Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME 2023)
19-21 July 2023, Tenerife, Canary Islands, Spain

Enhanced Yoga Posture Detection using Deep Learning and Ensemble Modeling

Shah Imran
Computer Science and Engineering
Brac University)
Dhaka, Bangladesh
shah.imran@g.bracu.ac.bd

Zarif Sadman
Computer Science and Engineering
Brac University
Dhaka, Bangladesh
zarif.sadman@g.bracu.ac.bd

Abidul Islam
Computer Science and Engineering
Brac University
Dhaka, Bangladesh
abidul.islam@g.bracu.ac.bd

Dewan Ziaul Karim
Computer Science and Engineering
Brac University
Dhaka, Bangladesh
ziaul.karim@bracu.ac.bd

***Abstract*—Yoga is one of the best at-home exercises for maintaining our physical health. However, yoga is all about successfully performing the 82 Yoga Asanas throughout the course of six classes. Lamentably, not everyone has the knowledge or can perform yoga accurately. So to do yoga poses correctly we will have to find a yoga instructor, but it can be very hard and expensive to find yoga instructors considering all possible general situations and status. Using Deep Learning(DL) and modifying some pre-trained models to some extent can be a possible solution to detect yoga pose and class separately, which can eventually help general people. This work proposes a detailed experiment using two pre-trained CNN models along with an ensemble model to detect yoga poses accurately. The work was done on a total of 18488 images divided into 6 major yoga classes and 82 different poses. The aftermath of using ensemble modeling was instrumental as it was able to detect yoga poses with a 95% chance of assurance.**

***Index Terms*—Yoga poses, Transfer learning, Posture detection, Ensemble model, Deep learning**

I. Introduction

Yoga is a spiritual discipline that is based on tremendously tenuous science. According to multiple types of research and studies, yoga can have a massive impact on both the human body and mind. If yoga is performed correctly then, it is possible to use it as a treatment for stress-related disorders such as hypertension, back pain, asthma, and many others. Furthermore, performing a Yoga stance incorrectly may be damaging to one's health. However, there is an overlap[1][2] issue that affects the system's accuracy. To address this problem, we present a Deep Learning model that combines modified Keras Applications (DenseNet201, Xception, MobileNet) with ensemble modeling to determine difficult postures. Our model offers the following benefits:

- Better accuracy for 82 and 6 poses than any other research.
- Applying the ensemble model to boost the accuracy.

Classification is an important problem in the Computer Vision field. Many previous types of research have been conducted to classify different yoga postures accurately using image datasets, videos, and even real-time datasets. In this paper, we have used an image dataset named "Yoga-82" to classify different yoga postures. Although previously some works have been done using this dataset, our goal is to outperform all the previous works that used the Yoga-82 dataset in terms of accuracy. To achieve the goal, different transfer learning models are modified to get the best accuracy and then ensemble modeling is used to boost the accuracy even more.

II. Related Works

A. Previous Works

According to T. K. K. Maddala *et al.*[3] their motive was to advance with a goal of improvement in accuracy. Over the course of four weeks, they used a nine-camera mocap (Motion Capture) system to collect data on 42 different yoga poses performed by ten different people in ten different orientations. They've employed JADMs and RGB to color it (Red-Green-Blue). Also, according to S. Kothari *et al.*[2], OpenPose is being used for the first key-point extraction of human joint positions. The algorithm can efficiently classify yoga poses in both prerecorded films and in real-time (Long Short Term Memory). TensorFlow-Keras, OpenPose, NumPy, and Scikit Learn are among the Python packages used to create the models. This project's data set is part of the Open-Source collection and is freely accessible. There are 88 videos in all, each lasting 1 hour, 6 minutes, and 5 seconds. S. Liaqat *et al.*[1], Their research contains the uses of an advanced proposal of a

979-8-3503-2297-2/23/$31.00 © 2023 IEEE

hybrid approach that involves both machine learning and deep learning at the same time. According to the paper, using this hybrid approach the benefit was clearly visible through the accuracy and efficiency check. The paper mentioned that the DL method was used to reduce time in detecting the body postures and it was also being cleared that, no feature engineering was required. The proposed DL methods or classifiers were used as the input of Convolutional Neural Networks(CNN) and Long-Short Term Memory(LSTM) architecture. The usage of the hybrid approach helped the system to get a performance boost which was clearly visible through the accuracy of the whole process.

The main motive of G. G. Chiddarwar *et al's*[4] paper was to lessen the struggle of the self-learning process of the yoga poses. According to the paper, the best method can be distinguished by the usability of Android applications. Deep learning(DL) methods were used to support the proposed system in the paper to predict the yoga postures precisely. The paper was well explained and detailed though there was no mention of the achieved accuracy of the whole process. Z. Cao *et al*[5] have created a system that operates in real-time for multi-person 2D pose identification and limb orientation over the image domain. They used Part Affinity Fields (PAFs), which may be defined by a 2D vector field to encode the location, to accomplish the first bottom-up depiction of association scores. After that, they generated the ground truth confidence maps from annotated 2D key points to examine during the training. They could use a 3D posture to implement the approach and verify that all failure instances are replicated completely. J. Jose and S. Shailes *et al*[6] focused on Yoga Asana recognition in their study, because Yoga has great medical value when done correctly. They employed a data set developed by Anastasia Marchenkova, in which 700 photographs of ten yoga asanas were dispersed among ten courses. The photos were reduced to 100*100 pixels, and 4608 features were retrieved, according to the report. At last, the authors achieved an accuracy of 85 percent, despite the fact that Hu moments (73.5%), Histogram of Oriented Gradients(70.3%), and CNN were all significantly lower.

S. K. Yadav *et al*[7] tried to detect the main 6 asanas correctly using Deep Learning(DL). They gathered data by having 15 people (10 men and 5 women) do each of the six asanas. When using video input, they achieved more than 99 percent accuracy. For the first time, a deep learning pipeline from beginning to end has been used in a study to identify yoga poses from videos. M. Verma *et al*[8] acknowledges that yoga poses are miscellaneous and proposed to tackle this problem using fine-grained hierarchical categorization. This article uses hierarchical labeling of human poses instead of key points and skeletal annotations. Hierarchical labeling has advantages over flat N-way categorization. Their study has two sections of experiments. They started with Yoga-82 data. In the second section, three CNN architectures are described for performance analysis using hierarchical labels and the Yoga-82 class structure. This study uses the Yoga-82 dataset. This data set has 28.4k yoga posture pictures. These pictures include 82 classes and 20 subclasses. These 20 subclasses form six super-classes. The hierarchical organization enhances system performance. Third-level classifier accuracy increased from 74.9% to 79.35%. The data set's hierarchy information improves performance through increased learning supervision. Future class labels will have specific limitations.J. Palanimeera and K. Ponmozhi *et al*[9], proposed a system that will use machine learning modules and a large number of a huge dataset to perform training. According to the paper, the data set that was used for the system contained a total of 12 different and individual yoga poses performed by certified trainers. There were a few drawbacks to the proposed system one of them is the data collection procedure as a webcam was used to collect image data which is not an efficient tool to capture images.

B. Related Models & Classifiers

Facebook AI research introduced DenseNet[10] in 2016's "Densely Connected Convolutional Networks"[11] There are many versions with different depths, from 121 layers and 0.8 million parameters to 264 layers and 20.2 million parameters. Dense201[12] has 201 layers, is 80 MB, and each inference step takes 127.24 ms for CPU and 6.67 ms for GPU. Xception[13] is a novel deep convolutional neural network architecture. Xception proposes more efficiency which performs just better than Inception V3 on the ImageNet dataset.

MobileNets[14][15] uses depthwise separable convolutions to generate lightweight deep neural networks and it optimizes for speed. One ResNet model is ResNet50(Residual Neural Network)[16]. 1 Average Pool layer, 1 MaxPool layer, 48 Convolution layers. 3.8 109 Floating point operations are also included. ResNet-101[17] has 101 layers. The pre-trained network can classify 1000+ item types. Ensemble modeling[18] improves categorization and prediction models' computing capability. An ensemble model can learn better from training data by combining optimal aggregation combinations of basic models or "weak learners." Ensemble modeling enhances performance by reducing variance, bias, and noise. Large-data learning and feature selection benefit from ensemble modeling. Random Forest is a popular supervised learning approach for regression and classification using non-linear or real-time data. Random forests[19] have several decision trees to improve dataset predictions. To predict the ultimate output, the random forest votes on the most likely tree predictions. Because the random forest utilizes several trees to classify the dataset, some trees may correctly anticipate its class. All the trees collectively predict the right result. Each tree's predictions must have a minimal correlation.

III. Data Collection & Pre-Processing

The data set we used is "Yoga-82", which is a secondary data set to collect our images of various yoga poses. In this data set, we have a total of 82 different classes each defining a different individual pose. In these 82 files, we have 23945 different images of different poses. Images in these files are taken from different angles and these images contain

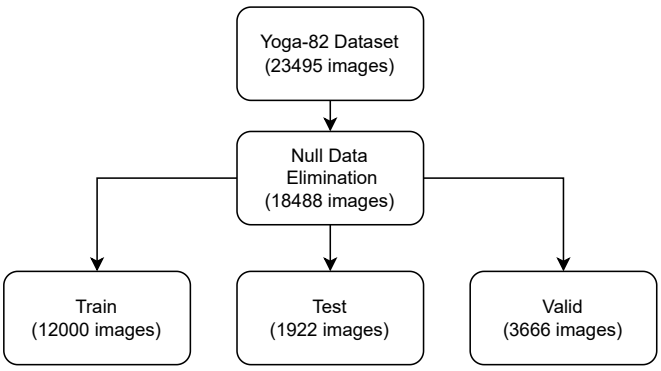

Fig. 1. Dataset split

different backgrounds such as both indoors and outdoors. To pre-process the data we used several Keras library functions for different models which process the data or the input images and make it ready according to the pre-trained model that is being used in the AI system. We have scaled them in (224, 224) format. Training data overfitting is far less likely when a model is subjected to a larger number of training examples. Deep learning models are designed to find the most efficient route from input parameters to output ones. To do this, it needs to pick out the right features and look for patterns that aren't obvious in the data.

As we used a large size of image dataset of different yoga poses, there were some null and garbage data present in the dataset. To eliminate these unusable data we excluded a few types of images and images which are not compatible with loading. After this null data elimination process, we gained a total of 18488 images as our dataset. We also did proper labeling for different poses and categories to help our model to train more easily and also, to improve the prediction accuracy of our model.

IV. Methodology

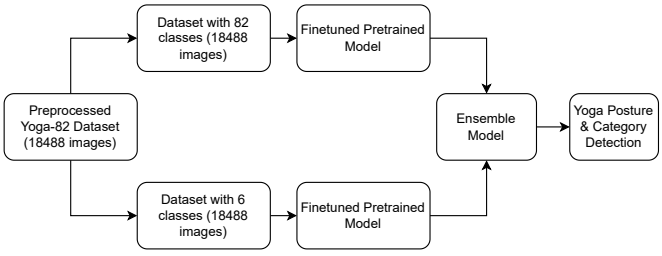

Fig. 2. Yoga Pose Detection Workflow

A. Dataset Preparation

After the dataset preprocessing, we created a dataset taking all of the 82 poses from the preprocessed dataset and labeled them into 6 categories, which include Balancing, Inverted, Reclining, Sitting, Standing, and Wheel. Then we split both datasets into a 7:2:1 (train:test: valid) ratio. From the workflow we can see, both of the datasets have exactly the same number of images (18,488 images) because both of them are the same dataset with different labeling.

B. Xception Model

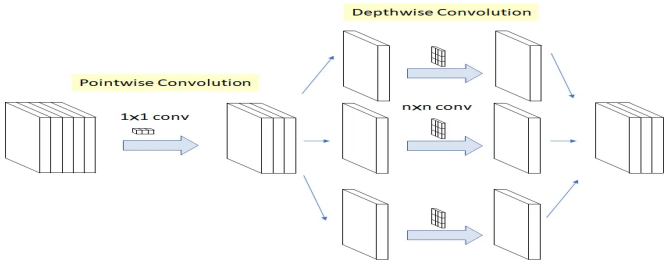

Fig. 3. Xception architecture

We used several pre-trained models to predict the poses and Xception[13] gave us the best result among them. However, Xception is a 71-layer deep convolutional neural network. Xception performs slightly better on the Imagenet dataset than Inception V3, and when used to a larger image classification dataset consisting of 350 million images and 17,000 classes, it achieves much better results than Inception V3. Because the Xception contains the same amount of parameters as the Inception V3 architecture, the performance increases are not the result of greater capacity but rather of more productive use of model parameters. Unfortunately, it was not enough to produce a good accuracy for the 82 poses. That is why we had to finetune the model.

C. Fine Tuned Pre-trained Model

We have used several Keras pre-trained CNN models (MobileNetV1, MobileNetV2, ResNet50, ResNet101, and Xception) to train both of the categories (82 classes and 6 classes). We wanted Top-2, Top-3 accuracy for category postures and Top-5, Top-10 accuracy for 82 yoga poses. However, using these pre-trained models did not give us high accuracy, so we had to fine-tune it. The steps we followed to fine-tune the Xception model:

- At first, we customized the Xception model by replacing its input layer with another trainable input layer.
- Then we set the hidden layers to non-trainable and added a trainable GlobalAveragePooling2D layer and a Dense layer at the end.
- After compiling with a learning rate of 0.0001, we trained the model with 12900 images.
- Then we compiled the model again while setting the nontrainable parameters to trainable.

D. Ensemble Model

Our vision was, if we can predict the category of any of the Yoga postures, we have to predict in less than 82 poses as each category consists of not more than 20 types of postures. However, to reach that stage, we applied the ensemble model[20]. We predicted 1922 images on both of the Keras models(for 82 classes and 6 classes) and combined the

prediction results for the ensemble model. We employed one hot encoding on these features and put them in the random forest to improve the prediction accuracy[21]. However, the random forest was able to gain a higher accuracy than any of the two Keras models. We were able to boost accuracy for the 82 classes and 6 classes.

V. Results & Analysis

To testify to the result analysis after the training and test for all the models for the 82 poses and 6 basic classes we attained different accuracy measurements from different models. For each model, we attained Top-5 accuracy and Top-10 accuracy for predicting 82 classes along with the validation accuracy. Also, we attained Top-2 accuracy and Top-3 accuracy for using 6 basic classes along with the validation accuracy which is given below in the following table:

TABLE I.
Accuracy table of Yoga-82(82 classes)

Models	Test Accuracy(%)	Top-5 accuracy(%)	Top-10 accuracy(%)
MobileNet	75.83	91.16	95.14
MobileNetV2	70.19	88.93	93.64
ResNet50	74.44	92.80	96.26
ResNet101	76.46	92.03	95.69
Xception	**82.52**	**94.54**	**96.81**

TABLE II.
Accuracy table for Yoga-82(6 classes)

Models	Test Accuracy(%)	Top-5 accuracy(%)	Top-10 accuracy(%)
MobileNet	82.86	93.33	95.49
MobileNetV2	86.46	95.15	95.99
ResNet50	86.25	95.34	97.75
ResNet101	86.06	95.17	98.09
Xception	**92.42**	**95.34**	**97.58**

In I and II test accuracy represents how accurately the models can classify the yoga categories(6 super-classes) and the yoga poses(82 classes) respectively. We have predicted Top-5 and Top-10 categorical accuracy for the 82 classes and Top-2 and Top-3 categorical accuracy for the 6 super-classes.

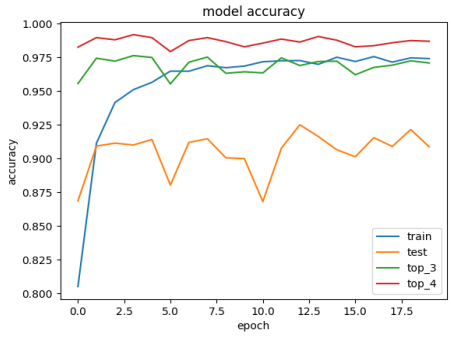

Fig. 4. Accuracy graph for 6 classes using Xception

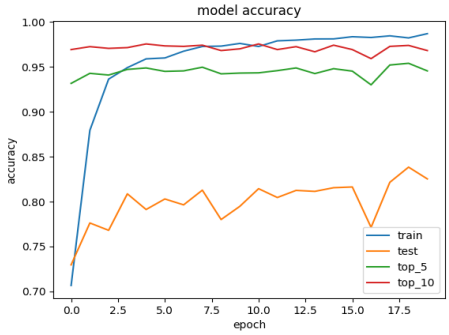

Fig. 5. Accuracy graph for 82 classes using Xception

Figure 4 and 5 represent the accuracy graph of the Xception, which is our best-performing model for predicting the categories and the poses of any Yoga postures.

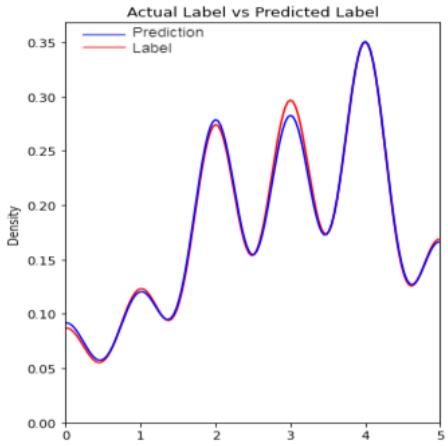

Fig. 6. Ensemble label vs prediction for 6 classes

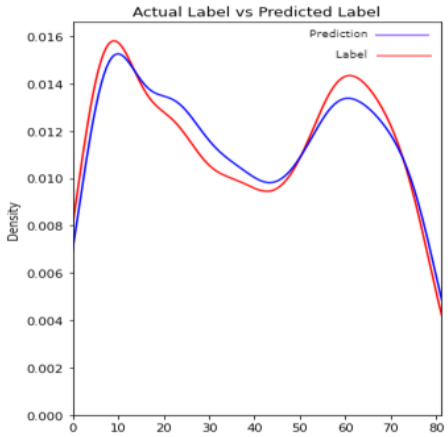

Fig. 7. Ensemble label vs prediction for 82 classes

However, from Figures 6 and 7, we can see the actual(Blue) vs the prediction(Red) values for the 6 and 82 classes respectively of the test dataset. Here, the Y-axis illustrates the test sample density of each category and the X-axis demonstrates the category of the test images. By observing the curves, we

can say that for 6 classes the model can predict the results very accurately. However, for 82 classes poses with fewer images are giving away incorrect predictions.

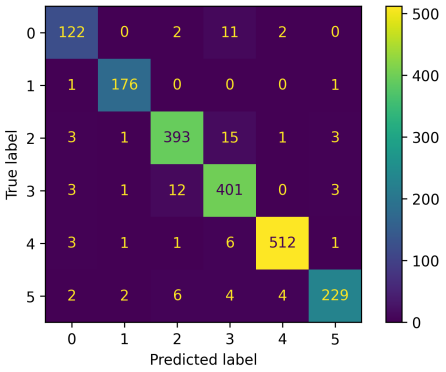

Fig. 8. Ensemble model confusion matrix for 6 classes

By observing the confusion matrix(Fig:8) based on the ensemble model, we can evaluate the predicted and true values of the test samples of the 6 classes more easily. For example, from (Fig: 10), we can see that for Category 2, the model predicted 122 images correctly and the other 15 images are not predicted correctly.

TABLE III.
Classification report for 6 classes

Category	pecision	recall	f1-score	support
0	0.92	0.89	0.90	137
1	0.97	0.99	0.98	178
2	0.94	0.95	0.95	416
3	0.93	0.95	0.94	420
4	0.99	0.97	0.98	524
5	0.95	0.94	0.94	247
accuracy			**0.95**	**1922**
macro_avg	**0.95**	**0.95**	**0.95**	**1922**
weighted_avg	**0.95**	**0.95**	**0.95**	**1922**

Furthermore, from the classification report(III) given above, we can see the classification report of the ensemble model where we have shown the test batches. From the [22]precision column, we can see how well each class is predicted divided by all the times the model has predicted the class(rightly or wrongly). However, in the [22]recall column, the scores represent the ratio of the number of correctly predicted members in a class divided by the number of total members. Then, the [22]f1 score indicates, how good precision and recall values. Finally, the support column represents the occurrences of individual classes in the data set.

A. Comparison with Previous Works

Although there have been many works regarding yoga pose estimation have already been done, there are very few works that have been done using the Yoga-82 dataset which is the same dataset that we used. M. Verma *et al*[8] used different transfer learning models and presented several hierarchical variants of DenseNet. We have managed to outperform their transfer models in terms of accuracy. Using state-of-the-art transfer learning models such as ResNet-50, ResNet-101, MobileNet, and MobileNet-V2, they managed to classify the 82 poses with respectively 63.44 %, 65.84%, 67.55%, and 71.11% accurately for Top-1 accuracy and respectively 82.55%, 84.21%, 86.81%, and 88.50% accurately for Top-5 accuracy. They also managed to improve the Top-1 accuracy using their proposed hierarchical variants of DenseNet to 79.35%. In our work, we have managed to outperform all of their transfer learning models and their proposed hierarchical variants of DenseNet in terms of accuracy for both Top-1 and Top-5 accuracy.

VI. Conclusion & Future Plan

Body posture detection is a popular topic in computer vision. In this research, we used deep learning to classify yoga positions from images. We constructed a deep learning system that trains and tests on a given dataset of 82 different yoga positions that can be divided into 6 primary super classes and then predicts the class and location of any given yoga posture as exactly as possible. We tried MobieNet, ResNet, and Xception to improve our system's classification of yoga positions. We also used ensemble modeling to classify yoga stance photographs more accurately. Using ensemble modeling for 82 classes and 6 basic superclasses, we increased our accuracy to 85% and 95% respectively. In short, we used one of our pre-trained models' top outputs and tried to improve it with our ensemble modeling for predicting yoga positions with categorization. In the future, We wish to use OpenPose to detect the Joint Angular Displacement Map (JADM). We also plan to develop an app for smartphones that can detect the correctness of a pose given image input. We have encountered overlapping issues with our present system. We will handle the overlapping issues by conducting additional studies.

References

[1] S. Liaqat, K. Dashtipour, K. Arshad, K. Assaleh, and N. Ramzan, "A hybrid posture detection framework: Integrating machine learning and deep neural networks," *IEEE Sensors Journal*, vol. 21, no. 7, pp. 9515–9522, 2021.

[2] S. Kothari, "Yoga pose classification using deep learning," 2020.

[3] T. K. K. Maddala, P. Kishore, K. K. Eepuri, and A. K. Dande, "Yoganet: 3-d yoga asana recognition using joint angular displacement maps with convnets," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2492–2503, 2019.

[4] G. G. Chiddarwar, A. Ranjane, M. Chindhe, R. Deodhar, and P. Gangamwar, "Ai-based yoga pose estimation for android application," *Int J Inn Scien Res Tech*, vol. 5, pp. 1070–1073, 2020.

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7291–7299.

[6] J. Jose and S. Shailesh, "Yoga asana identification: A deep learning approach," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 1110, 2021, p. 012 002.

[7] S. K. Yadav, A. Singh, A. Gupta, and J. L. Raheja, "Real-time yoga recognition using deep learning," *Neural Computing and Applications*, vol. 31, no. 12, pp. 9349–9361, 2019.

[8] M. Verma, S. Kumawat, Y. Nakashima, and S. Raman, "Yoga-82: A new dataset for fine-grained classification of human poses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 1038–1039.

[9] J. Palanimeera and K. Ponmozhi, "Classification of yoga pose using machine learning techniques," *Materials Today: Proceedings*, vol. 37, pp. 2930–2933, 2021.

[10] G. Huang, S. Liu, L. Van der Maaten, and K. Q. Weinberger, "Condensenet: An efficient densenet using learned group convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2752–2761.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[12] A. Jaiswal, N. Gianchandani, D. Singh, V. Kumar, and M. Kaur, "Classification of the covid-19 infected patients using densenet201 based deep transfer learning," *Journal of Biomolecular Structure and Dynamics*, vol. 39, no. 15, pp. 5682–5689, 2021.

[13] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[14] A. G. Howard, M. Zhu, B. Chen, *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[15] Q. Xiang, X. Wang, R. Li, G. Zhang, J. Lai, and Q. Hu, "Fruit image classification based on mobilenetv2 with transfer learning technique," in *Proceedings of the 3rd International Conference on Computer Science and Application Engineering*, 2019, pp. 1–7.

[16] A. S. B. Reddy and D. S. Juliet, "Transfer learning with resnet-50 for malaria cell-image classification," in *2019 International Conference on Communication and Signal Processing (ICCSP)*, IEEE, 2019, pp. 0945–0949.

[17] P. Ghosal, L. Nandanwar, S. Kanchan, A. Bhadra, J. Chakraborty, and D. Nandi, "Brain tumor classification using resnet-101 based squeeze and excitation deep neural network," in *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, IEEE, 2019, pp. 1–6.

[18] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, Springer, 2000, pp. 1–15.

[19] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS journal of photogrammetry and remote sensing*, vol. 67, pp. 93–104, 2012.

[20] R. Zwanzig, "Ensemble method in the theory of irreversibility," *The Journal of Chemical Physics*, vol. 33, no. 5, pp. 1338–1341, 1960.

[21] K. Fawagreh, M. M. Gaber, and E. Elyan, "Random forests: From early developments to recent advancements," *Systems Science & Control Engineering: An Open Access Journal*, vol. 2, no. 1, pp. 602–609, 2014.

[22] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *European conference on information retrieval*, Springer, 2005, pp. 345–359.