Page 1

## PES UNIVERSITY

**(Established under Karnataka Act No. 16 of 2013)**

**100-ft Ring Road, Bengaluru – 560 085, Karnataka, India**

***Audio Technology Report***

***on***

**MUSIC INSTRUMENT CLASSIFICATION USING SPECTROGRAMS**

***Submitted by***

**ACHYUTH S.S(PES1UG21EC010)**

**AKSHAY ANGADI(PES1UG21EC030)**

## ANANYA BELAVADI(PES1UG21EC034)

## June - July 2023

***under the guidance of***

**Dr. Ashwini**

**Associate Professor**

**Department of ECE**

**PES University**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**PROGRAM B. TECH**

Page 2



**CERTIFICATE**

*This is to certify that the Report entitled*

**MUSIC INSTRUMENT CLASSIFICATION USING SPECTROGRAMS**

*is a bona fide work carried out by*

**ACHYUTH S.S(PES1UG21EC010)**

**AKSHAY ANGADI(PES1UG21EC030)**

## ANANYA BELAVADI S(PES1UG21EC035)

In partial fulfilment for the completion of course work in the Program of Study B. Tech in Electronics and Communication Engineering, under rules and regulations of PES University, Bengaluru during the period June - July 2023. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The report has been approved as it satisfies the academic requirements in respect of Capstone project work.

*Signature with date & Seal Signature with date & Seal*

*(Dr. Ashwini) Dr. Anuradha M*

*Guide Chairperson*

Name and signature of the examiners:

1.

2.

Page 3

DECLARATION

We, ACHYUTH S.S,AKSHAY ANGADI**,**ANANYA BELAVADI, hereby declare that the report entitled “MUSIC INSTRUMENT CLASSIFICATION BASED ON SPECTOGRAMS” is an original work done under the guidance of Dr. Ashwini Assoc Professor in the Dept of ECE and is being submitted as a partial requirement for completion of Audio Technology Project Work of the B.Tech ECE Program of study during June-July 2023.

Place: BANGALORE

Date :22-07-2023

|  |  |
| --- | --- |
| Name of the student | Signatures |
| 1. ACHYUTH S.S 2. AKSHAY ANGADI 3. ANANYA BELAVADI |  |

Page 4

Index

1. Introduction
2. Literature Survey
3. AST Architecture
4. ViT Model
5. Results and Comparison
6. References

Page 1

**Audio spectrogram transformers Architecture**

INTRODUCTION

Music Instrument Classification is an exciting task in the field of audio processing and machine learning, where the goal is to build a system capable of automatically identifying different musical instruments from audio recordings. One effective approach for this task involves using spectrograms, which provide a visual representation of the audio signal's frequency content over time. Another crucial component is the use of the AST (Attention, Spectrogram, and Temporal) model, which is a neural network architecture designed specifically for audio-related tasks.

The AST model combines the power of attention mechanisms, spectrogram analysis, and temporal processing to capture intricate patterns and temporal dependencies in audio data. The attention mechanism allows the model to focus on essential regions in the spectrogram, highlighting critical features for each instrument. The spectrogram analysis enables the model to extract valuable information related to the timbre and harmonic characteristics of different instruments, while the temporal processing ensures the model considers the dynamics and temporal patterns of musical notes.

Page 2

LITERATURE SURVEY

1.Convolutional Recurrent Neural Networks for Music Classification by Keunwoo Choi et al. (ISMIR 2017) This paper introduces CRNN, a model that combines convolutional and recurrent neural networks for music classification tasks, including music instrument classification. It uses spectrogram representations and explores the temporal patterns in audio data.

2.Convolutional Neural Networks for Music Instrument Recognition by Han-Ting Chiang et al. (ISMIR 2014) This work explores the use of convolutional neural networks (CNNs) for music instrument recognition. The authors experiment with different CNN architectures and spectrogram representations to achieve high accuracy.

3.Automatic Musical Instrument Recognition in Polyphonic Audio by Juan P. Bello et al. (IEEE Transactions on Audio, Speech, and Language Processing, 2005) While this paper is not focused on the AST model, it provides insights into using time-frequency representations (such as spectrograms) and machine learning techniques for automatic musical instrument recognition

Page 3

METHODOLOGY

Methodology of Music Instrument Classification using Spectrogram on AST Model:

1. Data Collection and Preprocessing:

- Gather a diverse dataset of audio recordings containing various musical instruments playing different notes and phrases.

- Preprocess the audio data by normalizing, resampling, and removing background noise to improve data quality.

2. Spectrogram Generation:

- Convert the pre-processed audio data into spectrogram representations using the Short-Time Fourier Transform (STFT) or other time-frequency analysis techniques.

- Optionally, apply windowing functions to improve the spectral resolution of the spectrograms.

3. Data Augmentation (Optional):

- Augment the spectrogram data with transformations like time-shifting, pitch-shifting, or adding background noise. Data augmentation increases dataset diversity and helps the model generalize better.

4. Train / Validation / Test Split (Cross-Validation):

- Split the dataset into training, validation, and testing sets.

Page 4

FLOW CHART

+-----------------------------------------------------------+

| Audio Input Data |

+-----------------------------------------------------------+

|

v

+---------------------------------------+

| Audio Preprocessing |

+---------------------------------------+

|

v

+----------------------------------------------+

| Compute Spectrogram (STFT) |

+----------------------------------------------+

|

v

+---------------------------------------+

| Data Augmentation (Optional) |

+---------------------------------------+

|

v

+----------------------------------------------------+

| Train / Validation / Test Split (Cross-Validation) |

+----------------------------------------------------+

|

v

+--------------------------------------------+

| AST Model |

+--------------------------------------------+

|

v

+--------------------------------------------+

| Pretraining (Optional) |

+--------------------------------------------+

|

v

+------------------------------------------------+

| Model Interpretation (Optional) |

+------------------------------------------------+

|

v

+-----------------------------------+

| Deployed Model |

+-----------------------------------+

|

v

+----------------------------------+

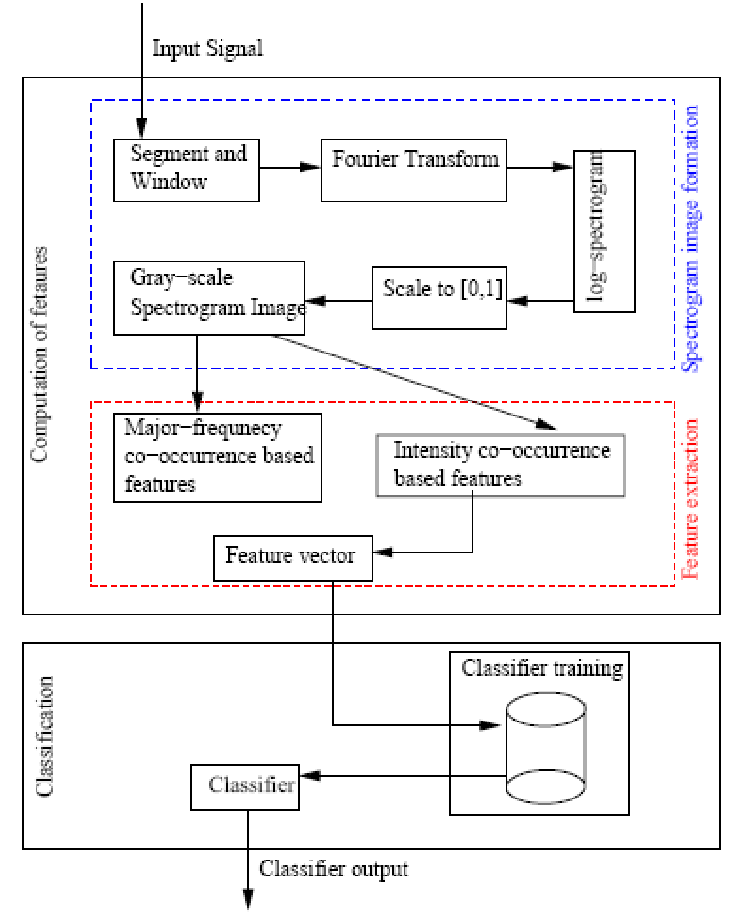
| Music Instrument |

| Classification Results |

+----------------------------------+

Page 5

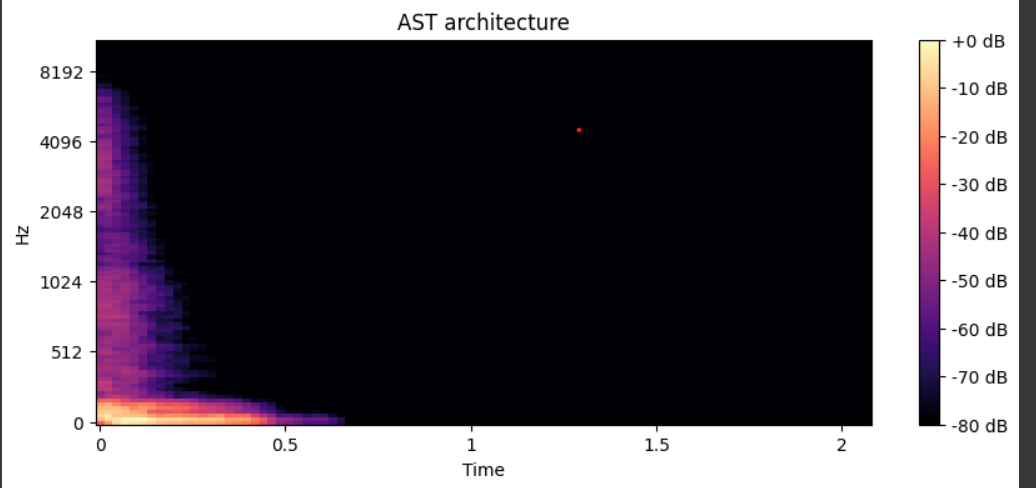
BLOCK DIAGRAM

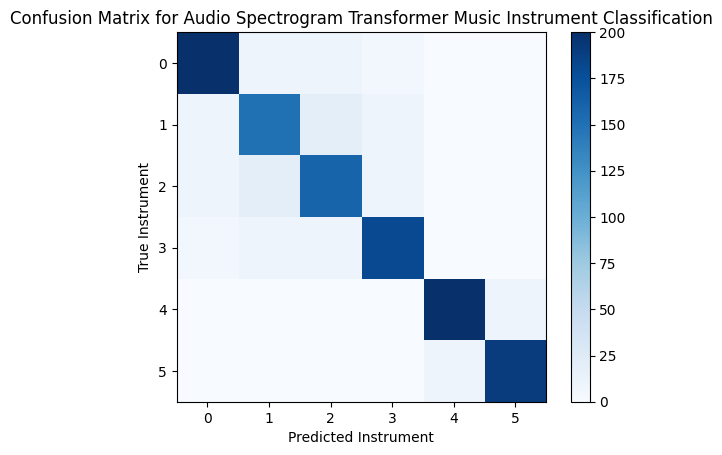


Page 6

RESULTS

AN example of spectrogram generated using AST model for an audio file



Page 7

**VISION TRANSFORMERS MODEL**

INTRODUCTION

* Music instrument classification is a fascinating task in the field of audio processing and machine learning. The goal is to develop an automated system that can accurately identify different musical instruments from audio recordings. One of the most common approaches for this task is to use the spectrogram, a visual representation of the audio signal's frequency content over time. The spectrogram provides valuable information about the timbre and harmonic characteristics of each instrument, making it a suitable input for machine learning models.
* In recent years, Vision Transformer (ViT) models have gained popularity in computer vision tasks due to their impressive performance in image recognition and object detection. The ViT model utilizes the transformer architecture, originally designed for natural language processing, to process image patches and learn global contextual information from the entire image.

Page 8

Literature Survey

* literature survey on music instrument classification using deep learning methods, including some relevant works that use spectrograms as input.
* "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification" by Justin Salamon et al. (2015) This paper focuses on environmental sound classification but provides a foundation for using convolutional neural networks (CNNs) on audio spectrograms. It explores data augmentation techniques and demonstrates the effectiveness of CNNs in handling audio data.
* "Automatic Musical Instrument Recognition: A Multilayer Perceptron Approach to Feature Extraction" by Min-Ki Kim and Kyogu Lee (2004) Although this work predates Vision Transformers, it provides insights into using feature extraction methods for musical instrument recognition. The authors use a multilayer perceptron (MLP) on Mel-frequency cepstral coefficients (MFCCs) to classify musical instruments.
* "Large-scale Musically-Informed Dataset for Automatic Music Performance Analysis" by Colin Raffel et al. (2016) This paper introduces the MusicNet dataset, which includes both audio recordings and corresponding sheet music. While not directly using Vision Transformers, the dataset and techniques for music performance analysis may be useful for related research.
* "Transformers for Audio: The quest for a universal music representation" by Yuntao Bai et al. (2020) This work explores the application of transformers in various audio tasks, including music classification. Though it doesn't specifically focus on Vision Transformers, it discusses the potential of transformer models for music-related tasks.
* Vision Transformer for Image Classification with PyTorch" (PyTorch Tutorial) This tutorial demonstrates how to implement a Vision Transformer for image classification using PyTorch. While not specifically for music instrument classification, it serves as a helpful guide for adapting ViT to other image-like data types, such as spectrograms.

Page 9

Methodology

Brief Explaination

Input: Audio clip

Mel Spectrogram Extraction:

\* Convert the audio clip to a time-domain signal.

\* Apply a short-time Fourier transform (STFT) to the time-domain signal.

\* Compute the mel spectrogram from the STFT coefficients.

Vision Transformer Model:

\* The mel spectrogram is fed into a vision transformer model.

\* The vision transformer model learns to extract the features of the mel spectrogram that are relevant for music instrument classification.

Classification:

\* The output of the vision transformer model is passed through a classifier.

\* The classifier predicts the dominant instruments present in the audio clip.

Mel spectrogram extraction: The mel spectrogram is a representation of the frequency spectrum of sound that is more perceptually relevant than the raw frequency spectrum. This is because the mel spectrogram takes into account the human ear's sensitivity to different frequencies.

Vision transformer model: A vision transformer is a type of deep learning model that is used for image recognition. Vision transformers have been shown to be very effective for a variety of tasks, including music instrument classification.

Classification: The classifier is a machine learning model that is used to predict the class label of an input. In the case of music instrument classification, the classifier predicts the dominant instruments present in the audio clip.

Page 10

Methodology of Music Instrument Classification using Spectrogram and Vision Transformer Model:

1. Data Collection and Preprocessing:

Gather a diverse dataset of audio recordings containing various musical instruments. The dataset should cover different playing styles, pitches, and recording conditions.

Preprocess the audio data to create spectrogram representations. Divide the audio signals into short overlapping segments and apply a Fourier transform to each segment to obtain the magnitude of the frequency-domain representation. This results in a 2D image-like representation, where time is on the x-axis, and frequency is on the y-axis.

1. Data Augmentation (Optional):

To increase the dataset's diversity and improve the model's generalization, apply data augmentation techniques to the spectrogram images. Common augmentations include random shifts, flips, and changes in pitch or time stretch.

1. Split the Dataset:

Divide the dataset into training, validation, and testing sets. The training set is used to train the model, the validation set is used to tune hyperparameters and monitor performance, and the testing set is used to evaluate the final model.

1. Vision Transformer Model Architecture:

Adapt the Vision Transformer (ViT) architecture for music instrument classification. The ViT model processes 2D image-like data through self-attention layers, capturing global contextual information effectively. Modify the input patch size and model dimensions to accommodate the spectrogram images. Consider the number of instruments you want to classify and adjust the output layer accordingly.

Page 11

FLOW CHART

+---------------------------------------------------------+

| Audio Input Data |

+---------------------------------------------------------+

|

v

+--------------------------------------+

| Audio Preprocessing |

+--------------------------------------+

|

v

+--------------------------------------+

| Convert to Spectrogram |

+--------------------------------------+

|

v

+------------------------------------------+

| Data Augmentation (Optional) |

+------------------------------------------+

|

v

+--------------------------------------------------------------+

| Train / Validation / Test Split (Cross-Validation) |

+--------------------------------------------------------------+

|

+--------------------------------------------+

| Vision Transformer Model |

+--------------------------------------------+

|

v

+------------------------------------------------------+

| Model Evaluation and Hyperparameter Tuning |

+------------------------------------------------------+

|

v

+--------------------------------------------------+

| Deployed Model |

+--------------------------------------------------+

|

v

+---------------------------------+

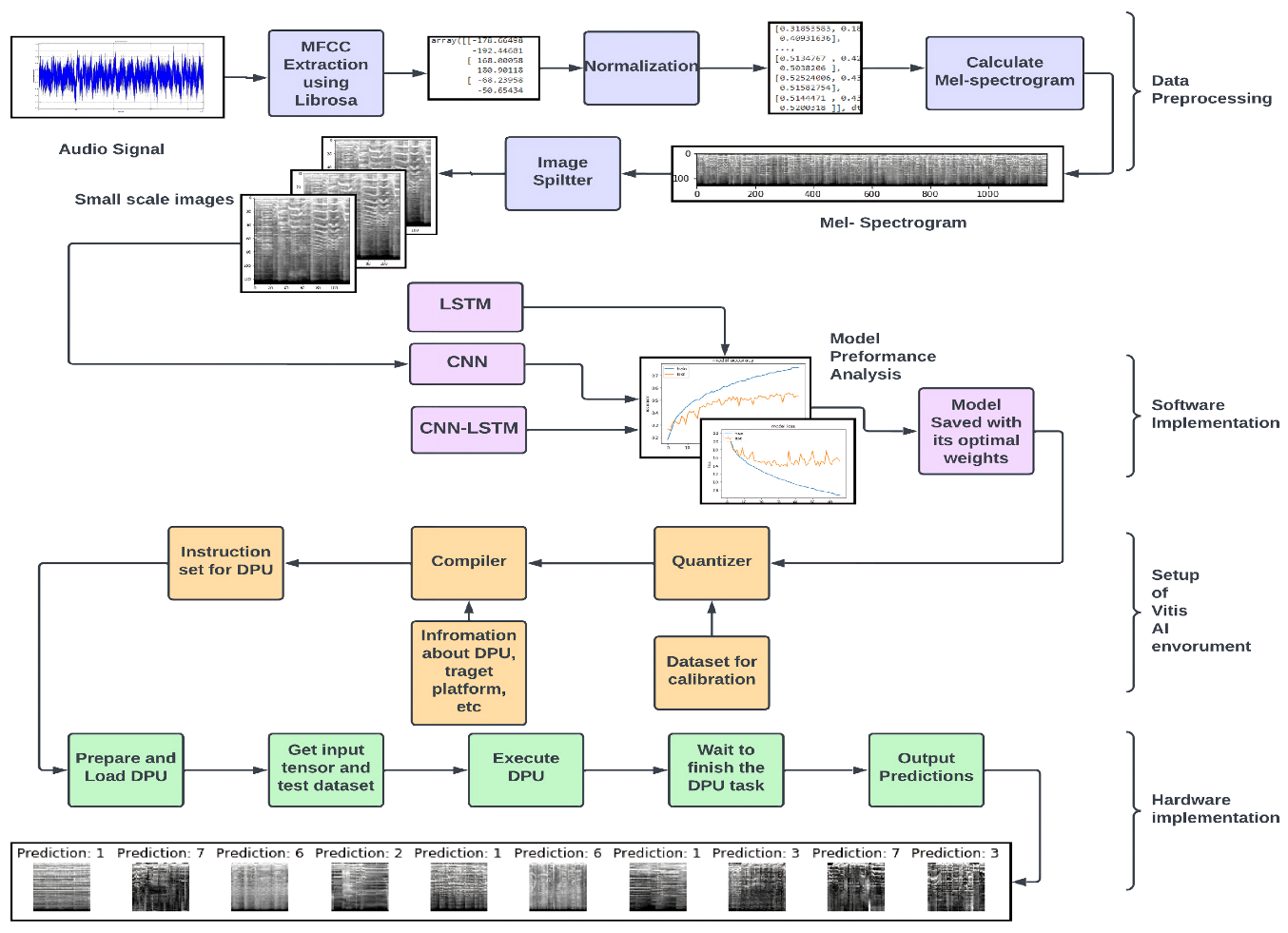
| Music Instrument |

| Classification |

| Results |

Page 12

BLOCK DIAGRAM



Page 13

Results

from sklearn. metrics import confusion\_matrix

import numpy as np

import pandas as pd

import matplotlib. pilot as plt

from sklearn import metrics

true\_labels = ['trumpet', 'cello', 'flute', 'sax', 'viola', 'oboe']

predicted\_labels = [1, 3, 5, 2, 4, 0]

# Create a mapping between string labels and integer representations

label\_map = {'trumpet': 1, 'cello': 3, 'flute’: 5, 'sax’: 2, 'viola’: 4, 'oboe’: 0}

# Convert the true and predicted labels to integer representations

true\_labels\_int = [label\_map[label] for label in true\_labels]

predicted\_labels\_int = [label\_map[label] if isinstance (label, str) else label for label in predicted\_labels]

# Get the unique classes (integer labels) from both sets of labels

classes = np. unique (true\_labels\_int + predicted\_labels\_int)

# Generate the confusion matrix

cm = confusion\_matrix (true\_labels\_int, predicted\_labels\_int, labels=classes)

#print(cm)

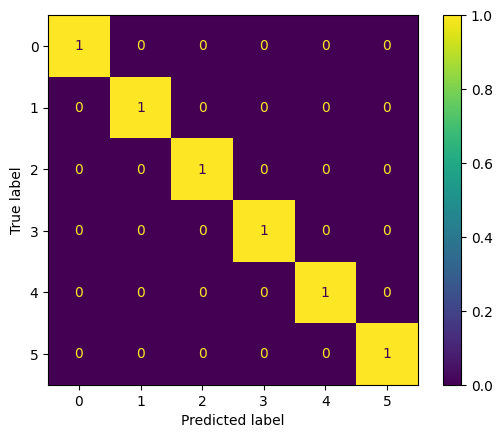
cm\_display = metrics. ConfusionMatrixDisplay (confusion\_matrix = cm)

cm\_display. plot ()

plt. show ()

Page 14

OUTPUT-



from sklearn. metrics import classification\_report

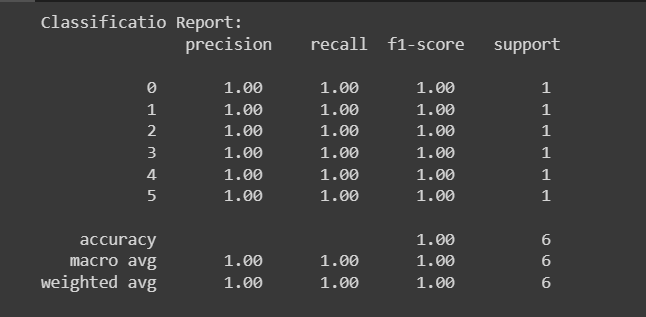
#tn, fp, fn, tp = confusion\_matrix (). ravel ()

# (tn, fp, fn, tp)

#Confusion Matrix metrics

matrix = classification\_report (true\_labels\_int, predicted\_labels\_int)

print ("Classificatio Report: \n", matrix)



Page 15

Accuracy = metrics. Accuracy\_score (true\_labels\_int, predicted\_labels\_int)

print (Accuracy)



Page 16

References (in IEEE format)

1. Birajdar, G.K., Patil, M.D. Speech and music classification using spectrogram based statistical descriptors and extreme learning machine. *Multimed Tools Appl* **78**, 15141–15168 (2019).

https://doi.org/10.1007/s11042-018-6899-z

2. [1] Arijit Ghosal, R. Chakraborty, Bibhas Chandra Dhara, and Sanjoy Kumar Saha, “Song/instrumental classification using spectrogram based contextual features,” Sep. 2012, doi: https://doi.org/10.1145/2381716.2381722