International Journal of Emerging Technologies in Engineering Research (IJETER)
Volume 9, Issue 1, January (2021)

Transfer Learning Models Based Environment Audio Classification

Jasmine Chhikara
Department of Electronics and Communication Engineering, Maharaja Surajmal Institute of Technology, Delhi, India
jasmine.chhikara@gmail.com

ABSTRACT

This work proposes a transfer learning model to train Convolutional Neural Networks (CNNs) for environmental audio classification. The model uses cyclic learning rate and adam optimizer with decoupled weight decay to stabilize the training process. Audio augmentation is applied to artificially increase the size of dataset which is followed by conversion of time-series audio samples to log-scaled mel-spectrogram. The log-scaled mel-spectrogram is then fed as input to the model to provide matching performance with the baseline results. The model produces significant results with reduced training epochs on same dataset size. Different types of audio augmentations are performed and a comparative study of three classification models including Xception, MobileNetV2 and DenseNet is presented here.

Index Terms – Convolutional Neural Network, Urban Sound Dataset, Cyclic Learning Rate, Mix-up, Adam with Decoupled Weight Decay.

1. INTRODUCTION

Noise is a growing problem in urban areas, and due to increasing urbanization more and more people are affected. Major sources of noise include transportation, construction, industry and recreational activities. The sum of all the noise is referred to as environmental noise or noise pollution. Noise pollution over sustained periods of time affects health and wellbeing in many ways. Noise can be a source of annoyance and increased stress, cause sleeping disturbance and increase risk of heart diseases. WHO has estimated that in Europe 1.6 million healthy life years (Disability-Adjusted Life Years, DALY) are lost annually due to noise pollution [1].

In India Central Pollution Control Board (CPCB) produced The Noise Pollution (Regulation and Control) Rules, 2000 under The Environment (Protection) Act, 1986 [2] to better regulate environmental noise pollution and monitor it. It highlights the duties of the state to restrict speakers, vehicles, and other sound sources to reduce and keep the environmental audio amplitude to an optimum level. It further goes on to set the volume levels (in decibels) for various regions to maintain a quality of life for its citizens. These broad regulations and objectives require analysis of sound sources in various regions and various time of the day to implement policies so that these regulations can be enforced. The noise analysis can help in creating noise maps of cities of various noise sources to achieve that objective. Deep learning models can help in training predictive models to help in discriminating various sounds from each other.

2. LITERATURE SURVEY

Salamon et al. [3] showed that deep CNN with audio augmentation can improve the classification accuracy significantly. In their work, they increased their dataset by performing five audio augmentations on each sample to create new samples. Then the CNN model was trained on the dataset which outperformed previous models. Audio augmentations are techniques to deform data while keeping the semantic meaning of the dataset. This showed that audio augmentations are one of the deciding factors in improving the model accuracy. Based on this inference, two more augmentations are used in this work which is based on policies by Park et al. [4] and Zhang et al. [5]. Two policies were introduced by Park – spectrogram masking and time warping, out of which we only used spectrogram masking as he points out that if there is resource constraint, we can leave time warping as it didn't improve the model's performance by a large factor. Spectrogram masking consisted of making blocks along time and frequency axis randomly. Augmentation policy by Zhang, called Mixup, is the addition of a fraction of data sample to input sample per batch. This operation showed improvement not only on computer vision tasks but also on audio related tasks. While Salamon's work produced better results but they came at the cost of training on a very large dataset which was created artificially through augmentations and over long training period of 50 epochs. To train in less number of epochs while matching state-of-the-art results, Smith [6] presented cyclic learning rate schedule. It consisted of triangular window cyclic learning rate policy which produced the best results with far less number of training epochs.

Adam method [7] has shown to converge faster than other methods but Wilson et al. [8] have suggested that adaptive methods do not generalize as well as Stochastic Gradient Descent (SGD) method. Loshchilov et al. [9] presented Adam with decoupled weight (AdamW) which is a modified Adam to show its improvement over Adam and showing matching results with SGD. They introduced weight decay in the Adam algorithm while removing L_2 regularization arguing that it is not the same as weight decay and has inferior performance than weight decay.

Batch normalization's (BN) error increases rapidly when the batch size becomes smaller, caused by inaccurate batch statistics estimation. This limits BN's usage for training larger models and transferring features to other tasks which require small batches constrained by memory consumption. Group Normalization (GN) [10] divides the channels into groups and computes within each group the mean and variance for normalization. GN's computation is independent of batch sizes, and its accuracy is stable in a wide range of batch sizes. Weight standardization (WS) [11] is targeted at micro-batch training setting. WS with GN is able to match or outperform the performances of BN trained with large batch sizes with only few lines of code.

3. DATASET

UrbanSound8k[3] is a collection of environmental audio sounds from www.freesound.org. It is a dataset consisting of 8,732 labeled sound excerpts with a maximum length of 4 seconds. Each audio clip belongs to one of ten classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. Each audio clip was manually labeled by Salamon et al (2017). The audio clips are in ".wav" format and vary in channels, sampling rate, bit depth and length. The dataset is divided into 10 folds. Each audio clip is also given salience rating of foreground or background. Fig. 1 shows spectrogram from each class.

Class	Samples	Duration (Average)	In Foreground
Air conditioner	1000	3.99 s	56 %
Car horn	429	2.46 s	35 %
Children playing	1000	3.96 s	58 %
Dog bark	1000	3.15 s	64 %
Drilling	1000	3.55 s	90 %
Engine idling	1000	3.94 s	91 %
Gun shot	374	1.65 s	81 %
Jackhammer	1000	3.61 s	73 %
Siren	929	3.91 s	28 %
Street music	1000	4.00 s	62 %

Table 1 UrbanSound8k - 8,732 Sound Clips of Maximum Length 4 Seconds Having 10 Different Classes

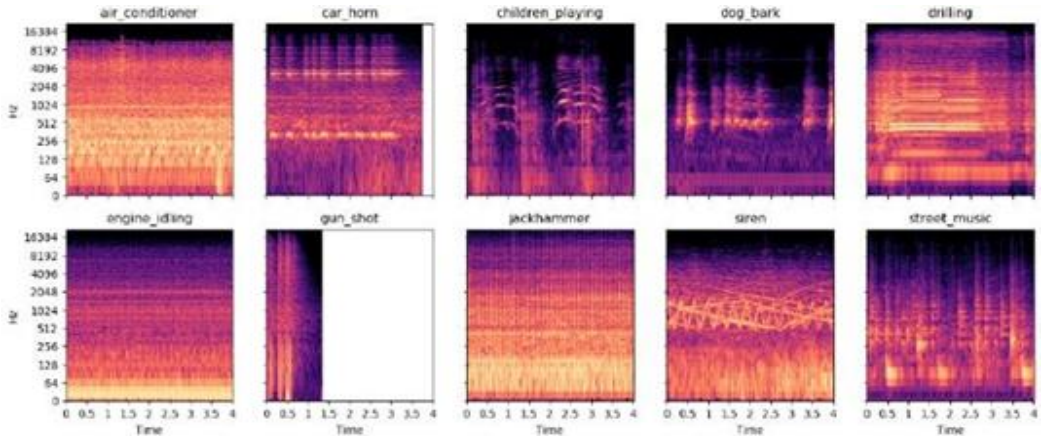

Figure 1 These Examples Show Log-Scaled Mel-Frequency Cepstral Coefficient Spectrograms of Different Classes in the Dataset

4. METHODOLOGY

On each input audio sample, 3 seconds audio clip is randomly taken and converted to normalized log-scaled mel-spectrogram. The convolutional neural network then predicts the probability of environmental sound classes and the highest probability class score is taken as the predicted class.

4.1. Data Preprocessing

Each audio has different properties but the model requires input with fixed properties so all the samples have mono channel (i.e., one channel), 16 bits bit depth, 44.1 kHz sampling rate and 3 seconds audio clip is randomly chosen from each sample. One or more of the following audio augmentations are then applied in each experiment:

- Time Stretching (TS) [3]: Slow down or speed up the audio sample (while keeping the pitch unchanged). For each sample one of five factors is randomly chosen: {0.81, 0.93, 0, 1.07, 1.23} where 0 represents no change.
- Pitch Shifting (PS) [3]: Raise or lower the pitch of the audio sample (while keeping the duration unchanged). For each sample one of nine values (in semitones) is randomly chosen: {-3.5, -2.5, −2, −1, 0, 1, 2, 2.5, 3.5} where 0 represent no change.
- Dynamic Range Compression (DRC) [3]: Compress the dynamic range of the sample using 4 parameterizations, 3 taken from the Dolby E standard and 1 (radio) from the icecast online radio streaming server where one was chosen randomly for each audio sample: {music standard, film standard, speech, radio}.
- Background Noise (BG) [3]: Mix the sample with another recording containing background sounds from different types of acoustic scenes. Each sample was mixed with 1 of 5 acoustic scenes randomly: {street- workers, street-traffic, street-people, park, none} where none represent any change. A weight *w* is chosen to parameterize the amount of noise to be added which is chosen from uniform distribution in the range [0.1, 0.3]. The operation is defined as

  $$z = (1-w).x + w.y$$

  where x is input sample, y is noise sample and z is output.
- Spectrogram Masking(SM): Mask multiple blocks of frequency and/or time in a spectrogram as described in [4] with the following values: { *mf*: 2, *mt*: 2, *freq*:6, *time*: 30, *p*: 0.2} where *mf* is number of masks in frequency axis, *mt* is number of masks in time axis, *freq* is maximum number of blocks that can be masked, i.e., *mask length* ϵ [0, *freq*] in each mask in frequency axis, *time* is maximum number of blocks that can be masked in each mask, i.e., *mask length* ϵ [0, *time*] in time axis and *p* is a factor in range (0, 1] for *time* such that $time' = time * p$.
- Mixup (MX): Mix the sample with another sample from dataset. This augmentation is done batch-wise such that the batch and its shuffled sample sequence are mixed with each other and output labels as well. {α: 0.8} is used to choose value in the range of [0, 1] for weighting the amount of mixup from beta distribution as described in [5].

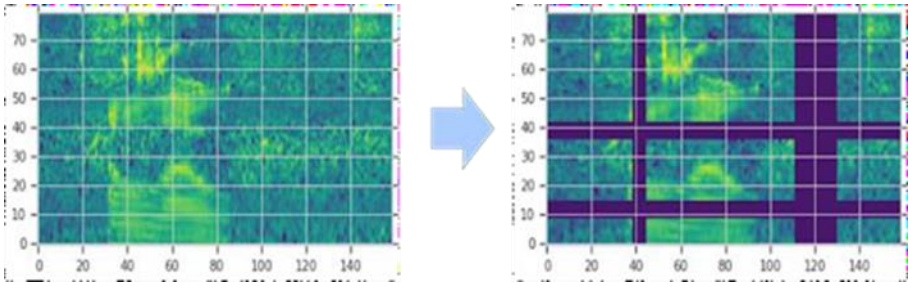

Figure 2 The Log Mel-Spectrogram is Augmented by Masking (Multiple) Blocks of Consecutive Time Steps (Vertical Masks) and Mel-Frequency Channels (Horizontal Masks). The Masked Portion of the Spectrogram is Displayed in Purple for Emphasis

Then log-scaled mel-spectrogram with 128 bands covering the audible frequency range (0-22050 Hz), using a window of 23 ms (1024 samples) and hop size of the same duration are extracted and normalized using equation

$$x' = x - mean \,/\, std$$

$$\forall\ x\ is\ spectrogram;\ mean\ \&\ standard\ deviation\ (std)\ of\ each\ spectrogram$$

This gives frequency-time patch (FT-patch) of audio X of 3 seconds, i.e., $X \in R^{128x128}$. The audio loading mel-spectrogram operations, TS, and PS are done by using Librosa library [14]. DRC is implemented by Sox [15].

4.2. Cyclic Learning Rate and Optimizer

A cyclic learning rate scheduler is chosen to change the learning rate during training with a half cycle of 4 epochs and AdamW optimizer's weight decay is also changed through the learning rate value at each batch.

Each learning rate scheduler requires an upper limit and a lower limit on the learning rate which is calculated by plotting a graph of loss vs learning rate. The lower limit is chosen where the loss just starts to reduce and the upper limit is chosen where the loss starts to increase again. For all experiments, $\alpha = 0.025$, *batch size* = 32 and *epochs* = 16.

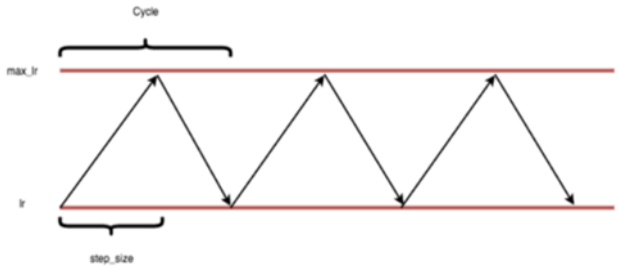

Figure 3 Triangular Window Cyclic Learning Rate Policy

4.3. Training

A convolutional neural network is used to predict the probability of environmental sound class for each spectrogram of audio. The final fully connected layer is replaced with one that has 10 outputs, after which a sigmoid nonlinearity is applied. The network was trained end-to-end using cross entropy loss & AdamW with default parameters $\beta1 = 0.9$, $\beta2 = 0.999$ (Kingma et al [7]) and cyclic weight decay & learning rate as described in earlier section. This training loop is implemented with Keras library which is now entirely integrated with Tensorflow [16].

5. EXPERIMENTS

The three experiments conducted with different model topologies discussed in previous section are as following:

5.1. Experiment 1: Xception Model

For the experimentation, TS, PS, DRC, BG, SM and MX audio augmentations are applied and learning rate ranged from $7.5x10^{-5}$ to $3.16x10^{-3}$. The network for training used Xception architecture [17] with 126 layers model as the base model and added average pooling layer and output layer on top of it. The network is initialized with Imagenet weights.

Figure 4 Learning Rate Range for Experiment 1

5.2. Experiment 2: Mobilenetv2 Model

In this experiment, TS, PS, DRC, BG, SM and MX audio augmentations are applied and learning rate ranged from $3.16x10^{-5}$ to $1.1x10^{-3}$. The network for training used MobileNetV2 architecture [18] with 88 layers model as the base model and added average pooling layer and output layer on top of it. The network is initialized with Imagenet weights.

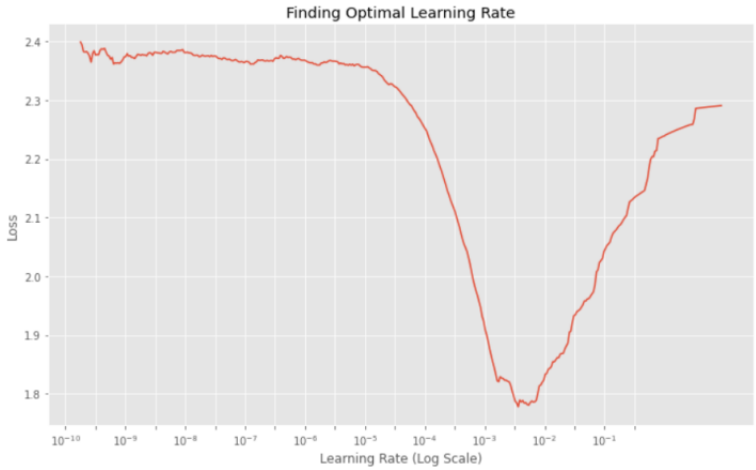

Figure 5 Learning Rate Range for Experiment 2

5.3. Experiment 3: Densenet Model

In this experiment, TS, PS, DRC, and BG audio augmentations are applied and learning rate ranged from $3.16x10^{-6} – 1.1x10^{-4}$. The network for training used a Densely Convolutional Network architecture (DenseNet) [19] with 169 layers model as the base model with BN layer replaced with GN layer, WS as kernel regularizer, and added average pooling layer and output layer on top of it. The network weights are randomly initialized.

Figure 6 Learning Rate Range for Experiment 3

6. RESULT AND DISCUSSION

The model performances are assessed on classification accuracy statistics. The model which achieved the highest performance on the test dataset is the Xception architecture with Accuracy of 0.81. MobileNetV2 achieved accuracy of 0.733 and DenseNet-169 achieved accuracy of 0.759. This shows that even though Imagenet weights are trained for handling images they can be used as starting point for other applications also using CNNs. A detailed report for each experiment on test dataset is given below. A comparison of our model performance with baseline model SB-CNN [3] is also given.

	Xception	MobileNetV2	DenseNet-169	SB-CNN
Air conditioner	0.86	0.74	0.75	0.49
Car horn	0.909	0.848	0.727	0.90
Children playing	0.8	0.65	0.67	0.83
Dog bark	0.88	0.86	0.87	0.90
Drilling	0.79	0.71	0.63	0.80
Engine idling	0.86	0.838	0.849	0.80
Gun shot	1.0	0.968	0.968	0.94
Jackhammer	0.82	0.656	0.906	0.68
Siren	0.59	0.566	0.506	0.85
Street music	0.85	0.57	0.83	0.84
Overall	0.81	0.733	0.759	0.79

Table 2 Performance Comparison of the Models with the Baseline Model SB-CNN on Accuracy Statistic

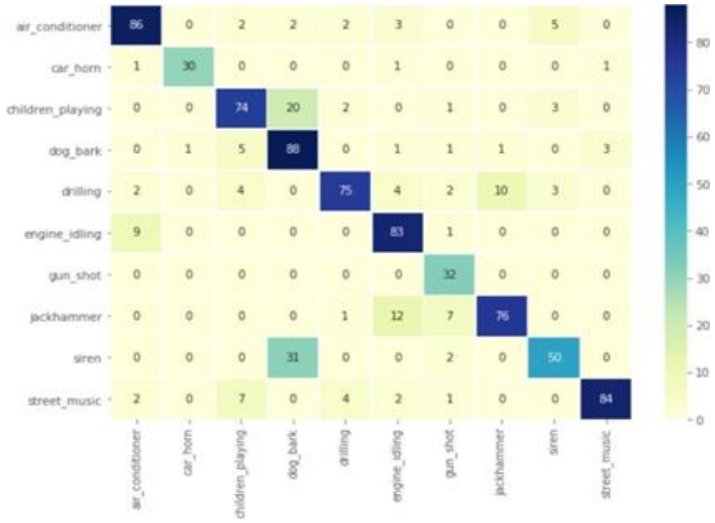

Figure 7 Confusion Matrix for Xception Model

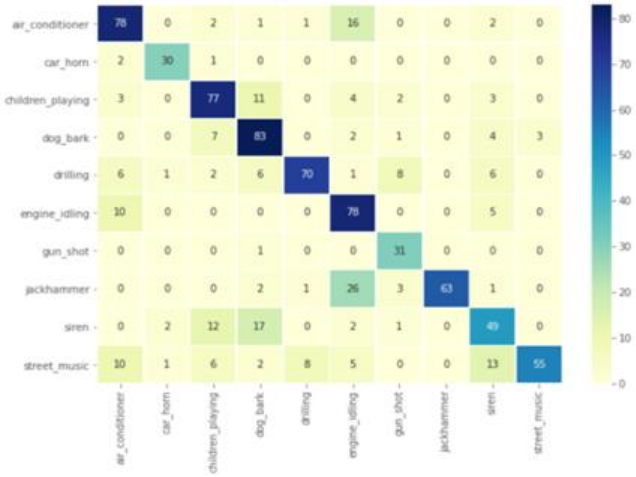

Figure 8 Confusion Matrix for MobileNetV2 Model

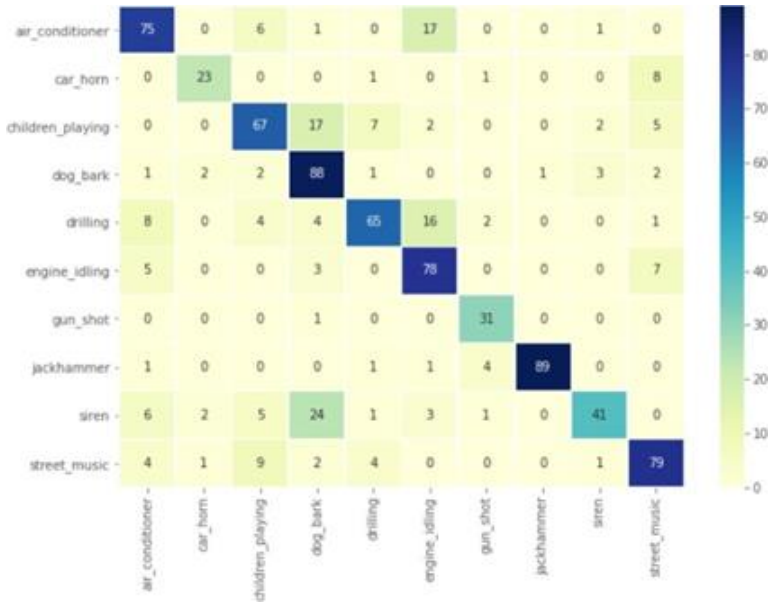

Figure 9 Confusion Matrix for DenseNet-169 Model

Three experiments are conducted on UrbanSound8k dataset using three different model architectures, namely, Densenet-169, MobileNetV2, Xception. Using normalization, audio augmentations and cyclic learning rate & optimizer, all three models achieving Accuracy statistic of more than 0.70 with the highest performance on Xception model architecture with Accuracy statistic of 0.81 outperforming the baseline model by Salamon et al [3]. All three models' performance bottlenecked due to over fitting of model on training dataset and experiment also showed that Imagenet initialization of weights of network performs better than that the network with random initializations.

REFERENCES

[1] L. Fritschi, A. L. Brown, R. Kim, D. Schwela, & S. Kephalopoulos, "Burden of disease from environmental noise: Quantification of healthy life years lost in Europe," World Health Organization (WHO), pp. 1-106, Dec. 2011.
[2] V. Sharma, "The noise pollution (regulation and control) rules, 2000," Gazette of India, Part-II Section 3 (ii), Feb. 2000.
[3] J.Salamon, & J.P.Bello,"Deep Convoution Neural Networks and Data Augmentation for Environmental Sound Classification," IEEE Signal Processing Letters, vol. 24(3), pp. 279-283, March 2017.
[4] D. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. Cubuk, Q. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in International Speech Communication Association, 2019, pp. 2613-2617.
[5] H. Zhang, M. Cisse, Y. Dauphin, D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in International Conference on Learning Representations (ICLR), 2018.
[6] L. Smith, "Cyclical Learning Rates for Training Neural Networks," in Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 464-472.
[7] D. Kingma, & J. Ba, "Adam: A Method for Stochastic Optimization," in International Conference on Learning Representations (ICLR), 2015.
[8] A. Wilson, R. Roelofs, M. Stern, N. Srebro, B. Recht, "The marginal value of adaptive gradient methods in machine learning," in Neural Information Processing Systems (NIPS), 2017, pp. 4149- 415.
[9] I. Loshchilov, F. Hutter, "Decoupled Weight Decay Regularization," in International Conference on Learning Representations (ICLR), 2019.
[10] Y. Wu, & K. He, "Group Normalization," International Journal of Computer Vision, vol. 128(3), pp. 742–755, March 2020.
[11] S.Qaio, H. Wang, et al., "Weight Standardization." Arxiv Preprint: 1903.10520, 2019.
[12] Standards and practices for authoring Dolby Digital and Dolby E bitstreams, Dolby Labortories, Inc., 2002.
[13] (2020) Icecast Streaming Media Server. [Online]. Available: icecast.imux.net/viewtopic.php?t=3462
[14] (2020) LibROSA. [Online]. Available: librosa.github.io/librosa
[15] (2020) SoX. [Online]. Available: sox.sourceforge.net
[16] (2020) TensorFlow. [Online]. Available: www.tensorflow.org
[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1800-1807.
[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, & L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520.
[19] G. Huang, Z. Liu, L. Van Der Maaten, & K. Q. Weinberger, "Densely Connected Convolutional Networks," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261-2269.

Author

Jasmine Chhikara is pursuing Ph.D. She is serving as Assistant Professor at Maharaja Surajmal Institute of Technology, Delhi, India. She possesses extensive knowledge in deep learning models and continuously works on innovative researches in the same domain.