

Implementing Musical Instrument Recognition using CNN and SVM

Prabhjyot Singh¹, Dnyaneshwar Bachhav², Omkar Joshi³, Nita Patil⁴

^{1,2,3}Student, Computer Department, Datta Meghe College of Engineering, Maharashtra, India

⁴Professor, Computer Department, Datta Meghe College of Engineering, Maharashtra, India

Abstract - Recognizing instruments in music tracks is a crucial problem. It helps in search indexing as it will be faster to tag music and also for knowing instruments used in music tracks can be very helpful towards genre detection (e.g., an electric guitar heavily indicates that a track is not classical). This is a classification problem which will be taken care of by using a CNN (Convolutional Neural Network) and SVM (Support Vector Machines). The audio excerpts used for training will be preprocessed into images (visual representation of frequencies in sound). CNN's have been used to achieve high accuracy in image processing tasks. The goal is to achieve this high accuracy and apply it to music successfully. This will make the organization of abundant digital music that is produced nowadays easier and efficient and helpful in the growing field of Music Information Retrieval (MIR). Using both CNN and SVM to recognize the instrument and taking their weighted averages will result in higher accuracy.

Key Words: CNN, SVM, MFCC, Kernel, Dataset, Features

1. INTRODUCTION

Music is made by using various instruments and vocals from a human in most cases. These instruments are hard to classify as many of them have a very minor difference in their sound signature. It is hard even for humans some time to differentiate between similar sounding instruments. Instrument sounds played by different artists are also different as they have their own style and character and also depends on the quality of the instrument. Recognizing which instrument is playing in a music will help to make recommending songs to a user more accurate as by knowing persons listening habit we can check if they prefer a particular instrument and make better suggestions to them. These suggestions could be tailored to a persons liking. Searching for music will also be benefited as for the longest time only way songs or music has been searched is by typing name of a song or an instrument, this can be changed as we can add more filters to the search and fine tune the parameters based on instruments. A person searching for an artist can easily filter songs based on instruments playing in the background. It can also help to check which instruments are more popular and where this will help to cater more precisely to a specific portion of the populous. Using CNNs to classify images has been extensively researched and has produced highly accurate results. Using these detailed results and applying them to classify instruments is our goal. Ample research has also been carried out to classify instruments without the use of CNNs. Combining the results from CNN classification with classification by SVM to get better results is the eventual goal. The weighted averages of the predictions will be used. SVM have been used for classifications for a long time with great results and to apply these to instrument recognition in music is the key idea which may help to achieve satisfactory results. Every audio file is converted to an image format by feature extraction using LibROSA python package, melspectrogram and mfcc are to be used.

2. LITERATURE SURVEY

Yoonchang Han et al [1] present a convolutional neural network framework for predominant instrument recognition in real-world polyphonic music. It includes features like MFCCs, Histogram, Facial expression, Finger features like Minutiae. The various classifiers used in this are SVM, KNN, NN and Softmax. The IRMAS dataset is used here for training and testing purposes. Evaluation parameters used here and their corresponding values are Precision=0.7 and recall=0.25, F1 measure=0.25.

Emmanouil Benetos et al. [2] have proposed automatic musical instrument identification using a variety of classifiers is addressed. Experiments are performed on a large set of recordings that stem from 20 instrument classes. Several features from general audio data classification applications as well as MPEG-7 descriptors are measured for 1000 recordings. Here, the feature used is Branch-and-bound. The first classifier is based on non-negative matrix factorization (NMF) techniques. A novel NMF testing method is proposed. A 3-layered multilayer perceptron, normalized Gaussian radial basis function networks and support vector machines (SVM) employing a polynomial kernel are also used as classifiers for testing.

Athanasia Zlatintsi et al. [3] proposed new algorithms and features based on multiscale fractal exponents, which are validated by static and dynamic classification algorithms and then compare their descriptiveness with a standard feature set of MFCCs which efficient in musical instrument recognition. Markov model is used for experimental results. The experiments are carried

out using 1331 notes from 7 different instruments. The first set of experiments used Gaussian Mixture Models (GMMs) up to 3 mixtures. The second set of experiments used Hidden Markov Models (HMMs). For the modeling of structure of the instruments' tone, a left-right topology is used. The combination of the proposed features with the MFCCs gives slightly better results than the MFCCs alone for most cases.

Arie A. Livshin et al. [4] reveals that in every solo instrument recognition using a custom dataset of 108 different solos performed by different artists have been used. Features used were, Temporal Features, Energy Features, Spectral Features, Harmonic Features, Perceptual Features, initially 62 features of given types were used but later reduced to 20. Classification is performed by first Linear Discriminant Analysis (LDA) on learning set and then multiplied the coefficient matrix with test set and classified using the K nearest neighbour (KNN). Best K is found using leave-one-out. In order to find the best feature Gradual Descriptor Elimination has been used. It repeatedly uses LDA to eliminate the least significant descriptor.

Lin Zhang et al. [5] proposed the system which is made up of two parts, the peripheral auditory system, and the auditory central nervous system. The entire system comprises 27 parallel channels. In the posteroventral cochlear nucleus (PVCN) model. Self Organizing Map Network (SOMN) is used here as a classifier. A large solo database that consists of 243 acoustic and synthetic solo tones over the full pitch ranges of seven different instruments. Sampling frequency used is 44.1 KHz. The entire system comprises 27 parallel channels. In the posteroventral cochlear nucleus (PVCN) model. Self Organizing Map Network (SOMN) is used here as a classifier.

Manuel Sabin et al. [6] has proposed a methodology in which project learns and trains using a guitar, piano, and mandolin by a specifically tuned neural network. Each of the recordings were exported as a '.wav' file. The final topology includes the bias nodes along the top, in which the actual version has 5,001 nodes including the bias node in the input layer, 115 including the bias node in the hidden layer, and 3 in the output layer. For this problem, the learning rate is set significantly low - i.e. 0.005. Momentum is an optimization meant to prevent very slow convergence and usually the percentage of the previous jump that the momentum takes is large, as it is in our program at 0.95.

Juan J. Bosch et al. [7] address the identification of predominant music instruments in polytimbral audio by previously dividing the original signal into several streams. Here, Fuhrmann algorithm, LRMS and FASST algorithm are used for methodology. In this proposed method, SVM is used as a classifier. Dataset included here is proposed by Fuhrmann. Evaluation parameters used here and their corresponding values are Precision=0.708 and recall=0.258 and F1 measure=0.378.

Toni Heittola et al. [8] also proposes a novel approach to musical instrument recognition in polyphonic audio signals by using a source-filter model and an augmented non-negative matrix factorization algorithm for sound separation. In the recognition, Mel-frequency cepstral coefficients are used to represent the coarse shape of the power spectrum of sound sources and Gaussian mixture models are used to model instrument-conditional densities of the extracted features. The method is evaluated with polyphonic signals, randomly generated from 19 instrument classes. GMM is used as a classifier here. Nineteen instrument classes are selected for the evaluations of the following instruments are accordion, bassoon, clarinet, contrabass, electric bass, electric guitar, electric piano, flute, guitar, harmonica, horn, oboe, piano piccolo, recorder, saxophone, trombone, trumpet, tuba. The instrument instances are randomized into training (70%) or testing (30%) set. This method gives good results when classifying into 19 instrument classes and with the high polyphony signals, implying a robust separation even with more complex signals.

Philippe Hamel et al. [9] proposed a model able to determine which classes of musical instrument are present in a given musical audio example, without access to any information other than the audio itself. A Support Vector Machine (SVM), a Multilayer Perceptron (MLP) and a Deep Belief Network (DBN) classifiers are used here. To test for generalization instruments are divided into three independent sets: 50% of the instruments were placed in a training set, 20% in a validation set and the remaining 30% in a test set. MFCCs are used for feature extraction. It also involve the use of spectral feature: centroid, spread, skewness, kurtosis, decrease, slope, flux and roll-off. To compare three different models F-score is used as a performance measure. The F-Score is a measure that balances precision and recall. DBN performed better than other 2 model.

3. METHODOLOGY

The model proposed in this project is a combination of Convolutional Neural Network and Support Vector Machine. The model consists of convolutional layers shown below, followed by one fully connected layer and one SoftMax layer and SVM. It has been shown that the deeper the convolutional layer is, the more abstract the features it learns [15]. MFCC is used for feature

extraction for SVM. The results obtained from both the CNN and SVM are added to get the weighted average, which will give the better performance in terms of instrument identification.

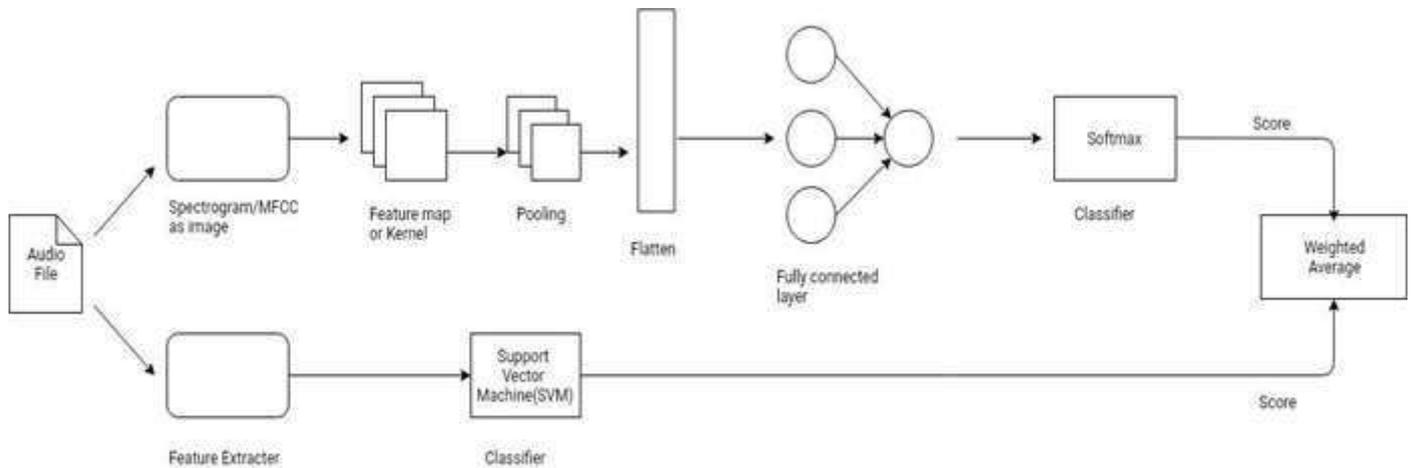


Fig-1: System Architecture

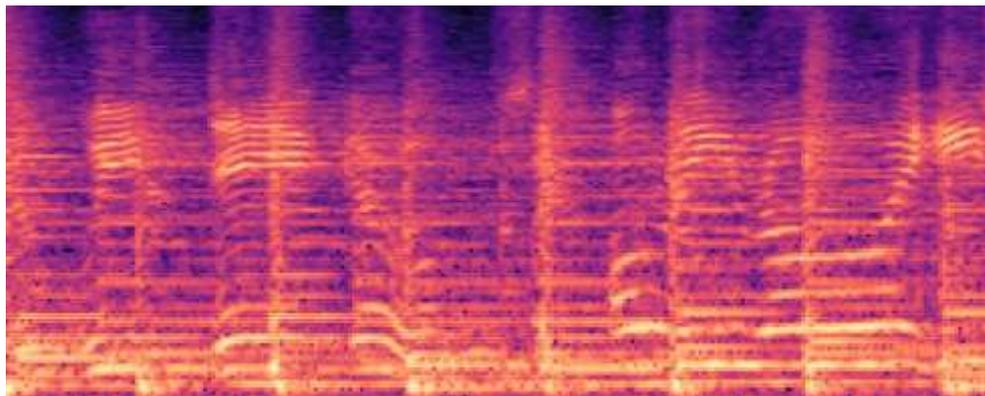


Fig.3 Sample Spectrogram

3.1 CNN

Convolutional Neural Networks are similar to Neural Networks as they have weights which change overtime to improve learning. The big difference is that CNNs are 3-D as they are images and hence are different from ANNs which work on 2-D vectors. CNNs are pretty straightforward to understand if previous knowledge of ANN is present. CNNs use the kernel also known as feature map or a filter. This filter is slided over the input image and as it moves block by block, we take the cross product of the filter with the image. Filter moves over the entire image and this process is repeated. As we can see in the there is not just a single filter but lots of them depending on the application or model. Here we show a single set of convolution layer i.e. bunch of filters but there are usually multiple convolution layers in between. And as we progress through multiple layers the output of the previous layers become the input for the next layer, and we again perform cross product to get the finer details and helps the model learn more in-depth. It takes a lot of computing power to handle this much data as images in real world have lots of details and have higher size hence increasing the size of each filter and computation required.

To overcome this, we pool these filters. Pooling is basically reducing the parameters of each filter. This is also called as down-sampling or sub-sampling. It works on each filter individually. *Reducing* the size improves the performance of the network. The most used technique for pooling is Max Pooling. In Max Pooling we basically again move a window over the filter but instead of taking a cross product, just select the maximum value present. It also has other advantages such as making influence of orientation change and scaling of image lesser. Orientation and scaling can make a lot of difference if not taken into consideration.

Flattening is the intermediate between convolution layers and fully connected layers. As we have 3-D data after pooling we convert it to 1-D vector for the fully connected layer. This process is done by flatten.

Fully connected layer is the one in which the actual classification happens. We extract features from the image in the Convolution part. This part is like a simple ANN which takes the features from the image as an input and adjust the weights based on back propagation.

CNNs have been researched a lot in the past few years with amazing results. \$ Karen Simonyan and Andrew Zisserman of the University of Oxford created a 19-layer CNN model more commonly known as VGG Net. It uses 3x3 filters, pad of 1 and 2X2 max pooling layers, with stride 2.

Google Net was one of the first CNN architecture which didn't just stack convolution and pooling layers on top of each other. It had an error rate of only 6.67%. They did not just add more layers to the model as it increases the computation power.

3.2 SVM

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. SVM is strong classification algorithm because it is simple in structure and it requires a smaller number of features. SVM is currently considered the most efficient family of algorithm in machine learning because it is computationally efficient and robust in high dimension. During the training of SVM, feature extractor converts each input value to feature set and these feature sets capture basic information about each input.

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N—the number of features) that distinctly classifies the data points.

Kernel: In machine learning, kernel methods are a class of algorithms for pattern analysis, whose best-known member is the support vector machine (SVM). The general task of pattern analysis is to find and study general types of relations (for example clusters, rankings, principal components, correlations, classifications) in datasets. In its simplest form, the kernel trick means transforming data into another dimension that has a clear dividing margin between classes of data. For many algorithms that solve these tasks, the data in raw representation have to be explicitly transformed into feature vector representations via a user-specified feature map: in contrast, kernel methods require only a user-specified kernel, i.e., a similarity function over pairs of data points in raw representation.

Radial Basis Function Kernel: In machine learning, the radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification.

Parameters of SVM:

C: Parameter C balances the trade-off between the model complexity and empirical error. Since the empirical error is estimated from the samples in the training set, therefore it is related to the training error. When C is large, the SVM tends to be overfitting; and when C is small, the SVM tends to be underfitting. We have set the c value 110.

Gamma: The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. We have set the gamma value to 0.1.

4. FEATURE SELECTION

Identifying the components of the audio signal is important. So, feature extraction is used to identify the basic important data of given input and it also removes the unnecessary data which comes in the format of emotion, noise, etc.

Spectrogram is used for CNN and MFCC, used for SVM, is a feature used in automatic speech and speaker recognition. Standard 32ms windows is going to use for the extraction of features.

In this section our goal is to achieve a smaller feature set which will give us the quick result i.e. recognizing the input in real-time. It will also not compromise the recognition rate.

5. DATASET

IRMAS dataset is divided into Training and Testing data. Audio files are sampled at 44.1 kHz in 16 bit stereo wav format. Total of 9,579 audio files are present. The instruments present for recognition are acoustic guitar (gac), electric guitar (gel), organ (org), piano (pia) and human singing voice (voi).

The dataset is very well constructed by taking into consideration change of styles in the past decades, also different ways in which people play the instrument and the production values associated. In the training dataset 3 seconds excerpts are present out of 2000 distinct recordings.

The training set contains 6705 audio files and 2874 audio files are kept for testing. The length of the sound excerpt for testing is between 5-20 seconds.

6. RESULT & ANALYSIS:

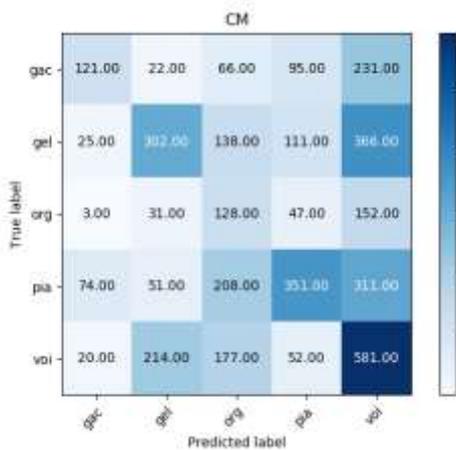


Fig.4 CNN Confusion Matrix

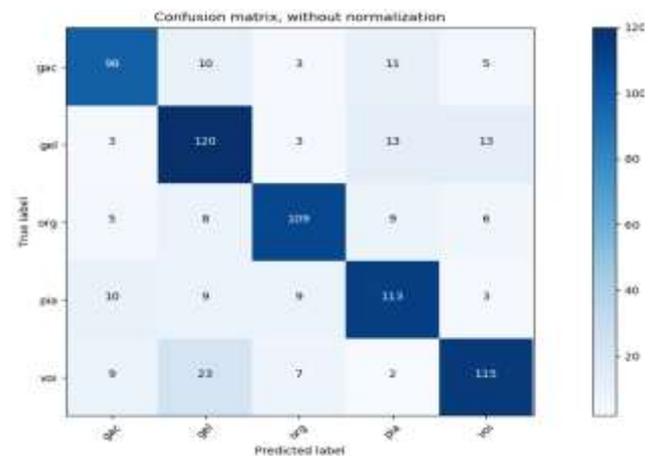


Fig. 5 SVM Confusion Matrix

Result for CNN:

	Precision	recall	f1-score	support
<i>gac</i>	0.50	0.23	0.31	535
<i>gel</i>	0.49	0.32	0.39	942
<i>org</i>	0.18	0.35	0.24	361
<i>pia</i>	0.54	0.35	0.43	995
<i>voi</i>	0.35	0.56	0.43	1044

Result for SVM:

	precision	recall	f1-score	support
<i>gac</i>	0.78	0.77	0.78	127
<i>gel</i>	0.79	0.79	0.75	152
<i>org</i>	0.83	0.08	0.81	137
<i>pia</i>	0.78	0.78	0.77	144
<i>voi</i>	0.74	0.74	0.77	156

From the results shown before we can see that in our case the SVM is giving better results than the CNN. This is contrary to fact that CNN should have been more accurate. This is due to the limitations put on the implementation of the project because of the

lack of a high-performance GPU available in the system for running the CNN. We were not able to get high number of epochs to run as it caused overheating of the mobile GPU in the laptop.

Also, from the results of the CNN we can see that the main problem encountered by the model was to distinguish between instruments of the same type. In our dataset we had guitar and electric guitar which are of the same type and piano and organ which are very similar. Human voice was almost always classified easily as it is very different from the instruments.

Another limitation was that in test data many audio files had human singing in them. As dataset consists of audio files from many years, they have voice in many of them. As these files are from songs there is lot of singing in them. This resulted in noisy data which explains why many times files were wrongly classified as human voice.

A great insight observed from the result is that pianos were classified were very well. This is great because there were two string-based instruments and piano being a percussion-based instrument which is not that far from the string category wasn't affected much. During testing we saw that rap songs were mostly classified as human voice and then the instrument playing in the background in most of the cases.

7. CONCLUSIONS

In musical instrument recognition, we try to identify the instrument playing in a given audio excerpt. Automatic sound source recognition is an important task in developing numerous applications, including database retrieval and automatic indexing of instruments in music.

We saw that the SVM was performed better than and expected and the CNN did not due the hardware limitations and this one weak link that we would like to improve in the future. Another thing is to add more instruments into the four that we have now.

The precision we get for CNN is 0.50 for acoustic guitar, 0.49 for electric guitar, 0.18 for organ, 0.54 for piano, 0.35 for voice. The precision we get for SVM is 0.78 for acoustic guitar, 0.79 for electric guitar 0.78, 0.83 for organ, 0.78 for piano, 0.74 for voice.

The easiest class was human voice, piano, electric guitar, guitar, organ in the given order. This was due to the fact that distinguishing humans singing from instruments is easier. This gives a great opportunity to make this a multi-label classification in the future.

We would like to improve our system as it has not reached its full potential due hardware limitations. It took a lot of time to run the CNN model. Also, as instruments such as guitar, electric guitar and piano and organ are similar in nature, it was hard for models to distinguish them. SVM model performed very well for the data.

In the future we like to implement and add the following features:

- Support for more instruments.
- More accurate results.
- Ability for users to save their history.
- To implement a system for users to add more data to the dataset with their submissions.

REFERENCES

- [1] Yoonchang Han, Jaehun Kim, and Kyogu Lee "Deep convolutional neural networks for predominant instrument recognition in polyphonic music", Journal Of Latex Class Files, Vol. 14, No. 8, May 2016
- [2] Emmanouil Benetos, Margarita Kotti, and Constantine Kotropoulos, "Large Scale Musical Instrument Identification", Department of Informatics, Aristotle Univ. of Thessaloniki Box 451, Thessaloniki 541 24, Greece
- [3] Athanasia Zlatintsi and Petros Maragos, "Musical Instruments Signal Analysis And Recognition Using Fractal Features", School of Electr. and Comp. Engin., National Technical University of Athens, 19th European Signal Processing Conference (EUSIPCO 2011)

- [4] Arie Livshin, Xavier Rodet, "Instrument Recognition Beyond Separate Notes -Indexing Continuous Recordings", ICMC 2004, Nov 2004, Miami, United States.
- [5] Lin Zhang, Shan Wang, Lianming Wang, "Musical Instrument Recognition Based on the Bionic Auditory Model", 2013 International Conference on Information Science and Cloud Computing Companion
- [6] Manuel Sabin, "Musical Instrument Recognition With Neural Networks", Computer Science 180 Dr. Scott Gordon 24 April 2013
- [7] Juan J. Bosch, Jordi Janer, Ferdinand Fuhrmann and Perfecto Herrera, "A comparison of sound segregation technique for predominant instrument recognition in musical audio signals", Universitat Pompeu Fabra, Music Technology Group, Roc Boronat 138, Barcelona, 13th International Society for Music Information Retrieval Conference (ISMIR 2012)
- [8] Toni Heittola, Anssi Klapuri and Tuomas Virtanen, "Musical Instrument Recognition in Polyphonic Audio Using source filter model for sound separation", Department of Signal Processing, Tampere University of Technology, 2009 International Society for Music Information Retrieval.
- [9] Philippe Hamel, Sean Wood and Douglas Eck, "Automatic Identification of instrument classes in Polyphonic and Poly-Instrument Audio", Departement d'informatique et de recherche opérationnelle, Université de Montréal, 2009 International Society for Music Information Retrieval.