Monocular Depth Estimation using Transfer learning-An Overview

K. Swaraja*, V. Akshitha, K. Pranav, B. Vyshnavi, V. Sai Akhil, K. Meenakshi, Padmavathi Kora, Himabindu Valiveti, Chaitanya Duggineni

GRIET, ECE Department, Hyderabad, India

Abstract. Depth estimation is a computer vision technique that is critical for autonomous schemes for sensing their surroundings and predict their own condition. Traditional estimating approaches, such as structure from motion besides stereo vision similarity, rely on feature communications from several views to provide depth information. In the meantime, the depth maps anticipated are scarce. Gathering depth information via monocular depth estimation is an ill-posed issue, according to a substantial corpus of deep learning approaches recently suggested. Estimation of Monocular depth with deep learning has gotten a lot of interest in current years, thanks to the fast expansion of deep neural networks, and numerous strategies have been developed to solve this issue. In this study, we want to give a comprehensive assessment of the methodologies often used in the estimation of monocular depth. The purpose of this study is to look at recent advances in deep learning-based estimation of monocular depth. To begin, we'll go through the various depth estimation techniques and datasets for monocular depth estimation. A complete overview of multiple deep learning methods that use transfer learning Network designs, including several combinations of encoders and decoders, is offered. In addition, multiple deep learning-based monocular depth estimation approaches and models are classified. Finally, the use of transfer learning approaches to monocular depth estimation is illustrated.

1 . Introduction

Estimation of monocular depth, which seeks to estimate the depth of each pixel given a single input RGB image, is a key job in computer vision. Many downstream applications, such as robotics, scene interpretation, 3D reconstruction, autonomous driving, medical imaging, and virtual reality, benefit from it. To estimate scene depth maps, active depth estimation techniques use lasers, structured light, besides extra reflections on the surface of the object. Obtaining dense and precise depth maps, on the other hand, generally necessitates a significant investment in both labour and computational resources. As a consequence, image-based depth estimation has been widely employed in research and may be used to an extensive variety of applications.

The principal objective of our work is to develop a model that can determine the depth of a picture. Previously, depth maps were calculated using depth signals such as vanishing points, focus and defocus, in addition to shadow. The bulk of these strategies, however, were only functional in a few cases. After the introduction of computer vision, several features made by hand and graph models from probabilistic analysis were utilized to predict maps of monocular depth utilizing constraint and non-constraint learning in the machine learning development. Geometry-based

approaches, sensor-based approaches, and deep-learning approaches may all be used to forecast them.

1.1 Methods of Depth Estimation

Geometry-dependent methods: Getting better 3D structures from a few photos depending on geometric limits is a common technique to sense depth that has been extensively researched in the last four decades. Structure from motion besides stereo vision similarity are two forms of this technology. SFM is a typical approach for predicting 3D structures after the succession of 2D picture arrangements that has been effectively employed in 3D reconstruction and SLAM. By examining a scene from two different perspectives, stereo vision matching may reconstruct 3D structures. Two cameras simulate the route of human eyes, and a cost function is used to create disparity maps of images.

Sensor- dependent methods: We use depth sensors, such as RGB-D cameras and LIDAR, to directly obtain the depth data of the related picture. The pixel-level dense depth map of an RGB picture may be obtained instantly using RGB-D cameras. However, LIDAR is frequently employed in the autonomous vehicle sector for depth measuring, and produce a scarce 3D map. We can't employ these depth sensors in tiny robotics applications like drones because of their size and power consumption. Their benefits include cheap cost, compact size, and a broad range of monocular camera

* Corresponding author: kswaraja@gmail.com

applications, among others. Because of these benefits, calculating a dense depth map after a single picture has gotten a lot of interest lately, and it's been thoroughly investigated using deep learning.

Deep learning-based approaches: When it comes to deep learning-involved techniques, through the quick advancement of deep learning, deep neural networks have shown their superior results in image processing tasks such as picture categorization, object recognition, and semantic segmentation, from others. Current research has demonstrated that using deep learning, the pixel-level depth map can be retrieved sequentially with a single picture.

1.2 Datasets

There are several monocular depth estimation datasets available, each with dissimilar kinds and depth varieties for indoor as well as outdoor applications. The following are some of the most frequent datasets used in deep learning approaches for estimation of monocular depth:

- The KITTI dataset contains a dataset of outdoor for estimation of monocular depth and deep learning-based object recognition and tracking.
- The NYU Depth V2 dataset is a deep learning-based indoor dataset for estimation of monocular depth.
- The Make3D dataset is alternative dataset for outdoor for deep learning-based estimation of monocular depth.
- The datasets above, KITTI and NYU Depth V2, besides Make3D, are altogether based on real-world scenes. Computers have created various virtual datasets, such as the dataset of SceneNet RGB-D and the dataset of SYNTHIA.

Remaining part of the work is focused on the literature survey of monocular depth estimation methods based on deep learning algorithms in section 2, and some deep learning methods and models are illustrated under section 3. Transfer learning techniques for monocular depth estimation is discussed in section 4. At last conclusions are provided in section 5.

2 Related Work

In this review, we looked at a number of studies and discussed our research on monocular depth estimation using deep learning algorithms. Monocular picture depth estimation based on a range of neural networks, including as convolutional neural networks (CNNs), recurrent neural networks, and others, has shown to be successful in recent years (RNNs). Typically, CNN-based models define depth estimation as a per-pixel regression issue that is solved using fully convolutional networks (FCNs).

Recurrent neural networks (RNNs) are a kind of neural network that uses hidden states and cyclic connections to describe the temporal behavior of sequential input. RNNs learn dependencies throughout the network by propagating backwards in time across several instances. Kyung Hyun Cho et al.[1] developed a novel neural network system named as RNN Encoder-Decoder, which consists of 2 recurrent neural networks (RNN). The first

RNN encodes a series of codes into a fixed-length vector depiction, which is then decoded into a new series of symbols by the second RNN. The encoder and decoder of the anticipated model are trained to maximize the conditional probability of a target arrangement if a source arrangement is given. When we structure depth estimation as a per-pixel regression issue, fully convolutional networks (FCN) have obtained excellent results in object identification, semantic segmentation, and depth estimation. In FCN, the skip connection structure is used to combine coarse and abstract data in distinct levels. This allows us to collect feature information from the input picture, create a pixel-level depth picture, and learn how to extract features in the depth estimation issue. The fully connected network (FCN) suggested by Laina et al.[2] eliminates completely linked layers and replaces them with efficient residual up-sampling blocks.

On common benchmark datasets, the deep learning-based algorithms stated above improve accuracy. Meanwhile, academics are working to increase the accuracy of the CNN approach. Yokila Arora et al.[3] stated that we may estimate depth using a single CNN architecture that was tailored for the purpose of depth prediction by merging information from surrounding pixels in the same local area. By keeping the down sampling and early up-convolutional layers constant, their network employs transfer learning. The weights for those layers come from a TensorFlow model that has already been trained. Furthermore, they use residual learning, notably via the skip and projection layers, in their methodology. As a result, the gradient issue may be resolved. The fully convolutional network (FCN) has recently shown promise in the areas of object identification, semantic segmentation, and depth estimation. The skip connection structure used in FCN has been proven to combine coarse and abstract information in discrete layers, enabling feature information from an input picture to be recovered while maintaining the output picture's size. As a consequence, we use a skip connection structure to train feature extraction and create pixel-level depth pictures in the depth estimation problem. The FCN network topology is used in the literature research for depth estimation as well, although the depth estimation and pose estimation parts of the network are separated. The deep estimation network element of their algorithm takes a single photo as input, while the posture estimation network takes the front and rear frames of a sequence or image pair as input.

Ground truth depth measurements are collected using depth sensors. However, in many circumstances, gathering the data might be expensive. A number of strategies have been suggested to minimize the requirement for ground truth labels, including supervised, unsupervised, semi-supervised, conditional random field (CRF), joint semantic segmentation, and information-assisted depth estimation. Unsupervised Monocular Depth Estimation was suggested by P. Zama

Ramirez et al.[4]. The popularity of single-view depth estimate has skyrocketed.

Due to the fact that the unsupervised technique does not need any ground truth during training, issues such as scale ambiguity and inconsistency have a lower performance. As a result, semi-supervised approaches are presented for improving estimate accuracy by lowering reliance on costly ground truth. By combining the unsupervised image reform error with the involvement from sparse depth ground truth tags, Kuznietsov et al.[5] executed a network in a semi-supervised way. The inverse depth maps amongst the left and right pictures are estimated using semi-supervised algorithms that train stereo image pairings. The disparity map is then generated using the expected inverse depth and utilized for inverse warping the left picture from the right picture to create the left image. Ramirez et al. [6] suggested a methodology that employs a second decoder stream for estimating semantic labels and supervised training. In addition, a cross-domain break term dependent on the anticipated semantic picture is used to increase the evenness of the anticipated depth map, resulting in improved performance.

Luo et al.[7] proposed a Deep3D-based view synthesis network to guess the right picture with the left picture. Furthermore, to restore the disparity map, a stereo similarity network is created to accept the raw left pictures and synthesized right pictures. The raw right pictures supervise the view synthesis network during training to increase the construction quality, and the left pictures are rebuilt from the estimated right pictures using expected disparity maps. We can use supervised approaches to tackle a variety of real-world computing problems more quickly and accurately. Ground truth depth values for each input picture are used to improve accuracy. Because the supervisory indication of supervised techniques is reliant on the depth maps of ground truth, estimation of monocular depth might be considered a regressive issue. Deep neural networks are used to guess depth maps with single photos using deep learning. To oversee the training of networks, the disparities between anticipated and actual depth maps are employed. CNN-based algorithms have increased the performance of depth estimation, thanks to the efficiency of deep learning. These models are trained with the use of supervision signals from supervised techniques employing depth sensors or synthetic datasets.

Convolutional networks (CNN) were introduced by Eigen et al.[8] to estimate the depth of an image using dense depth map regression to forecast the target depth values. This design is made up of two parts: a refinement network and an AlexNet-based coarse prediction network. Techniques like Conditional Random Fields (CRF), hierarchical CRF refinement, and the multi-scale CRF method are subsequently used to enhance the methodology. Depth estimation may be used with other tasks, such as semantic segmentation, to improve accuracy. With an end-to-end training, Xu et al. [9] incorporated the continuous CRF algorithm into a deep

CNN architecture. To improve the information transmission between needed characteristics, a planned attention method combined by the CRF method is presented. The use of random forests (RFs) models may upsurge the precision of depth estimation. We looked at a few different elements of monocular depth estimation in this work. We employed a deep learning-based approach for estimation of depth. To obtain ground truth depth values and obtain high performance, we employed a CNN deep learning model and a supervised deep learning-based technique. Monocular depth estimation relies on image processing techniques [10-28] and also helpful during communication [29-33].

3 Models and Methods based on Deep Learning for Estimation of Monocular Depth

3.1 Monocular depth estimate using deep learning algorithms

An encoder-decoder network that uses the RGB picture as the input and the depth map as the output is used for estimation of monocular depth dependent on deep learning. Convolution and pooling layers are used by the encoder network to record depth information, while deconvolution layers are used by the decoder network to revert the predicted pixel-level depth map of the similar dimension as the input. So as to keep the attributes of each scale, the relevant encoder and decoder layers are joined using skip-connections. The depth loss functions govern and train the whole network, which is then brought together when the necessary depth map is formed.

For estimation of monocular depth, there are three kinds of deep learning approaches: supervised and unsupervised, besides semi-supervised. Supervised techniques take a single picture and the depth information associated with it as input for training. In this case, the trained network may directly output the depth information. However, a large quantity of high-quality depth data is required, which is challenging to apply to all application scenarios. To get around the necessity for high-quality depth estimates as seed data, a number of semi-supervised techniques have been developed. Self-supervised techniques only need a small number of unlabeled pictures to train the networks for depth estimation, while semi-supervised techniques need a big quantity of labelled data and a small quantity of unlabeled data for training. By connecting numerous input modalities, these algorithms automatically extract depth information.

As a result, our objective in this work is to provide an experimental study of each important element by decoupling the widely utilized components in the supervised depth estimate technique and then examining each element's individual effects using controlled trials.

3.2 Monocular depth estimate with deep learning models

Convolutional neural networks (CNNs), recurrent neural networks (RNNs), variational auto-encoders (VAEs), and generative adversary networks are some of the deep learning models that have shown to be successful in monocular depth estimation (GANs). Various estimate algorithms of monocular depth have been presented throughout the years.

Fig 1 The general pipeline of deep learning for monocular depth estimation

3.2.1 CNN

Convolutional neural networks can extract spatial elements that reflect depth in a scene automatically. It's a form of feed-forward neural network that uses less parameters than typical approaches to extract depth data and rebuild depth maps. Convolutional layer, pooling layer, fully connected layer, and activation function are the major components of CNN, which allow it to learn the 2-d spatial properties of an input picture. To avoid pure linear combinations, the convolutional layer alters the input into depth features, the pooling layer diminishes the dimensions of the feature map at input, the fully connected layer is frequently positioned at the termination of the CNN to produce the outcome, in addition the activation function is usually a continuously differentiable nonlinear function. AlexNet, VGG, ResNet, DenseNet, and other lightweight networks like MobileNet and GhostNet are examples of CNNs. The present CNN-based depth estimate network relies on these to function.

3.2.2 RNN

RNN is a sequence-to-sequence method that contains memory abilities that is used to learn sequential characteristics from video sequences in monocular depth estimation. Recurrent neural networks are made up of three components: an input unit, a hidden unit, and an output unit, with the hidden unit's input consisting of the outputs of both the current and prior hidden units. For monocular depth estimation, representative RNNs are added into deep learning models, which are commonly paired with CNNs to obtain spatial-temporal information to recover depth.

3.2.3 GAN

The ground truth depth maps must be used to teach the supervised depth estimation model 3D mapping and scale data. However, since obtaining depth maps of ground truth in actual settings is challenging,

researchers developed GAN to produce crisper and more realistic depth maps than existing models. The generator and the discriminator are the two components that make up GAN. The discriminator decides if the depth map of input is true or false, while the generator predicts it as a depth estimation network. GAN-based depth estimation algorithms can provide well-organized ground truth depth maps. Few deep learning-based approaches, such as supervised and unsupervised, besides semi-supervised, have been suggested to ease the requirement for ground truth labels.

4. Monocular Depth Estimation Transfer Learning Techniques

4.1. Convolutional Neural Networks (CNNs)

Researchers constructed CNN-dependent estimation networks of monocular depth that learn depth information layer by layer via their convolution kernels besides recover depth maps using deconvolution to fulfil scene interpretation demands. CNN is divided into two types: completely linked CNN and CNN without completely linked layers. Convolution layers and fully connected layers make up the CNN with FC layer. The input RGB picture is first processed via six convolution layers, then the data is processed via fully connected layers, and the final output is resized to match the size of the ground truth depth map, as shown in the diagram. Although this model may overfit hundreds of photos and be used as a classification model, it fails to provide satisfactory results on the validation set. Our objective in this project is to recreate rather than categorize the jumbled picture. As a result, we utilized a convolutional neural network that did not include a fully connected layer. To rebuild the ground truth depth map picture in convolutional neural networks, we employ convolution layers rather than fully connected layers. The U Net architecture is a convolutional neural network with a few tweaks to the CNN architecture that was used to generate the output. To assist the training process, we first apply convolution to the input picture, which comprises batch normalization and a ReLU activation layer, as shown in the diagram below.

We then use max pooling, which is a pooling technique that chooses the maximum element from the feature map area covered by the filter. As a consequence, the max-pooling layer's output would be a feature map that included the most prominent features from the prior feature map. After extracting image characteristics, we employed a transfer learning methodology to build a depth map using convolution and up-projection. To up sample the feature pictures and produce the output depth picture, we employed a bi-linear up-sampling methodology.

4.2 CNN Architectures

Convolutional Neural Networks (CNNs) are a kind of multi-layer neural network that is meant to identify visual patterns directly from pixel pictures with little or no pre-processing. The ImageNet project is a massive visual database intended for use in visual object

identification, in which software applications compete to categorise and recognise items and situations properly. We'll now look at some of the CNN architectures.

4.2.1 ALEX NET

AlexNet has eight layers, five of which are convolutional and three of which are fully linked. It was put together by Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton[34]. "One of the biggest convolutional neural networks on ImageNet subsets to date," according to the researchers.

Fig 2 AlexNet architecture

It was the first to put convolutionary layers directly over each other rather than stacking a pooling layer over each convolutionary layer; it was somewhat similar to LeNet-5, but for bigger and deeper layers." The 11x11 and 5x5,3x3 configurations, max pooling, decommissioning, data increase, ReLU and SGD activation are all included. After each convolutionary and completely connected layer, ReLU activations have been implemented.

4.2.2 VGGNet

Simonyan and Zisserman [35] created VGGNet. VGGNet. VGG-16 consists of 13 convolutionary layers and 3 fully linked layers, all follow the lineage of AlexNet ReLU. Due to their remarkably similar construction, these convolutionary levels are quite attractive. The network picks up more layers on AlexNet and uses smaller filters (2 para 2 and 3 para 3). It's 2-3 weeks training on 4 GPUs. A deeper variation, VGG-19, were also developed. The VGG's weight configuration is public and has been utilized as a basic feature extractor for many different applications and challenges. It has 138 million parameters, though, and takes up about 500MB of storage, which may be quite difficult to manage.

Fig 3 VGGNet architecture

4.2.3 ResNet

Kaiming He et al.[36] built the Residual Neural Network (ResNet) with a very deep CNN consisted of 152 layers. In recent CNNs, we have seen a growth in the number of layers in design and increased performance. As the network depth increases, however, "precision gets saturated and declines fast." As illustrated in Fig.8 is ResNet-50, ResNets have 26M's parameters as the fundamental block for ResNets. In Fig.9, the simplified block diagram version is provided: The research team at Microsoft dealt with this challenge with the ResNet - constructing richer models by employing skip connections. The approach is to utilise a module to avoid performance deterioration by adding additional layers.

The residual module presented in Fig.10 includes a series of turns, a batch norm, ReLU activations and a residual connection x. The residual connection is also known as the skip connection since it gives information on the residual module function. The skip connection allows gradients to spread to early network layers obvious. This accelerates the learning process by eliminating the issue of disappearance. ResNet may take activation from one layer and then feed it to another layer further deeper in the neural network through these skip connections. ResNet allows us to improve our performance due to these qualities.

Fig 4 ResNet architecture

4.2.4 EfficientNet

Efficient Net is a convolutive neural network construction and scaling approach using a compound coefficient to equally measure all dimensions of depth, breadth and resolution. The Efficient Net scales approach evenly measures the network width, depth, and resolution using a set of preset scaling coefficients, unlike the typical methodology that scales these elements arbitrarily. For instance, if 2N times as many computing resources can be used, then the network depth may be just increased by αN , width by βN and picture size by μN . α , β and α coefficients, found by a little grid search of the initial little model, remain constant. Efficient Net employs the compound coefficient – in a principal manner, to consistently scale network breadth, depth and resolution. The compound scaling approach is presented with the premise that, if the input picture is bigger, the network will need more layers to widen the receptive field and more channels to capture the larger

picture with fine grain patterns. The EfficientNet-B0 base network is constructed on the inverted residual bottleneck blocks of MobileNetV2, in addition to the squeeze and excitement blocks. Efficient networks also transfer effectively with a lesser magnitude order and achieve the most advanced accuracy at CIFAR-100 (91.7 per cent), flowers (98.8 percent), and another three data sets for learning transfer.

Fig 5. Simplified ResNet architecture

Fig 6 Residual module of ResNet

4.2.5 DenseNet

One of the latest advances in visual object identification neural networks is DenseNet. While there are a few important differences, DenseNet is relatively similar to ResNet. ResNet utilises a supplementary approach (+) which fuses the preceding layer (identity) with the layer of future, whereas DenseNet fuses (.) the preceding layer output with the future one. DenseNet has a typical network classification. The figure displays a dense 5-layer block with a growth rate of $k = 4$ and the Resnet structure as typical.

The output of the preceding layer becomes an input of the second layer via the composite process. The composite operation comprises the convolution layer, batch layer, batch standardization and non-linear activation layer. This means that the network is connected directly with $L(L+1)/2$. A letter L indicates the number of levels in the architecture. There are many variations of the Dense Net, such as DenseNet-121, DenseNet-160, DenseNet-201 and others. The accuracy of the various CNN designs is shown in Fig.13. AlexNet, VGG types, ResNet types, EfficientNet, and DenseNet have been compared. In comparison with other CNN

designs ResNet is more precise from the picture above. So we chose ResNet-50 instead of other designs. We have also experimented with employing designs EfficientNet and DenseNet comparable to ResNet and comparing ResNet outcomes.

5. Conclusion

We intend to contribute to the growing area of monocular depth assessment research based on deep learning. Consequently, the essential investigation into monocular depth estimates is examined in terms of training methodologies. A monocular depth estimation is described by a variety of profound learning approaches and models along with transfer learning strategies.

References

1. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., *arXiv preprint arXiv:1406.1078*, (2014).
2. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F. and Navab, N., *Fourth international conference on 3D vision (3DV)* (pp. 239-248) (2016).
3. Arora, Y., Patil, I. and Nguyen, T., (2017).
4. Ramirez, P.Z., Poggi, M., Tosi, F., Mattoccia, S. and Di Stefano, L., *Asian Conference on Computer Vision* (pp. 298-313). Springer, Cham, (2018).
5. Kuznetsov, Y., Stuckler, J. and Leibe, B., *IEEE conference on computer vision and pattern recognition* (pp. 6647-6655) (2017).
6. Ramirez, P.Z., Poggi, M., Tosi, F., Mattoccia, S. and Di Stefano, L., *Asian Conference on Computer Vision, Springer* (pp. 298-313), (2018).
7. Ji, R., Li, K., Wang, Y., Sun, X., Guo, F., Guo, X., Wu, Y., Huang, F. and Luo, J., *IEEE transactions on pattern analysis and machine intelligence*, 42(10), pp.2410-2422, (2019) .
8. Wan, L., Eigen, D. and Fergus, R., *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 851-859) (2015).
9. Pilzer, A., Xu, D., Puscas, M., Ricci, E. and Sebe, N., *International Conference on 3D Vision (3DV)* (pp. 587-595) (2018).
10. Kuraparthi, S., Kollati, M., Kora, P. *Traitement du Signal*, **36(6)**, 565-573 (2019).
11. Swaraja, K., *J. Adv. Res. Dyn. Control Syst.(JARDCS)*, Special, (11) (2017).
12. Swaraja, K., Madhaveelatha, Y. and Reddy, V.S.K., *Int J Appl Eng Res*, **11(1)**, 216-225 (2016).
13. Swaraja, K., Latha, Y.M., Reddy, V.S.K. and Paramkusam, A.V., *2011 Annual IEEE India Conference*, (pp. 1-4) (2011).
14. Swaraja, K., Madhaveelatha, Y. and Reddy, V.S.K., *IEEE International Conference on Computational Intelligence and Computing Research*, (pp. 1-4) (2014).

15. Swaraja, K., Madhaveelatha, Y. and Reddy, V.S.K., *ARPN Journal of Engineering and Applied Sciences*, **10(4)**, pp.1822-1827 (2015).
16. Kora, P., Meenakshi, K., Swaraja, K., Rajani, A., & Islam, M. K., *Informatics in Medicine Unlocked*, **17**, 100257 (2019).
17. Kora, Padmavathi, and K. Sri Rama Krishna, *Information systems design and intelligent applications*, Springer, pp. 309-317, (2016).
18. Kumari, C.U., Kora, P., Meenakshi, K., Swaraja, K., Padma, T., Panigrahy, A.K. and Vignesh, N.A., *International Conference on Innovative Computing and Communications*, Springer (pp. 425-433), (2020).
19. Kumari, C.U., Kora, P., Meenakshi, K. and Swaraja, K., *3rd International Conference on Computing Methodologies and Communication (ICCMC)*, (pp. 114-117), (2019).
20. Swaraja, K., Meenakshi, K. and Kora, P., *Biomedical Signal Processing and Control*, **68**, p.102688, (2021).
21. Sravan, V., Swaraja, K., Meenakshi, K., Kora, P. and Samson, M., *4th International Conference on Trends in Electronics and Informatics (ICOEI)*, (48184), (pp. 1063-1068), (2020)
22. Meenakshi, K., and GN Beena Bethel. *First International Conference on Networks & Soft Computing (ICNSC2014)*, (2014).
23. Meenakshi, K., Swaraja, K. and Kora, P., *Soft computing and signal processing*, Springer, Singapore, (pp. 477-485), (2019).
24. Kora, P. and Kalva, S.R., *SpringerPlus*, **4(1)**, pp.1-19 (2015).
25. Kumar, S.K., Reddy, P.D.K., Ramesh, G. and Maddumala, V.R., *Traitement du Signal*, **36**, pp.233-237 (2019).
26. Kumari, C.U., Prasad, S.J. and Mounika, G., *3rd International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 1095-1098) (2019).
27. Dhanalaxmi, B., Naidu, G.A. and Anuradha, K., *Procedia Computer Science*, **46**, pp.432-442 (2015).
28. Jayahari, L., Sasidhar, P.V., Reddy, P.P., BaluNaik, B., Gupta, A.K. and Singh, S.K., *Journal of King Saud University-Engineering Sciences*, **26(1)**, pp.21-31 (2014).
29. T. Pavani, A. Hemanth, I. S. Narayana and Y. R. Rao, *Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 54-57 (2021).
30. Valiveti, H.B. and Polipalli, T.R., *International Journal of Computer Aided Engineering and Technology*, **10(5)**, pp.599-608, (2018).
31. Valiveti, H.B. and Kumar, B.A., *International Journal of Communication Systems*, **34(6)**, p.e4285, (2021).
32. H. B. Valiveti, S. Ganesh, B. A. Kumar and D. K. Yadav, *Computers, Materials & Continua*, **67(2)**, pp. 2049–2059, (2021).
33. Koduru, A., Valiveti, H.B. and Budati, A.K., *International Journal of Speech Technology*, **23(1)**, pp.45-55 (2020).
34. Krizhevsky, A., Sutskever, I. and Hinton, G.E., *Advances in neural information processing systems*, **25**, pp.1097-1105, (2012).
35. Simonyan, K. and Zisserman, A., *arXiv preprint arXiv:1409.1556*, (2014).
36. He, K., Zhang, X., Ren, S. and Sun, J., *IEEE conference on computer vision and pattern recognition*, (pp. 770-778), (2016).