International Journal of Computer Science & Communications
Vol. 10, No. 1, 2025, pp. 1-13
http://ijcsc.ielas.org/index.php/ijcsc
ISSN: 2545-4153

Application of Gender Classification Using CNN Model with Python Program

Abdul Qadeer Rasooli

*Selçuk University, Institute of Science, Department of Computer Engineering
ORCID https://orcid.org/0000-0003-1950-3209
E-mail: rasooli.csf@gmail.com

Received: 2025-09-02 Accepted: 2025-09-22 Published online: 2025-09-23

Abstract

For the past ten years, researchers have found great interest in the automatic classification of gender from face photos. For an automatic classification system to be successful, feature extraction and classification techniques are crucial. Today's rich face image databases have made it possible to apply numerous effective machine learning and deep learning techniques. When using classical machine learning methods, it is crucial to extract precise features from the datasets in order to produce promising classification results. On the other hand, features can be automatically extracted from raw data directly using deep learning models. In addition to classifying, this automates the feature extraction procedure. When compared to conventional machine learning techniques, deep neural networks can improve classification performance by exploring hidden and unpredictable feature sets. Many scientists have turned to convolutional neural networks (CNN), one of the most successful types of deep models, to help them solve the gender classification challenge. It can address the issue of face cues changing from one origin to another, which complicates precise feature extraction. Modern pretrained CNN architectures come in several forms that work incredibly well for picture categorization issues. In general, CNNs perform better when there are more input data points. In this study, we classified the images according to gender like Male, Female, Baby, and Girl. We trained the CNN model to classify the images using a Python program. Multi-class image classification is a computer vision task that involves categorizing images into more than two classes or categories. In other words, the goal is to teach a machine learning model to recognize and distinguish between multiple classes or types of objects within images.

Keywords: CNN, Classification, Logistic regression, Random Forest, Decision Tree algorithms.

1. Introduction

In computer vision, multi-class picture categorization is a key task, playing a crucial role in enabling machines to comprehend and categorize diverse visual content. This process involves the development and implementation of algorithms capable of automatically assigning predefined labels or classes to input images based on their content (Shah & Sajnani, 2020). The primary goal is to equip computational systems with the ability to recognize and interpret visual patterns, akin to human perception, thereby facilitating a broad range of applications. From everyday technologies like image organization on smartphones to complex applications in fields such as medical

diagnostics, where the identification of specific patterns in images is vital, multi-class image classification forms the backbone of various artificial intelligence endeavors. This discipline is driven by the overarching objective of harnessing machine learning techniques to automate the interpretation of visual data, contributing to efficiency, decision-making processes, and advancements in diverse industries (Ezat et al., 2020).

The goals of the application we designed for gender image classification using CNN10 in Python typically involve accurately predicting the gender (*Male, Female, Baby,* and *Girl*). From input images, leveraging convolutional neural networks (CNN10) for feature extraction and classification of each of them in separate classes.

Multi-class image classification is a fundamental aspect of computer vision that endeavors to automatically classify images into discrete classes from a predetermined collection. The primary goal is to outfit machines with the capability to recognize and interpret the content of pictures, enabling them to understand the visual world in a manner analogous to human perception. This facilitates a broad spectrum of applications, ranging from everyday technologies like photo organization in smartphones to more sophisticated applications (Yıldız et al., 2021). Moreover, multi-class image classification is instrumental in enhancing decision-making processes across diverse domains. By accurately categorizing images, this technology aids in automating tasks that would otherwise be labor-intensive and time-consuming (Yıldız et al., 2021).

2. Method

One widely used method for multi-class image classification is convolutional neural networks (CNN10). CNN10 is a family of deep learning models that is very useful for jobs involving images because it is primarily made for processing and evaluating visual data. Multi-class image classification was also performed using a Decision Tree, Logistic Regression, and Random Forest algorithm under orange tools.

Models suggested. The high-level block diagram of the CNN-10 models utilized in this paper's picture classification is described in this section. Important elements including the data set, cross-validation, and data augmentation methods utilized in this work are included in this diagram. The procedures used in this work are described in the steps that follow, as illustrated in Figure 1.

1) Dataset: There are four categories of photos in my dataset. The labeled photos in this collection represent a variety of images, including *Male*, *Female*, *Baby*, and *Girl*. The CNN models are trained and evaluated using these images as the input data.

2) Data Augmentation: Data augmentation techniques are used to improve the dataset's robustness and diversity. These methods entail giving the preexisting photographs adjustments like rotation, scaling, and cropping the information.

3) Cross-Validation: Cross-validation is used to evaluate the CNN models' performance and capacity for generalization. The dataset is separated into several subsets or folds. Every fold serves as the validation set once during the different iterations of training and evaluation. This methodology guarantees a thorough assessment of the CNN models' efficacy across various data subsets.

4) CNN-10: The architecture of CNN consists of ten convolutional layers. It is intended to extract increasingly abstract representations by first capturing hierarchical information from input photos.

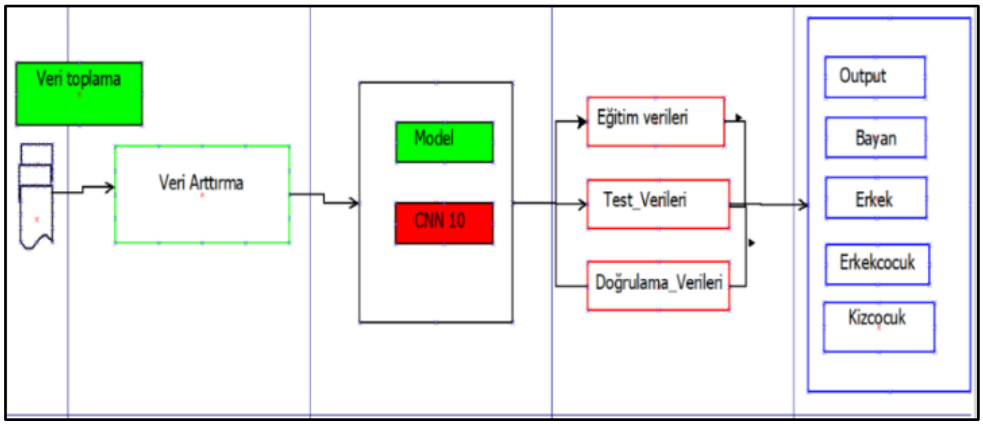

Figure 1. Block Diagram of Image Classification

A typical block diagram for image classification involves three main stages: input images for preprocessing, feature extraction using convolutional neural networks (CNNs), and classification using a machine learning algorithm. The input images are initially processed to enhance relevant features, followed by the extraction of discriminative features by CNN layers, and ultimately, a classifier assigns the images to predefined classes based on these features.

2.1. Problem statement

There are several issues with multi-class image classification, including imbalanced datasets with low representation of some classes, which might result in biased models. Model complexity must be carefully balanced to avoid overfitting and underfitting, which endanger the model's capacity to generalize successfully to new data. The scarcity of sufficient training data for specific classes can hinder accurate classification, while the inherent variability and ambiguity in real-world images, compounded by interclass

confusion, contribute to misclassifications. The computational complexity of training deep neural networks for large-scale multi-class problems can be resource-intensive. Additionally, ensuring model robustness to adversarial attacks and addressing transferability issues from pre-trained models to new domains are ongoing concerns. These challenges necessitate innovative approaches in model architecture, data augmentation, regularization, and domain-specific fine-tuning to enhance the overall performance and reliability of the classification of images in many class systems.

2.2. Multi-Class Image Classification

Multi-class image classification involves training a machine learning model to recognize and categorize images into more than two classes or categories. The primary objective is to develop a model capable of accurately assigning a label to an input image from a set of multiple predefined classes. This task requires a dataset with images labeled with their corresponding classes for training. The model gains the ability to identify characteristics in the input photos and map them to the relevant classes during training. Deep learning architectures such as convolutional neural networks (CNNs), which automatically build hierarchical representations of visual data, are commonly used for multi-class picture categorization. After that, a different collection of photos is utilized to evaluate the trained model's generalization and accuracy in classifying unknown data. Performance metrics including accuracy, precision, recall, and F1 score are used to measure the model's efficacy (Yuda et al., 2020).

Below flowchart depicting a proposed convolutional neural network (CNN) architecture designed for gender classification. It outlines the preprocessing of input images and the sequence of the convolutional layer, max pooling layer, and fully connected layers leading to the final classification output.

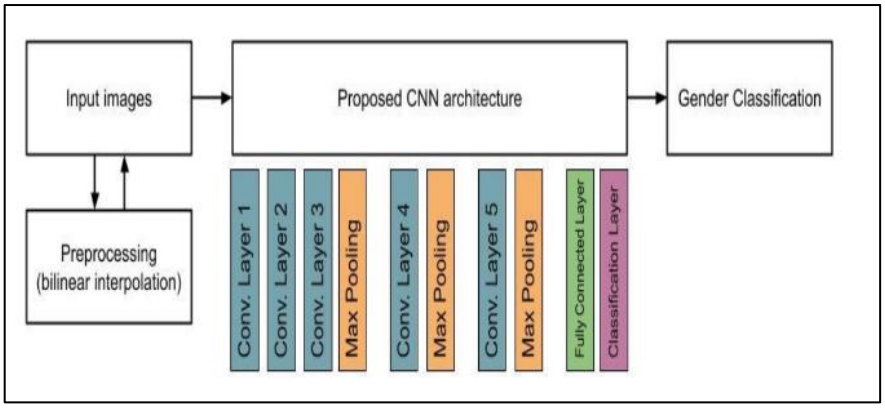

Figure 2. Proposed Convolutional Neural Network Architecture

2.3. Convolutional Neural Network

A subclass of deep neural networks called convolutional neural networks (CNNs) is used to process and analyze visual data, such as images. When it comes to computer vision tasks like object identification, image segmentation, and image classification, CNNs have shown to be incredibly efficient. These are the CNN algorithm's main features (Shah & Sajnani, 2020).

Figure 3 demonstrates the block diagram of a convolutional neural network (CNN) for image classification, the block diagram is composed of layers representing the input images, convolutional layers for feature extraction, pooling layers for downsampling, fully connected layers for classification, and output layer for prediction. The network processes input images through convolutional operations to learn hierarchical features, utilizes pooling to reduce spatial dimensions, and employs fully connected layers for classification based on the extracted features. The final output layer produces predictions, enabling the CNN to classify images into predefined categories (Laroui et al., 2019).

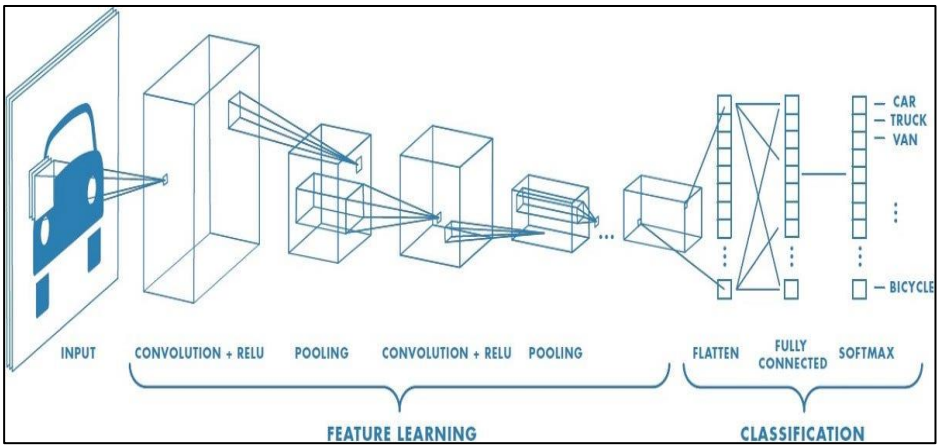

Figure 3. Block diagram of Image Classification (Yuda et al., 2020).

3. Result and Discussion

The tests carried out on four distinct classes of photos utilizing certain deep learning and machine learning methods are presented in this part.

The recommended method is implemented in a Jupyter Notebook environment using the programming language Python (version 11.5). The popular deep learning frameworks TensorFlow (2.15.0 version), Numpy library (1.26.1 version), Pandas library (2.1.4 version), and Seaborn (0.13.0 version) are used in the experimentation. The

evaluation parameters are detailed in Table 1. The effectiveness of the method suggested in this article was evaluated using the conventional assessment techniques previously discussed. Table 2 displays the results of the training and validation sets' performance evaluations.

Parameters

Choosing appropriate parameter values are vital to the model's performance, as it influences the network's ability to learn and generalize patterns from the data. Fine-tuning these parameters frequently requires trial and error, therefore methods like grid search or random search might be useful to quickly investigate a variety of options. Overall, careful parameter setting is crucial for achieving optimal results in training deep learning CNNs. Table 1 displays some parameters and their corresponding values utilized in our project (Laroui et al., 2019).

Table 1. Parameter Setting

Parametre	**Value**
Filtre	32,16
Çekirdek_boyutu	3
Aktivasyon	Relu, Softmax
LeakyRelu function	0.5
Dropout function	0.25, 0.2
Yoğun	256
Havuzlama_boyutu	2

The performance function in a CNN algorithm evaluates how accurately the model can classify or predict outcomes based on input data. It measures the network's efficiency in recognizing and decoding patterns in the data. The goal is to optimize the performance function during the training process by adjusting parameters and weights, ultimately enhancing the capacity of the model to generalize and produce the best prediction on fresh, untested data. Table 2 demonstrates the performance of the CNN model under different metrics.

Table 2. Performance of CNN Model with Different Metrics

Model	**Accuracy_Score**	**Percision_Score**	**Recall_Score**	**F1-Score**
CNN10	0.88	0.26	0.38	0.31

Training Accuracy (Train Accuracy): This metric assesses how accurate a machine learning model is using the training set of data. It is calculated by comparing the model's likelihood on the training data to the actual labels and determining the

proportion of correct predictions. Training accuracy indicates how well the model has been educated from the exercise data.

Validation Accuracy (Val Accuracy): Validation accuracy is a metric used to assess the performance of a model on a separate dataset called the validation set. This set is distinct from the training data and is not used during the training process. Validation accuracy helps to evaluate how efficiently the model adapts to fresh, clean data. When a model performs well on training data but is unable to generalize to new samples, it is critical for identifying overfitting.

Below figure 4. demonstrates the train accuracy and validation accuracy of the CNN model. The confusion matrix is a tabular representation that shows the counts of actual positive, actual negative, false positive, and false negative predictions in order to assess how well a classification model performs. It offers a thorough analysis of the model's performance, making it easier to evaluate classification parameters like accuracy, precision, and recall. Figure 4 demonstrates the prediction of the confusion matrix using the CNN model (Peryanto et al., 2022).

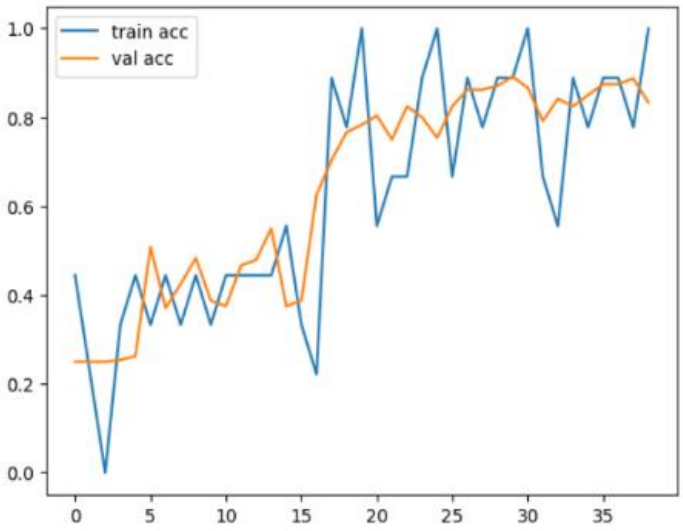

Figure 4. Train Accuracy and Validation Accuracy with CNN

Below is Figure 5 demonstrates the confusion matrix, which represents the forecast and real class labels for a different class classification problem using the CNN model (Peryanto vd., 2022).

International Journal of Computer Science & Communications
Vol. 10, No. 1, 2025, pp. 1-13
http://ijcsc.ielas.org/index.php/ijcsc
ISSN: 2545-4153

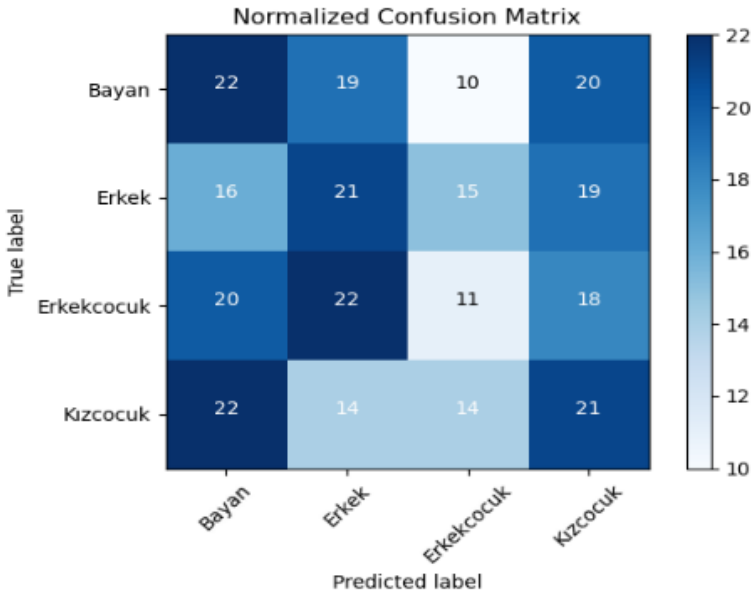

Figure 5. Confusion Matrix of CNN Model

The predicted and true class labels for a different-class classification task using a CNN model are represented by the confusion matrix. The figure consists of four classes: *Male, Female, Baby*, and *Girl.*

For *Female*, out of 71 images of the *Female* class, the model detected correctly 22 images as *Female* (actual positives). But it was incorrectly labeled 19 images as *Male* 10 images as *Baby,* and 20 images as *Girl* (false negatives). For *Male*, out of 71 images of the *Male* class, the model detected correctly 21 images as *Male* (actual positives). But it was incorrectly labeled 16 images as *Female* 15 images as *Baby,* and 19 images as *Girl* (false negatives). For *Baby*, out of 71 images of the *Baby* class, the model detected correctly 11 images as *Baby* (actual positives). But it was incorrectly labeled 20 images as *Female* 22 images as *Male,* and 18 images as *Girl* (false negatives). Finally for *Girl*, out of 71 images of the *Girl* class, the model detected correctly 21 images as *Girl* (actual positives). But it was incorrectly labeled 22 images as *Female* 14 images as *Male,* and 14 images as *Baby* (false negatives).

Orange (version 3.26.0) is utilized to put the recommended strategy into practice in the orange tools environment. Through testing, we used different algorithms of machine learning. Table 3. provides the information about the parameters. The effectiveness of the method suggested in this article was evaluated using the conventional assessment techniques previously discussed.

Table 3. Performance Metrics

Model	AUC	CA	F1-Score	Precision	Recall	MCC
Neural Network	0.986	0.877	0.877	0.87	0.87	0.83
Logistic Regression	0.989	0.90	0.908	0.909	0.908	0.87
Random Forest	0.96	0.83	0.835	0.836	0.835	0.78
Decision Tree	0.84	0.75	0.757	0.758	0.757	0.67

The performance function typically refers to the evaluation metrics used to assess the quality of a predictive model. These metrics assist users in measuring a model's performance on a particular job, such as regression or classification. Depending on the particular type of analysis, common performance indicators include the area under the ROC curve (AUC-ROC), recall score, accuracy score, precision score, and F1 score. Table 3 demonstrates the performance metric of Neural Network, Logistic Regression, Random Forest, and Decision Tree algorithms (Peryanto et al., 2022).

Figure 6 demonstrates the result of the Random Forest classification algorithm, according to the result of the random forest algorithm prediction performance in the *Female* and *Baby* classes is better than in the *Male* and *Girl* classes.

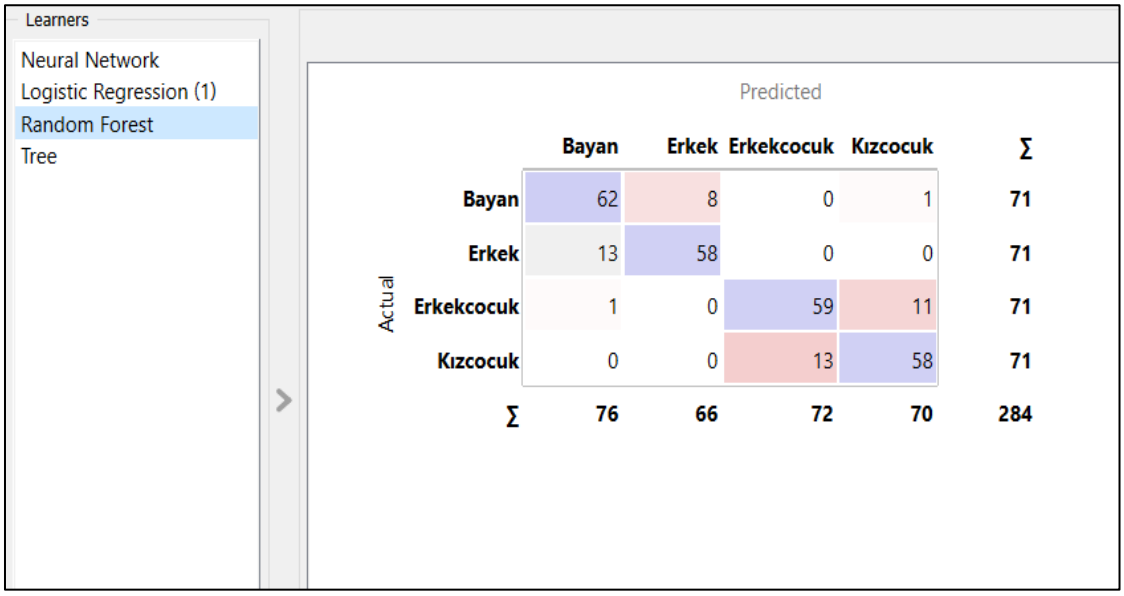

Figure 6. Confusion Matrix with Random Forest Algorithm

According to the result of Figure 6 for *Female*, out of 71 images of the *Female* class, the model detected correctly 62 images as *Female* (actual positives). But it was incorrectly labeled 8 images as *Male* and 1 image as *Girls* (false negatives).

For *Male*, out of 71 images of the *Male* class, the model detected correctly 58 images as *Male* (actual positives). But it was incorrectly labeled 13 images as *Female* (false negatives). For *Baby*, out of 71 images of the *Baby* class, the model detected correctly 59 images as *Baby* (actual positives). But it was incorrectly labeled 1 image as *Female* and 11 images as *Girls* (false negatives). For *Girl*, out of 71 images of the *Girl* class, the model detected correctly 58 images as *Girl* (actual positives). But it was incorrectly labeled 13 images as *Baby* (false negatives).

Figure 7 demonstrates the result of the Logistic Regression classifier, according to the result of the logistic regression algorithm, prediction performance in the *Baby* class is better than *Female*, *Male,* and *Girl* classes.

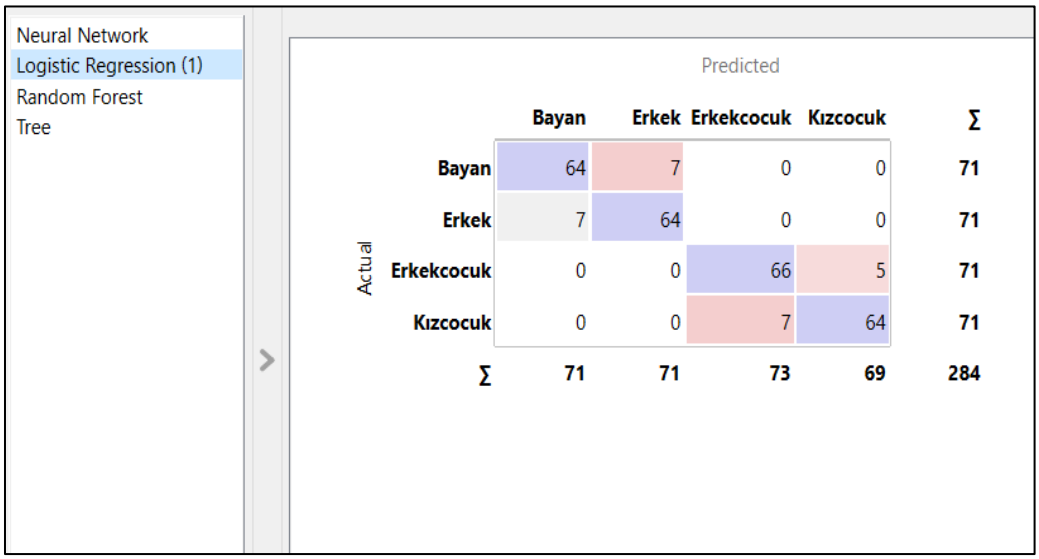

Figure 7. Confusion Matrix with Logistic Regression Algorithm

According to the result of Figure 7 for *Female*, out of 71 images of the *Female* class, the model detected correctly 64 images as *Female* (actual positives). But it was incorrectly labeled 7 images as *Male* (false negatives).

For *Male*, out of 71 images of the *Male* class, the model detected correctly 64 images as *Male* (actual positives). But it was incorrectly labeled 7 images as *Female* (false negatives). For *Baby*, out of 71 images of the *Baby* class, the model detected correctly 66 images as *Baby* (actual positives). But it was incorrectly labeled 5 images as *Girl* (false negatives). For *Girl*, out of 71 images of the *Girl* class, the model detected correctly 64 images as *Girl* (actual positives). But it was incorrectly labeled 7 images as *Baby* (false negatives).

Figure 8 demonstrates the incorrect classification result of the logistic regression classifier under the *Baby* class.

International Journal of Computer Science & Communications
Vol. 10, No. 1, 2025, pp. 1-13
http://ijcsc.ielas.org/index.php/ijcsc
ISSN: 2545-4153

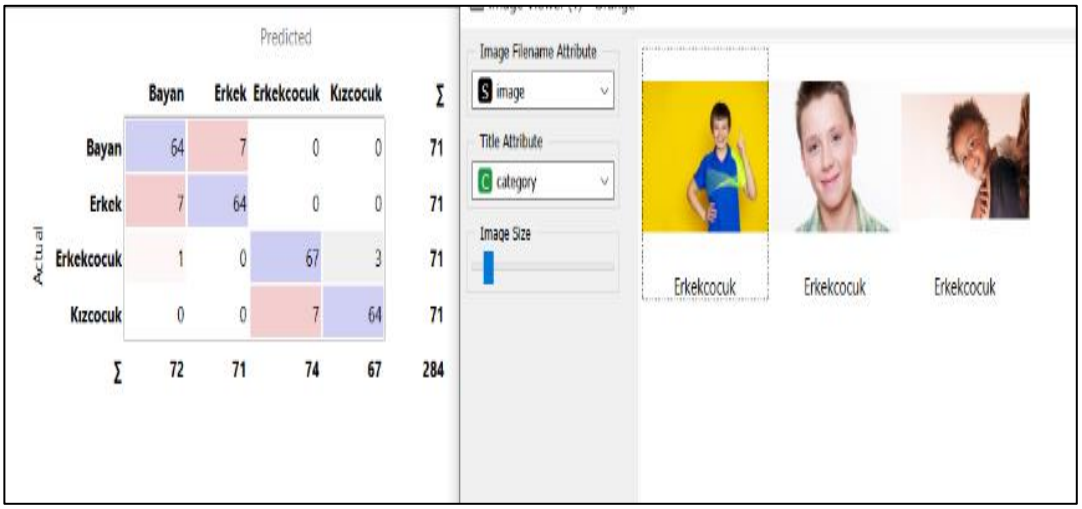

Figure 8. Mis Classification of Logistic Regression Algorithm

The logistic regression misclassified images of three people originally labeled as *Baby,* but this algorithm incorrectly predicted as *Girl* and classified in the *Girl* class.

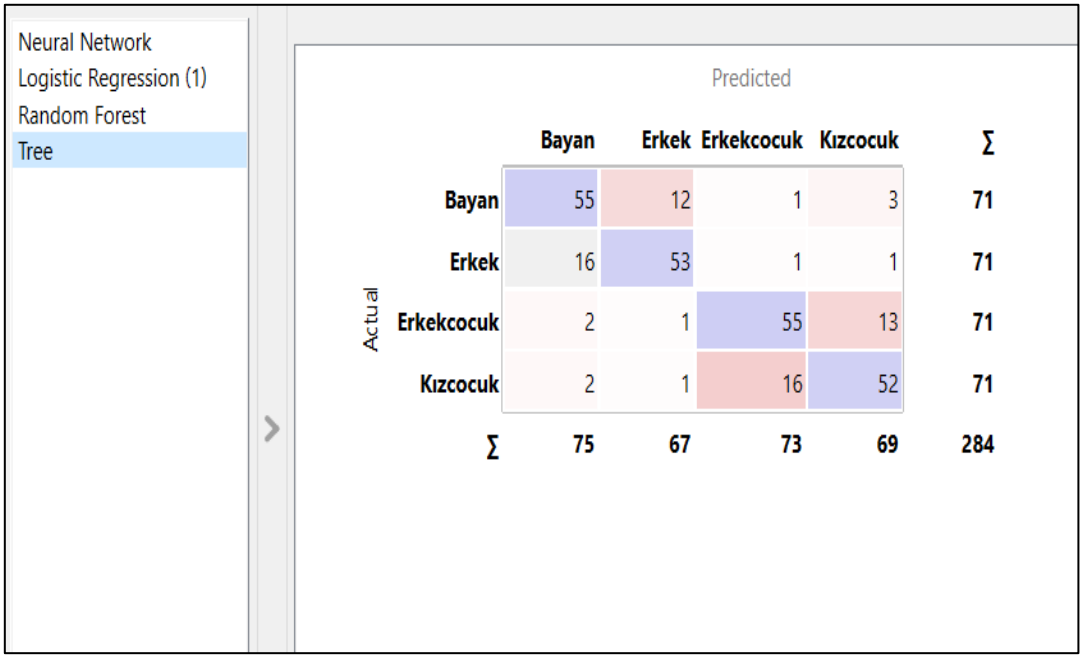

Figure 9. Confusion Matrix with Decision Tree Algorithm

According to the result of Figure 9 for *Female*, out of 71 images of the *Female* class, the model detected correctly 55 images as *Female* (actual positives). But it was

incorrectly labeled 12 images as *Male* 1 image as *Baby,* and 3 images as *Girl* (false negatives).

For *Male*, out of 71 images of the *Male* class, the model detected correctly 53 images as *Male* (actual positives). But it was incorrectly labeled 16 images as *Female* 1 image as *Baby,* and 1 image as *Girl* (false negatives). For *Baby*, out of 71 images of the *Baby* class, the model detected correctly 55 images as *Baby* (actual positives). But it was incorrectly labeled 2 images as *Female* 1 image as *Male,* and 13 images as *Girl* (false negatives). For *Girl*, out of 71 images of the *Girl* class, the model detected correctly 52 images as *Girl* (actual positives). But it was incorrectly labeled 2 images as *Female* 1 image as *Male,* and 16 images as *Baby* (false negatives).

4. Conclusion

The experimental study has demonstrated that the CNN model, a deep learning method, offers a straightforward and efficient solution for the multi-class picture classification problem. With a broad variety of data sets, the CNN model, a deep learning algorithm, can adapt to changing requirements. As demonstrated on a local computer using Python scripts, a basic training database including data has been created and uploaded to a stronger computer for the training procedure. As previously mentioned, the CNN model for the deep learning algorithm has been trained, and the model weight has been adjusted. The fresh data set has required adjustments to the model. The model's accuracy has grown with a small number of training examples.

In this study, using the CNN-10 deep learning algorithm, we investigated the problem of various classes of picture categorization with Python and also we explored the task of different class image classification using Logistic Regression, Random Forest, and Decision Tree algorithm with orange tools. To assess each algorithm's performance, we carried out in-depth tests.

In conclusion, our research offers insightful information on how various tools and algorithms perform in comparison when it comes to multiclass image classification. The superior accuracy was achieved by the Orange tool and we suggest the Orange tool for small datasets also we recommend the CNN-10 algorithm for the classification of large image datasets. Since the CNN model performs better with huge datasets than the orange tools.

References

1. Akbulut, Y., Şengür, A., & Ekici, S. (2017, September). Gender recognition from face images with deep learning. In 2017 International artificial intelligence and data processing symposium (IDAP), 1-4. IEEE
2. Cengil, E., Cinar, A., & Ozbay, E. (2017). Image classification with caffe deep learning framework. 2017 International Conference on Computer Science and Engineering (UBMK). http://doi.org/10.1109/ubmk.2017.8093433
3. Ezat, W. A., Dessouky, M. M., & Ismail, N. A. (2020). Derin öğrenme algoritması kullanarak çoklu sınıf görüntü sınıflandırması. Fizik Dergisi: Konferans Serisi, 1447(1). https://doi.org/10.1088/1742-6596/1447/1/012021
4. Laroui, S., Omara, H., Lazaar, M., & Mahboub, O. (2019). CNN mimarilerinde uygulanan performans özelliklerinin karşılaştırmalı çalışması. Haziran 2020. https://doi.org/10.4108/eai.24-4-2019.2284238
5. Peryanto, A., Yudhana, A., & Umar, R. (2022). Çiçek görüntülerinin sınıflandırmasında evrişimli sinir ağı ve destek vektör makinesi. Khazanah Informatika: Jurnal Ilmu Komputer Dan Informatika, 8(1), 1–7. https://doi.org/10.23917/khif.v8i1.15531
6. Shah, V., & Sajnani, N. (2020). Multi-class image classification using CNN and Tflite. *International Journal of Research in Engineering, Science and Management*, *3*(11), 65–68. https://doi.org/10.47607/ijresm.2020.375.
7. R. Waseem and W. Zenghui, "Deep convolutional neural networks for ımage classification: A Comprehensive Review," Neural Comput., vol. 29, no. 7, pp. 2352–2449, 2017, doi: 10.1162/NECO.
8. Shah, V., & Sajnani, N. (2020). CNN ve Tflite kullanarak çoklu sınıf görüntü sınıflandırması. Mühendislik, Bilim ve Yönetim Araştırmaları Uluslararası Dergisi, 3(11), 65–68. https://doi.org/10.47607/ijresm.2020.375
9. Yıldız, K., Güneş, E., & Bas, A. (2021). Kontrolsüz ortamlarda CNN tabanlı cinsiyet tahmini. Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 9(2), 890–898. https://doi.org/10.29130/dubited.763427
10. Yuda, R. P., Aroef, C., Rustam, Z., & Alatas, H. (2020). Evrişimli sinir ağları (CNN'ler) kullanarak yüz tanıma temelli cinsiyet sınıflandırması. Fizik Dergisi: Konferans Serisi, 1490(1). https://doi.org/10.1088/1742-6596/1490/1/012042
11. Lee, C. Y., P.W. Gallagher, and Z. Tu. Generalizing pooling functions in convolutionalal neural networks: Mixed, gated, and tree. in International Conference on Artificial Intelligence and Statistics, 2016.
12. Zhou, X., Yu, K., Zhang, T., & Huang, T. S. (2010). Image classification using super-vector coding of local image descriptors. Computer Vision – ECCV 2010 Lecture Notes in Computer Science, 141–154. http://doi.org/10.1007/978-3-642-15555-0_11.