IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 13, 2022, accepted February 23, 2022, date of publication February 28, 2022, date of current version March 8, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3155652

Scene Graph Generation With Structured Aspect of Segmenting the Big Distributed Clusters

AMJAD REHMAN KHAN[1], (Senior Member, IEEE), HAMZA MUKHTAR[2], TANZILA SABA[1], (Senior Member, IEEE), OMER RIAZ[3], MUHAMMAD USMAN GHANI KHAN[2], AND SAEED ALI BAHAJ[4]

[1]Artificial Intelligence & Data Analytics Laboratory, CCIS, Prince Sultan University, Riyadh 11586, Saudi Arabia
[2]National Center of Artificial Intelligence, KICS, UET Lahore, Lahore 54000, Pakistan
[3]Department of Information Technology, Faculty of Computing, The Islamic University of Bahawalpur, Bahawalpur 63100, Pakistan
[4]MIS Department, College of Business Administration, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia

Corresponding author: Omer Riaz (omer.riaz@iub.edu.pk)

This work was supported by the Artificial Intelligence & Data Analytics Laboratory (AIDA), CCIS, Prince Sultan University, Riyadh, Saudi Arabia.

ABSTRACT Accurate fruit counting is one of the significant phenotypic traits for crucial fruit harvesting decision making. Existing approaches perform counting through detection or regression-based approaches. Detection of fruit instances is very challenging because of the very small fruit size compared to the whole size image of a tree. At the same time, regression-based counting techniques contributes impressive results but presents inaccurate results while number of instances increases. Moreover, most approaches lack scalability and are applicable only on one or two fruit types. This paper proposes a fruit counting mechanism that combines loose segmentation and regression counting that works on six fruit types: Apple, Orange, Tomato, Peach, Pomegranate and Almond. Through relaxed segmentation, fruit clusters are segmented to extract the small image regions which contain the small cluster of fruits. Extracted regions are forwarded for the regression counting of fruits. Relaxed segmentation is achieved through a state-of-the-art deconvolutional network, while modified Inception Residual Networks (ResNet) based nonlinear regression module is proposed for fruit counting. For segmentation, 4,820 original images, including corresponding mask images, of all six fruit types are augmented to 32,412 images through different augmentation techniques, while 21,450 extracted patches are augmented to 89,120 images used for the regression module training. The proposed approach attained a counting accuracy of 94.71% for individual fruit types higher than techniques reported in literature.

INDEX TERMS Deep learning, segmentation, fruit counting, agricultural yield estimation, economic growth, agriculture, technological development.

I. INTRODUCTION

Yield estimation is becoming increasingly important in digital agriculture, which assists farmers to streamline harvesting resources which boost the cost-cutting for harvesting, enabling them to market the yield in a better way to get higher profits. With prior estimation of yield, farmers can make substantiate decisions to arrange the labor and machinery for ripping the crop, early order of required packing stuff, manage logistics to transfer and prepare sizable storage and processing facilities [1]. With prior decisions, farmers can devise better marketing and sales strategy to get a higher price. On the other hand, manual fruit estimation in orchards is quite labor-intensive, giving inaccurate numbers and infeasible at a large scale [2].

Agri vision using image processing through Computer vision is a growing field that can also assist in yield estimation. However, traditional image processing techniques are inefficient due to varying lighting conditions, color complexion, lack of robustness, occlusion and process hand-engineered features against each specific scenario [3], [4]. With Deep learning techniques, such as Convolutional Neural Networks (CNN), limits of image processing have been extended, which solves the complex Computer Vision problem, such as classification, detection and segmentation [5]. In addition, deep learning

The associate editor coordinating the review of this manuscript and approving it for publication was Jeon Gwanggil.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

techniques are efficient enough to generalize across various fruit types and environments that are dynamic in lighting conditions.

Significant progress has been made to devise different approaches to formulate an efficient and accurate system for orchard fruit counting. Several object detection methods have been developed based on localization and classification of fruits [6], [7]. Still, lack of accuracy when instance size is small and image capturing from closer are not practically feasible. Counting through object estimation based on density maps is also effective [8]. Explicit object counters, which use unmanned aerial vehicles (UAVs) as input sources, are also developed [9], [58].

Regression models give state-of-the-art results in object counting because of direct optimization of respective loss function for count prediction [8], [36]. In contrast, optimization of detection-based models is required to perform complex tasks, such as shape, size, and spatial location of the object instances. Perfect detection brings perfect count; however, bounding box and pixel-level annotated training data is expensive to acquire as compared to point level annotations which give approximate localization of object, and exact shape is not required for counting [37], [38].

With the emergence of CNNs, segmentation has become crucial for image analysis tasks in various fields including precision agriculture [10]. Semantic segmentation is a clustering phenomenon to group the objects of the same category together [11]. Fruits are entangled in clusters; therefore, individual fruit instances are not always obvious, amplifying the indulgence of the segmentation-based counting method for tiny fruit instances. We have developed a convolutional-deconvolutional segmentation network for the binary segmentation of the image to separate the fruit instances from other regions. As deep learning models are data-hungry, data augmentation techniques are applied equally to respective point-level masked images. Segmented image extracts tiny patches containing fruit clusters from the original RGB image through connected component analysis. The extracted patches go to the counting module to get the count on individual patches, which is summed up to get a total count for a single image. Our experiment illustrates that the proposed loose segmentation-based counting model obtains better and more efficient counting output than detection-based regression methods.

Further, section II explores the related work; section III presents the proposed approach in depth. Section IV, exhibits experimental results and analysis. Finally, section V, concludes the research.

II. RELATED WORK

Object detection has been in the spotlight over the past few years, and fabulous work has been reported [8], [14]. This problem is tried to be solved through unsupervised way through clustering of objects based on motion similarities [12] or structural similarity [13], but such unsupervised approaches have accuracy limitations, and to achieve higher accuracy supervised approaches are considered. Broadly, counting solutions are of three categories [15]: (1) clustering-based counting, (2) regressing based counting, and (3) object detection-based counting.

Clustering, unsupervised learning, based approaches are the initial work on counting problems. Objects are clustered based on similar features, such as texture, appearance, color and motion [16], and objective is to maximize the likelihood, which groups the individual object instances on low-level features. For example, a motion analysis-based mechanism is proposed for the moving objects where a parallel KLT tracker is used to observe the motion and appearance of feature points, and clustered are made based on observed features [17]. However, unsupervised approaches use lower level features and perform inaccurately on counting when we see in contrast with state-of-the-art Deep learning approaches.

Regression-based counting approaches are very accurate and efficient because the counting mechanism is learnt explicitly rather than optimizing object localization. They learn the direct mapping from image features to count labels, and for this learning, a huge amount of annotated data is required. [8] proposed a method, called glance, which explicitly learns the counting by mapping labelled counts on the image. Regression-based approaches are inefficient and give low accuracy when object instances are in large numbers [14].

Counting through object detection, draw the bounding boxes on detected objects, and just count the bounding boxes. Ground-truth labels are given in bounding boxes around objects for training [18]–[21]. Perfect detection leads to perfect counting. However, Chattopadhyay et al [14] manifested that detection method can perform poorly because the model needs to learn the object shape, size and localize it regardless of occluded real work conditions. Therefore, methods of detection based on pixel-level ground truth are also proposed [2], [22].

Song *et al.* [23] suggested a counting method with two models: first, bag-of-words model to discover the fruit instances in an image; second, aggregate model to sum up the count using a statistical approach on a bunch of given images. Maldonado et al [24] presented a method for green orange fruits counting based on correlation between visible fruits and whole fruits on trees. Feature extraction is performed by combining the techniques such as Gaussian blur thresholding, histogram, color conversion, spatial filtering and Sobel operator. Input image is converted into a bas-relief representation on which filtering is applied and forward to SVM which decides whether the object is fruit and counts the positive decisions. However, large adjustable parameters and manual feature extraction are prolonged and not robust in occluded conditions. Linker [33] suggested an estimation procedure based on light distribution. Dorj *et al.* [34] used color features to recognize fruit instances, conversion of RGB image to HSV, and different preprocessing techniques used for counting.

Rahnemoonfar *et al.* [7] proposed an inception-ResNet based estimation approach that maps the labelled count on images and reduces detection and localization cost. Training is performed on synthetic data and tested on read tomato images. Chen et al [9] suggested a deep learning approach that directly maps total count to input images. Candidate regions are extracted through a convolutional network-based blob detector, another convolutional network is employed to estimate the count in each extracted region, and a regression model map estimated count to a final count. Qureshi et al [26] proposed two methods: first, texture base segmentation based on K-Nearest neighbor classification and segmentation, and second, segmentation-based method which uses a support vector machine for classification. Bargoti *et al.* [27] presented a segmentation-based approach that consists of a multilayer perceptron and convolutional neural network. Segmentation is generated using watershed segmentation and individual fruits are counted through conducted Hough transform.

Liu et al [28] presented a segmentation and 3D localization model for counting. A fully convolutional network is used for segmentation and localization using an incremental structure motion algorithm. Ponce at al [29] proposed the counting method based on mathematical morphology which segment the olives to extract feature representation. Häni *et al.* [30] proposed a semantic segmentation model based on U-Net architecture and CNN for classification. Bellocchio et al [31] presented a weakly-supervised framework for explicit counting without supervised labels, only label whether instances belonging to the fruit class is required. Proposed an objective function to keep track of the predictions at different spatial locations of image. Roy et al [35] presented a counting approach where a semi-supervised clustering based on coloring is performed for fruit identification and spatial characteristics based on unsupervised clustering. Xiong et al [63] used YOLOv2 for fruit detection and linear regression for fruit counting.

Tu et al [32] presented a counting framework based on detection through multiple-scale faster-RCNN which detects the lower features effectively by incorporating feature maps for regions of interest. First, high and Lower-level features are extracted through a multiple scale detector, then RGB and depth detectors are trained which are finally combined through late fusion methods.

The main contributions of this research are

a. Estimated the orchard's yield by counting fruit instances. The primary focus is to build a highly accurate deep learning mechanism and develop a generalized approach so that model can be trained without prior knowledge about the type of the fruit.

b. Counted the fruit instances without segmenting the individual instances, we formulate the estimation problem as a non-linear regress problem which is helpful for many reasons. First, regression on small patches is more efficient than segmenting out the individual fruit instances. Second, from the supervised learning point of view, annotating individual instances is more challenging than annotating the segmented regions containing a small number of fruit instances. Finally, generalization of the model is very important to learn directly from annotated data without explicit information about the fruit type.

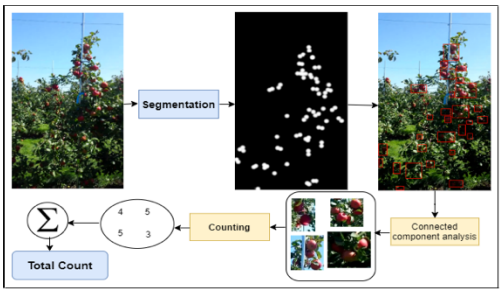

FIGURE 1. Workflow for fruit yield estimation: (1) loosely segment the fruit cluster from RGB images through segmentation model, (2) extract small cluster regions containing fruits via connected component analysis, (3) count is made for each patch, and (4) aggregate all the individual patch counts for the overall fruit count against single image.

c. Finally, deep learning inspired models are implemented to generalize the solution across different datasets, light conditions and variable sizes.

d. Driven by the inabilities of proposed techniques, we have devised a completely data-driven counting method based on loose semantic segmentation and direct regression form images.

III. PROPOSED APPROACH

This section illustrates the proposed loose binary semantic segmentation-based yield estimation approach where binary segmentation extracts small patches from an image containing a fruit cluster. The high-level design of our approach pursues a traditional computer vision workflow where the counting module follows the segmentation module. A two-step computational process for yield estimation is exhibited in Fig. 1. The proposed segmentation module generates the loosely segmented fruit cluster regions from RGB images on the first step. Then, responding fruit cluster regions are extracted from the input RGB image based on segmented regions. Finally, each extracted region is forwarded as input to the counting module to obtain the individual fruit count. At the end, individual counts are summed up to get the overall prediction count against a given input image, both segmentation and counting modules are built on deep learning architectures.

For each module, task-oriented convolutional architectures are introduced, trained without prior knowledge about the fruit type to build a generalized yield estimation approach that can be trained only from the data. Although both modules are trained separately, they are not independent entirely since binary masks produced by the segmentation module will be used to extract the sub patches containing fruit instances from original images. These extracted sub-images are used for the training of counting modules. In two subsections below, both modules are described along with rationale behind design.

A. SEGMENTATION

Regression counting on the whole image at once is computationally expensive when hundreds of trees are in an orchard. It requires many labelled samples that are pretty tedious to get and become extremely time-consuming when there are hundreds of fruit instances in a single image. As earlier established in [8], [14], regression-based counting achieves great results when the number of instances in images are small; however, accuracy gets compromised as the number of instances per image increases. Moreover, fruits grow in clusters, and processing the whole image is costly. So instead of processing the image as whole, counting over the non-overlapping patches containing clusters of fruit, is required. Therefore, disjoint patches of segmented fruit clusters are generated to provide thousands of small patches for the training of the counting module.

From the design point of view, output of the segmentation module is kept loose because instead of segmenting the individual fruit instances we want to segment the clusters so that corresponding patches can be extracted. Moreover, due to many fruit instances in the image, annotating the exact ground truth is highly tedious. It becomes even more expensive as the deep learning paradigm requires thousands of such annotated images. Since the background is almost uniform, learning for regression with loose and exact segmentation also becomes similar, and eventually, chances of involving the distinctive features from background are very low. Zhou et al [39] testifies the claim by visualizing the network that reveals saliency in the foreground. Fruit instances are very small compared to the whole image and partially occluded; therefore, soft segmentation of fruit clusters is suitable.

Due to the fewer training parameters, we have used the SegNet architecture [40], [41] for generating the loose segmentation of fruit clusters instead of deconvolutional networks with fully connected (FC) layers [61] having many more training parameters. As for cardinality of categories and the domain's nature, dealing with the loose segmentation is less complex than multi-class semantic segmentation because variations in pixel intensities are restricted in a single image. Additionally, main purpose is not to obtain an overall highly accurate segmentation mask, rather aim is not to miss any fruit cluster region in the image for the training of the counting module eventually. Fig. 2 illustrates the used segmentation network. The front-side convolutional substructure of the segmentation network is based on VGG architecture [25] where five 2×2 max-pooling operations are followed by convolutional and nonlinearity layers, which helps compress the feature map to 32 times before backend deconvolutional operation.

Class imbalance is very high in the fruit counting domain since the fruit cluster to background ratio incurs a big difference. Therefore, weighted categorical class-entropy is involved as a loss function that allows adjusting the weights depending on the misclassification to address the class imbalance problem.

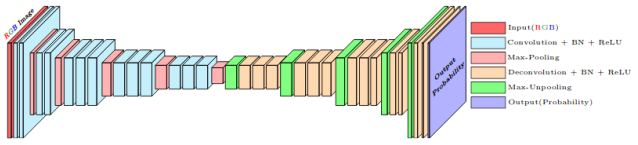

FIGURE 2. SegNet architecture [41] for fruit instances segmentation. Rectified linear unit [62] and batch normalization [44] is used after every convolutional and deconvolutional layer. In addition, 2×2 Max-pooling, with stride 2, is used, while 2×2 max-unpooling for un-pooling using corresponding pooling indices from the front-end network.

B. COUNTING

From Fig. 1, it can be seen that RGB images used for counting modules are extracted after segmenting out the fruit clusters by the segmentation module. Extraction of multiple patches from the original big image satisfies the need of a huge training dataset so that the model can learn input-output mapping. We have used the deep learning inspired counting approach to get the generalizable and robust counting solution. The combination of convolutional and pooling layers, CNN, is the deep learning approach that replicates the operational mechanism of the human vision system [42]. Input to the CNN is an image that goes through different convolutional and pooling layers and produces the representative feature map. Journey of the feature map between input to output layer goes from many hidden layers which consist of a stack of convolutional and pooling layers. Training of CNN goes through two stages: (1) feedforward and (2) backward propagation. Loss is calculated during the feedforward stage based on the predicted output from the produced feature maps and labelled outputs. In backpropagation, gradient of loss is calculated with respect to each weight parameter, and parameters are updated for next feedforward calculations based on gradient. Two staged processes go through many iterations and terminate when loss stops to decrease further.

Typically, CNN learns a feature map with two spatial and one channel dimensions simultaneously, increasing parameters. On the other hand, inception models ease this process and learn feature representation with fewer parameters because they work on spatial and cross-channel correlations. Although different inception models had been introduced with slight variation [43], [45], but, Inception-ResNet [46] outperformed the ImageNet dataset [47]. We used the modified Inception-ResNet-A with the proposed CNN network, which this performance influences. Usually, fruits are extremely crowded and vary in size due to natural variation in size and image capturing position incurs the size variability; therefore, high-level semantic feature plays a crucial role compared to receptive fields. Reason to this, indulgence of modified Inception-ResNet-A enlarges the receptive field [48]. The proposed network architecture is shown in Fig. 3.

First layer of the network is 5×5 convolutional followed by 2×2 with stride 2 max-pooling produces 64 feature maps. To reduce the dimensions of the first layer feature map, 1×1 convolutional is applied. Next, two 3×3 convolutional

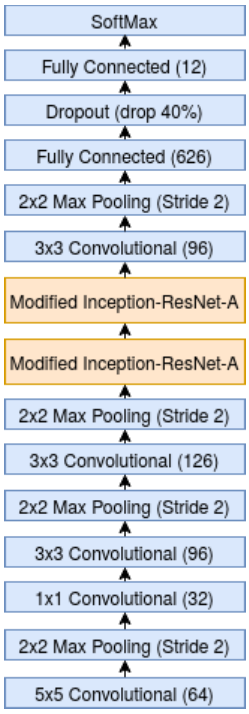

FIGURE 3. The architecture of proposed counting network.

followed by 2 × 2, stride 2, max-pooling layers are applied, producing 96 and 126 feature maps. Following, the modified inception layers come which take feature maps of multiple size through concatenating residual units [49], and the result of different filter sizes. Convergence of residual network is faster due to residual connections which skips connection to make a path for gradient flow.

Fig. 4 illustrates the architecture of the modified inception-ResNet-A model. Last layer, having 1 × 1 convolutional, calculates 126 feature maps instead of 256 as in original Inception-ResNet [46]. The Inception layer consists of three concatenated layers, and the result is added to activation of the previous layer which passes from a rectified linear unit. After the inception layer, 3×3 convolutional is again applied, followed by 2 × 2, stride 2, max-pooling, which increases the accuracy when used before a fully connected layer [50]. Size of the fully connected layer is 626. Deep learning models are prone to overfitting, which can be mitigated through dropout technique [51] where we have randomly dropped the 40% connections. Instead of regression output, we have applied SoftMax with 11 outputs because the number of fruit instances in extracted regions are less than 12 which makes SoftMax suitable.

IV. EXPERIMENTS & RESULTS

This section demonstrates the effectiveness of proposed methodology. First, we explain the used datasets for both modules. Next, training methodology for both modules along with training setup and implementation details. Lastly, evaluation and comparison with other proposed approaches are given.

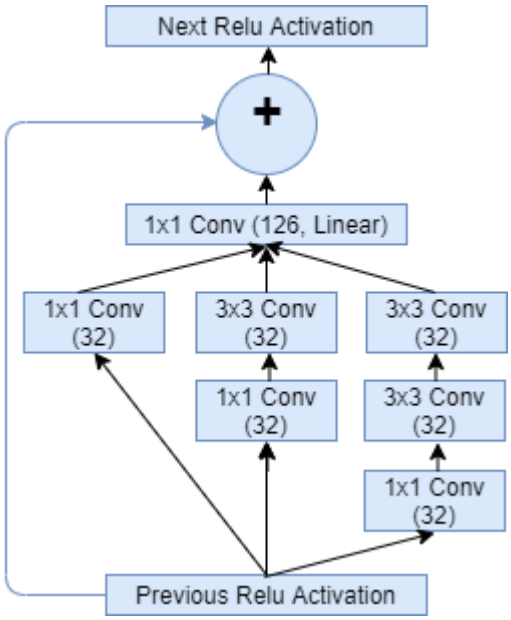

FIGURE 4. Modified Inception-ResNet-A module.

FIGURE 5. Samples for all fruit types are shown in following order: (1) Pomegranate, (2) Tomato, (3) Orange, (4) Apple, (5) Peach, and (6) Almond.

A. DATASET

The dataset consists of 6 different fruits including Apple, Almond, Orange, Peach, Tomato, and Pomegranate. Sample image of each fruit type is shown in Fig 5. 4820 original images are augmented to 32,412 images, and used 80% for the training of segmentation module, while remaining are used for validation. Although images for segmentation are gathered from different datasets including Google Images, and have different sizes, they are resized to 900 × 650 pixels. Fifty images of each fruit type are used to test the system's accuracy. Annotation of dataset especially for segmentation is very tedious to obtain for huge dataset, with this reason, different augmentation techniques are used to enlarge the dataset. Data augmentation is essential to teach the network the desired invariance and robust properties, when only few training samples are available.

We have applied the transformation with generalization ability [52]. Commonly used transformations, such as left-right flipping, elastic deformations [53], and rotation, are applied. These transformations are also applied with the same parameters on corresponding mask images. Breakdown of the images after augmentation against each fruit type used for segmentation training is given in Tab. 1.

For the training of the counting module, 21,450 sub patches are extracted from the original images, and each

TABLE 1. Type wise breakdown of the dataset used.

Fruit type	Original Images	After Augmentation	Training images	Testing images
Apple	1120	8174	6529	1635
Orange	950	5541	4432	1109
Tomato	680	4902	3922	980
Pomegranate	560	3829	3064	765
Almond	620	4783	3826	957
Peach	890	5183	4150	1036

patch contains 0 to 11 fruit instances. Maximum value of the assigned label was 11. We also used augmentation techniques, such as lift-right flipping, color changes, and rotation, to enlarge, but preserve the assigned label simultaneously. After augmentation, we have 89,120 sub-images divided into training, validation and test sets. 88, 820 sub-images are used for training, while 17,764 are used for validation which becomes almost 20% of the training set. Finally, 300 sub-images, having 50 images of each fruit type are used to test the counting accuracy.

B. TRAINING SETUP

For segmentation, the network was trained for 2,000 epochs, with batch size of 16, over the augmented dataset. To minimize the error, SGD-momentum was used with learning rate 0.02, momentum 0.8, and weight decay 0.0002. Xavier initializer was used to initialize the parameters [54]. For the training of the counting module, Adam optimizer was involved in having learning rate and weight decay equal to 0.0001. Then, network was trained for 150,000 epochs with batch size 32. Both networks were implemented using Keras on a machine having 16 GB RAM, and Nvidia 1080Ti GPU.

C. ANALYSIS & COMPARISONS

The proposed approach has been evaluated qualitatively and results are also compared in state of art techniques on fruit counting based on fruit types. Loss and accuracy graphs of both segmentation and counting modules are also presented in Fig 6 to Fig 9.

1) SEGMENTATION MODULE

Here, evaluation of segmentation against three metrics are given. First, performance of proposed segmentation module is assessed against generating loose binary segmentation, and precision, recall, and accuracy are calculated. Values for precision (∼87) and recall (∼84) are seemed low since the ground-truth masks are loosely annotated; however, loosely marked contours involve almost all the fruit patches in the image. We have visually examined the test segmentation result and find almost no fruit containing region undetected by the segmentation network. Nevertheless, higher segmentation accuracy is achieved. The precision, recall and accuracy score are shown in Tab. 2; TP is the number of true positives (correct segmentation), TN is the true negative, FP is the number of false positives (false segmentation), and FN is the number of false negatives.

TABLE 2. Evaluation matrix of binary segmentation.

Metric	%
Precision = TP / (TP+FP)	86.69
Recall = TP / (TP + FN)	84.48
Accuracy = (TP + TN) / All	95.52

FIGURE 6. Graph of training and validation accuracies of segmentation module.

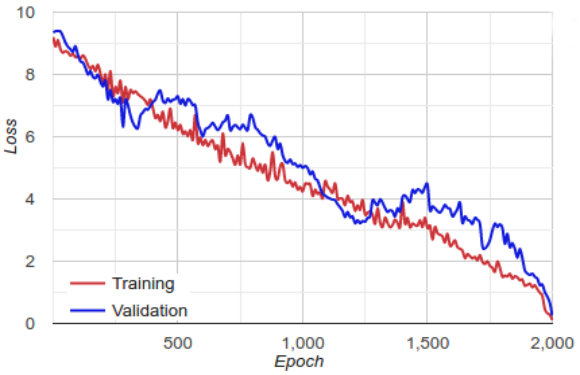

FIGURE 7. Training and validation loss graph of segmentation module.

The segmentation module is trained for 2,000 epochs and the final training and validation accuracies are 95.5% and 87.8% respectively. From the gap of training and validation accuracy, it can be concluded that the model is slightly overfitting the training data which is a curse associated with deep learning models. The accuracy graph in Fig. 6 shows the training and validation accuracies corresponding to epochs.

Segmentation loss for training is started decreasing from 9.2 and lowered to 0.11, while validation loss is reduced from 9.35 to 0.25 after 2,000 epochs. The graph in Fig 7 shows the loss journey throughout training.

2) COUNTING MODULE

Counting module is training for 150,000 epochs with approximately 89,000 images divided into training and validation sets with 80% and 20% ratios. Training and validation losses (Fig. 8) started reducing from 8.6 approximately, but training loss went to 0.07 and validation loss ended up at 0.12 at the final epoch. Validation loss went lower to training loss at some epoch but remained high most of the training time.

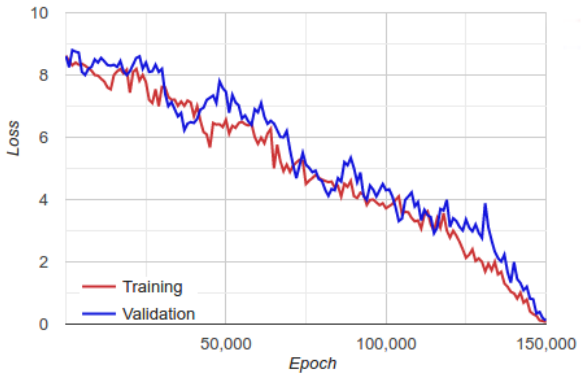

FIGURE 8. Loss graph of counting module for training and validation data.

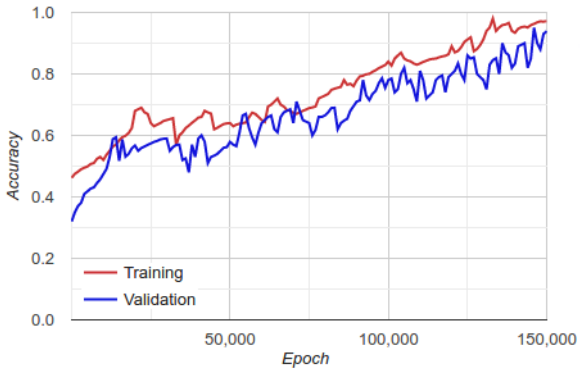

FIGURE 9. Training and validation accuracy graph of counting module.

TABLE 3. Accuracy against each category.

Fruit type	Testing Images	Actual Count	Predicted Counted	Accuracy
Apple	50	492	473	96.2
Orange	50	553	515	93.1
Peach	50	454	426	93.8
Tomato	50	604	557	92.3
Almond	50	671	602	89.6
Pomegranate	50	380	344	90.5
Overall	300	3154	2917	92.5

From Fig. 9, it could be seen that counting modules also faced overfitting as there is a difference between the training and validation accuracy where validation accuracy remained lower than training accuracy. At the end of the last epoch, training and validation accuracy ended at 97.2% and 93.9%, respectively.

Finally, the counting module is evaluated by comparing the predicted fruit count with ground-truth count. During training, 97.57% accuracy is achieved but achieved lower to 92.5% on average against all fruits during testing. Apple achieved the highest 96.2% accuracy, while Almond 89.5 slowest among all the fruit types. Below in Tab. 3, a breakdown of test accuracies is given against each fruit type.

In Tab. 4, we have compared achieved result in state of art on same datasets. Häni et al [56] used the same apple dataset had achieved 94% counting accuracy, while we achieved 96.2%. Dorj et al [34] had reached 93% accuracy on orange

TABLE 4. Results comparison in state of the art approaches.

Method	Apple (%)	Tomato (%)	Orange (%)	Almond (%)	Pomegranate (%)	Peach (%)
Roy et al [55]	91.3	-	-	-	-	-
Häni et al [56]	94	-	-	-	-	-
Rahnemoonfar et al [7]	-	91.03	-	-	-	-
Malik et al [57]	-	-	91.30	-	-	-
Dorj et al [34]	-	-	93	-	-	-
Wang et al [60]	-	-	85.6	-	-	-
Li et al [59]	-	-	84.6	-	-	-

counting, while we have gained 93.1%. We have achieved 92.3% on Tomato counting, while Rahnemoonfar et al [7] achieved 91.03%. Overall, we have performed 94.71% counting accuracy on all six fruit types.

V. CONCLUSION

Best to our knowledge, this was the first attempt to involve multiple fruit types to estimate fruit yield simultaneously. Almost all the known fruits have some common characteristics, such as circular shape, skin texture, and background, making it a suitable fit to count the lack of a big dataset for a single fruit type. Through shared features, we made a single pipeline for fruit counting. Moreover, relax segmentation mitigates the unnecessary process of the image regions where fruit instances are not present. It's very difficult to obtain the exact mask of the image, so loose segmentation allows to extract the cluster regions for further processing to count the instances. Use of SegNet makes the segmentation generation faster due to a smaller number of parameters. As established in the literature, regression method shows state-of-the-art results, and the involvement of inception-ResNet-A incurs higher accuracy and lower the computation cost.

In the future, we plan to involve more fruit types and build a counting mechanism for video which will eventually converted into mobile application.

ACKNOWLEDGMENT

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

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest to report regarding the present study and all authors contributed equally scientifically.

REFERENCES

[1] P. J. Cullen, V. P. Valdramidis, B. K. Tiwari, S. Patil, P. Bourke, and C. P. O'Donnell, "Ozone processing for food preservation: An overview on fruit juice treatments," *Ozone, Sci. Eng.*, vol. 32, no. 3, pp. 166–179, Jun. 2010, doi: 10.1080/01919511003785361.

[2] H. Mukhtar, M. Z. Khan, M. U. G. Khan, T. Saba, and R. Latif, "Wheat plant counting using UAV images based on semi-supervised semantic segmentation," in *Proc. 1st Int. Conf. Artif. Intell. Data Anal. (CAIDA)*, Apr. 2021, pp. 257–261.

[3] Z. S. Pothen and S. Nuske, "Texture-based fruit detection via images using the smooth patterns on the fruit," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Stockholm, Sweden, May 2016, pp. 5171–5176.

[4] S. Aich, A. Josuttes, I. Ovsyannikov, K. Strueby, I. Ahmed, H. S. Duddu, C. Pozniak, S. Shirtliffe, and I. Stavness, "DeepWheat: Estimating phenotypic traits from crop images with deep learning," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 323–332.

[5] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," in *Proc. Sci. Inf. Conf.*, Cham, Switzerland, 2019, pp. 128–144.

[6] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, May 2017, pp. 3626–3633.

[7] M. Rahnemoonfar and C. Sheppard, "Deep count: Fruit counting based on deep simulated learning," *Sensors*, vol. 17, no. 4, p. 905, Apr. 2017, doi: 10.3390/s17040905.

[8] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1324–1332.

[9] S. W. Chen, S. S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C. J. Taylor, and V. Kumar, "Counting apples and oranges with deep learning: A data-driven approach," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 781–788, Apr. 2017, doi: 10.1109/LRA.2017.2651944.

[10] F. Lateef and Y. Ruichek, "Survey on semantic segmentation using deep learning techniques," *Neurocomputing*, vol. 338, pp. 321–348, Apr. 2019, doi: 10.1016/j.neucom.2019.02.003.

[11] M. Thoma, "A survey of semantic segmentation," 2016, *arXiv:1602.06541*.

[12] V. Rabaud and S. Belongie, "Counting crowded moving objects," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jun. 2006, pp. 705–711.

[13] N. Ahuja and S. Todorovic, "Extracting texels in 2.1D natural textures," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[14] P. Chattopadhyay, R. Vedantam, R. R. Selvaraju, D. Batra, and D. Parikh, "Counting everyday objects in everyday scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1135–1144.

[15] C. C. Loy, K. Chen, S. Gong, and T. Xiang, "Crowd counting and profiling: Methodology and evaluation," in *Modeling, Simulation and Visual Analysis of Crowds*, vol. 11. New York, NY, USA: Springer, 2013, pp. 347–382.

[16] P. Tu, T. Sebastian, G. Doretto, N. Krahnstoever, J. Rittscher, and T. Yu, "Unified crowd segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Heidelberg, Germany, 2008, pp. 691–704.

[17] V. Rabaud and S. Belongie, "Counting crowded moving objects," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jun. 2006, pp. 705–711.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland, 2016, pp. 21–37.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[21] M. Z. Khan, S. Harous, S. U. Hassan, M. U. G. Khan, R. Iqbal, and S. Mumtaz, "Deep unified model for face recognition based on convolution neural network and edge computing," *IEEE Access*, vol. 7, pp. 72622–72633, 2019, doi: 10.1109/ACCESS.2019.2918275.

[22] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 11–19.

[23] Y. Song, C. A. Glasbey, G. W. Horgan, G. Polder, J. A. Dieleman, and G. W. A. M. van der Heijden, "Automatic fruit recognition and counting from multiple images," *Biosyst. Eng.*, vol. 118, pp. 203–215, Feb. 2014, doi: 10.1016/j.biosystemseng.2013.12.008.

[24] W. Maldonado and J. C. Barbosa, "Automatic green fruit counting in orange trees using digital images," *Comput. Electron. Agricult.*, vol. 127, pp. 572–581, Sep. 2016, doi: 10.1016/j.compag.2016.07.023.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[26] W. S. Qureshi, A. Payne, K. B. Walsh, R. Linker, O. Cohen, and M. N. Dailey, "Machine vision for counting fruit on mango tree canopies," *Precis. Agricult.*, vol. 18, no. 2, pp. 224–244, Apr. 2017, doi: 10.1007/s11119-016-9458-5.

[27] S. Bargoti and J. P. Underwood, "Image segmentation for fruit detection and yield estimation in Apple orchards," *J. Field Robot.*, vol. 34, no. 6, pp. 1039–1060, 2017, doi: 10.1002/rob.21699.

[28] X. Liu, S. W. Chen, S. Aditya, N. Sivakumar, S. Dcunha, C. Qu, C. J. Taylor, J. Das, and V. Kumar, "Robust fruit counting: Combining deep learning, tracking, and structure from motion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 1045–1052.

[29] J. M. Ponce, A. Aquino, B. Millan, and J. M. Andújar, "Automatic counting and individual size and mass estimation of olive-fruits through computer vision techniques," *IEEE Access*, vol. 7, pp. 59451–59465, 2019, doi: 10.1109/ACCESS.2019.2915169.

[30] N. Häni, P. Roy, and V. Isler, "A comparative study of fruit detection and counting methods for yield mapping in apple orchards," *J. Field Robot.*, vol. 37, no. 2, pp. 263–282, Mar. 2020, doi: 10.1002/rob.21902.

[31] E. Bellocchio, T. A. Ciarfuglia, G. Costante, and P. Valigi, "Weakly supervised fruit counting for yield estimation using spatial consistency," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2348–2355, Jul. 2019, doi: 10.1109/LRA.2019.2903260.

[32] S. Tu, J. Pang, H. Liu, N. Zhuang, Y. Chen, C. Zheng, H. Wan, and Y. Xue, "Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images," *Precis. Agricult.*, vol. 21, no. 5, pp. 1072–1091, Oct. 2020, doi: 10.1007/s11119-020-09709-3.

[33] R. Linker, "A procedure for estimating the number of green mature apples in night-time orchard images using light distribution and its application to yield estimation," *Precis. Agricult.*, vol. 18, no. 1, pp. 59–75, Feb. 2017, doi: 10.1007/s11119-016-9467-4.

[34] U.-O. Dorj, M. Lee, and S.-S. Yun, "An yield estimation in citrus orchards via fruit detection and counting using image processing," *Comput. Electron. Agricult.*, vol. 140, pp. 103–112, Aug. 2017, doi: 10.1016/j.compag.2017.05.019.

[35] P. Roy, A. Kislay, P. A. Plonski, J. Luby, and V. Isler, "Vision-based preharvest yield mapping for apple orchards," *Comput. Electron. Agricult.*, vol. 164, Sep. 2019, Art. no. 104897, doi: 10.1016/j.compag.2019.104897.

[36] D. Onoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland, 2016, pp. 615–629.

[37] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the point: Semantic segmentation with point supervision," in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland, 2016, pp. 549–565.

[38] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. M. Schmidt, "Where are the blobs: Counting by localization with point supervision," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 547–562.

[39] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

[40] S. Aich and I. Stavness, "Leaf counting with deep convolutional and deconvolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 2080–2089.

[41] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[42] F. Sílvio and L. A. Alexandre, "From the human visual system to the computational models of visual attention: A survey," *Artif. Intell. Rev.*, vol. 39, no. 1, pp. 1–47, 2013, doi: 10.1007/s10462-012-9385-4.

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.

[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[46] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," 2016, *arXiv:1602.07261*.
[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.
[48] X. Chen, R. Guo, W. Luo, and C. Fu, "Visual crowd counting with improved inception-ResNet—A module," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Kuala Lumpur, Malaysia, Dec. 2018, pp. 112–119.
[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[50] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.
[51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[52] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition," 2015, *arXiv:1501.02876*.
[53] P. Y. Simard, D. Steinkraus, and J. C. Plat, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. InIcdar*, vol. 3, 2003, pp. 1–6.
[54] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[55] P. Roy and V. Isle, "Vision-based apple counting and yield estimation," in *Proc. Int. Symp. Exp. Robot.*, 2016, pp. 478–487.
[56] N. Hani, P. Roy, and V. Isler, "Apple counting using convolutional neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 2559–2565.
[57] Z. Malik, S. Ziauddin, A. R., and A. Safi, "Detection and counting of on-tree citrus fruit for crop yield estimation," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 5, pp. 1–5, 2016.
[58] M. Z. Khan, S. U. Hassan, H. U. Draz, M. U. G. Khan, and R. Iqbal, "Detection and identification of vehicles from high-resolution aerial images using deep learning approaches with the tuned parameters," in *Unmanned Aerial Vehicles in Smart Cities*, 1st ed. Cham, Switzerland: Springer, 2020, pp. 65–83.
[59] H. Li, W. S. Lee, and K. Wang, "Immature green citrus fruit detection and counting based on fast normalized cross correlation (FNCC) using natural outdoor colour images," *Precis. Agricult.*, vol. 17, no. 6, pp. 678–697, 2016.
[60] C. Wang, W. S. Lee, X. Zou, D. Choi, H. Gan, and J. Diamond, "Detection and counting of immature green citrus fruit based on the local binary patterns (LBP) feature using illumination-normalized images," *Precis. Agricult.*, vol. 19, no. 6, pp. 1062–1083, Dec. 2018.
[61] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.
[62] V. Nair and G. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010, pp. 807–814.
[63] J. Xiong, Z. Liu, S. Chen, B. Liu, Z. Zheng, Z. Zhong, Z. Yang, and H. Peng, "Visual detection of green mangoes by an unmanned aerial vehicle in orchards based on a deep learning method," *Biosyst. Eng.*, vol. 194, pp. 261–272, Jun. 2020.

AMJAD REHMAN KHAN (Senior Member, IEEE) received the Ph.D. and Postdoctoral degrees (Hons.) from the Faculty of Computing, Universiti Teknologi Malaysia, with a specialization in forensic documents analysis and security, in 2010 and 2011, respectively. He is currently a Senior Researcher with the Artificial Intelligence & Data Analytics Laboratory, CCIS, Prince Sultan University, Riyadh, Saudi Arabia. He is also a PI in several funded projects and also completed projects funded from MOHE (Malaysia) and Saud Arabia. He is the author of more than 200 ISI journal articles and conferences. His research interests include data mining, health informatics, and pattern recognition. He received the Rector Award for 2010 Best Student from Universiti Teknologi Malaysia.

HAMZA MUKHTAR received the M.S. degree in computer science from the University of Engineering and Technology Lahore, Pakistan. He is currently working as a Research Officer with the Intelligent Criminology Laboratory, National Center of Artificial Intelligence, Al-Khawarizmi Institute of Computer Science, UET Lahore. His research interests include computer vision, natural language processing, machine learning, and deep learning.

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 working as an Associate Professor with the College of Computer and Information Sciences, Prince Sultan University (PSU), Riyadh, Saudi Arabia. She has published more than 100 publications in high ranked journals. Her research interests include bioinformatics, data mining, and classification. She was awarded best research of the year award at PSU, from 2013 to 2016. Due to her excellent research achievement, she is included in Marquis Who's Who (S & T) 2012. She won the Best Student Award with the Faculty of Computing, UTM, in 2012. She is also an editor of several reputed journals and on panel of TPC of international conferences.

OMER RIAZ received the B.S. degree in CS from the University of Engineering and Technology Lahore, in 2004, the master's degree in system level integration from the University of Edinburgh, in 2005, and the Ph.D. degree in high performance computing from the University of Strathclyde, U.K., in 2014. His research interests include study of challenges involve implementing time consuming algorithms on advance computer architectures (shared memory, distributed memory, and GPU) and numerical analysis and machine learning algorithms.

MUHAMMAD USMAN GHANI KHAN is currently a Professor with the Department of Computer Science & Engineering, University of Engineering and Technology Lahore, Pakistan. His recent works are concerned with multimedia, incorporating text, and audio and visual processing into one frame work. He is currently heading the Intelligent Criminology Laboratory, NCAI, KICS, UET Lahore.

SAEED ALI BAHAJ received the Ph.D. degree from Pune University, India, in 2006. He is currently an Assistant Professor with Prince Sattam Bin Abdulaziz University. His research interests include artificial intelligence, information management, forecasting, information engineering, big data, and information security.