentation. Nevertheless, further evaluation using clinically validated datasets and standard performance metrics is required to establish the system's reliability and clinical applicability.

III. EXISTING SYSTEM

Existing AI-based neurological analysis systems are commonly designed around a particular disease or a particular data modality. One system may concentrate on MRI image analysis, while another may focus on voice characteristics or EEG signals. Such separate approaches can require different interfaces and processing procedures. They provide limited demonstration of how several modality-specific AI workflows can be connected through a common application layer.

The project therefore uses a unified application approach in which each neurological condition retains its own model and preprocessing requirements while sharing common web functions. This preserves the differences between modalities instead of forcing all inputs into one identical model.

IV. PROBLEM STATEMENT

Different neurological conditions require analysis of different types of biomedical data. Developing separate applications for MRI, voice, and EEG analysis can lead to different interfaces, input procedures, and processing workflows. A unified demonstration platform is therefore useful for showing how these modality-specific AI approaches can be integrated within one software system.

The project addresses the need for a common web-based environment in which a user can select a neurological screening module, upload the supported data, perform preprocessing or feature extraction, obtain a model prediction, and review prediction history. The system also needs to validate uploaded files and connect the application layer with the corresponding model artifacts.

V. OBJECTIVES

The main objectives of the proposed system are:

1. To develop a unified Flask-based educational web application for demonstrating AI-based screening workflows for Alzheimer's disease, Parkinson's disease, and epilepsy.
2. To process MRI images using an EfficientNetB0 model, WAV audio using feature extraction with a Random Forest classifier, and EEG data using feature extraction with an ExtraTrees classifier.
3. To provide common functions such as user authentication, file validation, result presentation, and prediction-history storage using SQLite.

4. To evaluate the implemented application through functional, integration, input-validation, system, and basic usability testing without making unsupported clinical-performance claims.

VI. PROPOSED SYSTEM

The proposed system follows a web-based architecture in which the user interacts with NeuroVision AI through a browser. The Flask application handles routing, authentication, file uploads, validation, and communication with the appropriate prediction module. The selected predictor performs preprocessing or feature extraction and loads the corresponding trained model artifact. The prediction result is returned to the web interface and stored in SQLite for history viewing.

The major modules are Alzheimer's screening from MRI images, Parkinson's screening from WAV audio, and epilepsy screening from EEG data. Although these modules use different algorithms, they are presented through one common application interface.

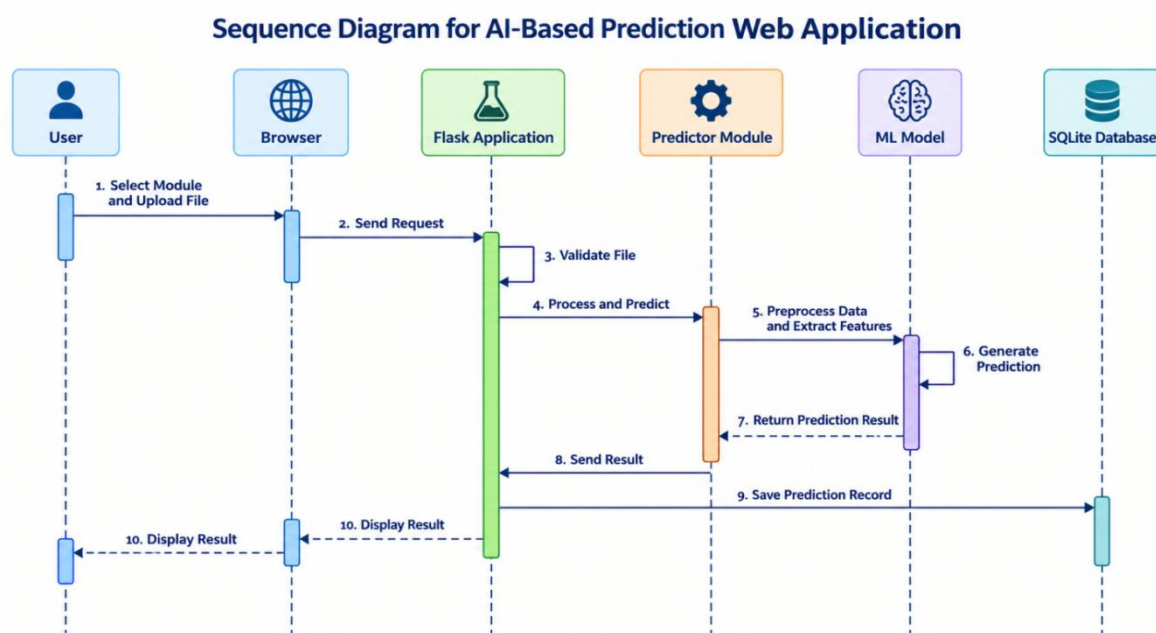


Figure-6.1: Architecture and interaction flow of the NeuroVision AI system

Table-6.1: The major machine learning models are:

Module	Algorithm / Processing
Alzheimer's Screening	EfficientNetB0 → MRI image prediction
Parkinson's Screening	Librosa/SoundFile feature extraction → Random Forest
Epilepsy Screening	41 statistical/frequency-domain features → ExtraTrees

VII. METHODOLOGY

The proposed system follows a modular methodology involving data upload, input validation, preprocessing or feature extraction, model prediction, result presentation, and prediction-history storage. Each disease module uses a processing pipeline appropriate to its input modality.

A. Data Input and Validation: The application accepts MRI images for Alzheimer's screening, WAV audio for Parkinson's screening, and EEG data in supported CSV/TXT form for epilepsy screening. Uploaded files are validated before processing.

B. Alzheimer's Module: The Alzheimer's pipeline accepts an MRI image, performs the required preprocessing, and uses a TensorFlow/Keras EfficientNetB0 model for prediction. The module returns a prediction result with the associated confidence information described in the project implementation.

C. Parkinson's Module: The Parkinson's pipeline accepts WAV audio, converts the audio to mono at 8 kHz, and extracts a 320-element feature vector using Librosa and SoundFile-based processing. The extracted representation is supplied to a Random Forest classifier, which returns a prediction label, confidence, and probability distribution.

D. Epilepsy Module: The epilepsy pipeline accepts EEG data and generates 41 statistical and frequency-domain features. The feature extraction includes power integration and band-energy calculations. These features are supplied to an ExtraTrees classifier for seizure/normal prediction.

E. Application and Database Integration: Flask routes connect the browser interface with the predictor modules. Jinja2 templates are used for the web pages, while SQLite stores user information and prediction history. A save_prediction helper is used for recording prediction results.

Table-7.1: Technologies used

Technology	Purpose
Python	Application and machine-learning implementation
Flask	Web application backend and routing
TensorFlow / Keras	EfficientNetB0-based MRI prediction
Scikit-learn	Random Forest and ExtraTrees classifiers
Librosa / SoundFile	Parkinson's audio processing and feature extraction
SciPy	Signal-processing and feature calculations
SQLite	User information and prediction history
Jinja2 / HTML	Web interface templates and result presentation

F. MACHINE LEARNING MODELS

- Alzheimer's Screening:** EfficientNetB0 is used for MRI image prediction through TensorFlow/Keras. The model is integrated into the application so that the uploaded MRI is processed and the prediction is displayed through the web interface.
- Parkinson's Screening:** The audio module uses WAV input and extracts a 320-element feature vector after audio preprocessing. A Random Forest classifier uses the extracted representation to produce the Parkinson's prediction output.
- Epilepsy Screening:** The EEG module generates 41 statistical and frequency-domain features, including power integration and band-energy calculations. An Extra-Trees classifier then produces the seizure/normal prediction.

The implemented predictors are connected to the application through separate model artifacts rather than being embedded directly in the Flask route logic. For Parkinson's screening, the trained Random Forest model is loaded from the project model directory, while the epilepsy predictor uses the Extra-Trees model artifact. This separation keeps the web layer responsible for request handling and presentation while the predictor modules perform modality-specific processing. The approach also makes the three workflows easier to maintain independently while preserving a common user experience.

For the Alzheimer's workflow, the uploaded MRI image is passed through the image-processing path and then to the EfficientNetB0 model implemented with Tensor Flow / Keras. For the Parkinson's workflow, WAV input is converted to mono at an 8 kHz sampling rate before the 320-element feature representation is produced. For epilepsy, the EEG input is processed to generate 41 statistical and frequency-domain features, including power integration and band-energy calculations. The resulting feature vectors are then supplied to the corresponding classical machine-learning classifiers.

VIII. RESULTS AND DISCUSSIONS

Table-8.1: Functional evaluation of the implemented system

Module / Function	Implemented Result
User Authentication	Functional login and registration workflow
Alzheimer's MRI Flow	MRI upload, preprocessing, EfficientNetB0 prediction and result display
Parkinson's WAV Flow	WAV validation, feature extraction, Random Forest prediction and result display.
Epilepsy EEG Flow	EEG input processing, 41-feature extraction, Extra-Trees prediction and result display
Prediction History	Prediction records stored and displayed using SQLite
System Integration	Flask routes, predictor modules, templates and database integrated
Input Validation	Supported file types and upload handling tested

The implemented system was tested through unit, integration, functional, system, input-validation, and usability-oriented testing. The reported testing covered authentication, the three disease-module workflows, prediction history, and integration between the application components. The application demonstrates the intended end-to-end software workflow from module selection and file upload to prediction, result display, and history storage.

Unlike a conventional experimental study that reports accuracy, precision, recall, F1-score, or AUC, the Phase 2 project report does not provide final numerical clinical-performance metrics for the three modules. Therefore, no unsupported numerical accuracy values are reported in this paper. The results are limited to the functional implementation and software-integration evidence available in the project.

Advantages: The proposed platform provides a common interface for three different biomedical data modalities, separates modality-specific processing from shared application services, supports prediction-history storage, and provides a practical educational demonstration of integrating deep learning and machine learning workflows in one web application. Its modular structure also allows future expansion with additional screening modules.

The output screens supplied with the implementation provide additional evidence of the end-to-end behavior of the three pipelines. The screenshots show that each module accepts its intended modality, displays the processed input or signal, reports the selected model, and presents a prediction together with a confidence or probability distribution. These outputs are useful as implementation evidence; however, the displayed confidence of an individual uploaded sample must not be interpreted as the overall accuracy of the trained model.

Table-8.2: Values displayed in the supplied demonstration outputs

Module	Observed demonstration output	Displayed value
Alzheimer's MRI	Prediction / confidence	Mild Demented / 100.0%
Parkinson's voice	Prediction / confidence	Positive / 99.0%
Epilepsy EEG	Prediction / confidence	Seizure / 99.6%
Alzheimer's MRI	Inference time / input size	0.84 s / 224 × 224
Parkinson's voice	Inference time / audio	0.32 s / 3.68 s
Epilepsy EEG	Samples / sampling rate	178 / 173.61

For the Alzheimer's demonstration, the interface reports a Mild Demented prediction with a displayed confidence of 100.0%. The same output screen identifies EfficientNetB0 and TensorFlow as the model and framework, respectively, and reports an inference time of 0.84 seconds with an input size of 224×224 . The screen also shows the original MRI, an AI attention map using Grad-CAM, a class-distribution panel, and a processing pipeline containing image upload, skull stripping, intensity normalization, feature extraction, CNN inference, classification, and report generation. These elements demonstrate the intended presentation and processing flow implemented in the application.

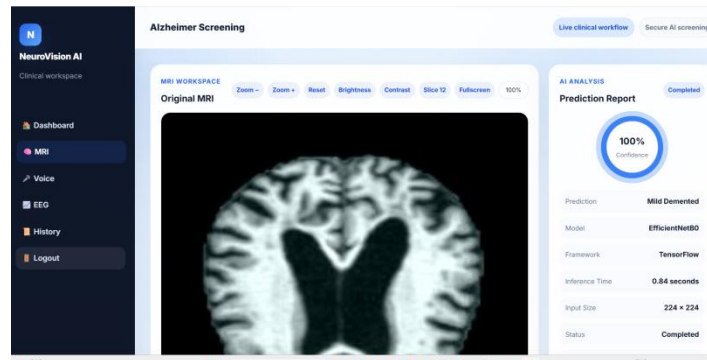


Figure-8.1: Alzheimer's screening output showing MRI input, EfficientNetB0 prediction, confidence and processing flow.

The Parkinson's output demonstrates the audio-processing path. The screen reports a Parkinson's Positive prediction with 99.0% displayed confidence and identifies Random Forest as the classifier and scikit-learn as the framework. It also reports an inference time of 0.32 seconds, an input duration of 3.68 seconds, and an 8,000 Hz sample rate. The accompanying probability distribution and waveform visualization show that the application presents both the classification result and the uploaded audio signal within the same interface.

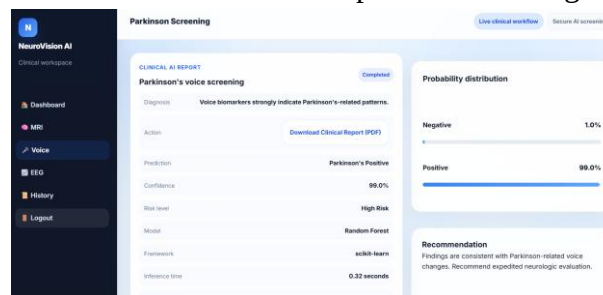


Figure-8.2: Parkinson's screening output showing Random Forest prediction, probability distribution and audio workflow.

For the epilepsy demonstration, the output screen reports Seizure Detected with 99.6% displayed confidence using the Extra Trees classifier. The screen shows 178 samples and a sampling rate of 173.61, together with a probability distribution for Normal and Seizure classes. The uploaded and processed EEG traces are also visualized, providing a direct view of the signal used for prediction. This supports the report's description of a pipeline in which EEG data are processed into statistical and frequency-domain features before classification.

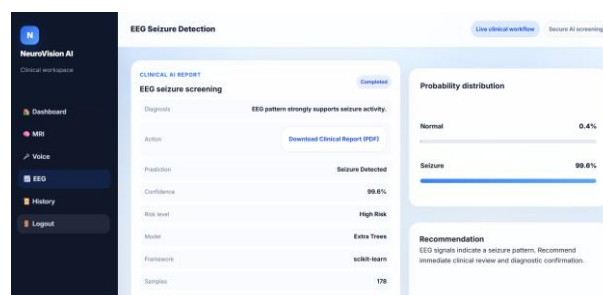


Figure-8.3: Epilepsy screening output showing Extra Trees prediction, probability distribution and EEG information.

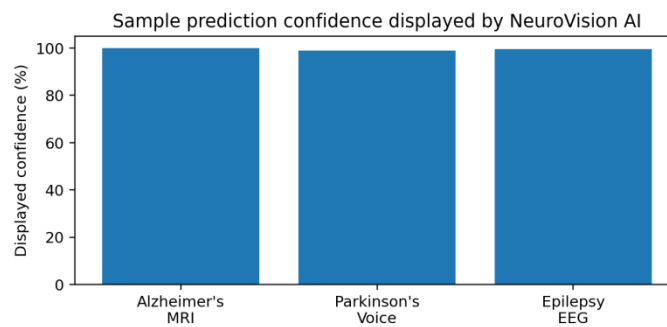


Figure-8.4: Displayed prediction confidence for the three supplied demonstration outputs; these values are not model accuracy metrics.

Figure-8.4 summarizes the confidence values displayed by the three supplied example outputs. The values are close to 100% for all three individual examples. This graph is intentionally presented as a visualization of the application output rather than as a performance comparison. A confidence value associated with one uploaded sample does not establish generalization, sensitivity, specificity, or clinical validity. Such claims would require evaluation on an appropriately separated and clinically validated test dataset.

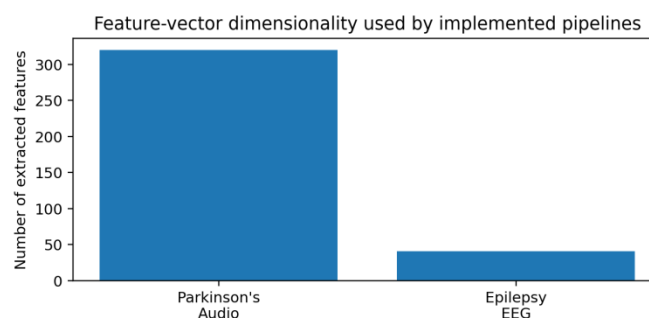


Figure-8.5: Feature-vector dimensionality reported for the Parkinson's and epilepsy pipelines.

Figure-8.5 compares the feature-vector dimensionality explicitly described in the implementation. The Parkinson's audio pipeline produces a 320-element feature vector, whereas the epilepsy pipeline generates 41 statistical and frequency-domain features. The difference reflects the modality-specific processing requirements and does not indicate that one classifier is superior to the other.

From a software-integration perspective, the results also show the benefit of keeping the three processing pipelines separate. MRI processing can use an image-oriented deep-learning model, while voice and EEG data can use feature-based classifiers without forcing all three modalities into a single representation. The Flask application provides the common layer for authentication, validation, routing, presentation, and history, while the predictor modules retain their modality-specific requirements.

Overall, the supplied results support the functional objective of integrating three independent modality-specific screening workflows into one web application. The screens demonstrate successful movement from input handling through preprocessing or feature extraction, model inference, and result presentation. Prediction history and database integration further support the software-level objective of retaining prediction records. At the same time, the evidence remains limited to the implemented demonstration and the supplied example outputs; it does not replace a formal clinical or statistical model evaluation.

LIMITATIONS

The proposed NeuroVision AI system is designed primarily as an educational and demonstration platform for AI-based neurological disease screening and is not intended for clinical diagnosis. The project evaluation focuses on functional implementation and software integration; therefore, clinical accuracy and validation by medical specialists have not been established. Furthermore, no clinical trials or real-world deployment evaluations have been conducted to assess the system's reliability under practical healthcare conditions.

Although the application integrates Alzheimer's, Parkinson's, and epilepsy screening within a common interface, it uses separate modality-specific processing pipelines rather than a single jointly trained multimodal AI model. For real-world deployment, the system would require larger and more diverse clinically validated datasets, enhanced security mechanisms, controlled biomedical data handling, reliable computing infrastructure, and comprehensive clinical and regulatory validation.

CONCLUSION

This paper presents NeuroVision AI, a Flask-based educational platform that integrates three neurological screening workflows within a common software application. The system uses EfficientNetB0 for MRI-based Alzheimer's screening, Random Forest with audio feature extraction for Parkinson's screening, and Extra-Trees with EEG feature extraction for epilepsy screening. Shared services provide authentication, file validation, result presentation, and prediction-history storage through SQLite. The implemented testing demonstrates the integration and functional operation of the application. However, the project does not establish clinical accuracy, and the system should therefore be treated as an educational demonstration rather than a medical diagnostic tool. Future work can focus on clinically validated datasets, improved models, real-time processing, security, deployment, and specialist validation.

FUTURE WORK

Future development can include larger and clinically validated datasets, improved deep-learning models, real-time analysis, cloud deployment, enhanced application security, scalable database infrastructure, clinical validation, and additional neurological screening modules. These improvements would be required before considering practical clinical use.

REFERENCES

- [1] an, M., and Le, Q. V., "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proceedings of the 36th International Conference on Machine Learning (ICML), vol. 97, pp. 6105–6114, 2019.
- [2] Kabir, E., Siuly, and Zhang, Y., "Epileptic Seizure Detection from EEG Signals Using Logistic Model Trees," Brain Informatics, vol. 3, no. 2, pp. 93–100, 2016. DOI: [10.1007/s40708-015-0030-2](https://doi.org/10.1007/s40708-015-0030-2)
- [3] Ntahobari, M., Kuhlmann, L., Boley, M., and Razavi Hesabi, Z., "Enhanced Extra Trees Classifier for Epileptic Seizure Prediction," in Proceedings of the 2022 5th International Conference on Signal Processing and Information Security (ICSPIS), pp. 175–179, IEEE, 2022. DOI: [10.1109/ICSPIS57063.2022.10002677](https://doi.org/10.1109/ICSPIS57063.2022.10002677)
- [4] Çelik, B., and Akbal, A., "Distinguishing Parkinson's Patients Using Voice-Based Feature Extraction and Classification," arXiv preprint arXiv:2501.14390, 2025. DOI: [10.48550/arXiv.2501.14390](https://doi.org/10.48550/arXiv.2501.14390)
- [5] Tan, M., and Le, Q. V., "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," arXiv preprint arXiv:1905.11946, 2019
- [6] McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., and Nieto, O., "librosa: Audio and Music Signal Analysis in Python," in Proceedings of the 14th Python in Science Conference (SciPy 2015), pp. 18–24, 2015. DOI: [10.25080/Majora-7b98e3ed-003](https://doi.org/10.25080/Majora-7b98e3ed-003)
- [7] Python Software Foundation, "sqlite3 — DB-API 2.0 Interface for SQLite Databases," Python Documentation. Available: <https://docs.python.org/3/library/sqlite3.html>
- [8] Pallets Projects, "Flask Documentation," Flask: A Python Web Application Framework. Available: <https://flask.palletsprojects.com/en/stable/>

Copyright & License:

© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.