

# CONDITIONAL GANS AS A SOLUTION TO IMAGE-TO-IMAGE RENDERING PROBLEMS

Aleem Ali<sup>1</sup>, Pooja Malik<sup>2</sup>

<sup>1</sup>Assistant Professor, Department of Computer Science, Shiv Nadar University, U.P, India

<sup>2</sup>Associate Professor, Department of CSE, Glocal University, Saharanpur, U.P, India

Email: <sup>1</sup>aleem@theglocaluniversity.in, <sup>2</sup>pooja.malik@snu.edu.in

**Abstract:** In many existing solutions of image-to-image rendering problems, the only focus is to find the closest output of the Generative Adversarial Network (GAN). In this research article, authors propose a generative adversarial network, a solution to pixel-to-pixel rendering problems and reduced the loss function to the maximum under all interactions. For achieving the best result, we have considered the mean square loss function in the generator and binary cross for the discriminator. Our proposed model deals with not only images but also read sketches where the edges are not sharp too. We have used a facade dataset to test our proposed model.

**Keywords:** Generative Adversarial Network (GAN), Image to image rendering, U Net Architecture, Generator and discriminator loss, Convolutional Neural Network (CNN), Deep Learning (DL).

## 1. Introduction

Generative adversarial networks (GAN) are generally used for unsupervised learning in which the neural network tries to generate the best possible data from the training set to the desired result [1]. These networks grasp the mapping from the input picture to the output picture and also present “loss function” to synchronize this mapping. GANs therefore, make sustainable strategy for the problems that conventionally would require loss formulations. The GAN structure normally adopts a game-theoretic strategy [2]. It generally comprised of two neural networks to coach and contend with one another, one generator and other is discriminator. The purpose behind picking up the word “adversarial” is due to their consistent clash during the entire training cycle. The generator (i.e. the forger) endeavors to develop a kind of currency almost alike to real-world currency by way of acquiring knowledge about the newest ploy to beguile the police, i.e., the discriminator. The police should ceaselessly refresh their data to identify the forged currency. Both the networks are persistently refreshing their knowledge and receiving feedback on their effective alterations. This entire process proceeds up to the police fail to recognise the genuine data. In the paper [3], the counterfeiter is creating legitimate examples. In this paper also, we are trying to illustrate an approach, which is constructive at integrating photos from label maps, regenerating objects from

edge maps, and colorizing images. This technique helps to generate a rational image without much artistic skills and experience in image rendering. For these methods, we need to build a non-differentiable method that can generate replicative sketches for the given photos. Later all the other necessary editing has to be made such as graph-cut composting to make the generated image close to the true image.

## 2. Related Work

Generally, GANs effort is dependent on the maximum likelihood estimation (MLE), Markov chain and approximate inference. The MLE is the foundation of Boltzmann Machine (RBM) [4] and its sisters' models [6]. Generated output with the traditional methods rapport the knowledge distribution with the testing distribution. These following models contain few serious limitations, supposed not to be much summed up. In view of unsupervised and semi-supervised learning, GANs models were developed by Ian Goodfellow [7]. Y. LeCun [8] presented adversarial testing which is the better stimulating and intriguing notion of the previous decade in machine learning and AI.

A conditional GAN or cGAN pixel to pixel model is studied in paper [9] in which input is a source image and output is condition-based. The output image is a logical transformation of the plain image. The discriminator is a function of adaptive failure and will be discarded once the generator is equipped.

ConvNet is a Deep Learning methods which supposed to receive input image and attributing value to different aspects and differentiates one from another. The pre-processing needed in a ConvNet is considerably lower than any other neural algorithm. Although the filters are applied in simple ways. ConvNets are prepared enough to know such filters [10-11]. A ConvNet's architecture is equivalent to the human brain neuron patterns. The whole visual area is a collection of field overlaps [12]. The RGB image is taken as an input and separated by its three color planes i.e., Red, Green, and Blue. This type of neural network is very significant when designing an architecture that is not only good in terms of learning but also expandable to huge datasets.

Generally, a class convergence on a training dataset is noticed as minimizing selected loss function on the training data. Convergence signals, the two-player game ends in a

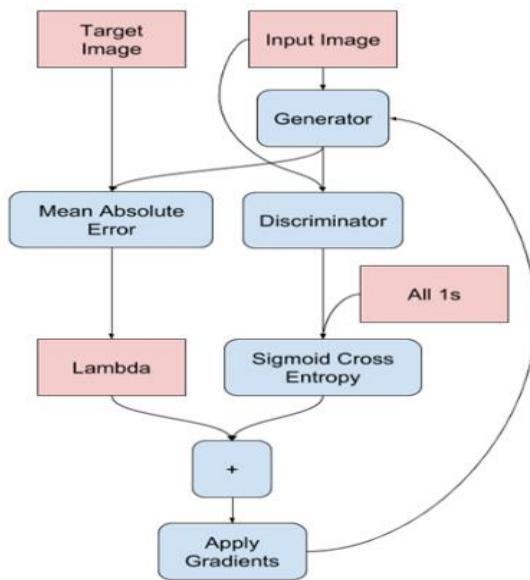
GAN. It seeks to strike a balance between the models [13]. The paper's [14] inputted diverse noise vectors to each steps of upsampling. Modifying noise vectors for each resolution, the users can regulate "high-stage details" with the aid of using low noise and "low-stage details" with the aid of later steps. The practical add-on might be as much original, while the computational cost is pretty high.

In the paper [15], the Pix2Pix methods tackle with issues like transforming line drawings to resulting paintings, permitting operators to contain few control of artistic regulator in order to refining sketches. In view of the Cycle GAN methods diminishes the requisite for balancing training samples. The well-known practices of this kind of method is to swapping cattle for ships and grapes for banana. Nevertheless, switching from Ludo to PUBG is unquestionably more pleasing to the youth-gen. The key element behind Pix2Pix method is the U-Net architecture.

Salimans, Tim, et al. [16] presented the improved expression for testing a classifier using GAN methods. The classifier is collectively educated to categorize images and come across to identify bogus images, meanwhile the generator attempts to create classifiable photos. These photos are not supposed to be bogus. The unlabelled data may be applied to educate the generator, and affords more data for the classifier to understand. This research paper is not centred on the philosophy of it. It is stated that "if you may create something, you should know already about it". This research work is an illustration of how GANs can assist in different activities. The authors proposed a self-focussed methods for GANs [17], an approach to complement universal cognition. The self-attention mechanism determines "what each pixel thinks of each other pixel." Consequently, it permits the model to build universal relationships. This task is completed with the assistance of the exterior creation of the flattened image with itself.

### 1.1 Standard GAN Loss Functions

This strategy comprises of two-loss functions, the initial one named as the Minimax GAN Loss, and the other named as the Non-Saturating GAN Loss [18]. The Typical diagram of the GAN Loss Diagram is shown in figure 1.



**Figure 1. Typical diagram of GAN Loss Diagram**

### Discriminator Loss

In the Standard GAN loss function, the discriminator classifies the original and duplicate pictures assigned by maximizing the loss function as mentioned in eq. 2.1. The discriminator pursues the highest the log probability average for original pictures and the inverted probability log for duplicate pictures [19-20]. The loss to the discriminator is the same under both schemes.

$$\text{Maximize } \log D(x) + \log (1 - D(G(z))) \quad (2.1)$$

In view of stochastic ascent, this will need variations to model weights. For real images, it is usually pragmatic as a conventional binary sorting difficulties with labels 0 and 1. The approach is suited for lowest the average entropy of binary crosses, also termed as log loss.

$$\text{Minimize } y_{\text{true}} * -\log (y_{\text{predicted}}) + (1 - y_{\text{true}}) * -\log (1 - y_{\text{predicted}}) \quad (2.2)$$

### Minimax GAN Loss

Minimax is known as the optimization approach of two-player games designed to minimize the loss of the other player's worst case. It denotes the simultaneous minimax optimization of the models of the generator/discriminator [21]. Both players are the generator and discriminator and have the chance to keep on frequently changes their loads. The min and the max term refers to minimizing the cost of the generator and maximizing the loss of the discriminator. Therefore,

$$\min\max (D, G) \quad (2.3)$$

Again, the identifier pursues the highest log probability of the original pictures and the log inverse probability for duplicate pictures.

$$\text{Discriminator: highest } \log D(x) + \log (1 - D(G(z))) \quad (2.4)$$

Therefore, this will encourage the creator to produce models with a short likelihood of being duplicate

$$\text{Generator: lowest } \log (1 - D(G(z))) \quad (2.5)$$

It clearly states that if the generator couldn't train as swiftly as the differentiator, the discriminator triumphs, the game finishes, therefore, the model is not supposed to be refined successfully.

## 1.2 Substitute GAN Loss Functions

Loss function selection is a burning subject of study, and several loss functions have been suggested and estimated. The least-square loss, Wasserstein loss, Cross-Entropy loss, etc. are many common substitute loss functions that cast-off in numerous GAN implementations.

## 1.3 Least Squares GAN Loss (LSGAN)

This strategy is based on observing the limitations of cross-entropy loss once developed images look dissimilar from factual images. This can result in fading gradients and cannot updates the model [22]. Here, the discriminator finds to lower the square sum difference for factual and false images between the predicted and actual values.

$$\text{Discriminator: lowest } (D(x) - 1)^2 + (D(G(z)))^2 \quad (2.6)$$

In the following equation 2.7, the expected generator looks for minimizing the sum squared difference concerning the predicted and real values.

$$\text{Generator: minimize } (D(G(z)) - 1)^2 \quad (2.7)$$

Generally, least square loss strategy provides a greater penalty to higher error values, producing a major rectification.

For generator's appropriation  $p_g$  over information  $x$  [23], a sharing noise variable  $p_z(z)$ , signifying distribution to information plane as  $G(z; \theta_g)$ , ( $G$ -differentiable function characterized having parameters  $\theta_g$ ). Another single scalar's output is achieved by multilayer perceptron  $D(x; \theta_d)$ .  $D(x)$  characterizes the likelihood which  $x$  originates other than  $p_g$ . Again, to learn  $D$  to increase the likelihood of allotting the right mark to learning approaches and tests from  $G$ . We at the same time train  $G$  to minimize  $\log (1 - D(G(z)))$ .

The weighted function for  $V(G, D)$  is shown below in eq. 2.8

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log (1 - D(G(z)))] \quad (2.8)$$

Image GAN contains various constraints and higher depth than the  $70 \times 70$  Patch GAN.

## 3. System Model

This model is a kind of conditional GAN model. Here also, the discriminator is supplied the source and the final image, and further, conclude whether the final one is a reasonable source image conversion. In this specialized model, the generator is made competent to learn through adversarial loss, which encourages it to generate reasonable images in the final domain.

### GENERATOR (U-net Architecture)

**Upsampling phase:** In this phase, the balancing route is comprised of four building blocks. Where individual block has a  $3 \times 3$  complexity layer along with activation function and each  $2 \times 2$  maximum pooling.

**Downsampling phase:** The expanding path is again comprised of four building blocks. Every individual block has complexity layer along stride two. They are concatenated using the matching feature map from the balancing route. The downsampling phase also has a  $3 \times 3$  convolution layer along with an activation function.

### DISCRIMINATOR (PatchGAN Architecture)

The system model considers two pictures as an input, the source, and a target pic. Both the pictures are linked at the channel level. The configuration in PatchGAN uses Convolution-Batch Norm-Leaky ReLU layers block and the number specifies the number of filters. It is designed with the size of the receptive field, sometimes referred to as the effective receptive field [24-31]. The model output can be a distinct value or a square value activation map predicting about each patch in the input picture is real or fake.

## 4. Implementation

This CNN model is implemented using U-Net architecture which is developed for the segmentation of medical imaging.

### 4.1 U-Net architecture

The architecture has an enormous count of feature maps in the upsampling because of its symmetry which allows information to be transferred. This network is comprised of two parts which are the contracting path, also known as the downsampling path, and the other one the expanding or upsampling path.

This appropriate data will move through skip connections to the upsampling path. The expanding path contains 4 blocks. The reason for expanding the route is to allow accurate localization shared with circumstantial data from the route of the contract.

### 4.2 Patch GAN

The PatchGAN performs a fixed  $2 \times 2$  and a static kernel size of  $4 \times 4$ . Therefore, we can calculate the size of the

receptive field starting with one pixel in the output picture and functioning towards the input. Here the produced size is considered the size of the activation map for previous layers and kernel size is the filter size to be implemented.

#### 4.3 Training Phase

Using the patch GAN approach as we can train and generate high-resolution images. The training phase steps of the GAN model are mentioned below step by step:

**Step 1:** For each input source generates an output.

**Step 2:** Now, the differentiator proceeds the picture and generates the picture as the initial response. The next response is this input picture.

**Step 3:** In this step, we determine losses of the model.

**Step 4:** In this step, gradients of loss are determined w.r.t the both the identifiers and employ it to upgraded.

**Step 5:** In this step, log the losses to Tensor Board.

The real testing circle starts from here

1. Recurrence over the quantities of epochs.
2. On each epoch, it rectifies the demonstration and initiates creating pictures to point out its improvement.
3. On each epoch, it repeats over the preparation dataset, printing a '!' for each model.
4. It keeps a designated spot on each 20 epochs.

#### 4.4 TESTING

Testing of the model is a crucial phase in which inferring the logs from a GAN is refined than any easy regression model. These are the simple steps of the testing phase.

**Step 1:** Ensure that no approach has "won". Whichever the gen\_gan\_loss becomes very low, then the model is ruling the other one, and we aren't training the model proficiently.

**Step 2:** For testing point of view,  $\log(2) = 0.69$  considered a decent placing point because of showing a perplexity of two. Therefore, differentiator is normal value.

**Step 3:** Lower assessment 0.69' projects that differentiator is showing improvement over irregular, along with joint set of original and duplicated pictures.

**Step 4:** Also for the gen\_gan\_loss, lower assessment 0.69 is showing that gen' is showing improvement over irregular on flooding discriminator.

**Step 5:** As long as the learning phases advance, the gen\_loss supposed to go lower.

#### 5. Results and Discussions

The results present in the loss analysis are the improved gradient data at refreshing the loads of the generator. We can learn from the results that the loss function might not

deliver an appropriate gradient for G to learn well. At the very initial phase of training to model, if G is found lacking, D immediately can discard data or values with good confidence so as they're dissimilar from the under investigation data. This includes the Generator Loss, Generator Total Loss, Discriminator Total Loss described below in the following figures 2, 3, and 4. The learning further might embrace and addition of Wasserstein loss to eliminate the weight clipping is said Wasserstein Gradient Penalty loss (WGAN GP). Figure 5 and figure 6 are generated using the test dataset.

```
for inp, tar in test_dataset.take(5):
    generate_images(generator, inp, tar)
```

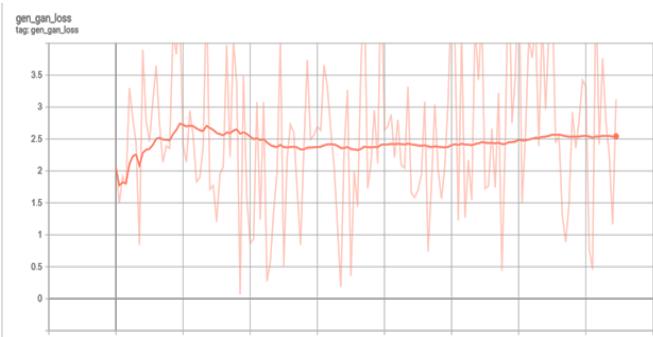


Figure 2 Illustrating Generator Loss

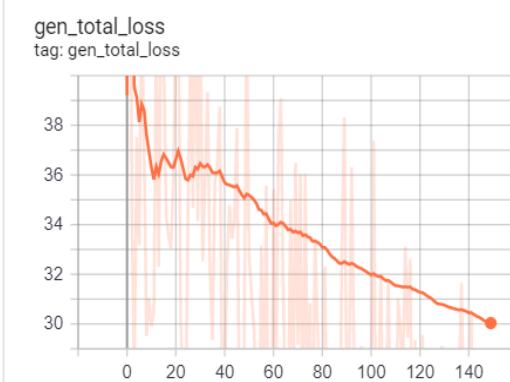


Figure 3 Illustrating Generator Total Loss

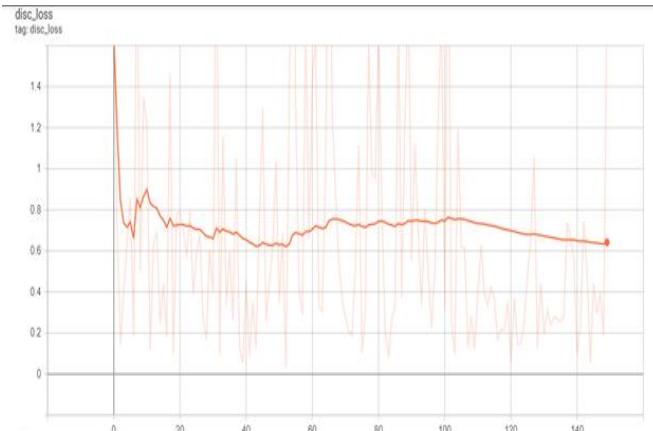


Figure 4 Illustrating Discriminator Total Loss

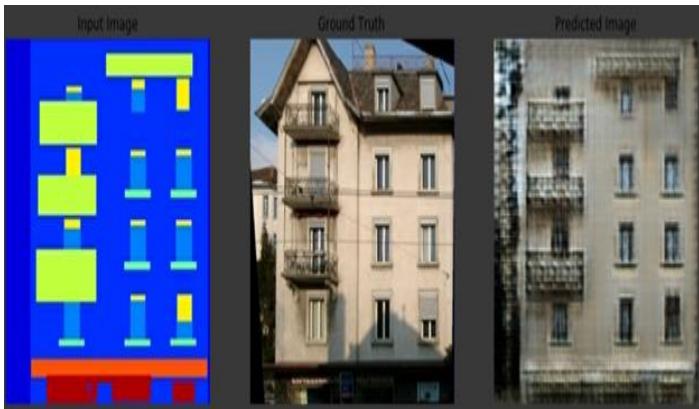


Figure 5 Generated using Test Dataset

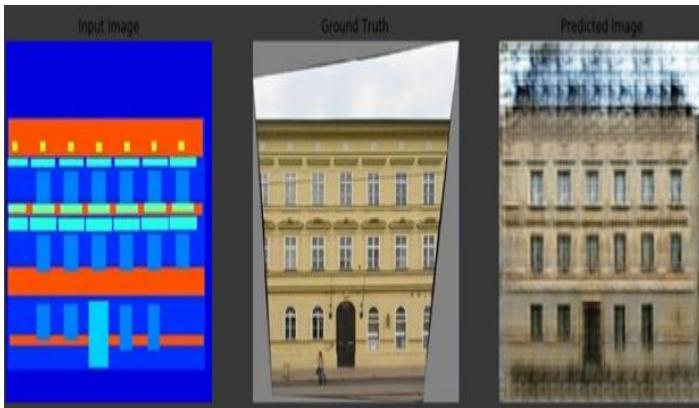


Figure 6 Generated using Test Dataset

## 6. Conclusion

The output of this research paper concludes that GANs are a promising strategy for many pix-to-pix transformations, particularly relating to highly structured graphical outputs. These specialized networks learn a loss analysis adapted to changes that make them applicable in a wide variety of settings. Using U-net as a generator has been an immense advancement for forwarding low-level features through the network and rebuilding it at the output.

## References

- [1] Jason Brownlee, Generative Adversarial Networks with Python Deep Learning Generative Models for Image Synthesis and Image Translation, Machine Learning Mastery, Vermont Victoria 3133, Australia. ACN: 626 223 336, 2020.
- [2] Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, David Foster, published in 2019.
- [3] I. Goodfellow, ‘NIPS 2016 tutorial: Generative adversarial networks’, arXiv Prepr. arXiv1701.00160, 2016.
- [4] P. Smolensky, Information processing in dynamical systems: Foundations of harmony theory. Cambridge: MIT Press, 1986.
- [5] G. E. Hinton, S. Osindero, and Y.-W. Teh, ‘A fast learning algorithm for deep belief nets’, Neural Comput., vol. 18, no. 7, pp. 1527–1554, 2006.
- [6] R. Salakhutdinov and G. Hinton, ‘Deep boltzmann machines’, in Proceedings of the Twelfth International Conference on Artificial intelligence and statistics, pp. 448–455, 2009.
- [7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ‘Generative adversarial nets’, in Advances in neural information processing systems, pp. 2672–2680, 2014.
- [8] Y. LeCun, ‘RI Seminar: The Next Frontier in AI: Unsupervised Learning’, 2016.
- [9] Antonia Creswell, Tom White, Vincent Dumoulin, et. al., “Generative Adversarial Networks: An Overview”, IEEE Signal Processing Magazine, vol. 35, Issue 1, pp. 53 – 65, Jan. 2018.
- [10] K. Teilo, An Introduction to Convolutional Neural Networks, pp. 0-11, NOVEMBER 2015.
- [11] V. Dumoulin and F. Visin, A guide to convolution arithmetic for deep learning, pp. 1-28, 2016.
- [12] O. Abdel-hamid, L. Deng and D. Yu, Exploring Convolutional Neural Network Structures and Optimization Techniques for Speech Recognition, pp. 3366-3370, August 2013.
- [13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., "Generative adversarial nets" in Advances in Neural Information Processing Systems, vol. 27, pp. 2672-2680, Red Hook, NY, USA, 2014.
- [14] Karras, Tero, Samuli Laine, and Timo Aila. “A style-based generator architecture for generative adversarial networks.” **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. pp. 1-12, June 2019, USA.
- [15] Zhu, Jun-Yan, et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks.” **Proceedings of the IEEE international conference on computer vision**. 2017.
- [16] Salimans, Tim, et al. “Improved techniques for training gans.” **Advances in neural information processing systems**. 2016.
- [17] Zhang, Han, et al. “Self-attention generative adversarial networks.” **arXiv preprint arXiv:1805.08318**. 2018.
- [18] A. Lucas, S. Lopez-Tapia, R. Molina and A. K. Katsaggelos, “Generative adversarial networks and perceptual losses for video super-resolution”, **IEEE Trans. Image Process.**, vol. 28, no. 7, pp. 3312-3327, Jul. 2019.
- [19] M. Arjovsky, S. Chintala and L. Bottou, “Wasserstein generative adversarial networks”, **Proc. 34th Int. Conf. Mach. Learn.**, vol. 70, pp. 214-223, 2017.
- [20] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong and L. Lin, “Unsupervised image super-resolution using Cycle-in-Cycle generative adversarial networks”, **Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)**, pp. 81-814, Jun. 2018.
- [21] Ying Jin, Yunbo Wang; Mingsheng Long; Jianmin Wang; Philip S. Yu; Jiaguang Sun, A Multi-Player Minimax Game for Generative Adversarial Networks, IEEE International Conference on Multimedia and Expo (ICME), 6-10 July 2020, United Kingdom, Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, Stephen Paul Smolley, Least Squares
- [22]

- Generative Adversarial Networks, Computer Vision and Pattern Recognition, pp. 1-16, 2017.
- [23] Ian J. Goodfellow, Jean Pouget-Abadie et. al., Generative Adversarial Nets, pp. 1-9, Canada.
- [24] Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. Journal of machine learning research 9(Nov):2579– 2605.
- [25] He Huang, Philip S. Yu and Changhu Wang, An Introduction to Image Synthesis with Generative Adversarial Nets, Computer Science, pp 1-17, China, 2018.
- [26] Ravi Raj Choudharya, K K Jisnua, Gaurav Meena, Image DeHazing Using Deep Learning Techniques, Procedia Computer Science 167, pp. 1110–1119, 2020.
- [27] Iftikhar Husain, Aleem Ali, Fuzzy Matrix Approach to Study the Maximum Age Group of Stressed Students Studying in Higher Education, International Journal on Emerging Technologies, 12(1), pp. 31-35, 2021.
- [28] Aleem Ali, Neeta Singh “M/M/1/n+Flush/n model to enhance the QoS for Cluster Heads in MANETs” published in “International Journal of Advanced Computer Science and Applications (IJACSA)”, U.K. 2018.
- [29] Aleem Ali, Neeta Singh, P. Verma, A Modified M/M/c/K+Flush/K Analytical Model for improving Delay Aware QoS of MANET, IOSR Journal of Engineering (IOSRJEN), Vol 9, Issue 10, pp. 57-64, Oct. 2019.
- [30] P. Malik and A. Singh Baghel, "A Summary and Comparative Study of Different Metrics for Machine Translation Evaluation," **2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)**, Noida, India, 2018, pp. 55-60, doi: 10.1109/CONFLUENCE.2018.8442777
- [31] P. Malik and A. S. Baghel, "An improvement in BLEU metric for English-Hindi machine translation evaluation," **2016 International Conference on Computing, Communication and Automation (ICCCA)**, Greater Noida, India, 2016, pp. 331-336, doi: 10.1109/CCAA.2016.7813740.