IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

MIC_FuzzyNET: Fuzzy Integral based ensemble for Automatic Classification of Musical Instruments from Audio Signals

KARAM KUMAR SAHOO [1], RIDAM HAZRA [1], MUHAMMAD FAZAL IJAZ [2] (Member, IEEE), SEONGKI KIM [3] (Member, IEEE), PAWAN KUMAR SINGH [4,5] (Senior Member, IEEE) MUFTI MAHMUD [6,7,8] (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, National Institute of Technology, Mahatma Gandhi Road, A-Zone, Durgapur, West Bengal-713209 India
[2]Department of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, Korea
[3]National Centre of Excellence in Software, Sangmyung University, Seoul 03016, Korea
[4]Department of Information Technology, Jadavpur University, Jadavpur University Second Campus, Plot No. 8, Salt Lake Bypass, LB Block, Sector III, Salt Lake City, Kolkata-700106, West Bengal, India
[5]School of Science and Technology, Nottingham Trent University, Clifton, Nottingham NG11 8NS, UK
[6]Department of Computer Science, Nottingham Trent University, Clifton, Nottingham NG11 8NS, UK
[7]Medical Technologies Innovation Facility, Nottingham Trent University, Nottingham NG11 8 NS, UK
[8]Computing and Informatics Research Centre, Nottingham Trent University, Nottingham NG11 8 NS, UK

Corresponding author: Muhammad Fazal Ijaz (email: fazal@sejong.ac.kr) and SeongKi Kim (email: skkim9226@smu.ac.kr)

This research was supported by the MSIT (Ministry of Science, ICT), Korea, under the National Program for Excellence in SW, supervised by the IITP (Institute of Information communications Technology Planing Evaluation) in 2022(2019-0-01880).

ABSTRACT Music has been an integral part of the history of humankind with theories suggesting it is more antediluvian than speech itself. Music is an ordered succession of tones and harmonies that produce sounds characterised by melody and rhythm. Our paper proposes an ensemble deep learning musical instrument classification (MIC) framework, named as MIC_FuzzyNET model which aims to classify the dominant instruments present in musical clips. Firstly, the musical data is converted to three different spectrograms: Constant Q-Transform, Semitone Spectrogram and Mel Spectrogram, which is then stacked to form 3 channel 2D data. This stacked spectrogram is fed to transfer learning models namely, EfficientNetV2 and ResNet18 which output the preliminary classification scores. A fuzzy rank ensemble model is finally employed that assigns the classifier ranks, on the testing data in order to achieve final enhanced classification scores which reduces error and biases for the constituent CNN architectures. Our proposed framework has been evaluated on the Persian Classical Music Instrument Recognition (PCMIR) dataset and Instrument Recognition in Musical Audio Signals (IRMAS) dataset. It has achieved considerably high accuracy, making our proposed framework a robust MIC model.

INDEX TERMS Musical Instrument classification, MIC_FuzzyNET, Fuzzy integral, Spectrogram, Transfer Learning, PCMIR dataset, IRMAS dataset.

I. INTRODUCTION

Sound plays an integral role in how living beings perceive the world around them and communicate with each other. Although conscious communication is multi-sensory and involves tactile as well as visual cues in addition to audio cues, most of the time we gather and analyse information about the surrounding through sound cues without many wilful attempts. However, the past few decades have witnessed considerable innovation and research by amalgamating science with the study of sound waves.

Audio comes in many forms including random noises, verbal speech, wildlife and environmental sounds, and music, which this paper deals with. Our research proposes an intelligent system which can able to classify musical sounds into their instruments based on spectrogram features. As humans, we grow up listening to various genres of music and artists. We can distinguish various instrument classes, such as percussion, wind, and string instruments, to name a few. With the escalation of online streaming platforms, both audio and video data are generated at a tremendous rate, meaning there must be services to analyse multimedia data.

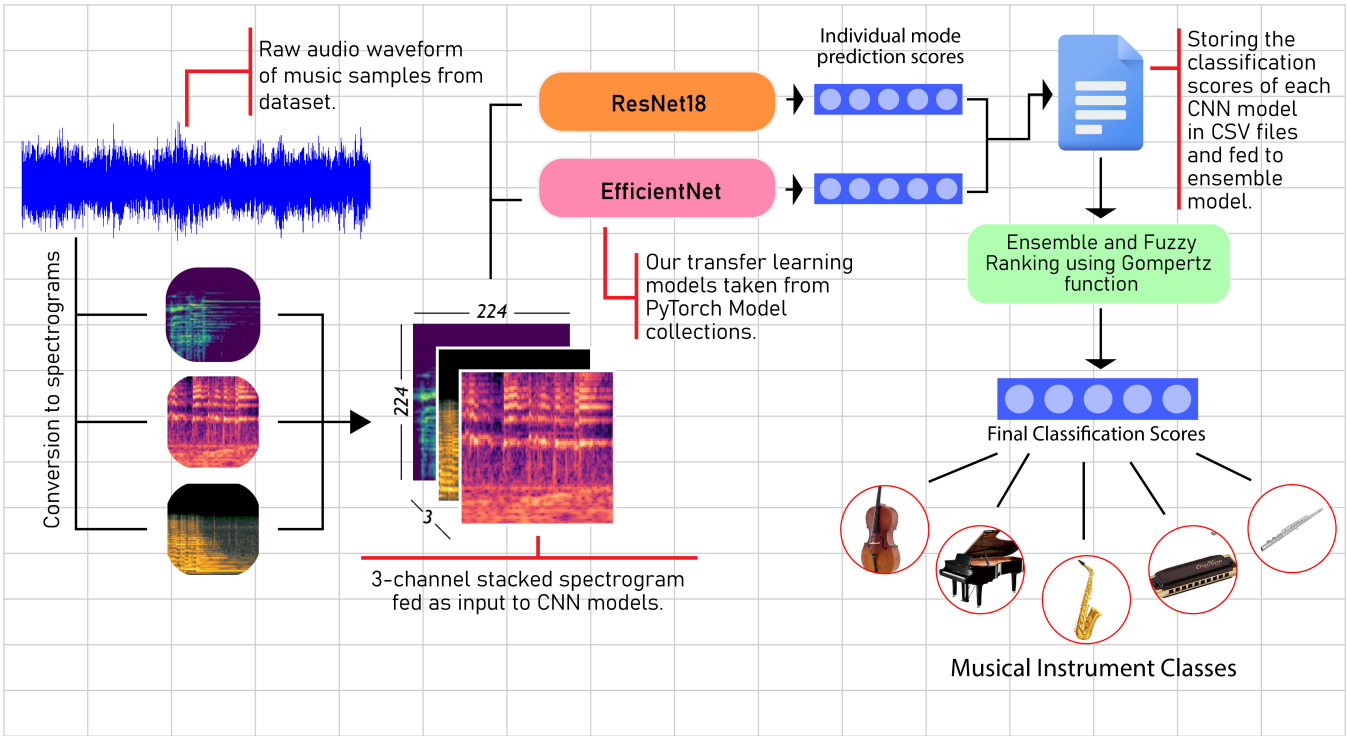


Figure 1: Overall framework of our proposed MIC_FuzzyNET model for classifying musical instruments from musical excerpts.

Music Instrument Classification (MIC) falls under the canopy of Music Instrument Retrieval (MIR) domain, which mainly deals with the analysing audio content such as feature extraction, classification, descriptor generation and segmentation to name a few. The development of MIC frameworks can assist music indexing, Human-Computer interaction systems, genre classification, and recommendation systems, among other potential applications.

Any audio classification pipeline has two important phases: picking the right set of features for feeding the classification network and designing an efficient Machine Learning or Deep Learning architecture. Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs) have had a lot of success and perform well for Machine learning and vision problems. DNN architectures are employed in various domains such as biometrics, healthcare, image classification, segmentation and generation, Natural Language Processing and audio classification and understanding.

To train any deep learning model, audio data must be pre-processed and features must be extracted to feed the model. Sound has predominantly two kinds of features, the first is temporal features, and the second is spectral features which are obtained by converting the temporal features into the frequency domain with Fourier transforms. Zero-crossing rate, maximum amplitude, signal energy, and minimum energy are some examples of temporal features. On the other hand spectral features are represented using Mel Frequency Cepstral coefficients (MFCC), chroma-stft, spectral density, constant-Q transform (CQT) spectrograms, semitone spectrograms, central bandwidth, and central rolloff, among other features. Keeping into account the effectiveness and past success of transfer learning CNN models, our proposed research has taken into consideration 3 spectral features namely, Mel spectrogram, CQT spectrograms, Semitone spectrograms and processed them into a format recognised by CNN architectures. These architectures are pre-trained on huge datasets and these weights are reused for extracting essential patterns from the modified input spectrogram for classifying the dominant instrument from the audio/music sample. A pictorial view of our entire framework has been represented in Figure 1.

A. MOTIVATION AND CONTRIBUTIONS

1) A lot of data is needed for building an end-to-end deep learning model. However, there is a shortage of labelled and organised data for the purpose of MIC problem, as a result of which we used pre-trained transfer learning models namely, EfficientNetV2 and Resnet.
2) Training deep learning models directly on audio samples is computationally expensive because of massive sampling rates. We have stacked three spectrogram layers: Semitone, CQT and Mel spectrogram, together into a 3D stacked spectrogram to extract features using convolution layers thereby providing the classification models with a better understanding of the audio data.
3) The use of a single transfer learning model to train the

spectrograms may result in an imbalance. We use the ensemble approach to gain a weighted opinion of all base classifiers in order to reduce noise and get better as well as unbiased prediction scores. As a result, it is an innovative approach to the MIC problem.

4) A modified Gompertz function is employed to allocate fuzzy ranks to the prediction scores of the different models. The Gompertz function saturates exponentially to an asymptote, with prediction scores seldom falling below zero. Because fuzzy rankings based fusion applies adaptive priority weights to each model prediction scores, it is different and more efficient than typical ensemble pipelines.
5) We have compared our performance and evaluation metrics with some recent approaches for MIC problem and inferred that our MIC_FuzzyNet framework transcends them, as a result, proving the novelty of our ensemble approach.
6) Two open-source datasets are used to train and evaluate the MIC_FuzzyNet model: Persian Classical Music Instrument Recognition (PCMIR) and Instrument Recognition in Musical Audio Signals (IRMAS). In comparison to other machine learning as well as deep learning frameworks, we are able to achieve the state-of-the-art accuracy.

II. LITERATURE REVIEW

In 1995, Kaminsky and Materka [29] used principal component analysis, short term RMS energy envelope and ratio/product transformations for classification of monophonic instruments with the help of K-nearest neighbour classifier and neural networks. In 2000, Eronen and Klapuri [16] came up with a system that took into account various temporal and spectral features to classify pitch-independent musical instruments and integrate this into a transcription system.

Essid et al. [17] studied various audio features and used inertia ratio maximisation and genetic algorithms with feature space projection for choosing the most relevant set of features. They employed Gaussian Mixture Models (GMMs) and Support Vector Machines(SVM) for the musical instrument classification phase with 75% accuracy in their baseline GMM model. Heittola et al. [23] used a sound separation technique on polyphonic music data using a source-filter model and Mel-frequency cepstral coefficients were the choice of audio features. They achieved an accuracy of 59% on 6 note polyphonic music.

Many researchers focused on instruments of a certain genre or culture for uplifting art and research on their history. Mousavi et al. [36] curated the PCMIR dataset which we will also be using in this paper. He used Fuzzy entropy measure for feature selection and a Multi Layer Perceptron for classification of Persian instruments. Shetty and Hegde [46] worked on classifying 10 different Carnatic musical instruments by extracting Linear prediction coefficients (LPC) and MFCC features and comparing different Deep learning and Machine Learning models for classification. Bosch et al. [6], in their paper, presented approaches combining source separation and instrument recognition, where they learned that there is 32% improvement of the micro F1-measure over the original algorithm.

Han et al. [22]'s approach to identify instruments went around extracting the various features from mel-spectrogram using convolutional layers of CNN. They experimented with various activation functions, out of which, 'ReLU' (alpha = 0.33) gave the best classification result with the overall F score of 0.602 on IRMAS training data, which we have used as well. Goel et al. [20] showed how we can use musical genres to distribute and manage music datasets to increase the accuracy in finding a music item a person wants to listen to. They presented research for creating an appropriate model for genre recognition in audio files using machine learning classifiers on the IRMAS dataset. Then the classification of genre using Synthetic Minority Oversampling Technique (SMOTE) algorithm has been characterised in the confusion matrix. They achieved a maximum accuracy of 81.56% using the ensemble classification model.

	Total Samples	Training	Testing	Validation
Ud	339	271	35	33
Tar	461	368	47	46
Santur	443	354	45	44
Kamancheh	363	290	37	36
Ney	435	348	44	43
Setar	369	295	38	36

Table 1: Distribution of samples in different classes in the train, test and validation subsets for PCMIR dataset.

III. EXPERIMENTAL SETUP

Deep learning models have been loaded from Keras models API and trained using resources from Google Colaboratory. GPU used is Tesla K80 with 2496 CUDA cores and VRAM of 12GB DDR5, single core hyper threaden Xeon processors clocked at 2.3GHz and 12.6GB of available RAM, all of which is provided by Google Colaboratory workspace. Initial data visualisation and dataset analysis have been performed in local machine with AMD Ryzen 5 4600H CPU, 8GB of DDR4 RAM and 4GB of GTX GeForce 1650 GPU.

IV. DATASET DESCRIPTION

A. PCMIR

The PCMIR dataset was designed to study few important musical instruments used in Persian Music. The dataset consists of music samples belonging to 7 musical instruments: Kamancheh, Tar, Ney, Tonbak, Santur, Setar and Ud. This dataset is primarily important because it is a maiden research conducted for classifying Persian musical instruments. For the transfer learning phase we have split the dataset into training, testing and validation data in the ratio 8:1:1. The data distribution is given in Table 1.

	Total Samples	Training	Testing	Validation
Cello	388	310	40	38
Flute	451	360	46	45
Organ	682	545	69	68
Piano	721	576	73	72
Saxophone	626	500	64	62

Table 2: Distribution of samples in different classes in the train, test and validation subsets for IRMAS dataset.

B. IRMAS

The IRMAS dataset consists of polyphonic musical samples with presence of 2 or more predominant musical instruments. There are 3 second musical extracts in $.wav$ format of 16 bits. There are a total of 11 classes in the original dataset which are: electric guitar, organ, piano, saxophone, trumpet, violin , cello, clarinet, flute, acoustic guitar, and human singing voice. Since these are polyphonic in nature we have considered only 5 classes of instruments which are cello, flute, organ, piano and saxophone. By leveraging the pre-trained weights from the transfer learning models, 5 instrument classes were taken from the dataset to show that our proposed model can perform well even with limited datasets. Furthermore, the 5 instruments belongs to a different class of musical instruments, adding to the robustness of our proposed framework.

We have evenly and randomly split the samples in a $8:1:1$ ratio as shown in Table 2 into training-testing-validation subsets. This particular ratio was decided to enable the model to train on enough data samples since our proposed framework deals with small datasets, while optimizing the number of unseen testing samples on which our framework infers the performance of the ensemble model. The data samples are not collected in studio environment but collected across different genres, artists and decades which is why there is great variety in quality of data points.

V. METHODOLOGY

The proposed framework has been divided into the following subsections, namely feature extraction, creating stacked spectrogram, CNNs, model training with transfer learning and finally assigning the fuzzy ranks to the CNN models for ensemble learning.

A. FEATURE EXTRACTION

An eclectic choice of features from the musical samples we have, is of prime necessity if we want to extract more information for instrument classification. There is availability of various temporal and spectral features corresponding to audio data which include Mel spectrograms, LPCC, MFCC to name a few. However there are few spectrograms which are easy to visualize and can be used for our MIC problem. For our proposed model, we have chosen three features which are CQT spectrogram, Semitone Spectrogram and Mel Spectrogram.

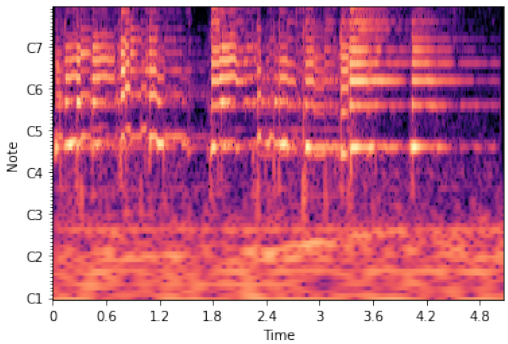


Figure 2: A CQT spectrogram corresponding to a Setar audio sample.

1) Constant Q Transform Spectrogram

A spectrogram is a visual representation of the signal strength of a signal versus time at various frequencies present in a particular waveform. In a spectrogram, it can be seen whether there is more or less energy but it can also seen how energy levels vary over time. In the field of signal processing, the CQT, which goes by CQT[15], transforms a data series to its corresponding frequency domain. It is derived from the Fourier transform and also closely related to the complex Morlet wavelet transform.

The transform can be taken as a series of filters δf_k, spaced logarithmically in frequency, with the k-th filter having a spectral width δf_k equal to a multiple of the previous filter's width:

$$\delta f_k = 2^{1/n} \cdot \delta f_{k-1} = \left(2^{1/n}\right)^k \cdot \delta f_{\min}, \tag{1}$$

where, δf_k is the bandwidth of the k-th filter, $\delta f_{\min}$ is the central frequency of the lowest filter, and n is the number of filters per octave.

In CQT, the frequency will be converted into a log scale and the colour dimensions (amplitude) into decibels to form a spectrogram. Figure 2 shows a CQT spectrogram corresponding to a Setar audio sample.

2) Mel Spectrogram

A detailed graph called a spectrogram contains data on the frequency, duration, and amplitude of sound waves. Colors are utilised as the third dimension in spectrograms, which are typically two dimensional. A Fourier Transform is applied to each of the broken-up, little temporal chunks or frames that make up the audio stream. The colour scales of the spectrograms indicate the frequency's amplitude or power in the resulting frequency versus time graph. Humans hear frequencies logarithmically rather than linearly. As a result, a 100Hz difference in the Mel Scale corresponds to what a human would typically perceive in the actual world. This issue is resolved by the Mel scale, which converts a tone's perceived frequency to its actual frequency.

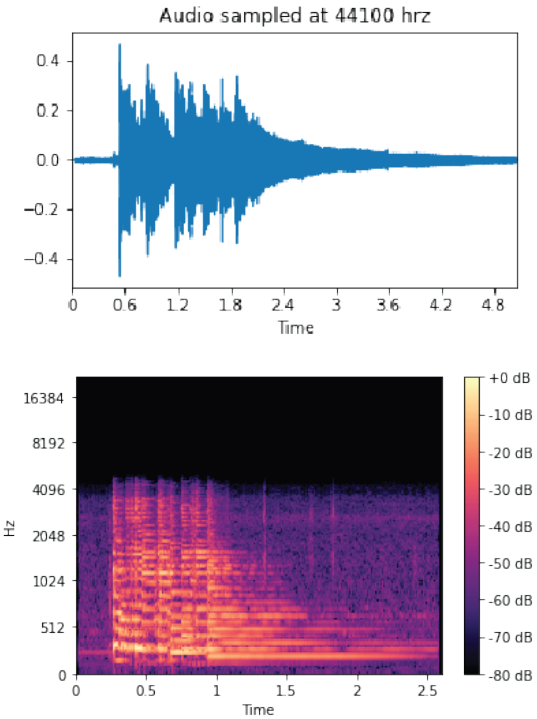


Figure 3: An audio sample and its corresponding Mel spectrogram representation.

Mel spectrograms hold sound information which the human ear could perceive. The Mel scale and Hertz(Hz) are related by the given formula:

$$m = 2595\, log_{10}(1 + \frac{f}{700}) \tag{2}$$

Figure 3 shows the raw audio waveform of a clip taken from Ud instrument class of the PCMIR dataset along with its Mel spectrogram which has been extracted using Librosa library in Python.

3) Semitone Spectrogram

The smallest musical interval employed in Western music is the semitone, commonly referred to as a half step or half-tone. When performed harmonically, a semitone is thought to be the most discordant[39]. A semitone is the distance in pitch between two notes that are close to one another on a 12-tone scale. When a test signal is run through a signal processor, such as a filter, the results are typically analysed using spectrograms to show the performance. For particular spine notes, the semitones filter determines melodic semitone intervals. The filter can highlight repetitions, steps, leaps, and the direction of intervals in the rendered notation.

Significance of chosen spectrograms

The hertz values are remapped to the Mel scale in the Mel spectrogram. So, Mel spectrograms are better suited for applications that need to replicate human hearing perception, such as music.

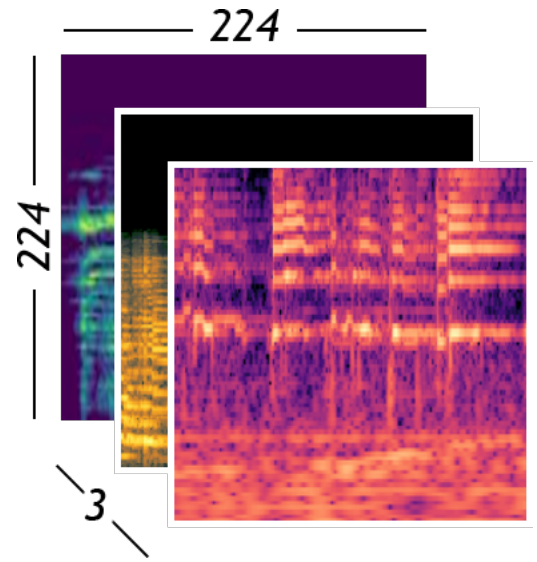


Figure 4: Pictorial representation of the 3D stacked spectrogram employed as our feature vector in the present work.

The CQT has a few characteristics that make it a better fit for musical data when compared to the rapid Fourier transform. Since the output of the transform is essentially amplitude/phase against log frequency, fewer frequency bins are required to adequately cover a given range. When frequencies cover several octaves, this is advantageous. This reduction in output data is significant because the human hearing range extends from 20 Hz to around 20 kHz or about ten octaves.

The majority of musical instruments, nowadays, employ the 12-tone chromatic scale, which divides an octave into 12 evenly spaced parts on a logarithmic scale. A semitone[53], the tiniest interval in music, is seen as dividing each part. For most instruments, tuned to an evenly tempered scale, the middle octave note A is tuned at 440Hz. To calculate the short-time mean-square power (STMSP) for each band, the subsequent filtered time samples are added together. However, the semitone spectrogram generates 85 filters with one-semitone bandwidths and the MIDI pitches keeping [24, 108] as center frequencies when it is launched using the default set of parameters. In 1972 Deutsch[13] claimed that although people seem to perceive in octaves, pitch organisation within octaves varies culturally. The equal-tempered scale, which splits the octave into twelve equally spaced semitones, has served as the main organisational framework for Western music. It is to be noted that the smallest pitch unit in Western music is the semitone.

B. CREATING STACKED SPECTROGRAM

The novelty of our proposed algorithm, partly lies in the presentation of audio features into a 3D matrix, the structure of which is similar to that of images. Semitone, Mel and Constant Q transform spectrograms are the selected features for our MIC_FuzzyNET model and they are stacked together

in a 3D matrix, where each spectrogram is analogous to a channel in an image.

The transfer learning models namely, EfficientNetV2 and Resnet18, are extremely effective in classifying image datasets. They can easily pick up low level and high level features from images and determine the classes or segment images with high levels of accuracy. We are exploiting this property of the CNN models in our proposed paper. By synthesising analogous data structures using the constituent spectrograms, the CNN models can similarly pick up features and perform convolutions, pooling and classification on the stacked spectrograms. Hence, we have successfully reduced the data dimensions from audio sampled at 44kHz to 3D matrices of size (3x224x224). Notwithstanding the 3 dimensional shape of inputs, the CNN models perform 2D convolutions and not 3D convolutions. 2D convolutions implies that the kernel traverses in 2 dimensions only (i.e., along the height and width of the image or similar input). Despite the kernel being a 3D matrix, it will move only along the height and breadth of the image while the third dimension overlaps with the channel dimension of the images. Fig 5 represents the 2D convolution operations performed on 3D data with 3D kernels.

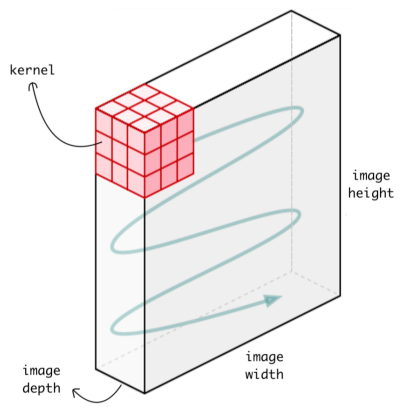


Figure 5: Diagrammatic representation of 3D kernel on 3D input data with 2D convolution operations.

Using these three spectrograms, we are able to explore different modalities of audio data. All the three constituent spectrograms represent different properties of the audio data, hence the stacked spectrogram gives us the power to simultaneously perform convolutions on all three feature vectors and make more informed and accurate predictions. Figure 4 shows the pictorial representation of the 3D stacked spectrogram for a clip taken from the Ud instrument class of the PCMIR dataset.

C. CONVOLUTIONAL NEURAL NETWORKS

In the field of deep learning, CNN is a type of Artificial Neural Network (ANN) which is generally used in image recognition and processing because it is designed especially to process pixel data. CNN are powerful Artificial Intelligence (AI) that perform both generative and descriptive tasks, by using machine vision that have both image and video recognition, along with recommender systems and the Natural Language Processing (NLP).

If we consider the history of CNN, LeNet, named after Yann LeCun, was one of the very first CNNs which helped immensely the field of Deep learning. This pioneering work was named LeNet5 [34] after many previous successful and building efforts since 1988. In those days, the LeNet architecture was used in research works related to character recognition tasks like reading pin codes, digits, etc. AlexNet, developed in 2012, showed that AI, a branch of deep learning, which uses multi-layered neural networks, needs to be look at. The availability of large sets of data, such as the ImageNet dataset with many labeled pictures, and vast compute resources enabled researchers to make complex CNN that would perform computer vision tasks that were previously impossible.

CNNs are generally used to take the benefit of their ability to develop an internal representation of a two-dimensional image. This way it allows the models to learn position and scale invariant structures in the data, which is very important when working with image datasets. CNNs work well with data that has a spatial relationship, such as in a document of text, there is an ordered relationship between words, or in the time steps of a time series etc. Although ConvNets achieve state-of-the-art (SOTA) results on problems such as document classification, mostly used in sentiment analysis, entity extraction and related problems in NLP. This paper [10] showed how deep ConvNets surpassed other traditional machine learning algorithms like Random Forest, SVM and Gradient Boosting classifiers, especially in terms of classification accuracy.

The reason why CNNs are highly rated is because of their architecture, which has no need for feature extraction. The core concept of CNN is, it uses convolution of image and filters to get invariant features which are passed onto the next layer and therefore it learns feature extraction on it's own. The features in the next layer are convoluted with different filters to generate more invariant and abstract features and the process continues till one gets the final output which is invariant to occlusions. The CNN input is traditionally two-dimensional, a field or matrix, but also it works with one-dimensional, allowing it to develop an internal representation of a one-dimensional sequence. Now, CNNs can extract informative features from images, eliminating the necessity of manual image processing methods, which is the traditional one, used for years.

The formula for convolution can be written as follows:

$$s[t] = (x * w)[t] = \sum_{a=-\infty}^{a=\infty} x[a]w[a+t] \tag{3}$$

where, *s[t]* is Feature map, *x* is input and *w* is kernel. There are generally three-way layering in a CNN: Convolutional Layers, Pooling Layers and Fully Connected Layers. When

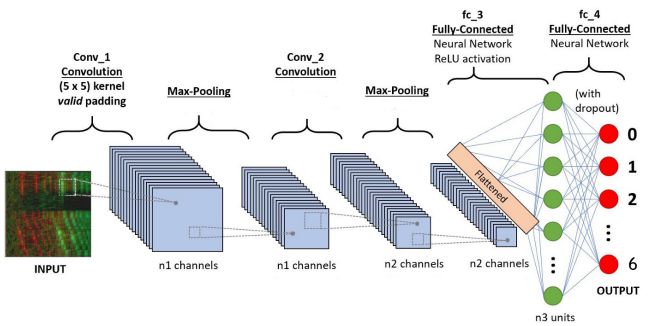


Figure 6: A pictorial representation of a CNN sequence to classify stacked spectrogram.

an image is given as input in the ConvNet model, each layer generates several activation functions that are passed onto its subsequent layers. The convolutional layer extracts primary features like horizontal or diagonal edges. This output is passed onto its subsequent layer which as we move deeper into the network, can identify even more complex features such as objects, faces, etc. The initial convolutional layer or layers learn characteristics like edges and straightforward textures. Later, convolutional layers pick up elements like more intricate patterns and textures. The last convolutional layers pick up on properties like objects or their components[1]. The completely linked layers acquire the ability to link the activations from the specific classes to be predicted to the high-level characteristics. Based on the activation map of the ultimate convolution layer, the classification layer outputs a collection of confidence scores (a value between 0 and 1), that specifies how likely the image is to belong to a class or set of desired output. For example, if we have a Network model that detects cats, dogs, and cars from just their images, it is possible that the output of the final layer contains any of those already considered input images. Figure 6 shows the schmatic diagram of the various layers in CNN sequence, which has been used to classify stacked spectrogram.

The pooling layer is used to reduce the spatial size of the convolved features, like the convolutional layer. There are mainly two styles of pooling: Average pooling and Max pooling. In Max pooling, the maximum value of a pixel from a part of the image covered by the kernel is selected. This layer discards the noisy activation altogether and also performs de-noising alongside dimensionality reduction. On the opposite hand, Average pooling returns the average of all the values from the portion of the image covered by the kernel. It generally performs dimensionality reduction, as a noise suppressing mechanism, as a result of that, Max pooling performs tons better than Average pooling. At the last stage of the network, the fully connected layers are used, after feature extraction and consolidation has been performed by the convolutional and pooling layers. These are used to create final non-linear combinations of features and then for creating the final predictions by the network.

As we have already covered, CNNs do well when it comes to picture classification. Additionally, earlier studies have already demonstrated how well-known CNN architectures, including AlexNet, VGG, Inception, and ResNet, performed when applied to audio-based classifications. To create a spectrogram, which served as an input to the network models, the audio time signal is typically decomposed using a short-time Fourier transform. Then, we use transfer learning to accomplish the goal with relatively little data. After the majority of the layer weights are frozen during the transfer learning process, a pre-trained network is employed, with only a few of the last layers being retrained using the audio training data. The next step is to train CNNs using spectrogram and raw audio inputs. Layer-wise Relevance Propagation (LRP) is then chosen to investigate further how the models choose features and make judgments. Thereafter, results also show that spectrogram inputs result in higher accuracy over raw audio inputs. The audio signal is then pre-processed with the Mel spectrogram in order to describe it in a more detailed way. In order to effectively identify the audio data, both transfer learning along with a smaller CNN architecture can be used at the same time.

D. MODEL TRAINING WITH TRANSFER LEARNING

Creating newer architectures for every classification problem in deep learning is bottlenecked by the lack of sufficient and properly labelled data. To tackle this problem, transfer learning is employed where models pre-trained on millions of data points are reused for problems with small datasets. Transfer learning refers to the exploitation of information gathered by training on very large sized datasets to problems with less data. Models trained to differentiate between different plants can be used to also classify food images. Deep neural networks only recognise sophisticated patterns like color and specific features in the later layers; the initial few layers only recognise general patterns like shape, edges, noise, etc. To identify or predict our own datasets, we can use the initial layers, remove the latter layers, and add our own layers. This is achieved by transferring the weights from the previous model to a newer model. Not only does transfer learning leverage the usefulness of small datasets, but it also reduces training time by freezing weights of some layers and training only a subset of layers for the new problem. CNN models can sometimes be very resource demanding with lots of computations which makes it unsuitable for systems with less computational power. Hence, transfer learning models can be used where the re-training of the entire model is unnecessary. In our chosen models, we have only modified the valence layer which is the linear dense layer by changing the output classes from their default values to that of the number of instrument classes according to the datasets. The training method is quicker than updating the complete CNN architecture using forward and backward propagation since we are only training the final layer of the CNN models rather than the full architecture. As a result, our suggested model performs better in terms of time complexity.

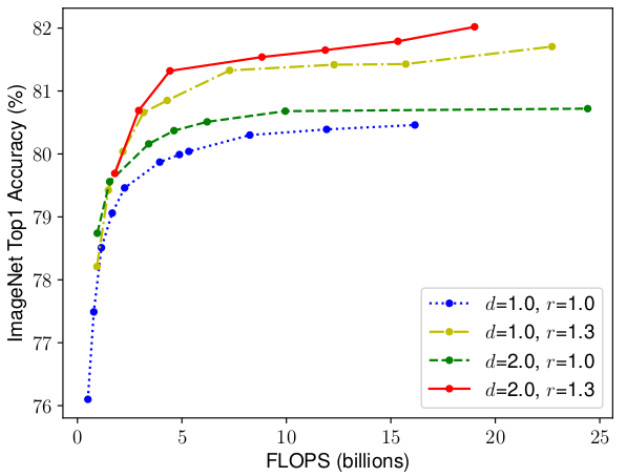


Figure 7: Effect of coupling different factors while scaling in a CNN network.

We have employed pre-trained transfer learning model in our proposed work for cross-domain datasets. Both the EfficientNetV2 as well as ResNet18 models are initially trained on ImageNet but our dataset has spectrogram images. However, leveraging transfer learning models for datasets where the source and target domains are different have been experimentally verified[38] and is an effective choice for small-sized datasets as already seen in the literature.

Both the datasets are divided into training, testing, and validation subsets in the ratio 8:1:1. The training and validation datasets containing the stacked 3 channel spectrogram data are used to fine-tune the 2D CNN transfer learning models. The test data samples are unseen by the model during training. Both the ResNet18 as well as EfficientNet models can be accessed from the Pytorch Model Zoo. The two transfer learning models are described below as follows:

Stage	Operator	Stride	#Channels	#Layers
0	Conv 3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1 & Pooling & FC	-	1280	1

Table 3: A tabular representation of the EfficientNetV2 architecture used in the present work.

We used early stopping to reduce overfitting and used fewer epochs, with an empirical cap of 20 epochs, to ensure that our training phase did not overfit the training data. Additionally, we used an ideal 8:1:1 train-validation-test split in order to allocate more data to the training phase.

1) EfficientNetV2

Convolutional networks have paved their way into computer vision community and have retained a permanent spot, however the problem of model scaling remains quiet a problem. Model scaling refers to the problem of increasing performance accuracy at the cost of increasing model depth and complexity in architecture. Often times the tuning of model depth and layer sizes becomes time and resource demanding which is why this new class of CNNs, EfficientNet[52], were created by Google in 2019.

The EfficientNet class of CNNs have a mobile-size architecture with reduced parameters and Floating Point Operations per Second compared to contemporary state of the art CNN architectures. They employ a compound scaling methodology to maximise the gain in accuracy proportional to model size.

The main idea behind EfficientNet comes from the observation that a model can be scaled wither by increasing layer depth, input resolution or width of network. However changing any one factor after a certain point saturates the accuracy, which is experimentally observed. Hence, change in any one factor must be coupled with tweaks in the other properties to maximise the gain in accuracy. Hence during convolutional network scaling, resolution, width and model depth must be scaled proportionately, the result of which is shown in Figure 6. In our proposed paper, we have used a newer version of EfficientNet called EfficientNetV2 which use progressive training and fused-MB convolutional layers[56]. These Fused-MB Convolutional layers are characteristic to the EfficientNetV2 which has lesser parameters and FLOPS while also being able to use modern GPU/CPU accelerators. Unlike traditional EfficientNet models which compound scales all the stages(resolution, depth and width) equally, the EfficientNetV2 scales the layers in a non-uniform fashion. This is because the early layers only are responsible to capture high level features and don't require much scaling.

The EfficientNetV0 has achieved 87.3% accuracy on the ImageNet dataset with faster training times compared to state of the art architectures. Hence, it is a perfect choice for our transfer learning phase. Table 3 gives a tabular view of the model architecture which contains the new fused-MB convolutional layers along with the MBConv layers from the initial EfficientNetB0 models.

2) ResNet18

ResNet also called Residual Network was developed in 2015 and is a 2D CNN model that employs the concept of residual mapping which is effective against the "degradation problem" in deep neural networks. The optimisation phase of the CNN model is greatly enhanced by the residual mapping. The ResNet-18 is pre-trained on the ImageNet dataset which has been trained on millions of images, making it a good CNN model for transfer learning. The input size of images for the model is 3x224x224. For ResNet architectures, the performance is greatly impacted by the depth of the network(total number of layers). ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-110, are few depth wise implementations of ResNet architectures. The ResNet18 model (see Figure 8 for further details) used in our proposed paper

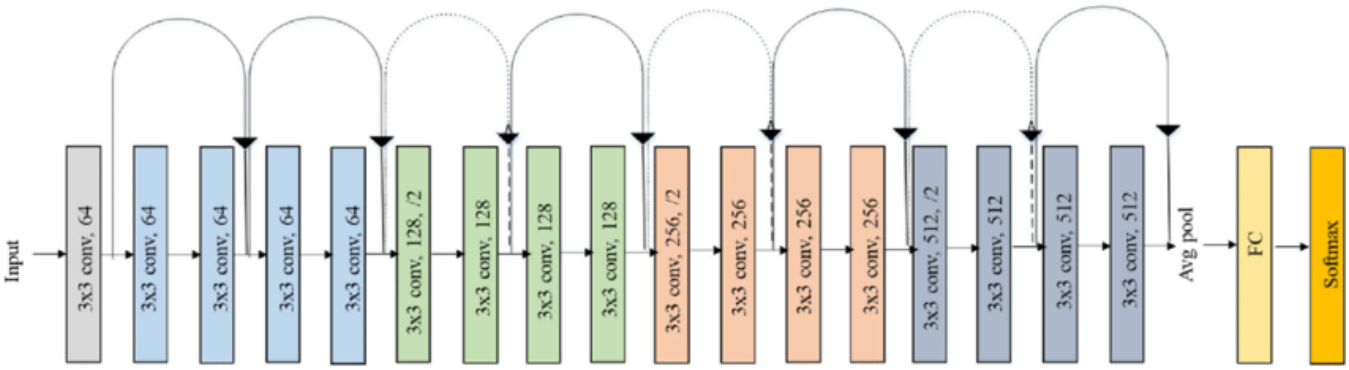


Figure 8: ResNet-18 CNN transfer learning model in pictorial form.

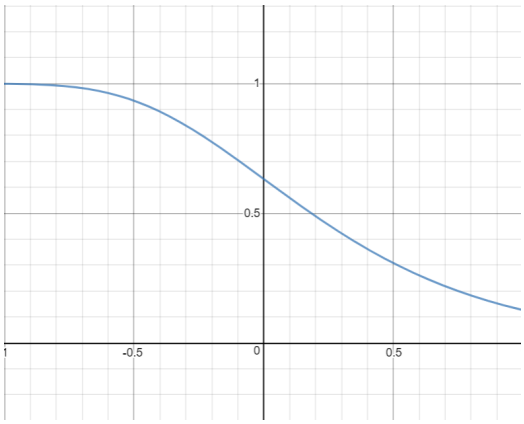


Figure 9: Graphical representation of the modified Gompertz function used to determine the fuzzy ranks of constituent classifiers.

is an ideal trade off between performance and computational complexity.

Because the chosen CNN models, in the present work, have substantially fewer parameters, the system needs less memory. It is to note that the Resnet18 model needs 41MB to load, whereas the Resnet152 model needs 214MB. Despite this significant increase in model size, it was unable to deliver a commensurate performance advantage. Additionally, the EfficientNetV2 model requires 21MB of space to load. Our models perform better than other well-known models like AlexNet (216 MB), Densenet161 (106 MB), and VGG16 (489 MB), both in terms of performance as well as memory requirements.

E. FUZZY RANKING

In the literature, the traditional ensemble method uses pre-calculated weights for the classifiers and assigns equal weight to the classification scores of all constituent CNN models. The main issue with such an ensemble is the creation of static weights, which can be challenging to control in the section when we categorise samples. However, each base classifier's predictions rankings are taken into account for each sample separately in the proposed fuzzy-rank framework. By using our ensemble technique, rankings for prediction can be obtained that are more favourable and accurate. Since this is a dynamic approach, it is not necessary to initialise new weights for various data samples.

Time series that increase gradually at the beginning and end of a period are described by the Gompertz function. Although it is now frequently used in biology, it was primarily used to explain the mortality rate in proportion to advancing age. The Gompertz function can be used to explain population increase, the development of cancerous tumours, the spread of bacterial colonies, and the number of people affected by an epidemic. We use the following equation to understand the function:

$$f(t) = ae^{-e^{b-ct}} \tag{4}$$

where, a represents an asymptote, b determines the x-axis displacement, c scales the y-axis, and e is Euler's number.

Figure 9 depicts the modified Gompertz Function that is used in our suggested approach. We have N number of prediction scores for each image in the test split of the database, where N is the number of constituent models. As previously mentioned, we used three CNN models for transfer learning, hence N=3. If the dataset's label count is L, then:

$$\sum_{l=1}^{L} S_l^{(n)} = 1; \forall n, n = 1, 2, 3, \ldots, N \tag{5}$$

The prediction scores for each class for each set of sample data, denoted by S in Equation 5, are taken into consideration for constructing the fuzzy ranks. The following formula gives the fuzzy ranks due to the nth constituent model for the lth class:

$$\begin{aligned} R_l^{(n)} &= (1 - \epsilon^{-\epsilon^{-2\times S_l^{(n)}}}) \\ &\forall l, n;\ n = 1, 2, ..., N;\ l = 1, 2, ..., L \end{aligned} \tag{6}$$

There may be k top classes that correspond to each class in the dataset; in our suggested strategy, we have selected "2" as these top classes. For the class l, the eqs. (7) to (8) is utilised

to determine the fuzzy ranks $(FRSl)$ and complement of confidence factor sum $(CCFSl)$. A penalty value of P_l^R and P_l^{CF} is applied on the appropriate class if the label l does not fall under the top K classes. By multiplying the (FRS_l) and $(CCFS_l)$ and selecting the class with the lowest value overall, as demonstrated in eq. (9), the final projected class for the data instance X is determined.

$$FRS_l = \sum_{i=1}^{N} \begin{cases} R_l^{(i)}, & if R_l^{(i)} \in K^{(i)} \\ P_l^R, & otherwise \end{cases} \tag{7}$$

$$CCFS_l = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} CF_l^{(i)}, & if R_l^{(i)} \in K^{(i)} \\ P_l^{CF}, & otherwise \end{cases} \tag{8}$$

$$class(\mathbf{X}) = min\{FRS_l \times CCFS_l\} \tag{9}$$

VI. RESULTS AND DISCUSSION

In the next section, a tabular data regarding the results are presented which has been acquired after working on the two aforementioned freely available MIC datasets. The final ensemble model, as well as the assessment measures and performance of the transfer learning models, are all thoroughly detailed. Combining an ensemble technique with deep learning 2D CNN models that take as input modified 3 channel inputs enabled us to attain state-of-the-art performance in categorising instruments from polyphonic music samples, according to our findings.

A. EVALUATION METRICS

Accuracy, Precision, Recall, and F-1 score are the evaluation measures used to assess performance. The most used metric for deep neural network challenges is training, validation, and testing accuracy, hence our proposed study also uses it. The next subsections thoroughly compare our suggested model and other earlier frameworks and architectures.

Basic parameters like True Positives, True Negatives, False Positives, and False Negatives can be used to produce the aforementioned evaluation metrics. The following are the related formulas:

$$Accuracy_x = \frac{\sum_x M_{x_x}}{\sum_x \sum_y M_{x_y}} \tag{10}$$

$$Precision_x = \frac{\sum_x M_{x_x}}{\sum_x \sum_y M_{y_x}} \tag{11}$$

$$Recall_x = \frac{\sum_x M_{x_x}}{\sum_y M_{x_y}} \tag{12}$$

$$F1\ Score_x = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{13}$$

The percentage of correctly classified data to all classified data is a model's accuracy. The aforementioned calculations indicate that high precision and recall will result from reducing the overall amount of false positives and false negatives. Since the F1-Score takes into account both recall and precision values, it is safe to claim that it is a good evaluation metric that becomes 1 if both recall and precision values become 1.

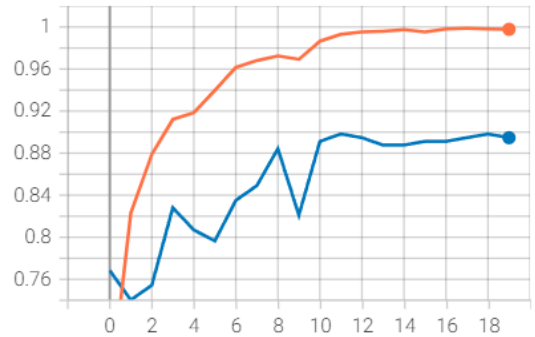
B. PERFORMANCE OF CNN MODELS

The two base 2D CNN models have been loaded from Pytorch and pre-trained on the most important image dataset, ImageNet[12] dataset. The pre-trained weights of the CNN models for the initial layers have been frozen to prevent training the entire model and re-initializing the weights. Since the ImageNet dataset has a total of 1000 classes, the classification layer also has an output dimensions of 1000. The final prediction dense layer have been modified to the shape (1, num_of_classes) for our training phase. Each base classifier has been trained for 25 epochs after which the model starts overfitting and converging. The model weights with best validation accuracy have been saved for the testing phase. The training and validation phase have been run with a different number of epochs and batch sizes and the hyperparameters have been chosen empirically for our proposed model.

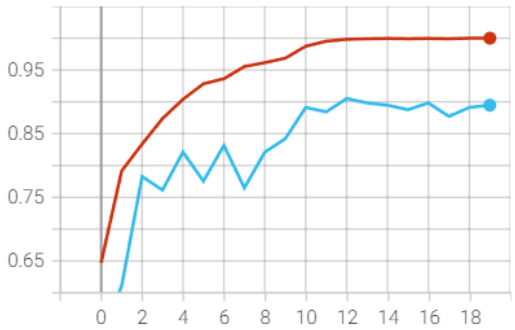
The EfficientNetV2 model achieved 99.86% training accuracy, 89.82% validation accuracy and testing accuracy of 85.67%. The training starts to converge after 10 epochs, so the epochs were experimentally set to 20 and not more. Since the audio excerpts in IRMAS dataset are polyphonic and even humans find it difficult at time to differentiate between instruments when melody contains multiple musical sounds from different instruments, testing accuracy of 85.67% is satisfactory. The ResNet18 model has 100% training accuracy, 91.57% validation accuracy and testing accuracy of 85.01% for the IRMAS dataset.

For the PCMIR dataset, the performance of 2D CNN models is remarkable due to the monophonic nature of music excerpts making it easier for the neural network to classify the instruments discretely. The Resnet18 model has 100% training accuracy at peak which converges after 10 epochs. The validation and testing accuracies are found to be 98.74% and 96% respectively. Coming to the EfficientNetV2 model, we have achieved the maximum training accuracy at 99.99%, validation accuracy of 98.74% whereas the final testing accuracy is found to be 96.34%.

Our CNN models can recognise more features for classification because we are simultaneously employing three separate spectrograms instead of only one. On the ImageNet dataset, which contains 3 channel RGB images, the transfer learning models have already been trained. We did not need to alter the model architecture as a result. Because the dynamic ensemble model incorporates transfer learning, it outperforms earlier efforts that did not take advantage of the simultaneous training of various spectral features. Table 14 gives a detailed analysis of the running time for training, validation and testing times for each datasets. It also highlights the time taken for the Fuzzy-rank based ensemble model to make its final classification.

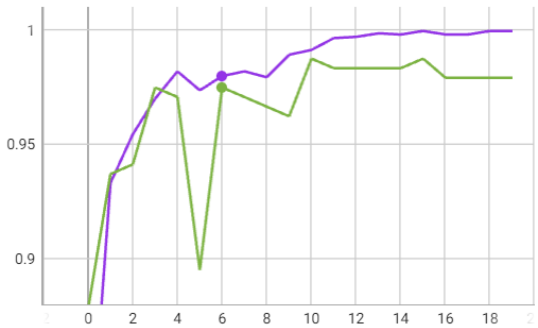


(a) EfficientNetV2

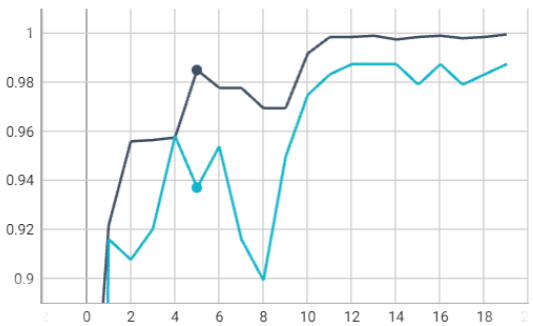


(b) ResNet18

Figure 10: Graph showing the CNN model training and validation curves on the IRMAS dataset using (a)EfficientNetV2 model and (b) ResNet18 model.



(a) EfficientNetV2



(b) ResNet18

Figure 11: Graphical representation of the variation of CNN model training and validation curves on the PCMIR dataset using (a)EfficientNetV2 model and (b) ResNet18 model.

Table 4: Performance of our MIC_FuzzyNET model compared to state-of-the-art MIC works for the PCMIR dataset.

Sl No.	Researcher	Methodology	Feature space used	Classification Accuracy
1	Mousavi et al. [36] (2019)	Muliti Layer Neural Network	Combined vector of MFCC, SC, SR, ZCR, EE Features	82.57%
2	**Sahoo et al. (2022)**	**MIC_FuzzyNET model**	**Mathematically synthesised 3 channel spectrogram using Mel, Constant Q Transform and Semitone spectrogram**	**98%**

Table 5: Performance of our proposed MIC_FuzzyNET model compared with state-of-the-art MIC works for the IRMAS dataset.

Sl No.	Researcher	Methodology	Feature space used	Classification Accuracy
1	Racharla et al. [43] (2019)	Support Vector machine	Mel-frequency cepstral coefficients	79%
2	Kim et al. [31] (2019)	Modified VGG-16 CNN model	Hilbert Spectrum Analysis–Intrinsic Mode Functions to generate spectrograms	80%
3	Hing and Settle [27] (2021)	Transfer Learning model	Mel Spectrogram	60.43%
4	**Sahoo et al. (2022)**	**MIC_FuzzyNET model**	**Mathematically synthesised 3 channel spectrogram using Mel, Constant Q Transform and Semitone spectrogram**	**88.36%**

C. PERFORMANCE OF ENSEMBLE MODEL

The categorisation scores obtained by the two transfer learning models in V-D are given fuzzy rankings by the ensemble model. Classification results from the previous transfer learning phase are saved for each sample in the test set. In this phase, we penalise the other class predictions

Table 6: Performance of the MIC_FuzzyNET model on the PCMIR dataset.

Class	Precision	Recall	F1 Score	Support
Kamancheh	1.0000	0.9730	0.9863	4097
Ney	0.9778	1.0000	0.9888	44
Santur	1.0000	0.9778	0.9888	45
Setar	0.9737	0.9737	0.9737	38
Tar	0.9787	0.9787	0.9787	47
Ud	0.9444	0.9714	0.9577	35
Accuracy	**0.9800**			

Table 7: Performance of the MIC_FuzzyNET model on the IRMAS dataset.

Class	Precision	Recall	F1 Score	Support
Cello	0.8571	0.9	0.8780	40
Flute	0.8444	0.8261	0.8352	46
Organ	0.8553	0.9420	0.8966	69
Piano	0.9315	0.9315	0.9315	73
Saxophone	0.9107	0.7969	0.8500	64
Accuracy	**0.8836**			

and award fuzzy ranks to the top k classes as explained in V-E. The ensemble model calculates the final prediction scores for the number of classes in each dataset. tables 6 to 7 display the ensemble's ultimate accuracy as well as the accuracy, recall, precision, and F1 score. Each dataset's musical instrument classes are listed in the class column. The six musical instrument classes included in the PCMIR dataset are as follows: Kamancheh, Ney, Santur, Sitar, Tar, and Ud, while the five classes used in the IRMAS dataset are cello, flute, organ, piano, and saxophone.

The classification performance of our ensemble model is displayed using the ROC curve, also known as the receiver operating characteristic curve. The ROC curve can be used for multi-class classification even though binary classification is its more popular application. The ground truth class is treated as a single label by the One versus All approach, whereas the other classes are treated as a group. The capacity of a model to distinguish between classes is measured by the ROC curve. The area under the ROC curve indicates the accuracy with which a class is correctly classified. The True Positive rate and False Positive rate are compared on the ROC curve.

The model is perfect if the ROC curve's area under it equals 1. It can accurately and completely distinguish between several classes. The lowest performing model is a ROC curve that has almost 0 area under the curve since it will predict incorrectly for each sample dataset. Figure 12 to Figure 13, respectively, presents the ROC curves produced by our suggested ensemble model for the IRMAS and PCMIR datasets.

$$TPR = \frac{TP}{TP + FP} \tag{14}$$

$$FPR = \frac{FP}{TN + FP} \tag{15}$$

The key takeaways of our framework is the reduction of errors of the individual models by the fuzzy rank ensemble

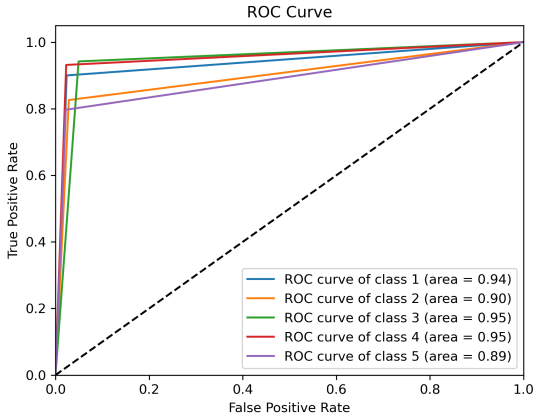


Figure 12: ROC curve created for the IRMAS dataset following ensemble classification

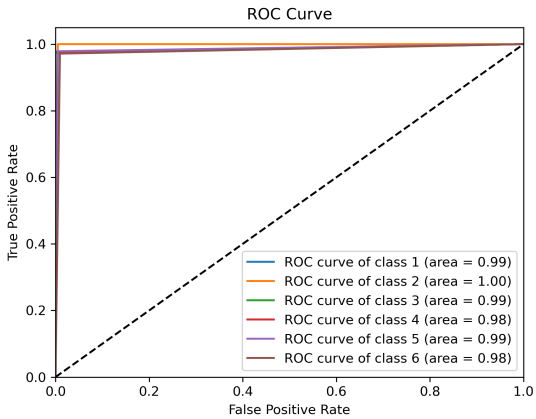


Figure 13: ROC curve created for the PCMIR dataset following ensemble classification

phase thereby ensuring more precise final classification of the musical instrument classes. Subsequently, our proposed framework does not use static weights as compared to traditional ensemble approaches and assign dynamic ranks to the classifiers, hence providing better results. The tables 4 to 5 provide a comparison of our model with the state-of-the-art frameworks with respect to performance, methodology and features used, corresponding to PCMIR and IRMAS datasets.

D. STATISTICAL SIGNIFICANCE TEST

We conducted a thorough investigation of our suggested model's performance on two benchmark MIC datasets in the preceding section and discovered that the proposed ensemble of the two base models surpasses each of them in terms of accuracy. We conducted a non-parametric statistical test[48] known as the Friedman test to specifically demonstrate the superiority and efficacy of our suggested ensemble model over the base models.

For each dataset under consideration, we chose 10 alternative subsets at random, each of which had 40 samples from the test data, with equal representation from each of

Phase	Model	IRMAS	PCMIR
Training and Validation	Resnet18	4 minutes	2 minutes 15 secs
Testing		4 secs	3 secs
Training and Validation	EfficientNetV2	5 minutes 45 secs	2 minutes 6 secs
Testing		3 secs	3 secs
Proposed StackNET ensemble model		8 secs	6 secs

Table 8: Execution time analysis of the different deep learning phases.

Degrees of freedom (*df*)	Significance level (α)							
	.99	.975	.95	.9	.1	.05	.025	.01
1	-------	0.001	0.004	0.016	2.706	3.841	5.024	6.635
2	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210
3	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345
4	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277
5	0.554	0.831	1.145	1.610	9.236	11.070	12.833	15.086
6	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812
7	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475
8	1.646	2.180	2.733	3.490	13.362	15.507	17.535	20.090
9	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666
10	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209
11	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.725

Figure 14: Chi-Square Right Tail Probability Table

Models	Mean ranks of classifier models for MIC datasets	
	PCMIR	IRMAS
Resnet	2.0	2.4
EfficientNetV2	2.6	2.4
MIC_FuzzyNET	1.4	1.2

Table 9: According to their accuracy on ten different subsets of each MIC dataset, mean ranks are awarded to the two base models as well as the proposed MIC_FuzzyNET model.

the class labels. Then, we determined the mean rank for each of the three models including the proposed ensemble across all 10 sample subsets using the formula given below, and the classification accuracies of each model over those samples are measured and ranked according to their accuracies.

$$R_y = \frac{1}{N}\sum_{x=1}^{N} r_y^x \tag{16}$$

where, for the x^{th} sample, r_y^x is the rank of the y^{th} classifier or model. Table 9 displays the classifiers' computed mean ranks.

All classifiers or models are the same, according to the null hypothesis (H_0). Their rank must therefore be equal. We determined the value of the Friedman statistic using the following formula[47] to support the null hypothesis.

$$x_F^2 = \frac{12N}{(k+1)k}\left[\sum_y R_y^2 - \frac{k(k+1)^2}{4}\right] \tag{17}$$

Dataset	Friedman Value
PCMIR	7.2
IRMAS	9.6

Table 10: Friedman statistic values calculated for each MIC dataset.

where, N is the total of sample datasets and k is the count of classifiers which in our proposed framework are 10 and 3 respectively. Table 10 displays the statistic's determined value for the two separate MIC datasets utilised in this research work.

The standard Friedman static value at significance level 0.05 is determined to be 5.991, which is much lower than the computed values, as shown in Table 10 as can be seen from the Chi-square table (shown in Figure **??**) at $k-1$ degrees of freedom, which in our case is 2, degrees of freedom (df). As a result, the null hypothesis can be rejected. The results obtained by the base models and suggested ensemble model are statistically significant, i.e., not equal, as may be inferred from the aforementioned statistical tests.

VII. CONCLUSION

This paper used transfer learning 2D-CNN models to create an ensemble learning-based framework for categorising musical instruments by synthesising a 3-channel spectrogram. The extraction of significant features from spectrograms of audio data has been demonstrated using models pre-trained on large image datasets, effectively converting audio processing and detection into a computer vision chal-

lenge. The proposed MIC_FuzzyNET model combines transfer learning, CNNs, and a fuzzy rank based ensemble technique using the Gompertz function and a 3-channel stacked spectrogram. Because the datasets used for training the deep CNNs are not large, transfer learning is a good option. The dynamic assignment of ranks to the classifiers allows predictions to be produced for new datasets without the need to initialise a new set of weights for the entire ensemble phase of the framework. The fuzzy ranking approach compensates for the faults made by each CNN classifier individually. Furthermore, our CNN models can collect characteristics from three spectrograms at the same time with no additional computational burden or time complexity by integrating three separate spectrograms into a single 3D matrix, comparable to RGB photographs. According to the experimental results, the MIC FuzzyNET model achieved state-of-the-art accuracies of 98 percent and 88.36 percent for both the PCMIR and IRMAS datasets. The MIC problem is tackled by combining transfer learning and ensemble approaches in a promising way.

There are few areas where our proposed MIC_FuzzyNET model can be improved which are as follows:

1) Better data augmentation techniques, such as voice conversion utilising a generative model[55] and speed perturbation, can improve the framework's generalisation.
2) Web scraping can be used to get more data from across the internet which can be used to create datasets with greater variety and instrument choices.
3) We can extend our proposed framework and make necessary modifications to not only classify but also segment different musical notes from polyphonic music and excerpts.

CONFLICT OF INTEREST

All the authors declare that there is no conflict of interest.

References

[1] S. Albawi, T. A. Mohammed, and S. Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.
[2] A. A. Alvarez and F. Gómez. Motivic pattern classification of music audio signals combining residual and lstm networks. *International Journal of Interactive Multimedia & Artificial Intelligence*, 6(6), 2021.
[3] U. Bagcı and E. Erzin. Automatic classification of musical genres using inter-genre similarity. *IEEE Signal Processing Letters*, 14(8): 521, 2007.
[4] D. Bhalke, C. Rao, and D. S. Bormane. Automatic musical instrument classification using fractional fourier transform based-mfcc features and counter propagation neural network. *Journal of Intelligent Information Systems*, 46(3):425–446, 2016.
[5] D. Bhattacharya, D. Sharma, W. Kim, M. F. Ijaz, and P. K. Singh. Ensem-har: An ensemble deep learning model for smartphone sensor-based human activity recognition for measurement of elderly health monitoring. *Biosensors*, 12(6):393, 2022.
[6] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera. A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In *ISMIR*, pages 559–564. Citeseer, 2012.
[7] H. Boyer, X. Amatriain, E. Batlle, X. Serra, et al. Towards instrument segmentation for music content description a critical review of instrument classification techniques. In *Proceedings of the 1st International Symposium on Music Information Retrieval; 2000 Oct 23-25; Plymouth, Massachusetts, USA.[Plymouth]: ISMIR; 2000. 9 p.* International Society for Music Information Retrieval (ISMIR), 2000.
[8] J. Chae, S.-H. Cho, J. Park, D.-W. Kim, and J. Lee. Toward a fair evaluation and analysis of feature selection for music tag classification. *IEEE Access*, 9:147717–147731, 2021.
[9] Y.-H. Cheng, P.-C. Chang, D.-M. Nguyen, and C.-N. Kuo. Automatic music genre classification based on crnn. *Engineering Letters*, 29(1), 2020.
[10] A. Chouiekh and E. H. I. E. Haj. Convnets for fraud detection analysis. *Procedia Computer Science*, 127:133–138, 2018.
[11] J. Colonna, T. Peet, C. A. Ferreira, A. M. Jorge, E. F. Gomes, and J. Gama. Automatic classification of anuran sounds using convolutional neural networks. In *Proceedings of the ninth international c* conference on computer science & software engineering*, pages 73–78, 2016.
[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. .
[13] D. Deutsch. Octave generalization and tune recognition. *Perception & Psychophysics*, 11(6):411–412, 1972.
[14] Q. Ding and N. Zhang. Classification of recorded musical instruments sounds based on neural networks. In *2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing*, pages 157–162. IEEE, 2007.
[15] Z. DING and L. DAI. A study of constant q transform in music signal analysis. *Technical Acoustics*, 2005.
[16] A. Eronen and A. Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 2, pages II753–II756. IEEE, 2000.
[17] S. Essid, G. Richard, and B. David. Musical instrument recognition by pairwise classification strategies. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1401–1412, 2006.
[18] R. Foulon, P. Roy, and F. Pachet. Automatic classification of guitar playing modes. In *International Symposium on Computer Music Multidisciplinary Research*, pages 58–71. Springer, 2013.
[19] F. Fuhrmann et al. *Automatic musical instrument recognition from polyphonic music audio signals*. PhD thesis, Universitat Pompeu Fabra, 2012.
[20] S. Goel, R. Pangasa, S. Dawn, and A. Arora. Audio acoustic features based tagging and comparative analysis of its classifications. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pages 1–5, 2018. .
[21] S. Goel, R. Pangasa, S. Dawn, and A. Arora. Audio acoustic features based tagging and comparative analysis of its classifications. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pages 1–5. IEEE, 2018.
[22] Y. Han, J. Kim, and K. Lee. Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):208–221, 2017. .
[23] T. Heittola, A. Klapuri, and T. Virtanen. Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In *ISMIR*, pages 327–332, 2009.
[24] P. Herrera, A. Yeterian, and F. Gouyon. Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques. In *International conference on music and artificial intelligence*, pages 69–80. Springer, 2002.
[25] P. Herrera-Boyer, G. Peeters, and S. Dubnov. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1): 3–21, 2003.
[26] P. Herrera-Boyer, A. Klapuri, and M. Davy. Automatic classification of pitched musical instrument sounds. In *Signal processing methods for music transcription*, pages 163–200. Springer, 2006.
[27] D. S. Hing and C. J. Settle. Detecting and classifying musical instruments with convolutional neural networks.
[28] C. Joder, S. Essid, and G. Richard. Temporal integration for audio classification with application to musical instrument classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 17 (1):174–186, 2009.
[29] I. Kaminsky and A. Materka. Automatic source identification of monophonic musical instrument sounds. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 1, pages 189–194. IEEE, 1995.

[30] P. Khunarsa, C. Lursinsap, and T. Raicharoen. Impulsive environment sound detection by neural classification of spectrogram and mel-frequency coefficient images. In *Advances in Neural Network Research and Applications*, pages 337–346. Springer, 2010.

[31] D. Kim, T. T. Sung, S. Cho, G. Lee, and C.-B. Sohn. A single predominant instrument recognition of polyphonic music using cnn-based timbre analysis. *International Journal of Engineering Technology*, 7(3. 34):590–593, 2018.

[32] B. Kostek and A. Czyzewski. Representing musical instrument sounds for their automatic classification. *Journal of the Audio Engineering Society*, 49(9):768–785, 2001.

[33] M. Kursa, W. Rudnicki, A. Wieczorkowska, E. Kubera, and A. Kubik-Komar. Musical instruments in random forest. In *International Symposium on Methodologies for Intelligent Systems*, pages 281–290. Springer, 2009.

[34] Y. LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20(5):14, 2015.

[35] J. Liu and L. Xie. Svm-based automatic classification of musical instruments. In *2010 International Conference on Intelligent Computation Technology and Automation*, volume 3, pages 669–673, 2010. .

[36] S. M. H. Mousavi, V. S. Prasath, and S. M. H. Mousavi. Persian classical music instrument recognition (pcmir) using a novel persian music database. In *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 122–130. IEEE, 2019.

[37] C.-R. Nadar, J. Abeßer, and S. Grollmisch. Towards cnn-based acoustic modeling of seventh chords for automatic chord recognition. In *International Conference on Sound and Music Computing. Málaga, Spain*, 2019.

[38] E. Otović, M. Njirjak, D. Jozinović, G. Mauša, A. Michelini, and I. Stajduhar. Intra-domain and cross-domain transfer learning for time series data—how transferable are the features? *Knowledge-Based Systems*, 239:107976, 2022.

[39] H. Papadopoulos and G. Peeters. Large-scale study of chord estimation algorithms based on chroma representation and hmm. In *2007 International Workshop on Content-Based Multimedia Indexing*, pages 53–60. IEEE, 2007.

[40] G. Peeters. Automatic classification of large musical instrument databases using hierarchical classifiers with inertia ratio maximization. In *Audio Engineering Society Convention 115*. Audio Engineering Society, 2003.

[41] J. Pons, O. Slizovskaia, R. Gong, E. Gómez, and X. Serra. Timbre analysis of music audio signals with convolutional neural networks. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2744–2748. IEEE, 2017.

[42] S. Prabavathy, V. Rathikarani, and P. Dhanalakshmi. Classification of musical instruments using svm and knn. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, pages 2278–3075, 2020.

[43] K. Racharla, V. Kumar, C. B. Jayant, A. Khairkar, and P. Harish. Predominant musical instrument classification based on spectral features. In *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 617–622. IEEE, 2020.

[44] C. Relkar and V. Tejwani. Musical instrument identification using machine learning. *Musical Instrument*, 2(9), 2019.

[45] K. K. Sahoo, I. Dutta, M. F. Ijaz, M. Woźniak, and P. K. Singh. Tlefuzzynet: fuzzy rank-based ensemble of transfer learning models for emotion recognition from human speeches. *IEEE Access*, 9: 166518–166530, 2021.

[46] S. Shetty and S. Hegde. Automatic classification of carnatic music instruments using mfcc and lpc. In *Data Management, Analytics and Innovation*, pages 463–474. Springer, 2020.

[47] P. K. Singh, R. Sarkar, and M. Nasipuri. Statistical validation of multiple classifiers over multiple datasets in the field of pattern recognition. *International Journal of Applied Pattern Recognition*, 2(1):1–23, 2015.

[48] P. K. Singh, R. Sarkar, and M. Nasipuri. Significance of non-parametric statistical tests for comparison of classifiers over multiple datasets. *International Journal of Computing Science and Mathematics*, 7(5):410–442, 2016.

[49] A. Solanki and S. Pandey. Music instrument recognition using deep convolutional neural networks. *International Journal of Information Technology*, pages 1–10, 2019.

[50] V. M. Souza, G. E. Batista, and N. E. Souza-Filho. Automatic classification of drum sounds with indefinite pitch. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2015.

[51] M. Taenzer, J. Abeßer, S. I. Mimilakis, C. Weiß, M. Müller, H. Lukashevich, and I. Fraunhofer. Investigating cnn-based instrument family recognition for western classical music recordings. In *ISMIR*, pages 612–619, 2019.

[52] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[53] S. E. Trehub, A. J. Cohen, L. A. Thorpe, and B. A. Morrongiello. Development of the perception of musical relations: Semitone and diatonic structure. *Journal of Experimental Psychology: Human Perception and Performance*, 12(3):295, 1986.

[54] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5): 293–302, 2002. .

[55] J. Wang, S. Kim, and Y. Lee. Speech augmentation using wavenet in speech recognition. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6770–6774, 2019. .

[56] Y. Xiong, H. Liu, S. Gupta, B. Akin, G. Bender, Y. Wang, P.-J. Kindermans, M. Tan, V. Singh, and B. Chen. Mobiledets: Searching for object detection architectures for mobile accelerators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3825–3834, 2021.

KARAM KUMAR SAHOO (email: karamsahoo@gmail.com) is currently pursuing his Bachelor's degree in Computer Science and Engineering from National Institute of Technology, Durgapur, India. His research interests include Deep Learning, Computer Vision and he is also a full stack MERN developer.

RIDAM HAZRA (email: rhazra0602@gmail.com) is currently pursuing his Bachelor's degree in Computer Science and Engineering from National Institute of Technology, Durgapur, India. His research interests are mainly Deep Learning, Artificial Intelligence and Natural Language Processing. Other than those, he is also interested in Python development and Cyber Security. He is student member of IEEE and IEEE Computer Society.

MUHAMMAD FAZAL IJAZ (email: fazal@sejong.ac.kr) received his B.Eng. degree in Industrial Engineering and Management from University of the Punjab, Lahore, Pakistan, in 2011, and Dr. Eng. degree in Industrial and Systems Engineering from Dongguk University, Seoul, South Korea, in 2019. From 2019 to 2020, he worked as an Assistant Professor in Department of Industrial and Systems Engineering, Dongguk University, Seoul, South Korea. Currently, he is working as an Assistant Professor in Department of Intelligent Mechatronics Engineering, Sejong University, Seoul, Korea. He has published numerous research articles in several international peer-reviewed journals, including IEEE Access, Sensors, Symmetry, Journal of Food Engineering, Applied Sciences, Asia Pacific Journal of Marketing and Logistics, and Sustainability. His research interests include Machine learning, Blockchain, Healthcare Engineering, Internet of Things, Supply chain management, Big data, and Data mining.

SEONGKI KIM (email: skkim9226@smu.ac.kr) is an Assistant Professor at Sangmyung University. He received his PhD degree in computer science and engineering from Seoul National University in 2009. He researched and developed software for the GPU, the GPGPU and dynamic voltage and frequency scaling (DVFS) at the Samsung Electronics from 2009 to 2014. He also worked at the Ewha Womans University and Keimyung University from 2014 to 2020. His current research interests include the areas of graphics, virtual/ augmented reality, an artificial intelligence, an algorithm optimization through the GPU, and high-performance computing with CPU and GPU.

PAWAN KUMAR SINGH (email: pawankrsingh@ieee.org) received his B. Tech degree in Information Technology from West Bengal University of Technology in 2010. He received his M. Tech in Computer Science and Engineering and Ph.D. (Engineering) degrees from Jadavpur University (J.U.) in 2013 and 2018 respectively. He also received the RUSA 2.0 fellowship for pursuing his post-doctoral research in J.U. in 2019. He is currently working as an Assistant Professor in the Department of Information Technology in J.U. He has published more than 100 research papers in peer-reviewed journals and international conferences. He also serves as Editorial Board Member, Reviewer, and Technical Program Committee Member for a number of IEEE and Springer journals and conferences. His areas of current research interest are Computer Vision, Pattern Recognition, Handwritten Document Analysis, Image & Video Processing, Feature Optimization, Machine Learning, Deep Learning and Artificial Intelligence. He is a senior member of the IEEE (U.S.A.), member of The Institution of Engineers (India) and Association for Computing Machinery (ACM) as well as a life member of the Indian Society for Technical Education (ISTE, New Delhi) and Computer Society of India (CSI).

MUFTI MAHMUD (email: mufti.mahmud@ntu.ac.uk) received the Ph.D. degree in information engineering from the University of Padova, Italy, in 2011. He is currently working as an Associate Professor of computer science with Nottingham Trent University, U.K. With over 18 years of experience in the industry and academia in India, Bangladesh, Italy, Belgium, and U.K. He is an Expert in computational intelligence, applied data analysis, and big data technologies with a keen focus on healthcare applications. As of July 2021, he has published over 170 peer-reviewed articles and papers in leading journals and conferences, and (co-)edited five volumes and many journal special issues on those domains. As an active researcher, he has secured research grants totaling > £3.3 million and has supervised 50 research students (Ph.D., master's, and bachelor's). He is also a Senior Member of ACM, a Professional Member of the British Computer Society, and a fellow of the Higher Education Academy, U.K. He was a recipient of the Vice-Chancellor's Outstanding Research Award 2020 at the NTU and the Marie-Curie Postdoctoral Fellowship in 2013. During 2020–2021, he has been serving as the Vice-Chair for the Intelligent System Application and Brain Informatics Technical Committees of the IEEE Computational Intelligence Society (CIS), a member of the IEEE CIS Task Force on Intelligence Systems for Health, an Advisor of the IEEE R8 Humanitarian Activities Subcommittee, the Publications Chair of the IEEE U.K., and Ireland Industry Applications Chapter, and the Project Liaison Officer of the IEEE U.K., and Ireland SIGHT Committee. He has also served as the Co-ordinating Chair for the local organization of the IEEE-WCCI2020; the General Chair of BI2020, 2021 and AII2021; and the Program Chair of IEEE-CICARE2020 and 2021. He serves as a Section Editor (Big Data Analytics) for the Cognitive Computation journal, an Associate Editor for the Frontiers in Neuroscience, and a Regional Editor (Europe) for the Brain Informatics journal.