STAR-GALAXY CLASSIFICATION USING MACHINE LEARNING ALGORITHMS AND DEEP LEARNING

*Amit Sadanand Savyanavar *, Nikhil Mhala, Shiv H. Sutar*

School of Computer Engineering & Technology, Dr. Vishwanath Karad MIT World Peace University, Pune
India

* Corresponding Author: e-mail: amitsavyanavar@gmail.com

Abstract: Cosmology is the study of the universe comprising stars and galaxies. Advancement in the telescope has made it possible to capture high-resolution images, which can be analysed using machine learning (ML) algorithms. This paper classifies the star galaxy dataset into two classes: star and galaxy using ML algorithms and compares their classification performance. It is observed that random forest provides better accuracy of 78% as compared to other ML classifiers. Further to improve the classification accuracy, we proposed a CNN (Convolution Neural Network) model and achieved an accuracy of 92.44%. Since the CNN model itself extracts the characteristics, it exhibits superior classification accuracy.

Key words: Cosmology, star-galaxy classification, Machine Learning, Convolution Neural Network.

1. INTRODUCTION

Modern cosmology [1] is a vast interdisciplinary science. Given the widespread use of technology and image capturing techniques, stars and galaxies have been categorized using a wide variety of machine learning (ML) algorithms. The source classification is the fundamental problem in astronomy. Now a days due to advancement in the technology, many ML algorithms have been proposed to classify the star and galaxy images. ML has shown better results in classifying the images in various fields. There are many ML methods [2] to classify stars and galaxies. Recent use of Deep Learning (DL) [3] provided better accuracy than the traditional ML algorithms. It has been validated on various datasets which are available and have better accuracies. CNN is one of the most efficient algorithms in classifying image datasets. ML algorithms and CNN learn from the training dataset to develop a desired input-output relation. This is crucial for tasks like speech and image recognition and strategy optimization, which have significant applications in the IT sector. Due to ability of the CNN architecture to learn and extract the relevant

features from the images, it has been providing the better results for image datasets. This has motivated us to leverage and propose a novel CNN based architecture to the classify the source as star/galaxy classes.

This paper discusses the classification accuracy achieved by the traditional ML algorithms. Also, the paper proposes a novel CNN architecture to improve the classification accuracy on the ARIES dataset. To the best of our knowledge, CNN model has not been proposed to classify the star/galaxy images on ARIES dataset. The proposed architecture is divided into three sub blocks. The first block extracts the relevant features, second layer enhances the extracted feature for better training followed by noise reduction block to provide the 92.44% classification accuracy for

ARIES dataset. Further, section 2 discusses the dataset, section 3 describes the ML algorithms used for classification, section 4 proposes the CNN model, section 5 discusses the results and analysis and finally the conclusion in section 6.

2. DATASET

The dataset we used comprises of 3986 star images and 3986 galaxy images from a project at ARIES (Aryabhatta Research Institute of Observational Sciences), Nainital, India [10]. The images were captured in the observatory situated at Devasthal, Nainital, India using the in-house 1.3m telescope. The captured images were of size 2kx2k. These images were reduced to 64x64 cut-outs to identify the sources in each image. For each image, segmentation was used to detect the source and labelling was done using the database. According to the label, cut-outs were stored in different directories. This data is generated from scratch using the real-world data and data samples of galaxy are shown in Figure 1 and stars in Figure 2. We use this dataset to train computer vision models to classify stellar sources like stars and galaxies in the telescope images.

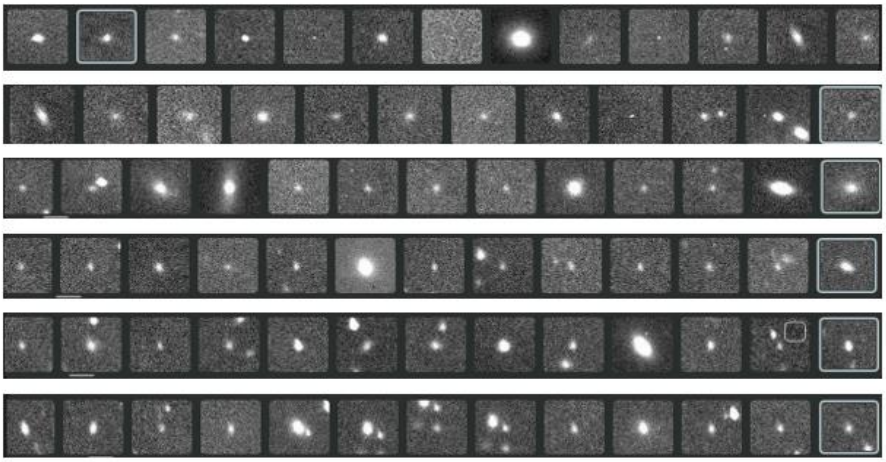

Fig. 1. Sample Galaxy images

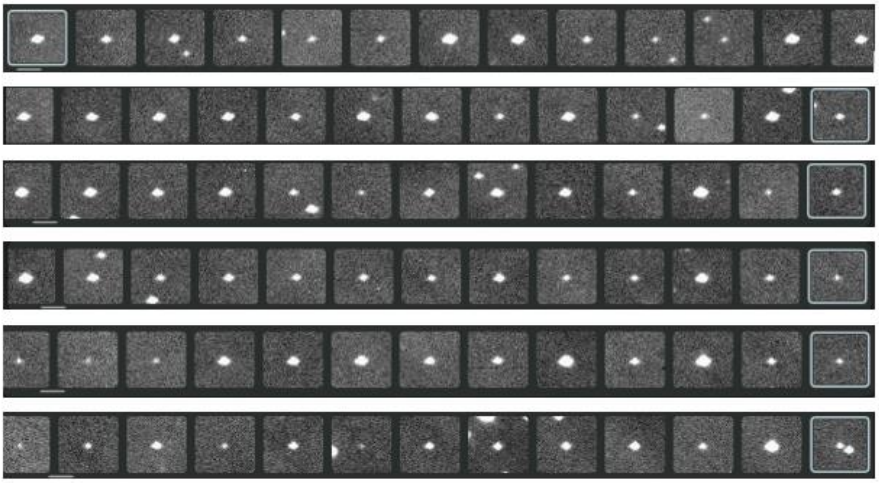

Fig. 2. Sample Star images

It is clear that we cannot classify stars and galaxies with our naked eye. Hence in order to classify them, ML and DL methods could be used.

3. ML ALGORITHMS FOR CLASSIFICATION OF STAR/GALAXY

K-nearest neighbours (KNN): KNN [4] is a popular supervised learning classifier in the community of categorization because of its effectiveness and simplicity. It is also one of the most basic ML methods that could effectively be used for text and image classification. It calculates the distance between the elements in the training set and the element that needs to be classified as a part of the test set. Calculating the k-nearest neighbours yields the anticipated class. The distances can be calculated using metrics such as the Euclidean measure. The amount of computation required for this method depends on the size of the training set. The model's output is discrete when the k-nearest neighbour method is employed. Each class is given a probability in the probabilistic KNN implementation. KNN has the benefit of being straightforward and effective in numerous domains. The complexity of computation does, however, rise as the quantity of data points begin to rise. In addition, KNN still exhibits inductive biases or model misfits as a result of its underlying assumptions, such as the notion that training data is spread equally across all classifications.

Random Forest (RF) classifier: An ensemble classifier called Random Forest [5] is built from a set of decision trees collectively referred to as the forest. Prediction is the result of adding several classifications that each tree produces. Based on the stochastic technique used to identify the attributes that will optimize information gain, each tree is distinct. Additionally, in order to develop new trees, the bootstrap statistical method is employed to create distinct datasets from the original one. As with the KNN algorithm, the output for the discrete case is the one that received the

most votes. To establish the final classification, the class assignment probabilities generated by each built tree are averaged using the arithmetic mean. The RF feature importance is then calculated by averaging the feature importance values for each tree in the ensemble. In comparison to Decision Trees (DT), the diversity of trees reduces bias, leading to more accurate models overall. Contrarily, RF consumes more time and memory than DT. The number of decision trees to be generated and the number of variables to be chosen and tested for the best split while building the trees are the two factors that must be established to construct the forest trees.

Naive Bayes (NB): Naive Bayes [6], a probabilistic ML technique is based on the Bayes Theorem, which is used to solve a variety of categorization problems. The Bayes Theorem is a fundamental formula in mathematics that can be used to estimate conditional probabilities. The posterior probability, or $P(X|Y)$, calculates the likelihood that X will occur when Y does. $P(Y|X)$ displays the frequency of Y when X happens. $P(X)$ stands for the likelihood that X will happen, and $P(Y)$ stands for the likelihood that Y will happen. The fundamental assumption of NB is that each attribute contributes equally and independently to the outcome. This probabilistic classifier is an instance of supervised learning. For each class, it produces the posterior probability. The input is classified into a class having the highest probability. It is simple to use and produces good results, but it requires extensive prior knowledge of probabilities and incurs high computing costs.

Support Vector Machines (SVMs): SVMs [7] which are a part of kernel-based methods, represents a considerable improvement in ML algorithms. When employed for classification, regression, or other tasks, SVMs create a hyperplane in a high- or infinite-dimensional space. In general, higher the margin, better the classifier. Hence the optimal separation corresponds to the hyperplane that is farthest from the nearest training data point of any class. The benefits of SVMs include the ability to build a strong predictive model with only a few free parameters, their robustness against some types of model violations and outliers, and their superior processing efficiency versus many other approaches. SVMs as well as other kernel-based algorithms have found success in several application areas, including text mining, fraud detection, insurance, chemistry, and bioinformatics. As a result, SVMs are now at the forefront of statistical learning and ML. SVMs are highly regarded by astronomers as well as statisticians, mathematicians, and computer scientists. They are also highly regarded by engineers and data analysts.

Logistic regression (LR): When the dependent variable is dichotomous i.e., binary data coded as 1 (yes or true), or 0 (no or false), building ML models using LR [8] is a statistical technique. It is a technique of predictive analysis that is built on the notion of probability. LR is used to analyse a dependent variable, along with one or more independent variables, in order to characterize the data. It's utilized to forecast the probability of a categorical dependent variable. The independent variables could be of the nominal, interval, or ordinal types. The word "logistic regression" refers to the idea of the logistic function that it uses. The value of this logistic function ranges from 0 to 1 and is also known as the sigmoid function. The

sigmoid function is used in ML to translate predictions into probabilities. Mathematically, sigmoid function can be expressed as

$$f(x) = 1/(1 + e^{-x})$$

LR performs better when the data can be separated into separate linear groups. It doesn't require a lot of processing resources because of its outstanding interpretability. There is no need for tuning because the input features can be scaled without any difficulty. It is simple to implement and train a model using LR. It offers both the direction of the relationship (positive or negative) and a measurement of the relevance of a predictor (coefficient size).

Decision Trees Classifier (DT): The Decision Tree Classifier [9] method repeatedly splits the dataset into a tree-like structure. At each split, a feature of the dataset is selected based on least impurity of the newly created nodes. The first step in creating a decision tree was defining an information gain (IG) function. IG is used to divide the dataset at each phase. Entropy, classification error and Gini are some examples of impurity functions that can be utilized. When the tree is fully formed, the probability distributions for each class are used to divide the feature space. Throughout the aforementioned branching process, some features pop up more frequently than others. The contribution of each feature to the prediction step can be measured using this frequency. Although DT is characterized by straightforward handling and interpretation, it may be biased and subject to revisions in the training set.

4. PROPOSED CNN BASED MODEL

Due to advancement in the technology and computing power of the system, CNN [3, 11, 12] has been used popularly in the image classification. The paper proposed a CNN based architecture as shown in Figure 3. The model contains total 8 convolution layers and three fully connected layers to flatten the features to classify images into two classes. The images are classified as star or galaxy in the last layer.

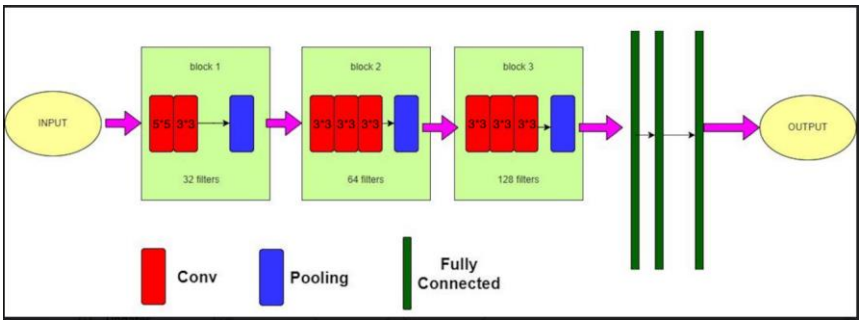

Fig. 3. Architecture of CNN Model

type	filters	filter size	padding	non-linearity	initial weights	initial biases
convolutional	32	5×5	-	leaky ReLU	orthogonal	0.1
convolutional	32	3×3	1	leaky ReLU	orthogonal	0.1
pooling	-	2×2	-	-	-	-
convolutional	64	3×3	1	leaky ReLU	orthogonal	0.1
convolutional	64	3×3	1	leaky ReLU	orthogonal	0.1
convolutional	64	3×3	1	leaky ReLU	orthogonal	0.1
pooling	-	2×2	-	-	-	-
convolutional	128	3×3	1	leaky ReLU	orthogonal	0.1
convolutional	128	3×3	1	leaky ReLU	orthogonal	0.1
convolutional	128	3×3	1	leaky ReLU	orthogonal	0.1
pooling	-	2×2	-	-	-	-
fully-connected	2048	-	-	leaky ReLU	orthogonal	0.01
fully-connected	2048	-	-	leaky ReLU	orthogonal	0.01
fully-connected	2	-	-	softmax	orthogonal	0.01

Fig. 4. CNN Model details

The network summary of the proposed architecture used to improve learning capability of the model is as discussed below.

1) Convolution + Convolution + pooling (2CP): The first block uses the filter size of 5 *5 and 3* 3 with 32 filters. The initial features extracted from the first block are reduced using the pooling layer. The pooling layer minimizes the number of parameters to be learnt by the system and the number of computations to speed up the training.

2) Convolution + Convolution + Convolution + pooling (3CP): In the second block there are total 3 convolution layers with 64 filters each having filter size as 3 * 3. The second layer enhanced the features extracted from the block one to provide better features. Lastly pooling is applied on the second block to further reduce the computations in the system.

3) Convolution + Convolution + Convolution + pooling (3CP): In the third block with the assistance of the same number of convolution layers and pooling layers features are extracted, but with a total number of filters increased from 64 to 128.

4) Fully-connected + Fully-connected+ Fully-connected (3F): The last block receives as input, the output of the preceding block. The last block contains 3 fully-connected layers. First 2 fully-connected layers flatten the input features into filters of 2048 and last layer classifies them into 2 classes as star or galaxy.

The use of above-mentioned blocks extracts the features from the star/galaxy dataset images. Since the datasets contains more noisy data, first 2CP block tries to extract the meaningful features from them. Further these features are enhanced by passing them to block 3CP with 64 filters to get the enhanced features from the previous 2CP block. The higher number of filters leads to more abstraction that the CNN model is able to extract. Hence 3CP block is further passed to extract more relevant features to 3CP block with filters increased from 64 to 128 respectively. Finally, the output of the 3CP block is fed into the fully connected block 3F to flatten the features into two classes as star or galaxy respectively.

5. RESULTS AND ANALYSIS

The result and analysis section are divided into two subsections as

a) Performance of ML based classifiers for classification and

b) Performance of CNN based models for classification

The performance of the aforementioned strategies is assessed in our paper using the ARIES dataset. To reduce overfitting induced upon by the magnitude of the dataset, we use the data augmentation approach. The data argumentation being commonly used to combat the over fitting. In the data argumentation, often the dataset is artificially increased by label-preserving transformations. The following transformations are applied on the images to raise the number of images.

• Rotation: The dataset of star-galaxies has the advantage that rotating the image does not change the observation like it is a star or galaxy. Hence images are randomly rotated by multiples of 90.

• Reflection: To exploit the mirror symmetry of the images, they are flipped horizontally with a probability of 0.5.

• Translation: Image translation helps the model to search for the observation inside the image. Since the dataset has translational symmetry, the images are translated by cropping it to 60*60 size and then shifting the images up to 4 pixels vertically and/or horizontally.

5.1. Performance of ML based classifiers for classification

Table 1 compares the performance of ML algorithms using accuracy, precision, recall and f1-score. The ratio of the total number of class predictions that were valid to the total number of test predictions is used to calculate accuracy. The precision denotes the ratio of positive predictions that are correct. Recall is used to calculate the ratio of actual positives that were predicted incorrectly. F1-score is the harmonic mean of precision and recall. The classification accuracy for the RF is 78.68% whereas for KNN, DTC, NB, LR and SVM it is 78%, 71%, 68%, 74% and 77% respectively. From Table 1 it is evident that RF provides better accuracy as compared to existing ML algorithms.

Table 1: Performance of ML algorithms

Algorithm	***Precision***	***Recall***	***F1-score***	***Accuracy***
RF	*99%*	*78%*	*87%*	*78%*
KNN	*98%*	*78%*	*87%*	*78%*
DTC	*80%*	*81%*	*81%*	*71%*
NB	*74%*	*83%*	*78%*	*68%*
LR	*88%*	*81%*	*84%*	*74%*
SVM	*99%*	*77%*	*87%*	*77%*

5.2. Performance of CNN based model for classification

The CNN model designed is divided into three sub blocks. The first block extracts the relevant features, second layer enhances the extracted feature for better training followed by noise reduction block to provide 92.44% classification accuracy for ARIES dataset.

Figure 5 depicts the loss function comprising of training loss and validation loss for 25 epochs. Training loss is the measure of how well a CNN model fits the training data and validation loss measures the model's performance on the validation data. AUC function which exhibits the strength of separability in a good model is shown in Figure 6. AUC gauges the model's accuracy by looking at the area below the curve. An efficient model displays an AUC close to 1. ROC curve in Figure 7 denotes the true positive and false positive rates.

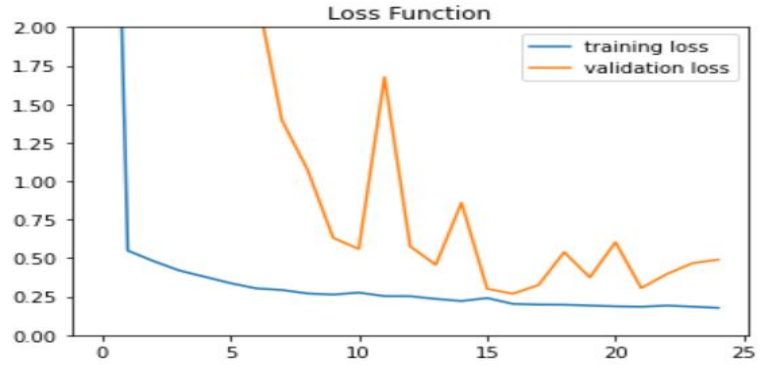

Fig. 5. Loss Function

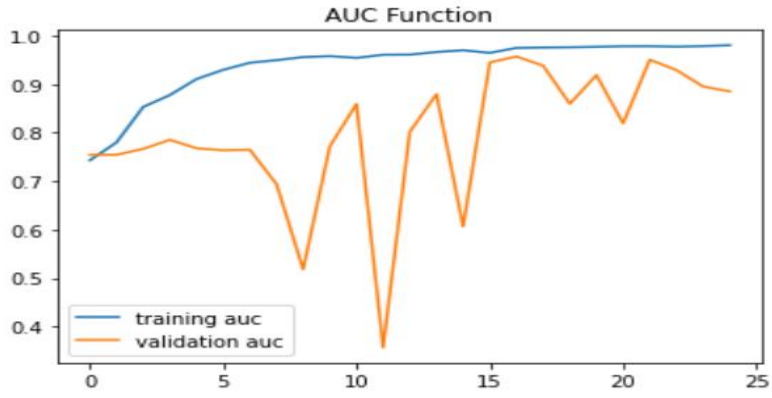

Fig.6. AUC Function

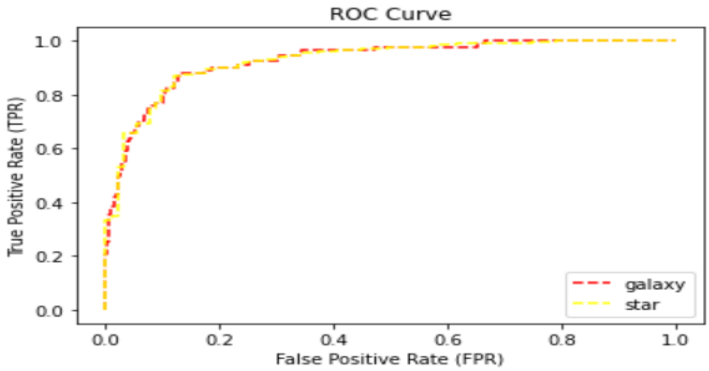

Fig. 7. ROC Curve

6. CONCLUSION

This paper has studied the start-galaxy classification using the traditional ML algorithms. The traditional ML algorithms extract the features from images and classified them into two classes. The existing ML algorithms like KNN, SVM and RF have shown significant accuracy on the ARIES dataset. However, they achieved a classification accuracy of 78%. Hence to further improve the accuracy, this paper proposed a CNN based deep learning model, which achieved an accuracy of 92.44% for the ARIES dataset. Further in future the different CNN models can be used to further improve the classification accuracy of the star galaxy datasets. Also, we can subcategorize the stars into seven different types given the features like temperature, colour and luminosity (brightness) as O, B, A, F, G, K, and M-class stars.

REFERENCES

[1] S. Karypidou, I. Georgousis and G. A. Papakostas, Computer Vision for Astronomical Image Analysis, *IEEE International Conference on Progress in Informatics and Computing (PIC)*, Shanghai, China, 2021, pp. 94-101, doi: 10.1109/PIC53636.2021.9687023

[2] Wen Xiao-Qing, Yang Jin-Meng, Classification of star/galaxy/QSO and star spectral types from LAMOST data release 5 with machine learning approaches, *Chinese Journal of Physics*, vol. 69, 2021, pp. 303-311, ISSN 0577-9073, https://doi.org/10.1016/j.cjph.2020.03.008

[3] Srikanta Swamy Pushpalatha, Shrishail Math, Human activity recognition using hybrid model, *International Journal on Information Technologies and Security*, vol.14 , no.4, 2022, pp. 55-66.

[4] Arka Banerjee, Tom Abel, Nearest neighbour distributions: New statistical measures for cosmological clustering, *Monthly Notices of the Royal Astronomical Society*, vol. 500, no. 4, February 2021, pp. 5479–5499, https://doi.org/10.1093/mnras/staa3604

[5] Sathya D., Primya T., Vinothini S., Priya J., Jagadeesan D., Smart health system using stacking ensemble classification algorithm, *International Journal on Information Technologies and Security*, vol.14, no.3, 2022, pp. 67-78.

[6] Andrew J Wilson, Ben S Lakeland, Tom J Wilson, Tim Naylor, A naive Bayes classifier for identifying Class II YSOs, *Monthly Notices of the Royal Astronomical Society*, vol. 521, no. 1, May 2023, pp. 354–388, https://doi.org/10.1093/mnras/stad301

[7] Y. Zhang, and Y. Zhao, Applications of Support Vector Machines in Astronomy, *Astronomical Data Analysis Software and Systems XXIII*, 2014, vol. 485, pp. 239.

[8] Leire Beitia-Antero, Javier Yanez and Ana I. Gómez de Castro, On the use of logistic regression for stellar classification, *Experimental Astronomy*, 2018, vol. 45, pp. 379–395, https://doi.org/10.1007/s10686-018-9591-4

[9] Manana Chumburidze, Nana Shonia, The Algorithms of Strategic Financial Management, *International Journal on Information Technologies and Security*, vol. 14, no.1, 2022, pp. 29-36.

[10] https://www.kaggle.com/datasets/divyansh22/dummy-astronomy-data

[11] Munir Siraj, Jami Syed and Wasi Shaukat, Knowledge graph based semantic modelling for profiling in industry, *International Journal on Information Technologies and Security*, vol. 12, no.1, 2020, pp. 37-50.

[12] Ali Usman, Rauf Abdul, Shoukat Ali and Ul Abu, Big data analytics for a novel electrical load forecasting technique, *International Journal on Information Technologies and Security*, vol. 11, no.3, 2019, pp. 33-40.

Information about the authors:

Amit Sadanand Savyanavar – Received M.E. and Ph.D. degree in CSE from Shivaji University, Kolhapur, India. Currently working as Associate Professor at Dr. Vishwanath Karad MIT World Peace University, Pune, India. Areas of research are Mobile Computing, Operating Systems, Computer Vision and Distributed Systems.

Nikhil Mhala – Received M.Tech. degree in CSE from Walchand College of Engineering, Sangli, India and Ph.D. in Computer Science and Engineering from NITK, Surathkal, Mangalore, India. Currently working as Assistant Professor at Dr. Vishwanath Karad MIT World Peace University, Pune, India. Areas of research are Visual Cryptography, Image Processing and Computer Vision.

Shiv H. Sutar – Received M.E. degree in IT from University of Pune, Pune, India. Currently working as Assistant Professor at Dr. Vishwanath Karad MIT World Peace University, Pune, India. Areas of interest are Wireless Sensor Networks, Internet of Things, and High-Performance Computing.

Manuscript received on 07 April 2023