Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Identification of Musical Notes using Support Vector Machine

Siddhi Jagtap, Jagruti Wagh, Aakash Kasabekar, Kiran Mogal, Tanaya Nene, Sankirti Shiravale

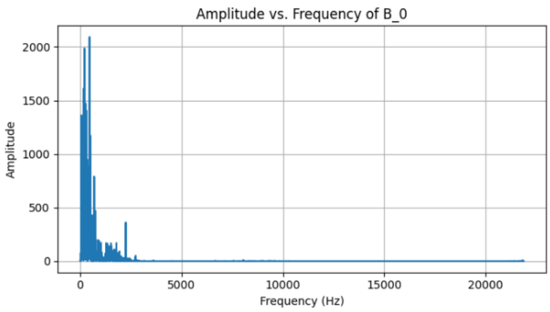
Marathwada Mitra Mandal's College of Engineering, Pune, India
Email: siddhijagtap802@gmail.com

Music is the universal language that profoundly unites cultures and generations all over the world. The ability to recognize notes in music is essential in many fields, including composing, teaching music, and digital audio processing. It is essential for both music education and composition to be able to accurately identify and classify musical notes and chords. Using Mel-frequency Cepstral Coefficients (MFCC) and Support Vector Machine (SVM), we present a novel Music Note Recognition System in this study which identifies the notes present in the input audio file. To meet the needs of musicians, non-musicians as well as audio enthusiasts, the system attempts to accurately identify musical notes from audio samples. The experiments show that the recommended method is effective. Comparison analysis shows the advantages of SVM classification and MFCC-based feature extraction over existing techniques, which result in better performance. Beginners can use this system to learn and practice musical instruments, while musicians can use it to compose, transcribe, and analyze musical pieces. Furthermore, the system's durability and adaptability make it a good fit for integration with recent advancements in audio processing and music production software.

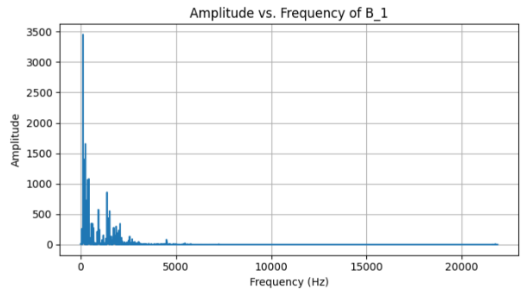
Keywords: Mel Frequency Cepstral Coefficients (MFCC); Machine Learning; Neural Networks; Fast Fourier transform (FFT); Support Vector Machine (SVM); Radial Basis Function (RBF)

1. Introduction

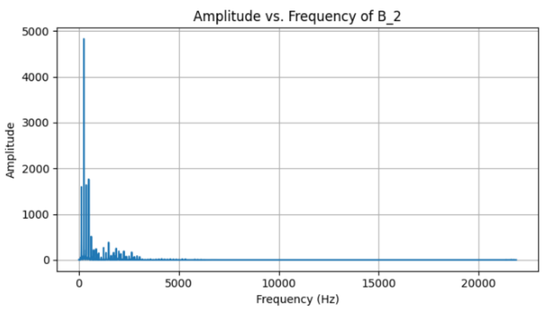
The project aims to develop a Musical Note Recognition System that is based on Machine Learning and can identify notes based on input from the user. The development of a Music Note Recognition System stemmed from a desire to enrich the learning experience for budding musicians. This technology finds utility across various domains within the music industry due to its versatility. By automatically identifying musical notes in audio files, it streamlines tasks such as sound editing, pitch correction, and sample modification, allowing producers to allocate more time to creative expression and less on technical intricacies, thereby enhancing the music production workflow.



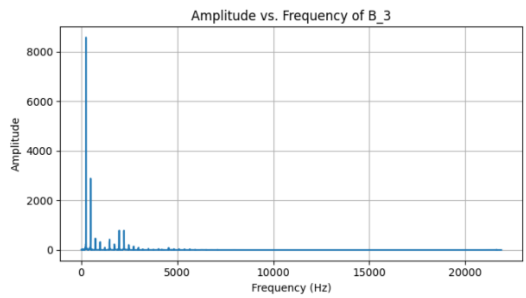
a. Note B0



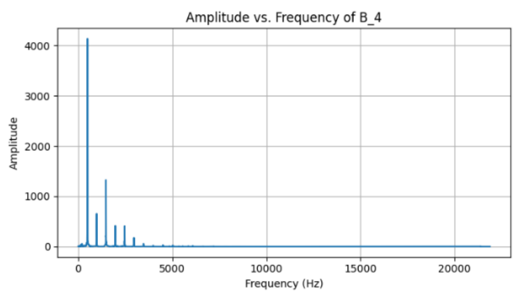
b. Note B1



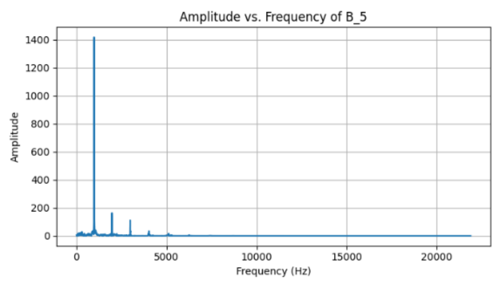
c. Note B2



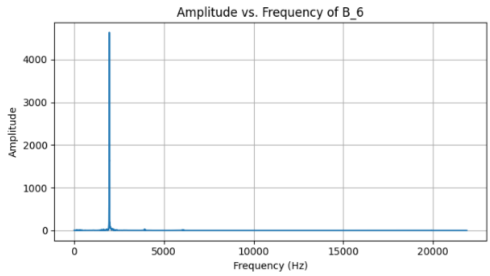
d. Note B3



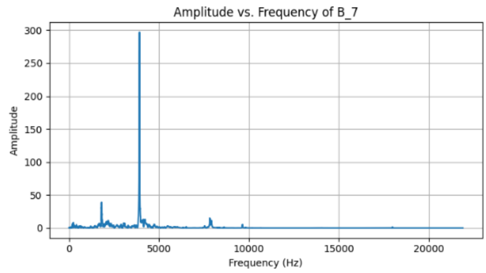
e. Note B4



f. Note B5



g. Note B6



h. Note B7

Figure 1. Amplitude vs. Frequency Variations

The fundamental components of music theory encompass notes, chords, scales, pitch and octaves. Notes, serving as the foundational units, possess distinct pitches and durations, forming the basis of musical compositions. Chords, created by playing multiple notes simultaneously, contribute depth and complexity to musical arrangements. Scales provide a framework for melodies and harmonies, being ordered sets of notes. Pitch is how we perceive the highness or lowness of any sound. Combining different pitches creates chords which increases with more sources. It's the quality of sound that matters. For instance, when played on a piano or a guitar, note A4 is heard as having the same pitch (Gfeller et al., 2020). Additionally, octaves facilitate smooth modulation of pitches while preserving the inherent qualities of individual notes, adding depth to musical compositions. Each octave on a piano keyboard consists of eight notes, distinguished by the arrangement of white and black keys.

On a piano, black keys indicate sharps and flats and white keys indicate natural notes (A, B, C, D, E, F, G). This allows for a more organized approach to exploring music. The fundamental connection between frequency values and octave intervals is illustrated by harmonic progression, which happens when the frequency of each octave doubles that of the one before it. The Amplitude vs. Frequency variation of note B across several octaves is displayed in Figure 1. It illustrates how a note's frequency varies in multiples. For instance, if B4 is at 494 Hz in frequency, B3 will be at 247 Hz, and B5 will be at 988 Hz. This demonstrates that as the octave changes, frequency doubles.

Originally, the project aimed to utilize the NSynth Dataset, comprising over 300,000 musical notes from diverse instruments. However, due to its complexity, focus shifted to a piano-specific dataset from Kaggle, better suited for model development. Recognizing that chords compromised accuracy, a switch was made to a dataset containing single piano notes, significantly enhancing model performance and training efficacy (Wang, 2023).

Several Python libraries, including Librosa and PyAudio, facilitated the processing and analysis of audio data. Mel-frequency Cepstral Coefficients (MFCC), which are useful for speech and music analysis applications because they can efficiently capture frequency in a limited number of coefficients, were used to extract features. Initially organized using a sequential model architecture, it became apparent that this approach was inadequate for capturing the complex patterns in musical data. Thus, a Support Vector Machine (SVM) architecture was adopted for its ability to handle high-dimensional data and define complex decision boundaries, crucial for music note classification (Fu et al., 2011; Ghosal et al., 2012). SVMs excel in handling non-linear relationships without overfitting, thus achieving higher accuracy and generalization in music note recognition tasks (Rajadnya & Joshi, 2021).

In conclusion, an iterative approach to feature engineering, model optimization, and dataset selection has yielded a reliable music note recognition system. By leveraging machine learning techniques, particularly Support Vector Machines (SVMs) and integrating Mel-frequency Cepstral Coefficients (MFCCs), accuracy and efficiency in real-time music note recognition tasks have been significantly improved.

Table 1 gives an overview of every Note and its frequency at respective Octave.

- The frequencies are in Hertz (Hz)
- Oct: Octave

Table 1. Frequencies of Notes at different Octaves

	Notes											
Oct	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
Oct 0	16.35	17.32	18.35	19.45	20.6	21.83	23.12	24.5	25.96	27.5	29.14	30.8
Oct 1	32.7	34.65	36.71	38.89	41.2	43.65	46.25	49	51.91	55	58.27	61.7
Oct 2	65.41	69.3	73.42	77.78	82.41	87.31	92.5	98	103.83	110	116.54	124
Oct 3	130.8	138.6	146.8	155.56	164.81	174.61	185	196	207.65	220	233.08	247
Oct 4	261.6	277.2	293.6	311.13	329.63	349.23	369.99	392	415.3	440	466.16	494
Oct 5	523.2	554.4	587.3	622.25	659.26	698.46	739.99	783.99	830.61	880	932.33	988
Oct 6	1046	1108	1174	244.51	1318.5	396.91	479.98	567.98	661.22	760	864.66	976
Oct 7	2093	2217	2349	489.02	637.02	793.83	959.96	135.96	322.44	352	729.31	951
Oct 8	4186	434.9	4698	4978.1	5274.1	5587.65	5919.9	6271.9	6644.8	704	7458.6	7902

2. Literature Review

After conducting a comprehensive literature analysis, we were able to pinpoint a glaring gap in the corpus of knowledge on note identification in music. It was discovered that this particular discipline has not received much attention because the majority of research has been done on more general topics like machine learning for music composition, digital signal processing, and genre classification.

Since there has not been much research done in the topic of note identification, our work offers a chance to add fresh perspectives and techniques to the relatively uncharted field. This insight inspired us to explore the nuances of music note recognition in order to fill this significant void in the literature and deepen our understanding.

According to Youssef & Woo (2008), which is highly relevant to our project's focus and deals with the identification of musical notes. It employs a Time-Delay Neural Network (TDNN) to effectively reduce noise. This article presented a novel note recognition algorithm. The next steps in the process were putting the adaptive pattern categorization subsystem into practice. The two distinct stages of this subsystem were the Self-Organizing Map (SOM) and the Learning Vector Quantization (LVQ). All of these steps worked together to coordinate the complex process of note recognition. The study's complex neural network architectures pose the risk of overfitting, in which the model becomes excessively adapted to its training set, making it more difficult for it to generalize to different musical contexts.

Another noteworthy study that was looked by Mounir et al. (2021) at the use of Spectral Sparsity Measure in the context of music was discovered during the literature review. This work presented a novel approach to enhance the identification process: note onset features were assessed without accounting for low-magnitude spectral components. The non-data-driven techniques rely on several criteria, including pre-processing, reduction, and input preparation, because of which, it is difficult to determine the best combination of parameter values.

Additionally, in the field of signal processing, Othman and Abidin (2021) have presented a different approach. Using samples from both acoustic and electric guitars, this research presents a real-time note recognition algorithm. The developed program can identify the pitch-related characteristics of the notes that the chosen instruments generate by using the signal processing facilities in MATLAB, particularly cepstral analysis (Bhalke et al., 2011). The study highlights the algorithm's potential for use in music production and analysis by showing how well it can identify and match notes from both acoustic and electric guitars. For researchers who are not familiar with MATLAB programming or do not have access to MATLAB tools, relying solely on MATLAB for method development may limit accessibility.

Furthermore, we discovered a study project by Raguraman et al. (2019) that builds an evaluation instrument for Music Information Retrieval (MIR) systems with the help of the Python package Librosa. This tool shed light on the complex process of evaluating musical content by demonstrating proficiency in differentiating key characteristics and providing an effective grading sheet. Together, these discoveries deepened our comprehension and shaped our strategy for creating a reliable music note recognition system. The accessibility of benchmark audio files, or "teacher" files, are critical to the evaluation process' efficiency. The model's capacity to precisely evaluate user performance across a broad spectrum of musical styles and genres may be hampered by restricted access to extensive and varied benchmark files.

Using MFCC features, we focused on creating and training an SVM with an RBF kernel for music note recognition (Varughese & Kalaivani, 2023). In order to accurately represent non-linear relationships in the data and represent the intricacy of musical notes, the SVM needs the RBF kernel. When working with high-dimensional data, such as MFCC features, SVMs perform well and are resistant to overfitting. Utilizing TensorFlow's Keras API, we first attempted to use the sequential neural network architecture (Puri & Mahajan, 2019; Suguna et al., 2023). However, moving from neural networks to SVM produced more encouraging outcomes. Utilizing an alternative machine learning technique, the methodical and structured approach stays constant in its goal of developing a dependable musical note classification system.

3. Dataset

We used a meticulous process to put together a wide range of dataset to create a trustworthy model for categorizing piano notes. Various octaves and dynamic ranges of distinct piano notes are represented by the diverse range of audio samples in this collection. Inside the 'piano-notes' directory, the dataset is neatly organized to preserve organization. Consistent with naming conventions, each audio file is labelled with the musical note it represents and the corresponding octave. As a result, supervised learning is guaranteed to have access to precise ground truth labels. In addition to extracting the musical note labels from the filenames, we used the Librosa library to load each audio sample and standardize its duration to one second. Ensuring the consistency and relevance of the dataset to our classification task was made possible by these crucial steps. We started with 88 clean notes in our dataset, which we further refined using more cleaning methods. However, we used data augmentation techniques to increase the dataset because we realized that there was a need for more comprehensive data coverage.

In order to ensure a rich and diverse dataset for training and assessing the model, we increased the number of audio notes to 440. Data augmentation techniques for audio include time stretching, pitch shifting, noise injection, spectrogram manipulation, and speed perturbation. We used the Pitch Shifting Data Augmentation technique. The pitch of the audio is changed while maintaining its duration. This technique is generally used to simulate variations of the voice or musical notes. The model's efficacy and accuracy in categorizing piano notes are greatly improved by this augmentation process.

The key to the model's ability to generalize well across different musical contexts is the diversity of our dataset. Through incorporating a variety of playing styles, tempos, and octave ranges, along with both sharp and flat notes, we improve the model's ability to identify finer details in piano compositions. A strong foundation for in-depth study on musical note classification in piano recordings is also provided by the dataset's size and variability, which enhance the model's resilience and enable it to capture the nuances of piano.

4. Methodology

The categorization of musical notes in audio samples is the core of our research's suggested methodology. To ensure uniformity and suitability for model training, the dataset, which consists of audio recordings of piano notes, is carefully pre-processed, including augmentation. Using the Librosa library, the project starts with loading the audio samples and applying a fixed length, which is essential for creating uniformity throughout the dataset. Musical notes are simultaneously extracted from the filenames to provide the ground truth labels needed for further supervised learning.

For the SVM model to receive meaningful input from raw audio data, effective feature extraction is necessary. The extracted features, which are primarily MFCCs, are used to train the SVM model for classification (Rajadnya & Joshi, 2021; Jang et al., 2019). The RBF kernel is employed for multiclass classification. Figure 2 represents an overview of the system.

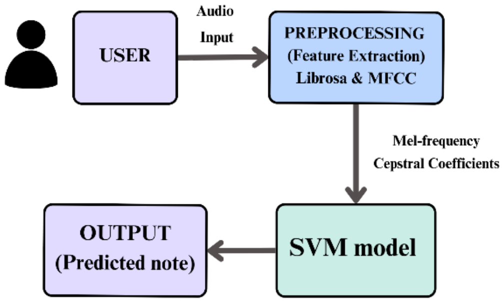


Figure 2. System Overview

4.1 Mel-Frequency Cepstral Coefficient

MFCCs offer information about the distribution of spectral energy. Details regarding the changes in spectral energy across various frequency bands are encoded by MFCC coefficients. A positive cepstral coefficient indicates that most energy is found in lower frequency regions; a negative cepstral coefficient indicates that most energy is found in higher frequency regions.

1. Divide the audio signal into discrete frames in order to record shifting frequencies over time.
2. To reduce spectrum leakage brought on by the Fourier Transform's assumption of infinite data, utilize windowing.
3. Calculate the frequency spectrum for each frame using an N-point FFT to create a periodogram. A periodogram is a tool used in signal processing to measure the strength or power of different frequencies present in a signal.

$$x_k = \sum_{m=0}^{n-1} x_m e^{-i2\pi km/n} \qquad k = 0,\ldots.n-1 \quad (1)$$

where,

$e^{-i2\pi/n}$ is a primitive n^{th} root of 1

the total number of samples in the signal is denoted by n

k has a range of 0 to $n-1$

4. To extract spectral features, Mel spaced Filter Banks are used, which are usually composed of 20–40 triangle filters.
5. Log filterbank energies are obtained by taking the logarithm of the spectrogram values.
6. Use the Discrete Cosine Transform (DCT) for decorrelation in order to handle the problem of highly correlated filter bank coefficients.
7. The resulting coefficients give a concise representation of the audio elements and are referred to as Mel Frequency Cepstral Coefficients (MFCCs).

4.2 Support Vector Machine

Experimental studies show SVM is superior to handle non-linear data classification. Music notes classification can benefit from the flexibility and efficiency of Support Vector Machines

(SVM), machine learning techniques that can handle vast volumes of data with nonlinear correlations.

SVM involves determining which hyperplane in the feature space best divides data points into different groups. Support vectors, or the data points that are closest to the hyperplane, are crucial in determining the hyperplane. The hyperplane maximizes the margin between the nearest points of several classes to offer dependable classification even in the presence of outliers.
The size of the hyperplane is determined by the number of features in the dataset.

SVMs can be divided into two parts:

1. Linear SVM:

In linear SVM, data points are divided into distinct classes by Linear Support Vector Machines using a linear decision boundary, also known as a hyperplane. This approach works best when the data can be divided into classes in a clear and comprehensive manner.

2. Non-linear SVM:

The nonlinearity present in the data is handled by using more powerful kernel functions which can transform original input into high dimensional space. A linear hyperplane, which in this transformed space corresponds to a non-linear decision boundary in the original feature space, can be used to partition the data. Radial basis function (RBF), sigmoid, and polynomial are examples of frequently used kernel functions.

This work suggests using the Radial Basis Function (RBF) kernel for musical note classification in a Support Vector Machine (SVM) architecture. The accuracy of note classification is increased by employing Mel-Frequency Cepstral Coefficients (MFCC) as feature inputs since the RBF kernel can handle the non-linear correlations present in audio data better. For tasks involving the retrieval of music information, this method presents an appropriate answer to the difficulties caused by the complex and variable character of audio signals.

This method calculates the proximity between two points, x_1x_1 and x_2x_2, to determine the degree of similarity between them. This similarity, which is mathematically computed by the RBF kernel and described as follows, is essential for identifying patterns in data:

$$K(x_1, x_2) = exp\left(-\frac{||x_1 - x_2||^2}{2\sigma^2}\right) \qquad (2)$$

where,

σ: hyperparameter i.e. variance

$x_1 - x_2$: Euclidean distance between 2 points

In the context of Music note recognition system, SVM plays a crucial role in classifying the extracted features from audio files into specific musical notes. Here is an overview of how SVM works in the context of note extraction:

1. Feature Extraction (MFCC with Librosa):

Librosa, a Python library for audio and music analysis, helps extract MFCCs from the audio files. In order to capture the spectrum properties of the music, audio and speech processing often uses representations called Mel-frequency cepstral coefficients, or MFCCs. They highlight important musical elements by offering a condensed representation of the audio signal's frequency content.

2. Data Representation:

To represent distinct notes, the SVM model uses the extracted MFCCs from the input audio as its input.

3. SVM Training:

The SVM model must next be trained on a labelled dataset in order to identify correlations between input MFCCs and musical notes.

4. Classification of notes:

Every audio frame is classified into one of several musical note classes by the trained SVM.

5. Note Sequence Reconstruction:

By utilizing Support Vector Machines (SVM) to process consecutive frames, the model can reconstruct the entire musical note sequence.

The performance of the model is evaluated using metrics such as accuracy. This methodology is a reliable tool for both professionals and music amateurs, as it ensures precise identification of musical notes inside a given audio file.

The efficiency of our method is shown not only in the training and assessment of the model but also in the real-world use case of musical note prediction from user-supplied audio files. We demonstrate the adaptability and practicality of our classification system by storing the trained model for later use. In summary, our methodology presents a methodical and all-encompassing approach to musical note classification in piano audio samples, offering significant insights into model development and real-world application.

4.3 System Architecture

The architecture consists of multiple phases, such as feature extraction, data preprocessing, model training, and inference. Piano notes recorded at different frequencies and durations make up the dataset from which raw audio samples are first taken. Before being subjected to further analysis, the samples are pre-processed.

1) Pre-processing:

To guarantee length uniformity which entails truncating longer audio clips to a set length and padding shorter ones. The audio signals are additionally converted into frequency domain representations using the Fast Fourier Transform (FFT) method to help with feature extraction (Othman & Abidin, 2021).
Feature extraction, the process of deriving meaningful representations from the audio signals, is crucial to the success of model learning. In this research, the Librosa library is used to calculate Mel-Frequency Cepstral Coefficients (Molau et al., 2001). MFCCs are useful features for musical note recognition tasks because they capture important aspects of audio signals, like pitch and timbre (Szczerba & Czyzewski, 2005; Guerrero Turrubiates et al., 2014). To improve the feature set, additional spectral features such as the magnitudes of FFT coefficients are used.

The dataset is then split into training and testing sets at a ratio of 1:5 in order to create and evaluate models, respectively. In order to prevent bias in the learning process, the training data is further standardized using methods such as StandardScaler to guarantee that features are on a similar scale. Moreover, LabelEncoder is used to encode the labels that correspond to musical notes in order to make them compatible with machine learning algorithms.

2) SVM model:

Support Vector Machines (SVMs) are chosen due to their ability to handle high-dimensional data and their strong ability to generalize to new cases. An SVM model with a radial basis function (RBF) kernel is trained on the standardized feature vectors in order to identify the underlying patterns in the data. The trained SVM model's satisfactory accuracy in identifying notes is evaluated using the test set.
The test set is used to evaluate the trained SVM model's generalization ability and accuracy in predicting musical notes. Note identification tasks are used to evaluate the model's performance using performance metrics like F1 score, recall, accuracy, and precision. Lastly, the trained model is used in practical situations, such as helping musicians detect played notes accurately or transcribing piano music.
Figure 3 represents the architecture of the system.

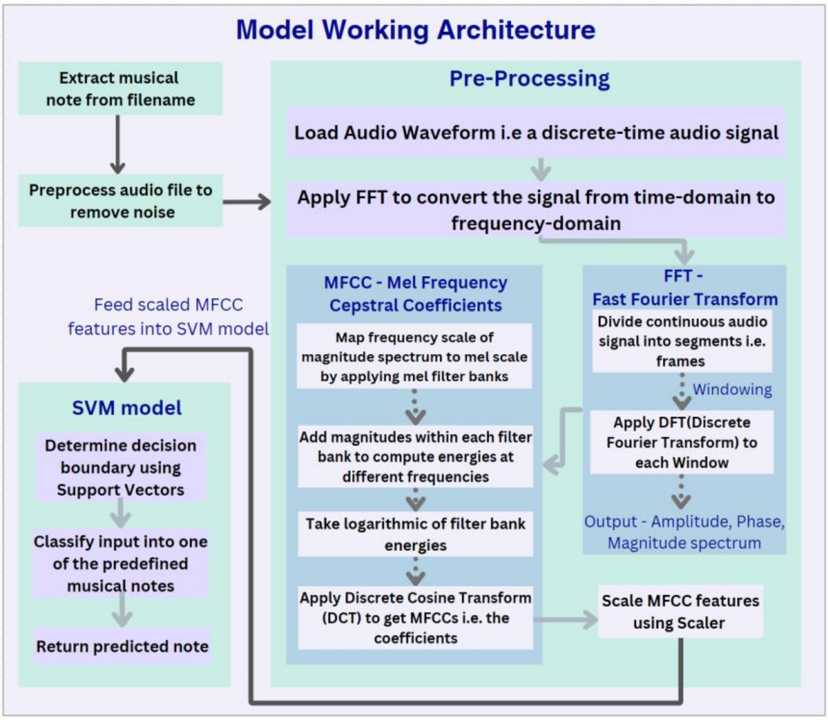


Figure 3. System Architecture

5. Results

The Proposed Music Note Recognition System performs well, with an accuracy rate for testing set is 90% as shown in Table 2. The Classification Report of other metrics such as precision, recall, f1-score and support are depicted in Table 3. Further, macro-average and weighted average are derived to calculate overall system performance. It is observed that the system performed well on recordings of higher quality inputs, but it struggled with deteriorated audio or background noise, which led to decrease in performance. It is observed that the system produces high precision for most of the note classes. F1-score is also good in maximum cases. Overall accuracy is 94% which depicts the quality and robustness of the implemented system. This demonstrates how sensitive the system is to the quality of the input and underscores the need for methods to increase robustness against erratic or poor-quality inputs.

The formulation of evaluation parameters used here are computed as:

$$\text{Macro Average Precision} = \sum_{i=1}^{n} \frac{Precision_i}{n} \quad (3)$$

$$\text{Macro Average Recall} = \sum_{i=1}^{n} \frac{Recall_i}{n} \quad (4)$$

$$\text{Macro Average F1-Score} = \sum_{i=1}^{n} \frac{F1-Score_i}{n} \quad (5)$$

$$\text{Weighted Average Precision} = \frac{\sum_{i=1}^{n} w_i Precision_i}{\sum_{i=1}^{n} w_i} \quad (6)$$

$$\text{Weighted Average Recall} = \frac{\sum_{i=1}^{n} w_i Recall_i}{\sum_{i=1}^{n} w_i} \quad (7)$$

$$\text{Weighted Average F1-Score} = \frac{\sum_{i=1}^{n} w_i F1-Score_i}{\sum_{i=1}^{n} w_i} \quad (8)$$

$$\text{Accuracy} = \frac{C}{N} \quad (9)$$

where,

n is the total number of distinct classes

w_i is the weight for class i

C is the number of correct predictions

N is the total number of instances

Table 2. Accuracy

Metric	Value
Training Accuracy	0.9487
Test Accuracy	0.9032

The possible effects of the system on accessibility, production, and education of music are highlighted by this assessment. However, we noticed some limitations, especially regarding differences in accuracy across music genres and the system's reliance on high-quality input. To solve these challenges, future research should focus on developing algorithms that can handle complicated compositions more adeptly and on improving flexibility by diversifying training datasets. Our approach may be useful in helping musicians in real life. But this approach is suitable for musicians at their learning phase in real life.

Table 3. Classification Report

Notes	Precision	Recall	F1-score	Support
A	1.00	1.00	1.00	8
A#	1.00	0.86	0.92	7
B	1.00	1.00	1.00	7
C	1.00	0.90	0.95	9
C#	0.86	1.00	0.92	7
D	1.00	1.00	1.00	7
D#	1.00	1.00	1.00	7
E	0.88	1.00	0.93	7
F	1.00	0.88	0.93	8
F#	1.00	1.00	1.00	7
G	1.00	1.00	1.00	7
G#	0.71	1.00	0.83	7
Macro Average	0.95	0.96	0.95	88
Weighted Average	0.96	0.94	0.94	88
Accuracy	0.94			

6. Conclusion:

In conclusion, a major development in the field of music is the creation and application of a system for identifying musical notes. Experimental results show machine learning algorithms play a vital role to build systems that can precisely identify musical notes from audio input. The results demonstrate how machine learning can be used to simplify tasks associated with transcription, instruction, and music creation. The system is essential for fostering inclusivity, efficiency, and creativity because of its accurate note detection capabilities, support for performance analysis, and aid in music composition. Highly accurate results are achieved for high quality, less overlapping notes. Also, the system is highly robust to the new input tune over traditionally predefined notation systems.

The Music Note Recognition System's future direction entails a thorough strategy meant to enhance its usefulness, accuracy, and adaptability. One of the main goals is to improve recognition accuracy for intricate arrangements, chords, various playing styles, and instrument tones (Ito & Arai, 2021). Moreover, through the utilization of pitch, harmonics, melodic phrasing, and interval training, we can harness this system effectively for vocal training (Sharma & Er, 2019). To make sure the system can handle a wider variety of musical genres and unusual instruments, this encompasses growing training datasets and researching transfer learning strategies. Just as this system employs the frequency spectrum for note recognition, it holds promise for tuning various instruments such as guitars (Deo Kumar et al., 2021).

Furthermore, efforts will be made to lower computational demands and optimize the system for real-time processing, allowing for smooth integration with music production software and live applications. The system's capabilities will be further enhanced by developments in identifying expressive elements in music and in improving the accessibility and usability of user interface design. Initiatives for standardization and compatibility, cross-domain application research, and partnerships with educational platforms, all demonstrate how the system has potential to have a larger influence and be widely used.

References

1. Bhalke, D. G., et al. Dynamic Time Warping Technique for Musical Instrument Recognition for Isolated Notes. 2011 International Conference on Emerging Trends in Electrical and Computer Technology, 2011, pp. 768–71. IEEE Xplore, https://doi.org/10.1109/ICETECT.2011.5760221.
2. Deo Kumar, Bishnu, et al. Design & Implementation of Digital Guitar Tuner Using MATLAB. 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2021, pp. 547–49. IEEE Xplore, https://doi.org/10.1109/ICACITE51222.2021.9404728.
3. Fu, Zhouyu, et al. A Survey of Audio-Based Music Classification and Annotation. IEEE Transactions on Multimedia, vol. 13, no. 2, Apr. 2011, pp. 303–19. IEEE Xplore, https://doi.org/10.1109/TMM.2010.2098858.
4. Gfeller, Beat, et al. SPICE: Self-Supervised Pitch Estimation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, 2020, pp. 1118–28. https://doi.org/10.1109/TASLP.2020.2982285.
5. Ghosal, Arijit, et al. Music Classification based on MFCC Variants and Amplitude Variation Pattern: A Hierarchical Approach. International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 5, no. 1, Mar. 2012, pp. 131–50. www.earticle.net, https://www.earticle.net/Article/A208805.
6. Guerrero-Turrubiates, Jose de Jesus, et al. Pitch Estimation for Musical Note Recognition Using Artificial Neural Networks. 2014 International Conference on Electronics, Communications and Computers (CONIELECOMP), 2014, pp. 53–58. IEEE Xplore, https://doi.org/10.1109/CONIELECOMP.2014.6808567.
7. Ito, Tsuyoshi, and Shuichi Arai. Harmonic Representation for CNN-LSTM Automatic Chord Recognition. 2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS), 2021, pp. 1–5. IEEE Xplore, https://doi.org/10.1109/ICORIS52787.2021.9649565.
8. Jang, Byeong-Yong, et al. Music Detection from Broadcast Contents Using Convolutional Neural Networks with a Mel Scale Kernel. EURASIP Journal on Audio, Speech, and Music Processing, vol. 2019, no. 1, Dec. 2019, p. 11. https://doi.org/10.1186/s13636-019-0155-y.
9. Molau, S., et al. Computing Mel-Frequency Cepstral Coefficients on the Power Spectrum. 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221), vol. 1, 2001, pp. 73–76 vol.1. IEEE Xplore, https://doi.org/10.1109/ICASSP.2001.940770.
10. Mounir, Mina, et al. Musical Note Onset Detection Based on a Spectral Sparsity Measure. EURASIP Journal on Audio, Speech, and Music Processing, vol. 2021, no. 1, Dec. 2021, p. 30. https://doi.org/10.1186/s13636-021-00214-7.
11. Othman, Kama Azura, and Any Arope Zainal Abidin. Signal Processing Application for Musical Notes Recognition. 2021 IEEE 19th Student Conference on Research and Development (SCOReD), 2021, pp. 135–39. IEEE Xplore, https://doi.org/10.1109/SCOReD53546.2021.9652762.
12. Puri, Surekha B., and Shrinivas P. Mahajan. Optimum Feature Selection for Harmonium Note Identification Using ANN. 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1–7. IEEE Xplore, https://doi.org/10.1109/ICCCNT45670.2019.8944560.
13. Raguraman, Preeth, et al. LibROSA Based Assessment Tool for Music Information Retrieval Systems. 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 2019, pp. 109–14. IEEE Xplore, https://doi.org/10.1109/MIPR.2019.00027.
14. Rajadnya, Vibhavari, and Kalyani Joshi. Raga Classification Based on MFCC and Variants. 2021 IEEE 2nd International Conference on Technology, Engineering, Management for Societal

Impact Using Marketing, Entrepreneurship and Talent (TEMSMET), 2021, pp. 1–6. IEEE Xplore, https://doi.org/10.1109/TEMSMET53515.2021.9768314.

15. Sharma, Shraddha, and Meng Joo Er. Design and Development of an Indian Classical Vocal Training Tool. 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS), 2019, pp. 58–62. IEEE Xplore, https://doi.org/10.1109/ICoIAS.2019.00017.
16. Suguna, G. C., et al. Automatic Detection of Musical Note Using Deep Learning Algorithms. Intelligent Communication Technologies and Virtual Mobile Networks, Springer Nature, 2023, pp. 381–92. https://doi.org/10.1007/978-981-19 1844-5_29.
17. Szczerba, Marek, and Andrzej Czyzewski. Pitch Detection Enhancement Employing Music Prediction. Journal of Intelligent Information Systems, vol. 24, no. 2–3, Mar. 2005, pp. 223–51. https://doi.org/10.1007/s10844-005-0324-6.
18. Varughese, Aaron, and Kalaivani S. Music Note Recognition Based Physical Therapy in Parkinson's Disease. 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), IEEE, 2023, pp. 82–85. https://doi.org/10.1109/IITCEE57236.2023.10090886. 9 Jagtap et al., Proceedings on Engineering Sciences, Vol. xx, No. x (202x) x-x, doi:
19. Wang, Wenjun. Music Chord Sequence Recognition Method Based on Audio Feature Extraction Algorithm. 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT), 2023, pp. 1286–89. IEEE Xplore, https://doi.org/10.1109/ICCECT57938.2023.10140586.
20. Youssef, Khalid, and Peng-Yung Woo. Music Note Recognition Based on Neural Networks. 2008 Fourth International Conference on Natural Computation, vol. 2, 2008, pp. 474–78. IEEE Xplore, https://doi.org/10.1109/ICNC.2008.831.