Comparison of Convolutional Neural Networks and K Nearest Neighbours for Music Instrument Recognition

Dhivya S[1] and Prabu Mohandas[2]

[1] Department of Computer Science and Engineering, Easwari Engineering College,
Chennai, India
dhivyasreedhar@gmail.com
[2] Intelligent Computing Lab, Department of Computer Science and Engineering,
National Institute of Technology, Calicut, India
prabum@nitc.ac.in

Abstract. Music Instrument Recognition is one of the main tasks of Music Information Retrieval. Identification of instruments present in an audio track provides information about the composition of music. Music instrument recognition in polyphonic music is a challenging task. Existing approaches use temporal, spectral, and perceptual feature extraction techniques to perform Music Instrument Recognition. In the proposed work a convolutional neural network and k nearest neighbour classifier framework are implemented to identify the musical instrument present in a monophonic audio file and the performance of the two models are compared. The model is trained on London Philharmonic dataset which consists of 6 different classes of musical instruments. Mel spectrogram representation is used to extract features for the Convolutional Neural Network model. For k-nearest neighbors, the Mel-frequency cepstral coefficients feature vectors are calculated to perform classification. This approach only works for monophonic music and cannot be used for polyphonic music. The model helps to label the unlabelled audio files so that manual annotation can be avoided. The model performed well with excellent result of 99.17% accuracy for the Convolutional Neural Network and 97% accuracy for the k-nearest neighbor architecture.

Keywords: Musical instrument recognition · convolutional neural network · k nearest neighbours · deep learning · multi-class classification

1 Introduction

Music is one of the most popular forms of art that is practiced and listened to by billions of people all over the world. Music can improve mood, decrease pain and anxiety, and can benefit our physical and mental health in numerous ways. Musical instrument recognition is the task of instrument identification by virtue of its audio [2]. Automatic recognition of musical instruments forms the basis of more complex tasks like melody extraction, music information retrieval, recognizing the dominant instruments from polyphonic audio [1], and so on. The task of efficient automatic music classification is of vital importance and forms the basis for various advanced applications of AI in the musical domain like music genre classification, automatic music transcription and recommender systems.

Music Instrument Recognition enhances the performance of other MIR tasks. It helps to find the type of musical instrument used which would significantly improve the performance of other MIR tasks like automatic music transcription, music genre identification and source separation. It will be very helpful for the people who are working on music data and also the present-day music companies, it can assist them on music recommendations for their users. Music Information Retrieval (MIR) is about retrieving information from music. MIR systems add significant value to existing music libraries and make them more easily accessible. They help in automatic music classification, indexing, searching and organisation [3].

Machine learning helps systems to learn from data, identify patterns and make decisions with minimal human interaction. Machine learning for audio signal processing has attracted a large amount of attention recently for its uses in speech recognition. Machine Learning techniques provide numerous ways to perform music categorization as per need. Music instrument classification can be done easily on monophonic sounds than polyphonic ones, where multiple instruments played together. Classifying instances into three or more classes is called multi-class classification. A multi-class classifier is implemented which takes an audio stream as the input and outputs the class of the musical instrument present in the stream. Most work is done on monophonic music which is less challenging. The timbre of the instruments are studied which in turn gives patterns for classification. The methods for Music Instrument Recognition can be classified as traditional Machine Learning techniques and deep learning techniques. Deep learning techniques for Music Instrument Recognition have been evolving rapidly in the last decade.

The primary goal is to classify 6 instruments from the given music data. A convolutional neural network (CNN) and a κ - Nearest neighbour (KNN) classifier is implemented to perform the classification. In the Convolutional Neural Network Classifier, the input audio stream is pre-processed to extract the mel-spectrogram. The features for the mel-spectrogram are used to perform the classification. The input of the model is the mel-spectrogram, and the output is an index corresponding to the predicted class. For the κ nearest neighbour classifier, the mfcc feature vectors are calculated, the number of neighbours is set and the classification process is done.

1.1 Motivation

Music Instrument Recognition can help in finding what kinds of instruments are present in a music clip and can distinguish the instruments with one another. The motivation behind this work is to come up with a system that can help musicians extract a particular instrument sound. It can help people who are working on music in music data transcription and identification. It can help present day music companies with recommendations for their users. It allows us to perform various music information retrieval tasks like pitch, timbre separation, genre classification, automatic music transcription and source signal separation. It assists people involved in musicology, psycho acoustics, signal processing and optical music recognition,

1.2 Objective

- Development of a model to train different audio files, the model should classify what instruments are used in the audio.
- A method to label unlabelled audio files to avoid manual annotation.
- Training of CNN and KNN models to perform Music Instrument Recognition.
- Performance analysis of both the models to get better understanding

1.3 Organization

The proposed work, analyzes the performance of CNN and k-nearest neighbour classfier. The entire chapter is organised as follows. Section 2 reviews the most popular existing works. The proposed methodology is explained in detail in section 3. The experimental setup of the work is explained in Section 4. Section 5 discusses the results of the experiments. Finally, the chapter concludes with section 6.

2 Literature Review

Musical Instrument Recognition using CNN and SVM [4], in this chapter the classification task was performed on the IRMAS dataset [5]. The IRMAS dataset consists of musical audio excerpts with annotations of predominant instruments present in the file. Music and Instrument Classification using Deep Learning Technics [6], implemented a multi-class classifier that identifies instruments in music streams. They use Google's AudioSet which provides human labelled data. It has a set of 10 second clips from YouTube, labelled with the audio instruments and any other sound label it contains. Musical Instrument Classification Using Neural Networks [3] implemented an automatic classification of musical instrument sounds with a dataset of 4548 files from 19 instruments of MIS database - The University of Iowa Musical Instrument Samples [7]. In Deep convolutional neural networks for predominant instrument recognition in polyphonic music [8], Music Instrument Classification in Polyphonic music is accomplished. They also used the IRMAS dataset. AN Artificial Neural Network is implemented for classification in [9]. They use the full London philharmonic orchestra dataset which contains twenty classes of instruments belonging to the four families - woodwinds, brass, percussion, and strings. Kratimenos et al. [16] explored a variety of data augmentation techniques focusing on different sonic aspects, such as overlaying audio segments of the same genre, as well as pitch and tempo-based synchronization. Eronen et al. [17] set up a system for pitch independent musical instrument recognition. A wide set of features covering both spectral and temporal properties of sounds was investigated, and their extraction algorithms were designed. Patil S.R. [18] has described a system for musical instrument recognition in monophonic audio signals where the single sound source is active at a time using a Gaussian mixture model (GMM). Ghosh et al. [19] proposed a Decision Tree based model for automatic recognition of musical instruments.

Singh et al. [4] used a combination of Convolutional Neural Network and Support Vector Machine. The SVM uses MFCC for feature extraction. The audio excerpts used for training will be pre-processed into images (visual representation of frequencies in sound). The results obtained from both the CNN and SVM are added to get the weighted average, which gave better performance in terms of instrument identification. Lara Haidar-Ahmad [6] implemented a model which consists of a CNN which takes input as an audio stream that is pre-processed to extract the mel-spectrogram, and outputs the class of pre-selected instruments. They focus on 3 instruments, and classify audio streams into one of 4 classes: “Piano”, “Drums”, “Flute” or “Other”, around 8,000 samples were trained. Lara Haider-Ahmed [6] obtained a precision of 70%, a recall of 65%, and a F1-score of 64%. In [3], probabilistic neural networks were used for classification for its flexibility and the straightforward design. The dataset used consists of 4548 tunes from 19 instruments of the MIS database. Probabilistic neural networks were used as classifiers. Mel-frequency cepstral coefficients(mfcc) were used as features. Multi-level quantization was applied to the features before doing the classification. The accuracy of 92% was achieved. Kratimenos et al. [16] utilized Convolutional Neural Networks for the classification task, comparing shallow to deep network architectures and an ensemble of VGG-like classifiers, achieving slightly above 80% in terms of label ranking average precision (LRAP) in the IRMAS test set. Eronen et al. [17] validated the usefulness of the features test data that consisted of 1498 samples covering the full pitch ranges of 30 orchestral instruments from the string, brass and woodwind families, played with different techniques. The correct instrument family was recognized with 94% accuracy and individual instruments in 80% of cases. Patil S.R. [18] obtained an accuracy of 93.18% (average) for a combination of MFCC as a feature and GMM as a classifier. Ghosh et al. [19] obtained an accuracy of 84.02% by Decision Tree (DT) for a set of 9 instruments belonging to different families. The accuracy for predicting the instrument family 96.07% for string family and for wind instrument the overall prediction accuracy is 90.78%.

Han et al. [8] uses a convolution neural network for the predominant instrument Recognition. The model is trained on the single labelled predominant instrument. They used dataset of 10k audio files. It consisted of 11 instruments. Convolutional neural networks were found to be more robust than conventional methods and thus obtained an F1 measure of 0.602 for micro achieving 19.6% performance improvement compared with other algorithms. Mahanta et al. [9] achieved an accuracy of 97% on the full dataset containing all 20 classes of different musical instruments. Table 1 shows the comparison of performance for different models implemented for Music Instrument Recognition.

Table 1. Comparison of models

Authors And Year	Model	Objective	Dataset	Accuracy/F1 score
Hing, Dominick Sovana, and Connor Settle[10] 2020	CNN	A multiclass instrument classifier using CNN	6705 training samples and 1400 test samples from IRMAS [5]	70.3 %
Yun, Mingqing, and Jing Bi[11] 2018	LSTM	A music instrument classifier using RNN with log mel-spectrogram	A dataset of 14 instruments with 200 training samples	81 %
J. Liu and L. Xie[12] 2010	SVM	SVM based classifier of musical instruments using MFCC features	2177 clips of 13 Chinese instruments and 13 western instruments	95.44%
S.Prabavathy, V.Rathikarani, Dhanalakshmi,[13] 2020	KNN	Proposed a KNN model for Music Instrument Classification	1284 samples were used from 16 musical instruments	98.22%
Anhari, Amir Kenarsari[14] 2020	RNN	Multi-instrument classifier using an attention based Bi-directional LSTM	20k audio clips from the OpenMic dataset	F1 score of 0.83
Kingkor Mahanta, Saranga, Abdullah Faiz, and Partha Pakray [9], 2021	ANN	An ANN model was trained to perform classification on 20 classes of musical instruments	13679 examples divided among 20 classes of musical instruments	99.7%

2.1 Performance Issues

Mahanta et al. [9] proposed a deep artificial neural network model that efficiently distinguishes and recognizes 20 different classes of musical instruments, even across instruments belonging to the same family. The model trains on the full London philharmonic orchestra dataset which contains twenty classes of instruments belonging to the four families viz. woodwinds, brass, percussion, and strings. They use use only the mel-frequency cepstral coefficients (MFCCs) of the audio data.

The dataset was divided into training and validation or testing sets in the ratio 8:2 using stratified splitting, such that the number of examples from each of the 20 classes split proportionally into two sets. The training and test sets contained 10,943 training examples and 2,736 test examples respectively after the split. MFCC features are extracted, from the constant length examples and feeding them into an ANN model to make predictions. The model uses an ANN architecture with 1690 input neurons which are connected to the first dense hidden layer having 512 neurons followed by ReLU activation function. The second and third hidden layers contain 1024 and 512 neurons respectively both followed by the ReLU activation function. A dropout layer with a rate of 0.3 is then added to induce regularization and avoid overfitting. After the dropout layer, the values pass through two more hidden layers containing 128 and 64 neurons respectively and a dropout layer with a 0.2 rate. The final output layer has 20 neurons for each class. They use the Rectified Linear Unit (ReLU) activation function for all the hidden layers. It simply activates the neurons containing a positive value after the aforementioned computations.

$$y = max(0, x) \tag{1}$$

The softmax function is used in the output layer. It provides the confidence scores of each class using

$$\mathit{\Sigma}(z_i) = e^{z_i} / \mathit{\Sigma}_{j=1}^{K} e^{z_j} \tag{2}$$

The scores add up to 1. The class having the highest confidence score is the model's predicted class for a particular set of input features. The model achieved an accuracy of 97%. During model training, the training accuracy peaked 0.9913 and validation accuracy 0.9726. The dataset is quite imbalanced as most instruments belong to a particular family, so data augmentation measures may be adopted to deal with the imbalance problem. Different learning rates and optimizers can be tried to produce different results. Expanding the target space by supporting the recognition of even more instruments including the piano, or the ukulele would be a notable improvement.

2.2 Problem Statement

• MFCCs and Mel-spectrograms provide excellent visual perceptions of sound, thus CNNs may prove to be quite efficient than ANN.
• The dataset may be imbalanced and most of the instruments belonged to one particular class of the family.
• Other optimizers and activation functions can give better results.
• Lots of variables in the pre-processing stage can be tweaked to provide better results

3 Proposed Methodology

The identification of instruments present in an audio track plays a vital role in music information retrieval as it provides information about the composition of music. Music instrument recognition in polyphonic music is a challenging task. The proposed work employs a CNN and k-nearest neighbour classifier to identify the musical instrument present in a monophonic audio file. This section gives a detailed description of the proposed methodology.

3.1 Proposed Block Diagram

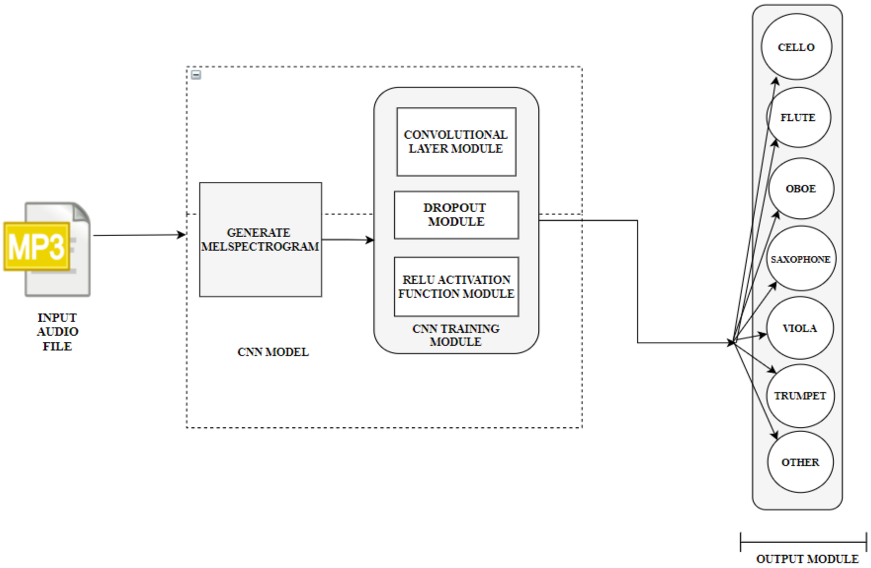


Fig. 1. Block diagram for CNN model

Figure 1 and Figure 2 depicts the block diagram of the proposed work. The input audio file is loaded to the processing module and the output is the class of the musical instrument it belongs to. In the CNN model the audio file is converted to a Mel-spectrogram and the extracted features are sent to the CNN training module. Inside the training module it first passes through the Convolutional layer gets convoluted, then through the dropout layer and then the Relu activation function. In the KNN model the audio file is resampled, the audio features are calculated. The audio files are normalized and the mfcc feature vectors are calculated using the librosa module and are inputted into the KNN Classifier for classification.

3.2 CNN based approach

Figure 3 depicts the CNN model architecture. It consists of three convolutional layers followed by a pooling layer, an activation function and a fully connected layer. The CNN model takes an image as the input. The audio files undergo some transformations so that they can be inputted as an image in the CNN model. In deep learning, a convolutional neural network (CNN/ConvNet) is a class of deep neural networks, which is most commonly used on images. It is composed of

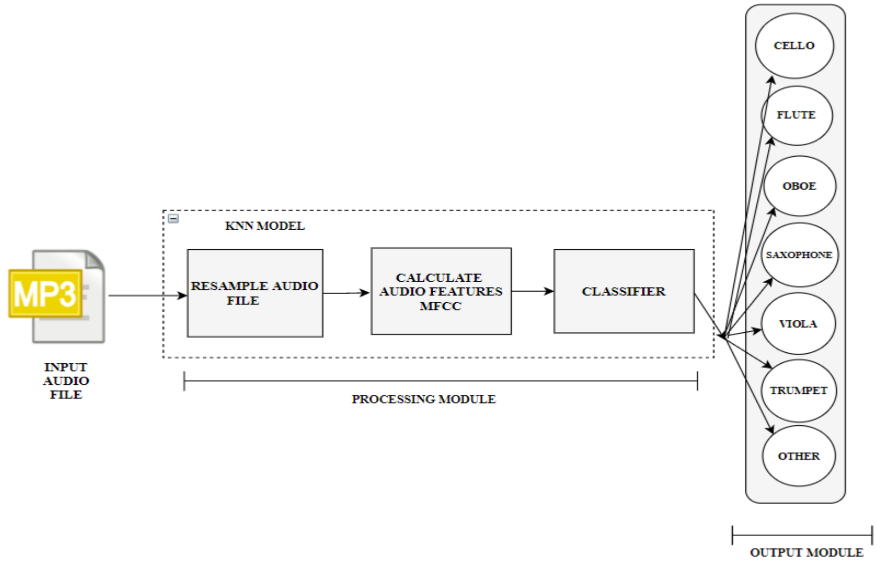


Fig. 2. Block diagram for KNN model

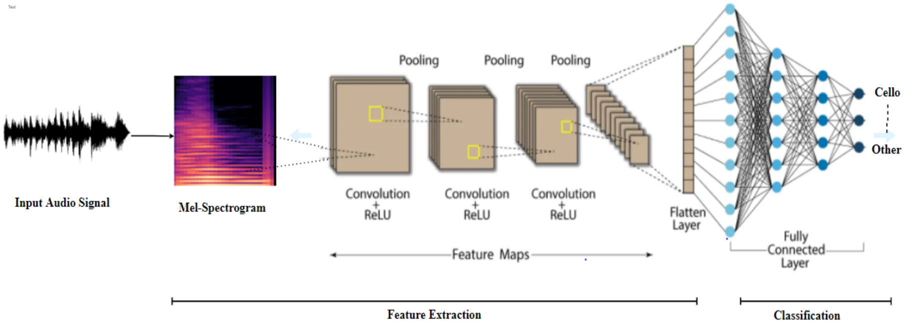


Fig. 3. CNN Model Architecture

many layers of neurons. The first layer extracts basic features such as horizontal or diagonal edges which is passed on to the next layer. The next layer then detects more complex features like corners or combinational edges. It identifies even more complex features as we move deep into the network. CNN is patterned to process multidimensional array data in which the convolutional layer takes a stack of feature maps, like the pixels of those colour channels, and convolves each feature map with a set of learnable filters to obtain a new stack of output feature maps as input. Based on the activation map of the final convolution layer, the classification layer outputs a set of numerical values between 0 and 1 that predicts which class the image belongs to. Figure 4 represents the two dimensional representation of an audio file.

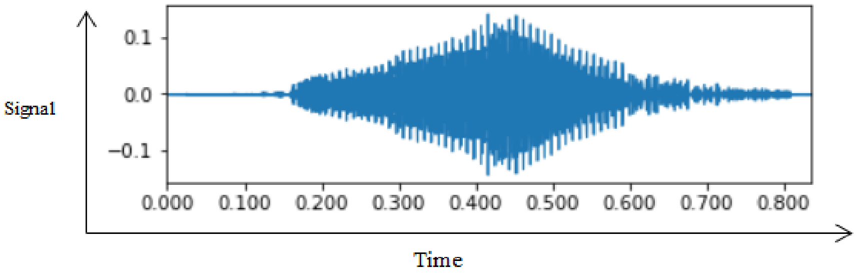


Fig. 4. Two Dimensional representation of an audio file

The Mel Scale is the logarithmic transformation of the frequency of a given signal. It is difficult for humans to differentiate higher frequencies than lower frequencies. Even if the distances between the differences of the two sounds are same, the human perception of the difference is not same. Hence, Mel Scale is fundamental in Machine Learning applications of audio.

Transformation from Hertz scale to Mel scale:

$$m = 1127 * log(1 + f/700) \tag{3}$$

Equation 3 is a formula to transform Hertz scale to Mel scale from O'Shaughnessy's book. The mel frequency cepstral coefficients (MFCCs) of a signal is used to describe the overall shape of a spectral envelope.

Mel spectrogram is a spectrogram that is converted to a Mel scale. A spectrogram is a visualization of the frequency spectrum of a signal, where the frequency spectrum of a signal is the frequency range that is contained by the signal. Each audio file in the dataset is converted into a spectrogram to perform the classification. Figure 5 depicts the mel-spectrogram generated for each class of musical instrument present in the dataset.

In a CNN, the input of a shape (number of inputs) x (input height) x (input width) x (input channels) becomes a feature map of shape (number of inputs) x (feature map height) x (feature map width) x (feature map channels), after passing through a convolutional layer.

Convolutional layer generally has the following attributes:

- The number of filters the convolutional layers will learn from.
- The dimensions of the kernel,The size of the input
- The activation function to be applied after performing convolution

The model uses 32 filters, has a kernel size of 3*3 and uses relu activation function.

The pooling layer is responsible for reducing the spatial size of the convolved feature. The pooling layer resizes the input spatially, using the MAX operation. The MaxPool operation down samples the input along its dimensions by taking the maximum value over an input window which is defined by the pool_size for each channel of the input. The model uses a pool_size of 3*3. Fully connected layers are responsible for connecting all neurons in one layer to neurons of another layer.

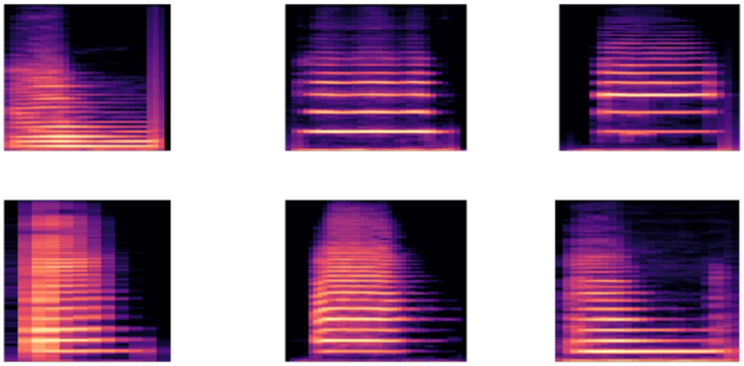


Fig. 5. Mel spectrograms of each musical instrument - cello, flute, oboe, saxophone, trumpet, viola

After uploading and reprocessing all the audio files, the labels of each sample are appended. The dataset is split into training, testing and validation sets. The input convolutional layer followed by the second and the third convolutional layer are initialized. After the image is passed into the input convolutional layer it gets convoluted to a different size. The feature maps passes through the pooling layer which reduces the size of the convolved feature. Finally the output layer is initialized and the model is compiled. The dense layer or the fully connected layer connects every other neuron and all the extracted feature maps together. The model is trained for the given number of epochs.

The input is the mp3 audio file and the output is the class of the musical instrument in the monophonic audio file. The musical instrument classes are initialized. All the audio files are loaded and the labels of each file are initialized. The dataset is split into training set, validation set and test set. The CNN model is initialized and compiled. The model is trained for the specified number of epochs.

3.3 KNN based approach

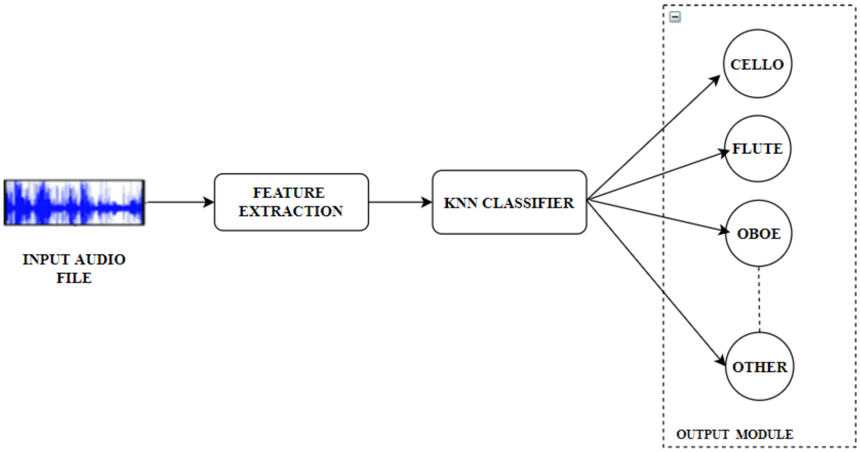


Fig. 6. KNN Model Architecture

K-Nearest Neighbors (KNN) is one of the simplest algorithms used for both classification and regression problems. Classification is done by a majority vote to its neighbours. Figure 6 depicts the KNN model architecture. The feature extraction process is done from the input audio file and the features are sent to the KNN classifier. All the audio files are normalised and the mfcc features are calculated for each audio clip using the librosa module. In the KNN classifier, the value of κ is initialized to the selected number of neighbours. The distance between the feature vectors of each pair of the audio clip is calculated and sorted. For κ entries from the sorted data, the mode of κ labels will be returned for classification problems. All the audio samples are loaded, pre-processed and their respective labels are appended. The value of κ is initialized and the Euclidean distances between the κ number of nearest neighbours are calculated. The distances of the inputs are sorted. For the κ nearest neighbours, simple majority is applied. The process is first performed for $\kappa = 1$, after finding the best value of κ from the error vs κ value graph the process is repeated for that value of κ.
The input is the mp3 audio file and the output is the class of the musical instrument in the monophonic audio file. The musical instrument classes are initialized. All the audio files are loaded and the labels of each file are initialized. All the labels are encoded to numerical values to normalize the labels. The dataset is split into training set and test set. The KNN model is initialized and compiled. The best value of κ is found out for the model based on the error vs κ value graph and the model is compiled again for that value of κ.

4 Experimental setup and analysis

4.1 Dataset and Annotations

The dataset consists of musical instrument samples from the Philharmonic website [15]. It is a balanced dataset and it consists of 6 different classes. The dataset consists of 600 files. The classes are: ‘cello’,‘flute’,‘oboe’ ,‘saxophone’ ,‘trumpet’ ,‘viola’. Each class consists of 100 recordings of each instrument. All audio files are in .mp3 format. The size of the dataset is 8.16 MB. Data set is divided into testing and training set. We pre-process the data before using it. To process we use sample rate 44100 Hz, an fft size of 2048, hop length of 512. The dataset includes musical audio excerpts with annotations of the musical instrument present.

4.2 Model training and testing

The given data set [15] is split into training, validation and testing set. The train set has 60% of the data and the test set and validation set has 20% each. The model is trained on the training set. The CNN model requires as an image as the input. The audio files have to be visualized using some transformations. The audio is pre-processed to extract the Mel-spectrograms. Mel-spectrogram is used as input of the model. The mel-spectrograms of all the audio files are stored separately. These files are then trained in the CNN model. Three convolutional layers which consist of 32, 64 and 128 filters, respectively are used to produce feature maps. RELU activation function was implemented after each convolutional layer. Three max pooling layers are used to reduce dimensionality without padding. A dropout rate of 0.25 was applied to reduce overfitting. ADAM optimizer with a 0.0001 learning rate was used and the CNN model was trained up to 30 epochs. Categorical cross-entropy was used as a loss function to optimize results. After the model is trained the accuracy and loss of the model is calculated. For the KNN algorithm, the dataset is loaded and the features and feature vectors are calculated. The features are scaled using StandardScaler function. The data is then split into train (75%) and test data (25%). The KNN Classifier is first trained for $\kappa = 1$ and the performance is evaluated. The best value of κ is calculated from the Error rate vs κ value plot and the model is trained for that value of κ. The performance is analysed and the confusion matrix is plotted.

5 Results and Discussion

5.1 Performance Evaluation of CNN model

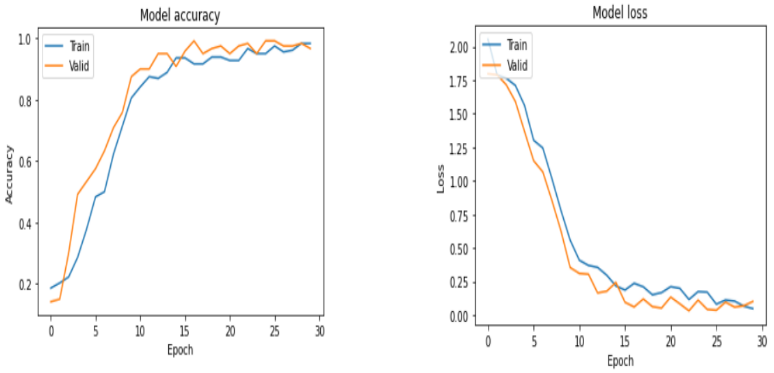
The proposed work is evaluated using different parameters. The training data of the philharmonic dataset has 360 audio samples and the validation and testing data have 120 each. We calculate the loss, accuracy, val_loss and val_accuracy for the CNN model as shown in Table 2. The loss function keeps decreasing with every epoch and the accuracy keeps increasing. The training set gave an accuracy

Table 2. Performance Evaluation of CNN model

Epoch	loss	accuracy	val_loss	val_accuracy
1	2.2948	0.1787	1.7985	0.1417
2	1.7934	0.1892	1.7900	0.1500
3	1.7808	0.2036	1.7166	0.3000
4	1.7390	0.2792	1.5918	0.4917
5	1.5956	0.3752	1.3686	0.5333
6	1.3288	0.4498	1.1500	0.5750
7	1.2465	0.4992	1.0673	0.6333
8	1.0706	0.6139	0.8537	0.7083
9	0.7975	0.7186	0.6244	0.7583
10	0.5712	0.7933	0.3536	0.8750
11	0.4287	0.8451	0.3099	0.9000
12	0.3388	0.8715	0.3042	0.9000
13	0.3661	0.8762	0.1645	0.9500
14	0.2531	0.9180	0.1773	0.9500
15	0.1930	0.9466	0.2428	0.9083
16	0.2002	0.9299	0.0947	0.9583
17	0.2264	0.9185	0.0600	0.9917
18	0.1820	0.9325	0.1194	0.9500
19	0.1623	0.9268	0.0621	0.9667
20	0.1998	0.9333	0.0518	0.9750
21	0.2379	0.9225	0.1336	0.9500
22	0.2067	0.9251	0.0831	0.9750
23	0.1224	0.9632	0.0312	0.9833
24	0.1788	0.9502	0.1110	0.9500
25	0.1674	0.9580	0.0418	0.9917
26	0.0936	0.9716	0.0357	0.9917
27	0.1142	0.9529	0.0956	0.9750
28	0.1053	0.9649	0.0589	0.9750
29	0.0628	0.9862	0.0687	0.9833
30	0.0637	0.9763	0.1009	0.9667

of 97% at the end of the 30th epoch. From the plot of accuracy in Figure 7a it can be seen that the model has not over-learned the training dataset, showing comparable skill on both the training and validation datasets. From the plot of loss in Figure 7b, it can be seen that the model has comparable performance on both training and validation datasets.

The test dataset gave an accuracy of 99.1% and loss value of 0.24.



(a) Model Acuracy Plot

(b) Model Loss Plot

Fig. 7. Performance plots for CNN

5.2 Performance Evaluation of KNN model

The KNN model is evaluated using different metrics. Precision, f1-score, recall , accuracy and support are calculated. Table 3 shows the the classification report for $\kappa = 1$. The Error vs κ value plot is plotted to find the best value of κ so that the model is not overfitted.

From the plot in Figure 8, it can be seen that the least stable error rate occurs around $\kappa = 7$ hence $\kappa = 7$ gives the best model. The classification report for $\kappa = 7$ is shown in Table 4 . Table 5 shows the comparison of f1 score, accuracy, recall, precision and the number of wrong predictions for 150 samples.

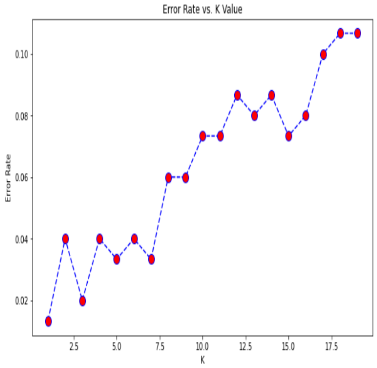


Fig. 8. Error vs κ value plot for KNN

Table 3. Classification report for $\kappa = 1$

Index	Precision	Recall	F1 score	support
0	1.00	0.96	0.98	25
1	1.00	1.00	1.00	25
2	1.00	0.96	0.98	25
3	1.00	1.00	1.00	25
4	0.96	1.00	0.98	25
5	0.96	1.00	0.98	25

Table 4. Classification report for $\kappa = 7$

Index	Precision	Recall	F1 score	support
0	0.96	1.00	0.98	25
1	0.96	0.96	0.96	25
2	1.00	0.96	0.98	25
3	1.00	0.96	0.98	25
4	0.93	1.00	0.96	25
5	0.96	0.92	0.94	25

Table 5. Comparison of results for $\kappa = 1$ and $\kappa = 7$

K value	Accuracy	Recall	Precision	F1 score	No. of samples	Wrong predictions
1	0.99	0.99	0.99	0.99	150	2
7	0.97	0.97	0.97	0.97	150	5

A confusion matrix is a table that is often used to describe the performance of a classification model (or ”classifier”) on a set of test data for which the true values are known. Figure 9 and Figure 10 show the confusion matrix for $\kappa = 1$ and $\kappa = 7$ respectively. Table 6 shows the comparison of accuracy for the CNN and KNN models for the 150 test audio samples.

Table 6. Comparison of results

Model	accuracy	Number of samples
KNN (κ=1)	0.99	150
KNN($\kappa = 7$)	0.97	150
CNN	0.9917	120

The CNN algorithm gave an accuracy of 99.17% on 120 test samples while the KNN algorithm ($\kappa = 7$) gave an accuracy of 97% on 150 test samples. Both the algorithms performed well for the unknown test samples.

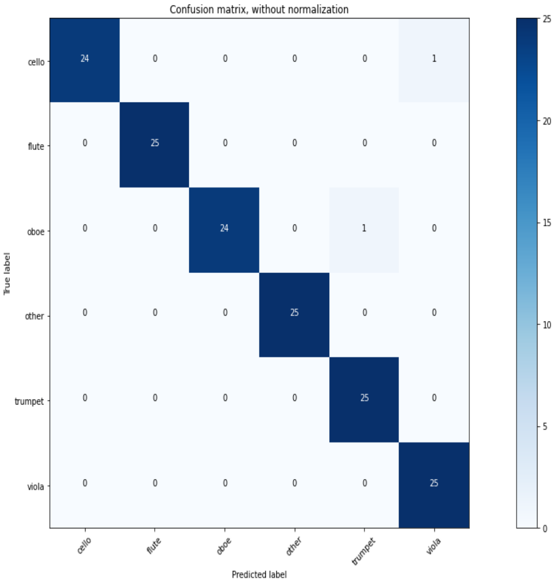


Fig. 9. Confusion matrix for $\kappa = 1$

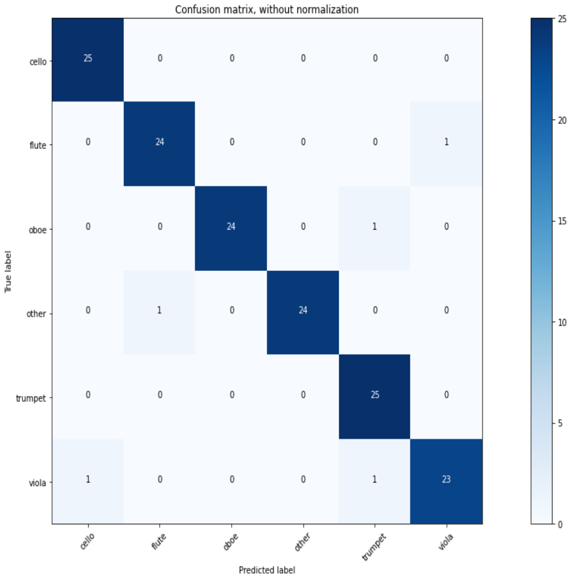


Fig. 10. Confusion matrix for $\kappa = 7$

6 Conclusion

After all the explanatory analysis of the result given, it is clear that both the models provided a satisfactory result. Both classification models performed with high accuracy. The performance of both the models have been analysed carefully. The mel-spectrogram representation of music provided sufficient features and information for the convolutional neural network fit to, and allowed the model to very accurately differentiate between musical instruments with very different timbres. After the 30 epochs, the research found the excellent result with 99.17% accuracy for the 120 samples used in the CNN model. The KNN model showed 97% accuracy for $\kappa = 7$, for the 150 test samples.

References

1. Essid, S., Richard, G., David, B,(2005), "Instrument recognition in polyphonic music based on automatic taxonomies," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, no. 1, pp. 68–80.
2. Pham, J., Woodford, T. and Lam, J.(2009), Classification of Musical Instruments by Sound.
3. Mustafa Sarimollaoglu, Coskun Bayrak, (2006)," Musical Instrument Classification Using Neural Networks,", Proceedings of the 5th WSEAS International Conference on Signal Processing, Istanbul, Turkey, pp. 151-154.
4. Prabhjyot Singh, Dnyaneshwar Bachhav, Omkar Joshi, Nita Patil,(2019),"Musical Instrument Recognition using CNN and SVM," in International Research Journal of Engineering and Technology (IRJET),vol. 06, no.3, pp. 1487-1491.
5. IRMAS Dataset, https://www.upf.edu/web/mtg/irmas . Last accessed 12 Sept 2021
6. Haidar-Ahmad, Lara.(2019),"Music and instrument classification using deep learning technics", Recall, 67(37.00), pp.80-00.
7. IOWA Dataset, http://theremin.music.uiowa.edu/MIS.html. Last accessed 12 Sept 2021
8. Y. Han, J. Kim and K. Lee,(2017), "Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 1, pp. 208-221.
9. Mahanta, S.K., Khilji, A.F.U.R. and Pakray, P.,(2021)," Deep Neural Network for Musical Instrument Recognition Using MFCCs," in Journal Computación y Sistemas, vol 25, no 2, pp. 351–360.
10. Hing, D.S. and Settle, C.J.(2020), Detecting and Classifying Musical Instruments with Convolutional Neural Networks.
11. Yun, M. and Bi, J.(2018), Deep Learning for Musical Instrument Recognition.
12. Liu, J., Xie, L., (2010),"SVM -based automatic classification of musical instruments," in International Conference on Intelligent Computation Technology and Automation, vol. 3, pp. 669-673.
13. Prabavathy, S., Rathikarani, V.,Dhanalakshmi, P., "Classification of Musical Instruments using SVM and KNN," in International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, no. 7,pp.2278-3075.
14. Anhari, A.K.,(2020)," Learning multi-instrument classification with partial labels,", arXiv preprint arXiv:2001.08864.
15. Philharmoic Dataset, https://philharmonia.co.uk/. Last accessed 12 Sept 2021

16. A. Kratimenos, K. Avramidis, C. Garoufis, A. Zlatintsi, P. Maragos, (2020) ,"Augmentation Methods on Monophonic Audio for Instrument Classification in Polyphonic Music," in 28th European Signal Processing Conference (EUSIPCO), pp. 156-160.
17. A. Eronen and A. Klapuri,(2000), "Musical instrument recognition using cepstral coefficients and temporal features," in IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings, vol.2, pp. II753-II756.
18. Patil S.R., Machale S.J.,(2020),"Indian Musical Instrument Recognition Using Gaussian Mixture Model," in Techno-societal 2018, Springer, Cham, (pp. 51-57).
19. A. Ghosh, A. Pal, D. Sil, S. Palit,(2018), "Music Instrument Identification Based on a 2-D Representation," in International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), pp. 509-513.