Research Article GU J Sci 37(1): 137-148 (2024) DOI: 10.35378/gujs.1203685

Gazi University

Journal of Science

http://dergipark.gov.tr/gujs

Detection of Bovine Species on Image Using Machine Learning Classifiers

Ali Tezcan SARIZEYBEK[1,*], Ali Hakan ISIK[2]

[1]*Burdur Mehmet Akif Ersoy University, Graduate School of Natural and Applied Sciences, 15000, Burdur, Türkiye*
[2]*Burdur Mehmet Akif Ersoy University, Faculty of Engineering and Architecture, 15000, Burdur, Türkiye*

Highlights

- This paper focuses on the classification process for cattle breeds.
- In the study, a hybrid approach using machine learning algorithms on image data is proposed.
- The methods used were compared and the best among the suggested methods was selected.
- A highly precise and efficient classification accuracy were obtained.

Article Info

Received: 14 Nov 2022
Accepted: 06 Mar 2023

Keywords

SMOTE
VGG16
XGBoost
Random forest
Cow classification

Abstract

There are too many cattle in the world and too many breeds of cattle. For someone who is new to cattle breeding, it may be difficult to tell which species their cattle are. In some cases, even an experienced person may not understand the breeds of two cattle that are similar in appearance. In this study, the aim is to classify the cattle species with image processing methods and mobile applications written in Flutter and TensorFlow Lite. For classifying breeds, The VGG-16 algorithm was used for feature extraction. XGBoost and Random Forest algorithms were used for classification and the combined versions of the two methods were compared. In addition, SMOTE algorithm and image augmentation algorithms were used to prevent the imbalance of the dataset, the performance results of the combined versions of the two methods were compared. Images of different cattle species from different farms were obtained and the dataset was prepared, different image processing models were trained, the trained models were tested and the performance analyses were made. As a result of performance tests, it is obtained that the best model is VGG16+Random Forest+SMOTE+Augmentation with 88.77% accuracy result for this study. In the mobile application, first the cattle is detected with a pre-trained object detection model, and then the breed classification of the cattle on the image is made with image classification model.

1. INTRODUCTION

There are too many cattle breeds in the world, some of these cattle species’ appearance is similar, but they have different features. It is hard to classify the cattle species without information about cattle breeds, so species estimation could be made with image processing methods so that people who do not know about how to identify cattle species might make this distinction. For this, a mobile application should be made on a mobile platform, so everyone could use it, then models should be run on this prepared mobile application. This application decided to be written in a cross-platform programming language, Flutter, so more users may use the application. The platform to be used for classification was chosen as TensorFlow Lite which was produced for mobile platforms, which has low model sizes and short response time. A dataset was prepared by taking the images of the 10 most common cattle breeds in Turkey and different classification models were trained with this dataset. After the training of the models, it was put into performance tests and the performance analyses were conducted. After the model was trained, the proposed model was added to the application prepared on the phone, and the trained image classification model was also added to the application. After the cattle were detected with the object detection model, the species classification was made with the prepared image classification model.

*Corresponding author, e-mail: atsarizeybek@mehmetakif.edu.tr

In the model to be created in this study, XGBoost and Random Forest algorithms were used as the classification algorithm and VGG16 network was used as the feature extractor. The use of these two methods together has not been combined in the field of image processing before in the literature, so this combination has an important place in terms of the originality of the study. SMOTE algorithm was used to balance the number of images per class of the specially prepared dataset; Since the SMOTE algorithm is mostly used for machine learning, it is thought that its use in the field of image processing in this study will make a big difference among other studies. Data augmentation methods were applied to balance the number of images in the same dataset and the performance results of the image classification method used in the study were compared. Some preprocessing methods were applied to prepare the dataset for image classification training. With the CLAHE algorithm, which is one of the preprocessing stages used, the histogram is equalized and contrast and brightness adjustments are provided on the image data. In short, VGG16+XGBoost, VGG16+XGBoost+SMOTE, VGG16+XGBoost+Augmentation, VGG16+XGBoost+SMOTE+Augmentation, VGG16+Random Forest, VGG16+Random Forest+SMOTE and VGG16+Random Forest+SMOTE+Augmentation methods were used in performance comparison.

1.1. Related Work

According to the literature view, in the study by Santony, Sensuse, Arymurthy and Fanamy (2015), cattle breeds were successfully classified by gray level co-occurrence matrix(GLCM-CNN), a dataset was prepared with 775 images containing 5 different cattle species. Dataset was divided into 70% training and 30% test. Training data has 620 and test data has 155 images. In the result of the experiment, the model trained with images which applied GLCM-CNN accuracy is 93.763% [1].

For human face classification, in the study by Ou, Wu, Qian and Xu (2005), the aim is making the classification of Asian and non-Asian people's faces. In the study, PCA was used for feature generation and ICA used for feature extraction. Support Vector Machines(SVM) were used for training. Dataset contains 750 human faces which is Asian and non-Asian [2].

In the study conducted by Sutojo et al. (2017), it was aimed to classify the cattle species by extracting color characteristics. The proposed method includes changing the background color, transforming the color space and performing resizing operations in the preprocessing stage. After the preprocessing stage, the texture features are extracted with GLCM (Gray Level Occurence Matrix), then the comparison of the images in the training dataset and the input image with the Euclidean Distance, and finally the calculation of accuracy, precision and recall scores over the confusion matrix with the CBIR and Euclidean Distance methods. In the study, a total of 100 training and 20 test image data, 20 training and 4 test images of each cattle, were used as data set, and these methods were passed through the recommended preprocessing stage. Then, feature extractions were made with GLCM and similarity calculations were made with Euclidean distance. As a result of the test, an average of 95% accuracy, 100% precision and 100% recall were obtained [3].

In the study by Bello et al. (2020), it was aimed to detect cattle separately, that is, individually. This classification was carried out using CNN. As a data set, 10 types of cattle were examined and a total of 1000 images were obtained by taking 100 images from each. 400 images were reserved for training and 600 images were reserved for testing. In the preprocessing stage, the contrasts in the images were improved by using Gaussian Filter and CLAHE (Constrast Limited Adaptive Histogram Equalization). In the results obtained using the RBM-based Deep Belief Network (DBN), the accuracies were obtained as 92.59% and 89.95% [4].

In the study by Jwade, Guzzomi and Mian (2019), the aim is making the breed classification with image classification methods. 1642 sheep images with 4 different breeds were used for the dataset. Two different transfer learning methods were used to determine the best method for sheep breed classification. One is the six-layers of fine-tuned VGG-16, and the other is the pre-trained VGG-16 method with SVM on top. Facial and whole body images of sheep are used, noise and different preprocessing effects are applied. Fine-tuned VGG16 for 10 epochs had the best classification accuracy of 95.38% with 1.7 standard deviation [5].

Considering the use of the methods to be used in this study in the literature: Almeida et al. In this study, it was aimed to detect breast cancer on mammography images, and for this purpose, a comparison of VGG16, which includes CNN architecture, and XGBoost, which is a Gradient Boosting algorithm, was made. The CBIS-DDSM dataset, consisting of a total of 10239 images collected from 1566 patients, was used. Since 3568 images had ROI areas which are not related to the study, the images were removed from the dataset, leaving a total of 6671 images. As a result, XGBoost performed much better than VGG16. Considering the use of the methods to be used in this study in the literature: Almeida et al. In this study, it was aimed to detect breast cancer on mammography images, and for this purpose, a comparison of VGG16, which includes CNN architecture, and XGBoost, which is a Gradient Boosting algorithm, was made. The CBIS-DDSM dataset, consisting of a total of 10239 images collected from 1566 patients, was used. Since 3568 images had ROI areas not related to the study, the images were removed from the dataset, leaving a total of 6671 images. As a result, XGBoost performed much better than VGG-16 [6].

2. MATERIAL METHOD

In this section, the algorithms that used, dataset and the mobile application prepared for this study is discussed.

2.1. Tensorflow Lite

TensorFlow Lite is a framework which was developed for use in mobile and embedded systems. Its lower size and shorter response time compared to TensorFlow models enable its use on mobile devices. In order to be portable, the feature range is narrowed according to TensorFlow. Flat buffer is used instead of Protocol Buffers, which is the data serialization format in TensorFlow models. TensorFlow models may be converted to TensorFlow Lite models with TFLite Model Converter [7]. CPU, GPU and TPU support is available for mobile and embedded devices. At the same time, it may be run on a desktop device, although it is not as efficient.

2.2. VGG-16

The VGG-16 network is used for feature extraction in the model. VGG-16 was presented to the literature in 2014, it is a CNN architecture using Imagenet infrastructure [8, 9]. The meaning of 16 in VGG-16 indicates that 16 layers are convolutional in the architecture [10]. Even if the dataset is too small, good performance results could be obtained. It consists of 16 convolutional networks with 3x3 receiver areas and 5 networks consisting of a 2x2 Max Pooling layer [11]. In the last layers, there are 2 Fully Connected layers and finally a Softmax layer, which has approximately 138 million parameters. On a heavy dataset like Imagenet, which contains 1000 classes and 14 million image data, 92.7% accuracy was obtained [12, 13]. The reason for using VGG-16 for feature extraction in this study is to reduce the size of the trained model and to expect more successful performance results. The reason for not using VGG19 is that since there are about 3000 image data in the dataset, it is known that the VGG-19 algorithm, which added 3 more convolutional layers, will not make much difference against VGG-16.

2.3. Random Forest

Random Forest algorithm was added to the literature by Breiman in 2001 [14, 15]. Random Forest Algorithm is an ensemble learning algorithm developed for classification and regression [16, 17], It is based on decision trees such as XGBoost [18]. It consists of numerous decision trees [19], and the classification result is obtained according to the classification result of the majority of the classification trees [20].

2.4. XGBoost

Extreme Gradient Boosting, namely XGBoost, is a decision tree-based supervised boosting algorithm added to the literature by Chen and Guestrin in 2016 [21]. Since it performs cross-validation in itself, it does not need to set hyperparameters such as epoch number [22]. It eliminates the overfitting problem by making regularization [23]. Because of the ability to work in parallel while trees are being formed, training is performed faster than other boosting algorithms [21, 24]. Many XGBoosts are used in the literature, and in most studies, higher results are obtained than other algorithms [25 - 27]. However, the datasets used in the studies are not image data, the dataset used in this study is entirely on image data. In this study, feature extraction was made with VGG-16, classification was made with XGBoost and Random Forest algorithms and comparisons were made.

2.5. Dataset

The dataset used in this study was custom-prepared. The images of the dataset were taken from different farms and videos and divided into classes. An average of 350 images were taken from each cattle species in Table 1, and a total of 3507 cattle images were collected in the dataset. Cattle breeds that will be classified are given in Table 1. In Figure 1, the distribution of the number of images according to the types of the data in the dataset is given.

Table 1. *Cattle Breeds That Will Be Classified*

Species	
Charolais	Hereford
Holstein	Angus
Manda	Jersey
Simmental	Swedish Red

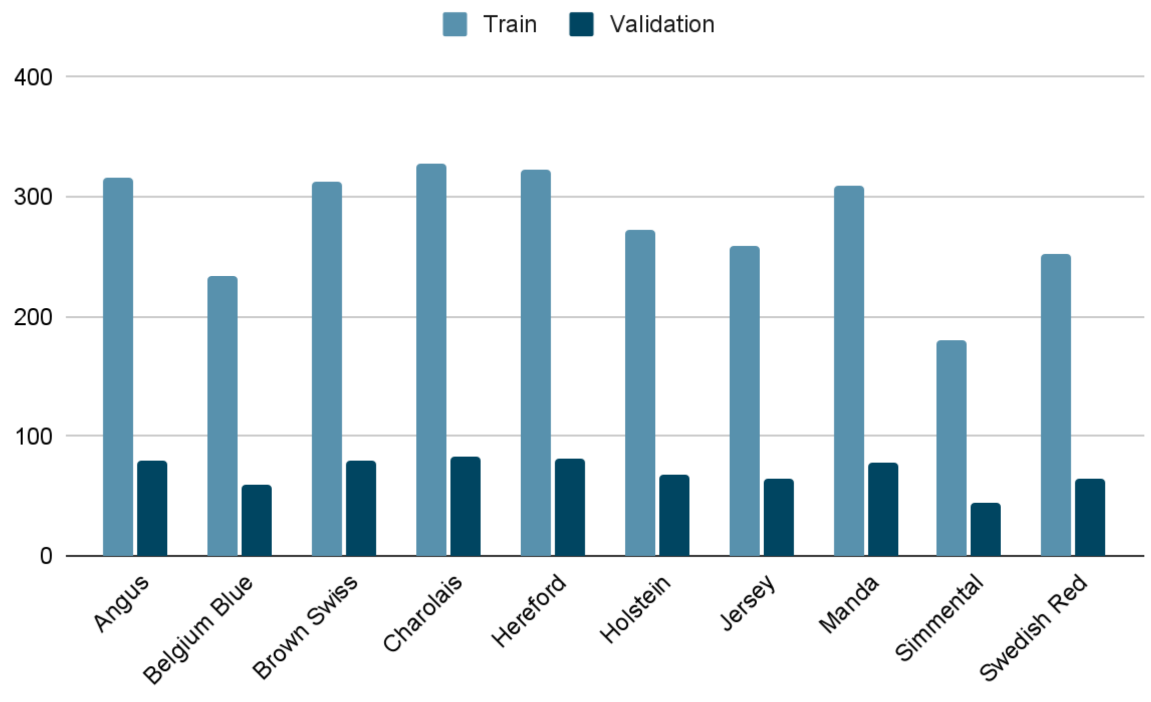

Figure 1. *Image distribution per class in the dataset before balancing step*

2.6. Data Preprocessing

In this study, the dataset was preprocessed, thus both the image features were revealed more and the dataset imbalances were eliminated. Augmentation methods and SMOTE algorithm were used to eliminate the imbalance in the dataset. Upsampling was done with Image Augmentation and SMOTE algorithms, thus the dataset imbalance was eliminated.

SMOTE is a preprocessing algorithm proposed as the upsampling method. It is used to eliminate the multi-class imbalance problem in the dataset. It balances the small number of data to the number of the largest number of data.

When an imbalance occurs in the dataset, problems may be encountered in the training phase, it is necessary to balance the number of data in the classes in the dataset, therefore, it is necessary to synthetically increase the data of the classes with the least number of data to the data number of the class with the highest number of data, i.e. upsampling.

To reduce the data of the classes with the highest number of data to the data number of the class with the least number of data, i.e. downsampling. Since the data will be sacrificed in the downsampling method, the quality of the dataset is reduced, so this method is not preferred much in datasets with high imbalances in the dataset. Since the upsampling method creates synthetic dirty data in the dataset, the quality of the dataset does not decrease and the data is balanced. Figure 2 shows how the SMOTE method generates new data against an imbalance, along with its function.

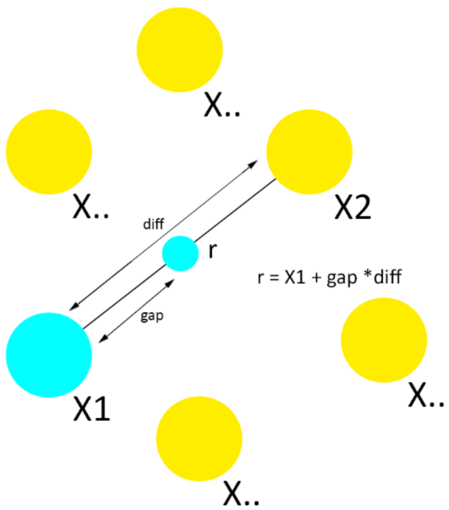

Figure 2. *Functional explanation of new data generation of SMOTE algorithm*

Since the dataset is unstable for training, upsampling was done with augmentation methods besides the SMOTE algorithm. For example, if the class with the most images in the dataset has 350 images, and if there are 250 images in another class, 100 more images are obtained with augmentation methods. Data augmentation steps applied to images are given in Table 2.

Table 2. *Data augmentation steps and descriptions*

Method	Description
Crop	Up to 20% of images are cropped in the middle.
Flip	Images were rotated between -15° and 15°.
Shear	Images are tilted +-15° horizontally and vertically.
Saturation	Saturation is given between -25% and 25%.
Brightness	Between -14% and 14% brightness increase-decrease process has been done.
Exposure	Exposure increase-decrease operations were performed between -25% and 25%.
Distortion	Noise is added to 5% of the pixels in the image.

20% of the obtained images are reserved as validation and 80% as training data. 2624 of them are training, 656 of them are validation data. In order to increase the processing speed of the images and to make the performance results of the models more effective, the images were preprocessed and 3 processes were added. These are, in turn, automatic orientation, resizing to 224x224, and automatic contrast adjustment using adaptive equalization. The distribution chart of the images according to the classes is given in Figure 3. As could be seen, the images have been tried to be distributed equally, and the class with the least images is Simmental with 6.5%.

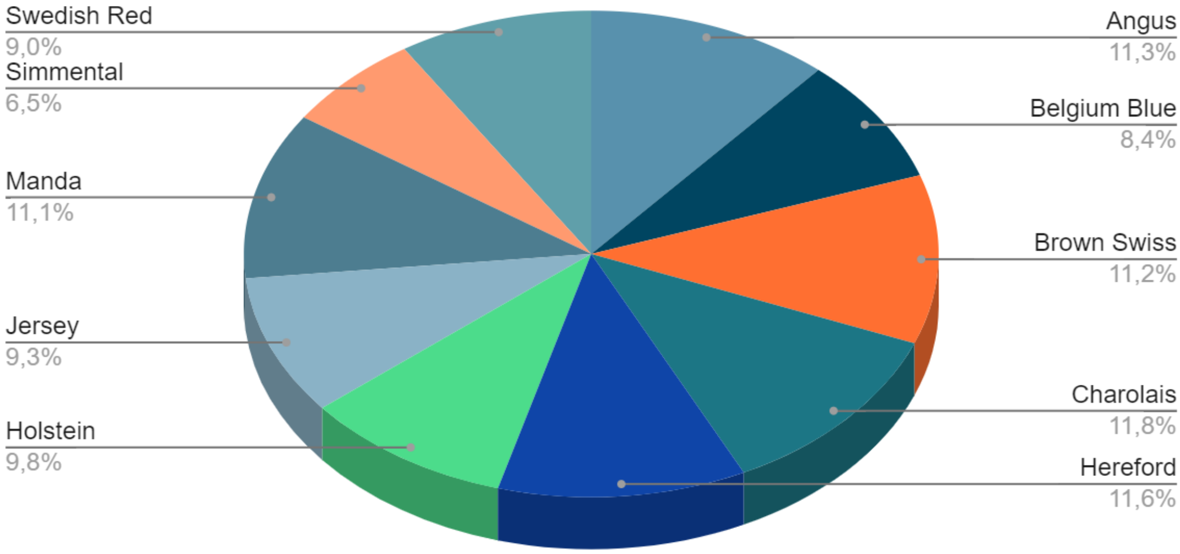

Figure 3. *Distribution of images according to classes*

The mobile application for species estimation was written in Flutter and was prepared to run Tensorflow Lite image classification models. Images are taken from the camera instantly and processed, sized as 240x240 pixels and transferred to the Tensorflow Lite model. The indices and estimation rates of the cattle species determined as output are obtained. The mobile application takes the indices from the species map and converts them to cattle species names and writes them one under the other in the block at the bottom of the application. It also writes the prediction rates next to the class names.

2.7. Mobile Application

After training the models, an application was prepared with Flutter, a cross-platform software language, since the best model was requested to be taken and run in the mobile application, and the models were run in this application. After the model was trained, the model was tested in the mobile application prepared with Flutter. The mobile application makes the classification in real time, continuously processes the images taken from the camera, transfers them to the model and makes inferences according to the result returned from the model. In order for the classification to be more successful, the pre-trained MobileNetV2 object recognition model from the Tensorflow Hub site was taken and transferred to the application. First, the application calls the object recognition model and sends the real-time image data received from the camera to the object recognition model, as output, the index numbers of the recognized objects, the xmin, xmax, ymin and ymax coordinate values of the object are returned as pixel values. Based on these coordinate values, a square is drawn on the camera image that appears in the application, the index number and the sequence number of the label names in the label map text file are taken, and the class to which the recognized object belongs is obtained and this name is placed just above the square. If the recognized object is in the cattle class, this time the image classification model we prepared is called and the image classified as cattle in the object recognition model is sent to the newly called model. Class index number and confidence rate are returned as output. As in object recognition, the application compares the index number from this data in the label map prepared for this model, and returns the cattle type class name with this sequence number in this index. At the end, a maximum of 3 of these results, especially the one with the highest confidence rate, are given at the bottom of the screen with confidence rates. This cycle continues until the user exits the screen. As a result, the classification of cattle species is realized. The logic of the Mobile Application is given in Figure 4 as a flowchart.

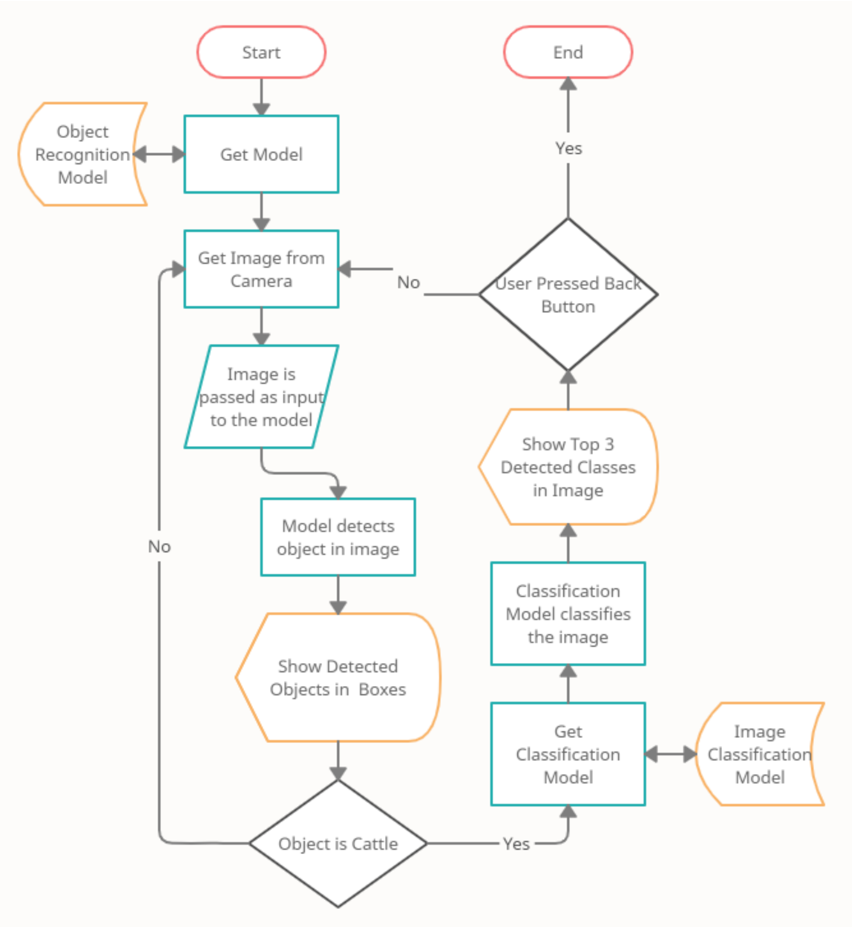

Figure 4. *Logic of the Mobile Application*

3. EXPERIMENTAL STUDIES

In this section, the training and performance analysis stages of the model is discussed.

Two training methods were used. First, the model was trained on pre-trained models using transfer learning with a custom-prepared dataset, and in the second method, the training was carried out by applying different combinations of VGG16+XGBoost and VGG16+Random Forest algorithms and SMOTE and Image Augmentation methods, which are the methods suggested in the study. In model training, the images reserved for training were trained on each model and performance analyzes were made. Models trained with the transfer learning method were trained in 100 epochs and 53 steps. The class names of the cattle species were taken from the file paths where the images were located and a class map was drawn accordingly. In the training of transfer learning models, the first layer is the model, the next two layers are Dropout(0.4) and a Dense layer that contains as many units as the number of classes. Softmax layer is added to the Dense layer as activation. Image sizes were set to 224x224 pixels in all models, Adam optimization algorithm was used in model optimization, default values were given as parameters.

In the performance analysis performed on the validation dataset after the training, the Accuracy, Precision, Recall and F1 Score values obtained for each model are given in Table 3.

Table 3. *Performance Analysis of Models*

Model	Accuracy	Precision	Recall	F1 Score
NasNet Mobile	0.84761	0.83888	0.85571	0,85415
Inception V2	0.80125	0.80999	0.79047	0,80974
Inception V3	0.84761	0.83597	0.85326	0,84167
VGG16+XGBoost	0.85755	0.86043	0.85755	0.85785
VGG16+XGBoost+SMOTE	0.85365	0.85796	0.85365	0.85387
VGG16+XGBoost+Augmentation	0.86330	0.86695	0.86330	0.86310
VGG16+XGBoost+SMOTE+Augmentation	0.85796	0.86215	0.85796	0.85812
VGG16+Random Forest	0.87050	0.87283	0.87050	0.87018
VGG16+Random Forest+SMOTE	0.85467	0.85346	0.85467	0.85221
VGG16+Random Forest+Augmentation	0.86474	0.86462	0.86478	0.86327
VGG16+Random Forest+SMOTE+Augmentation	0.88776	0.88848	0.88776	0.88690

After all models were trained, the Accuracy, Precision, Recall and F1 Score values were recorded for each model performed on the validation dataset, and it was found that the most suitable model for this problem was VGG16+Random Forest+SMOTE+Augmentation with 0.88776 Accuracy, 0.88848 Precision, 0.88776 Recall and 0.88690 F1 Score. Confusion matrix and heatmap of models trained with Random Forest and XGBoost are given in Figure 5.

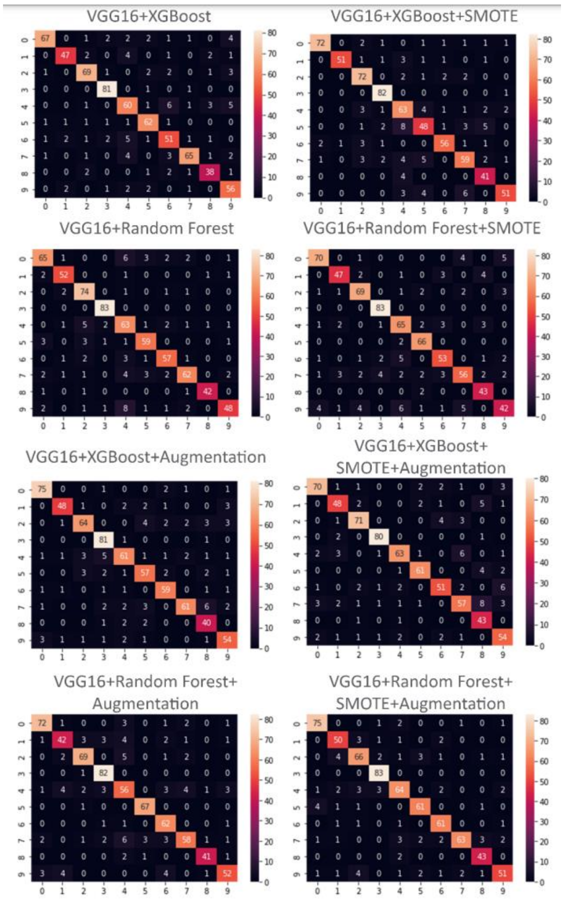

Figure 5. *Logic of the Mobile Application*

4. RESULTS

The fact that the Accuracy, Precision, Recall and F1 Score values are so close to each other shows that the dataset is balanced and the model training is performed properly. As seen in the Confusion Matrix, all classes are properly trained and there are no problems such as overfitting. In addition, as seen in both the confusion matrix and the performance results, the imbalance of the dataset was eliminated and it was understood that the upsampling method applied was concluded correctly. After the models were trained, it was desired to classify some images from the validation data, and the results are given in Figure 6. Models were registered as Keras Saved Model type. The Keras model was then converted to the Tensorflow Lite model, and the Tensorflow Lite model was quantized to make it work faster. After the Tensorflow Lite model was obtained, it was successfully tested in the mobile application prepared. VGG16+Random Forest+SMOTE+Augmentation model, which has the highest accuracy, precision, and sensitivity values in terms of species classification, was used for this application.

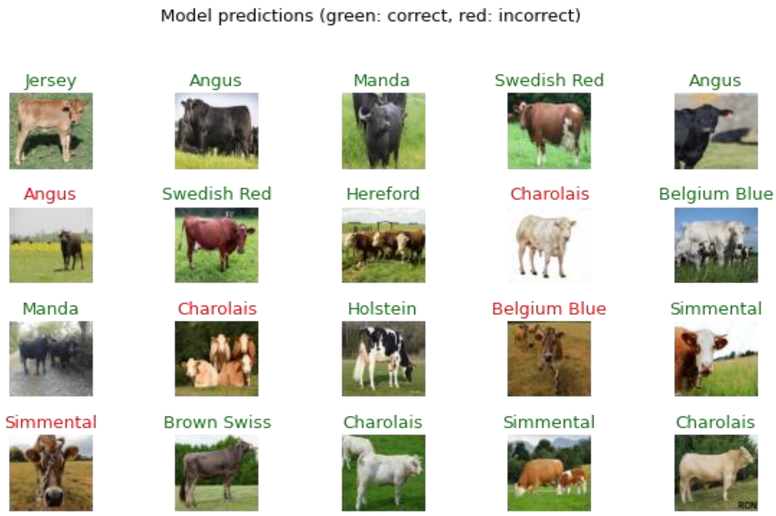

Figure 6. *Example Predictions*

After the best model is sent to the prepared mobile application, the mobile application has been tested and the classification is done successfully. According to the output obtained from the object recognition model, after obtaining that the object in the image is cattle, the image classification model prepared for this study is called and the classification is made with the image. According to the confidence rate in the output obtained from the classification model, the best 3 classes obtained for that image are written in the results section at the bottom of the screen. This classification process continues in real time until the user exits the classification screen in the application. The screenshot of the application, taken during the cattle species classification, is given in Figure 7.

Figure 7. *Screenshot of the Mobile Application*

4. CONCLUSION

The dataset of the image classification model was obtained separately for each species, and the performance values of the model were expected to be high by increasing the number of images obtained with data augmentation methods. Cattle verification was carried out by combining the obtained custom image classification model with the pre-trained object detection model. After checking the presence of cattle on the image with the pretrained object detection model, cattle species classification was carried out with the prepared image classification model. As a result, at the end of the study, it was understood that the most suitable model for estimating cattle breeds was VGG16+Random Forest+SMOTE+Augmentation with 88.77% Accuracy, 88.84% Precision, 88.77% Recall, 0.8869 F1 Score rates. The worst model is the Inception V2 model with 80.12% Accuracy, 80.99% Precision, 79.04% Recall and 0.80974 F1 Score. In order to further increase the performances, the number of images should be increased by at least 2 times. In future studies, besides estimating cattle species, their weights could be calculated and estimated using depth estimation methods. This species classification problem could be applied not only to cattle but also to other animal species. In addition, it could be used not only for the classification of mammals, but also for purposes such as determining whether insect species are harmful or harmless, and detecting flower species. Meetings of people with the same type of animals could also be arranged. In this study, there are 10 breeds of cattle, in order to increase this diversity, a system might be prepared in which the user may take images of cattle and upload them to a database. With the images taken from the user, the dataset may be expanded and new species could be obtained, the data of existing species could be more consistent. The dataset might be put into the training phase once a week and the diversity of users who will use the application might increase as the cattle species diversity increases. For this, image processing models should not be run in the mobile application, they should be used with the server so that the user might stay up to date. The real-time received image should be sent to the server, and the rendered output should be returned to the application. Since no cattle classification application has been seen in the literature before, the contribution of this study to the literature is to prove that cattle classification may be done with image processing methods.

ACKNOWLEDGMENT

This study was supported by Burdur Mehmet Akif Ersoy University Scientific Research Projects Commission. Project Number: 0831-YL-22.

This study was produced from the first author's thesis titled "Detection of cattle diseases with image processing methods".

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

[1] Santoni, M. M., Sensuse, D. I., Arymurthy, A. M., Fanany, M. I., "Cattle race classification using gray level co-occurrence matrix convolutional neural networks", Procedia Computer Science, 59: 493-502, (2015).

[2] Ou, Y., Wu, X., Qian, H., Xu, Y., "A real time race classification system", IEEE International Conference on Information Acquisition, 6, (2005).

[3] Sutojo, T., Tirajani, P. S., Ignatius Moses Setiadi, D. R., Sari, C. A., and Rachmawanto, E. H., "CBIR for classification of cow types using GLCM and color features extraction", 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 182-187, (2017).

[4] Bello, R. W., Talib, A. Z., Mohamed, A. S. A., Olubummo, D. A., Otobo, F. N., "Image-based Individual Cow Recognition Using Body Patterns", Image, 11(3): (2020).

[5] Jwade, S. A., Guzzomi, A., Mian, A., "On farm automatic sheep breed classification using deep learning", Computers and Electronics in Agriculture, 167, 105055, (2019).

[6] de Miranda Almeida, R. M., Chen, D., da Silva Filho, A. L., Brandao, W. C., "Machine Learning Algorithms for Breast Cancer Detection in Mammography Images: A Comparative Study", ICEIS, 660-667, (2021).

[7] David, R., Duke, J., Jain, A., Reddi, V. J., Jeffries, N., Li, J., Warden, P., "Tensorflow lite micro: Embedded machine learning on tinyml systems", Proceedings of Machine Learning and Systems, 3: 800-811, (2020).

[8] Simonyan, K., Zisserman, A., "Very deep convolutional networks for large-scale image recognition", arXiv preprint, arXiv:1409.1556, (2014).

[9] Shermin, T., Teng, S. W., Murshed, M., Lu, G., Sohel, F., Paul, M., "Enhanced transfer learning with imagenet trained classification layer", Pacific-Rim Symposium on Image and Video Technology, 142-155, (2019).

[10] Tammina, S., "Transfer learning using vgg-16 with deep convolutional neural network for classifying images", International Journal of Scientific and Research Publications, 9(10): 143-150, (2019).

[11] Srivastava, S., Kumar, P., Chaudhry, V., Singh, A., "Detection of ovarian cyst in ultrasound images using fine-tuned VGG-16 deep learning network", SN Computer Science, 1(2): 1-8, (2020).

[12] Rawat, J., Logofătu, D., Chiramel, S., "Factors affecting accuracy of convolutional neural network using VGG-16", International Conference on Engineering Applications of Neural Networks, 251-260, (2020).

[13] Kumar, A., Shaikh, A. M., Li, Y., Bilal, H., Yin, B., "Pruning filters with L1-norm and capped L1-norm for CNN compression", Applied Intelligence, 51(2): 1152-1160, (2021).

[14] Breiman, L., "Random forests", Machine Learning, 45(1): 5-32, (2001).

[15] Biau, G., Scornet, E., "A random forest guided tour", Test, 25(2): 197-227, (2006).

[16] Pal, M., “Random forest classifier for remote sensing classification”, International Journal of Remote Sensing, 26(1): 217-222, (2005).

[17] Segal, M. R., “Machine learning benchmarks and random forest regression”, Center for Bioinformatics and Molecular Biostatistics, (2004).

[18] Didavi, A. B., Agbokpanzo, R. G., Agbomahena, M., “Comparative study of Decision Tree, Random Forest and XGBoost performance in forecasting the power output of a photovoltaic system”, 4th International Conference on Bio-Engineering for Smart Technologies, 1-5, (2021).

[19] Oshiro, T. M., Perez, P. S., Baranauskas, J. A., “How many trees in a random forest?”, International Workshop on Machine Learning and Data Mining in Pattern Recognition, 154-168, (2012).

[20] Kulkarni, V. Y., Sinha, P. K., “Pruning of random forest classifiers: A survey and future directions”, 2012 International Conference on Data Science & Engineering, 64-68, (2012).

[21] Chen, T., Guestrin, C., “Xgboost: A scalable tree boosting system”, Proceedings of the 22nd acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 785-794, (2016).

[22] Brownlee, J., “XGBoost With python: Gradient boosted trees with XGBoost and scikit-learn”, Machine Learning Mastery, (2016).

[23] Priscilla, C. V., Prabha, D. P., “Influence of optimizing XGBoost to handle class imbalance in credit card fraud detection”, 2020 Third International Conference on Smart Systems and Inventive Technology, 1309-1315, (2020).

[24] Yadav, D. C., Pal, S., “Analysis of heart disease using parallel and sequential ensemble methods with feature selection techniques: heart disease prediction”, International Journal of Big Data and Analytics in Healthcare, 6(1): 40-56, (2021).

[25] Dhaliwal, S. S., Nahid, A. A., Abbas, R., “Effective intrusion detection system using XGBoost”, Information, 9(7): 149, (2018).

[26] Davagdorj, K., Pham, V. H., Theera-Umpon, N., Ryu, K. H., “XGBoost-based framework for smoking-induced noncommunicable disease prediction”, International Journal of Environmental Research and Public Health, 17(18): 6513, (2020).

[27] Sahin, E. K., “Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest”, SN Applied Sciences, 2(7): 1-17, (2020).