Indonesian Journal of Electrical Engineering and Computer Science
Vol. 32, No. 2, November 2023, pp. 704~714
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v32.i2.pp704-714

Evaluation of optimization techniques with support vector machine for identification of dry beans

Nabin Kumar Naik[1], Prabira Kumar Sethy[1], Rajat Amat[1], Santi Kumari Behera[2], Preesat Biswas[3]
[1]Department of Electronics, Sambalpur University, Burla, India
[2]Department of Computer Science and Engineering, Veer Surendra Sai University of Technology Burla, Burla, India
[3]Department of Electronics and Tele-Communication Engineering, Government Engineering College, Jagdalpur, India

Article Info

Article history:

Received May 10, 2023
Revised Jul 23, 2023
Accepted Aug 17, 2023

Keywords:

Dry bean
Identification
Image processing technique
Optimization technique
Support vector machine

ABSTRACT

Automated recognition or categorization of fruits or crops through image processing technology is a highly sought-after technique in agriculture. This approach has been demonstrated to outperform manual monitoring in humans. It reduces the time and expenses incurred by farmers and ensures that end users receive high-quality products. This study details the process for identifying dry beans. We employed a support vector machine (SVM) model and three optimization techniques: Bayesian optimization, random search optimization, and grid search optimization. This study aimed to determine the optimal optimization technique for identifying dry beans using an SVM model. Various metrics, including the validation accuracy, test accuracy, total validation cost, total test cost, and minimum classification error, were computed and analyzed to compare the performance of each optimization method. In comparison, Bayesian optimization was determined to be the most effective optimization technique for identifying dry beans, with a validation accuracy of 93.06%, test accuracy of 92.29%, and minimum classification error of 0.069384.

Corresponding Author:

Prabira Kumar Sethy
Department of Electronics, Sambalpur University
Burla, India
Email: prabirsethy.05@gmail.com

1. INTRODUCTION

Beans are a substantial crop worldwide and are well liked by cultivators and consumers. Dried beans comprise almost half of the leguminous grains directly consumed by humans in most developing nations [1]. Therefore, the use of multivariate and deep-learning techniques to apply classification methods is important in the agricultural industry. Categorizing different seed varieties and evaluating their quality is of utmost importance, as it significantly impacts crop production [2]. Dry beans exhibit high genetic diversity and are the most extensively cultivated edible legume crop worldwide. Seed quality plays an important role in crop production. Therefore, seed classification is crucial for both marketing and production as it aids in establishing sustainable farming systems [3]. Haricot beans, a highly nutritious food consumed worldwide, exist in various types that differ in characteristics, such as taste, size, and economic value. However, despite the differences between the types, bean grains are often mistaken for one another. Therefore, it is crucial to distinguish bean grains from different species. In the later phases of crop production, analyzing and sorting seeds is essential [4]. Customers in today's cutthroat marketplace require consistency and safety. Therefore, manufacturers must provide goods that can be sorted based on visual characteristics, such as shape, size, color, and freshness. Grain identification allows farmers to grow and sell profitable crops. This ensures that the grains they sell are on par with industry requirements, boosting their bottom line [5].

2. RELATED WORK

Krishnan *et al.* [6] created a multiclass classification model based on machine learning (ML) methods to classify seven distinct varieties of dried beans. Because ML algorithms can be skewed towards the majority in an unbalanced multiclass dataset, researchers used the random under-sampling technique to create a balanced dataset. A multiclass classification model was built using the attributes of seven different dried bean types from the UCI ML repository dataset. The characteristics of the dataset were subjected to a box-cox transformation (BCT) to determine its skewness. To select the optimal ML classification method, 22 alternative algorithms were tested on standardized and cleaned datasets. A 10-fold cross-validation method was used to verify the accuracy of the ML algorithm output. CatBoost ML had the greatest mean accuracy throughout validation, at 93.8% (92.05%-95.35%). Set out to find a dependable classifier with low noise consequences and developed an effective algorithm for classifying dry beans. Researchers have employed a dataset of dried bean grains, each with 16 characteristics, 12 dimensions, and four unique morphologies. Python's interquartile range (IQR) function was used to clean the dataset with any missing values. Logistic regression (LR), Naive Bayes (NB), k-nearest neighbour (KNN), decision tree (DT), random forest (RF), extreme gradient boosting (XGB), support vector machine (SVM), and multilayer perceptron (MLP) were among the eight popular classifiers used, along with both balanced and imbalanced classes. Frequency tables, bar graphs, box plots, and analysis of variance were used for descriptive analysis and data pre-processing. The results indicated that the XGB classifier performed the best regardless of whether the distribution of dry beans within each class was even or uneven. They found that the XGB classifier had an ACC of 93.0% for unbalanced classes and 95.4% for balanced classes [7]. Lawi and Adhitya [8] evaluated the quality of cocoa beans based on their digital pictures' morphology. The area, perimeter, major axis length, minor axis length, aspect ratio, circularity, roundness, and ferret diameter were some of the morphological or physical parameters they derived. Normal, broken, fractured, and skin-damaged beans are the four types of imperfect cocoa beans [8]. In this study, a multiclass ensemble least-squares SVM (MELS-SVM) was utilized as the classification model. The findings of the proposed model showed that the four cocoa bean classes could be accurately classified with a precision of 99.705% using only the morphological feature input parameters [8].

Further, using computer vision, Taspinar *et al.* [9] developed a system for classifying dried beans. A total of 33,064 images representing 14 bean species were collected to form the collection. Transfer learning was used to train three separate pre-trained convolutional neural networks (CNN) utilizing 33,064 photos of 14 different species of beans. In particular, they used CNN models InceptionV3, VGG16, and VGG19. SVM and LR models use the retrieved features as inputs. Six different models (InceptionV3+SVM, VGG16+SVM, VGG19+SVM, InceptionV3+LR, VGG16+LR, and VGG19+LR) were used to assign classes to the photos. The accuracy in classifying data was 79.60% for InceptionV3+SVM, 81.97% for VGG16+SVM, and 80.64% for VGG19+SVM. The VGG16+LR model exhibited the best data classification accuracy. With a classification accuracy of 84.48%. InceptionV3 surpassed its competitors. To determine the color of a typical high-quality coffee bean, Gonzales *et al.* [10] used a picture of a batch of beans as a reference. Researchers have removed coffee beans whose area values were much lower than those of typical coffee beans. Coffee beans with skin flaws and fading have a smaller surface area than healthy beans because their form of extraction is imperfect. gourmet coffee beans. Researchers effectively separated low-quality beans from high-quality ones using an image processing method that allowed them to recognize differences [10]. Nasirahmadi and Behroozi-Khazaei [11] used a multilayer perceptron artificial neural network (MLP-ANN) and machine vision methods to classify beans according to their colors. Among the bean varieties gathered were the Khomein1, KS21108, Khomein2, Sarab1, Khomein3, KS21409, Akhtar2, Sarab2, KS21205, and G11870, all of which are cultivated in Iran. The MLP-ANN classifier was then fed the retrieved six color features of the beans and the six color features of the spots on the beans [11]. High levels of sensitivity and specificity were achieved by the MLP-ANN classifier throughout the training, validation, and testing stages, with values of 100%, 97.33%, and 96%, respectively. These findings suggest that the MLP-ANN classifier is useful for properly categorizing beans into their respective classes [11]. The accuracy of automated temporal change detection between two photographs of beans was proposed by Alban [12] they used a strategy based on the analysis of texture and spatial color qualities that are important for bean seeds. To do this, they used wavelet transform and fuzzy-logic segmentation. Digital image processing based on color and texture analysis [12] is used in the suggested methodology, making it a more accurate and robust method for grading beans that are difficult to cook.

To achieve standardization in seed categorization, Koklu and Ozkan [13] developed a computer vision system to distinguish between seven registered types of dry beans that share comparable characteristics. To create the classification model, they took high-quality photographs of 13,611 grains representing seven types of registered dry beans. After the segmentation and feature extraction processes were applied to the collected bean images, 16 features (12-and four-shaped features) were obtained for each grain. Using 10-fold cross-validation, we constructed four distinct classification models: MLP, SVM, k-NN, and DT. The performance of the models was compared using established measures. Classification rates for Barbunya, Bombay, Cali, Dermason, Horoz, Seker, and Sira beans were 92.36%, 100.00%, 95.03%, 94.36%, 94.92%, 94.67%, and

86.84%, respectively, using the SVM model [13]. Nasution *et al.* [14] devised a system to determine the quality of coffee beans by measuring their human systems integration (HSI) values. A backpropagation artificial neural network was trained to distinguish high-quality seeds from those of lower grades using the HSI value acquired from the converted RGB values of the beans. Testing showed that the developed program could identify poorly trained coffee bean samples using a user interface built in MATLAB R2013a [14]. To create a smartphone app that can successfully recognize and classify seed photos using a CNN, a deep learning approach, Başol and Toklu [15] adjusted pre-trained ResNet50, Inceptionv3, Xception, and Inception-ResNetV2 models. Images were scaled down to 224×224 for the ResNet50 model and 299×299 for the Inceptionv3, Xception, and Inception-ResNetV2 models when comparing their accuracies on the Seed15 dataset. The ResNet50 model used a pool size of 4×4 in its average pooling layer, whereas Inceptionv3, Xception, and Inception-ResNetV2 models used pool sizes of 8×8. The Xception model was shown to have the best accuracy (88%) of the four pre-trained models, followed by the InceptionResNetV2 (61%), InceptionV3 (37%), and ResNet50 (91%) models [15].

3. MATERIAL AND METHOD

3.1. About dataset

For this experiment, seven different types of dried beans were used, which were chosen for their specific shape, size, and marketability. To standardize seed categorization, a computer vision system was developed to distinguish between the seven recorded varieties of dry beans that had many features. The categorization model was created by photographing 13,611 individual grains of seven distinct registered types of dry beans using a high-resolution camera. The computer vision system captured photos of beans, and then the images underwent segmentation and feature extraction. In total, 16 characteristics, including 12 spatial dimensions and four geometric shapes, were determined from the grains. Seven different types of dry beans were used in this study, which were chosen based on characteristics such as form, shape, type, structure, categorization by the Turkish Standards Institute, and consumer desires. Seker, Barbunya, Bombay, Cali, Dermosan, Horoz, and Sira have only a few names for these species. Table 1 [16] shows the common characteristics of all designated dry beans.

Table 1. Dry beans and their features

Dry beans	Features
Cali	Its seeds are white, somewhat plumper than dried beans, and shaped like kidneys.
Horoz	These dry beans are medium in size and shape, with a long, cylindrical shape and a white color.
Dermason	White in color with a rounded end and a round end on each bean, this kind of dry bean is flatter.
Seker	White, large, and round, these seeds are everything from little.
Bombay	Its flesh is white, its seeds are enormous, and its shape is round and rounded.
Barbunya	Its big, almost spherical seeds have a beige base with scarlet streaks or variegated, spotted coloring.
Sira	The seeds of this plant are tiny, white, flat, and physically symmetrical, with one end being flat and the other being rounded.

3.2. Feature extraction

Various characteristics of each dry bean were identified and subsequently extracted. Owing to the lack of a distinct color feature, the extracted features primarily revolved around dimensions and shape. Therefore, feature analysis was employed to determine the dimensional and shape-related features of dry beans. Geometry-related features were obtained from binary images using MATLAB software. The features extracted from each dry bean encompassed a total of 12 dimensional and four shape-related features, with the values being measured in terms of pixel count. They are the most useful dimensional features used in the classification of dry bean seeds [17]:

(a) Area (A): Area of a bean zone and the number of pixels within its boundaries.

$$A = \sum_{r,c \in R} 1, \text{ where r, c is the size of region R} \quad (1)$$

(b) Perimeter (P): Bean circumference was defined as the length of its border.
(c) Major axis length (L): distance between the ends of the longest line that can be drawn from a bean.
(d) Minor axis length (l): The longest line that can be drawn from the beam while standing perpendicular to the main axis.
(e) Aspect ratio (K): The relationship between L and l is defined.

$$K = \frac{L}{l} \quad (2)$$

(f) Eccentricity (Ec): eccentricity of an ellipse with the same moments as the region.
(g) Convex area (C): number of pixels in the smallest convex polygon that can contain the area of a bean seed.
(h) Equivalent diameter (Ed): diameter of a circle with the same area as the bean seed area;

$$d=\sqrt{\frac{4*A}{\Pi}} \quad (3)$$

(i) Extent (Ex): the ratio of pixels in the bounding box to the bean area;

$$Ex=\frac{A}{AB}, \text{ where } AB \text{ Area of bounding rectangle} \quad (4)$$

(j) Solidity (S) is also known as convexity. Ratio of pixels in convex shell to those in beans;

$$S=\frac{A}{C} \quad (5)$$

(k) Roundness (R): calculated using the formula;

$$R=\frac{4\pi A}{P^2} \quad (6)$$

(l) Compactness (CO): measures the roundness of an object;

$$CO=\frac{Ed}{l} \quad (7)$$

The shape features used in the classification of bean seeds were as [18].

$$\text{Shape factor1 (SF1)} =\frac{L}{A} \quad (8)$$

$$\text{ShapeFactor2 (SF2)} =\frac{l}{A} \quad (9)$$

$$\text{ShapeFactor3 (SF3)} =\frac{A}{\frac{L}{A}*\frac{L}{A}*\pi} \quad (10)$$

$$\text{ShapeFactor3 (SF3)} =\frac{A}{\frac{L}{A}*\frac{l}{A}*\pi} \quad (11)$$

3.3. Support vector machine

SVM is a relatively recent and promising technique for learning separate functions in pattern recognition (classification) tasks or for obtaining good function estimates in regression issues [19]. Instead of offering a regression model and an algorithm, SVMs offer a classification learning model and an algorithm [20]. The goal of employing an SVM is to identify a classification criterion (i.e., a decision function) that can accurately classify unknown data with good generalization at the testing stage [21]. A training set is said to be linearly separable if a linear discriminant function exists, with a sign corresponding to the class of each training example. Typically, infinitely separating hyperplanes exist if a training set can be linearly separated. Choose a separation hyperplane that maximizes the margin or leaves the greatest distance between it and the nearest example [22]. For example, consider a set of data points made up of n vectors xi, each linked to a value yi that indicates whether the element belongs to the fraud class (+1) or not (-1).

A linear hyperplane for a set of training data, x_i for i=1, 2, … n, is defined as;

$$w^T x+b=0 \quad (12)$$

where w is an n-dimensional vector and b is a bias term.
The optimal hyperplane is required to satisfy the constrained minimization as (13).

$$\min_{w,b} \ \emptyset(w)=\frac{1}{2}w^T.\ w$$
$$\text{Subject to } y_i\,(x_i.\ w+b)\geq 1, \text{ for i=1, 2, … m} \quad (13)$$

In these cases, the constraint can be enforced using a lagrange multiplier(α), as shown (14).

$$L\ (w, b, \alpha)=\frac{1}{2}w^T.\ w-\sum_{i=1}^{n}(y_i\,(w^T x_i+b)-1) \quad (14)$$

To find the minimum of (3) over w, b (while fixing all α_i), we set the gradient vector to zero.

$$W= \sum_{i=1}^{n} \alpha_i x_i y_i \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0 \quad (15)$$

The solution of the cost function, which yields the maximum hyperplane utilized to categorize citrus fruit types, is as (16).

$$L(w, b, \alpha) = \sum_{i=1}^{n} \alpha_I - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j$$
$$\text{Subject to } \sum_{i=1}^{m} \alpha_i y_i, \forall \alpha_i \geq 0 \quad (16)$$

In the absence of linear separability in the training set, optimal hyperplanes are ineffective for classifying credit card fraud detection. Therefore, soft margins were introduced to address this problem. In addition, slack variables are introduced to disregard some constraints. That is, the margin offers training points. Therefore, we wanted their margin of penetration to be as small as possible. Therefore, the margin should be composed of only the fewest possible points;

$$\min_{w,b,\varepsilon} \phi(w) = \frac{1}{2} w^T . w + C \sum_{i=1}^{m} \varepsilon_i$$
$$\text{Subject to } y_i (w, x_i + b) \geq 1 - \varepsilon_i, \text{ for } i=1, 2, \ldots n \quad (17)$$

where ε_i is the slack variable and the C penalty parameter of the error term. The solution of (6) is;

$$L(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j$$
$$\text{Subject to } 0 \leq \alpha_i \leq C, I=1, 2\ldots, m \quad (18)$$
$$\sum_{i=1}^{n} \alpha_i y_i, \forall \alpha_i \geq 0$$

higher values of C produced lower biases and higher variances, whereas lower values of C provided the opposite effects: higher biases and higher variances. Therefore, it is necessary to determine the ideal C value for the trade-off between the bias and variance.

3.4. Optimization techniques

To achieve optimal results using deep-learning algorithms, it is essential to tune their parameters appropriately. Selecting a powerful deep-learning algorithm and adjusting its parameters are critical for developing a high-accuracy classification model. However, manually performing parameter optimization can be extremely time consuming, particularly when the learning algorithm has many parameters. One of the biggest challenges in setting up an SVM model is selecting an appropriate kernel function and determining optimal parameter values. Poor classification outcomes often result from inappropriate parameter settings. Therefore, it is crucial to identify the most appropriate parameter settings for an SVM model to ensure the best possible classification results [23], [24].

Hyperparameters in neural networks are parameters that cannot be directly learned from the training data, but must be set beforehand. They can greatly impact the performance of the neural network; therefore, optimizing them to achieve better training results is important. The number of layers, learning rate, momentum, mini-batch size, activation functions, regularization parameters, and dropout rate are some of the hyperparameters that can be optimized to improve the neural network performance [25]. However, optimizing hyperparameters requires exploring a large search space, which can be computationally expensive and time consuming. This is especially true when training deep neural networks, which often have a large number of tuning hyperparameters. Nevertheless, these techniques can save time and resources, often leading to improved training outcomes [26]. This study employed three methods to tune the SVM parameters: Bayesian optimization, grid search, and random search techniques. The aim of parameter tuning is to reduce the testing time while maintaining test accuracy, and a feasible parameter setting must achieve an accuracy comparable to that of the original model.

3.4.1. Grid search optimization technique

The grid search method systematically tests a predefined subset of hyperparameter values by exhaustively evaluating all the possible combinations of hyperparameters within a specified range. The hyperparameters are typically defined using the minimum, maximum, and step sizes for each hyperparameter. The search can be conducted on a linear, quadratic, or logarithmic scale. The performance metrics were then used to evaluate the performance of each combination of hyperparameters. The grid search tunes the SVM hyper-parameters (such as C, γ, and degree) using a performance metric based on the cross-validation (CV) technique. The primary objective is to determine the optimal hyperparameter values that lead

to an accurate prediction of new data by the classifier while avoiding overfitting. To determine the optimal values of C and γ using k-fold cross-validation, we divided the available data into k subsets (typically k=10). One subset was used as the validation set and the model was trained on the remaining k-1 subsets. Next, we evaluated the performance of the SVM classifier on the validation set by using different values of C, γ, and other parameters. Next, we repeated this process for all k possible partitions of the data, and calculated the average cross-validation error for each combination of hyperparameters. Finally, the combination that achieved the highest cross-validation accuracy (or lowest error) was selected and used to train the SVM on the entire dataset [27], [28].

3.4.2. Random search optimization technique

A grid search can result in unnecessary computations of the hyperparameters that have little impact on the performance of the model. A more efficient technique called random search can be used to overcome this problem. Hyperparameters were randomly sampled from the search space in a random search, and the configuration with the best performance was selected. The intuition behind this approach is that a sufficiently large set of random samples is likely to include the global optimum or a close approximation with high probability. Furthermore, a random search usually proceeds more quickly than a grid search. Because it can be used with a computer cluster that is susceptible to failure and enables the experimenter to change the "resolution" on the fly, random search is more useful than grid search. Because the trials are i.i.d., which is not the case for a grid search, it is possible to add new trials to the set and disregard failed trials. Of course, sequential optimisation, which automates manual search, can probably enhance random search [29], [30].

3.4.3. Bayesian optimization technique

Bayesian optimization uses a probabilistic model to approximate the function that maps hyperparameters to performance measures. By iteratively selecting hyperparameters for evaluation based on this model and updating the model with the results of those evaluations, Bayesian optimization can efficiently search the hyperparameter space and find good solutions with fewer evaluations than grid or random searches. The Bayesian optimization algorithm updates the prior distribution as new sampling points are tested using this information to create a posterior distribution that better represents the true objective function. Based on this updated distribution, the algorithm determines the next set of hyperparameters to be tested, targeting the area where the global optimum is most likely to be found. This iterative process continued until a satisfactory set of hyperparameters was obtained. It may also be used to functions that require a lot of computation time, have challenging derivatives, or are not convex. The objective of optimisation in this study is to identify the greatest value at the sample point for an unidentified function, f;

$$x^{+} = \arg\max_{x \in A} f(x) \tag{19}$$

where a represents the x search space. The foundation for Bayesian optimisation is the Bayes' theorem, which states that given evidence data E, the posterior probability P (M|E) of a model M is equal to the likelihood P (E|M) of overserving E given the model M multiplied by the prior probability.

$$P(M|E) \alpha P(E|M) P(M) \tag{20}$$

Bayesian optimization balances exploration and exploitation by selecting the next set of hyperparameters for testing based on the balance between their potential for improvement and uncertainty. This approach allows the algorithm to efficiently search the hyperparameter space and find the global optimum without becoming trapped in a local optimum. As a result, Bayesian optimization has become a popular and effective approach for optimizing hyperparameters in ML models. Bayesian optimization outperforms grid and random searches when determining optimal hyperparameters with fewer samples. This is because it uses previous information to guide the search toward promising areas of the hyperparameter space, while still exploring other areas to ensure that the global optimum is not missed. By balancing exploration and exploitation, Bayesian optimization can efficiently determine the best set of hyperparameters [31], [32].

4. RESULTS AND DISCUSSION

In this experiment, three different optimization techniques were used with the SVM model to compare the performance of the SVM. The optimization techniques are random search, grid search, and bayesian optimization. The proposed method was implemented on an HP Victus laptop with a 12th generation Intel Core i7 CPU running on the Windows 11 operating system, including a built-in NVIDIA graphics processing unit, and MATLAB 2022a software. The training dataset comprised 70% of the data, the validation dataset comprised 20%, and the remaining 10% were reserved for testing the augmented dataset. After validating the

data, the validation accuracy was computed and the test accuracy was calculated after testing the data. The minimum classification error is determined after processing the data. The SVM models using all three optimization techniques were processed similarly. The four parameters of validation accuracy, test accuracy, total cost, and minimum classification error were compared to determine the best optimization technique using SVM.

The random search optimization technique randomly selects hyperparameters from a defined search space and evaluates the model's performance using cross-validation. The advantage of random search is that it can explore a wider range of hyperparameters than grid search. Among the three optimization techniques, the random search optimization technique with an SVM was validated and tested.

The random search optimization technique achieved validation and test accuracy of 92.62 and 92.21, in Figure 1, respectively. The minimum classification error in this model was 0.073836. The confusion matrix and AUC of the validated data are shown in Figures 1(a) and 1(b), respectively. The confusion matrix and AUC of test data are shown in Figures 1(c) and 1(d) continuously.

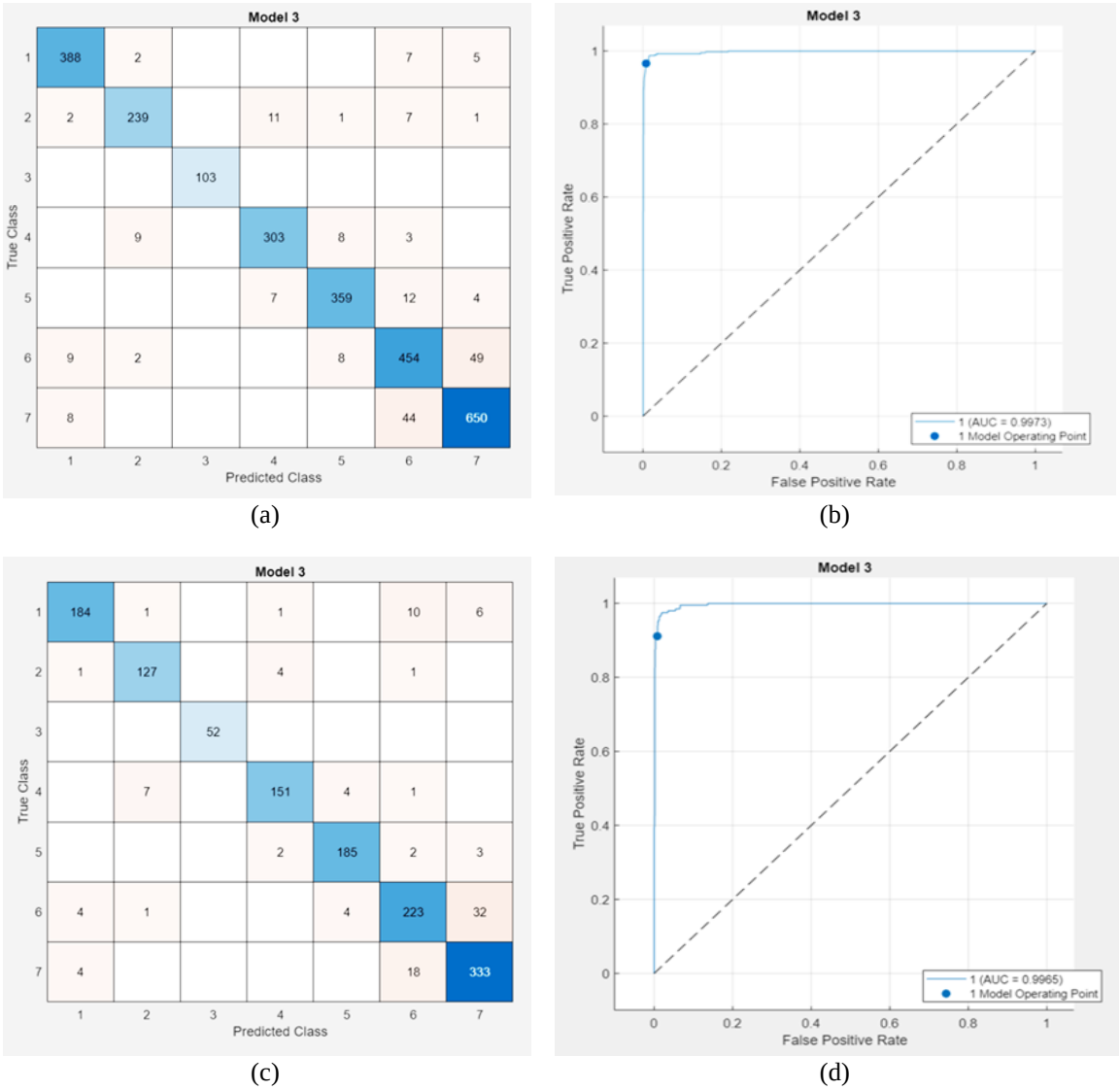

Figure 1. Performance of random search optimization with SVM; (a) confusion matrix of validated data, (b) area under the curve (AUC) of validated data, (c) confusion matrix of test data, and (d) AUC of test data

Bayesian optimization has emerged as a popular technique for addressing optimization problems in various domains where traditional numerical methods fall short. For example, one of its popular applications

is hyperparameter tuning, which aims to minimize the validation error of a machine-learning algorithm by tuning its hyperparameters. The process involves evaluating the objective function, which is the validation error, by training the ML model and assessing its performance on the validation data. In this context, Bayesian optimization was applied in conjunction with an SVM classifier to improve the accuracy of dry bean identification. The Bayesian optimization technique achieved validation accuracy and test accuracy of 93.06 and 92.29 in Figure 2, respectively. The minimum classification error in this model was 0.069384. The confusion matrix and AUC of the validated data are shown in Figures 2(a) and 2(b), respectively. The confusion matrix and AUC of test data are shown in Figures 2(c) and 2(d) continuously. Grid search is a systematic search based on a predefined subset of the hyperparameter space. It partitions the range of parameters to be optimized into a grid and evaluates all the points to obtain the optimal parameters. The grid search optimizes the SVM parameters using cross-validation as the performance metric.

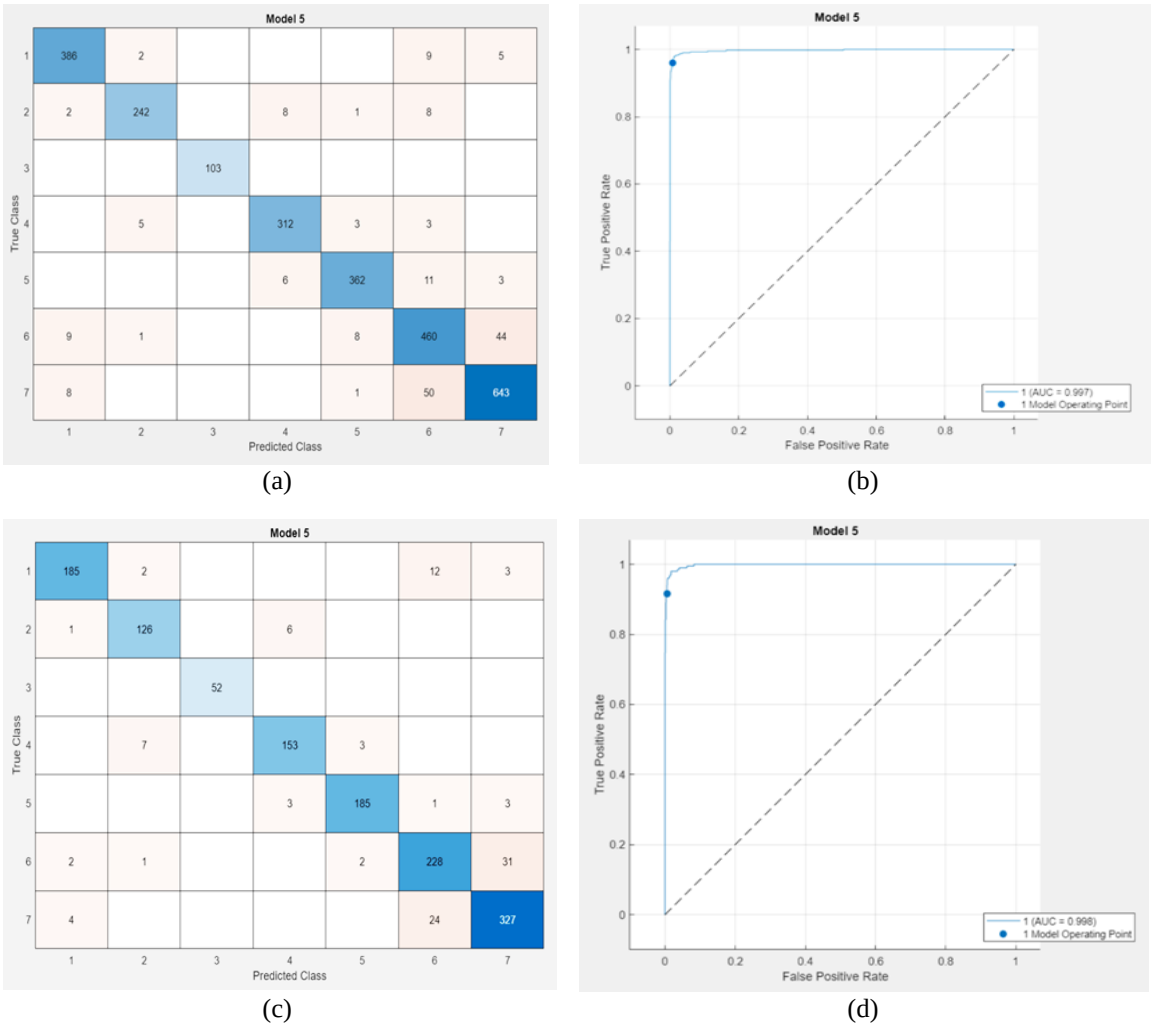

Figure 2. Performance of bayesian optimization with SVM; (a) confusion matrix of validated data (b) AUC of validated data, (c) confusion matrix of test data, and (d) AUC of test data

The grid search optimization technique achieved validation and test accuracy of 92.58 and 92.36, respectively. The minimum classification error in this model was 0.074208. The performance of grid search optimization with SVM is illustrated in Figure 3. The confusion matrix and AUC of the validated data are shown in Figures 3(a) and 3(b), respectively. The confusion matrix and AUC of test data are shown in Figures 3(c) and 3(d) continuously. The detailed validation accuracy, test accuracy, total cost, and minimum classification error of the random search, bayesian, and grid search optimization techniques with SVM are shown in Table 2.

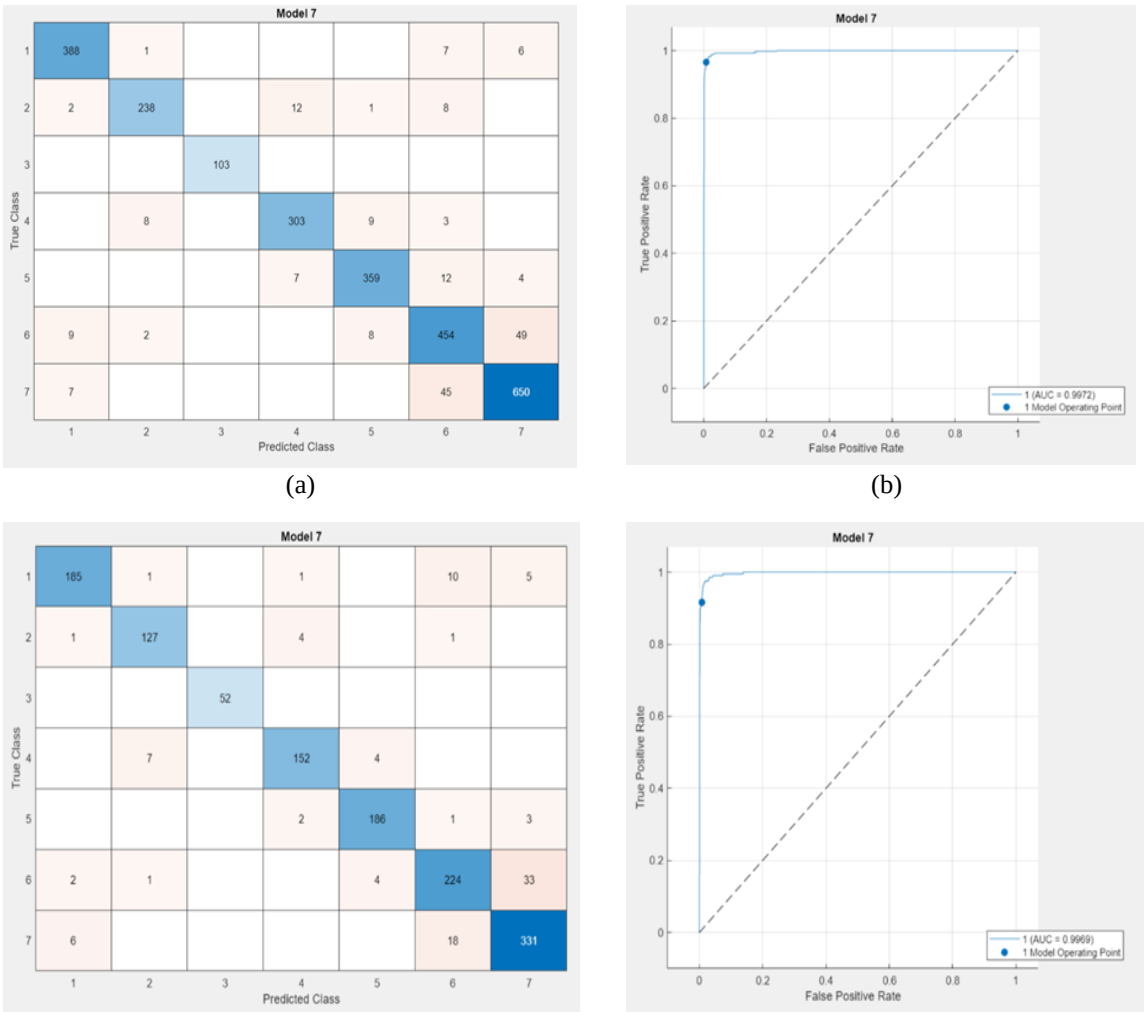

Figure 3. Performance of grid search optimization with SVM; (a) confusion matrix of validated data, (b) AUC of validated data, (c) confusion matrix of test data, and (d) AUC of test data

Table 2. Performance of SVM concerning different optimization techniques

Sl. no.	Optimization techniques	Validation accuracy	Total cost (validation)	Test accuracy	Total cost (test)	Minimum classification error
1	No optimization	90.02	221	89.21	219	0.19821
2	Random search	92.62	199	92.21	106	0.073836
3	Bayesian	93.06	187	92.29	105	0.069384
4	Grid search	92.58	200	92.36	107	0.074208

5. CONCLUSION

The classification of dry bean seed varieties is critical for seed uniformity and quality assurance. Compared with human inspectors, this system has two significant advantages. First, it produces a higher, reproducible, and objective sample classification. This study proposes a method for identifying dry beans using three different optimization techniques with an SVM. Without any optimization technique, the SVM method achieved a lower test accuracy than validation accuracy. Random and grid search optimization techniques have more classification errors than Bayesian optimization techniques. The validation accuracy of the Bayesian optimization technique was larger than that of the other two methods. The total validation and test costs of the Bayesian optimization technique were lower than those of the other methods. After comparing all parameters, such as validation accuracy, test accuracy, total validation cost, total test cost, and minimum classification error, it was concluded that the Bayesian optimization technique with SVM is the best optimization technique for identifying dry beans.

REFERENCES

[1] S. Mamidi *et al.*, "Investigation of the domestication of common bean (Phaseolus vulgaris) using multilocus sequence data," *Functional Plant Biology*, vol. 38, no. 12, pp. 953–967, 2011, doi: 10.1071/FP11124.

[2] V. L. B. Velasquez and P. Gepts, "RFLP diversity of common bean (Phaseolus vulgaris) in its centres of origin," *Genome*, vol. 37, no. 2, pp. 256–263, Apr. 1994, doi: 10.1139/g94-036.

[3] S. Beebe, P. W. Skroch, J. Tohme, M. C. Duque, F. Pedraza, and J. Nienhuis, "Structure of genetic diversity among common bean landraces of Middle American origin based on correspondence analysis of RAPD," *Crop Science*, vol. 40, no. 1, pp. 264–273, Jan. 2000, doi: 10.2135/cropsci2000.401264x.

[4] B. Laurent, B. Ousman, T. Dzudie, M. M. Carl, and T. Emmanuel, "Digital camera image processing of hard-to-cook beans," *Journal of Engineering and Technology Research*, vol. 2, no. 9, pp. 177–188, 2010.

[5] X. Chen, Y. Xun, W. Li, and J. Zhang, "Combining discriminant analysis and neural networks for corn variety identification," *Computers and Electronics in Agriculture*, vol. 71, no. SUPPL. 1, pp. S48–S53, Apr. 2010, doi: 10.1016/j.compag.2009.09.003.

[6] S. Krishnan, S. K. Aruna, K. Kanagarathinam, and E. Venugopal, "Identification of dry bean varieties based on multiple attributes using catboost machine learning algorithm," *Scientific Programming*, vol. 2023, pp. 1–21, Apr. 2023, doi: 10.1155/2023/2556066.

[7] M. S. Khan *et al.*, "Comparison of multiclass classification techniques using dry bean dataset," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 6–20, Jun. 2023, doi: 10.1016/j.ijcce.2023.01.002.

[8] A. Lawi and Y. Adhitya, "Classifying physical morphology of cocoa beans digital images using multiclass ensemble least-squares support vector machine," *Journal of Physics: Conference Series*, vol. 979, no. 1, Mar. 2018, doi: 10.1088/1742-6596/979/1/012029.

[9] Y. S. Taspinar, M. Dogan, I. Cinar, R. Kursun, I. A. Ozkan, and M. Koklu, "Computer vision classification of dry beans (Phaseolus vulgaris L.) based on deep transfer learning techniques," *European Food Research and Technology*, vol. 248, no. 11, pp. 2707–2725, Nov. 2022, doi: 10.1007/s00217-022-04080-1.

[10] A. J. G. Gonzales, J. A. D. Sosa, E. R. Arboleda, and E. M. Galas, "Filtering of faded coffee beans using image processing," *International Journal of Scientific and Technology Research*, vol. 8, no. 12, pp. 3634–3637, 2019.

[11] A. Nasirahmadi and N. Behroozi-Khazaei, "Identification of bean varieties according to color features using artificial neural network," *Spanish Journal of Agricultural Research*, vol. 11, no. 3, pp. 670–677, Jul. 2013, doi: 10.5424/sjar/2013113-3942.

[12] N. Alban, "Color and texture information processing to improve storage beans," *British Journal of Applied Science & Technology*, vol. 2, no. 2, pp. 96–111, 2012, doi: 10.9734/bjast/2012/796.

[13] M. Koklu and I. A. Ozkan, "Multiclass classification of dry beans using computer vision and machine learning techniques," *Computers and Electronics in Agriculture*, vol. 174, p. 105507, Jul. 2020, doi: 10.1016/j.compag.2020.105507.

[14] T. H. Nasution, M. Rumansa, and H. L. Adlin, "Designing the quality of coffee bean detection application using hue saturation intensity," *IOP Conference Series: Materials Science and Engineering*, vol. 648, no. 1, 2019, doi: 10.1088/1757-899X/648/1/012036.

[15] Y. Başol and S. Toklu, "A deep learning-based seed classification with mobile application," *Turkish Journal of Mathematics and Computer Science*, vol. 13, no. 1, pp. 192–203, Jun. 2021, doi: 10.47000/tjmcs.897631.

[16] M. Koklu and I. A. Ozkan, "Multiclass classification of dry beans using computer vision and machine learning techniques," *Computers and Electronics in Agriculture*, vol. 174, no. 1, 2020.

[17] J. Paliwal, N. S. Visen, and D. S. Jayas, "Evaluation of neural network architectures for cereal grain classification using morphological features," *Journal of Agricultural and Engineering Research*, vol. 79, no. 4, 2001, doi: 10.1006/jaer.2001.0724.

[18] A. R. Pazoki, F. Farokhi, and Z. Pazoki, "Classification of rice grain varieties using two artificial neural networks (mlp and neuro-fuzzy)," *Journal of Animal and Plant Sciences*, vol. 24, no. 1, pp. 336–343, 2014.

[19] Y. Ma and G. Guo, "Support vector machines applications," *Support Vector Machines Applications*, vol. 649, pp. 1–302, 2014, doi: 10.1007/978-3-319-02300-7.

[20] S. Suthaharan, "ML models and algorithms for big data classification," *Integrated Series in Information Systems*, vol. 36, 2016.

[21] R. Gholami and N. Fakhari, "Support vector machine: principles, parameters, and applications," in *Handbook of Neural Computation*, Elsevier, 2017, pp. 515–535.

[22] L. Bottou and C. Lin, "Support vector machine solvers large scale kernel machines," *Large scale kernel machines*, vol. 3, no. 1, 2007.

[23] D. J. Kalita, V. P. Singh, and V. Kumar, "A novel adaptive optimization framework for SVM hyper-parameters tuning in non-stationary environment: A case study on intrusion detection system," *Expert Systems with Applications*, vol. 213, p. 119189, Mar. 2023, doi: 10.1016/j.eswa.2022.119189.

[24] Y. Yang *et al.*, "The innovative optimization techniques for forecasting the energy consumption of buildings using the shuffled frog leaping algorithm and different neural networks," *Energy*, vol. 268, p. 126548, Apr. 2023, doi: 10.1016/j.energy.2022.126548.

[25] Q. Li, Z. Luo, H. Chen, and C. Li, "An overview of deeply optimized convolutional neural networks and research in surface defect classification of workpieces," *IEEE Access*, vol. 10, pp. 26443–26462, 2022, doi: 10.1109/ACCESS.2022.3157293.

[26] B. Bischl *et al.*, "Hyperparameter optimization: foundations, algorithms, best practices, and open challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 13, no. 2, Mar. 2023, doi: 10.1002/widm.1484.

[27] I. Syarif, A. Prugel-Bennett, and G. Wills, "SVM parameter optimization using grid search and genetic algorithm to improve classification performance," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 14, no. 4, p. 1502, Dec. 2016, doi: 10.12928/telkomnika.v14i4.3956.

[28] H. A. Fayed and A. F. Atiya, "Speed up grid-search for parameter selection of support vector machines," *Applied Soft Computing Journal*, vol. 80, pp. 202–210, Jul. 2019, doi: 10.1016/j.asoc.2019.03.037.

[29] S. Andradóttir, "Chapter 20 an overview of simulation optimization via random search," *Handbooks in Operations Research and Management Science*, vol. 13, no. C, pp. 617–631, 2006.

[30] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of machine learning research*, vol. 13, no. 2, pp. 281–305, 2012.

[31] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019, doi: 10.11989/JEST.1674-862X.80904120.

[32] A. H. Victoria and G. Maragatham, "Automatic tuning of hyperparameters using Bayesian optimization," *Evolving Systems*, vol. 12, no. 1, pp. 217–223, Mar. 2021, doi: 10.1007/s12530-020-09345-2.

BIOGRAPHIES OF AUTHORS

Nabin Kumar Naik received his B.Tech. degree in Electronics and Telecommunication engineering from Kalinga Institute of Industrial Technology, Deemed University, Bhubaneswar, India in 2006 and M.Tech. degree in VLSI Design from Anna University, Chennai, India in 2010. He is currently pursuing Ph.D. at the Department of Electronics, Sambalpur University, Burla, India. His areas of interest are image processing, artificial intelligence, machine learning, deep learning and very large-scale integration. He can be contacted at email: nabin.naik@silicon.ac.in.

Prabira Kumar Sethy senior member IEEE currently working as an Assistant Professor in the Department of Electronics at Sambalpur University since 2013. He has nine years of teaching, research and administrative experience and four years of Industry experience. Previously he worked as Engineer in Doordarshan, Prashar Bharati, from 2009 to 2013. He has received his Ph.D. and M. Tech degree from Sambalpur University and IIT (ISM) Dhanbad, respectively. His research area is image processing, machine learning, and deep learning. He has published 80 research papers in different reputed journals and conferences. In addition, he has two patents. He is a senior member of IEEE. He is a frequent reviewer of many journals and session chair of international conferences. He can be contacted at email: prabirsethy.05@gmail.com.

Rajat Amat currently pursuing his Master of Technology (M.Tech) in Embedded System Design from SUIIT, Sambalpur University, Jyoti Vihar, Burla. He has completed his Bachelor of Science (B.Sc.) in Physics and Master of Science (M.Sc.) in Electronics and Telecommunication (ETC) from Utkal University, Vani Vihar, Bhubaneswar and Gangadhar Meher University, Amruta Vihar, Sambalpur respectively. His research area is digital image processing, deep learning and VLSI. He can be contacted at email: amatrajat4@gmail.com.

Santi Kumari Behera senior member IEEE currently working as an assistant professor in the Department of Computer Science and Engineering at Veer Surendra Sai University of Technology since 2015. Her research area is image processing, machine learning, and deep learning. She has published 70 research papers in different reputed journals and conferences. In addition, he has one patent and one copyright. She can be contacted at email: b.santibehera@gmail.com.

Dr. Preesat Biswas received a Ph.D. in Electronics Communication Engineering form Dr.C.V. Raman University, Bilaspur (C. G) in 2022, M. Tech in Electronics and Telecommunication from KEC Bhubaneswar (BPUT, ODISHA) in 2013, and B. Tech in Electronics and Telecommunication from GKCET (BPUT, ODISHA) in 2009. He was also an inventor and holds Three patents. He published seven books. He authored and co-authored over 14 refereed journal and conference papers in various areas of wireless communication, signal processing, image processing, adaptive signal processing, speech processing, and deep learning. He can be contacted at email: preesat.eipl@gmail.com.