2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)

Automatic Music Generation System based on RNN Architecture

Sandeep Kumar
Department of Computer Science & Engineering
KoneruLakshmaiah Educational Foundation
Vaddeswaram, Andhra Pradesh, India
er.sandeepsahratia@gmail.com

KeerthiGudiseva
Department of Computer Science & Engineering
KoneruLakshmaiah Educational Foundation
Vaddeswaram, Andhra Pradesh, India
keerthigudiseva0611@gmail.com

AallaIswarya
Department of Computer Science & Engineering
KoneruLakshmaiah Educational Foundation
Vaddeswaram, Andhra Pradesh, India
aallaiswarya@gmail.com

Shilpa Rani
Department of Computer Science & Engineering
Neil Gogte Institute of Technology
Hyderabad, India
shilpachoudhary1987@gmail.com

KMVV Prasad
Department of ECE
Symbiosis Institute of Technology
Symbiosis International Deemed University
Pune, India
mvvprasad.kantipudi@gmail.com

Yogesh Kumar Sharma
Department of Computer Science & Engineering
KoneruLakshmaiah Educational Foundation
Vaddeswaram, Andhra Pradesh, India
dr.sharmayogeshkumar@gmail.com

2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS) | 978-1-6654-7657-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICTACS56270.2022.9988652

***Abstract*—Musicians or artists build on what has been generated utilizing the system and bring their original work. Music composition is an exciting topic that helps us to realize the composer's creativity. With the rapid improvement of the era, the form of music has ended up extra various and unfolds faster. The cost of making music, on the other hand, remains very high. Deep learning should really be capable of producing music that sounds like it was made by a person if it has sufficient data and the right algorithm. The purpose of this research is to set up a track-based and machine-learning-based device that can automatically put together songs. The device is composed of a set of piano MIDI records from the MAESTRO dataset that are used to build song segments. Fully connected and convolutional layers take advantage of the rich features in the frequency area to improve the quality of the music that is made.**

Keywords— Music generator, AI Music Composition, Automated Composition, Deep Learning.

I. Introduction

These days, the growing amusement market has positioned a better demand for songs [1]. The quality tune is essential for video making, online game making, video editing, entertainment, or even public places. Whereas, now and then, finding a suitable list of music may be challenging and steeply priced [2-4]. This could be resolved using automatic, deep-studying based on totally tune making. By the concept of Recurrent Neural Network (RNN), computers can study the patterns from present song portions and convert them into a possible map [5-7]. The conventional definition of music is much like what is defined as a set of notes that correlate, harmonize, and specific emotions. The most classical track becomes composed of sophisticated shapes and arrangements to deliver ideas and feelings [8]. At the same time, the current tune has advanced to serve as more of an entertainment-time amusement with a more excellent rhythmic, patterned, and much less complicated fashion which may be found out and generated with machine learning [9-11].

A. Motivation

Our job is to build a classifier using some music data that already exists. The model has to learn how to recognise patterns in music. Once the machine knows the above, the prototype should have been able to make new music [12]. To make new music, you must first understand how music is put together. We don't expect our model to make new music that is as good as that made by professionals, but we do expect it to make music that is pleasant to listen to. Machine learning is when computers can do tasks without being told how to do them, but they still think and act like machines [13-15]. Their ability to do some complicated things, like get features of an image or video, is still a long way behind that of humans.

Deep learning models give an exceptionally complicated approach to machine learning and are designed to address these challenges because they have been meticulously modelled after the human brain [16]. Using intricate, multi-layered "deep neural networks," data is conveyed between nodes (like neurons) that are closely linked. The outcome is a non-linear transformation of the data that gradually abstracts. Even while it requires a lot of data to "feed" and build such a system, it can start providing results almost quickly, and once the programmes are in place, there is no need for human interaction. There are various models to calculate in deep learning. The natural choice will be RNN or RNN variations since our music are a series of characters [17–19]. An ANN that processes sequential or time series data is known as a RNN as shown in Fig.1. These deep learning algorithms can be found in well-known apps like Siri, voice search, and Google Translate. They are frequently used for ordinal or temporal problems like language translation, natural language processing (NLP), speech recognition, and image captioning. Recurrent neural networks, like feedforward and convolutional neural networks learn from training data

(CNNs). They stand out due to their memory, which enables them to use information from earlier inputs to influence current input and output. While traditional deep neural networks assume that inputs and outputs are independent of one another, recurrent neural networks' output is dependent on the sequence's fundamental components [20–22]. In this research, a straightforward RNN is used to produce musical notes. Using a selection of piano MIDI files from the MAESTRO dataset, we can learn how to train a model. Our model may be trained to predict the next note in a series of notes if given the previous notes [23–25]. By calling the model more than once, we can produce a longer series of notes.

We reviewed the literature in the second section. We discussed the architecture and approach in the third section. We described the findings of benchmark datasets in section 4, and in the following subsections, we evaluated by contrasting the existing methods with the suggested method.

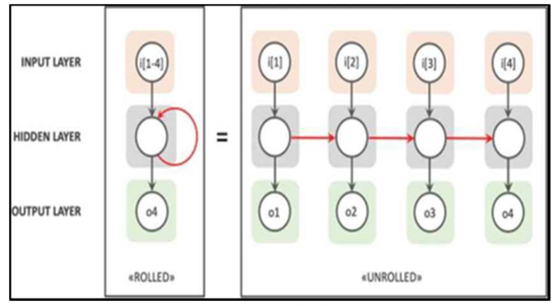

Figure 1. RNN Block Diagram.

II. LITERATURE SURVEY

This paper shows how to use a simple RNN to make musical notes. From the MAESTRO dataset, we can also learn how to train a model using a group of piano MIDI files. Various authors have worked in this field and used different methods and techniques. Table I displays a comparison of various approaches.

TABLE I. COMPARATIVE ANALYSIS OF DIFFERENT METHODS

S.NO	Author Name& Year of Publication	Methodology	Database	Remarks
1	Chih-Fang Huang et al., 2020	CVAE-GAN Model	Self database	Scoring Statistics Table for the Generated Music: A B C D Average 2.52 1.52 2.27 4.10 SD1.24 0.78 1.06 0.90 Variation1.53 0.61 1.12 0.81 Mode3 1 3 4 Median3 1 2 4
2	KeigoSakurai et al. 2021	Dimensional reduction	Self database	RESULTS FOR HIT RATE: AM151630MSNP30MP .07 .31 .29 .24 .24 .14 .11 .32 .29 .21 .19 .13 .09 .32 .26 .25∗ .28∗ .111 .12 .35 .29 .24 .25 .140
3	Seiji Ueda et al., 2018	Random walk model	Self database	Effectiveness: The approaches' effectiveness was the initial focus. Calculation time: Four-fold cross-validation was used to compare the average computation times for each method. Figures 9 and 10 show the results for faster computation times using RW-hybrid for the 30musicS dataset after two iterations.
4	CsabaSulyok, 2018	Constand and adaptive	Self database	Depicts the averaged mean and maximum values for each of the six configurations across the 20 test runs. According to the findings, adaptive operator configuration only offers slight advantages in unique circumstances.
5	ManyaSingh et al. 2021	Sequential and Convolutional neural network	Self database	Sequential neural network: Largest data set: 1000 Distribution of data sets: 80:20 28 data points were collected from the input layer for each song over the course of 30 seconds for a total of 1000 songs. Four layers total (3) Network of convolutional neurons: 2000 rows in the dataset Distribution of the dataset 70:20:10 50 pixels per second square spectrogram slices in the input layer 6 layers (4 Conv max pools and 1 completely)
6	Yi Zhao et al. 2020	WaveNet, WaveGlow, and the neural-source-filter (NSF) model	NSynth and MAESTRO Dataset	The findings imply that pre-training, even on speech data, may be able to enhance music sound production ability. Of the three synthesisers we looked at, NSF performed the best in scenarios involving fine-tuning and training from start. WaveGlow, meanwhile, displayed excellent promise for zero-shot cross-domain adaptability.
7	Tao Wang et al. 2020	RVAE-GAN neural network	Self database	Objective Evaluation: The average probability of a beat, abbreviated as APB, the standard deviation of the beat length, the standard deviation of the bar length. Subjective assessment: The musical tempo has some benefits, based on the score.
8	Brandon Royal, et al. 2020	preprocessing and reconstruction technique hash based	Nottingham dataset	Method Average (song 1) 4.14 Medeot et al. (song1)2.00 Johnson(song 1) 2.86 Chen et al(song 1) 3.21 Boulanger(song 1) 2.71

9	Sarthak Agarwal et al., 2018	PRECON-LSTM	Nottingham Music Database	ScoreMeanMean Realness Artif3.120 2.747 Gen Hum3.613 3.516 Comp
10	AdvaitMaduskar et al. 2020	Autoregressive GAN model	Bach's musical symphonies dataset	Outlined a generation model for note sequence generation using the GAN framework. Additionally, the deep convolution neural network was employed and tuned based on the properties of musical notes; this optimization approach allows the convolution network to focus on learning musical aspects and speed up tests.
11	Tianyu Jiang et al., 2019	Bidirectional LSTM network	Classical Piano Dataset	Bidirectional LSTM network was presented to produce harmonic music. Through the bidirectional learning of context information from notes on both horizontal and vertical levels, the model enhanced the quality of the music that was generated. In order to speed up the model optimization process, we restructure the loss function to prevent producing a large number of useless outcomes.
12	Hongyu Chen et al., 2019	DCGAN model	The Lakh MIDI Dataset	A GAN-based model that was trained on a dataset of Bach's orchestral symphonies produced the desired outcomes. The pitch was the conditional characteristic in use here. The generator develops its ability to depict various instrument timbres by employing pitch.
13	Muhammad Nadeem et al., 2019	LSTM	Nottingham MusicDatabase	The majority of participants enjoyed listening to the music created by the proposed musical data structure and RNN architecture. Participants seemed to enjoy the music this system produced, with 67% saying they did.
14	Belinda M. Dungan et al. 2020	Generative model	GAN-based Midinet	PERFORMANCE OF THE MELODY GENERATION MODELS: RCtimeavg DT DT 0.0679 0.7436 KNN KNN 0.0156 0.6524 Midinet 201.00 0.9682
15	Running Lang et al. 2020	SSCL model, which clusters music self-similarity matrix and then LSTM network	Self database	Epoch-100 True accSSCLaccDeepJacc 0.396 0.234 0.226 0.410 0.204 0.194 0.419 0.234 0.224 Epoch-500 True accSSCLaccDeepJacc 0.642 0.126 0.112 0.666 0.126 0.112 0.624 0.137 0.127

III. Proposed Methodology

In the proposed method, a single MIDI file is considered an input file and then the different structure of the music is generated. Fig. 2 depicts the proposed method's organizational structure. To begin with, use pretty midi to parse a single MIDI record and examine the note structure. Three variables will be used to denote the model's pitch, step, and length [26-28]. The important key of the sound as a MIDI word range is known as the pitch.

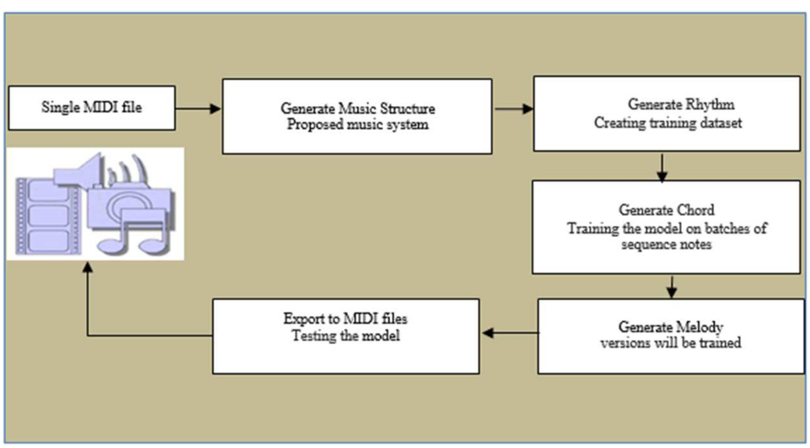

Figure 2. Block Diagram of Proposed System

The step takes the notes from the pattern MIDI file based on the amount of time since the preceding note or song's beginning. The sample copy is shown in Fig. 3. Plot the note pitch, beginning, and end over the song's duration in order to visualise the musical composition (i.e., Piano roll). Begin with the basic 100 notes [29-30]. In the proposed system, we used RNN to generate musical notes. The detailed description of RNN is given below and the architectural diagram is shown in Fig. 4.

A. *Recurrent Neural Network (RNN)*

For simulating sequence data, neural networks belonging to the RNN class are helpful. Feedforward networks' offspring, RNNs, exhibit behavior akin to that of human brains. Since the same set of weights are used again over a differential graph-like structure, this is why they are referred to as recurrent. Natural Language Processing is a great application for RNN.

	pitch	start	end	step	duration
0	75	1.081250	1.212500	0.000000	0.131250
1	63	1.082292	1.153125	0.001042	0.070833
2	68	1.083333	1.169792	0.001042	0.086458
3	60	1.090625	1.155208	0.007292	0.064583
4	48	1.094792	1.162500	0.004167	0.067708

Figure 3. Sample file

The potential to send information over the years-steps makes the RNN distinctive from the opposite participants of the own family of the neural community. The working style of RNN chooses sequential information as input and gives information in sequential/order form instead of accepting input and producing an outcome. As the song itself is sequential facts, it can be modeled using a sequential gadget getting to know the version, including the recurrent neural community. The outcome produced by the layers inside the neural networks again enters into the layer as the input.It immediately computes the value of the layer. Because of this, the neural network learns itself based totally on present data or information and previously collected information together. Recurrent Neural Networks contain many copies of the neural community linked and working in a series. This RNN model can help learn the track collection and generate the series of track records. The audio facts are also in sequence form or sequential manner. It could be treated as a signal that modifies timeand the time series facts where statistics points are accumulated in a sequencewith time values.

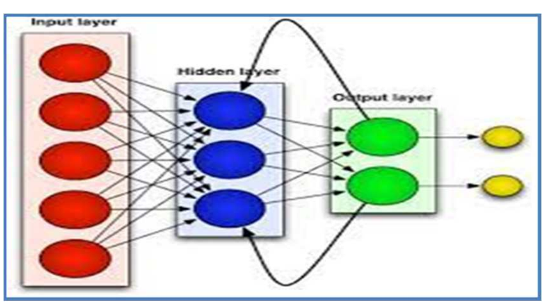

Figure 4. RNN Structure

IV. Result And Discussion

A. *Dataset*

More than 200 hours of paired audio and MIDI recordings from the International Piano-e-first Competition's ten years are included in the MAESTRO dataset. With a 3ms precision, individual musical compositions are extracted from audio and MIDI files and identified with the composer, title, and performance year. At least as excellent as CD quality is uncompressed audio (44.1–48 kHz 16-bit PCM stereo)

B. *Training and testing of the Dataset*

By removing notes from the MIDI documents, the training dataset is produced. We test with fewer, but higher-quality, files at first before using more. On collections of note sequences, we will train the model. Due to the input capabilities, each example will be composed of a collection of notes. In this manner, the taught version anticipates the subsequent item in a series. Set the length of each example's sequence. Test out unusual lengths (such as 50, 100, or 150) to find which one is best for the content. The vocabulary size is set to 128 to represent all of the pitches that Pretty Midi supports. Make the model, then train it. Each observed variable will have a separate output in the version's three outputs. You will employ a unique loss feature for pitch and time that is based on inferred squared errors that motivate the model to provide non-negative values. Values for the hyper-parameters are displayed in Fig. 5.

```
Model: "model"
________________________________________________________________________________
 Layer (type)                 Output Shape         Param #    Connected to
================================================================================
 input_1 (InputLayer)         [(None, 25, 3)]      0          []

 lstm (LSTM)                  (None, 128)          67584      ['input_1[0][0]']

 duration (Dense)             (None, 1)            129        ['lstm[0][0]']

 pitch (Dense)                (None, 128)          16512      ['lstm[0][0]']

 step (Dense)                 (None, 1)            129        ['lstm[0][0]']

================================================================================
Total params: 84,354
Trainable params: 84,354
Non-trainable params: 0
________________________________________________________________________________
```

Figure 5. Hyper-parameters of the model

We can see that the pitch loss is substantially larger than the duration and step losses after training and collecting the data. We also need to be aware that the pitching loss is the sum of all previous losses, which is how the loss is computed. Now pitch loss is the dominant factor.

C. *Epochs*

A hyper-parameter that regulates how frequently the learning algorithm iterates through the training dataset is the number of epochs. The underlying model parameters have been updated once every epoch for each sample in the training dataset.

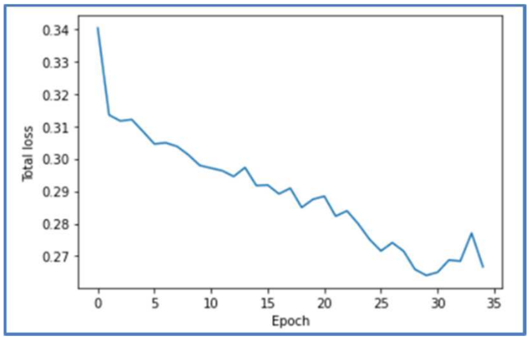

Figure 6. Graph of epoch history

In this model, we have used 50 epochs. For our proposed model music generation, after it has been trained, we limited it up to 50 epochs because we are getting a consistent result as the output. The results are shown in Fig. 6.

D. Learning Rate

The learning rate is a hyper-parameter that determines how much to alter the model each time the model weights are updated in response to the predicted error. A high learning rate speeds up the learning process but results in a less-than-ideal final set of weights. Although training will take significantly longer, a slower learning rate might enable the model to learn a more ideal or globally ideal set of weights.

E. Pitch

Pitch is a perceptual attribute of sounds that allows them to be ordered on a frequency-related scale, or, to put it another way, the pitch is the quality that allows humans to judge sounds as "higher" or "lower" in the sense of musical melodies. The audio representation of pitch, steps and duration is shown in Fig. 7-9.

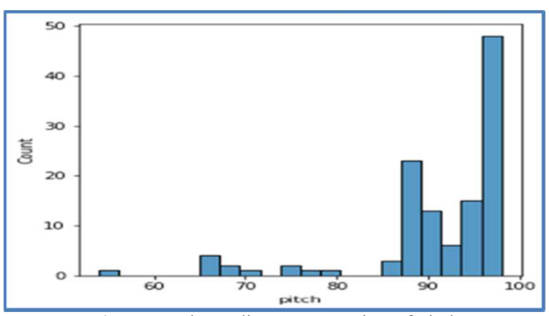

Figure 7. The audio representation of pitch

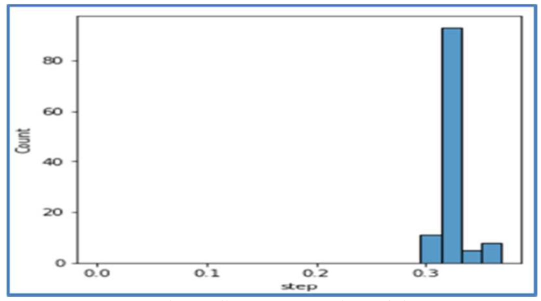

Figure 8. The audio representation of the steps-

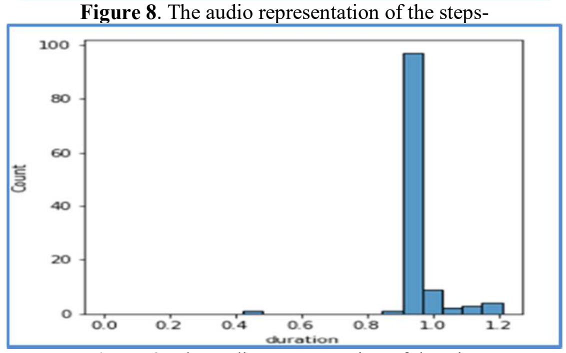

Figure 9. The audio representation of duration

The variation in pitch between two successive notes in a musical scale is known as a step or conjunct motion. Put another way, it is the space between two successive scale degrees. A skip, or disjunct motion, is any longer interval. The duration of a note, phrase, section, or composition refers to how long or short it lasts. Duration is the length of time that a pitch or tone is audible. A symphony can run for an hour or longer than a note, which can only last a fraction of a second.

F. Quantitative Analysis

We are now seeing the various categories like pitch, step, duration, start, and end over different seed values. We get different values for each iteration. In order to show the variations, we have chosen seed values from 30 and last till 40, and the results are shown in Fig. 10. Looking at those variations, we can say that every time we run, we get unique music.

SEED	PITCH	STEP	DURATION	START	END
30	26	0.313976	0.436626	0.313976	0.750601
31	69	0.417522	0.428845	0.417522	0.876368
32	66	0.387033	0.540044	0.387033	0.927077
33	76	0.320829	0.41971	0.320829	0.740000
34	85	0.511165	0.979517	0.511165	1.490681
35	100	0.252784	0.545995	0.252784	0.798779
36	21	0.148335	0.564492	0.148335	0.712827
37	76	0.186018	0.523883	0.186018	0.709901
38	67	0.267205	0.555200	0.267205	0.822405
39	75	0.083971	0.312730	0.083971	0.396701
40	60	0.274895	0.758532	0.274895	1.033428

Figure 10. Results on different seed values

The graph for the sample is constructed using the softmax distribution of pitch notes produced by the model. It generates continuous musical sequences because it always selects the note with the highest probability rather than just selecting the note with the highest probability. Now we can make some notes.

Additionally, we can experiment with next notes' temperature settings and starting order. The final result is given in Fig. 11. We can also download the generated audio files, visualize the generated notes, and check the step, pitch, and duration distribution by plotting.

	pitch	step	duration	start	end
0	101	0.282856	0.132292	0.282856	0.415148
1	50	0.026528	0.729401	0.309384	1.038786
2	95	0.091744	0.584931	0.401128	0.986060
3	84	0.117073	0.671026	0.518201	1.189227
4	87	0.102534	0.608209	0.620735	1.228944
5	95	0.110170	0.570209	0.730905	1.301114
6	95	0.110305	0.580784	0.841210	1.421994
7	99	0.108037	0.591342	0.949247	1.540589
8	95	0.109995	0.601920	1.059242	1.661161
9	72	0.109707	0.815779	1.168949	1.984727

Figure 11. Graph of Generated Notes

We can see changes in the distribution of the notice variables in the charts. The version prefers to provide a similar sequence of outputs to reduce the pitching loss because there is a feedback loop between its inputs and outputs. When employing Mean Squared Error loss. We may add unpredictability to pitch by raising the temperature in predict next note.

V. Conclusion

Production of music continues to be quite expensive. Deep learning should be able to produce music with a human-like sound with enough data and the right algorithm. The goal of this work is to build a machine learning and track-based automatic composition device. The musical instrument comprises of a set of piano MIDI files from the MAESTRO dataset that serve as the basis for the song portions. Convolutional layers that are fully related capture powerful abilities inside the frequency range and improve the quality of the track produced.

References

[1] Huang, Chih-Fang, and Cheng-Yuan Huang, "Emotion-based AI music generation system with CVAE-GAN," In IEEE Eurasia Conference on IoT, Communication and Engineering (ECICE), pp. 220-222, 2020.

[2] Sakurai, Keigo, Ren Togo, Takahiro Ogawa, and Miki Haseyama, "Music playlist generation based on graph exploration using reinforcement learning," In 3rd IEEE Global Conference on Life Sciences and Technologies (LifeTech), pp. 53-54, 2021.

[3] Ueda, Seiji, Atsushi Keyaki, and Jun Miyazaki, "A contextual random walk model for automated playlist generation," In 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 367-374, 2018.

[4] Sulyok, Csaba, "Genetic Operators in Evolutionary Music Composition," In 20th IEEE International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp. 253-259, 2018.

[5] Singh, Manya, Sanjeev Kumar Jha, Bhupendra Singh, and Banafsha Rajput, "Deep learning neural networks for music information retrieval," In IEEE International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), pp. 500-503, 2021.

[6] Zhao, Yi, Xin Wang, LauriJuvela, and Junichi Yamagishi, "Transferring neural speech waveform synthesizers to musical instrument sounds generation," In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6269-6273, 2020.

[7] Wang, Tao, Junzhe Liu, Cong Jin, Jianguang Li, and Shihao Ma, "An intelligent music generation based on VariationalAutoencoder," In International Conference on Culture-oriented Science & Technology (ICCST), pp. 394-398, 2020.

[8] Royal, Brandon, Kien Hua, and Brenton Zhang, "Deep composer: Deep neural hashing and retrieval approach to automatic music generation," In IEEE International Conference on Multimedia and Expo (ICME), pp. 1-6, 2020.

[9] Agarwal, Sarthak, VaibhavSaxena, VaibhavSingal, and Swati Aggarwal, "Lstm based music generation with dataset preprocessing and reconstruction techniques," In IEEE Symposium Series on Computational Intelligence (SSCI), pp. 455-462, 2018.

[10] Chen, Hongyu, Qinyin Xiao, and Xueyuan Yin, "Generating music algorithm with deep convolutional generative adversarial networks," In 2nd IEEE International Conference on Electronics Technology (ICET), pp. 576-580, 2019.

[11] Jiang, Tianyu, Qinyin Xiao, and Xueyuan Yin, "Music generation using the recurrent bidirectional network," In 2nd IEEE International Conference on Electronics Technology (ICET), pp. 564-569, 2019.

[12] Maduskar, Advait, AniketLadukar, Shubhankar Gore, and NehaPatwari, "Music Generation using Deep Generative Modelling." In IEEE International Conference on Convergence to Digital World-Quo Vadis (ICCDW), pp. 1-4, 2020.

[13] Waitt, Gordon, Theresa Harada, and Michelle Duffy, "'Let us have some music: Sound, gender and car mobility," Mobilities, vol. 12, no. 3, pp. 324-342, 2017.

[14] Dungan, Belinda M., and Proceso L. Fernandez, "Next Bar Predictor: An Architecture in Automated Music Generation," In IEEE International Conference on Communication and Signal Processing (ICCSP), pp. 109-113, 2020.

[15] Lang, Runnan, Songsong Wu, Songhao Zhu, and Zuoyong Li, "SSCL: Music Generation in Long-term with Cluster Learning," In 4th IEEE Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 77-81, 2020.

[16] Kumar S, Arpit Jain, Ambuj Kumar Agarwal, Shilpa Rani, and AnshuGhimire, "Object-Based Image Retrieval Using the U-Net-Based Neural Network," Computational Intelligence and Neuroscience, 2021.

[17] Kumar S, Shilpa Rani, Arpit Jain, Chaman Verma, Maria SimonaRaboaca, ZoltánIllés and BogdanConstantinNeagu, "Face Spoofing, Age, Gender and Facial Expression Recognition Using Advance Neural Network Architecture-Based Biometric System, " Sensor Journal, vol. 22, no. 14, pp. 5160-5184, 2022.

[18] Kumar S, Arpit Jain, Shilpa Rani, HammamAlshazly, Sahar Ahmed Idris, and Sami Bourouis, "Deep Neural Network Based Vehicle Detection and Classification of Aerial Images," Intelligent automation and soft computing, vol. 34, no. 1, pp. 119-131, 2022.

[19] S. Kumar, Arpit Jain, Shilpa Rani, DeepikaGhai, SwathiAchampeta, and P. Raja, "Enhanced SBIR based Re-Ranking and Relevance Feedback," in 10th International Conference on System Modeling & Advancement in Research Trends (SMART), pp. 7-12. IEEE, 2021.

[20] Harshitha, Guyana, S. Kumar, Shilpa Rani, and Arpit Jain, "Cotton disease detection based on deep learning techniques," in 4th Smart Cities Symposium (SCS 2021), vol. 2021, pp. 496-501, 2021.

[21] Sandeep Kumar, Arpit Jain, Anand Prakash Shukla, Satyendra Singh, Rohit Raja, Shilpa Rani, G. Harshitha, Mohammed A. AlZain, MehediMasud, "A Comparative Analysis of Machine Learning Algorithms for Detection of Organic and Non-Organic Cotton

Diseases, " Mathematical Problems in Engineering, Hindawi Journal Publication, vol. 21, no. 1, pp. 1-18, 2021.

[22] Sandeep Kumar, Shailu, Arpit Jain, “Enhanced Method of Object Tracing Using Extended Kalman Filter via Binary Search Algorithm” in Journal of Information Technology and Management, vol. 21, no. 2, pp. 1-19, 2021.

[23] Rani Shilpa, Sandeep Kumar, DeepikaGhai, and K. M. V. V. Prasad, “Automatic Detection of Brain Tumor from CT and MRI Images using Wireframe model and 3D Alex-Net,” in 2022 International Conference on Decision Aid Sciences and Applications (DASA), pp. 1132-1138, 2022.

[24] Rani Shilpa, KamleshLakhwani, and Sandeep Kumar, "Three-dimensional wireframe model of medical and complex images using cellular logic array processing techniques," in International Conference on Soft Computing and Pattern Recognition, pp. 196-207. Springer, Cham, 2020.

[25] Rani Shilpa, DeepikaGhai, and Sandeep Kumar, "Reconstruction of a wireframe model of complex images using syntactic pattern recognition," pp.8-13, 2021.

[26] Shukla A. P, “Training cellular automata for image edge detection,” Romanian Journal of Information Science and Technology, vol. 19, no. 4, pp. 338-359, 2016.

[27] Verma, P., Shukla, N., & Shukla, A. P., “Techniques of sarcasm detection: A review. In 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pp. 968-972, 2021.

[28] Sharma, P., & Shukla, A. P., “A review on brain tumor segmentation and classification for MRI images. In 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pp. 963-967, 2021.

[29] Misra, N. R., Kumar, S., & Jain, A, “A review on E-waste: Fostering the need for green electronics,” In International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pp. 1032-1036, 2021.

[30] Bhardwaj, A., Kaur, S., Shukla, A. P., & Shukla, M. K., “A Novel Method for Despeckling of Ultrasound Images Using Cellular Automata-Based Despeckling Filter,” International Journal of E-Health and Medical Communications (IJEHMC), vol. 12, no.5, pp. 16-35, 2021.