APWEN Journal of Engineering, Science and Technology
ISSN: 2714-2396, Volume 7 Issue 1

GENDER CLASSIFICATION BASED ON FACIAL INFORMATION USING EFFICIENT-NET

*[1]Adebayo, L. O., [1]Amusa, K. A., [1]Nuga, O. O. and [2] Aforesaid, D. O.

[1]Department of Electrical and Electronics Engineering, Federal University of Agriculture, Abeokuta, Nigeria.
[2]Department of Computer Science, Federal University of Agriculture, Abeokuta, Nigeria.
*adebayolajide@gmail.com; amusaka@funaab.edu.ng; nugaoo@funaab.edu.ng; aborisadedo@funaab.edu.ng

Received Date: 2nd April 2024 Accepted Date: 15th July 2024 Published Date: 15th Sept., 2024

ABSTRACT
Despite the significant progress in gender classification, several challenges persist. This paper explored using the EfficientNet-B0 model in achieving gender classification from facial information. The model used a dataset of Nigerians' facial images obtained by scraping search engines and online platforms. Collected images were manually annotated, resized, normalised, and split using stratified sampling for training, validation, and test sets. The model was compiled and optimised using the binary cross-entropy loss function and Adam optimiser. The implementation was done in a Python 3.6 environment with TensorFlow and Keras employed for construction and data augmentation. The model was evaluated using the scikit-learn library, and performance metrics were chosen, such as overall accuracy, precision, recall, and F1 score. Results obtained with pre-trained Efficient-Net showed that accuracy on the validation set was 49.11%. The precision for class 1 was 0.49, suggesting that roughly half of the positives were correctly identified while that of class 0 was 0.00. The weighted averages for precision, recall and F1-score were 0.24, 0.49 and 0.32, respectively. The accuracy after model fine-tuning was 0.76 (training dataset), 0.82 (validation dataset), and 0.81 (test dataset), while the overall accuracy was 0.81. Going by obtained results after model fine-tuning, it is obvious that the Efficient-Net model has the capability of classifying people gender based on facial information correctly.
Keywords: Gender classification, facial information, EfficientNetB0, dataset, fine-tuning.

1. INTRODUCTION

Recently, automatic gender classification has received more interest due to its widespread use in various fields such as social media, human-computer interaction, population census, biometrics, demographic statistics, surveillance systems, commercial development, mobile applications and video games. Gender classification is considered significant in these fields, depending on the differing attributes of masculinity and femininity that are socially constructed. Gender classification is considered a key component of artificial intelligence applications within computer vision and machine learning (Hager, 2022). Gender classification has been studied using controlled and uncontrolled datasets, where features such as the face, eyebrows, hand shape, body shape, fingernails, hair and muscle contours are used. Of all the aforementioned features, the face constitutes the most frequently adopted modality in the gender classification literature. Algorithms such as Eigenfaces, graph matching, and facial embedding are usually used to extract facial features.

The feature extraction process can be divided into two groups: the geometric-based and the appearance-based methods. Geometric-based approaches map the input images into geometric primitives such as points, lines, and curves to represent the main features. The geometrical primitives of the face, such as eyes, nose, mouth and ear, are called face geometrics and are adopted for distinctive feature localisation. The feature-based methods, on the other hand, involve working on the entire image or its components via the application of image filters to extract features (Barham et al., 2016). The appearance-based feature extraction method is superior to the geometric-based approach due to its utilization of the entire face image instead image portion around such fixed specific where geometric features are extracted.

In addition to the requirement of feature extraction for gender classification tasks, a controlled dataset is required. A number of pre-trained image batches of a specific size and a test batch are required. These datasets are mainly used to train deep neural networks and assess the performance of computer vision algorithms for major tasks. The training process of classifying gender includes the use of methods such as principal component analysis (PCA), support vector machine (SVM) and neural networks (NN). However, the convolution NN (CNN) has been shown to perform better than other machine learning algorithms for gender classification tasks (Tahmina et al., 2017). While the filters are optimized through automated learning in CNN, they are hand-engineered in other traditional algorithms. This constitutes a major edge for CNN as it is independent of human intervention in feature extraction. In addition, the use of an algorithm with pixel vector is often associated with loss of many spatial interactions between pixels, CNN is devoid of this as it effectively uses adjacent pixel information through convolution and then applies a prediction layer at the end (Tahmina *et al.*, 2017).

In the past decade, several machine learning models have been developed and proposed for gender classification task. One of the first methods for gender classification was SEXNET which classified images sampled at 30 x 30 pixels using a fully connected neural network with 3 layers of 900, 40 and 900 neurons (Bagrami and Halili, 2022). The major challenge with SEXNET is the requirement for full rotation of faces such that the eyes and mouth are leveled accordingly, for a successful gender classification. The method is reported to yield over 19% error in a test conducted on 90 faces (45 male and 45 females). The gender classification task was approached using an SVM-based algorithm by Moghaddam and Yang (2000). Subsample images of 21 x 21 pixels were used in the classification task using SVM with radial basis function kernel, which is a little bit smaller than that of a pixel. In a related effort, Patil (2019) showed in his work a 100% pass test after training of the facial-based gender classification using SVM on a pre-trained dataset of the USA superstars' dataset, the "IMDB-WIKI Dataset". Barham *et al.* (2016) proposed that face detection can be treated as a preprocessing step for classification tasks. Hence, they combined face detection and gender classification using facial features for better accuracy. The SVM was used as a classifier where features like mouth, eyes, and lips were identified by the conversion of colour to grayscale images, while areas of the skin were identified by the conversion of colour to grayscale images.

Irhebhude et al. (2021) presented a technique for categorizing gender amongst dark-skinned people where SVM was used on sets of locally and publicly gathered images. Viola-Jones algorithm was used to detect faces while feature extraction was done using histogram of oriented gradient (HOG) and rotation invariant local binary pattern (RILBP) methods. The model's training was carried out via the use of an SVM classifier, and PCA was performed on both the

HOG and RILBP descriptors to extract high-dimensional features. Various success rates were recorded. However, PCA on RILBP performed best with an accuracy of 99.6% and 99.8% on the public and local datasets, respectively. Most of the research reported in the literature for gender classification adopted the SVM as a technique for the training of the model aside from the recent method that was recently proposed by the Google team, known as the FaceNet (Irhebhude et al. (2021). These models, while effective to some extent, often require large computational resources and may not always provide the desired level of accuracy. Moreover, these models often struggle with handling variations in facial expressions, lighting conditions, age, and ethnicity, further complicating the gender classification task. Against this background, this research seeks to explore the use of EfficientNet, an advanced CNN model, for gender classification tasks. EfficientNet model is chosen to address the task of gender classification by leveraging on its capability to scale depth, width and resolution uniformly unlike what obtains in a regular CNN architecture.

2. EFFICIENTNET MODEL

Deep learning architectures aim to reveal more efficient methods using smaller models. EfficientNet is one of such architectures. This type of CNN has been optimised for better performance and efficiency. The main difference between EfficientNet and a typical CNN lies in the way the network size is scaled. In a regular CNN, scaling up the network usually involves arbitrarily increasing the network's depth, width, or resolution. However, this method does not always produce optimal results, as it does not consider the balance between the different dimensions. Instead, EfficientNet utilises a compound scaling method that uniformly scales principled all three dimensions (depth, width, and resolution). This is achieved using a compound coefficient to govern the resources used for different dimensions, ensuring a balance between all three, unlike in other state-of-the-art models.

The compound scaling approach employed in EfficientNet begins with a grid search to determine the relationship existing between different scaling dimensions of the baseline network. This is done when the baseline network is fixed. Through this process, the appropriate scaling factor is arrived at for each of the three dimensions (depth, width and resolution) involved in the network. The baseline network is scaled down via the obtained scale factor (for each depth, width and resolution dimension) to realize the target network (Tan & Le, 2019). The inverted bottleneck MBConv, first proposed in MobileNetV2 (Sandler et al., 2018), constitutes the EfficientNet model's primary building block (baseline network). However, owing to a sizeable computational budget, MBConv enjoys slightly more ample patronage than MobileNetV2 in terms of floating point operations per second.

In MBConv, blocks comprise a layer that expands prior to compression of the channels to realise direct connections between bottlenecks that link considerably fewer channels than expansion layers. This framework possesses in-depth independent convolutions that decrease computation by a factor of almost k^2 (k being the kernel size, i.e. the width and height of a two-dimensional convolution window) in comparison to traditional layers (Sandler et al., 2018).
The architecture of the baseline network EfficientNet-B0 of the EfficientNet model is depicted in Figure 1.

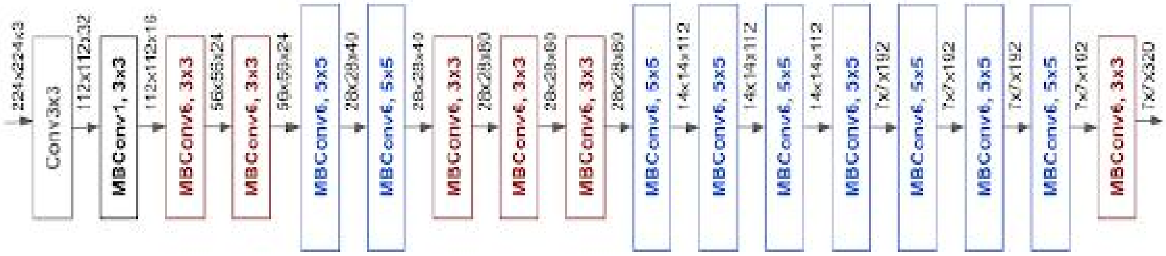

Figure 1: Architecture of EfficientNet-B0 model (Tan and Le, 2019)

3. METHOD

3.1 Model Selection and Pre-training

The foundation of the gender classification model presented in this paper is the EfficientNet-B0 architecture, which has been pre-trained on the extensive ImageNet dataset. Utilizing this pre-trained model offers a significant advantage of providing a robust initial feature representation for visual tasks. The pre-trained weights for EfficientNet-B0 was directly imported from the Keras library, thereby accelerating the model's convergence during training for our specific task. The dataset used initially for the tuning of the model is the UTKFace dataset to train the image-net weights. The model was trained, aligned and cropped using the UTKFace dataset (UTKFace, 2023).

3.2 Model Compilation, Optimization and Training

The gender classification model proposed in this paper is built upon the binary cross-entropy loss function, which is widely used as the loss function for binary classification problems. The Adam optimiser, which combines the features of AdaGrad and RMSProp, is used in the optimisation process to provide the needed good results for deep learning problems. The value of 0.001 is set as the learning rate for fine-tuning of the pre-trained weight.

3.3 Dataset Collection and Preprocessing

A dataset with Nigerian facial features is created using a custom JavaScript scraping script. The script automated the download of over 1,000 images from various online platforms such as Google Images, Flickr, and Wikimedia Commons. The script uses a set of carefully curated keywords related to Nigerian demographics, such as "Nigerian people", "Nigerian celebrities", "Nigerian models" and "Nigerian actors" among others, to ensure a broad spectrum of images in terms of gender, age, and ethnicity. It is essential to note that the scraping process was conducted in compliance with ethical standards, ensuring the privacy and consent of individuals whose images were used. A rigorous filtering process is employed to remove any low-quality images or duplicates, utilising the imagededup Python library for this purpose.

The preprocessing phase included the use of the OpenCV for face detection and cropping, focusing on the extraction of key facial features. This step is critical in eliminating background noise and concentrating on the needed facial attributes. The images were resized to 224 × 224 pixels and normalised to facilitate uniform input to the model. Additionally, data augmentation techniques, including rotations and flips, were used to enhance the robustness of the created dataset against various transformations. The images were manually labeled for gender and age group classifications. This dataset was then divided into training set that comprised 80% of the total images, this set served as the primary data source for model training, a validation set which was made up of 80% of the remaining dataset, this set is used for tuning model parameters and for initial performance evaluation as suggested by Sheoran et al. (2021), and test set, the remaining 20% of the unused dataset for the final evaluation of the model. The splitting process employed stratified sampling to ensure that the distribution of gender and age groups was consistent across the training, validation and test sets. This approach minimizes the risk of introducing bias during the model training and evaluation phases as reported by Wang et al. (2020) and Patil et al. (2021).

3.4 Model Implementation

The gender classification model proposed in this paper is implemented in Python 3.6 environment. It is selected due to its simplicity and the extensive support of analytical libraries that are crucial for tasks of this nature. This position reflects its widespread adoption in data science and machine learning (Masters et al., 2021; Rabaev et al., 2022).

Key libraries employed in the implementation are:

i. TensorFlow and Keras, which are the primary frameworks for building, training, and evaluating the deep learning model. TensorFlow provides a scalable and comprehensive ecosystem, while Keras, with its high-level API, simplifies neural network construction;
ii. OpenCV, a critical library for image processing tasks, including face detection necessary for preparing the dataset;
iii. Pandas and NumPy - Employed for data manipulation and linear algebra operations, facilitating efficient data handling and transformations.
iv. Matplotlib and Seaborn - These visualisation libraries assist in graphically representing data and model performance metrics.
v. scikit-learn - Used for generating classification reports and evaluating model performance with a variety of statistical measures.

The use of TensorFlow and Keras for model construction is in line with the recommendations of (Tan and Le, 2019; Tan and Le, 2021), while OpenCV's role in image processing mirrors the applications in Zheng et al. (2022). Keras' `Image-Data-Generator` is employed for data augmentation, which enhances the dataset with transformations to improve the model's generalization capabilities as proposed by Wang et al. (2020). The architecture is based on EfficientNet-B0 with ImageNet weights for feature extraction. The model is further refined with layers like global average pooling 2D, batch normalization and dropout. The output layer uses a sigmoid activation for binary classification of gender.

The Nigerian facial data augmentation is trained for a maximum of 15 epochs, with the early stopping mechanism to stop the training if it becomes unneeded and prevents over-fitting. This type of mechanism stops training if the validation loss stops decreasing for a specific number of epochs. To fit GPU memory constraints, the batch size is set to 32. The checkpoint with the lowest validation loss score is saved during the training for later use during evaluation and deployment. After training, the model undergoes the evaluation using the unseen Nigerian test set. As a result, multiple metrics including overall accuracy, precision, recall, and F1 score are calculated to obtain a comprehensive evaluation of the model. Following the initial evaluation, the model was subjected to a retraining phase using the curated Nigerian dataset. This dataset, enriched with a more diverse range of Nigerian faces, aims to mitigate the identified biases and improve model generalization. The retrained model was then re-evaluated using the same test set and metrics. A comparative analysis between the original and retrained models was used to quantify any reductions in bias.

4. RESULTS AND DISCUSSION

This section presented obtained results of activities highlighted in section 3 when finally carried out within the confinement of the set goal of this paper. This borders on created dataset of Nigerians' faces, results of model training with dataset and evaluation of the model performance.

4.1 Created dataset of Nigerians' faces

Figure 1 displays few of the faces that are scrapped from online platforms to create the dataset used for gender classification in this paper while Figure 2 depicts the result of data augmentation of one of the images in the dataset.

Figure 1: Snippet of the created and curated images in the facial dataset

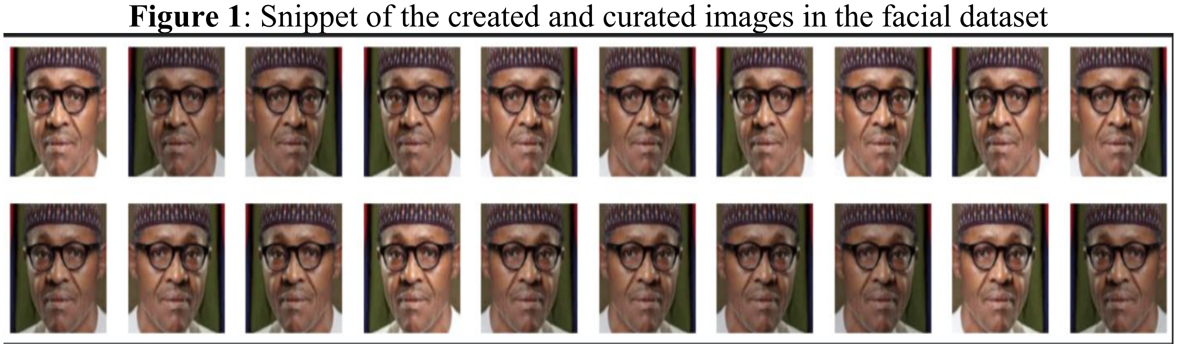

Figure 2: Result of data augmentation of an image in the created dataset

4.2 Results from pre-trained EfficientNet-B0 model

Figure 3 displays a snapshot of part of the training log obtained from the jupyter notebook ran on the UTKFace dataset for 25 epochs on the imagenet weights. It shows the dynamic learning rate in action. Figure 4 illustrates variations of the overall accuracy of the model with epoch for both training and validation dataset using pre-trained EfficientNet-B0 model.

```
Epoch 1/25
593/593 [==============================] - 260s 412ms/step - loss: 0.8273 - binary_accuracy: 0.5838 - val_loss: 0.9100 - val_binary_accuracy: 0.5268
Epoch 2/25
593/593 [==============================] - 144s 242ms/step - loss: 0.7085 - binary_accuracy: 0.6420 - val_loss: 0.7023 - val_binary_accuracy: 0.4740
Epoch 3/25
593/593 [==============================] - 144s 243ms/step - loss: 0.5035 - binary_accuracy: 0.7633 - val_loss: 1.6208 - val_binary_accuracy: 0.5270
Epoch 4/25
593/593 [==============================] - 145s 245ms/step - loss: 0.3839 - binary_accuracy: 0.8293 - val_loss: 0.3779 - val_binary_accuracy: 0.8256
Epoch 5/25
593/593 [==============================] - 143s 241ms/step - loss: 0.3452 - binary_accuracy: 0.8468 - val_loss: 0.8024 - val_binary_accuracy: 0.5257
Epoch 6/25
593/593 [==============================] - 143s 240ms/step - loss: 0.3265 - binary_accuracy: 0.8538 - val_loss: 1.5090 - val_binary_accuracy: 0.5255
Epoch 7/25
593/593 [==============================] - 144s 243ms/step - loss: 0.3125 - binary_accuracy: 0.8664 - val_loss: 0.9809 - val_binary_accuracy: 0.5257
Epoch 8/25
593/593 [==============================] - 144s 244ms/step - loss: 0.2999 - binary_accuracy: 0.8694 - val_loss: 1.7939 - val_binary_accuracy: 0.5260
Epoch 9/25
593/593 [==============================] - 146s 246ms/step - loss: 0.2868 - binary_accuracy: 0.8767 - val_loss: 1.7378 - val_binary_accuracy: 0.5260
Epoch 10/25
593/593 [==============================] - 143s 241ms/step - loss: 0.2838 - binary_accuracy: 0.8804 - val_loss: 0.7386 - val_binary_accuracy: 0.5257
Epoch 11/25
```

Figure 3: Training logs of the jupyter notebook ran on the UTKFace dataset for 25 epochs

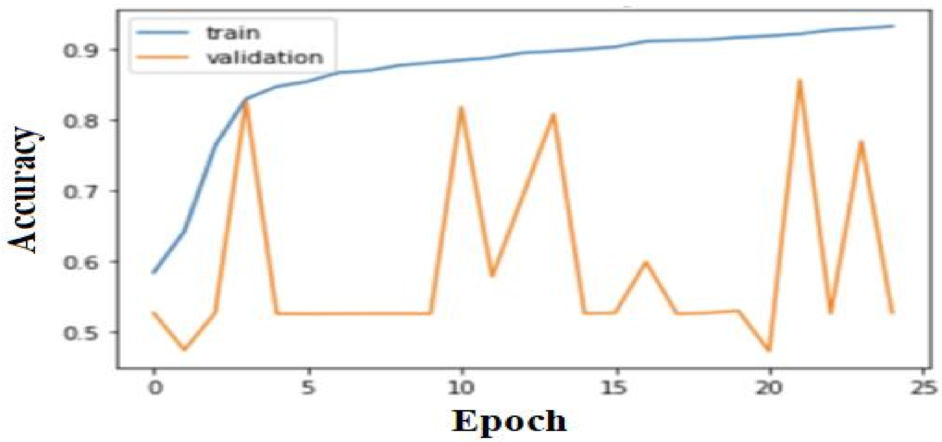

Figure 4: Variations of model's accuracy with epoch using pre-trained EfficientNet-B0 model

It is obvious from Figure 4 that the profile of training dataset is at sharp variance with that of validation set. The validation set served as a critical checkpoint in the model's training lifecycle. It provides feedback on the model's performance upon which informed decisions on hyper parameters and architecture adjustments can be based. The figure lays the first basis for the need for fine-tuning of the model in order to ensure improvement that is reflective of genuine learning rather than mere memorization of the training data.

To further justify the need for fine-tuning of the model, performance metrics adopted for the study are computed for each of class 0 (male) and class 1 (female) in the dataset used for gender classification. The initial model performance is assessed on the validation and test datasets, employing precision, recall, and F1-score metrics alongside accuracy. For training, the learning rate was 129 ms per step, loss recorded was 3.6691 and the value of the binary accuracy was 0.4764. In the case of the validation dataset, the model's learning rate was 152 ms per step, while the loss was 3.5659, and the binary accuracy was 0.4911. The results of the model's performance indices when test dataset was used had the values of the learning rate as 168 ms per step, loss of 5.2577 and binary accuracy of 0.2500.

Presented in Table 1 are obtained results of the evaluation of the pre-trained model's performance using curated Nigerian faces dataset.

Table 1: Values of performance metrics of the pre-trained model

	Precision	Recall	F1-Score	Support
		Training dataset		
Class 0	0.00	0.00	0.00	310
Class 1	0.49	1.00	0.66	282
Macro Average	0.24	0.50	0.32	592
Weighted Average	0.23	0.48	0.31	592
Accuracy			0.48	592
		Validation dataset		
Class 0	0.00	0.00	0.00	57
Class 1	0.49	1.00	0.66	55
Macro Average	0.25	0.50	0.33	112
Weighted Average	0.24	0.49	0.32	112
Accuracy			0.49	112
		Test dataset		
Class 0	0.00	0.00	0.00	12
Class 1	0.25	1.00	0.40	4
Macro Average	0.12	0.50	0.20	16
Weighted Average	0.06	0.25	0.10	16
Accuracy			0.25	16

Analysing precision and recall for different classes provides insights into potential biases, guiding the subsequent fine-tuning process. The model's accuracy on the validation set is found to be 49.11%. Precision for class 1 was 0.49, suggesting that roughly half of the positives were correctly identified. Recall for class 1 was 1.00, indicating all actual positives were identified. Conversely, class 0 had a precision and recall of 0.00, indicating a failure in identifying any true positives for this class. Weighted means for precision, recall and F1-score were 0.24, 0.49 and 0.32, respectively. Meaning of these values show a bias of the model in favour of class 1 rather than class 0, in such a high rate it requires immediate attention. These results highlight the need for resetting the model to amplify the discriminatory function on class 0 without losing the performance of class 1. The above changes are imperative to build a model that is capable of displaying invariant performance on all classes. The validation accuracy metrics for our model, fine-tuned on the Nigerian facial dataset, has 49.11%, which shows some performance dip as the training set has lower performance at 47.64%. These values indicate difficulties that the model encounters in its potential to generalize and properly distinguish characteristics within the classes. Macro and weighted averages metrics provide insights into the model's overall performance in terms of class imbalance. The macro average F1-score of '0.20' imply major performance differences between classes. The weighted average F1-score, like the others at 0.20, indicates that the class distribution is relatively balanced due to aligning of macro and weighted averages. That indicates that all classes demonstrate the same level of performance indicators though they have low scores.

Performance indicators uncover crucial hints. The model indicates impeccable recall for class 1 while class 0 has a recall of 0.00, i.e. maybe the model is biased towards class 1, and potentially at the expense of class 0. This inconsistency can signify that the model is over-fitting with respect to class 1 or that the class 0 is less represented or it is a challenging area for the model to learn. Also, as the poor F1-scores imply, there is a need for further improvement especially on the model's ability to balance precision and recall for class 0.

4.3 Results from fine-tuned EfficientNet-B0 model

Model performance is shown to depend on other factors besides accuracy often and it is crucial to use a strategic fine-tuning approach to boost performance of a model evenly across all classes without any inherent bias towards any specific class. Figure 5 shows the profile of the accuracy of the fine-tuned model with epoch. The fine-tuning phase addresses the under-representation and bias identified in the initial model evaluation. This involves augmenting the training dataset and adjusting the model's hyper parameters. The retraining process is guided by a revised loss function and learning rate adjustments, ensuring a balanced and fair model performance across different demographics. Presented in Table 2 is the results of model's performance after fine-tuning. In the results, the training learning rate was 56 ms per step with a loss of 0.7557 and a binary accuracy of 0.7551, while for the validation dataset, the learning rate was 57 ms per step with a loss of 0.4104 and a binary accuracy of 0.8214. The test dataset had its learning rate as 26 ms per step; loss of 0.4964 and binary accuracy of 0.8125.

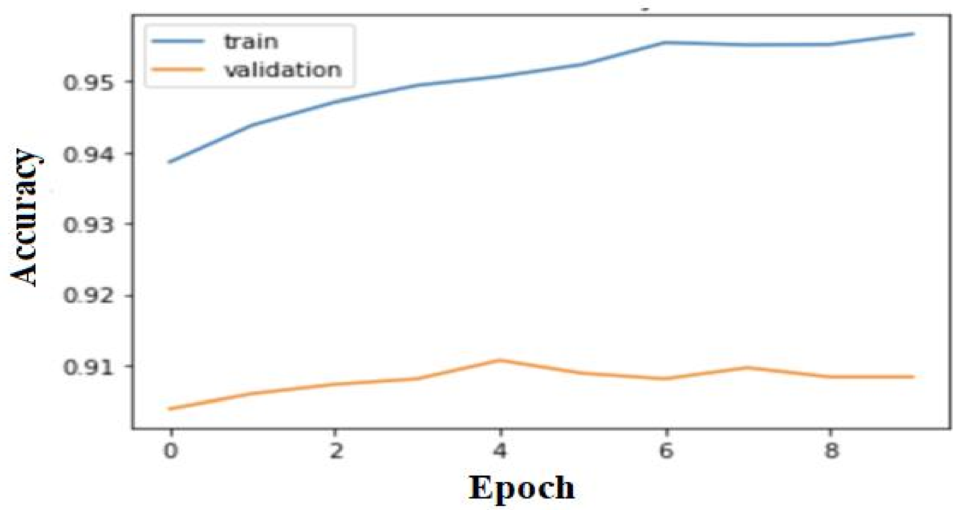

Figure 5: Variations of model's accuracy with epoch using fined-tuned EfficientNet-B0 model

It is obvious from entries of Table 2 that the fine-tuned model yields improved results in terms of the recorded values for performance metrics used, unlike what is obtained in the pre-trained model as reported in Table 1. The accuracy improved from initial 0.48 to 0.76 (training dataset); 0.49 to 0.82 (validation dataset) and 0.25 to 0.81 (test dataset). Although the difference in the accuracy of the fine-tuned model when validation and test datasets are used appears marginal, the key victories are seen in the precision and recall metrics. Precision for class 1 saw a significant increase, indicating that the fine-tuned model's ability to correctly predict gender classification had improved for the previously underperforming class. Recall for class 1 experienced the most notable improvement, suggesting that the model was now more capable of identifying all actual positives in this class. The F1-scores for both classes levelled, indicating a more balanced performance between precision and recall and thus, a more equitable model.

Table 2: Values of performance metrics of the fine-tuned model

	Precision	Recall	F1-Score	Support

Training dataset				
Class 0	0.73	0.84	0.78	306
Class 1	0.80	0.66	0.72	286
Macro Average	0.76	0.75	0.75	592
Weighted Average	0.76	0.76	0.75	592
Accuracy			0.76	592
Validation dataset				
Class 0	0.82	0.88	0.85	64
Class 1	0.82	0.75	0.78	48
Macro Average	0.82	0.81	0.82	112
Weighted Average	0.82	0.82	0.82	112
Accuracy			0.82	112
Test dataset				
Class 0	0.67	0.80	0.73	5
Class 1	0.90	0.82	0.86	11
Macro Average	0.78	0.81	0.79	16
Weighted Average	0.83	0.81	0.82	16
Accuracy			0.81	16

After fine-tuning, both the macro and the weighted averages for precision, recall, and the F1-score are seen increasing, confirming that the improvements of the model's performance do not come at the expense of over favouring one class over the other. The fine-tuned model’s performance metrics go beyond just numerical gains; it signifies a journey towards fair usage of machine learning in gender classification. The fine-tuning process highlights the importance of a holistic approach to machine learning development. It emphasizes that performance metrics must be interpreted within the context of the model's application and the societal implications of its deployment. The positive results from fine-tuning also prompted a reflection on the iterative nature of machine learning development, where continuous improvement is both possible and necessary.

Comparative analysis of the results obtained in this work with two of those available in the literature shows that Efficient-Net fares better when employed for gender classification task. Table 3 presents the result of the comparative analysis of Efficient-Net with two other methods in the literature for gender classification.

Table 3: Comparative analysis of EfficientNet-B0 with two other methods in the literature

Gender Classification Works	Methods Employed	Accuracy (%)
Gündüz and Cedimoğlu (2019)	CNN	72.20
Gündüz and Cedimoğlu (2019)	AlexNet	65.63
This work	EfficientNet-B0	81.00

5. CONCLUSION

A gender classification system using facial data utilising the EfficientNet-B0 model was presented in this paper. A significant achievement of this study was the notable rapidity in improving model accuracy with a relatively small number of training epochs. The model reached a high level of performance, evidenced by substantial gains in accuracy, precision, recall, and F1-score, within just 55 epochs (35 initial + 20 for fine-tuning). This efficiency underscores the potential of EfficientNet in gender classification tasks, particularly in scenarios where computational resources or time are limited. The EfficientNet model demonstrated its prowess, not just in accuracy but also in learning speed, validating its application in gender classification. This outcome is significant as it suggests that sophisticated models like EfficientNet can be trained effectively with limited data and computational time, broadening their applicability in practical settings.

The accuracy rate obtained when the result of gender classification using EfficientNet-B0 is compared with those utilizing CNN and AlexNet revealed that the method not only compared well but showed improved performance. With this procedure, the accuracy rate achieved with EfficientNet-B0 compared well with the accuracy rates of similar studies in the literature, and better result was observed. In future studies, it is planned to make use of other higher variants of EfficientNet model with the dataset containing more images.

REFERENCES

[1] Hager, K. 2022. Image Classification, Deep Learning and Convolutional Neural Networks. *International Journal of Analytical and Experimental Modal Analysis,* 2(1):14–34.

[2] Barham, P., Chen, J., Chen, Z., and Davis, A. 2016. TensorFlow : A system for large-scale machine learning. *International Journal of Computer Applications,* 1(18):11–19.

[3] Tahmina S, Daniel M, Gunasegaran P. Emergency Peripartum Hysterectomy. 2017. A 14-Year Experience at a Tertiary Care Centre in India. J Clin Diagn Res. 11(9):QC08-QC11. doi: 10.7860/JCDR/2017/26769.10613. Epub 2017 Sep 1. PMID: 29207784; PMCID: PMC5713806.

[4] Bagrami, E., and Halili, F. 2022. Python Software for Gender Detection Using Artificial Intelligence Libraries TensorFlow. *International Research Journal of Modernization in Engineering Technology and Science,* 4(3):1–6.

[5] Moghaddam B. and Yang M.-H. (2000). Gender Classification with Support Vector Machine. Mitsubishi Electric Research Laboratory 201 Broadway Cambridge, MA 02139 USA https://www.researchgate.net/publication/2460511_Gender_Classification_with_Support_Vector_Machine

[6] Patil, B. G. 2019. Human Gender Classification using Machine Learning. *International Journal of Engineering Research and Technology (IJERT),* 8(12):474–477.

[7] Irhebhude M., K. Adeola and F. Abdullahi. 2021. Northern Nigeria Human Age Estimation from Facial Images using Rotation Invariant Local Binary Pattern Features with Principal Component Analysis. *Egyptian Computer Science Journal,* 45(1):12-28.

[8] Mingxing T., and L. Quoc. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 9(5):1-6.

[9] Sandler M., A. Howard, M. Zhu, A. Zhmoginov and L. -C. Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

[10] Atila, U. U. Murat, A. Kemal, U. Emine. (2021). Plant leaf disease classification using EfficientNet deep learning model. *Ecological Informatics*. 61:101182.

[11] 10.1016/j.ecoinf.2020.101182.
[12] UTKFace dataset. 2023. https://www.kaggle.com/datasets/jangedoo/utkface-new
[13] Sheoran V., J. Shreyansh and B. Tanisha. 2021. Age and Gender Prediction using Deep CNNs and Transfer Learning. Arxiv
[14] Wang J., Y. Liu, H. Zhanqiang, H. Weifeng and L. Junwei. 2020. Multi-Label Classification of Fundus Images With EfficientNet. *IEEE Access.* 8:212499-212508.
[15] 10.1109/ACCESS.2020.3040275.
[16] Patil J., T. Rohit, D. Yash, K. Rohit and T. Nikhil. 2021. Age and Gender Detection Using CNN. *International Journal of Scientific Research in Science and Technology.* 29-33. 10.32628/IJSRST21835.
[17] Masters D., L. Antoine, E.-R. Zach and L. Carlo. 2021. Making EfficientNet More Efficient: Exploring Batch-Independent Normalization, Group Convolutions and Reduced Resolution Training. Preprint
[18] Rabaev I. A. Izadeen, W. Odai and L. Marina. 2022. Automatic Gender and Age Classification from Offline Handwriting with Bilinear ResNet. *Sensors. 22. 9650. 10.3390/s22249650.*
[19] Zheng B., Gao A., Huang X., Li Y., Liang D., and Long X. 2022. A modified 3D EfficientNet for the classification of Alzheimer's disease using structural magnetic resonance images. *IET Image Process,* 17:77-87.
[20] Gündüz, G., and Cedimoğlu, İ. H., 2019, Derin Öğrenme Algoritmalarını Kullanarak Görüntüden Cinsiyet Tahmini, Sakarya University Journal of Computer and Information Sciences, 2(1):9-17.