2024 3rd International Conference for Innovation in Technology (INOCON)
Karnataka, India. Mar 1-3, 2024

Fine-tuned EfficientNet and MobileNetV2 Models for Intel Images Classification

Arshleen Kaur
Chitkara University Institute of Engineering and Technology, Chitkara University,
Punjab, India
arshleen.kaur@chitkara.edu.in

Vinay Kukreja
Chitkara University Institute of Engineering and Technology, Chitkara University,
Punjab, India
vinay.kukreja@chitkara.edu.in

Nitin Thapliyal
Computer Science and Engineering, Graphic Era Hill University,
Dehradun, Uttrakhand, India, 248002
nitinthapliyal@gehu.ac.in

Manisha Aeri
Computer Science and Engineering Graphic Era Deemed to be University,
Dehradun, Uttarakhand, India.
maniaeri16@gmail.com

Rishabh Sharma
Chitkara University Institute of Engineering and Technology, Chitkara University,
Punjab, India
rishabh.sharma@chitkara.edu.in

Shanmugasundaram Hariharan
Proffessor, Department of Computer Science and Engineering,
Vardhaman College of Engineering,
Hyderabad, India
hariharan.s@vardhaman.org

2024 3rd International Conference for Innovation in Technology (INOCON) | 979-8-3503-8193-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/INOCON60754.2024.10512279

***Abstract*— This study looks at using two types of deep learning models, MobileNetV2 and EfficientNet, for sorting pictures in Intel's image collections. The study looks at how well it works by testing with different numbers for learning rate; these are 0.01 and 0.09 faster than the last ones, seen as goals to reach success high in speed needed fast. The study is about checking how good the models' guesses are since it shows how well they work. The MobileNetV2 model did very well, reaching the best accuracy of 99.67% at point 50 in training sessions. Instead, the EfficientNet showed a very good accuracy of 92.11% at the same time period. The side-by-side look shows that MobileNetV2 does better than EfficientNet in image sorting jobs, winning in this special situation. Changing the learning rate helps us understand how sensitive our models are to fine-tuning. This can help make training smoother and better. The results show how important it is to choose the right model and adjust its settings to get better accuracy when classifying images. This could have many uses in different areas, like seeing what objects are shown on images or finding out patterns from them.**

Keywords—Intel images, natural scenes, image classification, MobileNetV2, EfficientNet.

I. INTRODUCTION

Image classification is a key part of computer vision. It's used in many things such as recognizing faces and finding objects to checking medical images on cars without drivers [1]. With the introduction of more complex deep learning tools, image classifying systems have much improved. They can now find strange details and structures in pictures better [2]. One important use of image classification is identifying and sorting pictures in the Intel Image Classification dataset. The Intel Image Classification set is a chosen group of pictures covering different scenes taken by satellites. It is used as a standard set for researchers and users who work on sorting images. This dataset includes six distinct categories: Each category like 'buildings,' 'forest,' 'glacier', and others has a lot of images [3], [4]. This makes it hard for computer programs that sort these images to do well. In this data, the goal of image classification is to create and use models that can correctly put pictures into their own categories. This job is really important for looking at pictures from satellites, watching the environment and putting labels on types of land. Scientists use fancy computer programs to sort out pictures very well. Their work helps improve how we see things and get info from far away using computers [5].

The need to put images in groups, like tasks that use datasets such as the Intel Image Classification set, is important [6], [7]. This happens because we rely more and more on pictures for different things every day. Here are several reasons explaining why there is a need for image classification, often using datasets provided by organizations like Intel: The big amount of pictures and videos created every day, makes it impossible to check them all by hand. Image classification helps machines understand pictures automatically. It lets them sort and organize big sets of data using only common words in English. Identifying objects in pictures is a basic job for computer vision, which comes under AI and ML fields [8]. Training models from image classification tasks can be used in different activities like recognizing objects, finding images and self-driving systems [9]. In cases like the Intel Image Classification set, pictures often come from satellites or gadgets that see far away. It is very important to label these pictures. They help in making maps of land cover, checking the environment and planning cities. Image sorting gives useful details for choosing what to do in areas like farming, tree care and city building. Automatic finding of things and parts in pictures helps people make wise choices using picture information [10], [11]. Video watching and safety devices use picture grouping to find objects, people or strange actions. This app is very important for keeping everyone safe and guarding key buildings. Image being sorted is a big part of modern tech. It affects many different areas too. This lets machines understand and group pictures, making AI improve plus better decision-making in many uses. The Intel Image Classification set, along with others, helps researchers and workers improve image classification methods [12], [13].

In this study, we look at using two popular deep learning designs called MobileNetV2 and EfficientNet. We use them to solve the image classification problem in an Intel data set [14]. These models are known for their ability to use computer power and settings well, making them good for use in real-life situations. In addition, these models are tested with various learning rate values. This gives us information about how much they change when you adjust the settings or hyperparameters. In this study, we want to use common words in English. We aim to correctly classify Intel images

and compare two popular deep-learning models for image classification. The results of this study could guide future work to improve the accuracy and speed of picture-sorting systems in different areas.

II. Literature Review

The main problem in this study is the difference of sizes between layers [15]. This often needs methods like zero-padding or projection, making things harder to understand and costing more computer power. To fix this problem, the writers suggest changes to the starting blocks. They swap deeper parts of ResNet and use a method called symmetric factorization with small 1x1 convolutions to reduce how many parameters there are. This decrease, about 6 million parts, helps to make the time go down a lot. It saves up to 30 seconds in each cycle (epoch). The suggested new action function (NMAF) deals with the problem of vanishing gradient linked to Relu. By turning on bad things and giving out tiny negative numbers instead of zero, NMAF makes how fast something comes together better and more correct. Tests show a big better correctness of 5%, 15% and 5%. For clean sets it is one less, six less for data full sound. The usefulness of RSRNet is shown in different ways using measurements and opinions on three unique sets of information [16]. The study shows that RSRNet is better for classifying images in many types of remote sensing. It beats other top methods. The whole method of RSRNet works on getting better pictures, making the best classifier and mixing different types. So it's a strong tool that can be used in lots of ways to make multisource remote sensing image classification tasks more accurate and reliable. In the end, RSRNet is a big help to this area. It gives us all we need when it comes to problems with classifying images from different sources in remote sensing. This cool design and top performance show that it can be a big part of improving remote sensing tasks and working with images from different sources. This part introduces a new way of trusted computer use [17]. It suggests to separate the process so that we can improve privacy in deep neural network models when they are sent to safe places for running tasks. The main goal behind this method is to get the split computing performance done without having to change deep neural network models. The suggested way of dividing a model into parts not only makes it safer but also reduces the extra work related to this process. This research highlights the need for custom optimization plans for deep learning models in edge computing settings [18]. When we use image cleaning, extra data creation, computer speed boosting and fine-tuning of settings the suggested system gets better results during training. The side-by-side study of InceptionV3, VGG16 and MobileNet shows what each one is good or bad at. This helps to pick the best model for use in edge computing tasks that run on devices like smartphones and tablets. The use of a Model Optimizer improves the trained model and makes it work well on small devices. This paper helps make edge computing better. It gives useful advice for using deep learning models in places with limited resources. MobileNet is the model with lowest accuracy (85%). It takes longest time to load its model - 71 seconds. The VGG16 shows the most trust with an accuracy of 91% and takes least time to load its model (50 seconds). InceptionV3 comes in the middle, reaching an average accuracy of 87% and taking around 52 seconds to load its model. This paper talks about a new way to sort biology pictures by improving the design of CNNs [19]. Using bigger pictures instead of small ones and adding Inverted Residual Block parts, the method beats older ways to classify images. The big check-up on if it works well shows that the new method is powerful. It makes decisions more correctly while saving cost and number of parts needed for computer work. This work helps improve the area of classifying biology pictures. It shows an improved CNN design adjusted for what this field needs. The suggested way could improve picture recognition jobs in many biological uses. This writing introduces a new way to make Support Vector Machines faster for sorting big pictures with lots of colors by using powerful GPUs [20]. The book form of GPU is made to fix the slow nature that SVMs have. This happens mostly with big and hard datasets. The suggested approach makes operations happen at the same time and better use of memory. This leads to faster times for doing things. The tests done on various hyperspectral image examples show that the GPU usage works well. This proves its ability to make SVM-based classifications be finished more quickly for studying hyperspectral images. This job helps in improving the computer speed of machine learning systems for large datasets and high-dimensional data.

III. Materials and Methods

The section contains information on the dataset and method used for Intel image classification.

A. Dataset

The Intel images dataset is curated from Kaggle [21]. The dataset contains 24300 images of neutral scenes containing six different classes namely, buildings, glaciers, forests, sea, mountains and streets. The dataset already contained a higher number of images, hence no augmentation was applied.

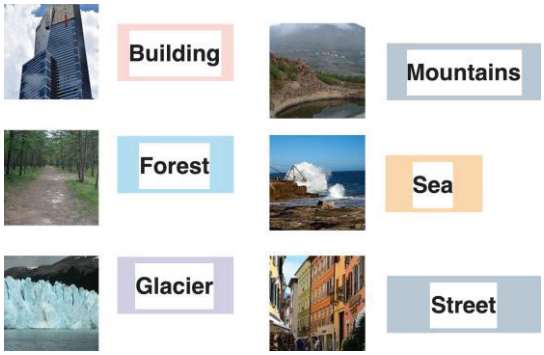

Fig. 1. Intel images in the dataset

B. Methodology

Methodology for Intel Image Classification Using MobileNetV2 and EfficientNet Models: MobileNetV2 is a lightweight, easy-to-use neural network design that uses less computer power. It's known for being efficient in this area.

Get the MobileNetV2 model design from a deep learning tool (for example, TensorFlow or PyTorch) [22]. EfficientNet is a model that uses resources well while still keeping high performance. Bring in the EfficientNet model design and pick a suitable version (like, for example, EfficientNetB0). Use trained weights for MobileNetV2 and EfficientNet on a big group of images (like ImageNet) [23].

Keep the first layers frozen so you don't lose lessons from before. Set up the model with the right loss function (like categorical crossentropy). Use an optimizer like Adam and track results using accuracy. Teach models different learning speeds (example: 0.01 and 0.; to see how it affects their

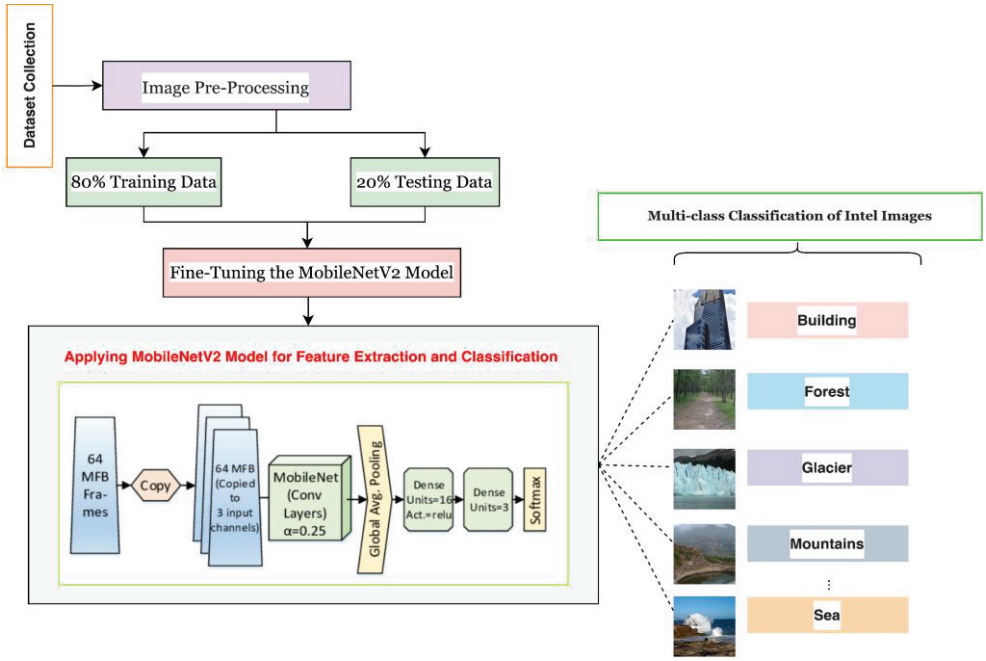

Fig. 2. Proposed MobileNetV2 model for Intel image processing and classification

training needs., Use data boosting methods (like turning or flipping) when training to make the model more tough. Start the training early to stop it when the model's performance on a test set stops improving. Use the training data to teach MobileNetV2 and EfficientNet models in separate steps. Check how well the models work on a new dataset to see if they can perform in different situations. Check correctness, exactness, remembrance and F1-score to fully judge how well a classification is doing. Talk about the things you need to think about when using the chosen model in real-life situations. This clear way shows how to do image sorting on the Intel Image Classification set using MobileNetV2 and EfficientNet models. It gives a clear way to teach models, test them and compare their results. This helps us understand how each model works best.

IV. Results and Discussion

The results have been discussed in this section which compares the MobileNetV2 and EfficientNet models.

A. Results of the MobileNetV2 and EfficientNet models

The results of two different models i.e. MobileNetV2 and EfficientNet have been shown below.

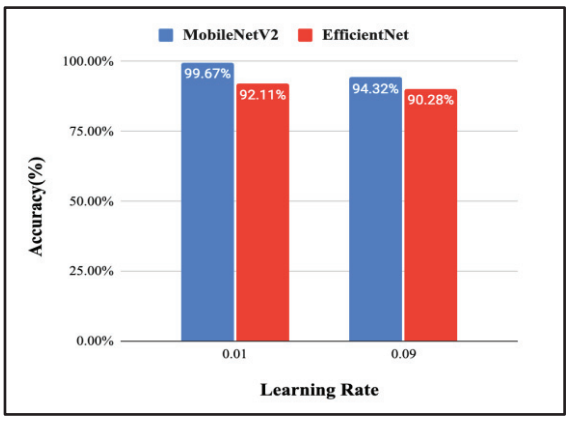

Fig. 3. Accuracy of the MobileNetV2 and EfficientNet model for intel image classification

Fig. 3 shows the accuracy of MobileNetV2 and EfficientNet at different learning rates. The highest accuracy of 99.67% has been achieved by the MobileNetV2 model.

Fig. 4 illustrates the precision of MobileNetV2 and EfficientNet at different learning rates. The highest precision and lowest precision of 0.998 and 0.893 has been achieved by the MobileNetV2 and efficientNet models, respectively.

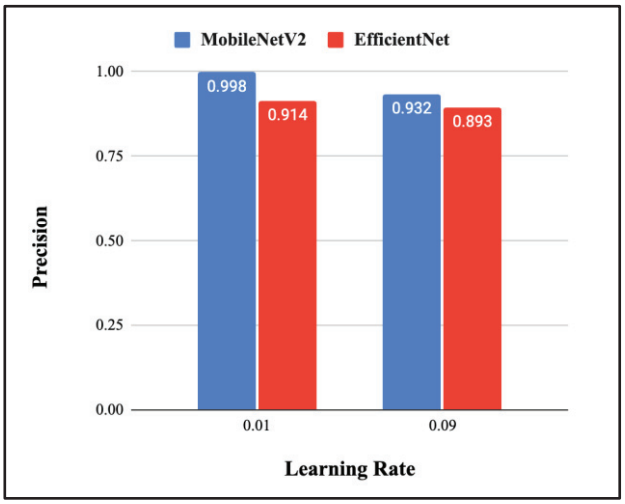

Fig. 4. Precision of the MobileNetV2 and EfficientNet model for intel image classification

Fig. 5 illustrates the recall of MobileNetV2 and EfficientNet at different learning rates. The highest recall and lowest recall of 0.983 and 0.883 has been achieved by the MobileNetV2 and efficientNet models, respectively.

Fig. 6 illustrates the F1-score of MobileNetV2 and EfficientNet at different learning rates. The highest F1-score and lowest F1-score of 0.988 and 0.873 have been achieved by the MobileNetV2 and EfficientNet models, respectively.

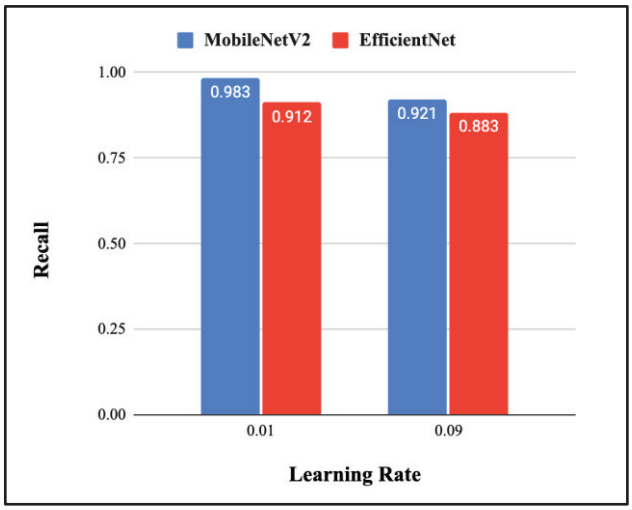

Fig. 5. Recall of the MobileNetV2 and EfficientNet model for intel image classification

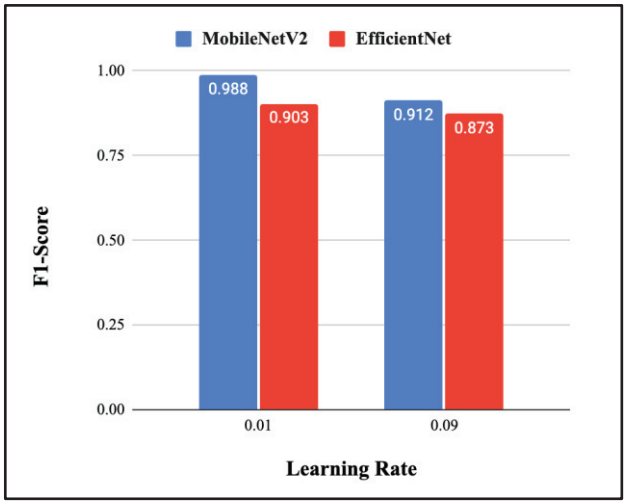

Fig. 6. F1-Score of the MobileNetV2 and EfficientNet model for intel image classification

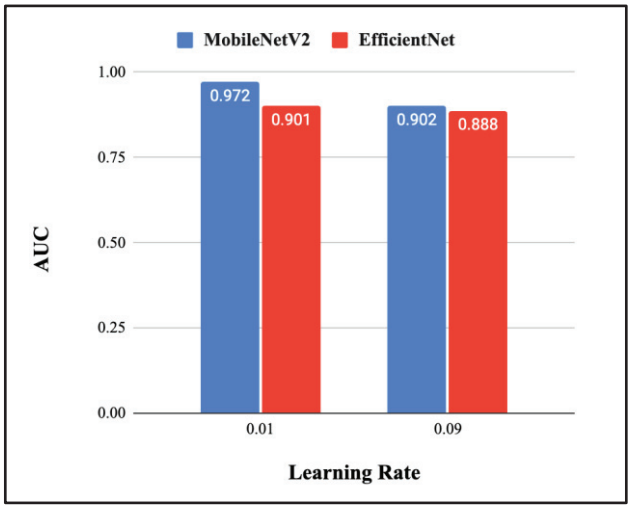

Fig. 7. AUC of the MobileNetV2 and EfficientNet model for intel image classification.

Fig. 7 illustrates the AUC of MobileNetV2 and EfficientNet at different learning rates. The highest AUC and lowest AUC of 0.972 and 0.888 have been achieved by the MobileNetV2 and EfficientNet models, respectively.

B. Comparison of VGG19 with existing techniques

The comparison of VGG19 with existing techniques have been shown in Fig. 8 which identifies that the proposed model MobileNetV2 outperformed by showing 99.67% accuracy.

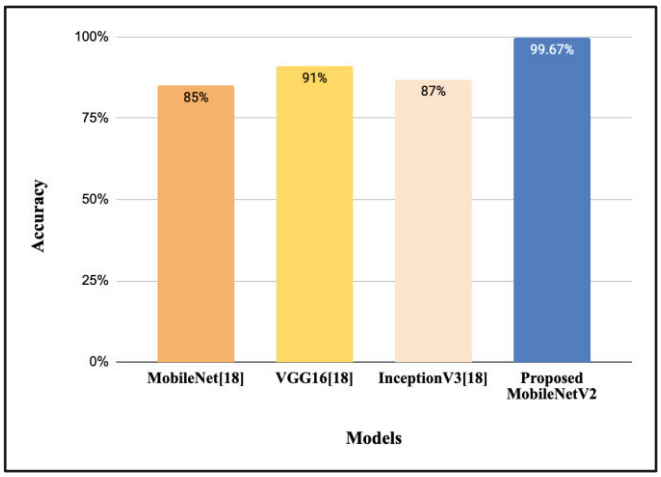

Fig. 8. Comparison of VGG19 with existing techniques

V. CONCLUSION AND FUTURE SCOPE

This study delves into the realm of image classification using the Intel image dataset, employing two prominent deep learning models: MobileNetV2 and EfficientNet. The main goal was to check how well the models worked with different learning rate values. These were 0.01 and 0. The other name for it is "Leaning problem." The findings show how well each model works when it comes to being accurate. This helps people in computer vision see what's best for them. MobileNetV2 was the best performer, reaching a remarkable 99.67% accuracy at round 50. This wonderful result shows how good the model is at correctly sorting Intel pictures. In contrast, EfficientNet showed good results but only reached 92.11% accuracy at the same time point. By comparing these two models, we see how different learning speed values affect their training process and final accuracy results. The better fitting that MobileNetV2 gets means it's good for sorting pictures in the Intel data set. It's important to remember that picking a model relies on many things like computer power, work needs and special data features. The study's results help to continue the talk about choosing a model and adjusting settings in deep learning apps. The shown reliability rates give a goal for later work. They offer the base to explore more and improve in the area of image labelling. As deep learning keeps changing, these ideas will probably help make models for many different image tasks better and more accurate.

REFERENCES

[1] M. Kräter*et al.*, "AIDeveloper: Deep Learning Image Classification in Life Science and Beyond," *Adv. Sci.*, vol. 8, no. 11, p. e2003743, Jun. 2021, doi: 10.1002/advs.202003743.

[2] B. Rueckauer, C. Bybee, R. Goettsche, Y. Singh, J. Mishra, and A. Wild, "NxTF: An API and Compiler for Deep Spiking Neural Networks on Intel Loihi," *J. Emerg. Technol. Comput. Syst.*, vol. 18, no. 3, pp. 1–22, Jan. 2022, doi: 10.1145/3501770.

[3] Y. Tianxu, "Convolutional Neural Network FPGA-accelerator on Intel DE10-Standard FPGA," 2021. Accessed: Jan. 04, 2024. [Online]. Available: https://www.diva-

portal.org/smash/record.jsf?pid=diva2:1584060

[4] M. Bansal, M. Kumar, M. Sachdeva, and A. Mittal, "Transfer learning for image classification using VGG19: Caltech-101 image data set," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 4, pp. 3609–3620, 2023, doi: 10.1007/s12652-021-03488-z.

[5] C. Kyrkou and T. Theocharides, "EmergencyNet: Efficient Aerial Image Classification for Drone-Based Emergency Monitoring Using Atrous Convolutional Feature Fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1687–1699, 2020, doi: 10.1109/JSTARS.2020.2969809.

[6] T. Kaur and T. K. Gandhi, "Deep convolutional neural networks with transfer learning for automated brain image classification," *Mach. Vis. Appl.*, vol. 31, no. 3, p. 20, Mar. 2020, doi: 10.1007/s00138-020-01069-2.

[7] S. Sharma, K. Guleria, S. Kumar, and S. Tiwari, "Benign and Malignant Skin Lesion Detection from Melanoma Skin Cancer Images," in *2023 International Conference for Advancement in Technology (ICONAT)*, Jan. 2023, pp. 1–6. doi: 10.1109/ICONAT57137.2023.10080355.

[8] Z. Wu *et al.*, "Scheduling-Guided Automatic Processing of Massive Hyperspectral Image Classification on Cloud Computing Architectures," *IEEE Trans Cybern*, vol. 51, no. 7, pp. 3588–3601, Jul. 2021, doi: 10.1109/TCYB.2020.3026673.

[9] S. Sharma, A. Kataria, and J. K. Sandhu, "Applications, Tools and Technologies of Robotic Process Automation in Various Industries," in *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, Mar. 2022, pp. 1067–1072. doi: 10.1109/DASA54658.2022.9765027.

[10] D. Xue *et al.*, "An Application of Transfer Learning and Ensemble Learning Techniques for Cervical Histopathology Image Classification," *IEEE Access*, vol. 8, pp. 104603–104618, 2020, doi: 10.1109/ACCESS.2020.2999816.

[11] R. Sharma and V. Kukreja, "Amalgamated convolutional long term network (CLTN) model for Lemon Citrus Canker Disease Multi-classification," in *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, Mar. 2022, pp. 326–329. doi: 10.1109/DASA54658.2022.9765005.

[12] D. Yao *et al.*, "Deep hybrid: Multi-graph neural network collaboration for hyperspectral image classification," *Defence Technology*, vol. 23, pp. 164–176, May 2023, doi: 10.1016/j.dt.2022.02.007.

[13] R. Sharma, V. Kukreja, R. K. Kaushal, A. Bansal, and A. Kaur, "Rice Leaf blight Disease detection using multi-classification deep learning model," in *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Oct. 2022, pp. 1–5. doi: 10.1109/ICRITO56286.2022.9964644.

[14] S. Sharma, A. Kataria, J. K. Sandhu, and K. R. Ramkumar, "Credit Card Fraud Detection using Machine and Deep Learning Techniques," in *2022 3rd International Conference for Emerging Technology (INCET)*, May 2022, pp. 1–7. doi: 10.1109/INCET54531.2022.9824065.

[15] A. A. Yahya, K. Liu, A. Hawbani, Y. Wang, and A. N. Hadi, "A Novel Image Classification Method Based on Residual Network, Inception, and Proposed Activation Function," *Sensors* , vol. 23, no. 6, Mar. 2023, doi: 10.3390/s23062976.

[16] J. Wang, W. Li, Y. Wang, R. Tao, and Q. Du, "Representation-Enhanced Status Replay Network for Multisource Remote-Sensing Image Classification," *IEEE Trans Neural Netw Learn Syst*, vol. PP, Jun. 2023, doi: 10.1109/TNNLS.2023.3286422.

[17] D. M. Kang, H. Faahym, S. Meftah, S. L. Keoh, and M. M. A. Khin, "Practical Deep Neural Network Protection for Unmodified Applications in Intel Software Guard Extension Environments," in *Critical Infrastructure Protection XVII*, Springer Nature Switzerland, 2024, pp. 177–192. doi: 10.1007/978-3-031-49585-4_9.

[18] E. Kristiani, C.-T. Yang, and C.-Y. Huang, "iSEC: An Optimized Deep Learning Model for Image Classification on Edge Computing," *IEEE Access*, vol. 8, pp. 27267–27276, 2020, doi: 10.1109/ACCESS.2020.2971566.

[19] J. Qin, W. Pan, X. Xiang, Y. Tan, and G. Hou, "A biological image classification method based on improved CNN," *Ecol. Inform.*, vol. 58, p. 101093, Jul. 2020, doi: 10.1016/j.ecoinf.2020.101093.

[20] M. E. Paoletti, J. M. Haut, X. Tao, J. P. Miguel, and A. Plaza, "A New GPU Implementation of Support Vector Machines for Fast Hyperspectral Image Classification," *Remote Sensing*, vol. 12, no. 8, p. 1257, Apr. 2020, doi: 10.3390/rs12081257.

[21] P. Bansal, "Intel Image Classification." Jan. 30, 2019. Accessed: Jan. 04, 2024. [Online]. Available: https://www.kaggle.com/datasets/puneet6060/intel-image-classification

[22] S. Sharma and K. Guleria, "A Deep Learning Model for Early Prediction of Pneumonia Using VGG19 and Neural Networks," in *Mobile Radio Communications and 5G Networks*, Springer Nature Singapore, 2023, pp. 597–612. doi: 10.1007/978-981-19-7982-8_50.

[23] S. Sharma and K. Guleria, "A Deep Learning based model for the Detection of Pneumonia from Chest X-Ray Images using VGG-16 and Neural Networks," *Procedia Comput. Sci.*, vol. 218, pp. 357–366, Jan. 2023, doi: 10.1016/j.procs.2023.01.018.