

C O M M U N I C A T I O N S

FACULTY OF SCIENCES
UNIVERSITY OF ANKARA

DE LA FACULTE DES SCIENCES
DE L'UNIVERSITE D'ANKARA

Series A2-A3: Physical Sciences and Engineering

VOLUME: 65

Number: 2

YEAR: 2023

Faculty of Sciences, Ankara University
06100 Beşevler, Ankara –Türkiye
ISSN 1303-6009 E-ISSN 2618-6462

C O M M U N I C A T I O N S

FACULTY OF SCIENCES
UNIVERSITY OF ANKARA

DE LA FACULTE DES SCIENCES
DE L'UNIVERSITE D'ANKARA

Series A2-A3: Physical Sciences and Engineering

Volume: 65

Number: 2

Year: 2023

Owner

Sait HALICIOĞLU, Dean of Faculty of Sciences

Editor in Chief

Fatma KARAKOÇ, Ankara University

Managing Editor

Şengül KURU, Ankara University

Area Editors

İnanç ŞAHİN (Physics) Ankara University, Türkiye	İman ASKERZADE (Askerbeyli) (Computer Eng.) Ankara University, Türkiye
Handan OLĞAR (Engineering Physics) Ankara University, Türkiye	Ziya TELATAR (Electronic Eng.) Başkent University, Türkiye
H. Volkan ŞENAVCI (Astronomy) Ankara University, Türkiye	M. Emin CANDANSAYAR (Geophysical Eng.) Ankara University, Türkiye

Editors

Ramiz ALIGULIYEV ANAS, Azerbaijan Murat EFE Ankara University, Türkiye Mustafa E. KAMASAK Istanbul Technical University, Türkiye Javier NEGRO Univesidad de Valladolid, Spain Emre YENGEL King Abdullah Uni. of Sci.Tech., Saudi Arabia	Gabriela CIUPRINA Politehnica Univeristy of Bucharest, Romania Osman EROGLU TOBB Economy and Tech. Uni., Türkiye İlhan KOSALAY Ankara University, Türkiye Miroslav VOZNAK VŠB – Tech.Uni. of Ostrava, Czech Republic A. Egemen YILMAZ Ankara University, Türkiye	Sara CRUZ Y CRUZ SEPI-UPIITA IPN, Mexico H. Gokhan İLK Ankara University, Türkiye İsa NAVRUZ Ankara University, Türkiye Hakan TORA Atılım University, Türkiye Kutluay YUCE Ankara University, Türkiye
--	---	--

Language Editor

Fatma KARAKOÇ, Ankara University

Production Editor

Şengül KURU, Ankara University

This Journal is published two issues in a year by the Faculty of Sciences, University of Ankara. Articles and any other material published in this journal represent the opinions of the author(s) and should not be construed to reflect the opinions of the Editor(s) and the Publisher(s).

Correspondence Address:

COMMUNICATIONS
EDITORIAL OFFICE
Ankara University, Faculty of Sciences,
06100 Beşevler, ANKARA – TÜRKİYE
Tel: (90) 312-216 87 00 Fax: (90) 312-216 89 00
e-mail: commun@science.ankara.edu.tr
<http://communications.science.ankara.edu.tr/index.php?series=A2A3>

Print:

Ankara University Press
İncitaş Sokak No:10 06510 Beşevler
ANKARA – TÜRKİYE

C O M M U N I C A T I O N S

FACULTY OF SCIENCES
UNIVERSITY OF ANKARA

DE LA FACULTE DES SCIENCES
DE L'UNIVERSITE D'ANKARA

Series A2-A3: Physical Sciences and Engineering

Volume: 65

Number: 2

Year: 2023

Research Articles

Ş. ÖZSARI, F.Z. ORTAK, M.S. GÜZEL, M.B. BAŞKIR, G.E. BOSTANCI, ML based prediction of COVID-19 diagnosis using statistical tests.....	79
H.A. ILGIN, F.A. AYDEMİR, B. CEDİMOĞLU, M.N. AYDIN, H. SİLLELİ, Comparative analysis of mature tomato detection by feature extraction and machine learning for autonomous greenhouse robots.....	100
S. SERTTAŞ, E. DENİZ, Disease detection in bean leaves using deep learning.....	115
U.N. AKTAN, M. DİKMEN, An optimized artificial neural network for estimating design effort of jigs and fixtures used in aviation industry	130
M.E. TUNALIOĞLU, H.Ö. ÇILDIROĞLU, A.U. YILMAZER, On the geometric phases in entangled states.....	142
B. ÖZBEY, A novel alternative in wireless and passive sensing: the bended nested split-ring resonator.....	152
M. KALKAN, Y. AR, Classification of human activities by smart device measurements.....	166

ML BASED PREDICTION OF COVID-19 DIAGNOSIS USING STATISTICAL TESTS

Sifa OZSARI¹, Fatima Zehra ORTAK², Mehmet Serdar GUZEL¹, Mukerrem Bahar BASKIR³,
Gazi Erkan BOSTANCI¹

¹Department of Computer Engineering, Ankara University, Ankara, TÜRKİYE

²MSc at Big Data and Business Analytics, Istanbul Technical University,
İstanbul, TÜRKİYE


³Department of Statistics, Bartın University, Bartın, TÜRKİYE

ABSTRACT. The first case of the novel Coronavirus disease (COVID-19), which is a respiratory disease, was seen in Wuhan city of China, in December 2019. From there, it spread to many countries and significantly affected human life. Deep learning, which is a very popular method today, is also widely used in the field of healthcare. In this study, it was aimed to determine the most suitable Deep Learning (DL) model for diagnosis of COVID-19. A popular public data set, which consists of 2482 scans was employed to select the best DL model. The success of the models was evaluated by using different performance evaluation metrics such as accuracy, sensitivity, specificity, precision, F1 score, kappa and AUC. According to the experimental results, it has been observed that DenseNet models, AdaGrad and NADAM optimizers are effective and successful. Also, whether there are statistically significant differences in each performance measure/score of the architectures by the optimizers was observed with statistical tests.


1. INTRODUCTION

The Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) virus, which causes the novel Coronavirus disease (COVID - 19), belongs to the family of coronaviruses, which are large enveloped, positive single-stranded RNA viruses that can infect humans and animals [1]. This disease spread rapidly around the world and have had a serious impact on the health and life of many people [2]. COVID-19, which emerged in November 2019 and defined as an epidemic by the World Health Organization (WHO), is very contagious. The lack of vaccine when it first appeared


Keywords. COVID-19, deep learning, CT images, statistical analysis.

✉ ozsaris@ankara.edu.tr-Corresponding author;  0000-0002-0531-4645

✉ fatmazolehraortak@gmail.com;  0000-0002-6420-9116

✉ mguzel@ankara.edu.tr;  0000-0002-3408-0083

✉ baharbaskir@yahoo.com;  0000-0002-1107-0659

✉ ebostanci@ankara.edu.tr;  0000-0001-8547-7569.

is one of the main reasons why the virus is dangerous. Therefore, it is very important to detect the disease quickly and isolate the infected person immediately in order to prevent the spread of the disease [3]. Reverse Transcription Polymerase Chain Reaction (RT-PCR) is the gold standard to diagnose COVID-19 [3–6]. It is performed by detecting the RNA virus that causes disease from sputum or nasopharyngeal swab [3]. However, in addition to the limited number of materials, a certain period of time must pass for the results to get. Chest imaging methods like Computed Tomography (CT) or X-ray are an effective techniques and could be used for diagnosis [2, 3, 5, 7–9]. While X-ray shows visual signs associated with COVID-19 [10], CT images have a high sensitivity for diagnosing COVID-19 [4].

In recent years, Artificial Intelligence (AI) has a major and fast growth in solving the complex subjects in some fields including engineering, medicine, economy. Specifically, Deep Learning (DL) a major area of AI have become very popular in medical applications. Previously, most things were done manually by doctors. With DL, this time-consuming process has started to improve [11]. This has attracted great interest in the proposal and development of deep learning-based studies for the diagnosis of COVID-19 using both CT and X-ray samples such as [12–18].

One of the studies performed on CT images is [15] by Ardakani et al. In their study, they used ten Convolutional Neural Network (CNN) (AlexNet [19], VGG-16, VGG-19 [20], SqueezeNet [21], GoogleNet [22], MobileNetV2 [23], Residual Neural Network (ResNet)-18, ResNet-50, ResNet-101 [24], and Xception [25]) to diagnose COVID-19. 1020 CT slices were used. The number of COVID-19 patients was 108 (laboratory proven) and the number of patients without COVID-19 was 86. The non-COVID-19 group included those with other atypical and viral pneumonia diseases. In all networks, Stochastic Gradient Descent with Momentum (SGDM) was used for the optimizer, 0.01 for the initial learning rate and 5 for the validation frequency. 80% of the data set was employed for training and 20% was considered for validation. The training and validation data set is the same for all networks. The data set was shuffled at each epoch. When the training process did not change remarkably, the training process was stopped. It is noted that among the networks, Xception and ResNet-101 and provided the best performance. Another study is [14]. In this study, authors constructed a multi-view deep learning fusion model, based on the modification of ResNet-50 architecture. They aimed to differentiate the COVID-19 patient with using computed tomography images. Chest CT images of 495 patients were obtained from different hospitals located in China. The data sets were randomly divided into the training set (395 cases), the validation set (50 cases) and the test set (50 cases). For the training set, 294 cases were diagnosed as COVID-19 and 101 were diagnosed as pneumonia. In validation and test sets, 37 cases were diagnosed as COVID-19 and 13 other was pneumonia. RMSprop optimizer with a learning rate of 1×10^{-5} and batch-size of 4 was used to update the parameters of network during training phase. In the study carried out by Singh et al. [17], a CNN model was employed to classify whether the patients

as infected or not. They used the chest CT images. Hyperparameters of CNN, which were kernel size, kernel type, number of epochs, learning rate, padding, stride, hidden layer, activation functions, momentum and batch size, were regulated by using Multi Objective Differential Evolution (MODE) algorithm. Jaiswal et al. [26] proposed a Dense Convolutional Network (DenseNet)-201 based [27] deep transfer learning model to classify patients as COVID-19 infected or not based on chest CT images. They utilized the proposed model to extract feature, followed by appropriate classifiers. A data set consisting of 2492 CT scans available on kaggle was used for experiments. Also, they augmented the data for obtaining higher accuracy. In the study conducted by Wang et al. [28], a weakly-supervised deep learning framework was developed for both COVID-19 classification and lesion localization problems. They segmented the lung area using a pre-trained U-Net [29], then in order to predict the possibility of COVID-19, the segmented 3D lung area was fed into a 3D deep neural network. 499 and 131 3D CT volumes were used for training and testing, respectively. In training of the network, they used Adaptive Moment Estimation (ADAM) [30] optimizer with a constant $1e - 5$ learning rate. Epoch size was taken as 100. Another relevant study using deep learning and CT images is [31]. Chen et al. built their study on UNet++ [32] and used Resnet-50 as the base of UNet++. ResNet-50 is pre-trained on the ImageNet dataset. All pre-training parameters of ResNet-50 were transmitted to UNet++. 46,096 anonymous images were used for model creation and validation. Ying et al. [33] designed a pre-trained ResNet-50 model with the addition of Feature Pyramid Network (FPN) to extract the top-K details from CT images. The data set consists of chest CT scans of 88 patients diagnosed with COVID-19, 86 healthy people and 101 patients infected with bacterial pneumonia. The model is capable of both determining the most important part of the images and interpreting the outputs of the neural network using FPN and attention modules (to learn the importance of every detail). In the study of Gozes et al. [34], they first extracted the relevant lung area using segmentation. For this, they trained U-Net architecture using 6,150 CT slices of cases with lung abnormalities. Then, they utilized Resnet-50-2D with fine-tuned parameters to detect coronavirus-related abnormalities. He et al. [35] aimed to develop deep learning methods that can give high diagnostic accuracy rate even the training CT samples are limited. They presented an Self-Trans approach. In order to reduce the over-fitting risk, contrastive self-supervised learning was combined with transfer learning for learning robust and unbiased feature representations. Besides, they published a public data set containing hundreds of COVID-19 positive CT scans.

This study, on the other hand, claims to obtain the best model by using different Deep Learning (DL) architectures with varying optimizers, tested on a public

CT data set [36]. While DenseNet-121, DenseNet-201, DenseNet-169 [27], MobileNetV2 [23], VGG16, VGG19 [20], U-Net [29] and ResNet-50 [24] were determined as architectures, different optimizers, involving Stochastic Gradient Descent (SGD), Adaptive Gradient Algorithm (AdaGrad) [37], ADAM, RMSProp and Nesterov-Accelerated Adaptive Moment estimation (NADAM) [38] were integrated into the models. In addition, statistical tests were used to examine whether there were statistically significant differences in each performance measure/score of the architectures by the optimizers.

The structure of the rest of this paper is as follows. In Section 2, the data set, architectures and optimizers used in the study are presented. In section 3, model parameters are detailed regarding models. Experimental results, success of the models and statistical tests are also discussed in this section, whereas, Section 4 concludes the study.

2. MATERIAL AND METHOD

In this section, information about the data set and models used in the study is detailed. Subsections involve Data set and Architecture.

2.1. Data set. In this study, a benchmark public data set (available in Kaggle) consisting of 2482 CT scans of patients [36] is employed. The data set contains 1252 SARS-CoV-2 infected CT scans and 1230 CT scans non-infected by SARS-CoV-2. Patients who are not infected by COVID-19 have other pulmonary diseases. The data was collected from hospitals of Sao Paulo, Brazil. Figure 1 illustrates sample CT scans from this data set. 2234 of these CT scans were used for training, whereas the remaining 248 scans were employed for testing.

2.2. Architectures.

2.2.1. DenseNet. DenseNet is one of the leading DL architecture, connecting each layer to following layers in a feed-forward approach, was proposed in [27]. With this architecture, it is aimed to deepen deep learning networks, to be more accurate, as well as to make them more efficient to train by using shorter connections between layers. There are several important advantages of DenseNets, which are detailed as follows [27]:

- They reduce the vanishing-gradient problem.
- They strengthen feature propagation.
- They encourage feature reuse.
- They quite decrease the number of parameters.

In addition, they showed that DenseNets scales to hundreds of layers without optimization difficulties.

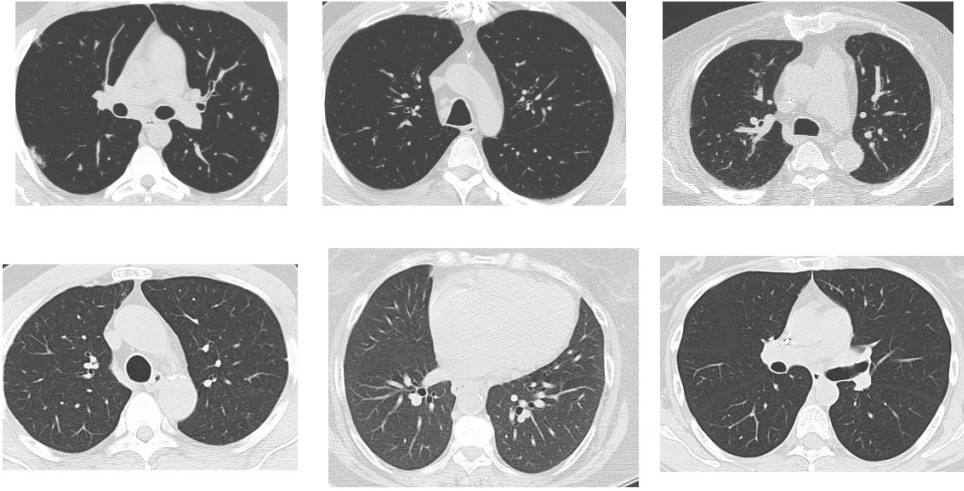


FIGURE 1. Sample CT scans [36].

2.2.2. *MobileNetV2*. MobileNetV2, introduced by [23], is a neural network architecture specifically developed for resource constrained environments, such as mobile platforms. MobileNetV2 architecture involves a novel layer, providing low-dimensional compressed representation for input data. Firstly, this representation is extended to high dimension and filtered with a “lightweight depth-wise convolution”. Afterwards, features are projected back to a low-dimensional representation by using a linear convolution filter. [23]. Although MobileNetV2 uses deep separable convolution, the point where it differs is that it has a bottleneck residual block rather than a deep separable convolution block.

2.2.3. *VGG*. VGG16 is a convolutional neural network model, achieved to win the first and second place in the localization and classification tracks respectively in “The ImageNet Large Scale Visual Recognition Challenge 2014”. It was proposed by [20]. During the training, the input of ConvNet is a fixed size 224×224 RGB image. The image passes through a convolutional layer stacks, in which filters (3×3) with a very small receptive are used. 1×1 convolution filters, which can be viewed as a linear transformation of the input channels (followed by nonlinearity), are used in one of the configurations. The padding is 1 pixel for 3×3 convolution layers and spatial pooling is performed with five max-pooling layers. Five max-pooling layers follow some convolution layers, but max-pooling (which is carried out over a 2×2 pixel window, with stride 2) does not follow all convolution layers. Three Fully-Connected (FC) layers follow a convolutional layers stack. In different

architectures, this convolutional layers stack may have different depth. The softmax, the classifier layer, is the final layer. In all networks, the configuration of fully connected layers is defined as identical can be seen in [20].

2.2.4. *U-Net*. U-Net [29] is a convolutional neural network developed for obtaining better segmentation performance in case of having limited amount of biomedical image data. It was proposed by Ronnenberger et al. [29]. In their study, they offered a network and training scheme based on the excessive use of data augmentation to employ existing annotated samples more effectively. It is an important change of the architecture they present to have a large number of feature channels in the upsampling section. These feature channels provide the network to pass context information to higher layers [29].

2.2.5. *ResNet*. ResNet architecture is one of the most popular deep neural networks available in many varieties with different number of layers. It was the winner of ILSVRC & COCO 2015 competitions on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation. ResNet was introduced by [24]. He et al. showed that the accuracy becomes saturated with increasing network depth, adding more layers to the network results in higher training error. In addition, they showed that this was not caused by overfitting contrary to popular belief. Due to the problem of vanishing/exploding gradients, training of deep networks is difficult. An identity shortcut connection that skips one or more layers was defined in ResNet. Thus, the number of layers can be increased without the problem of vanishing gradients.

2.3. **Optimizer**. Optimizer is used to reduce the loss, which is the difference between actual value and predicted value. The choice of these algorithm or method is very important. In this study, five different optimizers, which are SGD, AdaGrad, RMSProp, ADAM and NADAM, are used.

Stochastic gradient descent is a popular iterative method used to optimize an objective function. In SGD, which is a variant of Gradient Descent (GD), random samples are selected from training data in each iteration to update the parameter during optimization. Equation 1 is used for parameter updating:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x, y) \quad (1)$$

where, θ is a parameter, η is the learning rate, ∇ is the gradient and J is objective function. SGD performs one update at a time.

AdaGrad [37], which adapts the learning rate to the parameters, is an algorithm for gradient-based optimization [39]. Each parameter in AdaGrad has its own learning rate, so it eliminates manual adjustment of the learning rate. It decreases the learning rate of parameters proportionally to previous updates of parameters. AdaGrad makes large updates for infrequent parameters, while smaller updates for frequent parameters. The disadvantage of AdaGrad is that the system stops learning after a certain point due to the reduction of the learning rate.

RMSprop¹, which was proposed by Geoff Hinton, is an unpublished adaptive learning rate method. It is one of the algorithms developed due to the need to solve the decreasing learning rates problem in AdaGrad. RMSprop keeps the moving average of the squared gradients and divides the gradient by the root of this average. The update rule is as follows [39]:

$$\begin{aligned} E[g^2]_t &= \gamma E[g^2]_{t-1} + (1 - \gamma)g_t^2 \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}}g_t \end{aligned} \quad (2)$$

where, $E[g]$ is the running average. While Hinton recommends setting γ (moving average parameter) to 0.9, a good default value for η (learning rate) is 0.001.

The ADAM [30], which is an adaptive learning rate optimization algorithm, was introduced by Kingma and Ba in [30] study. They designed ADAM to combine the advantages of two popular methods, AdaGrad and RMSProp. Some of the advantages that ADAM has are as follows [30]:

- ADAM's step size is approximately limited by the step size hyper parameter.
- A fixed objective is not required.
- It works with sparse gradients.
- A form of step size annealing is naturally.
- The magnitudes of the parameter updates does not change with the rescaling of the gradient.

In the ADAM algorithm, adaptive learning rates are calculated for each parameter. It stores an exponentially decreasing average of past square of gradients, as in Adadelta and RMSprop. It also maintains an exponentially decreasing average of past gradients. ADAM update rule is given in the following equation [39]:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}}\hat{m}_t \quad (3)$$

where, \hat{m}_t and \hat{v}_t are first moment vector and the second moment vector respectively.

Finally, NADAM [38], which utilize ADAM optimizer with Nesterov Accelerated Gradient (NAG), is a variation of ADAM. NADAM can be used on noisy gradients as well as gradients with high curvatures. Equation 4 shows the equation of NADAM update rule.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}}\left(\beta_1\hat{m}_t + \frac{(1 - \beta_1)g_t}{1 - \beta_1^t}\right) \quad (4)$$

3. THE EXPERIMENTAL SECTION AND DISCUSSION

In this section, parameter values used in the study are explained and experiments have been carried out. Success of the models on data set is observed.

¹http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

3.1. Parameter Settings. In artificial neural networks, activation function generates the output of a node. Since it has an significant effect on success of neural network, the choice of activation function is very important for design of a neural network. In this study, while Rectified Linear Unit (ReLU) [40] is used in the hidden layer, sigmoid is used in the output layer. The batch-size is a hyperparameter that corresponds to the number of training examples to propagated through the network. Learning rate, that has an effect on updating weights of model at each iteration, is one of the important hyperparameters for deep neural networks. Iteration number, batch-size and learning rate values were set as follows:

- Iteration number: 50, 100
- Batch-size: 128
- Learning rate: Learning rate was determined using ReduceLROnPlateau. Started with 0.01, the minimum can be 0.00001.

3.2. Experimental Results. Models were trained separately on the training data set with each optimizer and iteration value. After each training, the models were tested. Accuracy, sensitivity (recall), specificity, precision, F1 score, kappa and Area Under the ROC Curve (AUC) were used to evaluate the performance of architectures. In equations 5, 6, 7, 8 and 9, the formulas for “accuracy”, “sensitivity”, “specificity”, “precision” and “F1 score” are given respectively.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (5)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (6)$$

$$Specificity = \frac{TN}{TN + FP} \quad (7)$$

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$F1 \text{ score} = 2 * \frac{precision * recall}{precision + recall} \quad (9)$$

For equations 5, 6, 7 and 8, TP corresponds to true positives, TN true negatives, FN false negatives and FP false positives. Table 1 indicates the performances of networks. In the table, iteration is abbreviated as iter, accuracy as acc, sensitivity as sens and specificity as spec, precision as prec and F1 score as F1.

TABLE 1. Performances of networks.

Networks	Optimizer	Iter	Acc	Sens	Spec	Prec	F1	Kappa	AUC
DenseNet-121	ADAM	50	0.94	0.94	0.95	0.94	0.94	0.89	0.94
DenseNet-121	AdaGrad	50	0.93	0.99	0.88	0.85	0.93	0.85	0.92
DenseNet-121	SGD	50	0.92	0.99	0.87	0.84	0.92	0.83	0.91
DenseNet-121	NADAM	50	0.90	0.90	0.90	0.89	0.90	0.79	0.89
DenseNet-121	RMSprop	50	0.92	0.96	0.89	0.87	0.92	0.83	0.91
DenseNet-121	ADAM	100	0.94	0.93	0.96	0.95	0.94	0.88	0.94
DenseNet-121	AdaGrad	100	0.91	0.99	0.85	0.81	0.91	0.81	0.90
DenseNet-121	SGD	100	0.91	0.99	0.85	0.81	0.91	0.81	0.90
DenseNet-121	NADAM	100	0.94	0.98	0.90	0.89	0.94	0.87	0.93
DenseNet-121	RMSprop	100	0.88	0.97	0.83	0.78	0.89	0.76	0.87
DenseNet-201	ADAM	50	0.93	0.96	0.90	0.89	0.93	0.86	0.92
DenseNet-201	AdaGrad	50	0.87	0.99	0.81	0.74	0.89	0.74	0.87
DenseNet-201	SGD	50	0.91	0.99	0.85	0.81	0.91	0.80	0.81
DenseNet-201	NADAM	50	0.94	0.93	0.95	0.94	0.94	0.87	0.93
DenseNet-201	RMSprop	50	0.89	0.97	0.83	0.78	0.90	0.77	0.88
DenseNet-201	ADAM	100	0.93	0.95	0.92	0.91	0.93	0.87	0.93
DenseNet-201	AdaGrad	100	0.91	0.99	0.86	0.83	0.92	0.82	0.91
DenseNet-201	SGD	100	0.93	0.97	0.89	0.87	0.93	0.85	0.92
DenseNet-201	NADAM	100	0.94	0.93	0.95	0.94	0.94	0.87	0.94
DenseNet-201	RMSprop	100	0.94	0.95	0.93	0.93	0.94	0.88	0.94
DenseNet-169	ADAM	50	0.93	0.96	0.91	0.89	0.93	0.86	0.93
DenseNet-169	AdaGrad	50	0.96	0.97	0.94	0.94	0.96	0.91	0.95
DenseNet-169	SGD	50	0.89	0.98	0.83	0.78	0.90	0.78	0.88
DenseNet-169	NADAM	50	0.88	0.82	0.94	0.94	0.88	0.76	0.88
DenseNet-169	RMSprop	50	0.79	0.99	0.71	0.57	0.83	0.58	0.78
DenseNet-169	ADAM	100	0.90	0.88	0.92	0.92	0.90	0.80	0.90
DenseNet-169	AdaGrad	100	0.93	0.98	0.88	0.86	0.93	0.85	0.92
DenseNet-169	SGD	100	0.86	0.99	0.79	0.72	0.88	0.73	0.85
DenseNet-169	NADAM	100	0.91	0.88	0.93	0.93	0.90	0.81	0.90
DenseNet-169	RMSprop	100	0.92	0.93	0.91	0.89	0.92	0.83	0.91

MobileNetV2	ADAM	50	0.76	0.98	0.69	0.52	0.81	0.52	0.75
MobileNetV2	AdaGrad	50	0.74	0.89	0.69	0.57	0.78	0.47	0.73
MobileNetV2	SGD	50	0.72	0.98	0.65	0.42	0.78	0.42	0.70
MobileNetV2	NADAM	50	0.91	0.95	0.88	0.86	0.91	0.82	0.90
MobileNetV2	RMSprop	50	0.87	0.93	0.82	0.78	0.88	0.73	0.86
MobileNetV2	ADAM	100	0.90	0.96	0.86	0.84	0.91	0.80	0.89
MobileNetV2	AdaGrad	100	0.72	0.71	0.74	0.73	0.72	0.44	0.72
MobileNetV2	SGD	100	0.90	0.97	0.75	0.64	0.84	0.63	0.81
MobileNetV2	NADAM	100	0.88	0.94	0.84	0.81	0.89	0.76	0.88
MobileNetV2	RMSprop	100	0.86	0.94	0.82	0.77	0.87	0.72	0.86
VGG16	ADAM	50	0.91	0.98	0.85	0.82	0.91	0.81	0.90
VGG16	AdaGrad	50	0.81	0.93	0.75	0.67	0.83	0.61	0.80
VGG16	SGD	50	0.86	0.92	0.82	0.78	0.86	0.70	0.85
VGG16	NADAM	50	0.89	0.97	0.85	0.81	0.91	0.79	0.89
VGG16	RMSprop	50	0.92	0.96	0.88	0.86	0.92	0.83	0.91
VGG16	ADAM	100	0.89	0.96	0.84	0.80	0.90	0.78	0.88
VGG16	AdaGrad	100	0.86	0.92	0.82	0.78	0.86	0.70	0.85
VGG16	SGD	100	0.87	0.93	0.84	0.80	0.88	0.74	0.87
VGG16	NADAM	100	0.92	0.96	0.90	0.89	0.93	0.85	0.92
VGG16	RMSprop	100	0.91	0.95	0.89	0.87	0.92	0.83	0.91
VGG19	ADAM	50	0.86	0.94	0.82	0.77	0.87	0.72	0.85
VGG19	AdaGrad	50	0.78	0.90	0.73	0.63	0.81	0.55	0.77
VGG19	SGD	50	0.73	0.80	0.71	0.65	0.76	0.46	0.73
VGG19	NADAM	50	0.89	0.93	0.87	0.84	0.90	0.79	0.89
VGG19	RMSprop	50	0.87	0.93	0.84	0.80	0.88	0.74	0.87
VGG19	ADAM	100	0.89	0.96	0.86	0.83	0.90	0.79	0.89
VGG19	AdaGrad	100	0.81	0.91	0.76	0.69	0.83	0.61	0.80
VGG19	SGD	100	0.73	0.83	0.71	0.63	0.76	0.47	0.73
VGG19	NADAM	100	0.89	0.93	0.87	0.85	0.90	0.79	0.89
VGG19	RMSprop	100	0.88	0.94	0.84	0.81	0.89	0.76	0.88
U-Net	ADAM	50	0.80	0.98	0.73	0.61	0.84	0.60	0.79
U-Net	AdaGrad	50	0.66	0.98	0.60	0.31	0.75	0.30	0.64
U-Net	SGD	50	0.64	0.99	0.59	0.26	0.74	0.26	0.63

U-Net	NADAM	50	0.82	0.97	0.75	0.64	0.84	0.63	0.81
U-Net	RMSprop	50	0.83	0.98	0.75	0.65	0.85	0.64	0.82
U-Net	ADAM	100	0.78	0.98	0.71	0.56	0.82	0.55	0.77
U-Net	AdaGrad	100	0.80	0.98	0.72	0.59	0.83	0.59	0.79
U-Net	SGD	100	0.68	0.98	0.62	0.36	0.76	0.35	0.67
U-Net	NADAM	100	0.80	0.98	0.73	0.61	0.84	0.60	0.79
U-Net	RMSprop	100	0.75	0.98	0.68	0.50	0.80	0.49	0.74
ResNet-50	ADAM	50	0.67	0.98	0.61	0.33	0.75	0.32	0.66
ResNet-50	AdaGrad	50	0.60	0.98	0.57	0.33	0.72	0.18	0.58
ResNet-50	SGD	50	0.50	0.98	0.60	0.33	0.66	0.12	0.52
ResNet-50	NADAM	50	0.61	0.98	0.57	0.20	0.72	0.19	0.59
ResNet-50	RMSprop	50	0.69	0.95	0.64	0.42	0.76	0.38	0.68
ResNet-50	ADAM	100	0.68	0.96	0.63	0.37	0.76	0.35	0.67
ResNet-50	AdaGrad	100	0.61	0.98	0.57	0.33	0.72	0.20	0.60
ResNet-50	SGD	100	0.52	0.99	0.52	0.35	0.68	0.15	0.50
ResNet-50	NADAM	100	0.69	0.95	0.63	0.41	0.76	0.37	0.68
ResNet-50	RMSprop	100	0.64	0.98	0.59	0.26	0.74	0.26	0.62

When the Table 1 is examined, DenseNet-169 have the highest accuracy rate with 96%. For this rate, the optimizer is AdaGrad and the number of iterations is 50. When the maximum values produced by other architectures are examined, NADAM was used in 8 experiments, RMSprop in 4 experiments and ADAM optimizer in 3 experiments. 7 of these results were obtained from 50 iterations and 8 of them were obtained from 100 iterations. ResNet-50 yielded the lowest accuracy value of 50% with SGD optimizer at 50 iterations. In addition, the highest accuracy rate of 69% of ResNet-50 is considerably lower than other architectures. DenseNet's are very successful with results of 94% and above.

When the AUC values are examined, with a value of 0.95, DenseNet-169 have the best result in 50 iterations and AdaGrad optimizer. Considering the best values of other architectures, NADAM was used in 6 results, ADAM and RMSprop optimizer was used in 3 results. 5 of them belong to 50 iterations and 7 of them to 100 iterations. ResNet-50 gave the worst AUC value of 0.5 at 100 iterations with SGD optimizer. The AUC value of 0.68 of ResNet-50 is lower than the best results yielded by other architectures. According to all the AUC values, DenseNets are more efficient.

When the results obtained with the kappa evaluation method are analyzed, it is seen that DenseNet-169 is the most successful architecture with AdaGrad optimizer and 50 iterations. This kappa value is 0.91. ResNet-50 is the lowest rate network

with 0.12. At this value, SGD was used as optimizer and the number of iterations was 50.

In precision values, the best result belongs to DenseNet-121 with 0.95. This value was obtained by using 100 iterations and ADAM optimizer. The worst result is 0.20 and it was produced by ResNet-50. It is seen that NADAM was used as optimizer and the number of iterations was 50. When all precision values are examined, it is seen that NADAM was used as optimizer and the number of iterations was 50 in most of the best results produced by the models. Again, it is clear that DenseNets are more successful in these results.

According to the specificity results, DenseNet-121 is the best architecture, while ResNet-50 is the most unsuccessful network. DenseNet-121 yielded 0.96 in 100 iterations with ADAM optimizer. In the worst result, SGD optimizer was used and the number of iterations was 100. It is seen that NADAM was used as optimizer and the number of iterations was 50 in most of the best values that the architectures had.

The most effective architectures for sensitivity values are DenseNets, U-Net and ResNet-50 with the value of 0.99. DenseNet-121 gave this value as a result of the experiments using AdaGrad and SGD optimizer in 50 and 100 iterations, respectively. DenseNet-201 yielded 0.99 sensitivity value with AdaGrad optimizer at 50 and 100 iterations and with SGD optimizer at 50 iterations. In DenseNet-169, RMSprop was used in 50 iterations and SGD optimizer in 100 iterations. SGD optimizer was used in ResNet-50 and U-Net networks and the iteration numbers were 100 and 50 respectively.

Finally, considering the F1 scores, the best result is 0.96, the worst result is 0.66. In the 0.96 value produced by DenseNet-169 architecture, AdaGrad optimizer was used and the number of iterations was 50. 0.66 belongs to ResNet-50 network with SGD optimizer 50 iterations. It is seen that DenseNet models are superior in F1 score values too.

When the evaluation is made considering all the results in the table, it is seen that DenseNets are the most successful architecture while ResNet-50 is a less effective network. Although AdaGrad was the optimizer for the majority of the highest results, the NADAM optimizer in general also produced effective results. The success of 50 iterations shows that effective results can be obtained with a small number of iterations. Figure 2 and 3 show Grad-CAM [41] visualization of 4 and 5 images, respectively, classified as COVID-19 using DenseNet169 50 iterations and AdaGrad optimizer. Figure 4 shows the normalized confusion matrix and ROC curve of DenseNet169 with 50 iterations and AdaGrad optimizer.

3.3. Statistical Significance in Each Performance Evaluation. In this study, each architecture was evaluated by the well-known performance measures. These performance measures were calculated for five optimizers with 50 and 100 iterations. One of the important concerns is whether there were statistically significant

TABLE 2. Comparisons of the optimizers by the architecture-performances ($iter = 50$).

Performance Score (PS)	Assumption		Comparing the optimizers	
	Normality	Variance-homogeneity	Hypothesis testing ($p - value$)	Pairwise comparisons
PS^1 : Acc-score	✗ ($p - value < 0.005$)	✓ ($p - value = 0.508$)	No difference/Similar effects ($p - value = 0.052$)	—
PS^2 : Sens-score	✗ ($p - values < 0.005$)	✓ ($p - value = 0.649$)	No difference/Similar effects ($p - value = 0.545$)	—
PS^3 : Spec-score	✓ ($p - values > 0.01$)	✓ ($p - value = 0.890$)	Significant difference/Dissimilar effects ($p - value = 0.011$)	Significant differences: •NADAM-AdaGrad (95%CI = (0.64, 17.86), $p - value = 0.031$) •SGD-NADAM (95%CI = (-18.48, -1.27), $p - value = 0.018$)
PS^4 : Prec-score	✗ ($p - value < 0.005$)	✓ ($p - value = 0.873$)	Significant difference/Dissimilar effects ($p - value = 0.030$)	Significant differences: •ADAM-SGD (95%CI = (5.0, 22.5), $p - value = 0.022$) •NADAM-SGD (95%CI = (1.5, 30.0), $p - value = 0.049$)
PS^5 : F1-score	✓ ($p - values > 0.01$)	✓ ($p - value = 0.636$)	Significant difference/Dissimilar effects ($p - value = 0.014$)	Significant difference: •SGD-NADAM (95%CI = (-11.49, -0.26), $p - value = 0.037$)
PS^6 : Kappa-score	✗ ($p - value < 0.005$)	✓ ($p - value = 0.350$)	No difference/Similar effects ($p - value = 0.052$)	—
PS^7 : AUC-score	✗ ($p - value < 0.005$)	✓ ($p - value = 0.449$)	Significant difference/Dissimilar effects ($p - value = 0.015$)	Significant differences: •AdaGrad-ADAM (95%CI = (-10.0, -2.0), $p - value = 0.030$) •ADAM-SGD (95%CI = (5.0, 13.5), $p - value = 0.014$) •NADAM-SGD (95%CI = (2.0, 17.0), $p - value = 0.035$)

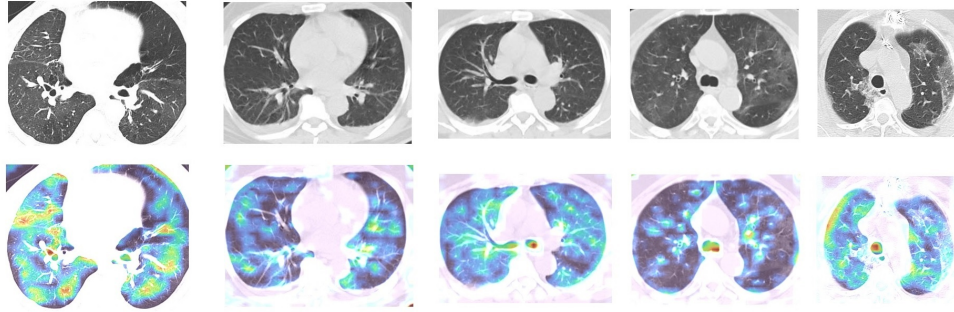


FIGURE 2. Grad-CAM [41] visualization. First row is original images with COVID-19; second row is Grad-CAM visualizations.

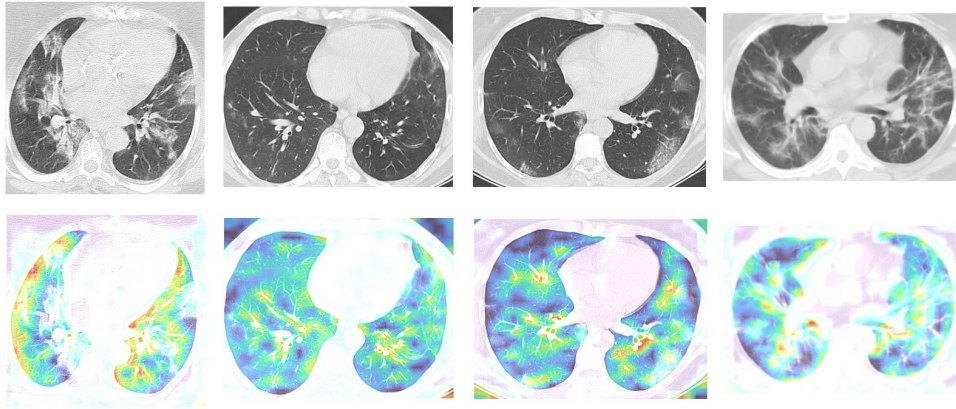


FIGURE 3. Grad-CAM [41] visualization: First row is original images with COVID-19; second row is Grad-CAM visualizations.

differences in each performance measure/score of the architectures by the optimizers. Performance-based data was organized and prepared as in Figure 5. Normality test was performed by using Shapiro-Wilk test, where α -level equals 0.01. Variance-homogeneity was investigated using Bartlett test for normally-distributed data and Levene test for non-normally distributed data. Comparisons of k -paired samples ($k > 2$) were analyzed by using two-way variance analysis where normality and variance-homogeneity assumptions were provided. Turkey test was conducted to perform pairwise-comparisons. Besides, Friedman test was used when normality and/or variance-homogeneity assumptions were not provided. Besides of comparing grand and estimated medians, pairwise-comparisons of k -paired samples were

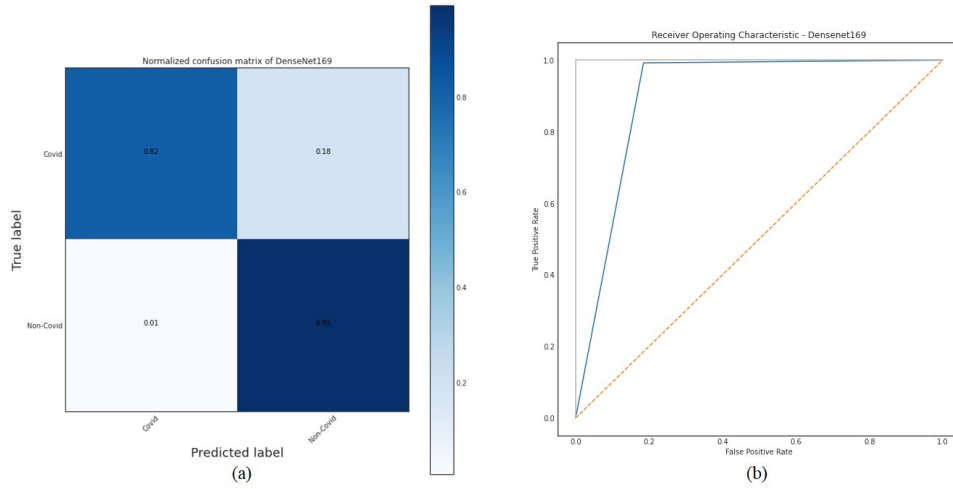


FIGURE 4. Normalized confusion matrix and ROC curve: The left image is normalized confusion matrix; the right one is ROC curve

performed by using Wilcoxon sign-rank tests. Significance level was selected as 0.05 to compare the optimizers. The lists of assumption-results and statistical inferences were summarized as in Table 2 and Table 3, where iteration number equals 50 and 100 respectively.

TABLE 3. Comparisons of the optimizers by the architecture-performances ($iter = 100$).

Performance Score (PS)	Assumption		Comparison of k-paired samples ($p - value$)	Pairwise-comparisons
	Normality	Variance-homogeneity		
PS^1 : Acc-score	✓ ($p - values > 0.01$)	✓ ($p - value = 0.655$)	Significant difference/Dissimilar effects ($p - value = 0.011$)	Significant differences: <ul style="list-style-type: none"> •SGD-ADAM (95%CI = (-12.57, -0.18), $p - value = 0.042$) •SGD-NADAM (95%CI = (-13.32, -0.93), $p - value = 0.018$)
PS^2 : Sens-score	✗ ($p - values < 0.005$)	✓ ($p - value = 0.565$)	No difference/Similar effects ($p - value = 0.636$)	—

PS^3 : Spec-score	✓ (p - values > 0.01)	✓ (p - value = 0.988)	Significant difference/Dissimilar effects (p - value = 0.000)	Significant differences: <ul style="list-style-type: none"> •ADAM-AdaGrad (95%CI = (1.81, 10.69), p - value = 0.003) •NADAM-AdaGrad: (95%CI = (2.43, 11.32), p - value = 0.001) •SGD-ADAM: (95%CI = (-13.57, -4.68), p - value = 0.000) •SGD-NADAM: (95%CI = (-14.19, -5.31), p - value = 0.000) •SGD-RMSprob: (95%CI = (-10.94, -2.06), p - value = 0.002)
PS^4 : Prec-score	✓ (p - values > 0.01)	✓ (p - value = 0.964)	Significant difference/Dissimilar effects (p - value = 0.000)	Significant differences: <ul style="list-style-type: none"> •NADAM-AdaGrad: (95%CI = (1.29, 16.46), p - value = 0.016) •SGD-ADAM: (95%CI = (-20.08, -4.92), p - value = 0.000) •SGD-NADAM: (95%CI = (-21.96, -6.79), p - value = 0.000) •SGD-RMSprob: (95%CI = (-15.46, -0.29), p - value = 0.039)
PS^5 : F1-score	✓ (p - values > 0.01)	✓ (p - value = 0.814)	Significant difference/Dissimilar effects (p - value = 0.005)	Significant difference: <ul style="list-style-type: none"> •SGD-ADAM (95%CI = (-10.14, -0.36), p - value = 0.031) •SGD-NADAM (95%CI = (-10.64, -0.86), p - value = 0.015)

PS^6 : Kappa-score	✓ (p -values > 0.01)	✓ (p -value ₄ = 0.898)	Significant difference/Dissimilar effects (p -value = 0.001)	Significant difference: <ul style="list-style-type: none"> ●NADAM-AdaGrad (95%CI = (0.85, 21.65), p-value = 0.029) ●SGD-ADAM (95%CI = (-24.03, -3.22), p-value = 0.006) ●SGD-NADAM (95%CI = (-25.28, -4.47), p-value = 0.002)
PS^7 : AUC-score	✓ (p -values > 0.01)	✓ (p -value = 0.745)	Significant difference/Dissimilar effects (p -value = 0.001)	<ul style="list-style-type: none"> ●SGD-ADAM (95%CI = (-13.30, -2.20), p-value = 0.003) ●SGD-NADAM (95%CI = (-14.05, -2.95), p-value = 0.001) ●SGD-RMSprob (95%CI = (-11.55, -0.45), p-value = 0.029)

4. CONCLUSION

In this study, the success of optimizers in diagnosing disease from COVID-19 CT images using different optimizers in different architectures was examined. In addition, the number of iterations was set at two different values, 50 and 100. DenseNet-169, DenseNet-121, DenseNet-201, MobileNetV2, U-Net, ResNet-50, VGG16 and VGG19 were used as models. The efficiency of ADAM, AdaGrad, SGD, NADAM and RMSprop optimizers was observed. Accuracy, sensitivity, specificity, precision, F1 score, kappa and AUC were used as evaluation metrics. According to the results, DenseNets were quite successful, while ResNet-50 was the less effective architecture. While NADAM is the superior optimizer for the majority of architectures' own best results, the majority of the top values in evaluation metrics include AdaGrad optimizer. Considering that the number of images in the data set used in the study is insufficient, it should be noted that the models yield very good results.

The differences in architecture-performances can be effected by the selected optimizers. Thus, the optimizer-effects were analyzed for each performance metric of the architectures. As the results of statistical inferences, there were statistically significant differences in 4 out of 7 architecture-performance metrics and 6 out of 7 architecture-performance metrics by the optimizers, when iteration-numbers were 50 and 100. According to pairwise comparisons, it has been seen that these

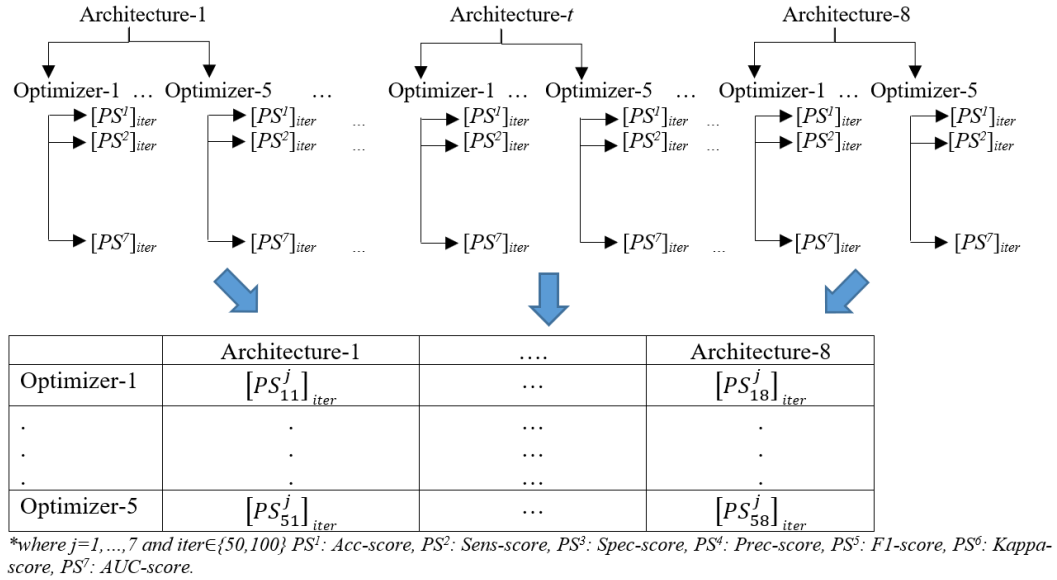


FIGURE 5. Data-preparation for statistical inference.

differences were mostly occurred by NADAM-optimizer. Compared to the performances of AdaGrad, NADAM has the best specification-performances for both 50 (p -value = 0.005) and 100 (p -value = 0.000) iterations. Besides, NADAM has better precision- (p -value = 0.000) and kappa- (p -value = 0.014) performances than AdaGrad, when iteration-number was 100.

Author Contribution Statements The authors contributed equally to this work.

Declaration of Competing Interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

- [1] Velavan, T. P., Meyer, C. G., The COVID-19 epidemic, *TM & IH*, 25 (3) (2020), 278, <https://doi.org/10.1111/tmi.13383>.
- [2] Minaee, S., Kafieh, R., Sonka, M., Yazdani, S., Soufi, G. J., Deep-covid: Predicting covid-19 from chest X-ray images using deep transfer learning, *Med. Image Anal.*, 65 (2020), 101794, <https://doi.org/10.1016/j.media.2020.101794>.
- [3] Amyar, A., Modzelewski, R., Li, H., Ruan, S., Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: Classification and segmentation, *Comput. Biol. Med.*, 126 (2020), 104037, <https://doi.org/10.1016/j.combiomed.2020.104037>.

- [4] Ai, T., Yang, Z., Hou, H., Zhan, C., Chen, C., Lv, W., Tao, O., Sun, Z., Xia, L. Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases, *Radiology*, 296 (2020), E32-E40, <https://doi.org/10.1148/radiol.2020200642>.
- [5] Islam, M. M., Karray, F., Alhajj, R., Zeng, J., A review on deep learning techniques for the diagnosis of novel coronavirus (covid-19), *IEEE Access*, 9 (2021), 30551-30572, <https://doi.org/10.1109/ACCESS.2021.3058537>.
- [6] Xu, X., Jiang, X., Ma, C., Du, P., Li, X., Lv, S., Yu, L., Ni, O., Chen, Y., Su, J., et al., A deep learning system to screen novel coronavirus disease 2019 pneumonia, *Engineering*, 6 (10) (2020), 1122-1129, <https://doi.org/10.1016/j.eng.2020.04.010>.
- [7] Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., Cai, M., Yang, J., Li, Y., Meng, X., et al., A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19), *Eur. Radiol.*, (2021), 1-9, <https://doi.org/10.1007/s00330-021-07715-1>.
- [8] Kanne, J. P., Chest CT findings in 2019 novel coronavirus (2019-nCoV) infections from Wuhan, China: key points for the radiologist, *Radiological Society of North America*, (2020), <https://doi.org/10.1148/radiol.2020200241>.
- [9] Rubin, G. D., Ryerson, C. J., Haramati, L. B., Sverzellati, N., Kanne, J. P., Raouf, S., Schluger, N. W., Volpi, A., Yim, J. J., Martin, I. B. K., et al., The role of chest imaging in patient management during the COVID-19 pandemic: a multinational consensus statement from the Fleischner Society, *Radiology*, 296 (11) (2020), 172-180, <https://doi.org/10.1016/j.chest.2020.04.003>.
- [10] Kanne, J. P., Little, B. P., Chung, J. H., Elicker, B. M., Ketai, L. H., Essentials for radiologists on COVID-19: an update—radiology scientific expert panel, *Radiological Society of North America*, (2020), <https://doi.org/10.1148/radiol.2020200527>.
- [11] Bhattacharya, S., Maddikunta, P. K. R., Pham, Q. V., Gadekallu, T. R., Chowdhary, C. L., Alazab, M., Piran Md. J., et al., Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey, *Sustain. Cities Soc.*, 65 (2021), 102589, <https://doi.org/10.1016/j.scs.2020.102589>.
- [12] Shan, F., Gao, Y., Wang, J., Shi, W., Shi, N., Han, M., Xue, Z., Shen, D., Shi, Y., Lung infection quantification of COVID-19 in CT images with deep learning, *arXiv preprint arXiv:2003.04655*, (2020), <https://doi.org/10.1002/mp.14609>.
- [13] Huang, L., Han, R., Ai, T., Yu, P., Kang, H., Tao, Q., Xia, L., Serial quantitative chest CT assessment of COVID-19: a deep learning approach, *Radiology: Cardiothoracic Imaging*, 2 (2) (2020), e200075, <https://doi.org/10.1148/ryct.2020200075>.
- [14] Wu, X., Hui, H., Niu, M., Li, L., Wang, L., He, B., Yang, X., Li, L., Li, H., Tian, J., and others, Deep learning-based multi-view fusion model for screening 2019 novel coronavirus pneumonia: A multicentre study, *Eur. J. Radiol.*, 128 (2020), 109041, <https://doi.org/10.1016/j.ejrad.2020.109041>.
- [15] Ardakani, A. A., Kanafi, A. R., Acharya, U. R., Khadem, N., Mohammadi, A., Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks, *Comput. Biol. Med.*, 121 (2020), 103795, <https://doi.org/10.1016/j.compbimed.2020.103795>.
- [16] Apostolopoulos, I. D., Mpesiana, T. A., Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks, *Phys. Eng. Sci. Med.*, 43 (2) (2020), 635-640, <https://doi.org/10.1007/s13246-020-00865-4>.
- [17] Singh, D., Kumar, V., Kaur, M., and others, Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks, *Eur. J. Clin. Microbiol. Infect. Dis.*, 39 (7) (2020), 1379-1389, <https://doi.org/10.1007/s10096-020-03901-z>.
- [18] Khan, A. I., Shah, J. L., Bhat, M. M., CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest X-ray images, *Comput. Meth. Prog. Bio.*, 196 (2020), 105581, <https://doi.org/10.1016/j.cmpb.2020.105581>.

- [19] Krizhevsky, A., Sutskever, I., Hinton, G. E., Imagenet classification with deep convolutional neural networks, *Adv Neural Inf Process Syst*, 25 (2012), 1097-1105, <https://doi.org/10.1145/3065386>.
- [20] Simonyan, K., Zisserman, A., Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*, (2014), <https://doi.org/10.48550/arXiv.1409.1556>.
- [21] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., Keutzer, K., SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size, *arXiv preprint arXiv:1602.07360*, (2016), <https://doi.org/10.48550/arXiv.1602.07360>.
- [22] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., Going deeper with convolutions, *CVPR*, (2015), 1-9, <https://doi.org/10.48550/arXiv.1409.4842>.
- [23] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C., Mobilenetv2: Inverted residuals and linear bottlenecks, *CVPR*, (2018), 4510-4520, <https://doi.org/10.48550/arXiv.1801.04381>.
- [24] He, K., Zhang, X., Ren, S., Sun, J., Deep residual learning for image recognition, *CVPR*, (2016), 770-778, <https://doi.org/10.48550/arXiv.1512.03385>.
- [25] Chollet, F., Xception: Deep learning with depthwise separable convolutions, *CVPR*, (2017), 1251-1258, <https://doi.org/10.48550/arXiv.1610.02357>.
- [26] Jaiswal, A., Gianchandani, N., Singh D., Kumar, V., Kaur, M., Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning, *J. Biomol. Struct. Dyn.*, (2017), 4700-4708, <https://doi.org/10.1080/07391102.2020.1788642>.
- [27] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q., Densely connected convolutional networks, *CVPR*, (2020), 1-8, <https://doi.org/10.48550/arXiv.1608.06993>.
- [28] Wang, X., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., Liu, W., Zheng, C., A weakly-supervised framework for COVID-19 classification and lesion localization from chest CT, *IEEE Trans. Med. Imaging.*, 39 (8) (2020), 2615-2625, <https://doi.org/10.1109/TMI.2020.2995965>.
- [29] Ronneberger, O., Fischer, P., Brox, T., U-net: Convolutional networks for biomedical image segmentation, *MICCAI*, (2015), 234-241, <https://doi.org/10.48550/arXiv.1505.04597>.
- [30] Kingma, D. P., Ba, J., Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*, (2014), <https://doi.org/10.48550/arXiv.1412.6980>.
- [31] Chen, J., Wu, L., Zhang, J., Zhang, L., Gong, D., Zhao, Y., Chen, Q., Huang, S., Yang, M., Yang, X., et al., Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography, *Sci. Rep.*, 10 (1) (2020), 1-11, <https://doi.org/10.1038/s41598-020-76282-0>.
- [32] Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., Liang, J., Unet++: A nested u-net architecture for medical image segmentation, *DLMI and ML-CDS*, (2018), 3-11, <https://doi.org/10.48550/arXiv.1807.10165>.
- [33] Song, Y., Zheng, S., Li, L., Zhang, X., Zhang, X., Huang, Z., Chen, J., Wang, R., Zhao, H., Zha, Y., et al., Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images, *TCBB*, (2021), <https://doi.org/10.1109/TCBB.2021.3065361>.
- [34] Gozes, O., Frid-Adar, M., Greenspan, H., Browning P. D., Zhang, H., Ji, W., Bernheim, A., Siegel, E., Rapid ai development cycle for the coronavirus (covid-19) pandemic: Initial results for automated detection & patient monitoring using deep learning ct image analysis, *arXiv preprint arXiv:2003.05037*, (2020), <https://doi.org/10.48550/arXiv.2003.05037>.
- [35] He, X., Yang, X., Zhang, S., Zhao, J., Zhang, Y., Xing, E., Xie, P., Sample-efficient deep learning for COVID-19 diagnosis based on CT scans, *medrxiv*, (2020), <https://doi.org/10.1101/2020.04.13.20063941>.
- [36] Soares, E., Angelov, P., Biaso, S., Froes, M. H., Abe, D. K., SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification, *medRxiv*, (2020) <https://doi.org/10.1101/2020.04.24.20078584>.

- [37] Duchi, J., Hazan, E., Singer, Y., Adaptive subgradient methods for online learning and stochastic optimization, *JMLR*, 12 (7) (2011).
- [38] Dozat, T., Incorporating nesterov momentum into adam, *ICLR*, (2016), 1-4.
- [39] Rude, S., An overview of gradient descent optimization algorithms, *arXiv preprint arXiv:1609.04747*, (2016), <https://doi.org/10.48550/arXiv.1609.04747>.
- [40] Nair, V., Hinton, G. E., Rectified linear units improve restricted boltzmann machines, *ICML*, (2010), 807-814.
- [41] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., Grad-cam: Visual explanations from deep networks via gradient-based localization, *Proc. IEEE Int. Conf. Comput. Vis.*, (2017), 618-626 <https://doi.org/10.48550/arXiv.1610.02391>.

COMPARATIVE ANALYSIS OF MATURE TOMATO DETECTION BY FEATURE EXTRACTION AND MACHINE LEARNING FOR AUTONOMOUS GREENHOUSE ROBOTS

Hakkı Alparslan İLGİN¹, Fevzi Anıl AYDEMİR¹, Berkay CEDİMOĞLU¹,
Muhammet Nurullah AYDIN¹, Hasan SİLLELİ²

¹Department of Electrical and Electronics Engineering, Ankara University,
Ankara, TÜRKİYE






²Department of Agricultural Machinery and Technologies Engineering,
Ankara University, Ankara, TÜRKİYE

ABSTRACT. Accurate detection of tomatoes grown in greenhouses is important for timely harvesting. In this way, it is ensured that mature tomatoes are collected by distinguishing them from the unripe ones. Insufficient light, occlusion, and overlapping adversely affect the detection of mature tomatoes. In addition, it is time consuming for people to detect mature tomatoes at certain periods in large greenhouses. For these reasons, high-performance automatic detection of tomatoes by greenhouse robots has become an increasingly studied area today. In this paper, two feature extraction methods, histogram of oriented gradients (HOG) and local binary patterns (LBP), which are effective in object recognition, and two important and commonly used classifiers of machine learning, support vector machines (SVM) and k-nearest neighbor (kNN), are comparatively used to detect and count tomatoes. The HOG and LBP features are classified separately and together by SVM or kNN, and the success of each case are compared. Performance of the detection is improved by eliminating false positive results at the postprocessing stage using color information.

1. INTRODUCTION

In recent years, studies in the field of agriculture which are supported by artificial intelligence have increased. Therefore, the development of fruit harvesting robots that fulfill this purpose has accelerated. These robots can detect and recognize a fruit

Keywords. Tomato detection, harvesting robots, machine learning, smart agriculture.

✉ ilgin@ankara.edu.tr – Corresponding author;  0000-0003-0112-4833
✉ anilaydemir9989@gmail.com;  0000-0002-6530-3040
✉ berkaycedimoglu@gmail.com;  0000-0002-2179-9566
✉ naydinn6@gmail.com;  0000-0002-4026-9739
✉ Hasan.Silleli@agri.ankara.edu.tr;  0000-0003-2242-3402.

© 2023 Ankara University
Communications Faculty of Sciences University of Ankara Series A2-A3: Physical Sciences and Engineering

autonomously with computer vision. On the other hand, developing an artificial visual system as successful as human perception is not an easy task [1, 2].

There are many studies that have been carried out for the visual system of harvesting robots and fruit detection. In [1], mature tomatoes were detected by using color and HOG features. Arivazhagan et al. [3] proposed a method for fruit detection using hue, saturation, value (HSV) color space and texture features. Bulanon et al. [4] developed a method for apple recognition using luminance and red, green and blue color difference. Liu et al. [5] used a simple linear iterative clustering method to segment apple images followed by color feature extraction to detect apples. Drg-Drb color index was utilized to segment apples from their surroundings in [6]. There are also various studies that use techniques other than color space models. Tanigaki et al. [7] used a three-dimensional (3D) vision sensor which is equipped with red and infrared lasers to locate and recognize the fruits and obstacles, prevented by sunlight. Ji et al. [8] utilized a support vector machine (SVM) classifier, for apple recognition. Circular Gabor texture analysis for feature extraction, and neural network with SVM classifier were used in [9] to detect peach fruit. Song et al. [10] proposed bag-of-words (BoW) model and novel statistical approach for locating and counting the pepper fruits. Histogram of oriented gradients (HOG) and local binary patterns (LBP) were used in [11] to detect plants.

In this paper, tomato detection is mainly carried out in three steps which are brightness and contrast enhancement as preprocessing, classification of features, and elimination of false positives as postprocessing. HOG and LBP features were classified separately and together with SVM or kNN, and the results of feature classification combinations were compared. Thus, six different cases for tomato detection were studied. Images are first preprocessed by means of illumination and contrast enhancement to improve the results in terms of detection success under different conditions. Naïve Bayes classifier (NBC) is then used to classify the pixels of the image as black and white, which represent possible background and tomato pixels, respectively. Morphological operations are applied to the black and white image to combine neighboring big groups of possible tomato pixels. Vertical and horizontal limits of those pixels give the coordinates of the region of interest (ROI) on the enhanced image, where the search for tomatoes is carried out with 64×64 sliding windows at a certain step size. LBP and HOG features of sliding windows in ROI are extracted to be used separately and together in detection process via SVM or kNN. At the postprocessing stage, false color removal (FCR) method is utilized to eliminate false positive results. Lastly, non-maximum suppression (NMS) is used to merge overlapped detections of the same tomato.

In the next section, preprocessing and obtaining the ROI is explained. Feature extraction and classification methods are presented in section 3. Postprocessing with

FCR and non-maximum suppression is discussed in section 4. Experimental results and conclusion are given in last two sections.

2. PREPROCESSING AND ROI

At the first stage of the preprocessing, contrast and lighting enhancement is applied to reduce the negative effect of illumination and contrast conditions such as light fluctuation and low contrast which may reduce the success of the detection process. There are various methodologies for contrast enhancement such as general histogram equalization (GHE), local histogram equalization (LHE), singular value equalization (SVE), dynamic histogram equalization (DHE) and contrast limited adaptive histogram equalization (CLAHE) [12]. CLAHE is frequently used in image enhancement applications where contrast and illumination are of prime importance [13, 14]. In this paper, before the training and detection processes, illumination and contrast enhancement algorithms were performed for all images. Since hue, saturation, intensity (HSI) is the one of the most suitable color spaces for these processes, original images in red, green, blue (RGB) color space are converted to HSI color space. Hue (H), saturation (S) and intensity (I) components of HSI color space are obtained as follows, respectively:

$$H = \begin{cases} \theta, & B \leq G \\ 360 - \theta, & B > G \end{cases} \quad (1)$$

where θ is angle in degrees and is given as

$$\theta = \cos^{-1} \left(\frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right) \quad (2)$$

$$S = 1 - \frac{3}{R + G + B} [\min(R, G, B)] \quad (3)$$

$$I = \frac{R + G + B}{3} \quad (4)$$

where R , G and B are red, green and blue components of the original RGB color space. The illuminance is enhanced by applying the natural logarithm to the intensity component I [1]. CLAHE is applied to the modified intensity component for contrast enhancement. Since CLAHE enhances the contrast by correlating the intensities of neighboring pixels, less noise is seen in the output image compared to AHE. In this way, brightness bursts in the image are prevented [15-17]. After illumination and

contrast enhancement, the image is converted from the HSI color space back to the RGB color space by the equations given below:

$$G = I(1 - S) \quad (5)$$

$$B = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad (6)$$

$$R = 3I - (G + B) \quad (7)$$

An example for the enhancement by natural logarithm and CHALE is given in Fig. 1, where an original greenhouse video frame and illumination and contrast-enhanced video frame are shown, respectively.



FIGURE 1. Video frame color enhancement (a) Original frame (b) Enhanced frame.

In Fig. 1 (a) a dark video frame with low contrast captured in a greenhouse is shown. After contrast and illumination enhancement, it becomes brighter with enhanced contrast for each color component as shown Fig. 2. Red, green and blue color histograms of the original and enhanced frames are shown in Fig. 2 (a), (b) and (c), respectively.

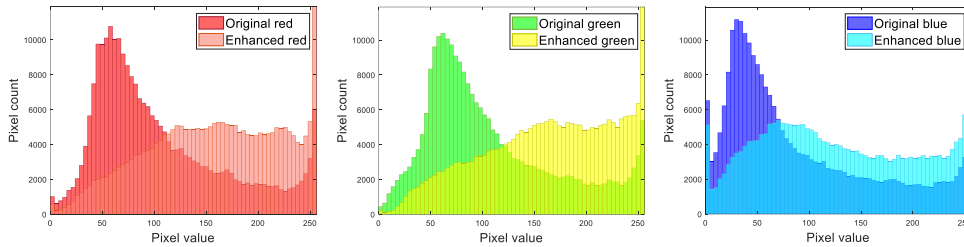


FIGURE 2. Color histograms of the original and enhanced frames in Fig. 1.

As shown in the histograms in Fig. 2, colors are distributed more evenly in the enhanced image than in the original one. Therefore, enhanced image has better contrast.

After illumination and contrast enhancement, ROI, which is the scanning region is obtained using NBC and morphological operations, which is explained in the next subsection.

2.1. Region of Interest (ROI). Images are cropped before they are scanned for tomatoes. Cropped areas are called ROI, which has been investigated in studies such as [1, 18, 19]. ROI contributes to the extraction of a functional feature by reducing the number of background pixels in the image and positively affects the success of object detection and computational complexity. The most critical point of ROI acquisition is the extraction of color features. These features are used in the training of the NBC model and also in the classification of the pixels to determine the pixel labels as either tomato or background in the image during the pixel detection process to acquire ROI. There are three color features related to red color components used in NBC [1], which are given as

$$c_1 = R - G \quad (8)$$

$$c_2 = R - B \quad (9)$$

$$c_3 = \frac{R}{R+G+B} \quad (10)$$

The first two features show how high or low red is relative to green and blue, respectively. The closer the third feature is to one, the more dominant the red color is compared to the other colors. After color features are calculated for training, threshold value for all three is obtained empirically. In Fig. 3, sorted color features are shown. As can be seen in the graphs, drastic change of the slopes of the curves begins around the sixty thousandth value for each of the three ordered color features and increases to higher values than its normal course. Therefore, an empirical threshold determination was performed by taking into account the slope change points for each feature, and accordingly, the pixels above the thresholds were labeled as tomatoes, and the rest as background.

After training NBC, a binary image where white and black pixels represent tomato and background regions, respectively, are obtained by classifying the pixels of the enhanced RGB image. Then some morphological operations such as erosion and dilation are applied to the binary image to obtain ROI precisely [20]. Gaps

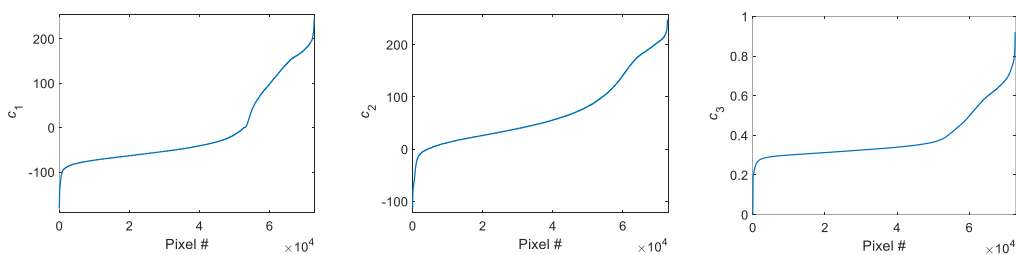
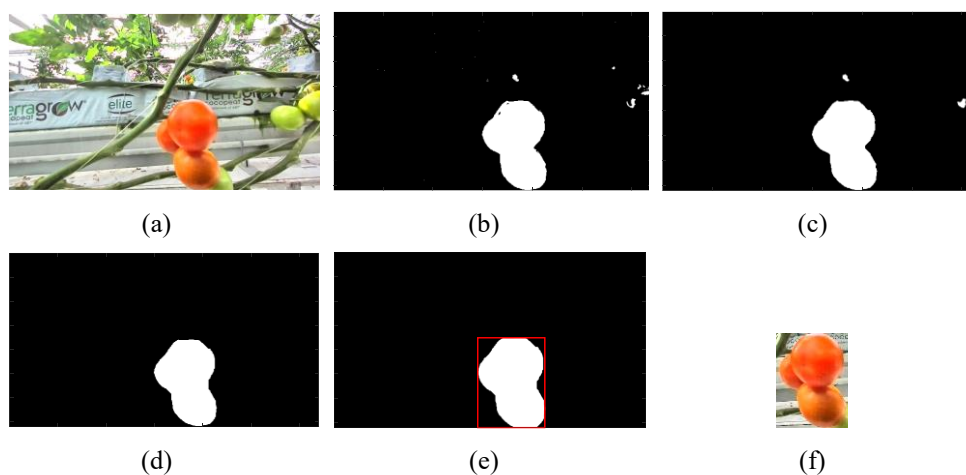
FIGURE 3. Sorted color features, c_1 , c_2 and c_3 .

FIGURE 4. The process of obtaining ROI (a) Enhanced image, (b) Binary image through NBC (c) Filling in large white pixel groups and removing small white pixel groups, (d) Removing white pixel groups other than the largest ones and filling the remaining white pixel groups with small discs, (e) Dilated white pixel group with red borders; (f) Cropped image (ROI).

between and within large groups of white pixels are filled, while small white pixel groups are removed through morphological operations. Finally, the binary image is cropped using the vertical and horizontal endpoints of white pixels, giving ROI. Binary image after pixel classification by NBC and a series of morphological operations to obtain ROI are shown in Fig. 4.

As seen in Fig. 4, NBC classifies pixels as white for tomato and black for background. Then, morphological operations give endpoints to crop the enhanced image. After cropping operation, ROI, which is the search area is obtained. Since it

is mostly smaller than the enhanced image, it reduces the processing time and increases the detection success.

In the next stage of the algorithm, tomatoes are searched in the ROI. For this purpose, a 64×64 sliding window is shifted in the search region with a certain vertical and horizontal step size. HOG and/or LBP features of each sliding window in the ROI are extracted. Then, the features of the sliding windows are classified using SVM or kNN as tomato or background. After the search process is finished, the ROI is downscaled by 10% and the same operations continue for 64×64 sliding window which is shifted by the same step size in the downscaled ROI. This is to ensure that large tomatoes that could not be detected in the previous scale can be detected in the downscaled ROIs, as tomatoes can be of different sizes. The downscaling process of ROI is repeated until its size is close to the size of the sliding window.

Feature extraction and classification are explained in detail in the next section.

3. FEATURE EXTRACTION AND CLASSIFICATION

For humans, detection and classification of objects are simple and effortless acts. However, it is difficult and complicated task for machines and robots to detect an object. In order to overcome this difficulty, classifier algorithm must use features, such as HOG, LBP, Bag of Visual Words (BoVW) and scale-invariant feature transform (SIFT), of the object to be detected. HOG feature extraction method was firstly used in pedestrian detection by Dalal and Triggs [21], and then has been widely used in object detection. HOG is a successful descriptor especially when used with SVM. LBP is also a widely used robust method for describing the texture properties of objects [22, 23]. In the LBP algorithm, the pixels of an image are labeled with decimal numbers called LBP codes, which encode the local structure around each pixel. The values obtained by subtracting each pixel from the eight pixels in its neighborhood are compared with a threshold value, and 1 and 0 are assigned for values greater and less than the threshold, respectively. The opposite is also possible. After the matrix containing 0's and 1's is obtained, the binary bit sequence is created by moving clockwise with the first digit in the upper left. The decimal equivalent of this bit sequence corresponds to the LBP value of the center pixel.

In this paper, besides using the HOG and LBP features separately, we use them together to increase the classification success. HOG features were extracted with 4×8 pixel cells, 2×2 cell blocks and 10 orientation bins. Using the labeled training data, SVM outputs an optimal hyperplane which categorizes the samples [24]. The results of SVM classification are compared with the results of kNN, which is also widely used in vision systems [25]. Generally, it is applied in pattern recognition and

data mining for classification thanks to its easy use and low error rate. It takes less effort to implement according to other classification techniques, and classifies unknown or new data by calculating the distance among existing and new data, and then checking the k -close neighbors, where k is number of neighbors. In our case, uniform distance was used, where all data points in each neighborhood were weighted equally. In addition, since our dataset are not very large, and SVM usually gives better results with fewer outliers than deep learning approach for relatively small data sets, deep learning models and methods are not used for classification.

3.1. SVM Training and Detection Processes. Before detection, SVM classifier is trained for HOG and LBP features separately. Features are extracted from a set of 64×64 pixel-resolution enhanced tomato and background training images to classify whether an image is tomato or not. These features are combined with corresponding labels to train SVM classifier. For classification, 64×64 images are also enhanced before extracting HOG and LBP features.

In the search area specified by the ROI, 64×64 sliding window is shifted with the fixed step size of 16 pixels in both vertical and horizontal directions. At each step, HOG and LBP features of the sliding window are extracted, and classified by SVM. When the sliding window is shifted all over the ROI, the ROI is downscaled with a scaling factor of 1.1. Then, tomatoes are searched in the downscaled ROI with the same processes with 64×64 sliding window. The final downscaled ROI is greater than or equal to 64×64 pixels in size.

4. POSTPROCESSING WITH FCR AND NMS

False and multiple positive detections are eliminated by FCR and NMS, respectively. FCR is used to remove false positive detections using color information [26]. During the search process in ROI, if sliding window is marked as tomato, then the result is checked by FCR whether it is false positive or not. If it is false positive, it is marked as background. For this aim, 64×64 -pixel sling window is binarized using the equation given below [1]:

$$I(x, y) = \begin{cases} 1, & 0.16R(x, y) - 0.093G(x, y) - 0.037B(x, y) - 11.032 \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where $I(x, y)$ is the binarized, namely black or white, pixel of the sliding window image with x and y vertical and horizontal pixel coordinates, respectively. If the equation is greater than or equal to zero, the pixel is classified as 1, which is tomato, otherwise it is classified as 0, which is background. After obtaining the binarized sliding window, the ratio of white pixels to the total number of pixels is calculated.

If it is less than a predetermined threshold, the label of the sliding window is switched to background from tomato [1]. Threshold value is determined empirically. All sliding windows in the original and all downscaled ROIs are classified as tomato or background by SVM or kNN classifier. After removing false positive detections by FCR, there may be more than one true positive results of the same tomato because of the sliding windows and downscaled ROIs. Therefore, NMS is applied as the last step of the process to eliminate positive over-detections [27]. NMS is based on the comparison of overlapping positive detections according to the classifier prediction score value and the selection of the detection with the highest score [28]. Confidence value and overlap threshold are important parameters used in NMS. The intersection over union (IOU) value of any of two sliding windows containing the detected tomatoes is calculated. After comparison with overlap threshold, over-detections of the same tomato are eliminated, leaving a single detection result. Thus, overlapping detections are filtered out, and the sliding window containing the highest score is selected as the only detection that includes the tomato. In this paper, best detection results were achieved when overlap threshold and confidence value are 0.27 and 0.7, respectively.

An example for elimination of unripe tomato detection by FCR and discarding over-detections by NMS is shown in Fig. 5. In both images, green bounding boxes are the results of tomato detections of sliding windows before FCR and NMS. In the left image, there are two tomato detections as the final result shown by red bounding boxes without applying FCR after NMS. As seen in this figure, when FCR is not used, unripe tomato is detected along with the mature one. However, FCR removes the detection of unripe tomato before NMS as shown in the image at the right.



FIGURE 5. Detection results with NMS and without FCR (left) and with FCR (right).

In the next section, test set, experimental results and comparison of the results are given.

5. EXPERIMENTAL RESULTS

In this paper, experiments were performed on NVIDIA Jetson AGX Xavier Developer Kit with Volta GPU w/512 CUDA Cores, 8-Core ARM v8.2 64 Bit CPU, 32 GB 256-Bit LPDDR4 RAM, and Ubuntu 18.04 operating system using Python 3.8 programming language. We used a public greenhouse tomato dataset from [29] with images sized 202×360 pixels in the experiments.

In the first stage of the experiments, tomatoes and background images were cropped in the size of 64×64 pixels from the data set for training. To expand the SVM training set, cropped images were rotated by 90, 180, and 270 degrees, and added to the training set beside other cropped images. Before the training, the cropped images were enhanced in terms of illumination and contrast. Then, HOG and LBP feature extraction were applied to the enhanced images. Finally HOG and LBP features were trained through SVM classification.

After training process, HOG and LBP features of the tomato and background test images were also extracted. Then test images were classified by SVM or kNN using only HOG, only LBP and HOG and LBP features together. To compare the performance of the feature extraction and classification algorithms, four metrics, recall, precision, F_1 and accuracy, were calculated from the confusion matrix. Classification of a 64×64 -sized image may result in four different detections, which are the elements of the confusion matrix as shown in Table 1.

TABLE 1. Confusion matrix.

Real Label	Background	TN	FP
	Tomato	FN	TP
		Background	Tomato
		Predicted Label	

True positive (TP) and true negative (TN) detections represent correct detections of tomato and background, respectively. On the other hand, false positive (FP) and false negative (FN) are undesirable results. FP is the detection of the background as

a tomato. Also, FN means that the detection result is the background, even though the actual image is a tomato.

In the first part of the experiments, confusion matrix results for classification of total of 450 images with the size of 64×64 , 150 of which are tomatoes and 300 of which are backgrounds, are given in Table 2. As seen in this table, the best result was obtained when HOG and LBP features were classified with SVM. In this case, all 300 backgrounds and 147 of 150 tomatoes were predicted correctly. Also, recall, precision, F_1 and accuracy metrics given in Table 3 were calculated from the confusion matrices in Table 2. The worst results were obtained when HOG features were classified by kNN, except the precision result of this approach is among the highest since there is no background misclassification.

TABLE 2. Confusion matrices of feature extraction and classification methods.

Feature	Classifier	
	SVM	kNN
HOG	$\begin{bmatrix} 300 & 0 \\ 5 & 145 \end{bmatrix}$	$\begin{bmatrix} 300 & 0 \\ 21 & 129 \end{bmatrix}$
LBP	$\begin{bmatrix} 298 & 2 \\ 7 & 143 \end{bmatrix}$	$\begin{bmatrix} 300 & 0 \\ 12 & 138 \end{bmatrix}$
HOG+LBP	$\begin{bmatrix} 300 & 0 \\ 3 & 147 \end{bmatrix}$	$\begin{bmatrix} 300 & 0 \\ 19 & 131 \end{bmatrix}$

TABLE 3. Metrics for detection results (%).

Feature	Classifier	Recall	Precision	F_1	Accuracy
HOG	SVM	96.67	100	98.31	98.89
LBP		95.33	98.62	96.95	98.00
HOG+LBP		98.00	100	98.99	99.33
HOG	kNN	86.00	100	92.47	95.33
LBP		92.00	100	95.83	97.33
HOG+LBP		87.33	100	93.24	95.78

In the second part of the experimental studies, tomatoes are searched in images with the size of 202×360 consisting total of 149 tomatoes. The search is carried out by means of 64×64 sliding windows on the ROI cropped from an 202×360 image from the test set. First, HOG and/or LBP feature extraction are performed for each sliding window which is shifted with a step size of 16 from left to right and top to bottom in the original resolution ROI. Then, the features of the sliding windows

in the ROI are classified by SVM or kNN whether they are tomatoes or not. ROI is then downscaled by the factor of 1.1, and the same operations for sliding windows in the reduced sized ROI are repeated. Processes are stopped when the ROI reaches its minimum size. Detection results for 75 images with 149 tomatoes are given in Table 4.

TABLE 4. Detection results for 75 images with 149 tomatoes.

Feature	Classifier	Number of Tomatoes	Tomato Detections	Missed Detections	Over-Detections
HOG	SVM	149	139	17	7
LBP		149	196	10	57
HOG+LBP		149	146	15	12
HOG	kNN	149	128	27	6
LBP		149	233	7	91
HOG+LBP		149	135	23	9

As seen in Table 4, best detection result in terms of missed detections is obtained when LBP features are classified with kNN. However, at the same time LBP with kNN gives the worst result in terms of over-detections. Total tomato detection with these two methods is 233, which is the worst of all. On the other hand, HOG with kNN gives the best result in terms of over-detections. However, missed detections of this combination is the worst of all. When we consider missed and over-detections together, HOG and SVM combination achieves the best result, which is followed by HOG+LBP and SVM combination where missed detections result is better.

An example of tomato detection by classifying HOG and LBP features with SVM and kNN is given in Figure 5 (a) and (b), respectively. As seen in this figure, while all tomatoes were detected with SVM, one of the tomatoes were missed by kNN.

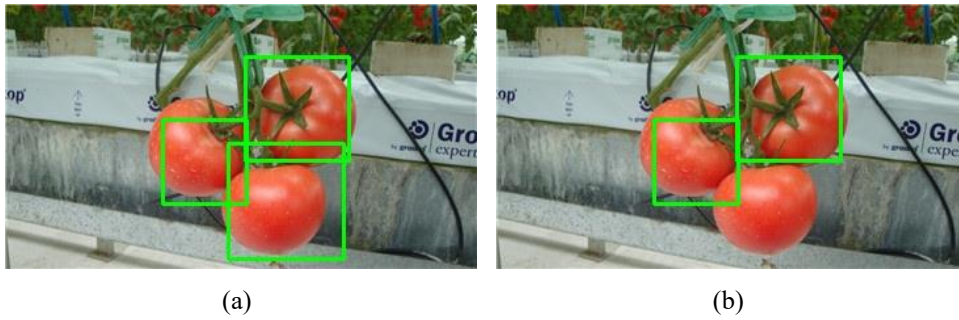


FIGURE 5. An example of the classification of HOG and LBP features by (a) SVM and (b) kNN for tomato detection.

6. CONCLUSION

In this paper, a comparative study was conducted for tomato detection with HOG and LBP feature extraction and SVM and kNN classifier algorithms. Before the detection process, images were enhanced to improve detection success. At the postprocessing stage, false positive results were eliminated by FCR. Finally, NMS was used to select a single detection result out of multiple overlapping detections for a tomato. In the first experiment, the detection success measured by various metrics was investigated for the images with a size of 64×64 pixels. The best result was obtained for the classification of HOG and LBP features by SVM. In the second experiment, where tomatoes were searched in a ROI extracted from full-sized images, the best results were obtained for the classification of HOG features by SVM, which is followed by HOG and LBP features classified by SVM. The kNN classifier, on the other hand, gives different results. In other words, while it gives the best result for missed detections for classifying LBP features, it obtains the worst result for over detections. In addition, while HOG classification by kNN achieves the best result for over detections, it obtains the worst result in terms of missed detections. In general, classification by SVM gives better results than kNN.

Author Contribution Statements The authors equally worked on the study. All authors read and approved the final copy of the manuscript.

Declaration of Competing Interests The authors declare that there is no conflict of interest regarding the publication of manuscript.

Acknowledgement The research was supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant No 7201372.

REFERENCES

- [1] Liu, G., Mao, S., Kim, J. H., A mature-tomato detection algorithm using machine learning and color analysis, *Sensors* 2019, 19 (2023), <https://doi.org/10.3390/s19092023>.
- [2] Onishi, Y., Yoshida, T., Kurita, H. et al., An automated fruit harvesting robot by using deep learning, *Robomech J.*, 6 (13) (2019).
- [3] Selvaraj, A., Shebiah, N., Nidhyananthan, S., Ganesan, L., Fruit recognition using color and texture features, *J. Emerg. Trends Comput. Inf. Sci.*, 1 (2010), 90-94.
- [4] Bulanon, D. M., Kataoka, T., Ota, Y., Hiroma, T., AE-Automation and emerging technologies: A segmentation algorithm for the automatic recognition of fuji apples at harvest, *Biosyst. Eng.*, 83 (4) (2002), 405-412.
- [5] Liu, X., Zhao, D., Jia, W., Ji, W., & Sun, Y., A detection method for apple fruits based

- on color and shape features, *IEEE Access*, 7 (2019), 67923–67933.
- [6] Mao, W., Ji, B., Zhan, J., Zhang, X. and Hu, X., Apple location method for the apple harvesting robot, *2009 2nd International Congress on Image and Signal Processing*, (2009),1-5.
- [7] Tanigaki, K., Fujiura, T., Akase, A., Imagawa, J., Cherry-harvesting robot, *Comput. Electron. Agric.*, 63 (1) (2008), 65-72.
- [8] Ji, W., Zhao, D., Cheng, F., Xu, B., Zhang, Y., Wang, J., Automatic recognition vision system guided for apple harvesting robot, *Comput. Elec. Eng.*, 38 (5) (2012), 1186-1195.
- [9] Kurtulmus, F., Lee, W.S., Vardar, A., Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network, *Precision Agric.*, 15 (2014) 57-79.
- [10] Song, Y., Glasbey, C. A., Horgan, G. W., Polder, G., Dieleman, J. A., van der Heijden, G. W. A. M., Automatic fruit recognition and counting from multiple images, *Biosyst. Eng.*, 118 (2014), 203-215, <https://doi.org/10.1016/j.biosystemseng.2013.12.008>.
- [11] Islam, M. A., Yousuf, Md. S. I., Billah, M. M., Automatic plant detection using HOG and LBP features with SVM, *Int. J. Comput.*, 33 (1) (2019), 26-38.
- [12] Hummel, R. A., Image enhancement by histogram transformation, *Comput. Graph. Image Process.*, 6 (1977), 184-195.
- [13] Ketcham, D. J., Lowe, R. W. and Weber, J. W., Real-time image enhancement techniques, *Seminar on Image Processing*, (1976), 1-6.
- [14] Pizer, S. M., Intensity mappings for the display of medical images, *Functional Mapping of Organ Systems and Other Computer Topics*, Society of Nuclear Medicine (1981).
- [15] Pizer, S. M., Amburn, E. P., Austin, J. D., et al., Adaptive histogram equalization and its variations, *Comput. Vis. Graph. Image Process.*, 39 (1987), 355-368.
- [16] Maison, Lestari, T., Luthfi, A., Retinal blood vessel segmentation using gaussian filter, *J. Phys.: Conf. Ser.*, 1376 (2019), 012023, <https://doi.org/10.1088/1742-6596/1376/1/012023>.
- [17] Umri, B. K., Utami, E. and Kurniawan, M. P., Comparative analysis of CLAHE and AHE on application of CNN algorithm in the detection of Covid-19 patients, *2021 4th Int. Conf. on Inf. and Comm. Tech. (ICOIACT)*, (2021), 203-208, <https://doi.org/10.1109/ICOIACT53268.2021.9563980>.
- [18] Wang, Q., Lu, Y., Zhang, X., Hahn, J., Region of interest selection for functional features, *Neurocomputing*, 422 (2021), 235-244.
- [19] Zhang, L., Sun, Q. and Zhang, J., Region of interest extraction via common salient feature analysis and feedback reinforcement strategy for remote sensing images, *GI Science Remote Sens.*, 55 (5) (2018), 745-762.
- [20] Vogt, P., Riitters, K. H., Estreguil, C., Kozak, J., Wade, T. G., Wickham, J. D. , Mapping spatial patterns with morphological, *Image Process.*, 22 (2) (2007), 171-177.
- [21] Dalal, N. and Triggs, B., Histograms of oriented gradients for human detection, *2005*

- IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, (2005), 886-893.
- [22] Tuncer, T. and Avcı, E., Yerel ikili örüntü tabanlı veri gizleme algoritması: LBP-LSB, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 10 (1) (2017) 48-53.
- [23] Chen, J., Kellokumpu, V., Zhao, G., Pietikainen, M., RLBP: Robust Local Binary Pattern, *Proceedings British Machine Vision Conference 2013*, 122 (2014),1-11, <http://dx.doi.org/10.5244/C.27.122>.
- [24] Cortes, C. and Vapnik, V., Support-Vector Networks, *Mach. Learn.*, 20 (1995), 273-297, <http://dx.doi.org/10.1007/BF00994018>.
- [25] Kuang, Q. and Zhao, L., A practical GPU based KNN algorithm, *Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCST '09)*, (2009), 151-155.
- [26] Li, H., Chen, L., Removal of false positive in object detection with contour-based classifiers, *2010 IEEE International Conference on Image Processing*, (2010), 3941-3944.
- [27] Rothe, R., Guillaumin, M., Van Gool, L., Non-maximum suppression for object detection by passing messages between windows, *Computer Vision ACCV 2014*, 9003 (2015), 290-306, https://doi.org/10.1007/978-3-319-16865-4_19.
- [28] Salscheider, N. O., FeatureNMS: Non-maximum suppression by learning feature embeddings, *25th International Conference on Pattern Recognition (ICPR)*, (2020), 7848-7854.
- [29] Liu, G., Mao, S., Open tomatoes dataset, (2019). Available at: <https://github.com/pandalgx/Tomato-dataset>. [Accessed February 2023].

DISEASE DETECTION IN BEAN LEAVES USING DEEP LEARNING

Soydan SERTTAS¹ and Emine DENİZ¹



¹Kütahya Dumlupınar University, Department of Computer Engineering,
Kütahya, TÜRKİYE



ABSTRACT. The care and health of agricultural plants, which are the primary source for people to eat healthily, are essential. Disease detection in plants is one of the critical elements of smart agriculture. In parallel with the development of artificial intelligence, advancements in smart agriculture are also progressing. The development of deep learning techniques positively affects smart farming practices. Today, using deep learning and computer vision techniques, various plant diseases can be detected from images such as photographs. In this research, deep learning techniques were used to detect and diagnose bean leaf diseases. Healthy and diseased bean leaf images were used to train the convolutional neural network (CNN) model, which is one of the deep learning techniques. Transfer learning was applied to CNN models to detect plant diseases with the difference of related works. A transfer learning-based strategy to identify various diseases in plant varieties is demonstrated using leaf images of healthy and diseased plants from the Bean dataset. With the proposed method, 1295 images were studied. The results show that our technique successfully identified disease status in bean leaf images, achieving an accuracy of 98.33% with the ResNet50 model.

1. INTRODUCTION

Accurate plant disease diagnosis is critical to well-being and health. Diseases, which are the primary factor affecting plants' growth, can reduce plant production by an average of more than 10% annually [1]. Not only do diseases lead directly to reduced plant yields, but also have a significant impact on produce quality and even raise concerns about the safety of food. To limit the use of chemical pesticides, save money, and reduce environmental pollution, early detection of plant-damaging diseases is essential. It is difficult for a person to detect problems with plant diseases

Keywords. Deep learning, bean leaf diseases, image processing, convolutional neural networks.

 soydan.serttas@dpu.edu.tr;  0000-0001-8887-8675

 emine.deniz@dpu.edu.tr-Corresponding author;  0000-0003-0670-3578.

with the naked eye [2]. Also, doing this over and over is a laborious and inefficient task. To accurately detect plant diseases, a plant pathologist must have good observational skills to be able to identify the characteristic symptoms. Therefore, the automatic identification of plant diseases figures prominently in identifying the illness type early and reducing production loss [3]. Recently, the increasing use of affordable smartphones among farmers has created an opportunity to classify diseases using images of diseased leaves [4].

Using deep learning and image processing, there has been much study on disease diagnosis in late years. Because deep learning is high in classification provides success. Machine learning algorithms such as K-Nearest Neighbors (KNN), K-Means, support vector machines (SVM) and artificial neural networks (ANN) have been used to detect plant diseases. Deep learning offers a new and modern methodology for processing images and data analysis. Deep learning has been successfully applied in several fields and has recently been used in agriculture as well [5]. There are promising developments in obtaining the most distinctive features in plants with CNN methods [2]. These developments have generally used fine-tuned transfer learning [6] methods. Deep Learning-based plant disease classification models include the use of various models such as AlexNet, GoogleNet, VGGNet in transfer learning. Within the scope of this study, three different pre-trained transfer models were used separately in the creation of the CNN model.

Pre-trained models are a common way for transfer learning to be expressed. The models used to solve the problem have previously been used to solve a similar problem and are trained on a large benchmark dataset. The pre-trained VGG-16, ResNet50 and MobileNetV2 models were used in this study.

The 1295 bean leaf images consisting of three classes, two diseased and one healthy, were used in this work. The images are divided into three sets. 128 are test sets, 1034 are training sets, and 133 are validation sets. Examples of diseased bean leaves are shown in Figure 1.

2. LITERATURE REVIEW

To detect plant diseases, it is necessary to examine research related to the identification of plant species. There are various studies on the detection of plant and leaf types. Vishnoi et al. introduced a number of methods related to obtaining images, pre-processing steps, techniques for identifying lesions in the images, extracting features, and classifiers [7]. The challenges have been outlined and the shortcomings of current systems have been examined. The work has also presented a range of computer vision techniques and has also provided an illustration of the research in the future. Unal et al. published a paper explaining modern learning techniques such as ANN and transfer learning [8]. This paper aims to classify plant

seedling images using two CNN architectures to test transfer learning. The Plant Seedlings dataset of Aarhus University was used in the paper. While the VGGNet architecture correctly classified 75% of plant images, the success rate of the AlexNet architecture was recorded to be close to 90%. Tuğrul classified five different types of rice using four different CNN architectures. VGG, ResNet, and EfficientNets architectures were trained and results were obtained. In this study, the VGG architecture achieved the best accuracy value of 97% [9]. Camgözlü and Kutlu studied five different leaf datasets with fixed background images, leaf images at different scales, and the combined version of these datasets with the ESA network [10]. The convolution filter size, the number of pooling layers, and the type of model to be used were determined by examining the variations in the image sizes of the datasets. In addition, the effect of whether leaf images had colored or grayscale background was examined. The results obtained as a result of these processes were evaluated by comparing the number of images, the number of species used and the obtained performances of the studies using CNN in the literature.

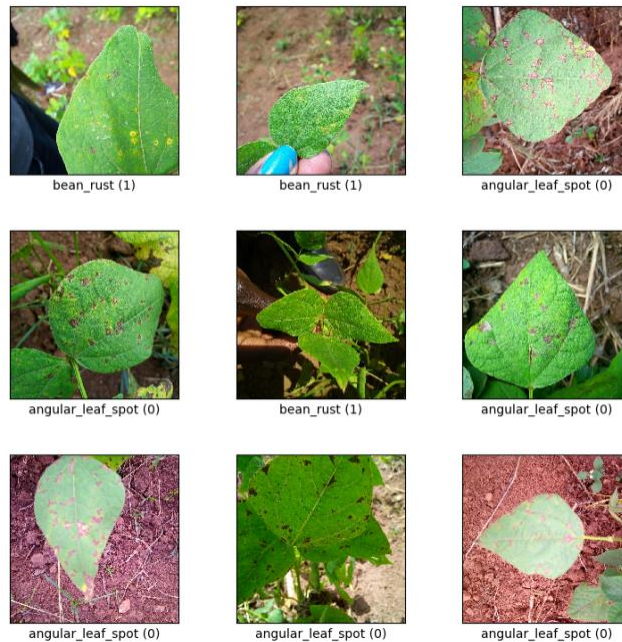


FIGURE 1. An example leaf image from the bean dataset in TensorFlow. It represents each plant-disease pair used.

There are various studies on the detection of leaf diseases. Yaman and Tuncer detected the disease in walnut leaves using deep learning and feature selection methods [5]. They collected 726 images of walnut leaves for their proposed method. The study involved testing 17 different deep learning models, and the two most successful ones, namely DarkNet53 and ResNet101, were chosen. A combination of features derived from both models was utilized to generate a hybrid feature extraction approach. The SVM algorithm was used to classify the selected features. Çetiner in his study examining apple leaf diseases obtained input layers from preprocessed plant disease images using ResNet152V2, DenseNet201, MobileNetV2, and ResNet50V2 pre-trained network models [2]. The proposed DenseNet201 architecture achieved an accuracy of 96%, ResNet50V2 achieved 94%, ResNet101V2 achieved 93%, ResNet152V2 achieved 94%, and MobileNetV2 achieved 97%. Göksu developed two deep learning models for the classification of corn diseases, including corn rust, gray leaf spot, leaf blight, and normal (healthy), based on corn leaves [11]. Model-1 was created using the EfficientNetB5 network. CNN layers were used to create the model called Model-2. Transfer learning was performed using the EfficientNetB5 network in Model-1, which achieved a success rate of 92.03% on test data. In Model-2, the success rate on test data was obtained as 89.88%. Sert proposed an approach to identify the type of disease in pepper and potato leaves [12]. This study presents an object detection approach in which Faster R-CNN and GoogLeNet architecture work together. The proposed Faster R-CNN-GC achieved an accuracy of 98.06% on the Plant Village dataset.

There are also various studies on the detection of bean leaf diseases. Önlü proposed an ANN model for bean leaf disease detection [13]. The network was constructed by integrating descriptive vectors from bean leaves with the transfer learning feature extraction and histogram oriented gradient feature extraction methods. In the work, the bean leaf dataset consists of images about bean rust, angular leaf spot and healthy classes. There are 1034 images in the training dataset, 128 images in the test dataset, and 134 images in the validation dataset. The model has achieved 98.33, 98.40 and 99.24% accuracy in training, validation, and test datasets, respectively. Abed and Esmaeel studied the detection of powdery mildew and bacterial brown spot diseases on bean leaves [14]. It's indicated that the developed methodology successfully detected the two types of leaf diseases with an accuracy of 100%. Abed et al. proposed a framework in real-time to determine the health condition of bean leaves using DNNs [15]. In the work, the U-Net architecture has been used to identify and locate the bean leaves within the input images. The architecture of this system relies on a ResNet34 encoder that was previously trained. To determine the healthiness of bean leaves, the performance of five deep learning models - VGG, VGG-16, ResNet34, Densenet121, and ResNet50 - has been evaluated. The performance of the framework has been evaluated by testing it on a

dataset comprising 1295 images that were classified into three distinct categories. These classes are bean rust, angular leaf spot, and healthy. The Densenet121 architecture with a Specificity of 96.82%, Sensitivity of 99.03%, Precision of 98.45%, CAR of 98.31%, and F1-Score of 98.74% has achieved the best success.

3. MATERIAL AND METHOD

Deep learning models are used to automatically classify bean leaf diseases and control large crop fields [16]. To apply a previously learned model to new tasks and contexts, transfer learning is frequently employed in image recognition. By learning a new feature space, transfer learning enables classifiers to maintain their performance on incoming data with new classes and distributions. Layers of a pre-trained model with reusable features may be trained in transfer learning on either an existing dataset or a new dataset [17]. The input used to train a much smaller network with fewer parameters uses features from this layer.

In transfer learning, freezing a layer refers to not changing the weights of that layer throughout training [18]. By doing this, the better characteristics that had already been extracted by the filters in the earlier layers will not be changed. On trainable or unfrozen layers, the new dataset is trained. While the present network trains using remaining trainable parameters, parameters in frozen layers remain untrainable. This yields a very high calculation time efficiency compared to backpropagation and updating the weights of all network layers. The number of trainable parameters lowers as the number of layers grows, which in turn reduces calculation time. By adding reusable features to already-trained models, it may be utilized as a feature extractor. The last classifier is the only block that is not frozen. When using the model as a feature extractor, the number of trainable blocks is zero, which ensures that the model is operated with as few trainable parameters as possible [19]. Figure 2 and Figure 3 illustrate the approaches used in this article. In Figure 2, new layers to be trained have been added to the previously trained data. These layers were trained and outputs were obtained. In the approach in Figure 3, results are obtained by retraining the last few layers of the previously trained model and adding new layers to be trained.

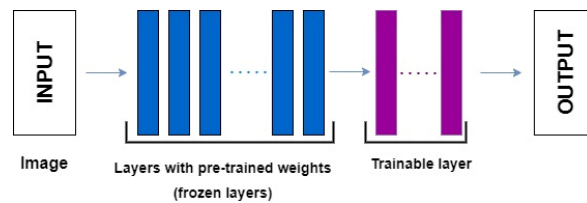


FIGURE 2. Transfer learning, the first technique.

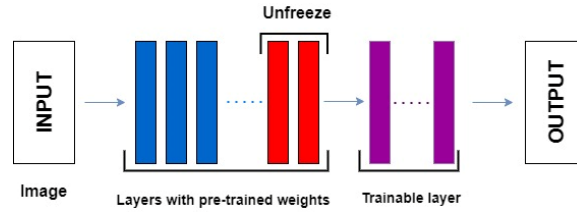


FIGURE 3. Transfer learning, the second technique.

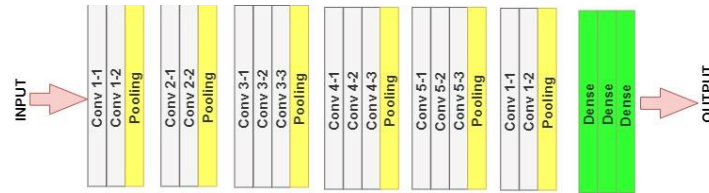


FIGURE 4. VGG-16 model.

VGG-16, ResNet50 and MobileNetV2 are used for the pre-trained model of Figure 2 and Figure 3. Simonyan and Zisserman introduced the CNN model VGG16, which is shown in Figure 4 [20]. The greatest distinguishing characteristic of VGG16 is that it consistently employs the same padding and pooling layer of the 2x2 filter with step 2 and focuses on the convolution layers of the 3x3 filter with step 1 rather than having many hyperparameters. The 16 in VGG16 indicates that there are 16 weighted layers. This network contains about 140 million parameters, making it a sizable one.

Figure 5 depicts the network architecture of MobileNetV2, which is made up of 19 original basic blocks known as bottleneck residual blocks [21]. A 1x1 convolution layer with an average pooling layer follows these blocks. A classification layer makes up the last layer.

There are many variants of ResNet that work on the same basic idea but differ in the number of layers [22]. One maximum pooling layer, one average pooling layer, and 48 convolution layers make up the ResNet50 model shown in Figure 6.

4. RESULTS

4.1. Dataset. Bean Dataset is a dataset of bean images that were taken outside by smartphones. It has three classes: one for health and two for diseases. Angular leaf spot and bean rust are two of the diseases mentioned. Uganda National Plant Resources Research Institute provided the annotations for the data that the Makerere

AI research lab gathered. Of 1295 images, 128 were used for testing, 1034 for training and 133 for validation.

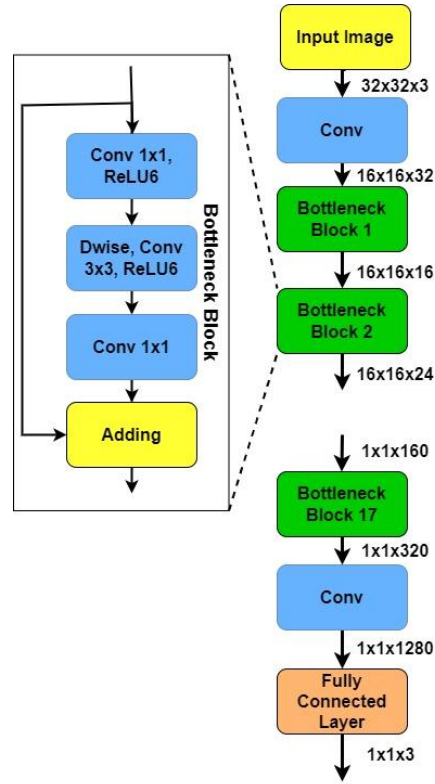


FIGURE 5. MobileNetV2 model.

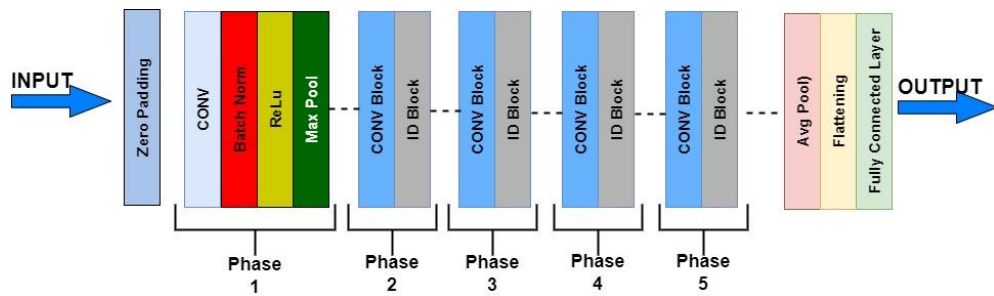


FIGURE 6. ResNet50 model.

4.2. Experimental Results. In the transfer learning technique shown in Figure 2, VGG16 was used for the pre-trained model and then one layer was added for the training and the accuracy graph in Figure 7 was obtained. The accuracy graph in Figure 8 was obtained by adding the second layer. Finally, the third layer was added and the accuracy graph in Figure 9 was obtained.

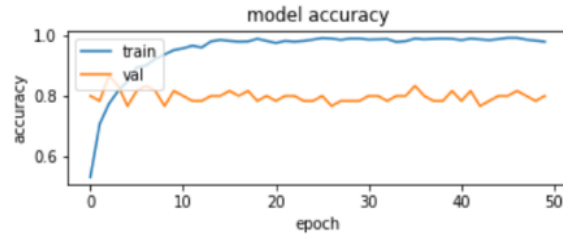


FIGURE 7. Accuracy graph when one layer is added to the VGG16 model by applying the first technique.

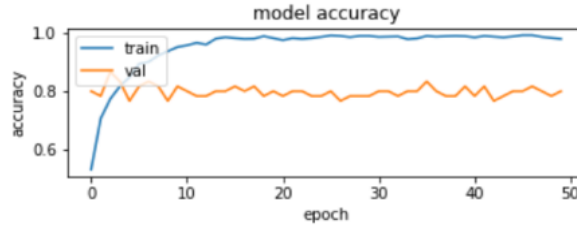


FIGURE 8. Accuracy graph when two layers are added to the VGG16 model by applying the first technique.

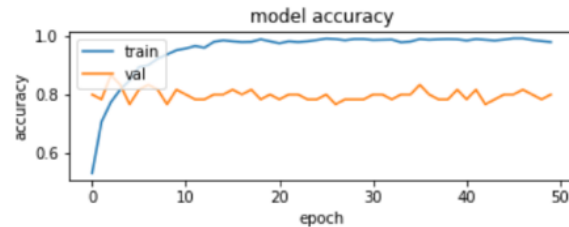


FIGURE 9. Accuracy graph when three layers are added to the VGG16 model by applying the first technique.

The pre-trained model in the transfer learning method depicted in Figure 3 was VGG16, and the last two layers of the VGG16 model were trained using later-added layers, respectively. By adding one layer, the accuracy graph in Figure 10 was obtained. The accuracy graph in Figure 11 was obtained by adding the second layer. Finally, the 3rd layer was added and the accuracy graph in Figure 12 was obtained.

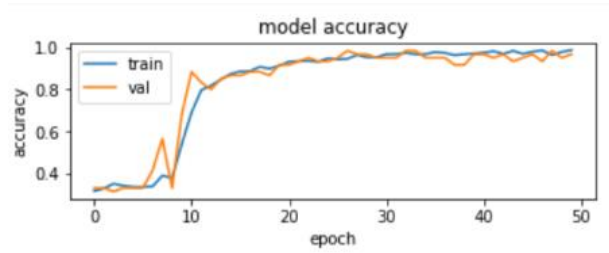


FIGURE 10. Accuracy graph when one layer is added to the VGG16 model by applying the second technique.

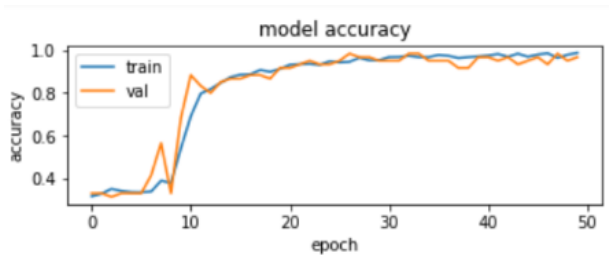


FIGURE 11. Accuracy graph when two layers are added to the VGG16 model by applying the second technique.

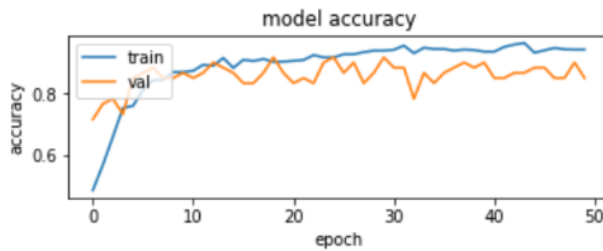


FIGURE 12. Accuracy graph when three layers are added to the VGG16 model by applying the second technique.

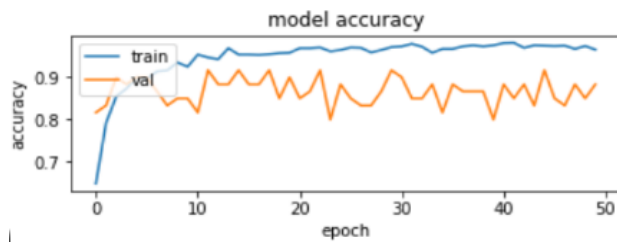


FIGURE 13. Accuracy graph when one layer is added to the MobileNetV2 model by applying the first technique.

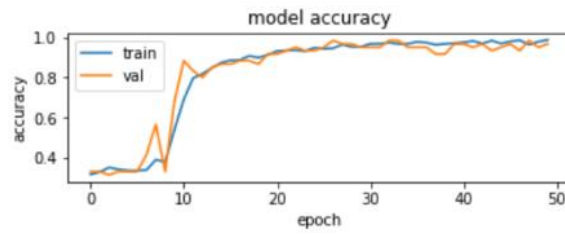


FIGURE 14. Accuracy graph when two layers are added to the MobileNetV2 model by applying the first technique.

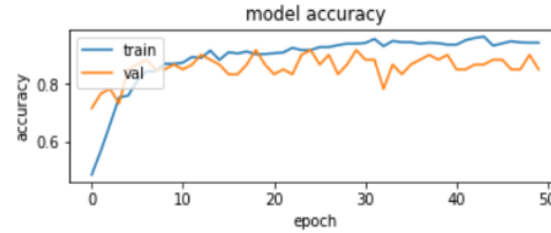


FIGURE 15. Accuracy graph when three layers are added to the MobileNetV2 model by applying the first technique.

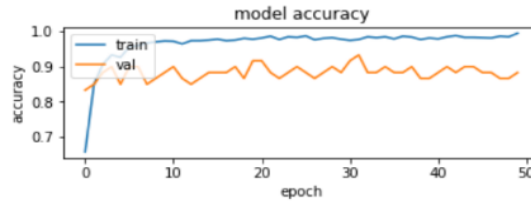


FIGURE 16. Accuracy graph when one layer is added to the MobileNetV2 model by applying the second technique.

In the transfer learning technique shown in Figure 2, MobileNetV2 was used for the pre-trained model and then one layer was added for the training and the accuracy graph in Figure 13 was obtained. The accuracy graph in Figure 14 was obtained by adding the second layer. Finally, the third layer was added and the accuracy graph in Figure 15 was obtained.

In the transfer learning technique shown in Figure 3, MobileNetV2 was used for the pre-trained model, and it was trained with the last two layers of the MobileNetV2 model and then added layers respectively. By adding one layer, the accuracy graph in Figure 16 was obtained. The accuracy graph in Figure 17 was obtained by adding the second layer. Finally, the third layer was added and the accuracy graph in Figure 18 was obtained.

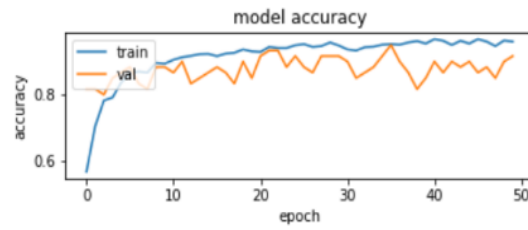


FIGURE 17. Accuracy graph when two layers are added to the MobileNetV2 model by applying the second technique.

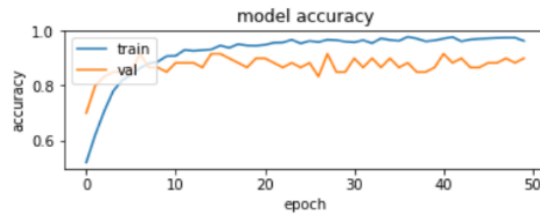


FIGURE 18. Accuracy graph when three layers are added to the MobileNetV2 model by applying the second technique.

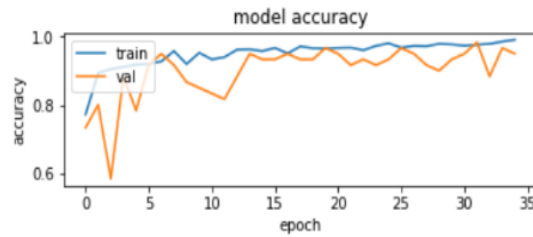


FIGURE 19. Accuracy graph when one layer is added to the ResNet50 model by applying the first technique.

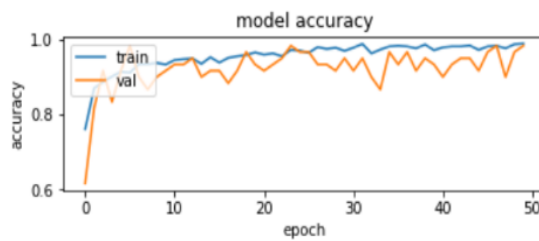


FIGURE 20. Accuracy graph when two layers are added to the ResNet50 model by applying the first technique.

In the transfer learning technique shown in Figure 2, ResNet50 was used for the pre-trained model and then one layer was added for the training and the accuracy graph in Figure 19 was obtained. The accuracy graph in Figure 20 was obtained by adding the second layer. Finally, the third layer was added and the accuracy graph in Figure 21 was obtained.

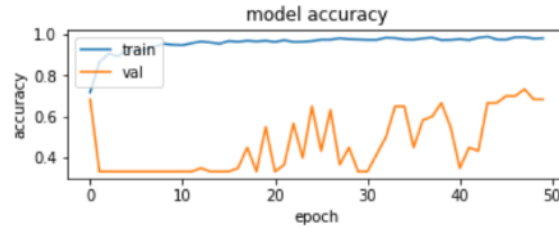


FIGURE 21. Accuracy graph when three layers are added to the ResNet50 model by applying the first technique.

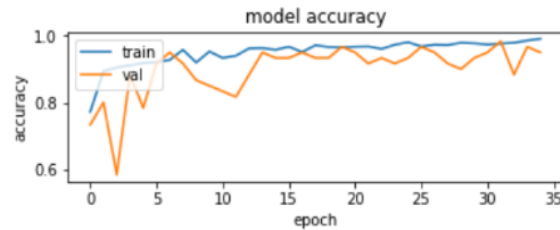


FIGURE 22. Accuracy graph when one layer is added to the ResNet50 model by applying the second technique.

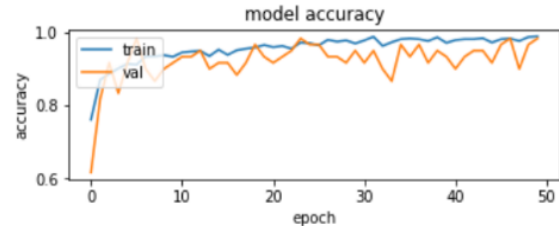


FIGURE 23. Accuracy graph when two layers are added to the ResNet50 model by applying the second technique.

In the transfer learning technique shown in Figure 3, ResNet50 was used for the pre-trained model and trained with the last two layers of the ResNet50 model and then added layers respectively. By adding one layer, the accuracy graph in Figure 22 was obtained. The accuracy graph in Figure 23 was obtained by adding the second

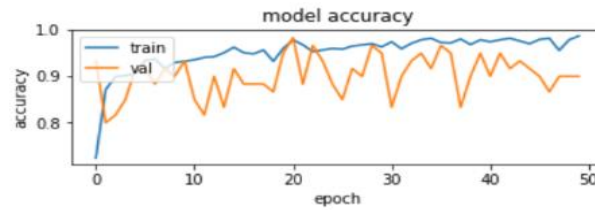


FIGURE 24. Accuracy graph when three layers are added to the ResNet50 model by applying the second technique.

layer. Finally, the third layer was added and the accuracy graph in Figure 24 was obtained.

The outcomes of the first and second techniques in three different models are displayed in Table 1. It can be shown that the ResNet50 model achieves the greatest accuracy value.

TABLE 1. Accuracy values of the first and the second techniques according to models.

Deep learning models	Accuracy values	
	1. Technic	2. Technic
VGG - 16	%81.67	%98.33
MobileNetV2	%88.33	%70.00
ResNet50	%91.67	%98.33

5. CONCLUSIONS

Plants are one of the important resources that provide an ecological balance for the planet. Plant diseases limit agricultural production, compromising access to food. Therefore, plants are healed quickly when diseases are detected early. The use of deep learning techniques in agriculture provides early detection of plant diseases. This article discusses the deep learning method used for plant leaf classification. In this method, three different CNN architectures were used and a different transfer learning technique was applied to each architecture. Different architectures and techniques were used to classify plant leaves, and results were obtained accordingly.

When the first technique was used, the VGG16 architecture correctly classified 82% of the plant images, while MobileNetV2 achieved an 88% success rate. The success rate of the ResNet50 architecture is close to 92%. That is, the ResNet50

architecture provided the highest success rate and achieved the highest success rate in this experiment. The VGG16 architecture correctly classified 98% of the plant images when the second technique was used, while the MobileNetV2 architecture achieved a 70% success rate. ResNet50 architecture, on the other hand, provided a success rate of 98% in this technique. In both techniques used in the experiment, the ResNet50 architecture provided the highest success rate. In future studies, it is aimed to detect more various diseases in different data sets.

Author Contribution Statements

Soydan SERTTAŞ: He contributed to the determination of the study subject, planning, execution, determination of the method, preparation of the material, conducting the analysis, evaluation of the results and making the study into an article.

Emine DENİZ: She contributed to the determination of the study subject, planning, execution, determination of the method, conducting the analysis, evaluation of the results and making the study into an article.

Declaration of Competing Interest The authors declare that there is no conflict of interest regarding the publication of this paper.

REFERENCES

- [1] Lee, S., Chan, C., Mayo, S. J., & Remagnino, P., How deep learning extract and learns leaf features for the plant classification, *Pattern Recognit.*, 71 (2017), 1-13, <https://doi.org/10.1016/j.patcog.2017.05.015>.
- [2] Çetiner, H., Yaprak hastalıklarının sınıflandırılabilmesi için önceden eğitilmiş ağ tabanlı derin ağ modeli, *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, 8 (15) (2021), 442-456.
- [3] Sagar, A., Dheeba, J., On using transfer learning for plant disease detection, *BioRxiv*, (2020), <https://doi.org/10.13140/RG.2.2.12224.15360/1>.
- [4] Kamilaris, A., Prenafeta-Boldú, F. X., Deep learning in agriculture: A survey, *Comput. Electron. Agric.*, 147 (2018), 70-90, <https://doi.org/10.1016/j.compag.2018.02.016>.
- [5] Yaman, O., Tuncer, T., Bitkilerdeki yaprak hastalığı tespiti için derin özellik çıkarma ve makine öğrenmesi yöntemi, *FÜMBD*, 34 (1) (2022), 123-132, <https://doi.org/10.35234/fumbd.982348>.
- [6] Liu, B., Zhang, Y., He, D., Li, Y., Identification of apple leaf diseases based on deep convolutional neural networks, *Symmetry*, 10 (1) (2017), 11, <https://doi.org/10.3390/sym10010011>.
- [7] Vishnoi, V., Kumar, K., Kumar, B., Plant disease detection using computational intelligence and image processing, *JPDP*, 128 (2021), 19-53, <https://doi.org/10.1007/s41348-020-00368-0>.

- [8] Unal, M., Bostancı, E., Guzel, M. S., Aydın, A., Modern learning techniques and plant image classification, *Commun. Fac. Sci. Univ. Ank. Series A2-A3*, 62 (2) (2020), 153-163.
- [9] Tuğrul, B., Classification of five different rice seeds grown in Turkey with deep learning methods, *Commun. Fac. Sci. Univ. Ank. Series A2-A3*, 64 (1) (2022), 40-50.
- [10] Camgözlü, Y., Kutlu, Y., Yaprak sınıflandırmak için yeni bir evrişimli sinir ağı modeli geliştirilmesi, *BŞEÜ Fen Bilimleri Dergisi*, 8 (2) (2021), 567-574.
- [11] Göksu, M., Sünnetci, K. M., Alkan, A., Derin öğrenme ağları kullanılarak mısır yapraklarında hastalık tespiti, *Comput. Sci.*, (Special) (2021), 208-216.
- [12] Sert, E., A deep learning based approach for the detection of diseases in pepper and potato leaves, *Anadolu Tarım Bilimleri Dergisi*, 36 (2) (2021), 167-178.
- [13] Önler E., Feature fusion based artificial neural network model for disease detection of bean leaves, *ERA*, 31 (5) (2023), 2409-2427, <https://doi.org/10.3934/era.2023122>.
- [14] Abed, S., Esmaeel, A. A., A novel approach to classify and detect bean diseases based on image processing, *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, (2018), 297-302, <https://doi.org/10.1109/ISCAIE.2018.8405488>.
- [15] Abed, S. H., Al-Waisy, A. S., Mohammed, H. J. et al. A modern deep learning framework in robot vision for automated bean leaves diseases detection, *Int. J. Intell. Robot. Appl.*, 5 (2021), 235-251, <https://doi.org/10.1007/s41315-021-00174-3>.
- [16] Muneer, A., Fati, S. M., Efficient and automated herbs classification approach based on shape and texture features using deep learning, *IEEE Access*, 8 (2020), 196747-196764, <https://doi.org/10.1109/access.2020.3034033>.
- [17] Rautaray, S. S., Pandey, M., Gourisaria, M. K., Sharma, R., Das, S., Paddy crop disease prediction- a transfer learning technique, *IJRTE*, 8 (6) (2020), 1490-1495, <https://doi.org/10.35940/ijrte.f7782.038620>.
- [18] Lu, J., Tan, L., Jiang, H., Review on convolutional neural network (CNN) applied to plant leaf disease classification, *Agriculture*, 11 (8) (2021), 707, <https://doi.org/10.3390/agriculture11080707>.
- [19] Sumalatha, G., Singothu, J. R., Rao, S. K., Transfer learning-based plant disease detection, *IJIEMR*, 10 (3) (2021), 469-477, <https://doi.org/10.48047/IJIEMR/V10/I03/99>.
- [20] Tammina, S., Transfer learning using vgg-16 with deep convolutional neural network for classifying images, *IJSRP*, 9 (10) (2019), 143-150, <https://doi.org/10.29322/ijssrp.9.10.2019.p9420>.
- [21] Falconí, L. G., Pérez, M., Aguilar, W. G., Transfer learning in breast mammogram abnormalities classification with mobilenet and nasnet, *2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*, (2019), 109-114, <https://doi.org/10.1109/iwSSIP.2019.8787295>.
- [22] Mukti, I. Z., Biswas, D., Transfer learning based plant diseases detection using ResNet50, *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, (2019), 1-6, <https://doi.org/10.1109/eict48899.2019.9068805>.

AN OPTIMIZED ARTIFICIAL NEURAL NETWORK FOR ESTIMATING DESIGN EFFORT OF JIGS AND FIXTURES USED IN AVIATION INDUSTRY

Umut Nazmi AKTAN¹ and Mehmet DİKMEN²

¹Product Lifecycle Management Process and Method, Turkish Aerospace Company,
Ankara, TÜRKİYE



²Computer Engineering Department, Başkent University, Ankara, TÜRKİYE

ABSTRACT. This paper investigates the usefulness of the machine learning methods to predict the design effort of jigs and fixtures used in the aviation industry. Reaching the best possible result by determining the ideal machine learning model to obtain the best estimate and the most appropriate set of inputs and parameters forms the basis of this study. To that end, most popular machine learning models that can be used for regression are combined with various data encoding methods. The best combination is optimized as well. The results showed that an optimized Artificial Neural Network architecture with binary encoding applied to the input data can be applied satisfactorily in the aviation industry for the solution of the given problem.

1. INTRODUCTION

An accurate estimation for design effort can make a significant difference in the time and cost expectation of a project. For this reason, each new method that can be applied in the estimation of design effort has a positive effect on the schedule of the relevant projects. In aviation industry, design efforts are relatively longer than the ones in most of the other industries. Since, the parts with very few details of a typical air vehicle are interchangeable, most jigs and fixtures (tools) require producing an aero structure and the subsystems. For example, a typical two people turboprop plane requires about 5000 different jigs and fixtures to manufacture.

Keywords. Tool design effort, aviation industry, artificial neural network, machine learning.

 nazmiumut@gmail.com;  0000-0002-6410-5720

 mdikmen@baskent.edu.tr -Corresponding author;  0000-0002-0584-5577.

Due to the need to reduce project costs, accurate estimation of design effort is crucial, as with most problems, this need can be met with machine learning methods. To that end, this paper focuses on the estimation of tool design efforts using machine learning (ML) methods, including Artificial Neural Networks (ANN) and optimization of its parameters to achieve the best result.

2. RESEARCH

In design process, technical requirements should be clearly stated so that the design can be done in the scheduled time span. Many input factors can affect the design effort. Function, complexity, and technical requirements can be considered as examples of the input factors. The more detailed and accurate the requirements for a design are defined, the more successful the estimation of the design time can be.

The most classic method of estimating the design time is the expert opinion. For example, by directing various problems to a team of expert designers, high-impact input factors affecting project costs can be identified [1]. The method of estimating the design cost by determining the changes on the new design with the existing design data that has already been completed can also be used effectively in the change management of similar design solutions [2]. In addition, it is possible to digitize with certain parameters by examining the dependent variables that affect the design at the highest rate in the design process. By examining the matrix structure obtained in this way, it is possible to establish a relationship between the design effort and the importance of the parameters [3].

Bashir and Thomson, who have more than one study on general design effort estimation, aimed to measure the design process by establishing a relationship between product complexity and design [4]. In addition, the authors tried to estimate the design effort by using parametric simulation, regression, and analogy methods [5], [6], [7]. On the other hand, the estimation of the production effort to be made after the design is easier since the available inputs are more detailed. In this regard, a method that can predict the production time of a workpiece to be machined on a Numerical Control (NC) machine by interpreting the parameters in Computer Aided Design (CAD) programs introduced [8].

One viable way to detect design and production effort is using digital twin-based architectures. In this context, the design efforts of the proposed integrated framework design and production processes can be estimated with a Model-Based Systems Engineering structure [9].

In a study of dataless design forecasting, a contribution was made in project management and project costing by using a fast and effective method that evaluates the tacit knowledge and experience of the design teams with an analytical method [10].

According to the research where 1178 articles were examined by natural language processing method in software effort detection studies published between 1996 and 2017, it was observed that the use of machine learning methods in software effort estimation increased in the last 15 years [11]. The study showed that ML methods were generally applied on software effort or measurement, not in tool design effort. In fact, to our knowledge, ML wasn't studied for the estimation of tool design effort yet. Therefore, a throughout analysis is needed to investigate the effectiveness of ML on this problem to reduce the need of expert opinion and time. To that end, this paper presents a study that aims to estimate the design effort of the production tools in the aviation industry, by using and comparing various optimized ML methods with state-of-the-art encoding techniques applied to the input data.

3. DATASET

In this study, Tool Order data set containing various input parameters related to the tool, which was collected from the data containing the necessary information for the realization of the tool design, was used [12]. Table 1 describes the input and output parameters.

TABLE 1. Tool Order dataset.

Parameter	Type	Data Type	Description
TOOLCODE	Input	Categorical	<u>Tool Code</u> ; describes the main function of the tool
PLANT	Input	Categorical	<u>Project Code</u> ; describes the project that the tool to be used
RFO	Input	Categorical	<u>Reason for Order</u> ; describes why the tool is requested
TOTYPE	Input	Categorical	<u>Tool Order Type</u> ; Describes whether tool is new or to be reworked or redesigned
TOTAL	Output	Real Number	<u>Total design time</u> (hours)

The data set, which was rearranged with expert tool designer in the previous study [12], was also used in this study. The expert examined all the data and identified incorrect inputs from other designers. For example, a designer can enter wrong work order code belonging to a project that he is not working on at that time. Such inconsistencies in the data set were removed from the data content so that, they would not cause any problems that would mislead the machine learning process.

In the previous study, Linear Regression (LR), Decision Tree Regression (DTR), Support Vector Regression (SVR) with Linear kernel function, Boosted Trees Regression (BTR) and Gaussian Process Regression (GPR) methods were utilized to solve the problem [12]. In addition, following encoding techniques were examined to digitize categorical inputs: Ordinal, Binary, One Hot, Dummy, Effect (Deviation) Frequency, Mean. The best results were obtained with binary encoding and training was performed with 80% of total data which were chosen randomly. As a result of this study shown in Table 2, the GPR method gave the best results. The term RMSE represents the Root Mean Square Error.

TABLE 2. Results of the Previous Study [12].

Machine Learning Method	RMSE
Linear Regression	8.764
Decision Tree Regression	8.863
Support Vector Regression (Linear Kernel)	9.051
Boosted Trees (Ensemble)	8.598
Gaussian Process Regression	8.418

In this paper, parameter optimization of all these methods including an ANN model is also performed. In addition, all models are verified with 10-fold cross validation instead of separating the data set randomly.

4. METHODOLOGY

To achieve the best result for estimating tool design effort, various machine learning methods should be investigated and optimized. Therefore, a framework specified in Figure 1 is proposed. Since, other encoding techniques for the methods given Table 2 were already examined in the previous study [12], only the best encoding technique for ANN is investigated in this paper.

ANN, which is a set of mathematical models inspired by nature, can be defined as a method of arranging the parameters of a set of nonlinear combined functions with input and output sets. The ANN architecture in this study can be summarized in Figure 2, where n and m are the number of neurons in input and hidden layer respectively, w and v are the real number weights between 0 and 1 [13]. Since, there is just one value to be estimated (i.e., the total design time), ANN contains a single output neuron. Also, b_j values are used as initial arbitrary constants (i.e., biases) to shift the regression functions. f_j 's represent the activation functions which are used to calculate the output (v_{fj}) of each neuron using the following sigmoid function.

$$f_j(x) = \begin{cases} g_j(x), & \text{if } \frac{1}{1 + e^{-g_j(x)}} > 0 \\ 0, & \text{otherwise} \end{cases}$$

where $g_j(x)$ represent the net input of a neuron calculated as:

$$g_j(x) \text{ is } g_j(x) = \sum_{i=1}^n x_i w_{x_i,j} + b_j$$

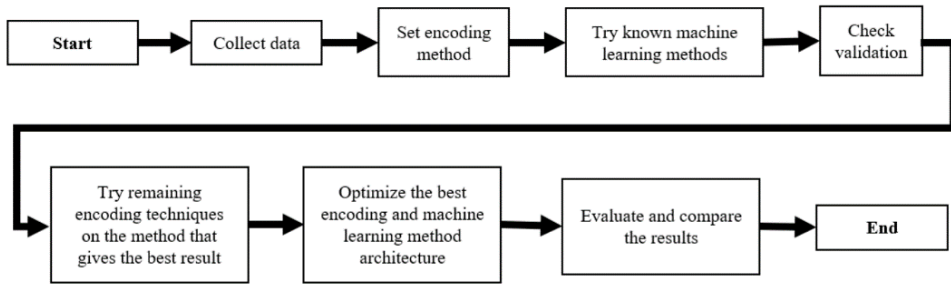


FIGURE 1. Framework to determine the best encoding and machine learning method.

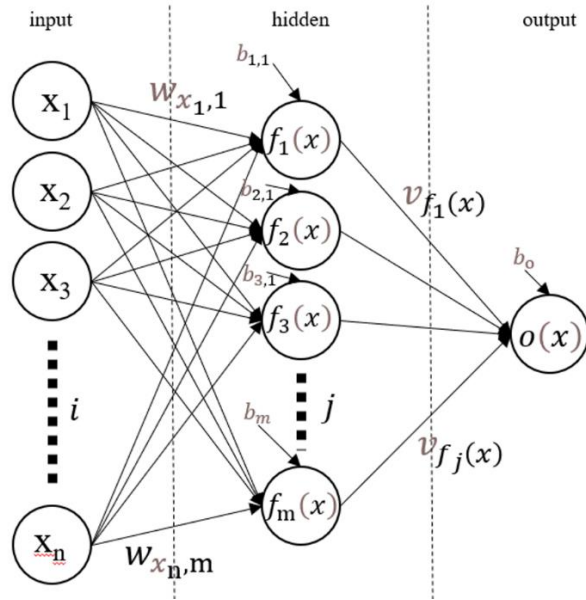


FIGURE 2. The initial ANN model diagram.

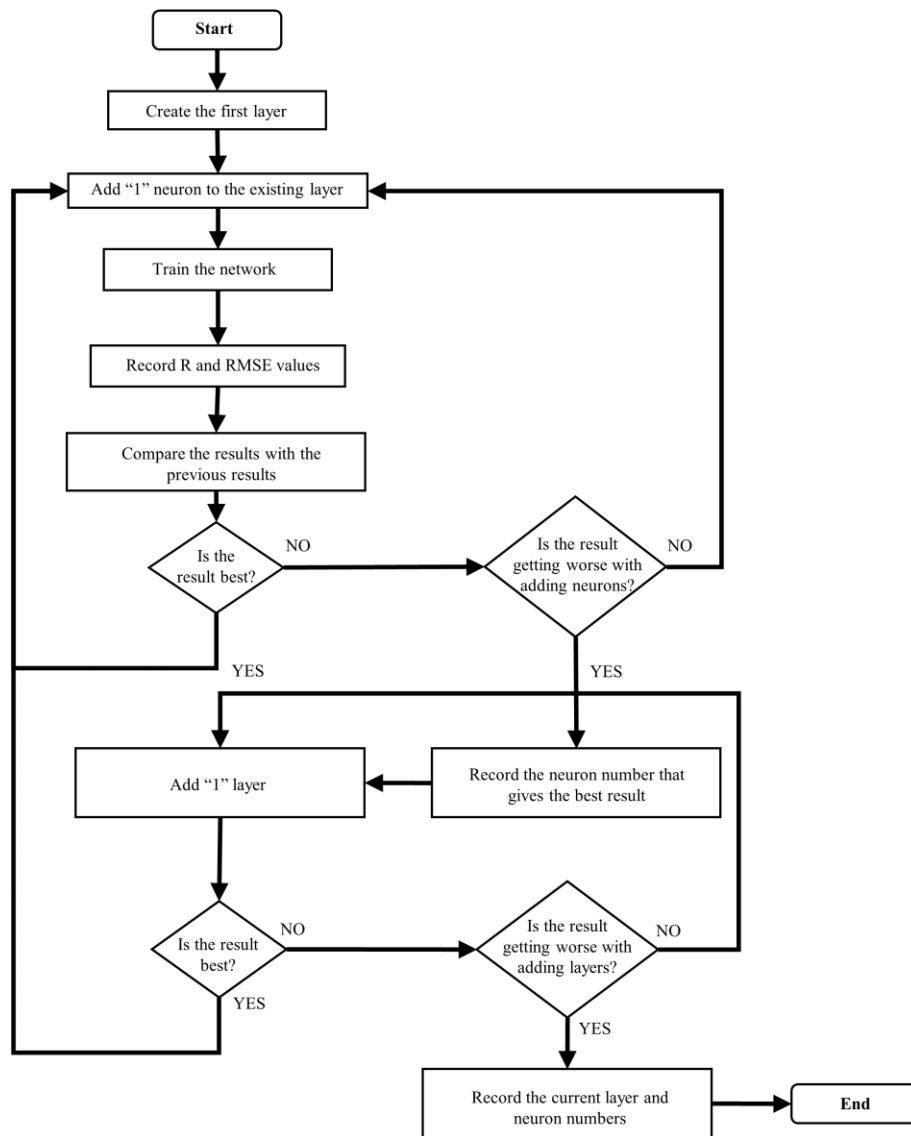


FIGURE 3. Suggested workflow for optimizing the ANN.

Training the ANN model was done by using Levenberg-Marquardt algorithm presented by Levenberg and Kenneth, which provides a fast convergence without

computing the exact Hessian matrix [14]. In addition, early stopping technique was employed to avoid the overfitting problem. For this purpose, 10% of the training data was randomly selected as the validation dataset and the prediction error on this dataset was observed. If no improvement is observed on validation error after 10 iterations, the training is stopped due to overfitting, and the best model weights were restored.

5. PARAMETER OPTIMIZATION OF MACHINE LEARNING MODELS

Each method except Linear Regression (LR) has parameters to optimize which are addressed in next paragraphs. The optimization procedure is limited to 500 iterations or 24 hours of computation time. For each parameter mentioned below, possible values are iterated. Only the parameter-value list that gave the best results is presented for each machine learning model.

The decisive parameter for optimizing DTR is evaluated as the range of each tree leaf. Additionally, surrogate decision splits and maximum surrogates per node were included in this evaluation process.

BTR requires four parameters Minimum leaf size, number of learners, number of predictors to sample and log scaled learning rate to be varied during optimization.

In the SVMR optimization process, 4 kernel functions were examined, which are Gaussian, Linear, Quadratic, and Cubic. The scales of these functions, which are log scaled, were also evaluated. Moreover, box constraint and epsilon (ϵ) values were considered.

In GPR optimization, 3 basis functions (Zero, Constant, and Linear) and 10 kernel functions were cycled (Non-isotropic Rational Quadratic, Isotropic Rational Quadratic, Non-isotropic Squared Exponential, Isotropic Squared Exponential, Non-isotropic Matern (Genton,2001) 5/2, Isotropic Matern 5/2, Non-isotropic Matern 3/2, Isotropic Matern 3/2, Non-isotropic Exponential, Isotropic Exponential).

ANN optimization involves the decision of how many hidden layers there should be and how many neurons each layer should have. Although there are techniques, such as using Genetic Algorithm to find the optimal parameter combination [15], trial and error technique is preferred in this study for its simplicity. It must be noted that the number of input neurons could be more than the number of inputs (i.e., 4) to the problem, since the exact number of inputs are increased after applying an encoding technique. Since, the input size is not same for all encoding techniques, the number of input neurons of the ANN varies depending on the encoding technique used. To conclude, to optimize the ANN structure, the workflow proposed in Figure 3 is presented. This procedure basically describes that layer size and neuron size (i.e., the number of neurons in a layer) are increased one by one until the best RMSE value is achieved. If, in any stage, adding a new neuron starts to worsen the result with a

tolerance of 10 additions, no new neuron is added anymore and the number of neurons that gave the best RMSE so far is accepted for that layer. If the inclusion of that layer has improved the result, then the procedure continues by adding another layer. Otherwise, the optimization stops and accepts the layer-neuron combination that gave the lowest RMSE value so far.

6. RESULTS AND DISCUSSION

The results of the ANN optimization procedure in Figure 3 are presented in Table 3, where rows represent the encoding techniques, and the columns correspond to the structure of the ANN. These results were obtained for each encoding technique within the scope of the proposed framework in Figure 1. Examining Table 3, the best result was obtained using binary encoding technique on an ANN with 2 hidden layers. The layers are consisted of 206 and 95 neurons, respectively. Since binary encoding produced 23 features, this ANN architecture has 23 input neurons. In general, the increase in the number of layers in ANN architecture had a positive effect on the result and having more than 3 hidden layers decreased the performance. On the other hand, the one hot encoding method quickly gave worse results with few neurons in the 2nd hidden layer. This can be caused by the fact that this technique generated the most features (i.e., 189) which increased the dimensionality of data and make it harder to generalize.

TABLE 3. ANN optimization results.

Encoding Method	Layer - 1		Layer - 2		Layer - 3		Layer - 4	
	#Neurons	RMSE	#Neurons	RMSE	#Neurons	RMSE	#Neurons	RMSE
Ordinal	337	8.672	35	8.507	17	8.493 ¹	60	8.534 ²
Binary	206	7.523	95	7.184 ³	87	7.307 ²	-	-
One Hot	4	8.052 ¹	11	8.226 ²	-	-	-	-
Dummy	202	7.999	152	7.972 ¹	16	8.058 ²	-	-
Effect	210	8.014 ¹	68	8.106 ²	-	-	-	-
Frequency	168	8.543	126	8.455	54	8.367 ¹	33	8.429 ²
Mean	119	8.170	323	7.974 ¹	330	8.239 ²	-	-

¹ Best result for a particular encoding method.

² Optimization got worse so stopped.

³ Optimal encoding and ANN architecture.

TABLE 4. Optimized results of all machine learning models.

Machine Learning Model	RMSE
Linear Regression	8.764
Decision Tree Regression	8.827
Support Vector Regression (Quadratic Kernel)	8.812
Boosted Trees (Ensemble)	8.571
Gaussian Process Regression	8.401
Artificial Neural Network	7.184

Optimized results of all machine learning models are presented in Table 4, where optimal value of each parameter is given in Table 5. Examining Table 4 together with Table 3, the results of all models were slightly improved after their parameter optimization. Despite those improvements, the optimized ANN architecture unquestionably obtained the best overall result. An interesting result in Table 4 shows that even a simple model like Linear Regression can achieve comparable results with other models. This might originate from the low representation capability of the current parameters. Unarguably, inclusion of additional features will represent the data better, thus more accurate results could be obtained.

TABLE 5. Optimal parameter values for each machine learning model.

DTR	SVR	BTR	GPR
min leaf size 24	kernel function quadratic	min leaf size 11	basis function constant
surrogate decision split off		number of learners 428	kernel function isotropic squared exponential
		number of predictors to sample 8	
		learning rate adaptive	

The business intelligence system of the organization that owns the data set performs project follow-ups daily. The tool design schedule is also included in this business intelligence system. When looking at the process on a long scale, for instance, in tool design project planning, it is preferred to use "day" units instead of "hour" units; and in some cases, "month" units in efforts calculation since the project calendars are usually expressed in years. Therefore, missing the correct design time by 10-15 hours is in fact not a bad estimation since the estimation error is still less

than 1 day. In other words, if design times were given as days instead of hours, RMSE values would be smaller than the ones in Table 4. For this reason, model performances should also be assessed in an alternative way. To that end, following rule of thumb is used: in aviation industry, estimates can be considered as correct if the time allocated to tool design engineering does not exceed 20% of the total duration of the project. Table 6 presents the estimation accuracy of each method when such an evaluation is done, where an estimation is assumed to be correct if its difference to actual design time do not exceed 20% of the actual design time. As a result, it is observed that ANN proved superior to other ML models by correctly estimating the 85.64% of the total data.

TABLE 6. Estimation accuracies of all machine learning models.

Machine Learning Model	Accuracy
Linear Regression	0.7704
Decision Tree Regression	0.7665
Support Vector Regression (Quadratic Kernel)	0.7525
Boosted Trees (Ensemble)	0.7890
Gaussian Process Regression	0.8049
Artificial Neural Network	0.8564

7. CONCLUSIONS

In this study, an optimized Artificial Neural Network architecture is proposed for the estimation of the design effort of production tools used in the aviation industry. This approach can also be adapted to many other design processes if requirements are well defined and prior effort data is available.

In the beginning of this study all machine learning methods applied on the dataset gave comparable results. However, ANN and binary encoding gave the most successful results due to the better representation capability of binary encoding and parameter optimization.

When the relationship between error rates and tool attributes are examined; in production projects where the product design was completed, known as build to print, the estimation performance has been found to be much better than the estimation of development projects. Comparing the relationship between the characteristic of the design and the error level; it has been seen that design studies involving generic activity are predicted with less error. For instance, preparing a periodic measurement document for a tool was much better predicted than designing

a tool from scratch. When the tool code and error rate were examined, no significant relationship was found between them. For this reason, it was evaluated that the main function of a tool alone was not decisive.

It is also thought that high dimensionality of the input data (i.e., more input neurons in input layer in ANN) might also degrade the performance of the proposed system. Therefore, dimension reduction techniques, such as Principal Component Analysis and Autoencoder methods, may also be considered as a future work to assess whether this is the case or not.

Author Contribution Statements Authors are equally contributed to the paper. All authors read and approved the final copy of the manuscript.

Declaration of Competing Interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement This study was supported by Turkish Aerospace Company.

REFERENCES

- [1] Benedetto, H., Vieira, D., Proposed framework for estimating effort in design projects, *Int. J. Manag. Proj. Bus.*, 11 (2) (2018), 257-274, <https://doi.org/10.1108/IJMPB-03-2017-0022>.
- [2] Martin, M. V., Ishii, K., Design for variety: a methodology for understanding the costs of product proliferation, *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 97607 (1996), <https://doi.org/10.1115/96-DETC/DTM-1610>.
- [3] Dentsoras, A. J., Information generation during design: information importance and design effort, *AI EDAM*, 19 (1) (2005), 19-32, <https://doi.org/10.1017/S089006040505002X>.
- [4] Bashir, H. A., Thomson, V., Estimating design complexity, *J. Eng. Des.*, 10 (3) (1999), 247-257, <https://doi.org/10.1080/095448299261317>.
- [5] Bashir, H. A., Thomson, V., Estimating design effort for GE hydro projects, *Comput. Ind. Eng.*, 46 (2) (2004), 195-204, <https://doi.org/10.1016/j.cie.2003.12.005>.
- [6] Bashir, H. A., Modeling of development time for hydroelectric generators using factor and multiple regression analyses, *Int. J. Proj. Manag.*, 26 (4) (2008), 457-464, <https://doi.org/10.1016/j.ijproman.2007.08.006>.
- [7] Bashir, H. A., Thomson, V., An analogy-based model for estimating design effort. *Des. Stud.*, 22 (2) (2001), 157-167, [https://doi.org/10.1016/S0142-694X\(00\)00015-6](https://doi.org/10.1016/S0142-694X(00)00015-6).
- [8] Ou-Yang, C., Lin, T., Developing an integrated framework for feature-based early

- manufacturing cost estimation, *Int. J. Adv. Manuf. Technol.*, 13 (9) (1997), 618-629, <https://doi.org/10.1007/BF01350820>.
- [9] Wang, H., Li, H., Wen, X., Luo, G., Unified modeling for digital twin of a knowledge-based system design, *Robot. Comput. Integr. Manuf.*, 68 (2021), 102074, <https://doi.org/10.1016/j.rcim.2020.102074>.
- [10] Thomson, A., Hird, A., Estimating design effort needs of product design projects using captured expert knowledge—a proposed method, *Proceedings of the Design Society*, 1 (2021), 1391-1400, <https://doi.org/10.1017/pds.2021.139>.
- [11] Sehra, S. K., Brar, Y. S., Kaur, N., Sehra, S. S., Research patterns and trends in software effort estimation, *Inf. Softw. Technol.*, 91 (2017), 1-21, <https://doi.org/10.1016/j.infsof.2017.06.002>.
- [12] Aktan, U., Dikmen, M., Estimation of design effort of jigs and fixtures used in aviation industry by machine learning methods, *2nd International Eurasian Conference on Science, Engineering and Technology*, (2020), 500-505.
- [13] Genton, M. G., Classes of kernels for machine learning: a statistics perspective, *J. Mach. Learn. Res.*, 2 (2001), 299-312.
- [14] Levenberg, K., A method for the solution of certain non-linear problems in least squares, *Q. Appl. Math.*, 2 (2) (1944), 164-168.
- [15] Soylu, K., Güzel, M. S., Emek Soylu B., Bostanci, G. E., Genetic hyperparameter optimization library development and its application on plant disease detection problem, *28th Signal Processing and Communications Applications Conference (SIU)*, (2020), 1-4, <https://doi.org/10.1109/SIU49456.2020.9302246>.

ON THE GEOMETRIC PHASES IN ENTANGLED STATES

Melik Emirhan TUNALIOĞLU¹, Hasan Ozgur CILDIROĞLU¹, Ali Ulvi YILMAZER¹

¹Department of Physics Engineering, Ankara University, Ankara, TÜRKİYE


ABSTRACT. Correlation relations for the spin measurements on a pair of entangled particles scattered by the two separate arms of interferometers in hybrid setups of different types are investigated. Concurrence, entanglement of formation, quantum fidelity, Bures distance are used to clarify how the geometric phase affects the initial bipartite state. This affect causes a quantum interference due to the movement of charged particles in regions where electromagnetic fields are not present. We shown that in some cases the geometric phase information is carried over to the final bipartite entangled state.


1. INTRODUCTION

In classical and quantum physics, the geometric phase is a unifying and innovative concept, and it appears as an additional phase factor due to the geometric and topological properties of the Hamiltonian parameter space for various driven systems. The geometric phase was first studied by Ehrenberg [1], Kato [2], Pancharatnam [3], Longuet-Higgins et al. [4], Aharonov and Bohm [5] and later generalized by Berry in 1984 [6]. Geometric and topological phases play a considerable role in physics and a vast literature exist today covering many plausible applications in different fields [7–16]. On the other hand, entanglement is another striking aspect of quantum mechanics and its dates back to the article published by Einstein et.al. [17] and its deep meaning was exhibited later by Bell [18].

The study of topological phases for entangled states has gained notable interest recently and it has been considered as a useful resource to analyze the quantum information processing. From this perspective, in this work, quantum correlations are briefly discussed for entangled particles in hybrid Aharonov-Bohm (AB), Aharonov-Casher (AC), He-McKellar-Wilkens (HMW), Dual Aharonov-Bohm (DAB) type Einstein, Podolsky and Rosen (EPR) setups. In this study, Clauser-Horne-Shimony-Holt (CHSH) inequality [19] for certain choices of spin measurements is discussed

Keywords. Geometric phase, CHSH inequality, concurrence, entanglement of formation, fidelity, Bures distance.

✉ tunalioglu@ankara.edu.tr-Corresponding author;  0000-0002-1143-9516

✉ cildiroglu@ankara.edu.tr;  0000-0003-0246-1102

✉ ali.u.yilmazer@eng.ankara.edu.tr;  0000-0002-2270-8486.

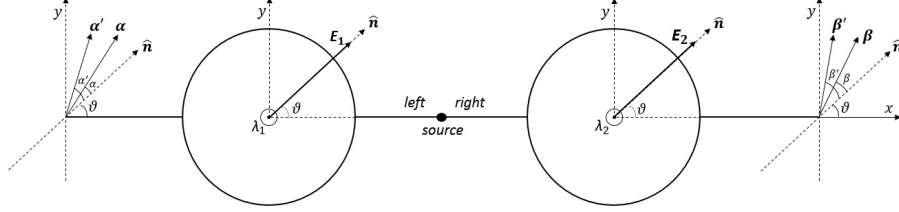


FIGURE 1. A hybrid AC-EPR setup. To investigate the result of the measurement of the spin components of the particles going left and right, one can define projection operators along arbitrary vectors α , α' , β and β' . Along these vectors the expectation values of joint measurements necessary for the CHSH inequality are calculated.

in a hybrid AC-EPR setup and then various entanglement measures such as concurrence (\mathcal{C}), entanglement of formation (EoF) are examined [20–23] in connection with the quantum states acquiring geometric and topological phases. At the same time, in order to better understand the possible role of geometric phase as an indicator of the entanglement concepts such as quantum fidelity (\mathcal{F}) and Bures distance (\mathcal{D}_B) are utilized.

2. AN EXAMPLARY MODEL: ENTANGLED STATE IN A HYBRID AC-EPR SETUP

In order to investigate the quantum spin correlations related to the Aharonov-Casher effect performed with an entangled spin pairs a hybrid AC-EPR experimental setup with two electric charge lines (λ_1, λ_2) provides a useful and practical configuration (Fig. 1).

Consider a pair of entangled spin- $\frac{1}{2}$ neutral particles produced in a spin singlet state by the source at the center:

$$|\psi(t)\rangle = \frac{1}{\sqrt{2}}[|\uparrow\rangle_L \otimes |\downarrow\rangle_R - |\downarrow\rangle_L \otimes |\uparrow\rangle_R]. \quad (1)$$

Thus we are considering a *gedanken* experiment with two magnetic dipoles (neutrons) from a single source, one to the right, one to the left, and are under the influence of two electric lines.

Our goal is to discuss the possible results of the spin measurements in various directions and to find the correlation functions associated with the joint probabilities; and furthermore we will investigate the CHSH violation, calculate the concurrence (\mathcal{C}), entanglement of formation (EoF) and quantum fidelity (\mathcal{F}), Bures distance (\mathcal{D}_B) step by step.

The initial singlet state in (1) will gain an AC phase as follows,

$$|\psi(t')\rangle = \frac{1}{\sqrt{2}}[e^{-i\mu\lambda_1}|\uparrow\rangle_L \otimes e^{i\mu\lambda_2}|\downarrow\rangle_R - e^{i\mu\lambda_1}|\downarrow\rangle_L \otimes e^{-i\mu\lambda_2}|\uparrow\rangle_R]. \quad (2)$$

This equation can be written more simply,

$$|\psi(t')\rangle = \frac{1}{\sqrt{2}}[e^{-i\mu(\lambda_1-\lambda_2)}|\uparrow\rangle_L|\downarrow\rangle_R - e^{i\mu(\lambda_1-\lambda_2)}|\downarrow\rangle_L|\uparrow\rangle_R]. \quad (3)$$

If one defines $\lambda_1 - \lambda_2 = \lambda_E$, then above equation becomes,

$$|\psi(t')\rangle = \frac{1}{\sqrt{2}}e^{-i\mu\lambda_E}[|\uparrow\rangle_L|\downarrow\rangle_R - e^{2i\mu\lambda_E}|\downarrow\rangle_L|\uparrow\rangle_R]. \quad (4)$$

The total phase factor can obviously be removed as it will not be included in the expectation value calculation, and the final spin wave function will be the following,

$$|\psi(t')\rangle = \frac{1}{\sqrt{2}}[|\uparrow\rangle_L|\downarrow\rangle_R - e^{2i\mu\lambda_E}|\downarrow\rangle_L|\uparrow\rangle_R]. \quad (5)$$

The quantum correlations can be examined by measuring the spin components of the particles moving to the right and left, along the directions as in the usual Bell tests. The correlation function (S) for the CHSH inequality associated with the angles $(\alpha, \beta, \alpha', \beta')$ for the spin measurements shown in Fig. 1 can be found to be as:

$$\begin{aligned} S(\alpha, \beta, \alpha', \beta') &= |E(\alpha, \beta) - E(\alpha, \beta')| + |E(\alpha', \beta) + E(\alpha', \beta')| \\ &= |-\cos \alpha \cos \beta - \sin \alpha \sin \beta \cos(2\mu\lambda_E) \\ &\quad + \cos \alpha \cos \alpha' + \sin \alpha \sin \alpha' \cos(2\mu\lambda_E)| \\ &\quad + |-\cos \alpha' \cos \beta' - \sin \alpha' \sin \beta' \cos(2\mu\lambda_E) \\ &\quad + \cos \beta \cos \beta' + \sin \beta \sin \beta' \cos(2\mu\lambda_E)| \leq 2. \end{aligned} \quad (6)$$

Here $E(\alpha, \beta)$'s are expectation values of the joint spin measurements defined as, $E(\alpha, \beta) = |\langle \psi(t') [P_+^l(\alpha) - P_-^l(\alpha)] \otimes [P_+^r(\beta) - P_-^r(\beta)] | \psi(t') \rangle|$ and $P_{\pm}^l(\alpha)$ being the spin projection operators along the angles $\pm(\alpha)$, etc. [24].

It is easily seen that the CHSH inequality is violated by quantum mechanics, if the appropriate angles are chosen for the S function. At this point, one notes that Bell-type inequalities have a generalized upper violation limit called Tsirelson limit ($|\langle E \rangle|_{QM\text{correlations}} \leq 2\sqrt{2}$) [25], and for appropriate choices of relevant angles below, one obtains:

$$S(0, \frac{\pi}{4}, \frac{3\pi}{4}, \frac{\pi}{2}) = \sqrt{2} + \sqrt{2}|\cos(2\mu\lambda_E)|. \quad (7)$$

This result reveals the clear dependence of the CHSH correlation function S on the geometric AC phase $2\mu\lambda_E$.

3. RELATED ENTANGLEMENT MEASURES

Now in order to discuss the entanglement measures in connection with the Bell-type correlations for the entangled states we continue to examine the above mentioned hybrid AC-EPR setup with two electric charge lines. Firstly, let us calculate concurrence. One needs to write a more suitable version of the wave function, namely $|\tilde{\psi}(t')\rangle$, according to the definition given in [21],

$$|\tilde{\psi}(t')\rangle = \sigma_y \otimes \sigma_y |\psi^*\rangle = \frac{1}{\sqrt{2}} [(|\downarrow\uparrow\rangle - e^{-2i\mu\lambda_E} |\uparrow\downarrow\rangle)]. \quad (8)$$

Pauli spin matrices here will spin flip the complex conjugate of the wave function, thus the concurrence becomes,

$$\mathcal{C}(|\psi(t')\rangle) = \left| \langle \psi(t') | \tilde{\psi}(t') \rangle \right| = \frac{1}{2} |(\langle \uparrow\downarrow | - e^{-2i\mu\lambda_E} \langle \downarrow\uparrow |)(|\downarrow\uparrow\rangle - e^{-2i\mu\lambda_E} |\uparrow\downarrow\rangle)|. \quad (9)$$

If we arrange the above result, and performs simple inner products the concurrence is found as

$$\mathcal{C}(|\psi(t')\rangle) = |-e^{-2i\mu\lambda_E}| = 1. \quad (10)$$

Thus, the entanglement of formation, EoF , is defined via the concurrence value and the function $h(x) = -x \log_2 x - (1-x) \log_2 (1-x)$ [21]. If we use the above result in the EoF calculation (10) we get

$$EoF = h\left(\frac{1 + \sqrt{1 - \mathcal{C}^2(\rho)}}{2}\right) = h\left(\frac{1}{2}\right) = 1. \quad (11)$$

In the case when both states ($|\psi(t)\rangle$ and $|\psi(t')\rangle$) are pure states then quantum fidelity can be calculated as an overlap of states,

$$\mathcal{F} = |\langle \psi(t) | \psi(t') \rangle| = \frac{1}{2} |\langle \uparrow\downarrow | \uparrow\downarrow \rangle - e^{2i\mu\lambda_E} \langle \uparrow\downarrow | \downarrow\uparrow \rangle - \langle \downarrow\uparrow | \uparrow\downarrow \rangle + e^{2i\mu\lambda_E} \langle \downarrow\uparrow | \downarrow\uparrow \rangle|. \quad (12)$$

Thus quantum fidelity for a process of hybrid AC-EPR scattering with spin entangled initial state turns out to be

$$\mathcal{F} = \frac{1}{2} |1 + e^{2i\mu\lambda_E}| = |\cos(\mu\lambda_E)|. \quad (13)$$

Furthermore, the Bures distance is given as $\mathcal{D}_B = \sqrt{2(1 - \mathcal{F})}$ [26], so \mathcal{D}_B between initial and final entangled states in a hybrid AC-EPR setup can be found as

$$\mathcal{D}_B(AC) = \sqrt{2 - 2|\cos(\mu\lambda_E)|}. \quad (14)$$

The state in (1) is a pure entangled state of two qubits, so by Gisin's theorem [27], it violates the CHSH inequality and our result in (7) confirms that prediction. Also by the equations (10), (11) and (13); one observes that the geometric phase has no effects on the concurrence and entanglement of formation, but on the contrary it has an explicit appearance in the quantum fidelity expression and the latter ranges between $0 \leq \mathcal{F} \leq 1$. Also $|\psi(t)\rangle$ and $|\psi(t')\rangle$ which are connected via an AC phase

TABLE 1. Summary of Aharonov-Bohm (AB), Aharonov-Casher (AC), He-McKellar-Wilkens (HMW), Berry and Dual Aharonov-Bohm (DAB) phases gained by entangled states and the values of Concurrence (\mathcal{C}), Entanglement of Formation (\mathcal{EoF}), Quantum Fidelity (\mathcal{F}), Bures Distance (\mathcal{D}_B). Here μ is magnetic dipole, λ_E is electric charge density, d is electric dipole, λ_B is magnetic charge density, γ is Berry phase and g is magnetic charge.

	$ \psi(t)\rangle$	$ \psi(t')\rangle$	\mathcal{C}	\mathcal{EoF}	\mathcal{F}	\mathcal{D}_B
AB	(1)	$\frac{1}{\sqrt{2}}e^{-i\phi_B}[\uparrow\rangle_L \downarrow\rangle_R - \downarrow\rangle_L \uparrow\rangle_R]$	1	1	1	0
AC	(1)	$\frac{1}{\sqrt{2}}[\uparrow\rangle_L \downarrow\rangle_R - e^{2i\mu\lambda_E} \downarrow\rangle_L \uparrow\rangle_R]$	1	1	$ \cos(\mu\lambda_E) $	$\sqrt{2 - 2\mathcal{F}}$
HMW	(1)	$\frac{1}{\sqrt{2}}[\uparrow\rangle_L \downarrow\rangle_R - e^{2id\lambda_B} \downarrow\rangle_L \uparrow\rangle_R]$	1	1	$ \cos(d\lambda_B) $	$\sqrt{2 - 2\mathcal{F}}$
Berry	(1)	$\frac{1}{\sqrt{2}}[\uparrow\rangle_L \downarrow\rangle_R - e^{2i\gamma} \downarrow\rangle_L \uparrow\rangle_R]$	1	1	$ \cos(\gamma) $	$\sqrt{2 - 2\mathcal{F}}$
DAB	(1)	$\frac{1}{\sqrt{2}}e^{-ig\phi_E}[\uparrow\rangle_L \downarrow\rangle_R - \downarrow\rangle_L \uparrow\rangle_R]$	1	1	1	0

as in (5) and the non-zero Bures distance between them is given by the equation (14).

In this context, AB, HMW, DAB type experimental hybrid setups can be designed similar to the hybrid AC-EPR setup and analogous calculations can be repeated for them. Thus, the summarized results, including a generalized version of the geometric Berry phase, can be obtained as written in Table 1. Also Fig. 2 tells us that the geometric phase information is preserved (observed) for the parameter values of the white areas but not preserved (seen) in the dark areas. In this plot exemplary μ , d , λ_E and λ_B are the relevant physical variables having certain values within some exemplary intervals chosen as $(\mu, d = [0, 4])$ and $\lambda_E, \lambda_B = [-4, 4])$ and each product $\mu\lambda_E$ and $d\lambda_B$ is an angle. It is to be noted that; since there is no geometric phase information in AB and DAB type hybrid experimental setups and neither in the Berry case, no figures are needed for them.

Quantum computers are analog machines, which are expected to become a part of our lives in the near future. Although the codes to be run in the quantum computer are applied on various quantum gates, they are actually represented by real variables. Applying a quantum gate to a qubit means subjecting it to a wave from an indiscriminate wave generator. All properties of this wave, such as amplitude and frequency, are real-valued variables. Errors may occur from time to time in these variables. These errors are different from those caused by the interaction of the qubit with the environment. Even if a required quantum gate is prepared and applied perfectly, however exactly the desired quantum state may not be achieved.

In fact, this is where quantum fidelity comes into play. Fidelity is a link between regular digital needs and quantum hardware. A quantum state will be less entangled the closer it is to the set of separable states, or more entangled the farther it is [28]. With fidelity calculations, one finds a way to track how well real circuits produced by a quantum computer fit the original state. Mathematically, it provides guidance

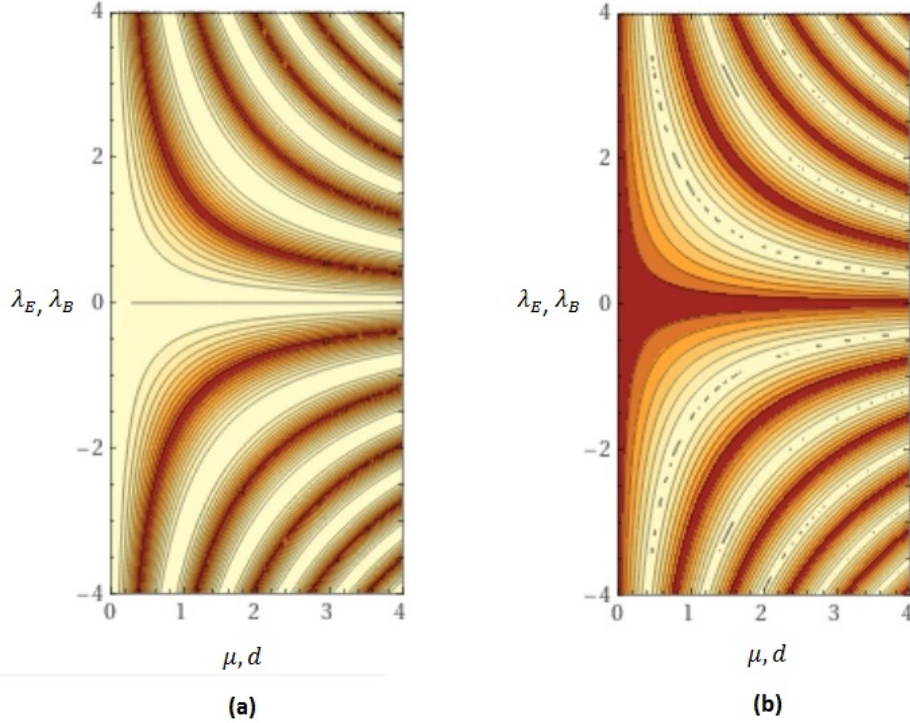


FIGURE 2. Plots of (a) Fidelity and (b) Bures distance for AC-EPR and HMW-EPR Hybrid Setups, respectively. The white areas are the regions where the absolute value of the fidelity expression given for AC ($|\cos(\mu\lambda_E)|$) and HMW setups ($|\cos(d\lambda_B)|$) in Table 1 attains unity, and in the dark areas are this absolute value becomes zero. Similarly, as to the Bures distance expression given for AC ($\sqrt{2-2|\cos(\mu\lambda_E)|}$) and HMW ($\sqrt{2-2|\cos(d\lambda_B)|}$), the region where the absolute value reaches unity is the dark areas, and conversely the white areas are the regions where the absolute value is zero. As a result, if the fidelity value is 1, the Bures distance value is 0 as expected.

on the degree between quantum states [29], and there are several benefits for making such a comparison. As a result fidelity has become one of the most widely used quantities to measure the degree of similarity between quantum states.

Besides, Bures distance describes the infinitesimal distance between the density matrices describing the quantum states, and this approach is a generalization of the Fisher distance. If it is limited to pure states only, it gives similar results as the Fubini–Study distance [30–34]. These distances can be used as a function of distance, metric. The Fisher metric is a Riemann metric that can be defined on

a smooth statistical manifold. It is generally used to calculate the difference in information between measurements.

In this regard, statistical distances such as the Bures distance are determined by the size of the statistical fluctuations that occur in the measurements made to distinguish between the initial state and the evolved state. These statistical fluctuations raise the interesting possibility that quantum mechanics may be partially responsible for the Hilbert-space structure, so Bures distance is thought to be a link between statistics and geometry. By counting the number of intermediate states, the distance between states can be found [35].

4. CONCLUSION

Summary Table 1 and Fig. 2 shows us that, AB and DAB phase information does not appear in the results of the measurements in the experimental setups to be made with entangled particles. However, the situation is different for AC, HMW and Berry phases. For these processes, geometric phase information appears in the joint spin measurements. Thus, it seems possible in the future to code an information via the geometric phase relevant in the process. It is understood that investigation of the entanglement properties by the geometric phase will provide a better understanding of the nature of entanglement and its role in quantum technologies [36].

Nevertheless, quantum logic gate operations can also be implemented with geometric phase. When a quantum system is in a cyclic evolution, it acquires a geometric phase determined by the path in which the system moves. Geometric phases are useful for combatting errors in quantum gate operations and can help quantum error correction codes reach below the error threshold. There are many studies examining how geometric phases can be used to better analyze entangled systems and to process the quantum information [37–39].

Since the geometric phase is not dependent on time and energy like the dynamic phase, but is connected to the closed path, it is not affected by changes such as noise distortions. Therefore, it can also be used for quantum logic gate applications [40, 41]. Obviously one of the main questions here is the entanglement content of the measured state, so after a physical process providing a geometric phase the different measures of entanglement are calculated for this purpose.

Otherwise, as is well known interferometers of different types, such as Mach–Zehnder, Hanbury Brown and Twiss, etc. are effectively used to study counter intuitive predictions of quantum mechanics [42–44]. In such works, one observes that the probability calculations based on the correlations between the number of particle counts at detectors have interesting similarities with the probabilities via spin measurements of the particles in certain directions as used in the present work.

Declaration of Competing Interests The authors did not receive support from any organization for the submitted work. The authors have no relevant financial

or non-financial interests to disclose.

Acknowledgement We thank Abdullah Vercin for helpful comments.

REFERENCES

- [1] Ehrenberg, W., Siday, R. E., The refractive index in electron optics and the principles of dynamics, *Proceedings of the Physical Society*, Section B 62 (1) (1949), 8–21, <https://doi.org/10.1088/0370-1301/62/1/303>.
- [2] Kato, T., On the adiabatic theorem of quantum mechanics. *J. Phys. Soc. Jpn.*, 5 (6) (1950), 435–439, <https://doi.org/10.1143/JPSJ.5.435>.
- [3] Pancharatnam, S., Generalized theory of interference, and its applications, *Proceedings of the Indian Academy of Sciences*, Section A 44 (5) (1956), 247–262, <https://doi.org/10.1007/BF03046050>.
- [4] Longuet-Higgins, H. C., Opik, U., Pryce, M. H. L., Sack, R., Studies of the Jahn-Teller effect .II. The dynamical problem, *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 244 (1236) (1958), 1–16, <https://doi.org/10.1098/rspa.1958.0022>.
- [5] Aharonov, Y., Bohm, D.: Significance of Electromagnetic Potentials in the Quantum Theory, *Phys. Rev.*, 115 (3) (1959), 485, <https://doi.org/10.1103/PhysRev.115.485>.
- [6] Berry, M. V., Quantal phase factors accompanying adiabatic changes, *Proceedings of the Royal Society of London A. Mathematical and Physical Sciences*, 392 (1802) (1984), 45–57.
- [7] Aharonov, Y., Casher, A., Topological quantum effects for neutral particles, *Phys. Rev. Lett.*, 53 (4) (1984), 319, <https://doi.org/10.1103/PhysRevLett.53.319>.
- [8] He, X. -G., McKellar, B. H. J., Topological phase due to electric dipole moment and magnetic monopole interaction, *Phys. Rev. A*, 47 (1993), 3424–3425, <https://doi.org/10.1103/PhysRevA.47.3424>.
- [9] Wilkens, M., Quantum phase of a moving dipole, *Phys. Rev. Lett.*, 72 (1994), 5–8, <https://doi.org/10.1103/PhysRevLett.72.5>.
- [10] Dowling, J. P., Williams, C. P., Franson, J. D., Maxwell Duality, Lorentz invariance, and topological phase, *Phys. Rev. Lett.*, 83 (1999), 2486–2489, <https://doi.org/10.1103/PhysRevLett.83.2486>.
- [11] Sponar, S., Klepp, J., Loidl, R., Filipp, S., Durstberger-Rennhofer, K., Bertlmann, R., Badurek, G., Rauch, H., Hasegawa, Y., Geometric phase in entangled systems: A single-neutron interferometer experiment, *Phys. Rev. A*, 81 (4) (2010), 042113, <https://doi.org/10.1103/PhysRevA.81.042113>.
- [12] Sponar, S., Klepp, J., Durstberger-Rennhofer, K., Loidl, R., Filipp, S., Lettner, M., Bertlmann, R., Badurek, G., Rauch, H., Hasegawa, Y., New aspects of geometric phases in experiments with polarized neutrons, *J. Phys. A Math. Theor.*, 43 (35) (2010), 354015, <https://doi.org/10.1088/1751-8113/43/35/354015>.
- [13] Lepoutre, S., Gauguier, A., Trenec, G., Buchner, M., Vigue, J., He-McKellar-Wilkens topological phase in atom interferometry, *Phys. Rev. Lett.*, 109 (12) (2012), 120404, <https://doi.org/10.1103/PhysRevLett.109.120404>.
- [14] Werner, S., Observation of Berry’s geometric phase by neutron interferometry, *Found. Phys.*, 42 (1) (2012), 122–139, <https://doi.org/10.1007/s10701-010-9526-z>.
- [15] Gillot, J., Lepoutre, S., Gauguier, A., Buchner, M., Vigue, J., Measurement of the He-McKellar-Wilkens topological phase by atom interferometry and test of its independence with atom velocity, *Phys. Rev. Lett.*, 111 (3) (2013), 030401, <https://doi.org/10.1103/PhysRevLett.111.030401>.
- [16] Cohen, E., Larocque, H., Bouchard, F., Nejadattari, F., Gefen, Y., Karimi, E., Geometric phase from Aharonov–Bohm to Pancharatnam–Berry and beyond, *Nat. Rev. Phys.*, 1 (7) (2019), 437–449, <https://doi.org/10.1038/s42254-019-0071-1>.

- [17] Einstein, A., Podolsky, B., Rosen, N., Can quantum-mechanical description of physical reality be considered complete?, *Phys. Rev.*, 47 (10) (1935), 777, <https://doi.org/10.1103/PhysRev.47.777>.
- [18] Bell, J. S., On the Einstein Podolsky Rosen paradox, *Phys. Phys. Fiz.*, 1 (3) (1964), 195, <https://doi.org/10.1103/PhysicsPhysiqueFizika.1.195>.
- [19] Clauser, J. F., Horne, M. A., Shimony, A., Holt, R. A., Proposed experiment to test local hidden-variable theories, *Phys. Rev. Lett.*, 23 (15) (1969), 880, <https://doi.org/10.1103/PhysRevLett.23.880>.
- [20] Bennett, C. H., DiVincenzo, D. P., Smolin, J. A., Wootters, W. K., Mixed-state entanglement and quantum error correction, *Phys. Rev. A*, 54 (5) (1996), 3824, <https://doi.org/10.1103/PhysRevA.54.3824>.
- [21] Wootters, W. K., Entanglement of formation of an arbitrary state of two qubits, *Phys. Rev. Lett.*, 80 (10) (1998), 2245, <https://doi.org/10.1103/PhysRevLett.80.2245>.
- [22] Werner, R. F., Entanglement Measures, *Encyclopedia of Mathematical Physics*, (2006), 233–236.
- [23] Plenio, M. B., Virmani, S., An Introduction to Entanglement Measures, *Quantum Info. Comput.*, 7 (1) (2007), 1–51, <https://doi.org/10.48550/arXiv.quant-ph/0504163>.
- [24] Cildiroglu, H. O., Yilmazer, A. U., Investigation of the Aharonov-Bohm and Aharonov-Casher topological phases for quantum entangled states, *Phys. Lett. A*, 420 (2021), 127753, <https://doi.org/10.1016/j.physleta.2021.127753>.
- [25] Cirel'son, B. S., Quantum generalizations of Bell's inequality, *Lett. Math. Phys.*, 4 (2) (1980), 93–100.
- [26] Yuan, H., Fung, C.-H. F., Quantum parameter estimation with general dynamics, *npj Quantum Information*, 3 (14) (2017), 1–6.
- [27] Gisin, N.: Bell's inequality holds for all non-product states, *Phys. Lett. A*, 154 (5) (1991), 201–202, [https://doi.org/10.1016/0375-9601\(91\)90805-I](https://doi.org/10.1016/0375-9601(91)90805-I).
- [28] Liang, Y.-C., Yeh, Y.-H., Mendonca, P. E., Teh, R. Y., Reid, M. D., Drummond, P. D., Quantum fidelity measures for mixed states, *Rep. Prog. Phys.*, 82 (7) (2019), 076001, <https://doi.org/10.1088/1361-6633/ab1ca4>.
- [29] Kozłowski, W., Wehner, S., Meter, R.V., Rijsman, B., Cacciapuoti, A. S., Caleffi, M., Nagayama, S., Architectural Principles for a Quantum Internet, *Internet Engineering Task Force Work in Progress* (2023), RFC 9340.
- [30] Jozsa, R., Fidelity for Mixed Quantum States, *J. Mod. Opt.*, 41 (12) (1994), 2315–2323, <https://doi.org/10.1080/09500349414552171>.
- [31] Hubner, M., Explicit computation of the Bures distance for density matrices, *Phys. Lett. A*, 163 (4) (1992), 239–242, [https://doi.org/10.1016/0375-9601\(92\)91004-B](https://doi.org/10.1016/0375-9601(92)91004-B).
- [32] Bures, D., An extension of Kakutani's theorem on infinite product measures to the tensor product of semifinite w^* -algebras, *Trans. Am. Math. Soc.*, 135 (1969), 199–212.
- [33] Helstrom, C. W., Minimum mean-squared error of estimates in quantum statistics, *Phys. Lett. A*, 25 (2) (1967), 101–102, [https://doi.org/10.1016/0375-9601\(67\)90366-0](https://doi.org/10.1016/0375-9601(67)90366-0).
- [34] Facchi, P., Kulkarni, R., Man'ko, V., Marmo, G., Sudarshan, E., Ventriglia, F., Classical and quantum Fisher information in the geometrical formulation of quantum mechanics, *Phys. Lett. A*, 374 (48) (2010), 4801–4803, <https://doi.org/10.1016/j.physleta.2010.10.005>.
- [35] Wootters, W. K., Statistical distance and Hilbert space, *Phys. Rev. D*, 23 (2) (1981), 357, <https://doi.org/10.1103/PhysRevD.23.357>.
- [36] Sandhya, S., Banerjee, S., Geometric phase: an indicator of entanglement, *EPJ Plus D*, 66 (6) (2012), 168, <https://doi.org/10.1140/epjd/e2012-30211-5>.
- [37] Vedral, V., Geometric phases and topological quantum computation, *Int. J. Quantum Inf.*, 01 (01) (2003), 1–23, <https://doi.org/10.48550/arXiv.quant-ph/0212133>.
- [38] Sjoqvist, E., Geometric phases in quantum information, *Int. J. Quantum Chem.*, 115 (19) (2015), 1311–1326, <https://doi.org/10.1002/qua.2494>.

- [39] Thomas, J., Geometric phase in quantum computation, (2016). Doctoral Thesis, George Mason University, United States.
- [40] Song, C., Zheng, S. -B., Zhang, P., Xu, K., Zhang, L., Guo, Q., Liu, W., Xu, D., Deng, H., Huang, K., Zheng, D., Zhu, X., Wang, H., Continuous variable geometric phase and its manipulation for quantum computation in a superconducting circuit, *Nat. Commun.*, 8 (1) (2017), 1061, <https://doi.org/10.1016/j.msea.2012.06.074>.
- [41] Ji, L.- N., Ding, C.- Y., Chen, T., Xue, Z.- Y., Noncyclic geometric quantum gates with smooth paths via invariant-based shortcuts, *Adv. Quantum Technol.*, 4 (6) (2021), 2100019, <https://doi.org/10.1002/qute.202100019>.
- [42] Zeilinger, A., General properties of lossless beam splitters in interferometry, *Am. J. Phys.*, 49 (9) (1981), 882–883, <https://doi.org/10.1119/1.12387>.
- [43] Grangier, P., Roger, G., Aspect, A., Experimental evidence for a photon anticorrelation effect on a beam splitter: A new light on single-photon interferences, *EPL*, 1 (4) (1986), 173–179, <https://doi.org/10.1209/0295-5075/1/4/004>.
- [44] Silverman, M. P., Quantum Superposition, Springer, Berlin, Heidelberg, 2008, 111–135.

A NOVEL ALTERNATIVE IN WIRELESS AND PASSIVE SENSING: THE BENDED NESTED SPLIT-RING RESONATOR

Burak OZBEY¹



¹Department of Electrical and Electronics Engineering, Ankara University,
Ankara, TÜRKİYE

ABSTRACT. In this paper, a new split-ring resonator variant, called the bended nested split-ring resonator (B-NSRR) is introduced. B-NSRR is a modified version of the nested split-ring resonator (NSRR) geometry, which has been successfully utilized in sensing of various physical quantities such as strain, displacement and moisture content due to its superior sensitivity, resolution and compactness in comparison to more traditional structures such as SRR and electrical SRR (ESRR). The B-NSRR geometry is demonstrated to allow an even more compact structure, while retaining the high sensitivity level of the NSRR. The performances obtained by the SRR, ESRR, NSRR and B-NSRR geometries are compared for displacement and moisture content sensing applications. Simulations are carried out to validate the findings, where modified versions of SRR-based structures are employed as displacement sensors and a comparison is made between their performances. Owing to its compactness and high sensitivity, it is shown that the B-NSRR is a reasonable alternative to available geometries in various sensing applications.

1. INTRODUCTION

Metamaterials are periodic structures which can be used to generate exotic behavior that cannot be achieved by natural materials. The interesting phenomena that have been proposed and/or demonstrated include left-handed materials with negative effective permittivity and permeability [1–3], superlenses [4], zero-index ultradirective materials [5], artificial magnetic conductors [6] and electromagnetic cloaks [7]. The potential of metamaterials and frequency-selective surfaces (FSS) in diverse areas has also motivated researchers to apply these results to well-known engineering problems to come up with novel superior designs. Examples of metamaterial-based structures in antenna and microwave engineering include thin sub-wavelength cavity resonators [8], phase shifters which can produce positive, negative or 0 phase shifts while maintaining the same short overall length [9], compact stopband filters [10],

Keywords. Split-ring resonator (SRR), nested split-ring resonator (NSRR), wireless sensor, passive sensor, displacement sensor, moisture content sensor, microwave sensing.

 ozbey@ankara.edu.tr-Corresponding author;  0000-0001-7485-2132.

broadband baluns [11], electrically-small antennas with increased radiation [12], and many more. Metamaterials can also be employed to generate a strong localization and enhancement of electromagnetic fields, which in turn can be used to obtain ultrasensitive and high-resolution sensor designs for detection of many different stimuli [13]. Metamaterial-based sensors that have been proposed in literature include a long-list of works, among which biosensors [14], thin-film sensors [15], strain sensors [16] pressure sensors [17, 18] and temperature sensors [18] can be cited.

Since their theoretical introduction in [19], the building blocks of metamaterials have been the split-ring resonator (SRR). A single SRR is a small-loop antenna, which leads to negative magnetic polarization and effective permeability due to cancellation of the incident field and the out-of-phase locally-scattered field 20 when operated at a slightly higher frequency. When the structure is used as a unit cell in two dimensional repeating patterns, the resulting SRR array acts as a μ -negative metamaterial, where μ is the magnetic permeability. The single structure, on the other hand, is useful by itself as a sensor, because of its high field localization. The SRR can be visualized as an LC resonator, whose capacitance can be increased by introducing additional rings of different total metal length. A variant of SRRs, called the electric split-ring resonator (ESRR) was introduced in [21]. Contrary to the SRR, this structure produces little or no response to magnetic field illumination, but exhibits strong sub-wavelength resonance characteristics to the incident electric field. In addition to numerous other SRR variants, both the original SRR and ESRR have been widely used as a sensor in their single structure form, due to their high sensitivity. Despite having been very popular in sensing applications, these structures have drawbacks. The most important of these is that the SRR and the ESRR structures do not enable designs which are compact enough. Compactness is critical in applications such as biosensing, where in vivo measurement of vital signs requires small structures. In addition, many biosensing applications dictate a relatively low frequency in order to avoid the background absorption that takes place in the soft tissue. In some other sensing applications, most prominently for the radio frequency identification (RFID)-based tags in antenna form, increasing the operation frequency leads to increased loss and lower sensitivity [22]. Therefore, it is important to have a compact design which allows for both low frequency operation and high sensitivity. For this purpose, a new SRR variant, called the nested split-ring resonator (NSRR) was proposed in [23], where it was demonstrated by experiments that the NSRR enables a much more compact footprint compared to classic SRR, as well as a higher sensitivity. The NSRR geometry is shown in Figure 1(a). Originally intended as a biological strain sensor to evaluate the progression of long-bone fracture healing [23], the NSRR structure has also been exploited in sensing critical damage parameters such as displacement [24–28] and two-dimensional average strain [29, 30] in civil engineering structures, as well as in biological applications to sense the moisture content in plants [31]. The NSRR structure has proven

to be a useful alternative since it offers a non-destructive, wireless and passive method which also demonstrates a high sensitivity and resolution.

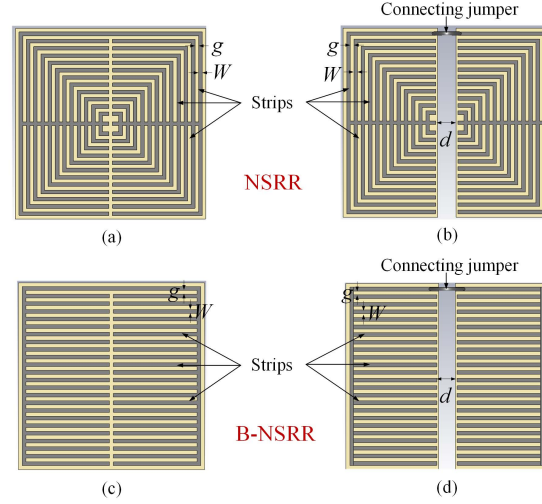


FIGURE 1. (a) Nested split-ring resonator (NSRR) geometry, (b) Modified NSRR geometry for displacement sensing, (c) Bended nested split-ring resonator (B-NSRR) geometry, (d) Modified B-NSRR geometry for displacement sensing.

In this work, we propose a novel NSRR-based sub-wavelength resonating structure, called the bended nested split-ring resonator (B-NSRR). The geometry of B-NSRR is shown in Figure 1(c). It includes a high number of thin coplanar metallic strips similar to the NSRR, however, the strips are not of the same size as in the NSRR, but of linearly decreasing length towards the center, making a 90° bend at the corners. This geometry increases the coplanar capacitance in comparison to the NSRR, while keeping the gap capacitance between opposing strips the same. This effect is especially pronounced when the number of strips is increased. Thus, the total capacitance is enhanced, which makes way for even a more compact design. As a variant of the NSRR, B-NSRR can be utilized in various sensing applications. Two of them, relative displacement sensing and moisture content sensing are demonstrated in this paper with full-wave electromagnetic simulations, where the change of the sensor resonance frequency is tracked versus monitored quantity. The sensing performances of the classic SRR, ESRR, NSRR and B-NSRR are compared for structures having approximately the same resonance frequency. The organization of this work is as follows: In Section 2, the B-NSRR geometry is discussed in terms of the NSRR and its equivalent circuit model. In Section 3, full-wave electromagnetic simulations of the four principal SRR geometries are presented for

wireless relative displacement and moisture content sensing. Section 4 concludes the paper.

2. THE B-NSRR GEOMETRY

2.1. NSRR Geometry and Its Equivalent Circuit Model. B-NSRR is a variant of the the NSRR geometry, which is shown in Figure 1(a). The NSRR consists of a number of opposing metallic strip pairs with a gap in between, which are aligned in vertical direction on a dielectric substrate. They are connected from the other end to a continuous uppermost strip, which makes the structure a combination of split-rings of different lengths which are connected in parallel. When the number of these “nested” split-rings, N , is relatively high, the capacitances coming from each split-ring is therefore added to yield a high capacitance value. The equivalent circuit model of the NSRR geometry is presented in Figure 2. The structure can be visualized as an LC resonator [32], where the inductance L_s is proportional to the length of each strip, while the capacitance has two forms: 1) The gap capacitance C_{gap} between each opposing strip, and 2) the capacitance between two parallel strips C_s [32]. The increase of both C_s and C_{gap} due to a high N in NSRR geometry increases the overall capacitance, which in turn leads to a decreased resonance frequency f_{res} , since $f_{res} = 1/\sqrt{L_{eff}C_{eff}}$, where L_{eff} and C_{eff} are the overall NSRR inductance and capacitance, respectively. For a given wavelength, increasing N thus allows for a more compact design in comparison to structures such as the SRR or the ESRR. The structure in this form can be used to measure physical quantities, which are correlated with electrical parameters of the medium. For instance, moisture content of the material on which the NSRR can be attached, or, the relative humidity of the medium can be measured, since variation of the water content in the material or in the air leads to a change in complex permittivity ϵ_r , which in turn affects C_{eff} . Likewise, mechanical changes on the NSRR structure also lead to a change in its electrical parameters. For example, strain induced on the NSRR is highly correlated with f_{res} , since it elongates or contracts the structure, which in turn changes C_{eff} and L_{eff} . Similarly, formation of a crack either on the medium behind the NSRR sensor or on the sensor itself, changes the electrical characteristics of the system. In order to measure relative displacement between two points, the NSRR structure can be split vertically into two symmetric halves, as shown in Figure 1(b). This way the halves become mechanically-independent. In order to maintain the electrical connection of the uppermost strip, the two halves are shorted by a thin jumper. This way, each NSRR half can be point-attached to a material under test, and if a displacement occurs between the two attachment points, this changes the gap d between the two structures. The variation of d is correlated with f_{res} , and by a calibration, it is possible to extract d wirelessly from the measurement of f_{res} [24]. It is possible to use either a one-channel measurement setup, where a transceiver antenna forms a coupled system with the NSRR, or, a two channel system where two antennas are

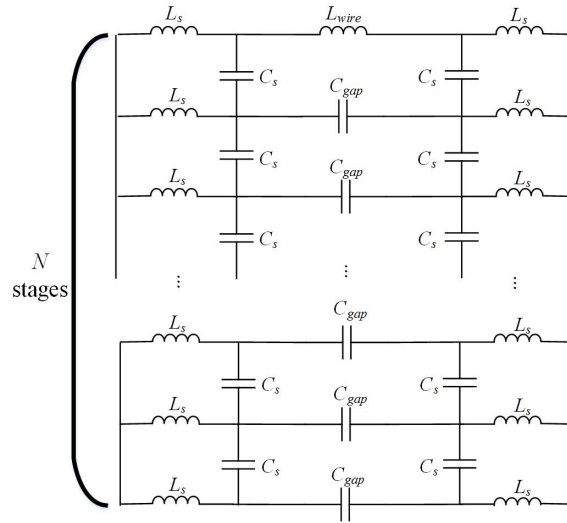


FIGURE 2. Equivalent circuit model of the NSRR geometry [32].

used where one of them sends a TX signal and the other receives the backscattered wave. In the first configuration, the NSRR resonance can be tracked directly from the spectrum of the transceiver antenna, but the interrogation distance is limited to near-field of the antenna [24, 27]. In the second configuration, it has been shown that the TX antenna can be taken to far field, but the RX antenna still has to be at a close distance [28] to maintain the high levels of sensitivity and resolution. In terms of the equivalent circuit model, the shorting wire brings an additional inductance, denoted as L_{wire} , which is in series with the uppermost strip as shown in Figure 2. The inclusion of the wire in the modified NSRR makes the geometry sensitive to magnetic field illumination. Normally a capacitance-dominated structure, only a horizontally polarized electric field is able to induce surface currents on the classic NSRR geometry. In the modified form, the jumper acts a magnetic loop to couple with incident magnetic fields, as well. How the resonance frequency shift occurs when the NSRR or the modified NSRR structure is employed as a sensor depends on the type of the sensing application. For example, for a strain or displacement sensor, the parameter d becomes subject to change, and therefore the dominating change occurs in the gap capacitance C_{gap} . On the other hand, for a moisture content sensor, an additional capacitance is brought in by the changing electrical properties of the medium.

2.2. Characteristics of the B-NSRR Geometry. Similar to the NSRR geometry, the proposed B-NSRR geometry also incorporates a high number of strips

to increase the overall capacitance. In essence, the field localization mechanism is very similar to the NSRR, since both geometries consist of nested split-rings which present parallel capacitances between the parallel and the opposing strips. Furthermore, the gap capacitance C_{gap} is also the same for both geometries, which is important in retaining the high sensitivity (as high 13 MHz/mm) and resolution ($\sim 1\mu\text{m}$) levels obtained in displacement sensing [24,28]. However, differently from the NSRR, the B-NSRR does not have a fixed strip length. Therefore, the strip inductance L_s and the parallel strip capacitance C_s do have a fixed value, but they change while moving from the center towards the top and the bottom of the structure. Here, the innermost strip produces the smallest L_s and C_s . On the other hand, the outermost strip produces the greatest L_s and C_s , which are around twice that are obtained by the NSRR structure, since the strip inductance and capacitance are linearly proportional to the length of the strip. Therefore, when the number of strips N is relatively small, the B-NSRR geometry is expected to generate a higher total capacitance and inductance. Here, the total parallel strip length is defined as the sum of each path length passing through the midsection of a neighboring parallel strip pair, as shown in Figure 3 with dashed lines.

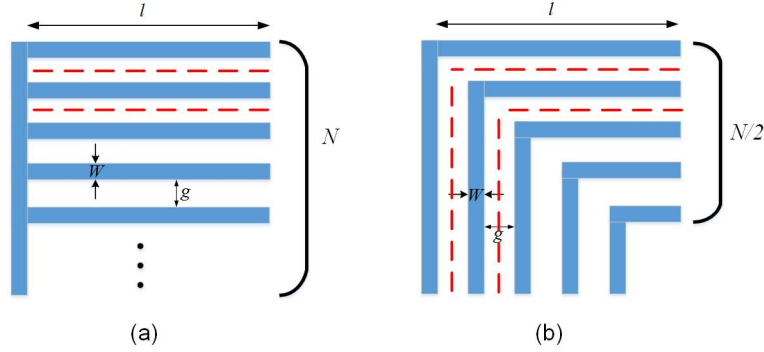


FIGURE 3. Calculation of the total parallel metallic strip length for (a) NSRR, and (b) B-NSRR.

The calculation of total parallel strip length for NSRR is straightforward. There are N parallel metallic strips, and $N - 1$ gaps in between. Therefore, total parallel strip length $l_{tot,NSRR}$ is given as:

$$l_{tot,NSRR} = \sum_{k=1}^{N-1} l = (N - 1)l \quad (1)$$

where l is the length of each strip. For B-NSRR, the midsection paths shown in Figure 3(b) make a 90° bend and have a varying size. The total parallel metal length $l_{tot,BNSRR}$ is then calculated as:

$$\begin{aligned}
l_{tot, BNSRR} &= 2 \times 2 [(l - g/2) + (l - g - w - g/2) + (l - 2g - 2w - g/2) + \dots] \\
&= \sum_{k=1}^{N/2-1} 4[l - (k - 1/2)g - (k - 1)w]
\end{aligned}$$

where W is the strip width and g is the gap between the strips. The summation is made from the first gap until the $(N/2 - 1)^{\text{th}}$ gap, which has the path with smallest length. The result is multiplied with 2 to include the 90° bend portion and further again with 2 due to horizontal symmetry of the structure. Calculated variations of total parallel metallic strip length with the strip number N for the two geometries are presented in Figure 4. Here, the parameter values are selected as $w = g = 0.194$ mm and $l = 4.61$ mm. The comparison of $l_{tot, NSRR}$ and $l_{tot, BNSRR}$ is important, since maximizing the total parallel metallic strip length implies maximizing C_s , which in turn increases the sensitivity of the structure. It can be observed from Figure 4 that the advantage of the B-NSRR is clear for the whole range of N and increases as N is increased before saturating a little bit. Therefore, it can be expected that the B-NSRR yields a higher gap capacitance.

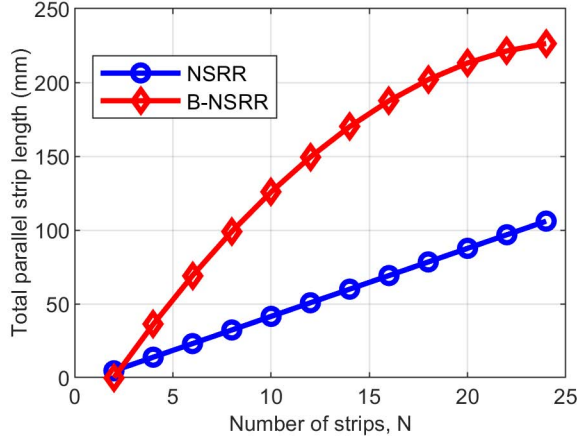


FIGURE 4. Variation of total parallel metallic strip length with strip number N for NSRR and B-NSRR.

For displacement sensing, the B-NSRR structure can also be split into two symmetric parts to form a modified B-NSRR geometry, as illustrated in Figure 1(d). The surface current densities induced on the modified NSRR and B-NSRR structures for a plane wave excitation are shown in Figure 5(a) and Figure 5(b), respectively. Both of the plots are normalized with respect to the same color scale. Both geometries are observed to have a high current density at their top strips shorted

by the jumper. The current distribution of the NSRR is tapered along the vertical axis, while the B-NSRR has a concentrated current density along the middle strip. The fact that the current density is higher close to the vertical gap in the B-NSRR in comparison to the NSRR can be considered as an important advantage in terms of displacement sensing.

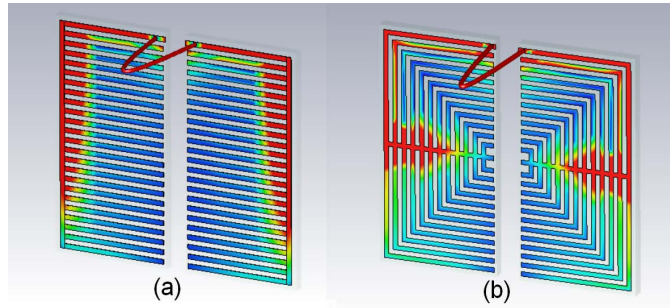


FIGURE 5. Surface current densities induced on the modified (a) NSRR and (b) B-NSRR structures for a plane wave excitation.

3. SRR STRUCTURES AS SENSORS

In this section, the sensing performances of the four SRR-based structures (classic SRR, ESRR, NSRR and B-NSRR) are analyzed with full-wave electromagnetic simulations run on CST Microwave Studio [33]. The designs used in the simulations are illustrated in proportion in Figure 6 and the dimensional parameters of the designs are presented in Table 1. The dielectric material is 0.508-mm-thick Rogers RO4003C with a dielectric constant of 3.55 for all structures. The dimensions are selected such that the resonance frequencies f_{res} of all 4 designs are in 1500–1800 MHz range. Although their f_{res} is around the same range, it can be observed that the ESRR design that corresponds to this frequency has the biggest size. The 3-ring SRR geometry has the second biggest size, while the NSRR and B-NSRR designs have a significantly more compact footprint.

The quality factor (Q) is an important metric for any resonating structure since it is an expression of the ratio of the stored energy to the loss. Therefore, the higher the Q , the better the sensitivity. In order to assess the quality factor of each SRR variant, each structure is excited by a plane wave, and its far-field reflection spectrum is analyzed. It is possible to look at either the absorption cross section (ACS) or the radar cross section (RCS), both of which yield a peak at the resonance frequency of the SRR. Here, we choose to compare the RCS for each structure. In Figure 7, the radar cross sections of each SRR variant with the design parameters given in Table 2 are compared. It can be understood that the B-NSRR yields the highest Q , the ESRR and the SRR come second and third, respectively, while

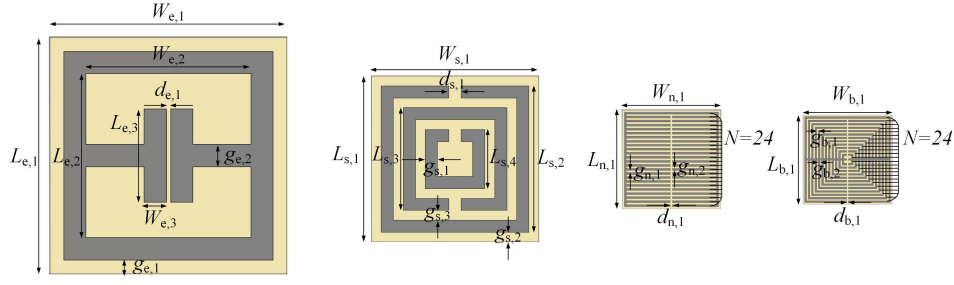


FIGURE 6. ESRR, SRR, NSRR and B-NSRR sensor designs used in this work (shown with a true proportion) with dimension parameters.

TABLE 1. Values of the dimensional parameters of sensors shown in Figure 6.

ESRR		SRR		NSRR		B-NSRR	
$W_{e,1}$	27.1 mm	$W_{s,1}$	18.9 mm	$W_{n,1}$	11.2 mm	$W_{b,1}$	10.2 mm
$W_{e,2}$	18.8 mm	$L_{s,1}$	18.9 mm	$L_{n,1}$	11.2 mm	$L_{b,1}$	10.2 mm
$W_{e,3}$	2.55 mm	$L_{s,2}$	16.7 mm	$d_{n,1}$	0.229 mm	$d_{b,1}$	0.194 mm
$L_{e,1}$	27.1 mm	$L_{s,3}$	11.7 mm	$g_{n,1}$	0.229 mm	$g_{b,1}$	0.194 mm
$L_{e,2}$	18.8 mm	$L_{s,4}$	6.73 mm	$g_{n,2}$	0.229 mm	$g_{b,2}$	0.194 mm
$L_{e,3}$	10.7 mm	$d_{s,1}$	1.40 mm	N	24	N	24
$d_{e,1}$	0.478 mm	$g_{s,1}$	1.37 mm				
$g_{e,1}$	1.59 mm	$g_{s,2}$	1.11 mm				
$g_{e,2}$	2.55 mm	$g_{s,3}$	1.11 mm				

the NSRR provides the smallest Q . It is an interesting fact which shows that the increased capacitance and inductance due to elongation of the strips is more effective in increasing Q than having a higher number of strips (N).

TABLE 2. Quality factor (Q) values for the SRR, ESRR, NSRR and B-NSRR designs.

	Q
SRR	38.7
ESRR	40.3
NSRR	20.9
B-NSRR	80.5

The SRR-based designs are first employed as moisture content sensors in this form, where the decrease of f_{res} is tracked when the water content of the dielectric slab placed behind the structure is gradually increased. The dielectric material loading the sensor mimics an oak tree trunk, whose relative permittivity and loss tangent values measured at different moisture content levels are presented in [34].

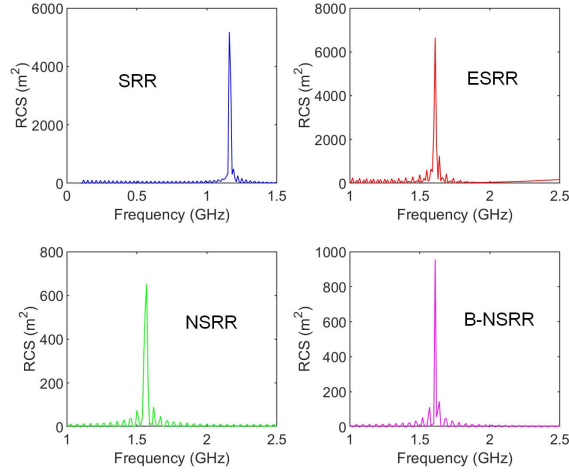


FIGURE 7. Comparison of RCS for SRR, ESRR, NSRR and B-NSRR for the designs whose parameters are given in Table 7.

The NSRR was employed as a moisture content (MC) sensor in [31], where a sensitivity of 1.1 MHz/%MC was demonstrated. Here, a comparison is made between the NSRR, B-NSRR and other SRR variants, and the results are displayed in Figure 8(a). The nonlinear decrease in f_{res} is evident for all structures. Sensitivity metric of the MC sensor can be defined as the average resonance frequency shift per MC change. The frequency shifts Δf_{res} can best be understood by subtracting the f_{res} obtained when MC is zero from all resonance frequencies, which is presented in Figure 8(b). It is observed that the ESRR yields the highest sensitivity with around 275 MHz shift for 90% MC change while the classic SRR and NSRR follow next. B-NSRR design has the lowest Δf_{res} , which is around 170 MHz. These results can be expected since moisture content sensing depends on variation of complex permittivity of the medium loading the SRR-based sensors, and it is inevitably proportional to the dimensions of the structures.

Although they can also be exploited in permittivity-based sensing applications, the real strength of the NSRR and B-NSRR lies in the measurement of one-dimensional strain and displacement, since the structure geometry is optimized especially for this purpose. The variation of C_{gap} with d yields a high sensitivity to mechanical changes occurring in the horizontal axis, which take place either as a high increase or drop in f_{res} . The variation of f_{res} with the gap parameter d is displayed in Figure 9(a), and the frequency shifts Δf_{res} for this case are presented in Figure 9(b). Here, only the results for ESRR, NSRR and B-NSRR are shown, since the classic SRR geometry is not suitable for measurement of relative displacement. All the remaining geometries are optimized for sensing horizontal displacement,

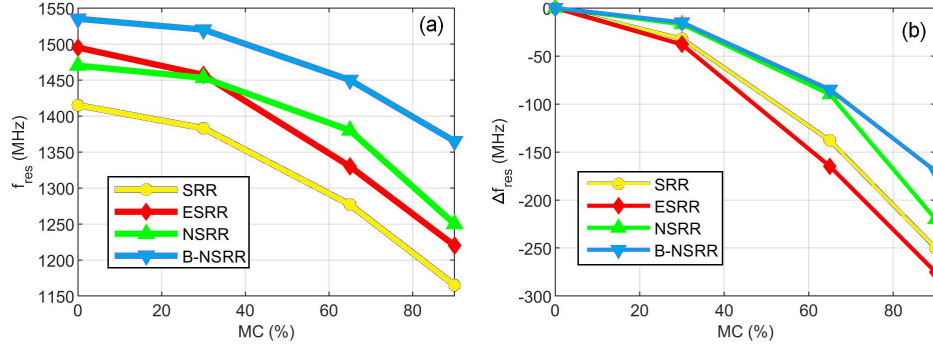


FIGURE 8. (a) Variation of f_{res} with moisture content for an oak sample, (b) the frequency shifts Δf_{res} .

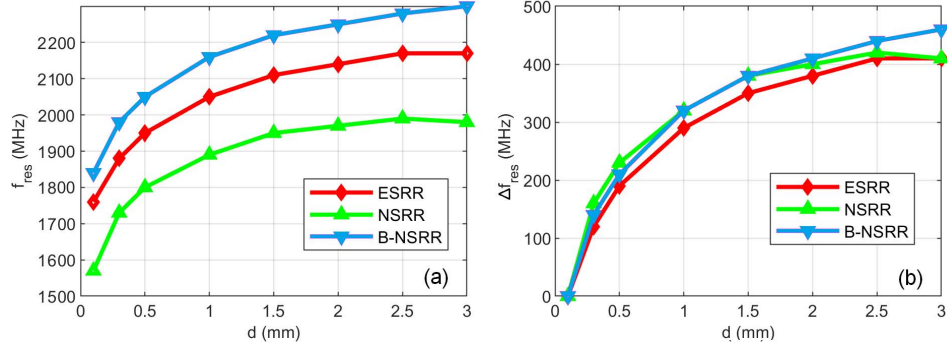


FIGURE 9. (a) Variation of f_{res} with the gap parameter d , (b) the frequency shifts Δf_{res} .

due to the presence of a small gap between the metallic parts. Similar to the modification procedure of the NSRR and B-NSRR, to obtain a displacement sensor, the ESRR structure is also split vertically into two mechanically-independent halves which are electrically shorted at the top and bottom. Sensitivity of the B-NSRR structure is observed to be the highest among all three structures, and it is around $450 \text{ MHz}/3 \text{ mm}=150 \text{ MHz}/\text{mm}$, while the other two are around $400 \text{ MHz}/3 \text{ mm}=133 \text{ MHz}/\text{mm}$. Although B-NSRR is the most compact structure, it also yields the highest displacement sensitivity, which shows its potential as a sensor.

4. CONCLUSION

We present a novel split-ring resonator structure called the bended nested splitting resonator (B-NSRR), which is primarily optimized for applications such as

moisture content, displacement or strain sensing. The structure is a variant of the previously developed NSRR geometry, and similarly, incorporates a high number of metallic strips, enabling better sensitivity and resolution compared to traditional SRR-based structures. Moreover, it offers a way to obtain much more compact designs, which is important in many applications. The sensitivity of the present NSRR structure is enhanced even more by introducing 90° bends on the strips, which leads to an increase in the total parallel metallic length, which in turn boosts the capacitance between the parallel strips. This way, better enhancement of fields and further miniaturization of the NSRR are achieved. Performance of the B-NSRR structure as a sensor of moisture content and relative displacement is compared to that of other SRR-based variants and it is shown that the B-NSRR structure can be utilized for both applications. However, it is observed to be the best option in displacement sensing among all SRR variants since it enables both a more sensitive and a more compact design.

Declaration of Competing Interests The author declares no known competing interests.

REFERENCES

- [1] Smith, D. R., Padilla, W. J., Vier, D. C., Nemat-Nasser, S. C., Schultz, S., Composite medium with simultaneously negative permeability and permittivity, *Phys. Rev. Lett.*, 84 (18) (2000), 4184–4187, <https://doi.org/10.1103/PhysRevLett.84.4184>.
- [2] Shelby, R. A., Smith, D. R., Schultz, S., Experimental verification of a negative index of refraction, *Science*, 292 (5514) (2001), 77–79, <https://doi.org/10.1126/science.1058847>.
- [3] Shelby, R. A., Smith, D. R., Nemat-Nasser, S. C., Schultz, S., Microwave transmission through a two-dimensional, isotropic, left-handed metamaterial, *Appl. Phys. Lett.*, 78 (4) (2001), 489–491, <https://doi.org/10.1063/1.1343489>.
- [4] Pendry, J. B., Negative refraction makes a perfect lens, *Phys. Rev. Lett.*, 85 (18) (2000), 3966–3969, <https://doi.org/10.1103/PhysRevLett.85.3966>.
- [5] Enoch, S., Tayeb, G., Sabouroux, P., Guerin, N., Vincent, P., A metamaterial for directive emission, *Phys. Rev. Lett.*, 89 (21) (2002), 213902, <https://doi.org/10.1103/PhysRevLett.89.213902>.
- [6] Erentok, A. and Luljak, P. L. and Ziolkowski, R. W., Characterization of a volumetric metamaterial realization of an artificial magnetic conductor for antenna applications, *IEEE Trans. Antennas Propag.*, 53 (1) (2005), 160-172, <https://doi.org/10.1109/TAP.2004.840534>.
- [7] Schurig, D., Mock, J. J., Justice, B. J., Cummer, S. A., Pendry, J. B., Starr, A. F., Smith, D. R., Metamaterial electromagnetic cloak at microwave frequencies, *Science*, 314 (5801) (2006), 977–980, <https://doi.org/10.1126/science.1133628>.
- [8] Engheta, N., An idea for thin subwavelength cavity resonators using metamaterials with negative permittivity and permeability, *IEEE Antennas Wireless Propag. Lett.*, 1 (2002), 10-13, <https://doi.org/10.1109/LAWP.2002.802576>.
- [9] Antoniadou, M. A., Eleftheriades, G. V., Compact linear lead/lag metamaterial phase shifters for broadband applications, *IEEE Antennas Wireless Propag. Lett.*, 2 (2003), 103-106, <https://doi.org/10.1109/LAWP.2003.815280>.
- [10] Falcone, F., Lopetegi, T., Baena, J. D., Marques, R., Martin, F., Sorolla, M., Effective negative-epsilon stopband microstrip lines based on complementary split

- ring resonators, *IEEE Microw. Wireless Compon. Lett.*, 14 (6) (2004), 280-282, <https://doi.org/10.1109/LMWC.2004.828029>.
- [11] Antoniadou, M. A., Eleftheriades, G. V., A broadband Wilkinson balun using microstrip metamaterial lines, *IEEE Antennas Wireless Propag. Lett.*, 4 (2005), 209-212, <https://doi.org/10.1109/LAWP.2005.851005>.
- [12] Ziolkowski, R. W., Kipple, A. D., Application of double negative materials to increase the power radiated by electrically small antennas, *IEEE Trans. Antennas Propag.*, 51 (10) (2003), 2626-2640, <https://doi.org/10.1109/10.1109/TAP.2003.817561>.
- [13] Chen, T., Li, S., Sun, H., Metamaterials application in sensing, *Sensors*, 12 (3) (2012), 2742-2765, <https://doi.org/10.3390/s120302742>.
- [14] Lee, H.-J., Lee, J.-H., Moon, H.-S., Jang, I.-S., Choi, J.-S., Yook, J.-G., Jung, H.-I., A planar split-ring resonator-based microwave biosensor for label-free detection of biomolecules, *Sens. Actuators, B*, 169 (2012), 26-31, <https://doi.org/10.1016/j.snb.2012.01.044>.
- [15] Al-Naib, I. A. I., Jansen, C., Koch M., Thin-film sensing with planar asymmetric metamaterial resonators, *Appl. Phys. Lett.*, 93 (8) (2008), 083507, <https://doi.org/10.1063/1.2976636>.
- [16] Melik, R., Unal, E., Perkgoz, N. K., Puttlitz, C., Demir, H. V., Metamaterial based telemetric strain sensing in different materials, *Opt. Express*, 18 (5) (2010), 5000-5007, <https://doi.org/10.1364/OE.18.005000>.
- [17] Tian, X., Lee, P. M. Tan, Y. J., Wu, T. L. Y., Yao, H., Zhang, M., Li, Z., Ng, K. A., Tee, B. C. K., Ho, J. S., Wireless body sensor networks based on metamaterial textiles, *Nat. Electron.*, 2 (6) (2019), 243-251, <https://doi.org/10.1038/s41928-019-0257-7>.
- [18] Ekmekci, E., Turhan-Sayan, G., Multi-functional metamaterial sensor based on a broad-side coupled SRR topology with a multi-layer substrate, *Appl. Phys. A: Mater. Sci. Process.*, 110 (2013), 189-197, <https://doi.org/10.1007/s00339-012-7113-1>.
- [19] Pendry, J.B., Holden, A.J., Robbins, D.J., Stewart, W.J., Magnetism from conductors and enhanced nonlinear phenomena, *IEEE Trans. Microw. Theory Tech.*, 47 (11) (1999), 2075-2084, <https://doi.org/10.1109/22.798002>.
- [20] Engheta, N., Ziolkowski, R. W., *Electromagnetic Metamaterials: Physics and Engineering Explorations*, Wiley, Hoboken, N.J., 2006.
- [21] Padilla, W. J., Aronsson, M. T., Highstrete, C., Lee, M., Taylor, A. J., Averitt, R. D., Electrically resonant terahertz metamaterials: Theoretical and experimental investigations, *Phys. Rev. B: Condens. Matter*, 75 (4) (2007), 041102, <https://doi.org/10.1103/PhysRevB.75.041102>.
- [22] Zhang, J., Tian, G. Y., Marindra, A. M. J., Sunny, A. I., Zhao, A. B., A Review of Passive RFID Tag Antenna-Based Sensors and Systems for Structural Health Monitoring Applications, *Sensors*, 17 (2) (2017), 265, <https://doi.org/10.3390/s17020265>.
- [23] Melik, R., Unal, E., Perkgoz, N.K., Santoni, B., Kamstock, D., Puttlitz, C., Demir, H.V., Nested metamaterials for wireless strain sensing, *IEEE J. Sel. Topics Quantum Electron.*, 16 (2) (2010), 450-458, <https://doi.org/10.1109/JSTQE.2009.2033391>.
- [24] Ozbey, B., Unal, E., Ertugrul, H., Kurc, O., Puttlitz, C. M., Erturk, V. B., Altintas, A., Demir, H. V., Wireless displacement sensing enabled by metamaterial probes for remote structural health monitoring, *Sensors*, 14 (1) (2014), 1691-1704, <https://doi.org/10.3390/s140101691>.
- [25] Ozbey, B., Demir, H. V., Kurc, O., Erturk, V. B., Altintas, A., Wireless measurement of elastic and plastic deformation by a metamaterial-based sensor, *Sensors*, 14 (10) (2014), 19609-19621, <https://doi.org/10.3390/s141019609>.
- [26] Ozbey, B., Demir, H. V., Kurc, O., Erturk, V. B., Altintas, A., Wireless sensing in complex electromagnetic media: Construction materials and structural monitoring, *IEEE Sensors J.*, 15 (10) (2015), 5545-5554, <https://doi.org/10.1109/JSEN.2015.2441555>.

- [27] Ozbey, B., Erturk, V. B., Demir, H. V., Altintas, A., Kurc, O., A wireless passive sensing system for displacement/strain measurement in reinforced concrete members, *Sensors*, 16 (4) (2016), 496, <https://doi.org/10.3390/s16040496>.
- [28] Ozbey, B., Range extension in coupling-based wireless passive displacement sensors for remote structural health monitoring, *IEEE Sensors J.*, 22 (21) (2022), 20268-20275, <https://doi.org/10.1109/JSEN.2022.3206475>.
- [29] Ozbey, B., Ertürk, V. B., Kurc, O., Altintas, A., Demir, H. V., Multi-point single-antenna sensing enabled by wireless nested split-ring resonator sensors, *IEEE Sensors J.*, 16 (21) (2016), 7744–7752, <https://doi.org/10.1109/JSEN.2016.2604020>.
- [30] Ozbey, B., Wireless surface strain mapping by passive electromagnetic resonators, *IEEE Sensors J.*, 23 (10) (2023), 10370-10377, <https://doi.org/10.1109/JSEN.2023.3264948>.
- [31] Ozbey, B., Eibert, T. F., Wireless non-destructive moisture content characterization of trees by highly-sensitive compact resonating probes, *IEEE Sensors J.*, 21 (5) (2021), 6125–6132, <https://doi.org/10.1109/JSEN.2020.3043304>.
- [32] Ozbey, B., Altintas, A., Demir, H. V., Ertürk, V. B., An equivalent circuit model for nested split-ring resonators, *IEEE Trans. Microw. Theory Tech.*, 65 (10) (2017), 3733–3743, <https://doi.org/10.1109/TMTT.2017.2699650>.
- [33] Dassault Systemes, CST Studio Suite, Vélizy-Villacoublay, France, 2019.
- [34] James, W. L., Dielectric Properties of Wood and Hardboard: Variation with Temperature, Frequency, Moisture Content, and Grain Orientation. Department of Agriculture, Forest Service, Forest Products Laboratory, 1975.

CLASSIFICATION OF HUMAN ACTIVITIES BY SMART DEVICE MEASUREMENTS

Muruvvet KALKAN¹ and Yilmaz AR¹

¹Department of Computer Engineering, Ankara University, Ankara, TÜRKİYE

ABSTRACT. The prevalence of activity detectors in users' personal mobile devices has been incorporated into an increasing interest in research into physical function recognition (HAR - Human Activity Recognition). With this research interest, different enterprises developed HAR systems working with measurement devices and still work on this subject. Although many HAR systems have been developed, there are still concrete practical limits. This situation is improved with modern techniques such as machine learning. A properly trained machine learning model predicts human activity from measured data. The data was measured at certain time intervals by sensors on smartphones. These different machine learning architectures were trained on sensor data that detected human activities, and their accuracy was calculated. A HAR system that predicts human activity is constructed separately with five approaches. KNN, Random Forest, Decision Tree, MLP and Gaussian Naive Bayes algorithms were used, and KNN produced the most accurate results.

1. INTRODUCTION

Detecting human motion is applied on various fields for scientific or commercial gain. Especially today, widely used mobile health applications come with the feature of detecting human activities. Various hardware of smart devices allows the data to be gathered. Modern ready-to-use smart phones and watches contain a diverse set of embedded sensors. For example, accelerometer, gyroscope, compass, WiFi, NFC and GPS [1]. The proliferation of such sensor-rich mobile devices is already in our daily lives. This provides an opportunity to unobtrusively capture information from human behavior in real time. It also provides easier development and the rapid growth of public mobile sensing applications. Thus, new possibilities are available for new mobile sensing research.

Among the sensors found on mobile device platforms, the accelerometer is one of the oldest and the most common. The accelerometer has gained immense popularity

Keywords. Human activity recognition, HAR, machine learning, accelerometer, gyroscope.

✉ kalkanm@ankara.edu.tr-Corresponding author; 📞 0000-0001-8056-1905

✉ ar@ankara.edu.tr; 📞 0000-0003-2370-357X.

in HAR research as it allows the recognition of a wide variety of human activities while having a relatively small energy consumption. Accelerometer-based HAR is used in numerous fields, including smart homes, healthcare, daily activity tracking, fitness tracking, elderly fall detection, and transportation mode detection [1]. Other motion-related sensors, such as the compass and gyroscope, are becoming more common and are often used to assist and complement the accelerometer. Especially gyroscope is useful in that sense, since it measures the state of the device according to gravity: oblique, vertical, horizontal. Motion sensors can also be paired with other sensors, such as GPS, GSM, WiFi and barometer, especially to recognize tasks beyond basic HAR.

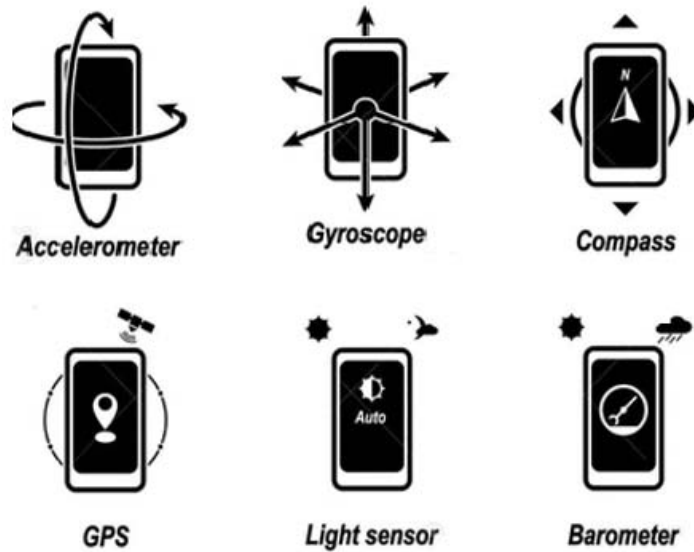


FIGURE 1. Sensors in smart mobile devices [2].

In this study, measurements are taken at small time intervals to make numerical calculations from the accelerometer and gyroscope. The body's acceleration/angular velocity is calculated from the mean, standard deviation of device acceleration/angular velocity by these measurements are also included in the original measurement data. Therefore, the success of ML classification algorithms performed on recognizing human activities with this data is tried in this work.

In the following sections certain points are made. In Problem Definition, the subject of the research is given. In Literature Review, the past scientific studies are examined on the subject of the domain and the proposed method. In Data set, input data with the size, type and the way it is produced is described in detail.

In Methodology, the proposed method with algorithms is given step by step and explained in detail. In Experimental Results, Discussion and Conclusion sections, the results of the experiment are displayed, compared and discussed, then all the work done is concluded.

2. LITERATURE REVIEW

Human activity recognition has always been an interested topic for all researchers whose area of expertise is feasible. Therefore, there are a lot of different methods to perform HAR depending, like video surveillance, sound tracking, image processing, motion detection with sensors etc. Depending on the input type, the methods narrow accordingly but overall they are various [3]. In this study, the method of motion detection with sensors is preferred, so sensor data is used as input and machine learning classification algorithms: KNN, Random Forest, Decision Tree, MLP, Naive Bayes are chosen for processing the data and performing HAR.

K-Nearest Neighbor (KNN) is one of the simplest algorithms belonging to the unsupervised class of machine learning algorithms. There are two important things to know about KNN. First, KNN is a non-parametric machine learning algorithm, excluding the number of nearest neighbors (K). This means that no assumptions are made about the data set when the model is used. Rather, the model is built entirely from the data provided. Second, KNN makes no generalizations between a training and test set, so all training data is also used when the model is asked to make predictions. KNN models calculate similarity using the distance between two points on a graph. The greater the distance between the points, the less similarity. There is more than one way to calculate the distance between points, but the most common distance measure is Euclidean distance. When applying this process, it assumes the similarity between the new case/data and existing cases and puts the new case in the category most similar to the existing categories. For this reason, KNN is known as a lazy machine learning algorithm. For this data set, the number of nearest neighbors is selected as five ($K = 5$) [4].

The Random Forest algorithm consists of the required number of decision tree algorithms that work in a collective just like the forests consisting of trees. Random Forest, which can be used for both regression and classification, belongs to the supervised category of ML algorithms. In this algorithm, which consists of a large number of decision trees, the decision trees are fused, and each decision tree is trained on a different observation sample, then their results are combined to produce an overall correct result [5].

The Decision Tree algorithm belongs to the supervised class of ML algorithms. Unlike other supervised ML algorithms, this algorithm can also be used in regression and classification problems. The purpose of using a Decision Tree is to create a training ML model that can be used to predict the class or exact value of the targeted variable by learning the simple decision rules generated from the training data. In Decision Trees, one starts at the root of the tree to predict the class of a

target record. The values of the root's attribute are compared with the attribute values of the target record. On the basis of the comparison, the branch that meets that value is followed and the next node is passed [6].

Multi-Layer Perceptron (MLP), a supervised ML algorithm, has a multi-layered network structure used especially in classification problems. When making forward calculations, a net input value is found by calculating the inputs transferred to the system. The output of the current phase is calculated by passing the obtained input value through the activation function. The calculated output is transferred to the next layer. These processes continue from the input layer to the next layer, the middle layer, and from the middle layer to the output layer. Finally, the output values are created in the output layer. Thus, the first stage of learning is completed. An error value will occur if the output value received from the network is different from the expected output value. With backward calculation, the error value is distributed and updates are made in each iteration and it is expected to get closer to the expected result [7].

Naive Bayes is a supervised ML algorithm based on the theorem put forward by Thomas Bayes and can be used to classify data. It is a classification algorithm based on probability methods. Predicts which class the target record or target data point may belong to, using probability calculations. The target record/data point is assumed to belong to the class with the highest value from the calculated probabilities. The more data entered into the algorithm for training purposes, the higher the accuracy of predicting the result. Naive Bayes is the oldest known ML classification algorithm, which is primitive, but easy to implement [8].

There are many academic studies on recognizing human activities. Ganapati Bhat and his team collected their data by getting the measurements from special wearable IoT devices they designed rather than smartphones [9]. They then trained their models on this data with reinforcement learning. Allan Stisen and his research associates studied heterogeneity in the measurements of the devices by taking measurements from different HAR devices [1]. In an evaluation run by Jindong Wang and his team, experiments on HAR using deep learning were discussed, compared and interpreted [10]. Also, Yilmaz and his colleague did a study on HAR using deep learning with genetic algorithms and proposed a novel approach [11]. In two studies, special HAR systems were developed to assist and support elderly people [12, 13]. There is also a recent study that delves into HAR field with smart phones sensors as input [14]. Lastly, a specific HAR system designed for the processors of mobile devices is designed using SVM (support vector machine) [15].

3. METHODOLOGY

The data set of this experiment consists of sensor measurements made with smartphones and human activities that they describe through these measurements. The

measurements are made with the accelerometer and gyroscope inside the smart devices. The data set consists of 10 thousand rows. There are 563 columns in total. The columns include:

- Acceleration from the triaxial accelerometer and estimated body acceleration.
- Angular velocity measured by the three-axis gyroscope.
- Variable values held in 561 columns, measured and calculated at specific time intervals (with frequencies).
- Subject: The code of the person doing the activity, namely which user it is.
- Activity: The activity that the user is doing.

The activity column is preferred to classify the data set. With the processed measurement data, it is determined which activity the movement is. For this purpose, the classification algorithms of machine learning (ML) were applied on the data set by accepting the Activity column as the class column.

In the data set, the Activity column can have one of the six classes in Table 1.

TABLE 1. The classes of the data sets.

Class	Activity
WALKING	Walking
WALKING_UPSTAIRS	Walking Upstairs
WALKING_DOWNSTAIRS	Walking Downstairs
SITTING	Sitting
STANDING	Standing
LAYING	Laying

The data set was created by a certain number of volunteers, between the ages of 18-49. Each of them carried out six different activities with a smart phone he wore around his waist. Experiments were video-recorded to tag actions later as they were performed. The data taken from the accelerometer and gyroscope were processed by passing through noise filters. All results were combined and the final version of the data set was divided into two as 70% training and 30% test data set.

In the experiment certain libraries of Python are preferred and all the calculations were done by them. They are Scikit-learn, NumPy, Pandas and Matplotlib. “Scikit-learn is one of the most popular ML libraries used by the data science community. Available in Python programming language, scikit-learn is very effective for supervised or unsupervised ML applications and data processing. However, Scikit-learn allows developers to use many ML algorithms. Scikit-learn is built on commonly known data processing Python libraries such as NumPy, Pandas and Matplotlib.” [16].

All models were run sequentially with the algorithm steps given on the data set. These steps in the program is given in the following Algorithm 1.

Algorithm 1. *Building the experimental model, training and evaluation.*

procedure MODEL CONSTRUCTION

Fetch the data set

Split them as training, test and validation data sets

Approximately 70% training, 20% validation, 10% test

model \leftarrow *get the model (KNN, Decision Tree, Random Forest, MLP, Gaussian Naive*

Bayes)

model.trainable \leftarrow *True*

metrics \leftarrow [*accuracy, loss, precision,*
recall, f1Score, roc]

model.compile(metrics)

Evaluate model by the test metrics

end procedure

The Algorithm 1 can be explained with following steps.

- (1) The data set is fetched.
- (2) The data set is partitioned to train and test data sets (70% - 30%, respectively).
- (3) The chosen ML model is loaded from Scikit-learn (One of KNN, Random Forest, Decision Tree, MLP or Naive Bayes)
- (4) The model is trained with training data set
- (5) Trained model produce predictions by the test data set and these predictions are saved.
- (6) Trained model produce prediction probabilities by the test data set and these prediction probabilities are saved.
- (7) Using produced predicted results, prediction probabilities and actual labels of the test data set, the following evaluation metrics are calculated: Accuracy, loss, precision, recall, F1-score and ROC AUC.
- (8) All steps from 3 to 7 are repeated for each algorithm and all of their scores are saved.
- (9) Saved scores are displayed in graphs with Matplotlib.

4. EXPERIMENTAL RESULTS

Algorithms run on the data set with 10.300 tuples were analyzed according to some evaluation metric results. The metrics examined are accuracy, loss, precision, recall, f1-score and ROC AUC values. Except for the loss, all of them are calculated from the values of prediction systems by 2:

- **TP:** Number of True Positives
- **TN:** Number of True Negatives
- **FP:** False Positives
- **FN:** False Negatives

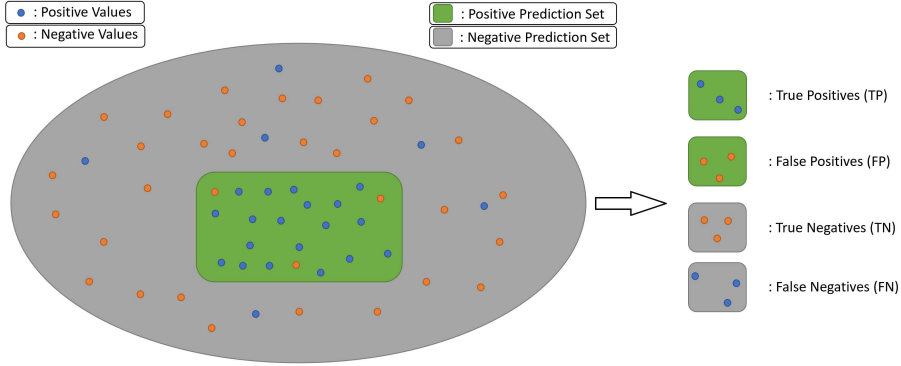


FIGURE 2. Prediction value definitions.

Accuracy is defined as the ratio of correctly detected data in the model to all data. It is not sufficient on its own to measure the relevance of the study, but it is the most important value, Equation 1.

Loss measures the difference between the true value of the sample and the predicted value in the model. The greater the difference, the greater the loss. It indicates how far are the predictions from real values.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

Precision is the ratio of actual positives to all predicted positives. It measures the precision of positively predicted values, Equation 2.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall shows how well the system is at not missing actual positives. The higher it is, the more true positives are accurately predicted, Equation 3.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

The F1-score value is the harmonic mean of the precision and recall values. Harmonic averaging is used to measure the balance between often contradicting precision and recall. F1-Score formula is given in Equations 4 and 5.

$$F_1 = 2 \bullet \frac{\textit{precision} \bullet \textit{recall}}{\textit{precision} + \textit{recall}} \quad (4)$$

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (5)$$

Area Under the Curve of Receiver Operating Characteristic (ROC AUC or AUC of ROC), the area under the ROC curve is expressed as AUC, see Figure 3. The larger this value in a model, the better the machine learning model is at classification. ROC curve is expressed on the 2D space with x-axis as the False Positive Rate (FPR) and y-axis as the True Positive Rate (TPR) of the model, see Equations 6 and 7.

$$TPR = \frac{TP}{TP + FN} \quad (6)$$

$$FPR = \frac{FP}{FP + FN} \quad (7)$$

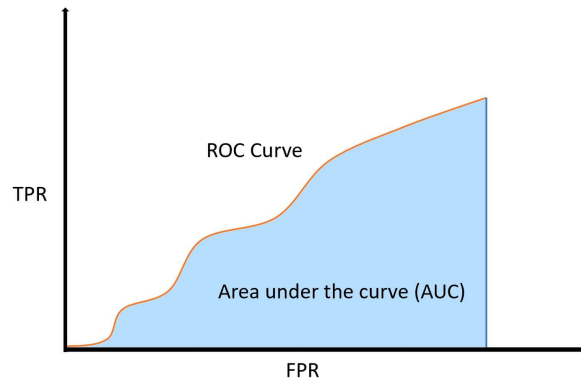


FIGURE 3. Area under the curve of receiver operating characteristic (ROC AUC).

All the metrics mentioned on the given models have been calculated separately. Analyzes are given according to the calculated values. Considering the accuracy values in the experiment, the KNN algorithm got the highest accuracy rate with 96.6%. MLP produced 94.56%, Decision Tree 89.09%, Random Forest 78.03% and Gaussian Naive Bayes algorithm 76.18% accuracies. The accuracy distribution of all algorithms is approximately between 76% and 96%. ROC AUC values range from 99% to 96%, and all models are close to each other, Figure 4.

When the loss data is examined, the MLP model got the least loss with a value of 0.15. Other models are followed by KNN with 0.16, Decision Tree with 0.31, and

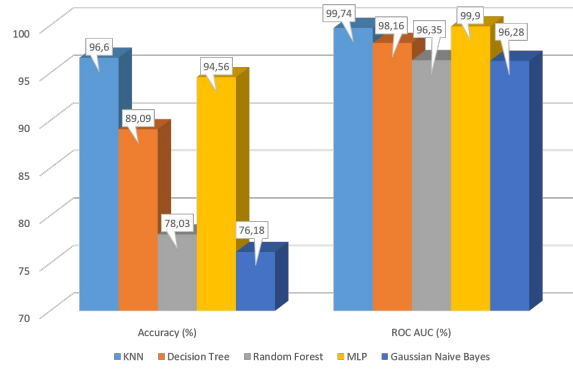


FIGURE 4. Graph of accuracy and AUC of ROC.

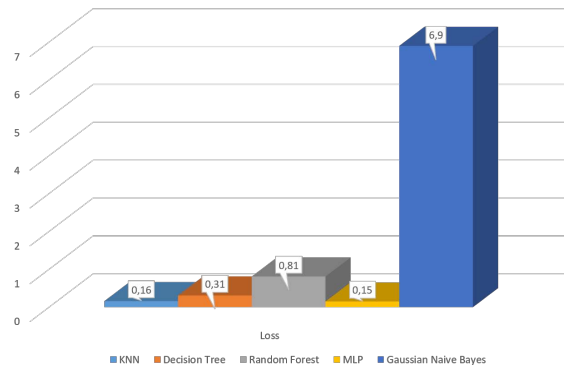


FIGURE 5. Graph of loss.

Random Forest with 0.81. With a value of 6.9, the Gaussian Naive Bayes algorithm has the highest loss, Figure 5.

Precision metric produced 96.82%, MLP 95.83%, Decision Tree 89.52%, Gaussian Naive Bayes 79.62%, Random Forest 78.07% on KNN. Recall values are 96.71% in the KNN model, 94.72% in MLP, 88.51% in Decision Tree, 77.59% in Random Forest, and 76.74% in Gaussian Naive Bayes. Finally, the F1-Score metric is produced as 96.75% by the KNN algorithm, 94.72% by MLP, 88.67% by Decision Tree, 77.4% by Random Forest, 75.42% by Gaussian Naive Bayes given in Table 2 and Figure 6.

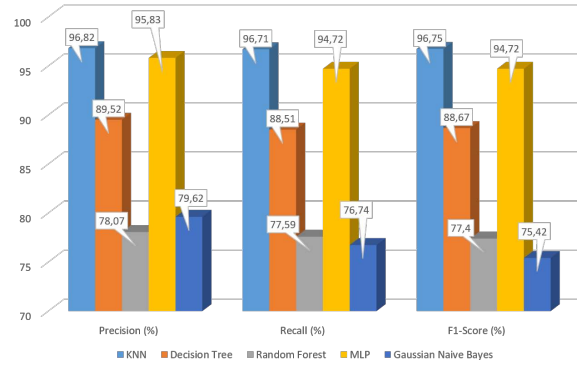


FIGURE 6. Graph of precision, recall and F1-score.

TABLE 2. The metrics results of the algorithms.

Algorithm	Accuracy%	Loss	Precision%	Recall%	F1-Score%	AUC of ROC%
KNN	96.60	0.16	96.82	96.71	96.75	99.74
Decision Tree	89.09	0.31	89.52	88.51	88.67	98.16
Random Forest	78.03	0.81	78.07	77.59	77.40	96.35
MLP	94.56	0.15	95.83	94.72	94.72	99.90
Gaussian Naive Bayes	76.18	6.90	79.62	76.74	75.42	96.28

5. DISCUSSION

When the study is evaluated in general, the model that produces the best result in almost all metric values is the KNN algorithm. While KNN is performing the classification operations, a distance is measured for each data to the neighbors of it, then a result is estimated by these distances. The label of the nearest neighbor is considered the class of the test data. Since the data set is a quite dense matrix, KNN with its own approach gave the best estimation values with a result of 96.6

The MLP classification algorithm iterates a lot of times to make itself to predict with the least error possible. Initially randomly assigned error weights are updated at each iteration and the network is optimized. In this way, with an accuracy rate of 94.56%, MLP can be taken as the second highest classification algorithm in the

study. Certainly, it points that the data set of the experiment is a good fit for MLP's way of classification.

The Gaussian Naive Bayes algorithm calculates the proximity to other data for the data to be classified and assigns it to the class with the highest probability. While it can give very good results with a small data set, the classification algorithm that can produce low values for large-volume data sets is 76% in the experiment. It has the lowest accuracy rate of 18. Considering it is a primitive algorithm compared to its peers, such a result is to be expected.

The Decision Tree model, which is easy to understand and interpret, followed with a rate of 89.09%. The model used is average for the experiment. Although not as strong as KNN and MLP, it produced satisfactory values.

Random Forest yielded disappointing results with 78.03%. Although it allows to create multiple decision trees on the data set and train each one separately, Random Forest could not provide satisfactory results for this experiment. Although it would be expected of Random Forest to produce far superior results to Gaussian Naive Bayes algorithm, their results are close, so Random Forest can be considered as a poor choice for this task.

When all algorithms are examined, the most powerful algorithm for the considered data set is the KNN model. It provided the best values in almost all metrics. However, the Random Forest algorithm could not meet the desired results and Gaussian Naive Bayes got the worst scores, especially the highest loss rate with a 6.9. The models that make the best classification in the study can be taken as KNN and MLP.

6. CONCLUSION

In this study, Classification of Smart Mobile Devices in the Recognition of Human Activities is provided by KNN, MLP, Random Forest, Decision Tree, and Gaussian Naive Bayes classification algorithms using the Scikit-learn library. In the application, the movement performed with various parameters is classified into six activities in total. Classification algorithms are evaluated by accuracy, loss, precision, recall, F1-score, and AUC-ROC curve. The KNN algorithm gave the best accuracy for the data set used, and the Gaussian Naive Bayes algorithm showed the lowest rate. Random Forest performed worst considering expectations. For the Loss metric, MLP produced the most accurate result, and the Gaussian Naive Bayes model produced the worst result. KNN algorithm produced the best results in all metrics.

Author Contribution Statements Conceptualization, M.K.; methodology, M.K.; software, M.K.; validation, M.K. and Y.A.; writing—original draft preparation, M.K. and Y.A.; writing—review and editing, M.K. and Y.A.; supervision, Y.A. All authors have read and agreed to the published version of the manuscript.

Declaration of Competing Interests The authors declare no conflict of interest.

REFERENCES

- [1] Stisen, A., et al., Smart devices are different: assessing and mitigating mobile sensing heterogeneities for activity recognition, *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, (2015), 127-140, <https://doi.org/10.1145/2809695.2809718>.
- [2] Accelerometer, gyroscope, compass, GPS, light sensor, barometer. Important phone functions. Black icon, (2023). Available: <https://stock.adobe.com/images/accelerometer-gyroscope-compass-gps-light-sensor-barometer-important-phone-functions-black-icon/170752038>. [Accessed: May 2023].
- [3] Vrigkas, M., Nikou, C. and Kakadiaris, I. A., A review of human activity recognition methods, *Front. Robot. AI*, 2 (2015), 28, <https://doi.org/10.3389/frobt.2015.00028>.
- [4] Cover, T. and Hart, P., Nearest neighbor pattern classification, *IEEE Trans. Inf. Theory*, 13 (1) (1967), 21-27, <https://doi.org/10.1109/TIT.1967.1053964>.
- [5] Breiman, L., Random forests, *Mach. Learn.*, 45 (1) (2001), 5-32, <https://doi.org/10.1023/A:1010933404324>.
- [6] Quinlan, J. R., Induction of decision trees, *Mach. Learn.*, 1 (1) (1986), 81-106, <https://doi.org/10.1007/BF00116251>.
- [7] Rosenblatt, F., The perceptron: a probabilistic model for information storage and organization in the brain, *Psychol. Rev.*, 65 (6) (1958), 386-408, <https://doi.org/10.1037/h0042519>.
- [8] Vikramkumar, B. and Vijaykumar, T., Bayes and naive bayes classifier, arXiv.1404.0933, (2014), <https://doi.org/10.48550/arXiv.1404.0933>.
- [9] Bhat, G., Deb, R., Chaurasia, V. V., Shill, H. and Ogras, U. Y., Online human activity recognition using low-power wearable devices, *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, (2018), 1-8, <https://doi.org/10.1145/3240765.3240833>.
- [10] Wang, J., Chen, Y., Hao, S., Peng, X. and Hu, L., Deep Learning for sensor-based activity recognition: A survey, *Pattern Recognit. Lett.*, 119 (2017), 3-11, <https://doi.org/10.1016/j.patrec.2018.02.010>.
- [11] Yilmaz, A. A., Guzel, M. S., Bostanci, E. and Askerzade, I., A novel action recognition framework based on deep-learning and genetic algorithms, *IEEE Access*, 8 (2020), 100631-100644, <https://doi.org/10.1109/ACCESS.2020.2997962.8>.
- [12] Chernbumroong, S., Cang, S., Atkins, A. and Yu, H., Elderly activities recognition and classification for applications in assisted living, *Expert Syst. Appl.*, 40 (5) (2013), 1662-1674, <https://doi.org/10.1016/j.eswa.2012.09.004>.
- [13] Janaki, M., Geethalakshmi, Dr. S. N., An efficient system for human activity recognition and monitoring for elderly people using machine learning, *SJIS*, 35 (1) (2023), 1194-1206.
- [14] Garcia-Gonzalez, D., Rivero, D., Fernandez-Blanco, E. and Luaces, M. R., New machine learning approaches for real-life human activity recognition using smartphone sensor-based data, *Knowl.-Based Syst.*, 262 (2023), 110260, <https://doi.org/10.1016/j.knosys.2023.110260>.
- [15] Anguita, D., Ghio, A., Oneto, L., Parra, X. and Reyes-Ortiz, J. L., Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine, *Proceedings of the 4th International Conference on Ambient Assisted Living and Home Care*, (2012), 216-223, <https://link.springer.com/chapter/10.1007/978-3-642-35395-6-30>.
- [16] SciKit-Learn, (2023). Available:<https://scikit-learn.org/stable/about.html>. [Accessed: May 2023].

APPENDIX

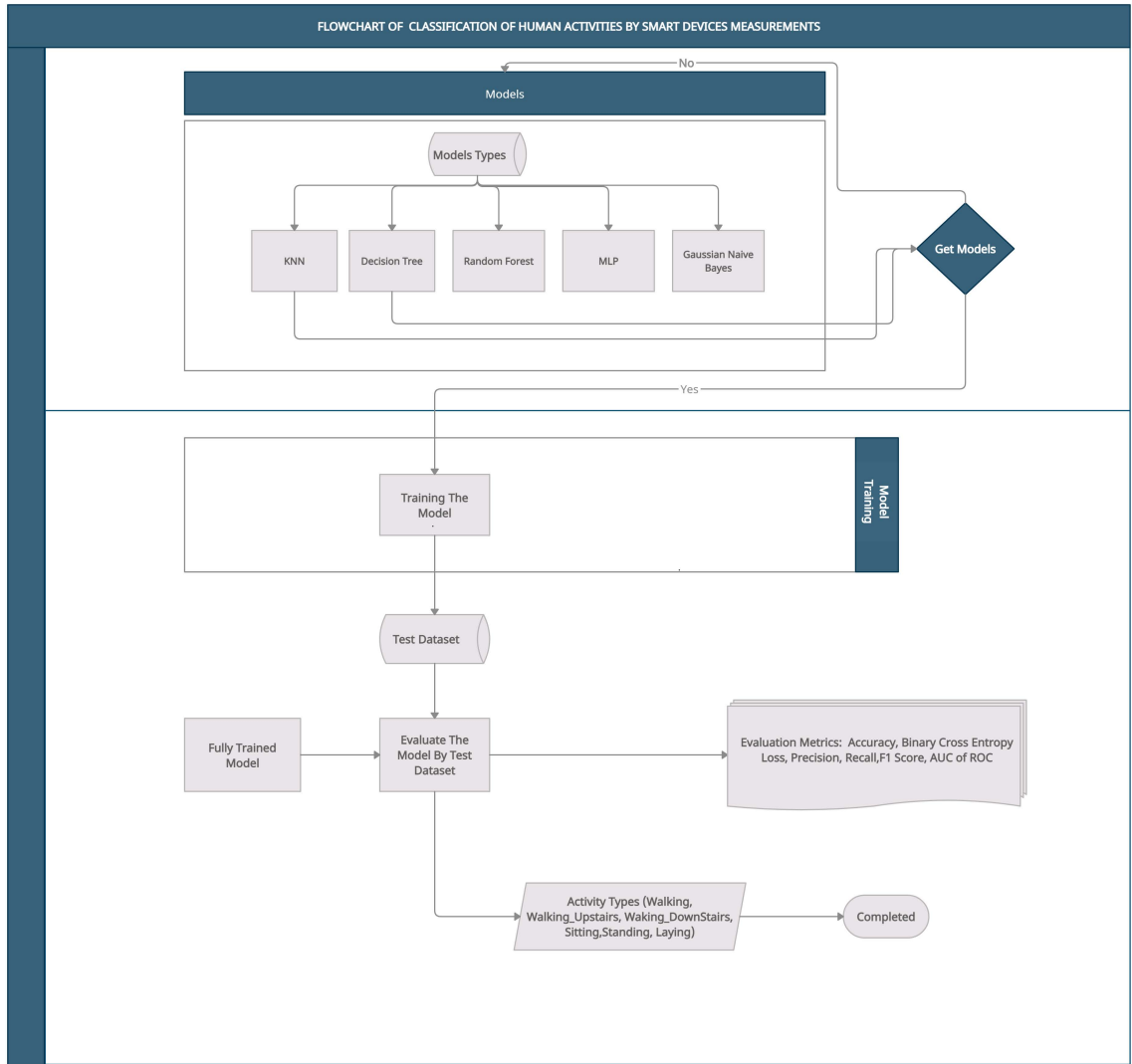


FIGURE 7. Flowchart representation of the experiment.

INSTRUCTIONS TO CONTRIBUTORS

Communications Faculty of Sciences University of Ankara Series A2-A3: Physical Sciences and Engineering is a single-blind peer reviewed open access journal which has been published since 1948 by Ankara University, accepts original research articles written in English in the fields of Physics, Engineering Physics, Electronics/Computer Engineering, Astronomy and Geophysics. Review articles written by eminent scientists can also be invited by the Editor.

Article-processing charges: The publication costs for Communications Faculty of Sciences University of Ankara Series A2-A3: Physical Sciences and Engineering are covered by the journal, so authors do not need to pay an article-processing and submission charges. The PDF copies of accepted papers are free of charges and can be downloaded from the website. Hard copies of the paper, if required, are due to be charged for the amount of which is determined by the administration each year.

Submission: All manuscripts should be submitted via our online submission: <https://dergipark.org.tr/en/journal/2457/submission/step/manuscript/new> Note that only two submissions per author per year will be considered. Once a paper is submitted to our journal, all co-authors need to wait 6 months from the submission date before submitting another paper.

Cover Letter: Manuscripts should be submitted in the PDF form used in the peer-review process together with **THE COVER LETTER** and the source file (Supporting File). In the cover letter the authors should suggest the most appropriate Area Editor for the manuscript and potential four reviewers with full names, universities and institutional email addresses. Proposed reviewers must be experienced researchers in your area of research and at least two of them should be from different countries. In addition, proposed reviewers must not be co-authors, advisors, students, etc. of the authors. In the cover letter, the author may enter the name of anyone who he/she would prefer not to review the manuscript, with detailed explanation of the reason. Note that the editorial office may not use these nominations, but this may help to speed up the selection of appropriate reviewers.

Preparing your manuscript: Manuscripts should be typeset using as DOC or LaTeX. Authors will submit their manuscript and the cover letter via our submission system. A template of manuscript can be reviewed in <https://dergipark.org.tr/tr/download/journal-file/20554> (or can be reviewed in [pdf form](#)).

Title Page: The title page should contain the title of the paper, full names of the authors, affiliations addresses and e-mail addresses of all authors. Authors are also required to submit their Open Researcher and Contributor ID (ORCID)'s which can be obtained from <http://orcid.org> as their URL address in the format <http://orcid.org/xxxx-xxxx-xxxx-xxxx>. Please indicate the corresponding author.

Abstract and Keywords: The abstract should state briefly the purpose of the research. The length of the Abstract should be between 50 to 5000 characters. At least 3 keywords are required.

Math Formulae: Formulas should be numbered consecutively in the parentheses ().

Tables: All tables must have numbers (TABLE 1) consecutively in accordance with their appearance in the text and a legend above the table. Please submit tables as editable text not as images.

Figures: All figures must have numbers (FIGURE 1) consecutively in accordance with their appearance in the text and a caption (not on the figure itself) below the figure. Please submit figures as EPS, PDF, TIFF or JPEG format.

Authors Contribution Statement, Declaration of Competing Interests and Acknowledgements should be given at the end of the article before the references.

References: The following format for the references should be used. Authors are urged to use the Communication.csl style (<https://dergipark.org.tr/en/download/journal-file/18514>) in Mendeley Desktop or Zotero automated bibliography. If manual entry is preferred for bibliography, then all citations must be listed in the references part and vice versa. Below, It has no relationship with the text, but can be used to show sample citations such as; for articles [1, 4], for books/booklets/theses [3], and for proceedings/conferences etc. [2].

[1] Demirci, E., Unal, A., Özalp, N., A fractional order SEIR model with density dependent death rate, Hacettepe J. Math. Stat., 40 (2) (2011), 287–295.

[2] Gairola, A. R., Deepmala, Mishra, L. N., Rate of approximation by finite iterates of q-Durrmeyer operators, Proc. Natl. Acad. Sci. India Sect. A Phys. Sci., 86 (2) (2016), 229–234.

[3] Lehmann, E. L., Casella, G., Theory of Point Estimation, Springer, New York, 2003.

[4] Özalp, N., Demirci, E., A fractional order SEIR model with vertical transmission, Math. Comput. Model., 54 (1-2) (2011), 1–6, <https://dx.doi.org/10.1016/j.mcm.2010.12.051>.

Peer-review policy: The Editor may seek the advice of two, or three referees, depending on the response of the referees, chosen in consultation with appropriate members of the Editorial Board, from among experts in the field of specialization of the paper. The reviewing process is conducted in strict confidence and the identity of a referee is not disclosed to the authors at any point since we use a single-blind peer review process.

Copyright: Copyright on any open access article in Communications Faculty of Sciences University of Ankara Series A2-A3: Physical Sciences and Engineering is licensed under a Creative Commons Attribution 4.0 International License (CC BY).

Declarations/Ethics With the submission of the manuscript authors declare that:

- All authors of the submitted research paper have directly participated in the planning, execution, or analysis of study;
- All authors of the paper have read and approved the final version submitted;
- The contents of the manuscript have not been submitted, copyrighted or published elsewhere and the visual-graphical materials such as photograph, drawing, picture, and document within the article do not have any copyright issue;
- The contents of the manuscript will not be copyrighted, submitted, or published elsewhere, while acceptance by the Journal is under consideration.
- The article is clean in terms of 'plagiarism', and the legal and ethical responsibility of the article belong to the author(s). Author(s) also accept that the manuscript may go through plagiarism check using iThenticate software;
- The objectivity and transparency in research, and the principles of ethical and professional conduct have been followed. Authors have also declared that they have no potential conflict of interest (financial or non-financial), and their research does not involve any human participants and/or animals.

Archiving: Research papers published in Communications Faculty of Sciences University of Ankara are archived in the Library of Ankara University (Volume 1-63) and Dergipark immediately following publication with no embargo.

Editor in Chief

Commun. Fac. Sci. Univ. Ank. Ser. A2-A3.

Ankara University, Faculty of Sciences

06100 Beşevler, ANKARA – TÜRKİYE

C O M M U N I C A T I O N S

FACULTY OF SCIENCES
UNIVERSITY OF ANKARA

DE LA FACULTE DES SCIENCES
DE L'UNIVERSITE D'ANKARA

Series A2-A3: Physical Sciences and Engineering

Volume: 65

Number: 2

Year: 2023

Research Articles

Ş. ÖZSARI, F.Z. ORTAK, M.S. GÜZEL, M.B. BAŞKIR, G.E. BOSTANCI, ML based prediction of COVID-19 diagnosis using statistical tests	79
H.A. ILGIN, F.A. AYDEMİR, B. CEDİMOĞLU, M.N. AYDIN, H. SİLLELİ, Comparative analysis of mature tomato detection by feature extraction and machine learning for autonomous greenhouse robots	100
S. SERTTAŞ, E. DENİZ, Disease detection in bean leaves using deep learning	115
U.N. AKTAN, M. DİKMEN, An optimized artificial neural network for estimating design effort of jigs and fixtures used in aviation industry	130
M.E. TUNALIOĞLU, H.Ö. ÇILDIROĞLU, A.U. YILMAZER, On the geometric phases in entangled states.....	142
B. ÖZBEY, A novel alternative in wireless and passive sensing: the bended nested split-ring resonator.....	152
M. KALKAN, Y. AR, Classification of human activities by smart device measurements.....	166