



A Deep Learning-Based Seed Classification With Mobile Application

YUSUF BASOL¹ , SINAN TOKLU² 

¹*Department of Computer Engineering, Faculty of Engineering, Duzce University, 81620, Duzce, Turkey.*

²*Department of Computer Engineering, Faculty of Technology, Gazi University, 06560, Ankara, Turkey.*

Received: 16-03-2021 • Accepted: 11-05-2021

ABSTRACT. Seed quality is an essential factor in agricultural production. Some seeds are inherently small so it is difficult to identify and classify differences between species. In the traditional method, these differences are classified by experts considering the morphological structure, texture and color. This method involves a classification process that is costly, subjective and time confusing, what makes it necessary to develop a process that can automatically detect the type of seeds. In this study, a mobile application has been developed that quickly detects and classifies seed images with high accuracy using CNN, one of the deep learning techniques.

2010 AMS Classification: 97R40, 68T45

Keywords: Deep learning, convolutional neural networks, seed classification, mobile application.

1. INTRODUCTION

Agriculture is an important sector for the economy on a global scale whereof various studies have been carried out to increase production efficiency over the years [17]. In the recent years, there is a notable need for advanced technologies in the agricultural sector to improve productivity, quality, and sustainability. The contribution of artificial intelligence studies in the agricultural sector has emerged because of this need.

Crop cultivation is through seeds within the agricultural sector. Without seeds it is not possible to produce or harvest [6]. Some seeds are inherently small so it is difficult to identify and classify differences between species. Within the traditional method, to identify and classify the seeds, the kind of the seed is decided by the experts by considering the morphological structure, shape, texture, and color of the seed. This method involves a classification process that is costly, of which decisions are subjective, and time-consuming. Therefore, it is necessary to develop an automated process that can quickly and accurately identify seed varieties. Such problems encountered in the identification and classification of seed species can be solved with deep learning by today's advanced machine learning technique.

Deep learning is one of the machine learning methods based on artificial neural networks and it is a modern approach successfully applied in various fields [11, 18]. Despite the increase in the number of layers and nodes in artificial neural networks, artificial intelligence methods could not be used in the 2000s due to hardware deficiencies [22]. However, today it has become popular again due to the decrease in computational costs thanks to the GPU and hardware developments [18]. With CNN, a specialized architecture of deep learning, it can be successfully applied to various classification problems, including seeds.

*Corresponding Author

Email addresses: basolyusuff@gmail.com (Y. Basol), stoklu@gazi.edu.tr (S. Toklu)

There are various studies that addresses subjects related to seeds by using deep learning. Recent studies have been conducted to perform seed classification of different crops, fruits, and vegetables by using machine learning techniques. Most of the studies have been carried out on a single seed type for different purposes such as seed classification, disease free seed detection and observation of growth stages. These studies are mentioned further in Section 2.

The number of studies that are carried out for the classification of multiple types of seeds is very limited. In addition, there is no system adapted to the mobile platform that allows the user to easily use the classified seeds. Therefore, there is a need for a highly accurate system that can overcome this deficiency using deep learning techniques.

In this study, an effective solution for seed identification has been produced by transforming the model trained with deep learning to a mobile application. A model was developed using deep learning libraries and python programming language from a dataset constructed from images taken of various seeds. The performances of the pre-trained InceptionV3, ResNet50, Xception, and InceptionResNetV2 CNN architectures on the dataset were monitored. The trained model with the highest performance has been converted for use in mobile devices. Thanks to the mobile application, it was ensured that the photographed or selected seed was identified and detailed information about its morphological structure, cultivation, and use was provided. More details on the work done are provided in Section 3. Parts of the study are described below.

Section 2 reviews previous work on this study. System structure and mobile application are explained in Section 3. Results and analyses are shown in Section 4. The conclusions are evaluated in Section 5.

2. RELATED WORK

There are multiple studies in the literature using artificial intelligence applications to solve various problems of the agricultural sector. The application of different crops, fruits, and vegetables with both traditional machine learning methods and deep learning techniques has been examined within these studies.

Gulzar et al. [6] created a dataset of 14 different seed species and a classification study was carried out using the CNN deep learning technique. The training process was carried out on the dataset created with the pre-trained VGG16 model and an accuracy of 99% was achieved. The aim is to make a mobile-based application by increasing the seed kind in future studies.

Verma et al. [23] developed a deep learning-based mobile application that can diagnose diseases in tomato plants. The training process was carried out with the PlantVillage dataset and pre-trained InceptionResNetV2, ResNet50, VGG16, VGG19, Xception CNN architectures. At the end of the training, the accuracy was 92.14% in the Inception-ResNetV2 model, 96.14% in the ResNet50 model, 92.67% in the VGG16 model, 92.38% in the VGG19 model, and 91.94% in the Xception model. The highest accuracy ResNet50 model was used in the mobile application. An android mobile application that enables the identification of a healthy-diseased plant with a plant leaf image taken as input has been made available for disease detection in the agricultural sector.

Foysal et al. [5] conducted a classification study based on deep convolutional neural networks (DCNN) using a dataset consisting of bacterial spot, late blight, leaf spot, spider mites, leaf curl virus disease images, and healthy tomato leaves images that can occur in tomatoes. After the training procedures, the proposed model achieved an accuracy of 76%.

Lammie et al. [11] developed a deterministically binary accelerated system with GPU and field programmable gate array (FPGA) using a public DeepWeeds dataset of weed images. FPGA-accelerated binary networks have been shown to reduce power consumption by 7 times, perform significantly better, and infer faster in weed images than their GPU-accelerated counterparts.

Koklu and Ozkan [10] developed a computer vision system to identify 7 different registered bean varieties with similar characteristics. The dataset was created by taking images of 7 different dried beans with a high resolution camera and the interface was designed using the MATLAB program. On four different classification models, 91.73% accuracy for multilayer perceptron (MLP), 93.13% for support vector machine (SVM), 87.92% for k-nearest neighbors (KNN) and 92.52% for decision tree (DT) were obtained. The SVM model with the highest results has mostly met the expectations of producers and customers to obtain uniform bean varieties. It has been suggested that the resulting application can be used more effectively and practically by moving it to the mobile platform.

Keya et al. [8] provided identification of rice, corn, sweet squash, beeswax, and gourd seeds by CNN technique. He created a dataset containing 1250 images in total, taking 250 images of each seed. As a result, 87% -89% training accuracy was obtained.

Ali et al. [1] proposed a system to classify 6 kinds of corn seed using machine learning approaches such as random forest (RF), bayes network (BN), LogitBoost (LB) and MLP. It was observed that the applied approaches performed well by achieving 97.22% accuracy with the RF technique, 97.67% accuracy with the BN technique, 97.78% accuracy with the LB technique and 98.93% accuracy with the MLP technique.

Kiratiratanapruk et al. [9] examined 14 different rice varieties in terms of preprocessing, feature extraction and classification. Firstly, seed orientation was made that aligns the seed bodies in the same direction. As second a quality screening method was used to detect unusual seed samples. As last, for classification purposes an attribute extraction of rice is performed based on physical attributes such as shape, color, and texture. Classification performances were compared using deep learning and statistical machine learning methods. According to the results, it was seen that the deep learning method performed better than the traditional method.

Salimi and Boelt [16] performed a seed quality assessment using the multispectral imaging (MSI) technique to classify five different damaged seed kinds in sugar beet seeds. The classification model, based on information derived by MSI, classified five different damaged seeds with an overall accuracy of 82%.

Prominent studies are shown in Table 1 in the literature [1, 5, 6, 8–11, 16, 23].

TABLE 1. Prominent studies

Area	Crop	Data Used	Method and Accuracy	Mobile App	Ref
Plant Disease Detection	Tomato	18160 images	InceptionResNetV2: 92.14% ResNet50: 96.14% VGG16: 92.67% VGG19: 92.38% Xception: 91.94%	Yes (Android)	[23]
	Tomato	3000 images	DCNN: 76%	No	[5]
Weed Classification	8 kinds of weeds	17508 images	DenseNet: 90.08%	No	[11]
Seed Classification	Rice (14 kinds)	+3500 images	InceptionResNetV2: 95.14% Xception: 93.82% VGG16: 89.45% VGG19: 90.94% InceptionV3: 85.32% KNN: 58.69% LR: 81.12% LDA: 79.77% SVM: 83.9%	No	[9]
	Dry beans (7 kinds)	13611 images	MLP: 91.73% SVM: 93.13% KNN: 87.92% DT: 92.52%	No	[10]
	Corn (6 kinds)	330000 images	MLP: 98.83% LB: 97.78% RF: 97.22% BN: 96.67%	No	[1]
	14 kinds of seeds	2733 images	VGG16: 99%	No	[6]
	5 kinds of seeds	1250 images	CNN: 87%-89%	No	[8]
	Sugar beet (5 kinds)	200 images	MSI: 82%	No	[16]

3. PROPOSED METHODOLOGY

The proposed model is shown in Figure 1. In the first stage, images of 15 different seeds were taken at high resolution and the dataset was created by labeling them appropriately. In the second stage, the training process was carried out with pre-trained CNN models and the best results were recorded. In the third stage, the model results were compared and the transformation process was made to make it applicable on the mobile platform. In the last stage,

the mobile application interface was designed and after the seed was identified by the application, detailed information about the morphological characteristics and use of the seed was provided to the user. These four stages are explained in detail in the sections below.

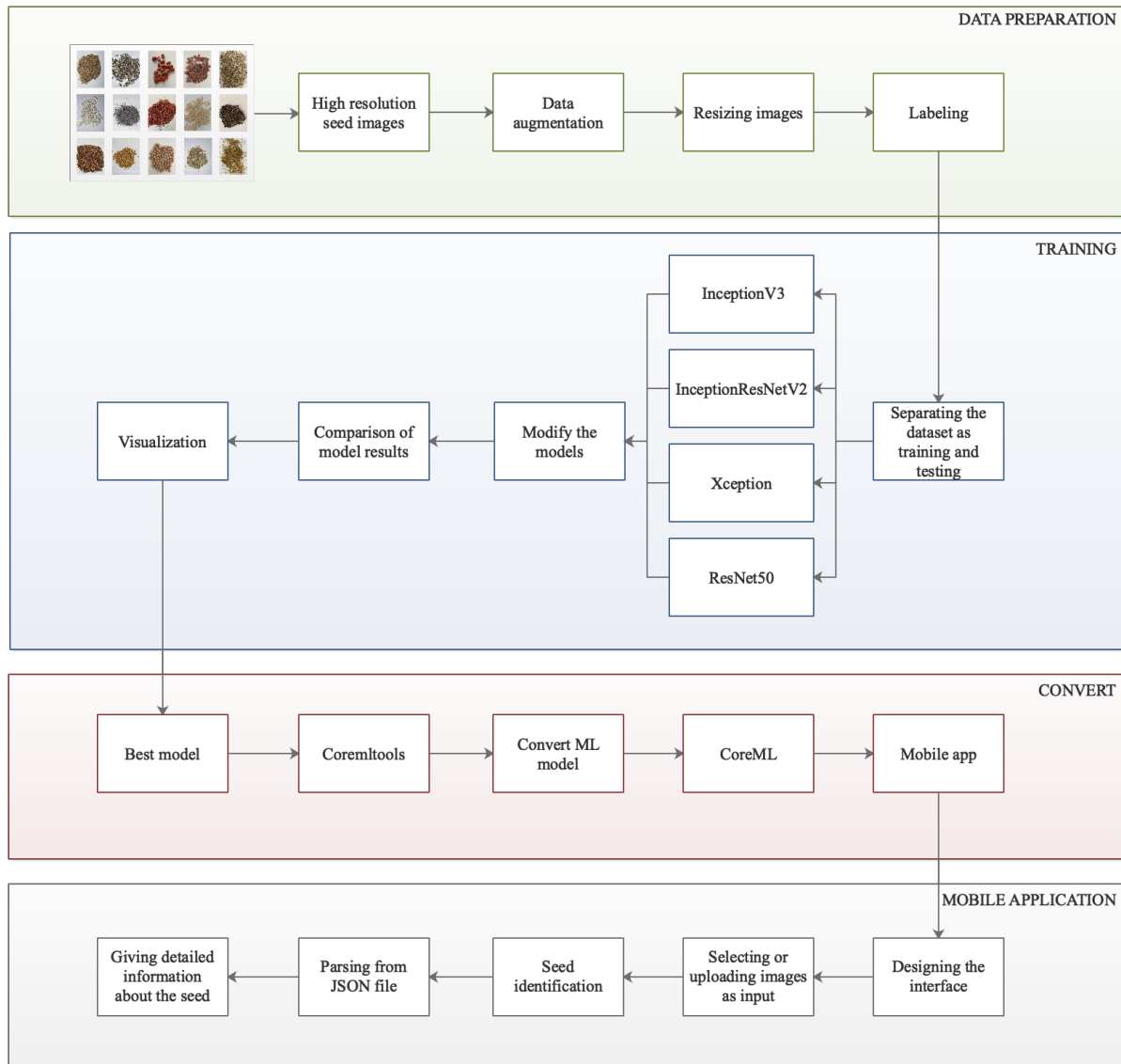


FIGURE 1. Proposed model

3.1. Dataset and Training. In this study, 15 commonly known seed species were selected to create the dataset, and the dataset was created by taking high resolution images. The dataset consists of anise, sunflower, broad bean, pea, rosemary, bean, poppy, spinach, pumpkin, black pepper, flax, corn, chickpea, fennel, and clover. Figure 2 shows sample seed pictures from the dataset.

The dataset is divided into 80% training data and 20% test data. The dataset contains a total of 3000 seed images, including 2400 images training data and 600 images test data. The images in the dataset are resized according to the pre-trained CNN models to be used (299x299x3 and 224x224x3) and labeled appropriately.

Seed images were captured in high resolution using a smartphone (iPhone 11). In order to avoid unbalanced classification, a dataset was created with 200 images from each seed class.



FIGURE 2. Seed images from the Seed15 dataset

There are many pre-trained CNN models within Keras. In this study, ResNet50, Inceptionv3, Xception, Inception-ResNetV2 pre-trained models were selected, and then the training processes were carried out by modifying the models. In addition, the accuracy rates of the models used in the Seed15 dataset were compared.

The training process started by adding appropriate modifications and additional layers to pre-trained models. The images were sized as 224x224 for the ResNet50 model, and 299x299 for the Inceptionv3, Xception, InceptionResNetV2 models. In the average pooling layer, the pool size was set to 4x4 for the ResNet50 model and 8x8 for the Inceptionv3, Xception, InceptionResNetV2 models. A 40% dropout was applied in the dropout layer to avoid memorization. In the flattening layer, neurons were flattened by Softmax activation function. Finally, a classification layer was added to represent the output of 15 classes.

In the model training process, the image was increased to prevent overfitting. 20% random horizontal and vertical panning, 20% magnification and random rotation of images were applied to the images.

Activation functions are used for transformation processes in multilayer neural networks [7]. The output of hidden layers is normalized by some activation functions. In this study, Softmax function, which produces output between [0,1] in the classification of many seed varieties, has been preferred for multiple classification. The expression of the softmax function is shown in Equation (3.1).

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (3.1)$$

There are many optimization algorithms used in deep learning applications [7]. In this study, the stochastic gradient descent (SGD) optimization algorithm was used and the learning rate parameter was selected as 1%, and the momentum parameter, which reduces the oscillation and accelerates the SGD in the relevant direction, was selected as 90%. In addition, the learning rate parameter has been adjusted to decrease over time so that the model does not make excessive leaps and miss global optimum. The learning rate is updated to 0.01 for the first five cycles, 0.002 for the number of five to ten cycles, and 0.0004 for the number of ten and fifteen cycles.

3.2. CNN Architectures. In this section, a short explanation is given about the CNN architectures used in training.

InceptionV3 was developed by a team at Google and is also called GoogleNet [19]. InceptionV3 is an improved version of InceptionV1 and InceptionV2 [20]. The image input size of the network is 299x299. Although inspired by

the LeNet5 architecture, a new element, also known as the inception module, has been implemented in InceptionV3 to reduce the number of parameters [4]. It performed well in detecting in the ILSVRC dataset [14].

ResNet50 is a convolutional neural network with a depth of 50 layers consisting of residual modules based on residual learning [24]. The image input size of the network is 224x224. Since optimizing classical CNNs consisting of convolution, pooling and fully connected layers often causes the lost gradient problem [2], a residual block has been introduced in order to reduce the difficulty in training the neural networks with high depth. Later, a team at Google combined the GoogleNet model with ResNet and created the InceptionResNetV2 model [12].

InceptionResNetV2 is a hybrid model [20]. Residual learning and inception blocks have been combined [21]. The image input size of the network is 299x299. It provided good accuracy in image classification comparison in the ILSVRC dataset.

Xception was developed by Francois Chollet as a new CNN model [12]. The image input size of the network is 299x299. Xception proposed a deeply decomposable convolution layer in order to make the network computationally efficient by decoupling the spatial and channel correlations [21]. The proposed model showed good results in the JFT data set [3].

3.3. Conversion the Model and Mobile Application. After the training process was completed, the models that gave the best results in the models made with ResNet50, Inceptionv3, Xception and InceptionResNetV2 were transformed. The python library coremltools was used to obtain the model (model.mlmodel) that can be used in the mobile application. In this way, the trained model file (model.hdf5) has been converted to mobile compatible CoreML form.

A suitable interface has been designed to use the transformed model in a mobile application. The interface of the mobile application is shown in Figure 3. In the first stage, a function has been added to import an image or use a captured image as input. In the second step, the input image is sent to the transformed model. The first three highest outputs from the model were obtained, and the estimation was shown to the user as a percentage. In the third step, a javascript object representation (JSON) file containing the information of all seeds was created and a page that provides detailed information about the seed was designed for the highest result. In this way, it is aimed to give the user detailed information about the morphological properties and use of the seed.

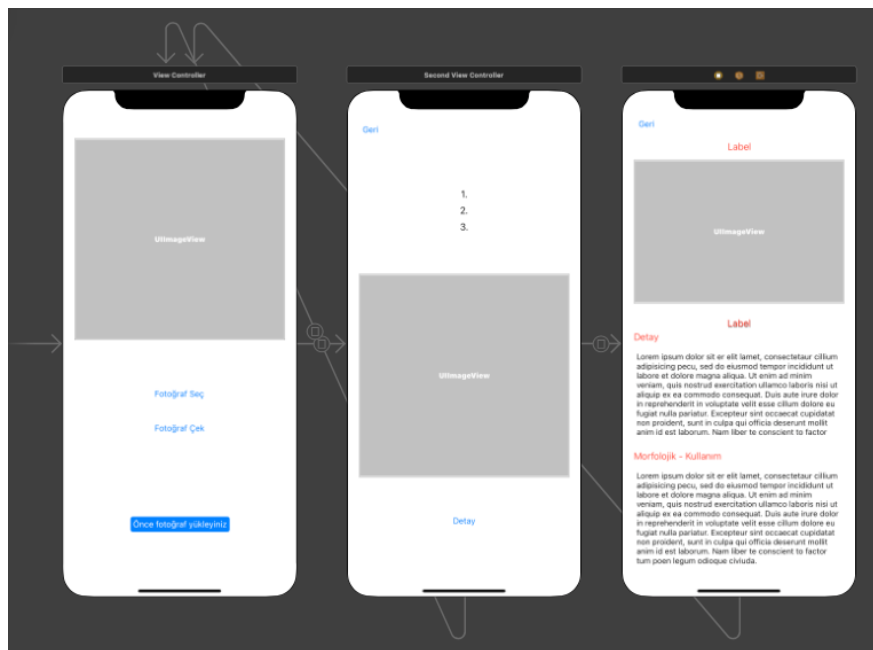


FIGURE 3. Mobile application general design

4. RESULT AND DISCUSSION

In the study, the accuracy values obtained from the models were compared. While Inceptionv3, Xception and InceptionResNetV2 architectures achieved 99% accuracy on the Seed15 dataset, 98% accuracy was achieved with the ResNet50 architecture. The accuracy values of the trained models are shown in Figure 4.

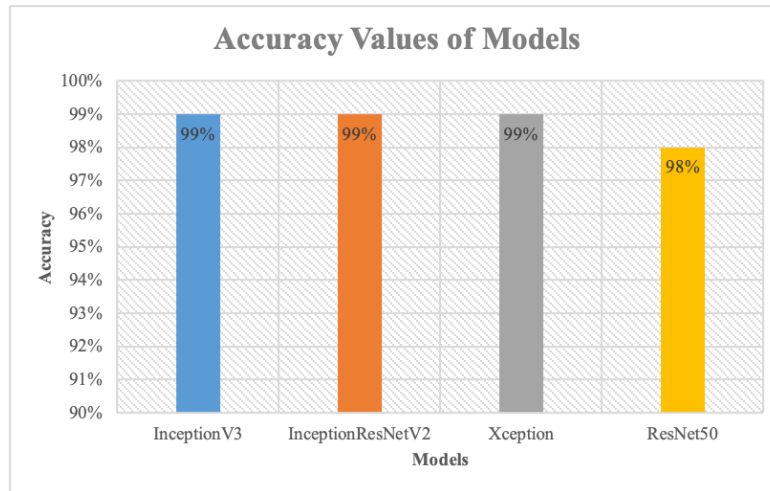


FIGURE 4. Accuracy values of the models

In the case of the value of epoch is the same, it has been observed that the accuracy values of the ResNet50, Inceptionv3, Xception, and InceptionResNetV2 models are increased. It was observed that the Xception model started with 88% accuracy, the InceptionResNetV2 model with 61% accuracy, the InceptionV3 model with 37% accuracy, the ResNet50 model with an accuracy close to 1%, and after the seventh iteration, there was no notable increase in the accuracy. Also, since the loss value approached the minimum value after the tenth iteration, it was not continued after the fifteenth iteration. The observation results are shown in Figure 5.

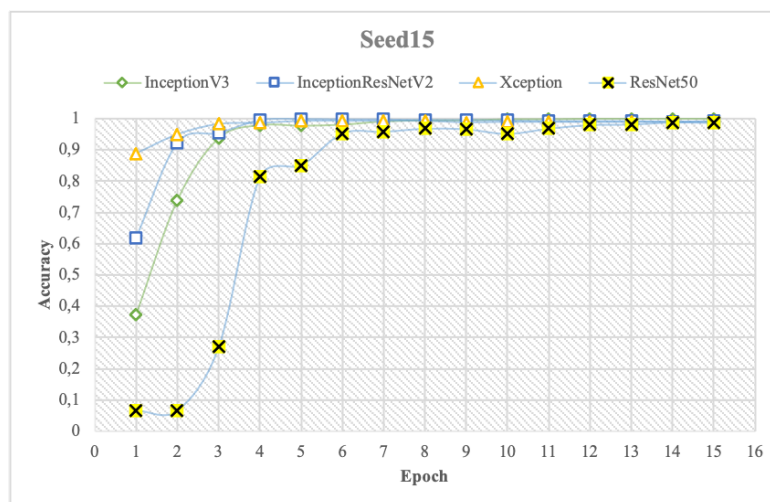


FIGURE 5. Comparison of models

The confusion matrices where the estimates and the actual values are compared were removed to evaluate the performance of the classification models. Classification estimates have one of four evaluations: true positive (TP), true

negative (TN), false positive (FP), false negative (FN). Examples where TP real value and predicted value are correct, TN real value is wrong and prediction is wrong, FP are examples where real value is wrong and predicted value is correct, FN are examples where real value is correct and predicted value is wrong. The confusion matrices of the ResNet50, Xception, Inceptionv3 and InceptionResNetV2 models have been removed separately. Comparison of the confusion matrix of the models is shown in Figure 6.

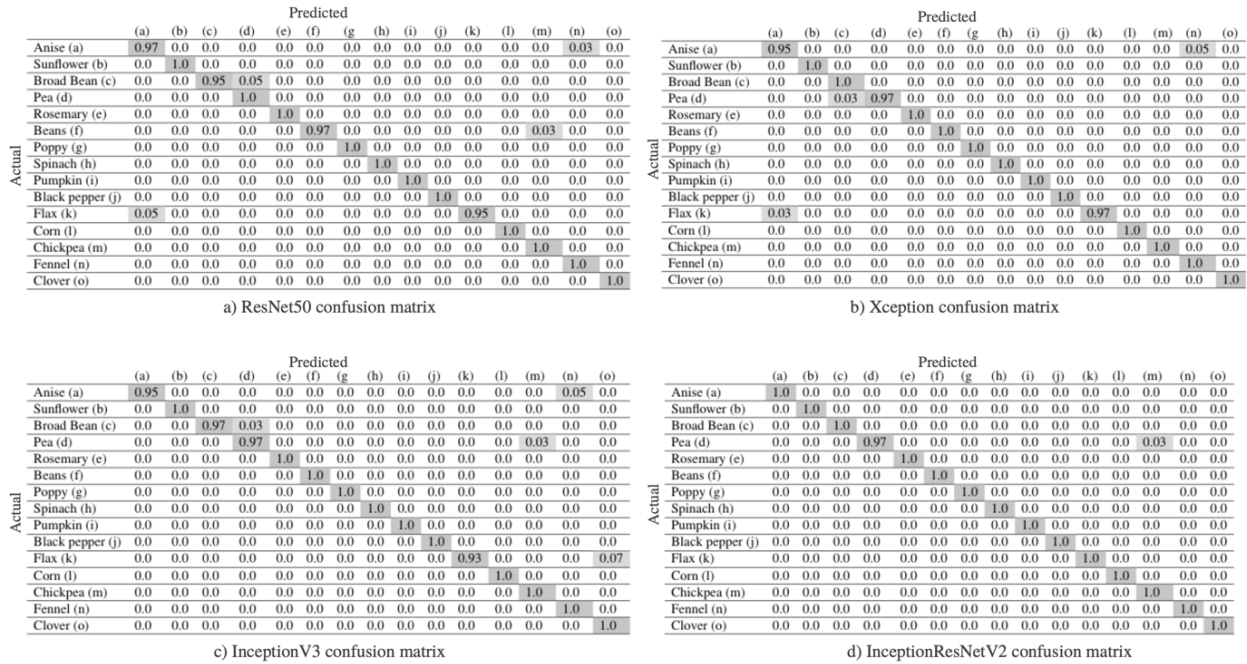


FIGURE 6. Confusion matrix comparison of models

In the study, accuracy rate, precision, recall, f1-score, and support metrics were extracted to check the performance of the proposed models. The support shows the number of samples of each class performed during the training of the model [6]. Accuracy shows the ratio of correctly estimated samples in the model to the total dataset [13]. Equation (4.1) shows the mathematical expression of accuracy.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

Precision shows how many of the values positively predicted in the model are actually positive [13]. Equation (4.2) shows the mathematical expression of precision.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

Recall shows how many of the samples that should be positively predicted in the model are positively predicted [13]. Equation (4.3) shows the mathematical expression of recall.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

The f1-score (4.4) shows the harmonic mean of the precision and recall values of the model [15]. Equation (4.4) shows the mathematical expression of f1-score.

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

The precision, recall and f1-score values of the ResNet50, Xception, Inceptionv3 and InceptionResNetV2 models are shown in Figure 7.

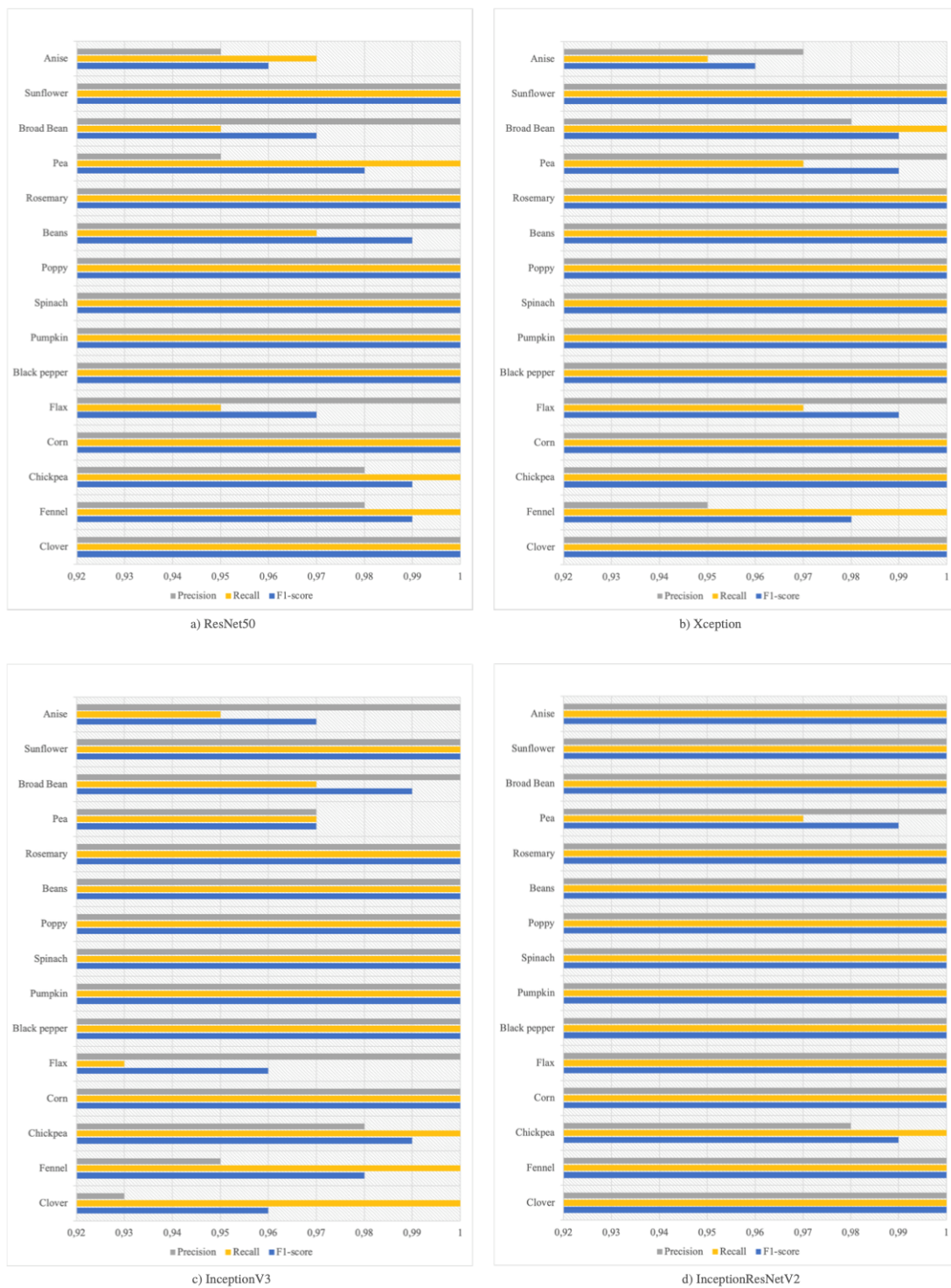


FIGURE 7. Precision, recall and f1-score values of the models

In Table 2, where ResNet50, Xception, Inceptionv3 and InceptionResNetV2 models are together, accuracy, macro average, weighted average, precision, recall, f1-score and support values are shown.

TABLE 2. ResNet50 (a), Xception (b), Inceptionv3 (c) and InceptionResNetV2 (d) classification report

	Precision				Recall				F1-score				Support (a,b,c,d)
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)	
Anise	0.95	0.97	1	1	0.97	0.95	0.95	1	0.96	0.96	0.97	1	40
Sunflower	1	1	1	1	1	1	1	1	1	1	1	1	40
Broad Bean	1	0.98	1	1	0.95	1	0.97	1	0.97	0.99	0.99	1	40
Pea	0.95	1	0.97	1	1	0.97	0.97	0.97	0.98	0.99	0.97	0.99	40
Rosemary	1	1	1	1	1	1	1	1	1	1	1	1	40
Beans	1	1	1	1	1	1	1	1	1	1	1	1	40
Poppy	1	1	1	1	0.97	1	1	1	0.99	1	1	1	40
Spinach	1	1	1	1	1	1	1	1	1	1	1	1	40
Pumpkin	1	1	1	1	1	1	1	1	1	1	1	1	40
Black pepper	1	1	1	1	1	1	1	1	1	1	1	1	40
Flax	1	1	1	1	0.95	0.97	0.93	1	0.97	0.99	0.96	1	40
Corn	1	1	1	1	1	1	1	1	1	1	1	1	40
Chickpea	0.98	1	0.98	0.98	1	1	1	1	0.99	1	0.99	0.99	40
Fennel	0.98	0.95	0.95	1	1	1	1	1	0.99	0.98	0.98	1	40
Clover	1	1	0.93	1	1	1	1	1	1	1	0.96	1	40
Accuracy									0.98	0.99	0.99	0.99	600
Macro average	0.98	0.99	0.99	0.99	0.98	0.99	0.99	0.99	0.98	0.99	0.99	0.99	600
Weighted average	0.98	0.99	0.99	0.99	0.98	0.99	0.99	0.99	0.98	0.99	0.99	0.99	600

It is necessary to evaluate the performance of the models and to check them before using them in the mobile application. The results showed that the models performed well in terms of precision, recall, and f1-score in most of the classes.

4.1. Mobile Application Assessment. The mobile application interface is designed to consist of three pages. The interface of the mobile application is shown in Figure 8. In the first page, two buttons have been added to select the image as input or to load a captured image. The second page is displayed by adding the Find button. The process of selecting the image as input runs in an average of 0.168 seconds, and the process of loading the input image takes an average of 0.194 seconds.

On the second page, the incoming photo has been captured and sent to the function to detect the seed from the model file. The first three highest estimates were taken from the function and the user was shown as a percentage. The third page is displayed by adding the detail button. The process of capturing the image and detecting the seed from the model file takes an average of 0.533 seconds.

On the third page, the parsing process is made from the JSON file, which contains all the seed information for the seed that is the model's highest estimate. The user was given detailed information about the morphological characteristics and use of the seed detected. Parsing the information of the highest predicted seed from the JSON file takes 0.002 seconds on average.

It has been seen that the functions perform these operations in a short time in terms of the running time of the functions in loading the seed image as input, detecting and parsing it from the JSON file. Thanks to the mobile application, the classification process was carried out at low cost, without subjective decisions and without spending much time. In addition, it has been shown that seed varieties can be applied in an automatic system that can determine quickly and with high accuracy.

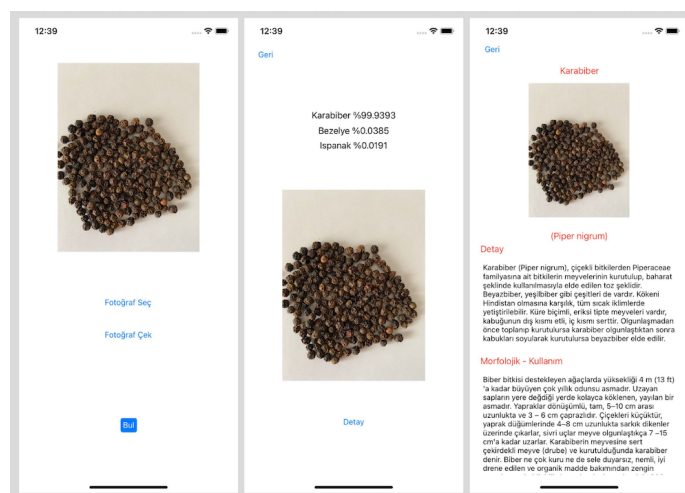


FIGURE 8. Mobile application interface

5. CONCLUSION

In this study, a mobile application has been developed that detects and classifies seed images with high accuracy using deep learning techniques. Using CNN, one of the deep learning techniques, a data set consisting of 15 types of seed images was created and the training process was carried out. The performances of the trained models were compared on the created dataset. Inceptionv3, Xception and InceptionResNetV2 models achieved 99% accuracy, and ResNet50 model 98% accuracy.

The model, which works with high accuracy, has been enabled to work on a mobile platform. It takes an average of 0.533 seconds for the image to detect the seed from the model file in the mobile application. Thanks to the mobile application, it has been realized that it can be used by anyone without the need for an expert to identify and classify seeds quickly and accurately. In addition, detailed information about the morphological characteristics and use of the seed detected was presented to the user with the mobile application. It has been ensured that the problems encountered in the identification and classification of seed species are realized in a short time, with low cost and without subjective decisions. In this way, the real-life applicability of an automatic system that can identify seed types has been demonstrated. In future studies, it is planned to make a more comprehensive application by increasing the number of seeds. It is aimed to develop smart decision-making systems by combining real-time data from sensors with Internet of Things (IoT)-based and deep learning-based applications. Thus, it can be ensured that a smart decision system is included in the mobile or web environment by monitoring the real-time environmental parameters.

CONFLICTS OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of this article.

REFERENCES

- [1] Ali, A., Qadri, S., Mashwani, W.K., Brahim, B.S., Naeem, S., et al., *Machine learning approach for the classification of corn seed using hybrid features*, Int. J. Food Prop., (2020), 1110–1124.
- [2] Bengio, Y., Simard, P., Frasconi, P., *Learning long-term dependencies with gradient descent is difficult*, IEEE Trans. Neural Networks, **5**(2)(1994), 157–166.
- [3] Chollet, F., *Xception: deep learning with depthwise separable convolutions*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), 1800–1807.
- [4] Dourado, C.M.J.M., da Silva, S.P.P., da Nobrega, R.V.M., Antonio, A.C., Filho, P.P.R., et al., *Deep learning IoT system for online stroke detection in skull computed tomography images*, Comput. Networks, **152**(2019), 25–39.
- [5] Ferdouse, A.F.M., Shakirul, I.M., Abujar, S., Akhter, H.S., *A novel approach for tomato diseases classification based on deep convolutional neural networks*, Proceedings of International Joint Conference on Computational Intelligence, (2020), 583–591.
- [6] Gulzar, Y., Hamid, Y., Soomro, A.B., Alwan, A.A., Journaux, L., *A convolution neural network-based seed classification system*, Symmetry 2020, **12**(12)(2020).

- [7] Kayıkçı, Ş., Başol, Y., Dörter, E., *Classification of turkish cuisine with deep learning on mobile platform*, UBMK 2019 - Proceedings, 4th Int. Conf. Comput. Sci. Eng., (2019), 296–300.
- [8] Keya, M., Majumdar, B., Islam, M.S., *A robust deep learning segmentation and identification approach of different bangladeshi plant seeds using CNN*, 11th International Conference on Computing, Communication and Networking, (2020), 1–6.
- [9] Kiratiratanapruk, K., Temniranrat, P., Sinthupinyo, W., Prempre, P., Chaitavon, K., et al., *Development of paddy rice seed classification process using machine learning techniques for automatic grading machine*, Journal of Sensors, (2020), 1–14.
- [10] Koklu, M., Ozkan, I.A., *Multiclass classification of dry beans using computer vision and machine learning techniques*, Computers and Electronics in Agriculture, **174**(2020).
- [11] Lammie, C., Olsen, A., Carrick, T., Rahimi, A.M., *Low-power and high-speed deep FPGA inference engines for weed classification at the edge*, IEEE Access, (2019).
- [12] Långkvist, M., Karlsson, L., Loutfi, A., *A review of unsupervised feature learning and deep learning for time-series modeling*, Pattern Recognition Letters, **42**(1)(2014), 11–24.
- [13] Powers, D.M.W., *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*, ArXiv abs/2010.16061, (2020).
- [14] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., et al., *ImageNet large scale visual recognition challenge*, International Journal of Computer Vision, **115**(3)(2015), 211–252.
- [15] Salas, A.H., Morzán-Samamé, J., Nunez-del-Prado, M., *Crime alert! crime typification in news based on text mining*, Lecture Notes in Networks and Systems, **69**(2020), 725–741.
- [16] Salimi, Z., Boelt, B., *Classification of processing damage in sugar beet (Beta vulgaris) seeds by multispectral image analysis*, Sensors (Switzerland), **19**(10)(2019).
- [17] Santos, L., Santos, F.N., Oliveira, P.M., Shinde, P., *Deep learning applications in agriculture: a short review*, Robot 2019: Fourth Iberian Robotics Conference Advances in Intelligent Systems and Computing, **1092**(2020), 139–151.
- [18] Schmidhuber, J., *Deep learning in neural networks: an overview*, Neural Networks, **61**(2015), 85–117.
- [19] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., et al., *Going deeper with convolutions*, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., (2015), 1–9.
- [20] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., *Rethinking the inception architecture for computer vision*, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., (2016), 2818–2826.
- [21] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A., *Inception-v4, Inception-ResNet and the impact of residual connections on learning*, 31st AAAI Conf. Artif. Intell., (2017), 4278–4284.
- [22] Şeker, A., Diri, B., Balık, H.H., *A review about deep learning methods and applications*, Gazi Mühendislik Bilim. Dergi., **3**(3)(2017), 47–64.
- [23] Verma, S., Chug, A., Singh, A.P., Sharma, S., Rajvanshi, P., *Deep learning-based mobile application for plant disease diagnosis*, Applications of Image Processing and Soft Computing Systems in Agriculture, (2019), 242–271.
- [24] Wu, S., Zhong, S., Liu, Y., *Deep residual learning for image steganalysis*, Multimedia Tools and Applications, **77**(2017), 10437–10453.