Categorizing the Students' Activities for Automated Exam Proctoring Using Proposed Deep L2-GraftNet CNN Network and ASO Based Feature Selection Approach

TANZILA SABA\textsuperscript{1}, (Senior Member, IEEE), AMJAD REHMAN\textsuperscript{1}, (Senior Member, IEEE), NOR SHAHIDA MOHD JAMAIL\textsuperscript{1}, SOUAD LARABI-MARIE-SAINTE\textsuperscript{1}, MUDASSAR RAZA\textsuperscript{2}, (Senior Member, IEEE), AND MUHAMMAD SHARIF\textsuperscript{2}, (Senior Member, IEEE)

\textsuperscript{1}Artificial Intelligence & Data Analytics Lab (AIDA), College of Computer & Information Sciences (CCIS), Prince Sultan University, Riyadh 11586, Saudi Arabia
\textsuperscript{2}Department of Computer Science, COMSATS University Islamabad, Wah Campus, Wah Cantonment 47040, Pakistan

Corresponding authors: Muhammad Sharif (muhhammadsharifmalik@yahoo.com) and Amjad Rehman (rkamjad@gmail.com)

This work was supported by the research project [Exam Activity Monitoring by Analyzing Students Behavior Using Deep Learning Model], Prince Sultan University, Saudi Arabia, under Grant SEED-CCIS-2020\{25\}.

ABSTRACT
Exam proctoring is a hectic task i.e., the monitoring of students’ activities becomes difficult for supervisors in the examination rooms. It is a costly approach that requires much labor. Also, it is a difficult task for supervisors to keep an eye on all students at a time. Automatic exam activities recognition is therefore necessitating and a demanding field of research. In this research work, categorization of students’ activities during the exam is performed using a deep learning approach. A new deep CNN architecture with 46 layers is proposed which contains the characteristics of deep AlexNet and SqueezeNet. The model is engineered first with slight modifications in AlexNet. After then, the squeezed branch-like structure of SqueezeNet is grafted/embedded at two locations in the modified AlexNet architecture. The model is named as L2-GraftNet because of dual grafting blocks. The proposed model is first converted to a pre-trained model by performing its training with SoftMax classifier on the CIFAR-100 dataset. Afterwards, the features of the dataset prepared for exam activities categorization are extracted from the above-mentioned pre-trained model. The extracted features are then fed to the atom search optimization (ASO) approach for features optimization. The optimized features are passed to different variants of SVM and KNN classifiers. The best performance results attained are on the Fine KNN classifier with an accuracy of 93.88\%. The satisfactory results prove the robustness of the proposed framework. Also, the proposed categorization provides a base for automated exam proctoring without the need for proctors in the exam halls.

INDEX TERMS
Action recognition, ASO, CNN, exams, L2-GraftNet, students.

I. INTRODUCTION
Students’ action modeling [1] a new field of study. It works on learning ‘how’ and ‘why’ of activities that take the human view of knowledge. In any instructional method, evaluation plays a key part as a means of measuring the students’ understanding of the topics associated with educational outcomes [2]. Student deception through real-time examinations, regardless of the degree of advancement of the community, is a common issue across the globe. There seems to be no widely held description of deception in exams, as with the principle of unprofessional conduct, but many types of actions are normally considered as necessary indicators of student academic deception, all of which have in particular that they are unethical acts aimed at enhancing the perceived success [3]. The conventional exam inspection in institutions is based on a written paper and involves the invigilators’ appearance on the spot. Figure 1 shows two different scenes.
of exam activities. The invigilators wander around desks to keep checking for any ethical activity by students. Manual tracking is a challenging method for assessing the behavior of students during written exams. Because of intellectual dishonesty, many students have captured annually [4].

In tests, deception means violating the exam rules. In a multitude of ways, students trick the examiner, including covering notes, transmitting hidden messages by hand and body movements, monitoring nearby student documents, passing notes to other students while the examiner is not watching, and illegally using technological devices such as mobile phones, calculators and digital watches [5], [6]. There are several explanations for the deceit of students in examinations. These causes are greatly divided into (a) psychiatric and (b) social problems [7], [8]. Psychiatric difficulties include a lack of time and anxiety factors in examinations. Family peer pressure and feeling incompetent are linked to psychiatric disorders [9]. Exam deception is implicated in the pathogenesis of potential dishonest conduct, both in higher education and subsequently in professional life [10]–[13]. Also, the use of unethical activities in exams can lead to serious implications for both individuals and society. Some of such implications contain [14]: (a) In one social background, developing a deceptive habit is therefore prone to leaking over to one another, (b) The idea that stealing contributes to an erroneous competence appraisal of the student often implies that the person’s basic requirements for learning skills are negatively impacted, and (c) reduction in reputation of the corresponding organization. To preserve the student’s integrity actively during the assessment process, real-time surveillance is very needful. This can help in the reduction of invigilators’ tough duty. Also, it will be difficult for the students to dodge the camera in front of the students as they can do when the invigilator sees towards other direction.

In this work, we propose an integrated exam control solution, focused on surveillance, that offers an automatic categorization of the behavior and activities of students during exams. The suggested approach is related to offline exam activity recognition (EAR) which incorporates the body movements of the exam’s participants across tests for the recognition of student acts. Six different actions regarding students’ actions during exams are considered for classification. Such actions involve: (a) seeing back, (b) watching towards the front, (c) normal performing, (d) passing gestures to other fellows, (e) watching towards left or right, and (f) other suspicious actions. 46 layers CNN-based deep learning model is proposed for image feature extraction. The extracted features are passed to a state-of-the-art feature subset selection approach. The selected features are finally supplied to multiple classifiers for the students’ activity recognition analysis. One major challenge faced in this research is the size of the prepared dataset. To cope up with this challenge, the proposed model is first pre-trained with a relatively big dataset i.e., CIFAR-100, and then the features of the prepared dataset are acquired from a fully connected layer in the proposed CNN model. Some additional challenges faced are inter-class similarity and object occlusion, which are effectively handled automatically by the deep learning approach. The comprehensive experimentation is performed by varying the number of selected features. The best recognition result with an accuracy of 93.88% is observed with 1000 selected features and a Fine KNN classifier. The major contributions are discussed as under:

1) A new 46 layers CNN-based architecture named as L2-GraftNet is proposed for image feature extraction. Because of the limited size of the dataset, the proposed model is first pre-trained on the CIFAR-100 dataset and then features are extracted from the Exam dataset.
2) ASO based Feature subset selection approach is utilized for reducing the dimensions of the extracted features.
3) The dataset CUI-EXAM is prepared to access the performance of the proposed model.
4) Numerous classifiers are employed to benchmark the best classifier. The best accuracy result proves the acceptability of the proposed framework.

Overall, the manuscript organization is described as follows: The manuscript basic terminologies and description of the proposed domain are presented in the introduction section. The existing studies are discussed in the second section regarding the literature review. Section three depicts the proposed approach and its steps in detail. The results are shown and the discussion on the results is provided in the fourth section. The manuscript is finally concluded in section five which is proceeded by references.

II. LITERATURE REVIEW

Many theoretical research studies have been carried out from old-time [9]–[11] to present on studying misconduct actions by students in exams and methods by which institutions may aim to tackle the issue [12]. Many works are presented to tackle the exam activities using traditional tactics [13]. Such tactics contain the improvement of question papers and challenging assignments given to the students. The EAR methods related to computer vision are based on two scenarios i.e., online EAR methods [14]–[17] and offline EAR approaches. The former EAR methods are related to scenarios of online exam structures where the students are present at remote locations [18]–[20]. The exams in such scenarios are monitored by proctors manually through cameras and other sensors [21], [22]. The online penalties are then imposed.
on illegal activities [23]. Some literature is found is related to online EAR using artificial intelligence and computer vision-based strategies. Bawarith et al. introduce an eye-tracking [24]–[26] mechanism to check whether the student is focused on his exam task or he is watching around [27]. Jalali and Fakhroddin [16] monitor exam activities using the webcam. The images captured from the webcam are subtracted with the reference image and if the subtracted outcome is found more than the threshold value, the activity in the image is considered cheating. Idemudia et al. [28] utilize face biometrics and head poses to detect legal and illegal exam activities. To the best of our insight, Offline automated EAR methods are very less in number in literature. Nishchal et al. [29] by combining both gesture and mood interpretation, aim to discover cheating behavior in an examination. The authors use an open pose tool for posture analysis and convolutional neural network (CNN) base Alexnet architecture for recognizing the exam deception activity. Four different classes are incorporated such as looking straight, turning backward, stretching the arms towards the back, and bending down. The total accuracy of the model is claimed as 77.8%. Aini et al. [30] employ background subtraction and pixel shift techniques to assess unusual motions during examinations. The presented dataset contains six classes to predict the type of exam deception. Hendryli and Fanany [31] employ multimodal decomposable models (MODEC) for features engineering and Multi-class Markov Chain Latent Dirichlet Allocation (MCMCLDA) approach as a predictor to categorize five classes belonging to the suspicious activities conducted during exams. These categories include visual contact, exchanging documents, making hand motions to engage with some other student, gazing at the reaction of another student, and using a visual aid; and one non-cheating practice. Ahmad et al. use two classes for cheating prediction. Speed Up Robust Features (SURF) approach is utilized for features mining. For developing further understanding of the domain, EAR can be associated with other activities recognition works. Such works include human activity recognition (HAR) [32]–[34], activities recognition in sports [35]–[37], Infants monitoring [38].

In the context of activities recognition, many methodologies found for feature extraction in the recent literature. Such methodologies include traditional feature extraction methods and deep learning approaches. Apart from feature extraction, several feature optimization approaches are used to reduce the dimensionality of the features. The mainstay methods are (a) filter approaches involving feature scoring include PCA [39], entropy [40], etc., (b) wrapper approaches involving feature learning include ant colony optimization (ACO) [41] genetic algorithms (GA) [42], etc. In the recent past, manifold learning [43], [44] is gaining attention for mapping high dimensional features to low dimensional feature spaces. Although manifold learning is easy to implement for classification procedures, the deep learning approaches show superior performances many times because of automatic feature learning [45]. The focus of this manuscript is on deep features extraction with learning-based feature selection for activity classification. Some useful classification methods include SVM [46]–[51], KNN [52]–[55] and Decision trees [56], [57].

III. MATERIAL AND METHODS
This section presents a new CNN-based model named as L2-GraftNet for EAR. The section describes the proposed model and the steps involved to perform EAR. The main steps of the proposed model include pre-training, Dataset augmentation, feature extraction from pre-trained L2-GraftNet, Feature subset selection from ASO based features subset selection, and classification. Figure 2 provides a brief overview of the proposed model of EAR using the proposed CNN network. The mentioned steps are discussed one by one in the accompanying section.

A. PROPOSED L2-GRAFNET
This work is contributed by a new CNN-based architecture, named as L2-GraftNet for EAR.

The proposed model is developed after studying two famous CNN networks such as AlexNet [58] and SqueezeNet [59]. The AlexNet comprises five convolutional and three fully connected layers. Along with these layers, the model also contains three pooling, seven ReLU, two dropouts, and SoftMax layers. The proposed model consists of 46 layers including input and output layers. Starting with AlexNet, several new layers including batch normalization and the fire modules like structures of SqueezeNet are added to form the proposed L2-GraftNet. Figure 3 depicts the architectural view of L2-GraftNet. The detailed architecture is described as follows: the input size is the same as that of AlexNet i.e., \(227 \times 227 \times 3\). The network starts with a convolution (C1) layer, after which the first grafting of fire block (GB1) (looks like Squeeze Net’s fire module) is embedded.

The GB1 contains ten layers in three branches that emerge from ReLU (R1) layer. The first branch consists of a batch normalization layer (BN2). The second branch contains four layers including, convolution (C2), Batch Normalization (BN1), convolution (C3), and leaky ReLU (LR1) layers. The third branch comprises three layers i.e., convolution (C4), BN3, and LR2 layers. The three branches are joined back with an addition (A1) layer. After the first grafting layer, two sub-blocks (B2 and B3) middled with a ReLU (R2) layer are added. The blocks B2 and B3 contain four layers in sequence (per block) that are cross channel normalization, pooling, batch normalization, and grouped convolution layers (having CN1, P1, BN4, and GC1-C5 of B1 while CN2, P2, BN5, and GC2-C6 layers of B2). Grafting of second fire block (GB2) is employed after B2. The structure of GB2 is the same as that of GB1. The proceeding layers contain four blocks (B2-B5) of convolution and ReLU layers. The blocks B3 (containing GC3-C10 and R4 layers) and B4 (containing GC4-C11 and R5 layers) lie simultaneously one after the other. Then the pooling (P3) and batch normalization (BN9)
layers are deployed. Block B5 and B6, middled by the dropout (D1) layer, contain fully connected layers (FC12 and FC13 respectively). The last four layers are (D1), fully connected (FC14), SoftMax (S), and output (O). The detail with layers configurations of L2-GraftNet is illustrated in Table 1.

A brief mathematical overview of some layers of the proposed network is given in the accompanying text. Further, CNN can be studied in detail from [60], [62].

The convolutional layer takes an image or a feature map $I_{i-1}$ as an input with $k_i$ channels, where $i$ belongs to the number of the layer [63]. The output will contain $k'_{i}$ channels and all channels are calculated as:

$$I_{k',i} = \mathcal{G}_i \left( \sum_{k} \delta_{1,k'k} * I_{k,i-1} + B_{k',i} \right)$$  \hspace{1cm} (1)

where * depict convolution function, $\delta$ is a filter with depth $k'_{i}$. $B$ and $\mathcal{G}$ represent nonlinear functions applied cell-wise.

The two types of ReLU layers are used in the proposed network such as normal ReLU and leaky ReLU [64]. The normal ReLU converts all negative values to zero mathematically represented as:

$$I_{x,y} = \text{maximum}(0, I_{x,y})$$  \hspace{1cm} (2)

where x, y are row and column numbers of the input matrix $I$. Instead of being zero, Leaky ReLU has a slight curve for values less than 0. When $x$ is less than 0, a leaky ReLU can have $y = 0.01x$.

Two types of normalization are employed in the proposed architecture such as BN and CN. BN is a technique that rationalizes channel neurons around the set quantity of the small batch. It measures the average and variance in chunks for each characteristic. It then extrapolates the average and splits the characteristics by taking the standard deviation of the small-batch [65]. Mathematically, the average of the batch is taken as:

$$\text{Average}_{\text{Batch}} = \frac{1}{k} \sum_{j=1}^{k} f_j$$  \hspace{1cm} (3)
where $\text{Batch} = \{f_1 \ldots f_k\}$ and $f$ represents a feature. The variance of the small-batch is given as

$$\text{Variance}_{\text{Batch}} = \frac{1}{k} \sum_{j=1}^{k} (f_j - \text{Average}_{\text{Batch}})^2$$  \hspace{1cm} (4)$$

The features are then normalized using the following equation

$$\hat{f}_j = \frac{f_j - \text{Average}_{\text{Batch}}}{\sqrt{\text{Variance}_{\text{Batch}} + \varphi}}$$  \hspace{1cm} (5)$$

where $\varphi$ is the constant used for mathematical steadiness. CN helps in generalization [58]. Promoting directional suppression is the rationale for using CN. In neuroscience, this phenomenon refers to the tendency of neurons to minimize the neighbors’ activity. This directional suppression in neural networks aims to perform spatial image contrast improvement such that the peak threshold value of pixels for the proceeding layers is used locally as the perturbation. The neighbor in CN represents the cross channel. Mathematically CN is depicted as:

$$\tilde{f} = \frac{f}{\left(\sum_{o=1}^{d} \delta_{o}\right)^{\frac{3}{2}}}$$  \hspace{1cm} (6)$$

where $f$ is feature before CN and $\tilde{f}$ is feature value after CN. Hyperparameters $\sum, \delta$ and $\delta_{o}$ depict the values used for normalization. $S$ shows the sum of square value and $\omega$ is dimensions of the channel.

**B. PRE-TRAINING THE NETWORK AND FEATURE EXTRACTION**

The dataset for training the proposed model is not huge to support deep learning. For this purpose, the proposed CNN model is first pre-trained on CIFAR-100 [66] dataset. CIFAR-100 comprises 100 categories. Each category
TABLE 1. The detail with layers configurations of L2-GraftNet.

| Layer # | Layer name | Feature maps Size | Filter depth | Stride | Padding | Pooling window size/other values |
|---------|------------|-------------------|--------------|--------|---------|----------------------------------|
| 1       | Input      | $227 \times 227 \times 3$ | $11 \times 11 \times 3 \times 96$ | [4 4]  | [0 0 0 0] | -                                |
| 2       | C1         | $55 \times 55 \times 96$ | -            | -      | -       | -                                |
| 3       | R1         | $55 \times 55 \times 96$ | -            | -      | -       | -                                |
| 4       | C4         | $55 \times 55 \times 96$ | $5 \times 5 \times 96 \times 96$ | [1 1]  | Same    | -                                |
| 5       | BN2        | $55 \times 55 \times 96$ | -            | -      | -       | -                                |
| 6       | BN3        | $55 \times 55 \times 96$ | -            | -      | -       | -                                |
| 7       | C2         | $55 \times 55 \times 48$ | $1 \times 1 \times 96 \times 48$ | [1 1]  | Same    | -                                |
| 8       | BN1        | $55 \times 55 \times 48$ | -            | -      | -       | -                                |
| 9       | C3         | $55 \times 55 \times 96$ | $11 \times 11 \times 48 \times 96$ | [1 1]  | Same    | -                                |
| 10      | LR1        | $55 \times 55 \times 96$ | -            | -      | -       | Scale 0.01                       |
| 11      | LR2        | $55 \times 55 \times 96$ | -            | -      | -       | Scale 0.01                       |
| 12      | ADD1       | $55 \times 55 \times 96$ | -            | -      | -       | -                                |
| 13      | CN1        | $55 \times 55 \times 96$ | -            | -      | -       | -                                |
| 14      | P1         | $27 \times 27 \times 96$ | -            | [2 2]  | [0 0 0 0] | Max pooling $3 \times 3$       |
| 15      | BN4        | $27 \times 27 \times 96$ | -            | -      | -       | -                                |
| 16      | GC1(C5)    | $27 \times 27 \times 256$ | Two groups of $5 \times 48 \times 128$ | [1 1]  | [2 2 2 2] | -                                |
| 17      | R2         | $27 \times 27 \times 256$ | -            | -      | -       | -                                |
| 18      | CN2        | $27 \times 27 \times 256$ | -            | -      | -       | -                                |
| 19      | P2         | $13 \times 13 \times 256$ | -            | [2 2]  | [0 0 0 0] | Max pooling $3 \times 3$       |
| 20      | BN5        | $13 \times 13 \times 256$ | -            | -      | -       | -                                |
| 21      | GC2(C6)    | $13 \times 13 \times 384$ | $3 \times 3 \times 256 \times 384$ | [1 1]  | [1 1 1] | -                                |
| 22      | R3         | $13 \times 13 \times 384$ | -            | -      | -       | -                                |
| 23      | BN7        | $13 \times 13 \times 384$ | -            | -      | -       | -                                |
| 24      | C7         | $13 \times 13 \times 192$ | $1 \times 1 \times 384 \times 192$ | [1 1]  | Same    | -                                |
| 25      | BN6        | $13 \times 13 \times 192$ | -            | -      | -       | -                                |
| 26      | C9         | $13 \times 13 \times 384$ | $3 \times 3 \times 384 \times 384$ | [1 1]  | Same    | -                                |
| 27      | BN8        | $13 \times 13 \times 384$ | -            | -      | -       | -                                |
| 28      | LR4        | $13 \times 13 \times 384$ | -            | -      | -       | Scale 0.01                       |
| 29      | C8         | $13 \times 13 \times 384$ | $5 \times 5 \times 192 \times 384$ | [1 1]  | Same    | -                                |
| 30      | LR3        | $13 \times 13 \times 384$ | -            | -      | -       | Scale 0.01                       |
| 31      | ADD2       | $13 \times 13 \times 384$ | -            | -      | -       | -                                |
| 32      | GC3(C10)   | $13 \times 13 \times 384$ | Two groups of $3 \times 3 \times 192 \times 192$ | [1 1]  | [1 1 1 1] | -                                |
| 33      | R4         | $13 \times 13 \times 384$ | -            | -      | -       | -                                |
| 34      | GC4(C11)   | $13 \times 13 \times 256$ | Two groups of $3 \times 3 \times 192 \times 128$ | [1 1]  | [1 1 1 1] | -                                |
| 35      | R5         | $13 \times 13 \times 256$ | -            | -      | -       | -                                |
| 36      | P3         | $6 \times 6 \times 256$ | -            | [2 2]  | [0 0 0 0] | Max pooling $3 \times 3$       |
| 37      | BN9        | $6 \times 6 \times 256$ | -            | -      | -       | -                                |
| 38      | FC12       | $1 \times 1 \times 4096$ | -            | -      | -       | -                                |
| 39      | R6         | $1 \times 1 \times 4096$ | -            | -      | -       | -                                |
| 40      | D1         | $1 \times 1 \times 4096$ | -            | -      | -       | 50% Dropout                      |
| 41      | FC13       | $1 \times 1 \times 4096$ | -            | -      | -       | -                                |
| 42      | R7         | $1 \times 1 \times 4096$ | -            | -      | -       | -                                |
| 43      | D2         | $1 \times 1 \times 4096$ | -            | -      | -       | -                                |
| 44      | FC14       | $1 \times 1 \times 100$  | -            | -      | -       | -                                |
| 45      | SOFTMAX    | $1 \times 1 \times 100$  | -            | -      | -       | -                                |
| 46      | OUTPUT     | -                  | -            | -      | -       | -                                |

contains 500 training and 100 testing images. In this work, both training and testing images are combined, and training is performed on 600 images per category. After converting the proposed network to a pre-trained model, the features are extracted on the prepared exam dataset. The features on all dataset images (total 11340 images) are acquired from the FC12 layer. The total feature acquired from this layer is 4096 per image. Thus, the total size of the feature matrix becomes $11340 \times 4096$.

C. FEATURES SELECTION WITH ASO

The extracted features in feature extraction phases are huge in number per image (4096). This can slow the training phase for classification. Besides, issues like ambiguous features and the curse of dimensionality can arise, as a result, a decrease in performance can occur. To address this problem, the extracted features are passed to the ASO algorithm for feature optimization. ASO is centered on molecular mechanics in which, by interaction, the atoms in the query areas associate.
one another. When two forces are identical at a displacement \( d_{ij} = 1.12T \), the period of stabilization is reached. Here, \( \Gamma \) depicts the length scale for atoms collision. The density of a given atom is theoretically a response, in which a heavy atom is stronger than a weaker atom [67]. The densities of atoms are impacted by the passage of atoms. The optimization function is used as follows:

\[
Root\ Root\ \ Mean\ \ Square\ \ Error\ =\ \ RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\rho_i - A_i)^2} \tag{7}
\]

where \( N \) illustrates the number of training samples, \( \rho_i \) is the \( i \)th categorization outcome and \( A_i \) is the actual value to be predicted. The main aim of the ASO algorithm is to lessen the value of the optimization function. An atom community of \( N \) is suggested (represented as \( \xi_i = \{\xi_{i1}, \xi_{i2}, \ldots, \xi_{iS}\}, i = 1, 2, \ldots, N \) in \( S \)-dimensional space for minimization solution. In this space, the Euclidean distance is represented as \( d_{ij} \), \( i \) and \( j \) depict the number of the atoms. By the force specified \( Force_{ij} \) (which is the change known as Lennard-Jones (L-J) potential), atoms communicate with one another as follows:

\[
Force_{ij} = \frac{24}{\Gamma^2} \left[ \frac{1}{2} \left( \frac{\xi}{d_{ij}} \right)^{14} - \left( \frac{\xi}{d_{ij}} \right)^{8} \right] \tag{8}
\]

The potential governs the atoms’ communication and thus, with each repetition, decides the atoms’ arrangements. The overall communication of \( i \)th atom is shown as:

\[
Force_i = \sum_{j=1, j \neq i}^{T} Force_{ij} \tag{9}
\]

For each cycle, the density fitness value of the atoms is reassessed as follows:

\[
Density\ Fitness_{ik} = e^{-\frac{\psi_{ik} - \psi_{ik}}{\psi_{ik} - \psi_{ik}}} \tag{10}
\]

where \( \xi \) shows the iteration number. The symbols \( b \) and \( w \) represent best, worst fitness values, respectively. Based on density fitness value, the density of \( i \)th atom is calculated as:

\[
Density_{ik} = e^{-\frac{\sum_{k=1}^{S} Density\ Fitness_{ik}}{\sum_{k=1}^{S} Density\ Fitness_{ik}}} \tag{11}
\]

The speeds of all atoms are revised by using the following criteria on each iteration:

\[
Velocity_{ik}(\xi+1) = random_{ik} Velocity_{ik}(\xi) + Acceleration_{ik} \tag{12}
\]

where \( random_{ik} \) represent a random number (0 to 1). The location of atoms is shown as:

\[
Location_{ik}(\xi+1) = Location_{ik}(\xi) + Velocity_{ik}(\xi) \tag{13}
\]

When the predetermined cumulative repetitions are achieved or the optimization function approaches a value lower than optimal, the search process stops. More detail on ASO can further be studied at [68], [69].

D. CLASSIFICATION

After features selection, the selected features are moved forward to, various classifiers. The chosen classifiers are the variants of (SVM) [70] and K-Nearest neighbor [71] classifiers. The variants of SVM include linear SVM(LSVM) [72], quadratic SVM (QSVM) [73], fine Gaussian SVM (FGSVM) [74], medium Gaussian SVM (MG SVM), coarse Gaussian SVM (CGSM), and cubic SVM (CSVM) [75]. SVM classifier and its kernel can be explored in [49]–[51]. The types of KNN classifiers used are cosine KNN (CKNN), coarse KNN (CrKNN), and fine KNN (FKNN)[55]. The detailed explanation of KNN and its variants can further be studied from [52]–[55]. The classifiers are evaluated on numerous performance evaluation metrics. Based on experimentation, the two best classifiers observed are CSVM and FKNN. FKNN is observed with highest accuracy while CSVM is found with best accuracy at second place. The detailed results and experiments are presented in the results section.

E. DATASET AUGMENTATION

Deep learning requires a huge dataset for robust performance at the training level. To improve the size of the dataset, augmentation is performed. Four different types of augmentations are chosen as mirroring, adding Gaussian noise, adding salt and pepper noise, and color-shifting (see Figure 4 for visualization). This increases the size of the dataset to four times.

IV. RESULTS AND DISCUSSION

The major aim of this study is to develop a deep CNN network that will work fine with the exam dataset. The proposed Deep L2-GraphNet CNN Network is used only to extract robust features on which after applying feature selection, SVM, KNN, and Tree-based classifiers are used to observe the system performance. This section represents the findings and the
analysis of the proposed framework. The section starts with the
description of the experimental environment setup. After
then, the dataset is described. The performance evaluation
protocol is then presented with dataset training visualizations.
At the last, the experiments are explained in detail. The
experiments are performed with windows 10 platform on core
i5 machine with 8GB memory, and NVIDIA GTX 1070 GPU
having 8GB inbuilt RAM. The programming tool selected is
MATLAB2020a.

A. DATASET
The provided dataset (CUI-EXAM) is collected with collabora-
tion from COMSATS University Islamabad (CUI), Wah
Campus Pakistan. The dataset consists of a total of 2268 snaps
of student activities obtained while offline examinations are
setup through Surveillance cameras. The dataset is annotated
from 50 acquired video streams. The bounding boxes of indi-
vidual students are cropped from the frames of video streams
manually and separated category-wise according to 6 classes.
The action categories are: (a) seeing back, (b) watching
towards the front, (c) normal performing, (d) passing gestures
to other fellows, (e) watching towards left or right, and (f)
other suspicious actions (see Figure 5 for sample dataset
snaps). The dataset is challenging because many bounding
boxes acquired are from images that are far from the camera
and have a blurring effect. To increase the size of the dataset,
dataset augmentation is performed by following the proce-
dure defined in section 3.5. Table 2 portrays the size of the
dataset before and after the augmentation.

B. PERFORMANCE EVALUATION PROTOCOLS
Different performance evaluation protocols are followed in
this research to access the robustness of the proposed work.
Most of these protocols are depending on the confusion
matrix formed during the testing phase of the classification
task. These protocols are depicted mathematically as
(14)–(20), as shown at the bottom of the page.

5-folds cross-validation scheme is used for training and
testing purposes. Figure 6 represents some intermediate
visualizations of features taken at various convolution layers
of the proposed L2-GraftNet.

C. OVERVIEW OF THE EXPERIMENTS PERFORMED
Using the suggested model, thorough experimentation is con-
ducted for multiple variations of selected features to seek the
optimal outcomes. There are a few major analyses mentioned
here. A concise narration of each experiment mentioned is
briefed in Table 3. Many experiments are performed with
varying numbers of features at the feature selection step.

Only accuracies of seven experiments are mentioned
in Table 2 and details of only the first five experiments are
provided in this manuscript. The outcome of ASO fitness
values with the selected number of features is shown in the
charts given in Figure 7.

| Experiment # | Selected Features | Best accuracy achieved (%) |
|--------------|-------------------|---------------------------|
| 1            | 100               | 92.43                     |
| 2            | 250               | 93.17                     |
| 3            | 500               | 93.50                     |
| 4            | 750               | 93.53                     |
| 5            | 1000              | **93.88**                 |
| 6            | 1500              | 93.78                     |
| 7            | 2000              | 93.87                     |

TABLE 2. CUI-EXAM dataset detail.

| Class                  | Original | Augmented |
|------------------------|----------|-----------|
| Back watching          | 406      | 2030      |
| Front watching         | 285      | 1425      |
| Normal                 | 560      | 2800      |
| Showing gestures       | 326      | 1630      |
| Side watching          | 379      | 1895      |
| suspicious             | 312      | 1560      |
| Total                  | 2268     | 11340     |

Accuracy (Ac) = \[ \frac{\text{TruePositiveValues} + \text{TrueNegativeValues}}{\text{TruePositiveValues} + \text{TrueNegativeValues} + \text{FalsePositiveValues} + \text{FalseNegativeValues}} \]

Sensitivity (Si) = \[ \frac{\text{TruePositiveValues}}{\text{TruePositiveValues} + \text{FalseNegativeValues}} \]

Specificity (Sp) = \[ \frac{\text{TrueNegativeValues}}{\text{TrueNegativeValues} + \text{FalsePositiveValues}} \]

The area under the ROC curve (AUC) = \[ \frac{\text{TruePositiveValues} + \text{TrueNegativeValues}}{\text{TruePositiveValues} + \text{TrueNegativeValues} + \text{FalsePositiveValues} + \text{FalseNegativeValues}} \]

Precision (Pr) = \[ \frac{\text{TruePositiveValues}}{\text{TruePositiveValues} + \text{FalsePositiveValues}} \]

F – measure (FM) = \[ 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} \times \text{Recall}} \]

G – Mean (GM) = \[ \sqrt{\text{TPR} \times \text{TNR}} \]
FIGURE 5. Some sample images from the CUI Exam dataset representing six classes: (a) seeing back, (b) watching towards the front, (c) normal performing, (d) passing gestures to other fellows, (e) watching towards left or right, and (f) other suspicious actions.

TABLE 4. Performance results of experiment 1 (100 features).

| Classifier | Ac (%) | Si (%) | Sp (%) | Pr (%) | FM (%) | GM (%) |
|------------|--------|--------|--------|--------|--------|--------|
| LSVM       | 48.02  | 49.61  | 47.67  | 17.13  | 25.46  | 48.63  |
| QSVM       | 76.77  | 84.63  | 75.06  | 42.52  | 56.61  | 79.70  |
| FG SVM     | 64.89  | 68.67  | 64.07  | 29.42  | 41.19  | 66.33  |
| MG SVM     | 62.03  | 69.21  | 60.46  | 27.62  | 39.49  | 64.69  |
| CG SVM     | 42.29  | 45.57  | 41.58  | 14.53  | 22.04  | 43.53  |
| CSVM       | 85.85  | 91.48  | 84.62  | 56.46  | 69.83  | 87.98  |
| CKNN       | 79.20  | 89.90  | 76.86  | 45.87  | 60.74  | 83.13  |
| CrKNN      | 52.69  | 60.79  | 50.92  | 21.26  | 31.51  | 55.64  |
| FKNN       | 92.43  | 95.76  | 91.71  | 71.58  | 81.92  | 93.71  |

The best fitness achieved is by using 1000 features and the fitness acquired in the seventh iteration. The detailed discussion on experiments is depicted in the upcoming text.

1) EXPERIMENT # 1: (100 SELECTED FEATURES)
The first experiment is performed by selecting 100 features by the ASO algorithm. The size of the feature matrix, containing all dataset images, becomes $4536 \times 100$. 5-fold cross-validation is applied on this feature matrix with ground truth class labels on the CUI-EXAM dataset. It means that each fold contains 80% random data is chosen for training and the remaining 20% data is selected for testing. This feature matrix is provided for automatic labeling to the prediction models that are the variants of SVM and KNN. The best result of 92.43 percent accuracy is obtained in this test by FKNN. The second-largest performance of 85.85 percent in terms of accuracy was reached by CSVM. In Table 4, the findings of the research for this experiment are presented. Similarly, Figure 8 contains a time plot and speed of training in this experiment.

2) EXPERIMENT # 2: (250 SELECTED FEATURES)
The first experiment is accomplished by picking 250 features by the ASO algorithm. The size of the feature matrix becomes $4536 \times 250$. 5-fold cross-validation is applied on this feature matrix with ground truth class labels on the CUI-EXAM dataset. This feature matrix is given to the variants of SVM and KNN. The best result of 93.17 percent accuracy is obtained in this test by FKNN. The second-highest performance of 90.03 percent in terms of accuracy is reached by CSVM. In Table 5, the outcomes of the research for this experiment are introduced. Similarly, Figure 9 covers a time plot and speed of training in this experiment.

3) EXPERIMENT # 3: (500 SELECTED FEATURES)
In this experiment, 500 features are chosen. The size of the feature matrix turns out to be $4536 \times 500$. Again, 5-fold cross-validation is operated on the CUI-EXAM dataset. This feature matrix is supplied for classification to the SVM and KNN based classifiers. The best result...
of 93.50 percent accuracy is attained in this test by FKNN. And the second-best performance of 91.41 percent in terms of accuracy was grasped by CSVM. In Table 6, the findings of the research for this experiment are presented. Also, it is observed that the performance of MGSVM is increased to almost six percent in terms of accuracy. Figure 10 encompasses a time plot and prediction speed of training in this experiment. The training time is increased as compared to previous experiments because of the increase in the number of features.

4) EXPERIMENT # 4: (750 SELECTED FEATURES)

In this experiment, 750 features are chosen. The size of the feature matrix is now $4536 \times 750$. This feature matrix is provided for the categorization of the SVM and KNN classifiers. The finest result of 93.53 percent accuracy is attained in this test by FKNN. It is observed that there is a slight difference in accuracy results when the number of features is increased from 500 to 750 that is 0.03 percent. The second-best execution of 92.21 percent in terms of accuracy was found by CSVM. In Table 7, the results of the research
FIGURE 7. ASO Fitness charts at different number of selected features: (a) 100, (b) 250, (c) 500, (d) 750 and (e) 1000.

FIGURE 8. Training time (sec) and prediction speed (obs/sec) plot with 100 features.

for this experiment are presented. Also, it is observed that the performance of MGSVM is now decreased in terms of accuracy. Figure 11 encompasses a time plot and prediction speed of training in this experiment. The training time is
increased as compared to previous experiments because of the increase in the number of features.

5) EXPERIMENT # 5: (1000 SELECTED FEATURES)
In this experiment, 1000 features are chosen. The size of the feature matrix is now $4536 \times 1000$. This feature matrix is provided for the categorization of the SVM and KNN classifiers. The finest result of 93.88 percent accuracy is attained in this test by FKNN. It is observed that there is again a slight difference of accuracy results when the number of features is increased from 7500 to 1000 that is 0.35 percent. The second-best performance of 92.58 percent in terms of accuracy was found by CSVM. In Table 8, the findings of the research for this experiment are shown. The third-highest accomplishment of MG SVM is further declined in terms of accuracy. Figure 12 covers a time plot and prediction...
TABLE 7. Performance results of experiment 4 (750 features).

| Classifier | Ac (%) | Si (%) | Sp (%) | Pr (%) | FM (%) | GM (%) |
|------------|--------|--------|--------|--------|--------|--------|
| LSVM       | 66.53  | 74.09  | 64.88  | 31.50  | 44.21  | 69.33  |
| Q SVM      | 86.34  | 91.13  | 85.30  | 57.47  | 70.49  | 88.17  |
| FG SVM     | 34.56  | 29.36  | 35.69  | 09.05  | 13.84  | 32.37  |
| MG SVM     | 88.27  | 91.77  | 87.51  | 61.57  | 73.69  | 89.62  |
| CG SVM     | 64.94  | 71.18  | 63.58  | 29.88  | 42.09  | 67.27  |
| CSVM       | 92.21  | 95.22  | 91.56  | 71.09  | 81.41  | 93.37  |
| CKNN       | 82.28  | 91.38  | 80.30  | 50.28  | 64.87  | 85.66  |
| CrKNN      | 57.40  | 64.73  | 55.80  | 24.20  | 35.23  | 60.10  |
| FKNN       | 93.53  | 96.95  | 92.78  | 74.55  | 84.28  | 94.84  |

FIGURE 11. Training time (sec) and prediction speed (obs/sec) plot with 750 features.

speed of training showing the effective training times of CKNN, CrKNN, and FKNN. All the experiments are performed on the same classifiers (mentioned in classification subsection of material and methods section) except experiment #5 where five more classifiers results are added to further analyze the results. The additional classifiers include Medium KNN (MKNN), Weighted KNN (WKNN) [54], [55], Fine Tree (FTree), Medium Tree (MTree), and Coarse Tree (CTree). Tree based classifiers can further be studied in detail from [56], [57]. It is observed that FKNN is dominant over all the mentioned classifier (see Table 8).

The performance outcomes with 1000 features are selected as best as the performance reduces or sometimes increases with a very small difference. Table 9 presents the confusion matrix of best results. The true positive values of six classes are highlighted in a diagonal. Overall, all classes show a good recognition rate. The class “Watching towards left or right” shows less accuracy. This is also depicted by ROC and AUC in Figure 13. Still, the AUC is found more than 0.94.

D. DISCUSSION

The manuscript is focused on EAR. For this purpose, the CUI-EXAM dataset is obtained. To increase the performance of the proposed approach, the proposed 46 layers L2-GraftNet is developed with extensive experimentation. The model and is first pre-trained on the CIFAR dataset which contains 100 classes. The features of the CUI-EXAM dataset are then obtained on the trained network. ASO-based feature selection is adopted on extracted features. The classification is applied in different experiments with varying the number of optimal features to observe the performance of the proposed work. The results of experiments provided in Tables IV-VIII depict the performance of the proposed system in terms of accuracy (Ac), sensitivity (Si), Specificity (Sp), Precision (Pr), F1-measure, and G-mean. These results are obtained with varying the number of features at the feature selection step. All the experiments are performed on the same classifiers. Five additional classifiers result in experiment 5 subsection are added because the performance of experiment 5 is found better with 1000 features.

It is analyzed that as the number of features increases, the performance also increases but after a certain level the rate of increase of performance becomes very low. For instance, the accuracy difference between 500 features and 750 features is 0.03 percent. Similarly, the accuracy difference between 750 features and 1000 features is 0.35 percent. However, the accuracy drops on 1500 and 2000 features slightly (see Table 3). Therefore, the framework with 1000 features is found better in all aspects. The performance in experiment 5 is measured on variants of three classifiers including SVM,
TABLE 8. Performance results of experiment 5 (1000 features).

| Classifier | Ac (%) | Si (%) | Sp (%) | Pr (%) | FM (%) | GM (%) |
|------------|--------|--------|--------|--------|--------|--------|
| LSVM       | 67.65  | 75.12  | 66.02  | 32.52  | 45.39  | 70.42  |
| QSVVM      | 87.51  | 92.76  | 86.37  | 59.74  | 72.67  | 89.51  |
| FG SVM     | 34.56  | 28.37  | 35.91  | 68.80  | 13.44  | 31.92  |
| MGSVM      | 82.35  | 85.96  | 81.57  | 50.42  | 63.56  | 83.74  |
| CG SVM     | 68.79  | 76.70  | 67.07  | 33.68  | 46.81  | 71.72  |
| CSV SVM    | 92.58  | 95.32  | 91.99  | 72.17  | 82.15  | 93.64  |
| CKNN       | 82.05  | 91.72  | 79.95  | 49.93  | 64.66  | 85.63  |
| GCKNN      | 57.16  | 68.13  | 54.77  | 24.72  | 36.28  | 61.08  |
| FKNN       | 93.88  | 96.85  | 93.23  | 75.73  | 85.00  | 95.02  |
| MKNN       | 81.69  | 91.43  | 79.57  | 49.39  | 64.13  | 85.29  |
| WKNN       | 90.41  | 96.11  | 89.16  | 65.91  | 78.20  | 92.57  |
| FTree      | 35.00  | 34.88  | 35.03  | 10.48  | 16.11  | 34.95  |
| MTree      | 32.51  | 36.70  | 31.60  | 10.47  | 16.30  | 34.05  |
| CTree      | 28.22  | 35.42  | 26.66  | 69.53  | 15.01  | 30.72  |

FIGURE 12. Training time (sec) and prediction speed (obs/sec) plot with 1000 features and FKNN classifier.

TABLE 9. Confusion matrix of best outcome with 1000 features and FKNN classifier.

| True Classes          | Seeing back | Watching towards front | Normal performing | Passing gestures to other fellows | Watching towards left or right | Other suspicious actions |
|-----------------------|-------------|------------------------|--------------------|----------------------------------|-------------------------------|-------------------------|
| Seeing back           | 1966        | 2                      | 11                 | 4                               | 45                            | 2                       |
| Watching towards front| 12          | 1300                   | 60                 | 17                              | 31                            | 5                       |
| Normal performing     | 22          | 57                     | 2621               | 10                              | 47                            | 43                      |
| Passing gestures to other fellows | 5    | 26                      | 17                 | 1546                            | 30                            | 6                       |
| Watching towards left or right | 43    | 34                      | 80                 | 17                              | 1699                          | 22                      |

KNN, and tree. Overall, tree-based classifiers performed very poor with all having accuracy less than or equal to 35%. The accuracy of SVM based classifiers is found in between 34.56% to 92.58%. Performance on nearly all KNN based
V. CONCLUSION

This work is based on a proposed CNN network i.e., L2-GraftNet for feature extraction along with the ASO algorithm for feature selection. The proposed L2-GraftNet is first converted to a pre-trained model by performing training on the CIFAR-100 dataset. The features are taken from the pre-trained model on the CUI-EXAM dataset. The CUI-EXAM dataset is first annotated and augmented before taking its image features. These features are forwarded to ASO based features optimization method and through various classifier variants of SVM and KNN, the model performance is observed on the selected features. 5-Fold cross-validation is applied for testing and training of the dataset. Many experiments are performed, apart, only five experiments are discussed in detail. It is observed that the least performance outcomes are obtained for experiments with 100 features having an accuracy of 92.43 percent by using the FKNN classifier. Similarly, the best performance outcomes are considered for experiments with 1000 features having an accuracy of 93.88 percent by using the FKNN classifier. From all classifiers’ outcomes, in all experiments, it is observed that FKNN shows high performance and CSVM depicts second to high performance in terms of selected performance measures.

Although the proposed system shows satisfactory outcomes in terms of performance measures, still there can be improvements performed for further enhancement of accuracy. In the future, the work can further be explored with more state-of-the-insight approaches such as manifold learning, LSTMs, Quantum deep learning, and brain-like computing approaches.

ACKNOWLEDGMENT

The authors would like to thank the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication.

REFERENCES

[1] A. R. Baig and H. Jabeen, “Big data analytics for behavior monitoring of students,” Procedia Comput. Sci., vol. 82, pp. 43–48, Jan. 2016.
[2] F. Rodrigues and P. Oliveira, “A system for formative assessment and monitoring of students’ progress,” Comput. Educ., vol. 76, pp. 30–41, Jul. 2014.
[3] J. Ramberg and B. Modin, “School effectiveness and student cheating: Do students’ grades and moral standards matter for this relationship?” Social Psychol. Educ., vol. 22, no. 3, pp. 517–538, Jul. 2019.
[4] Z. A. von Jena, “The cognitive conditions associated with academic dishonesty in university students and its effect on society,” UC Merced Undergraduate Res. J., vol. 12, no. 1, pp. 1–21, 2020.

[5] M. Ghizlane, B. Ilcham, and F. H. Reda, “A new model of automatic and continuous online exam monitoring,” in Proc. Int. Conf. Syst. Collabora. Big Data, Internet Things Secur. (SysCoBiTS), Dec. 2019, pp. 1–5.

[6] I. Blau and Y. Eshet-Alkalai, “The ethical disinnovation in digital and non-digital learning environments: Does technology promotes cheating among middle school students?” Comput. Hum. Behav., vol. 73, pp. 629–637, Aug. 2017.

[7] A. Asfirfan, A. Ghofur, and N. Azizah, “Cheating behavior in EFL classroom: A case study at elementary school in Sidenreng Rappang Regency,” OKARA: J.Bahasa dan Sastra, vol. 14, no. 2, pp. 279–297, 2020.

[8] P. M. Newton, “How common is commercial contract cheating in higher education and is it increasing? A systematic review,” Frontiers Educ., vol. 3, p. 67, Aug. 2018.

[9] W. J. Bowers, Student Dishonesty and Its Control in College. Bureau of Applied Social Research, Columbia University, 1964.

[10] A. Bushway and W. R. Nash, “School cheating behavior,” Rev. Educ. Res., vol. 47, no. 4, pp. 623–632, Dec. 1977.

[11] D. F. Crown and M. S. Spiller, “Learning from the literature on collegiate cheating: A review of empirical research,” J. Bus. Ethics, vol. 17, pp. 683–700, Apr. 1998.

[12] C. M. Ghaneem and N. A. Mozahem, “A study of cheating beliefs, engagement, and perception—The case of business and engineering students,” J. Academic Ethics, vol. 17, no. 3, pp. 291–312, Sep. 2019.

[13] S. W. Tabsh, H. A. El Kadi, and A. S. Abdellatif, “Faculty perception of engineering student cheating and effective measures to curb it,” in Proc. IEEE Global Eng. Educ. Conf. (EDECON), Apr. 2019, pp. 806–810.

[14] M. Korman, “Behavioral detection of cheating in online examination,” M.S. thesis, Lulea Univ. Technol., 2010.

[15] C. Voss and N. J. Haber, “Systems and methods for detection of behavior correlated with outside distractions in examinations,” Google Patent WO2018131152 B2, Feb. 13, 2018.

[16] K. Jalali and F. Noorbahbani, “An automatic method for cheating detection in online exams by processing the student’s webcam images,” in Proc. 3rd Conf. Elect. Comput. Educ. Technol. (E-Tech), Tehran, Iran, 2017, pp. 1–6.

[17] R. Metzger and R. Maudoodi, “Using access reports and API logs as additional tools to identify exam cheating,” in Proc. Soc. Inf. Technol. Teacher Educ. Int. Conf., 2020, pp. 294–299.

[18] N. van Halem, C. van Klaveren, and I. Cornelisz, “The effects of implementation barriers in virtually proctored examination: A randomised field experiment in Dutch higher education,” Higher Educ., Quart., vol. 2275, pp. 1–15, 2020.

[19] K. Butler-Henderson and J. Crawford, “A systematic review of online examinations: A pedagogical innovation for scalable authentication and integrity,” Comput. Educ., vol. 159, Dec. 2020, Art. no. 104024.

[20] S. Dendir and R. S. Maxwell, “Cheating in online courses: Evidence from online proctoring,” Comput. Hum. Behav. Rep., vol. 2, Aug. 2020, Art. no. 100503.

[21] X. Li, K.-M. Chang, Y. Yuan, and A. Hauptmann, “Massive open online proctor: Protecting the credibility of MOOCs certificates,” in Proc. 18th ACM Conf. Comput. Supported Cooperat. Work Social Comput., Feb. 2015, pp. 1129–1137.

[22] G. A. Adanur, R. Ismailova, A. Oumuraliev, and G. Muhamedjanova, “Learners’ perceptions of online exams: A comparative study in Turkey and Kyrgyzstan,” Int. Rev. Res. Open Distr. Learn. Educ., vol. 21, no. 3, pp. 1–17, 2020.

[23] A. Ullah, H. Xiao, and M. Lilley, “Profile based student authentication in online examination,” in Proc. Int. Conf. Inf. Soc. (i-Society), Dec. 2012, pp. 109–113.

[24] B. S. Bagepally, “Gaze pattern on spontaneous human face perception: An eye tracker study,” J. Indian Acad. Appl. Psychol., vol. 41, no. 3, pp. 128–131, 2015.

[25] A. Babenko and A. Kotenko, “The eye tribe,” Doctoral dissertation, Sumy State Univ., 2016.

[26] J. Raynowska, J.-R. Rizzo, J. C. Rucker, W. Dai, J. Birkeemeier, J. Hershitz, I. Selesnick, L. J. Balcer, S. L. Galetta, and T. Hudson, “Validity of low-resolution eye-tracking to assess eye movements during a rapid number naming task: Performance of the eyetrive eye tracker,” Brain Injury, vol. 32, no. 2, pp. 200–208, Jan. 2018.

[27] M. Buctuano, R. Debalucos, and J. G. Villalon, “Remotely view user activities and impose rules and penalties in a local area network environment,” Int. J. Comput. Sci. Inf. Technol., vol. 10, no. 6, pp. 69–79, Dec. 2018.
S. Rüping, “SVM kernels for time series analysis,” Tech. Rep. 2001, 43, 2001.

N. E. Ayat, M. Cheriet, and C. Y. Suen, “Automatic model selection for the optimization of SVM Kernels,” Pattern Recognit., vol. 38, no. 10, pp. 1733–1745, Oct. 2005.

B. Haasdijk, “Feature space interpretation of SVMs with indefinite Kernels,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 4, pp. 482–492, Apr. 2005.

Vibha and A. P. Singh, “Analysis of variants of KNN algorithm based on preprocessing techniques,” in Proc. Int. Conf. Adv. Comput., Commun. Control Netw. (ICACCCN), 2018, pp. 186–191.

A. Lamba and D. Kumar, “Survey on KNN and its variants,” Int. J. Adv. Res. Comput. Commun. Eng., vol. 5, no. 5, pp. 430–435, 2016.

L. Jiang, H. Zhang, and J. Su, “Learning k-nearest neighbor Naive Bayes for ranking,” in Proc. Int. Conf. Adv. Data Mining Appl., 2005, pp. 175–185.

Y. Xu, Q. Zhu, Z. Fan, M. Qiu, Y. Chen, and H. Liu, “Coarse to fine k nearest neighbor classifier,” Pattern Recognit. Lett., vol. 34, no. 9, pp. 980–986, Jul. 2013.

Y.-L. Chen, T. Wang, B.-S. Wang, and Z.-J. Li, “A survey of fuzzy decision tree classifier,” Fuzzy Inf. Eng., vol. 1, no. 2, pp. 149–159, Jun. 2009.

Priyanka and D. Kumar, “Decision tree classifier: A detailed survey,” Int. J. Inf. Decis. Sci., vol. 12, pp. 246–269, 2020.

A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 25, 2012, pp. 1097–1105.

Y.-N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size,” 2016, arXiv:1602.07360. [Online]. Available: http://arxiv.org/abs/1602.07360

J. Bouvrie. (2006). Notes on Convolutional Neural Networks. Accessed: Mar. 10, 2021. [Online]. Available: http://cogprints.org/5869/

Y. Li, Z. Hao, and H. Lei, “Survey of convolutional neural network,” Comput. Appl., vol. 36, no. 9, pp. 2508–2515, 2016.

J. Wu, “Introduction to convolutional neural networks,” Nat. Key Lab Novel Softw. Technol., Nanjing Univ., Nanjing, China, Tech. Rep., 2017, p. 23, vol. 5. [Online]. Available: https://cs.nju.edu.cn/wujx/paper/CNN.pdf

S. Balocco, M. González, R. Nanculef, P. Radeva, and G. Thomas, “Calibrated plaque detection in IVUS sequences: Preliminary results using convolutional nets,” in Proc. Int. Workshop Artif. Intell. Pattern Recognit., 2018, pp. 34–42.

Y. Liu, X. Wang, L. Wang, and D. Liu, “A modified leaky ReLU scheme (MLRS) for topology optimization with multiple materials,” Appl. Math. Comput., vol. 352, pp. 188–204, Jul. 2019.

S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. Int. Conf. Mach. Learn., 2015, pp. 448–456.

A. Krizhevsky and G. Hinton. (Sep. 2009). Learning Multiple Layers of Features From Tiny Images. Accessed: Mar. 10, 2021. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.9220&rep=rep1&type=pdf

M. H. Pham, T. H. Do, V.-M. Pham, and Q.-T. Bui, “Mangrove forest classification and aboveground biomass estimation using an atom search algorithm and adaptive neuro-fuzzy inference system,” Proc. ONE, vol. 15, no. 5, May 2020, Art. no. e0233110.

W. Zhao, L. Wang, and Z. Zhang, “Atom search optimization and its application to solve a hydrogeologic parameter estimation problem,” Knowl.-Based Syst., vol. 163, pp. 283–304, Jan. 2019.

J. Too and A. R. Abdullah, “Chaotic atom search optimization for feature selection,” Arabian J. Sci. Eng., vol. 45, no. 8, pp. 6063–6079, Aug. 2020.

W. S. Noble, “What is a support vector machine?” Nature Biotechnol., vol. 24, no. 12, pp. 1565–1567, Dec. 2006.

L. E. Peterson, “K-nearest neighbor,” Scholarpedia, vol. 4, no. 2, p. 1883, 2009.

Y.-W. Chang and C.-J. Lin, “Feature ranking using linear SVM,” in Proc. Causation Predict. Challenge, 2008, pp. 53–64.

I. Dagher, “Quadratic Kernel-free non-linear support vector machine,” J. Global Optim., vol. 41, no. 1, pp. 15–30, May 2008.

P. Virdi, Y. Narayanan, P. Kumari, and L. Mathew, “Discrete wavelet packet based elbow movement classification using fine Gaussian SVM,” in Proc. IEEE 1st Int. Conf. Power Electron., Intell. Control Energy Syst. (ICPEICES), Jul. 2016, pp. 1–5.

Z. Liu, M. J. Zuo, X. Zhao, and H. Xu, “An analytical approach to fast parameter selection of Gaussian RBF Kernel for support vector machine,” J. Inf. Sci. Eng., vol. 31, no. 2, pp. 691–710, 2015.

TANZILA SABA (Senior Member, IEEE) received the Ph.D. degree in document information security and management from the Faculty of Computing, Universiti Teknologi Malaysia (UTM), Malaysia, in 2012. She is currently serving as a Research Professor for the College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia. She has above 200 publications that have attracted 5000 plus citations. Her research interests include the recent years is medical imaging, MRI analysis, and soft-computing. She won Best Student Award in the Faculty of Computing, UTM, for 2012. Due to her excellent research achievement, she is included in Marquis Who’s Who (S & T) 2012.” She is also the editor of reviewer for reputed journals and on panel of TPC of international conferences. She has full command on variety of subjects and taught several courses at graduate and postgraduate level. She is also the Chair of the information system department for females. On accreditation side, she is a skilled lady with ABET & NCAAA quality assurance.

AMJAD REHMAN (Senior Member, IEEE) received the Ph.D. degree (Hons.) from the Faculty of Computing, Universiti Teknologi Malaysia, in 2010, with a specialization in forensic documents analysis and security. He is currently a Senior Researcher with the Artificial Intelligence & Data Analytics Lab, Prince Sultan University, Riyadh, Saudi Arabia. He received Rector Award for 2010 Best Student in the University. He is also the PI in several funded projects and also completed projects funded from MOHE Malaysia, Saudi Arabia. He is the author of more than 200 ISI journal articles, conferences, and H-index 42. His research interests include data mining, health informatics, and pattern recognition.

NOR SHAHIDA MOHD JAMAIL received the Ph.D. degree in software engineering from Universiti Putra Malaysia. She is currently with Prince Sultan University, Riyadh, Saudi Arabia, as an Assistant Professor. Her specialized are in software engineering, software process modelling, software testing, and cloud computing services. She had involved in the Machine Learning Research Group, Prince Sultan University. Her current research interests include cloud computing, quality in software engineering and software modelling, machine learning, and more on commercialization of research products. She is very committed in her academic career. Besides, she is also the Co-Chair, a Track Chair, and a Committee in several International Conference published by IEEE and Scopus indexed.
SOUAD LARABI-MARIE-SAINTE received the Engineering degree in operation research from the University of Sciences and Technology Houari Boumediene, Algeria, and the dual M.Sc. degree in mathematics, decision, and organization from Paris Dauphine University, France, and in computing and applications from Sorbonne Paris1 University, France, and the Ph.D. degree in artificial intelligence from the Computer Science Department, Toulouse 1 University Capitole, France, in 2011. She was an Associate Researcher with the Department of Computer Science, Toulouse 1 University Capitole. Then, she was an Assistant Professor with the College of Computer and Information Sciences, King Saud University. She is currently an Associate Professor with the Department of Computer Science, an Associate Director of Postgraduate Programs at the College of Computer and Information Sciences, and the Vice-Chair of the ACM Professional Chapter with Prince Sultan University. She taught several courses at the graduate and postgraduate levels. She published several articles in ISI/Scopus indexed and attended various specialized international conferences. She is currently an editorial board member and a reviewer of reputed journals and on the panel of TPC of international conferences. Her research interests include metaheuristics, artificial intelligence, healthcare, machine and deep learning, natural language processing, educational data mining, pattern recognition, and software engineering.

MUDASSAR RAZA (Senior Member, IEEE) received the Ph.D. degree from the University of Science and Technology of China (USTC), Anhui, Hefei, China. He is currently an Assistant Professor with COMSATS University Islamabad (CUI)-Wah, Pakistan, with an overall job experience of around 17 years at both graduate and undergraduate levels. He remained a very active research member of fields related to visual robotics, artificial intelligence, computer vision, and pattern recognition. He has more than 100 research publications in journals and conferences of international repute with cumulative impact factor more than 85 and more than 1860 citations. He was honored with “National Youth Award 2008” in the category of “Computer Science and Information Technology” by Prime Minister of Pakistan on International Youth Day, in August 2008. He has so far supervised and co-supervised 15 M.S. theses. He is also supervising four Ph.D. students. He has also supervised more than 50 research and development projects of undergraduate students. He is continuously being nominated for COMSATS Research Productivity Award for the last many years. He is also the academic editor and a reviewer of several prestigious journals, conferences, and books. He is also being invited on regular basis from top universities and institutes of Pakistan as a guest speaker on symposiums and workshops.

MUHAMMAD SHARIF (Senior Member, IEEE) is currently an Associate Professor with COMSATS University Islamabad–Wah, Pakistan. He has worked one year in Alpha Soft U.K. based software house in 1995. He is also the OCP of Developer Track. He has been in teaching profession since 1996 to date. He also headed the department from 2008 to 2011 and achieved the targeted outputs. He has more than 210 research publications in IF, SCI and ISI journals as well as in national and international conferences and obtained more than 245 impact factor. He has supervised three Ph.D. (CS) and more than 50 M.S. (CS) theses to date. His research interests include medical imaging, biometrics, computer vision, machine learning, and agriculture plants imaging. He awarded with the COMSATS Research Productivity Award, from 2011 till 2018. He has served in TPC for IEEE FIT 2014–2019 and also serving as an Associate Editor for IEEE Access, a Guest Editor of special issues, and a reviewer for well reputed journals.