Evaluating Transfer Learning Approaches Using MobileNetV2 Against a Custom CNN

Farima Torabzadeh
Student ID: 24846939
Faculty of Science and Engineering
Manchester Metropolitan University,
Manchester, UK

Abstract—This project investigates the performance of a custom Convolutional Neural Network (CNN) and a pretrained MobileNetV2 architecture with different activation functions (ReLU, Tanh, and Softsign) in for multi-class image classification. The MobileNetV2 with Tanh activation function achieved both the highest test accuracy and the lowest test loss, making it the top-performing model in the project. The results from evaluation metrics including loss curves, confusion matrices, classification reports and ROC curves, confirm the superiority of transfer learning and highlight the role of activation functions in model performance. These findings reinforce the importance of model selection and architectural choices in achieving optimal results in image classification tasks.

Keywords – Transfer learning, MobileNetV2, Convolutional Neural Network (CNN), Intel Image Classification, Deep learning

I. INTRODUCTION

Image classification plays a central role in computer vision and is widely used in face recognition, object detection and medical image analysis applications [2]. By advancing deep learning techniques, modern classification systems have significantly improved, allowing them to detect subtle patterns and complex structures in images more effectively [2]. In this project, compare the performance of a custom-built CNN with transfer learning based on MobileNetV2 using different activation functions.

Transfer learning uses data, deep learning recipes, and models developed for one task, and reapplies them to a different, but similar, task. It's a machine learning method where a model trained for one task is used as a starting point for another, enabling better performance and faster results [5]. In this work, transfer learning is used to fine-tune MobileNetV2 on the dataset. MobileNetV2 is a convolutional neural network architecture specifically designed for mobile and resource-constrained environments [4]. It introduces a novel building block called the inverted residual with linear bottleneck, which enables efficient representation learning by expanding compressed inputs into high-dimensional spaces, filtering them with lightweight depth-wise convolutions, and then projecting them back to a low-dimensional space using linear operations [4]. Transfer learning is applied to MobileNetV2 to assess its adaptability and effectiveness in this classification project.

The objective is to evaluate how architectural efficiency in a multi-class image task. This comparison helps us to identify whether a lightweight pretrained model with the help of transfer learning, can outperform a custom model trained from scratch and how activation functions yield the best results in terms of learning dynamics and test performance.

II. DATASET

A. Dataset Description

The Intel images dataset is curated from Kaggle as the foundation of the research. This dataset contains 24300 images of neutral scenes, which includes six different classes: building, glaciers, forests, sea, mountains and streets. To maintain consistency and equity in scoring, the datasets were divided into training (60%), validation (20%), and testing (60%) sets.

B. Data Augmentation

Image augmentation techniques were applied to the training data to improve the robustness and generalisation of the models to unseen land use scenes. These augmentations increase the diversity of the dataset by creating modified versions of the original images. This is particularly important in real-world environments where the appearance of a class (e.g. mountain or glacier) can vary due to orientation, lighting or slight distortions.

The transformation used includes random horizontal flips, rotations, and colour jittering to simulate changes in brightness, contrast, and saturation. All images are also resized to a fixed dimension and normalised to ensure stable training.

(A) (B)

Fig. 1. Intel images in the dataset before (A) and after Data Transformation(B)s

III. Activation Functions

The activation function role is very crucial in the neural network field. The following properties should be held by activation functions.
1. Non-linearity is important in the converging process of training of the network. It enables the models to learn complex relationships from the data.
2. It should not increase the model's computational complexity
3. Gradients should stay stable during training
4. Data should be well-distributed to support effective training [1]
Below are the following activation functions which we are going to use in this article.

A. Tanh

The Tanh (hyperbolic tangent) activation function is a commonly used non-linear activation function in neural networks, shown in Figure. Here are some of its metrics:

- Range
- Non-linear
- Smoothness
- Zero-centered

The Tanh compresses inputs to the range [-1,1], similar to the Logistic Sigmoid. However, it shares some of the drawbacks, such as diminishing gradient and computational cost.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{1}$$

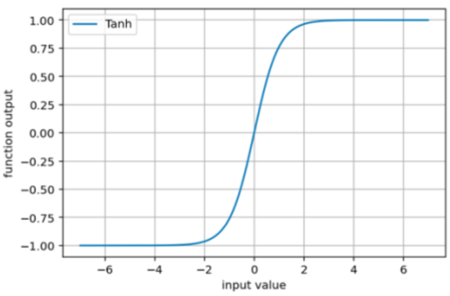

Fig. 2. Tanh Function[1]

B. Softsigns

The Softsign activation function is a smooth and continuous function that is commonly used in neural networks.

$$f(x) = x/(1 + abs(x)) \tag{2}$$

where abs(x) is the absolute value of x.

The following are the Softsign function properties:

- It maps any real value to a value between -1 and 1, helping the normalisation process of the input to the next layer.
- Being a smooth function with a continuous first derivative, it facilitates the optimisation process through gradient-based methods.
- Its outputs are centred around zero, meaning the output has a mean of zero.

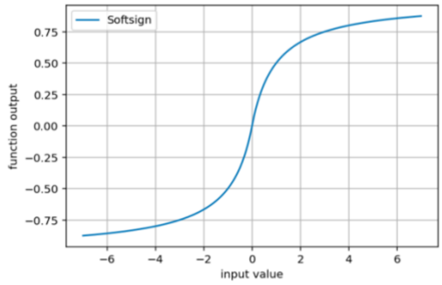

Fig. 3. Softsign Function[1]

While Softsign is less susceptible to the vanishing gradient problem than functions like Sigmoid or Tanh, it may still experience the gradient problem for large input values. Although it's not as commonly used as some other activations, Softsign can be effective in cases where normalisation or smoothness is important.

C. ReLU(Rectified linear unit)

The rectified linear unit is a frequently used activation function in the deep neural networks. It introduces non-linearity into the model, enabling it to learn deep, non-linear correlations between the inputs and outputs.

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases} \tag{3}$$

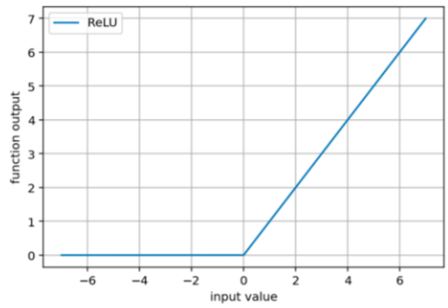

Fig. 4. ReLU Function[1]

ReLU helps address the vanishing gradient problem by maintaining a unit slope for positive inputs. It acts as a threshold function, outputting the input for positive values and zero for negative ones.

IV. METHODOLOGY

A. Proposed Methodology

The research objective is to assess the Convolutional Neural Networks compared to the MobileNetV2 with various activation functions. In this study, experiments are concducted on a standard image classification dataset and the effects of using Tanh, Softsign, and ReLU activation functions is investigated by measuring the accuracy and the training time of the MobileNetV2 model.

The results compared and analysed to provide insights into the effects of activation functions on the deep learning models performance.

B. Model Architecture

Based on the proposed methodology, three model architectures have been used in this study:

- **Custom CNN**

  The structure of the developed custom CNN consists of five convolutional blocks, each designed to progressively learn higher-level spatial features from the input images. These blocks include 5 convolutional blocks for hierarchical feature extraction, each combining convolutional layers, ReLU activations, batch normalisation, max pooling, and dropout to enhance stability and prevent overfitting.

  The model was trained over 15 epochs using the Adam optimiser and cross-entropy loss.

  The custom CNN model showed steady and significant improvement, achieving a test accuracy of 73.79%. Loss curves indicate consistent convergence without overfitting. The model performed particularly well on the "forest" class, although it struggled more with "street" and "sea" classes. The ROC AUC scores confirm the model's discriminative ability.

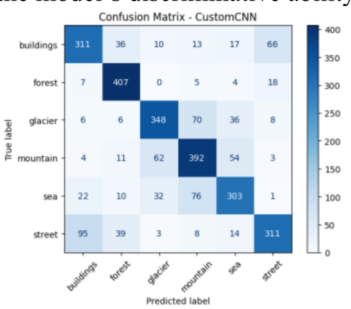

Fig. 5. Custom CNN confusion matrix

	precision	recall	f1-score	support
buildings	0.70	0.69	0.69	453.00
forest	0.80	0.92	0.86	441.00
glacier	0.76	0.73	0.75	474.00
mountain	0.70	0.75	0.72	526.00
sea	0.71	0.68	0.69	444.00
street	0.76	0.66	0.71	470.00
accuracy	0.74	0.74	0.74	0.74

Fig.6. CNN classification report

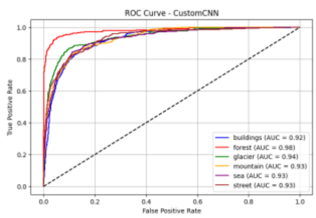

Fig7. CNN ROC plot

- **MobileNetV2**

  MobileNetV2 is employed as a base model using transfer learning. MobileNetV2 is a lightweight and efficient convolutional network architecture developed by Google, optimised for mobile and embedded vision applications.

  In this implementation, the pre-trained MobileNetV2 was loaded which weighted from ImageNet, and allows the model to reuse the learned visual features such as edges, textures, and shapes. All the convolutional layers were frozen in the feature extractor and replaced the classifier with a new fully connected block. This custom classifier consists of a dropout layer for regularisation, projecting to 128 features, various activation functions (ReLU, Tanh, and Softsign), and a final linear layer to produce predictions for the target classes.

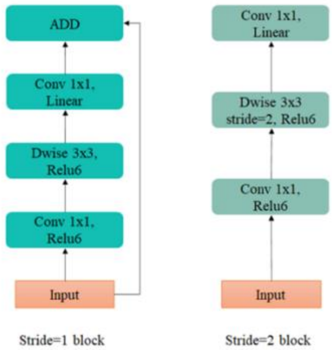

Fig. 8. Original MobileNetV2[6]

To ensure a fair comparison with our custom CNN model, the training configuration was kept consistent across models. Therefore, each of the modes was trained for 15 epochs using the Adam optimiser and cross-entropy loss, with the same learning rate of 0.0001 and batch size of 32.

- ReLU Activation Function

  MobileNetV2 with ReLU consistently delivered strong results, achieving a final test accuracy of 89.97% with minimal fluctuation across epochs. The loss curves indicate fast convergence and stable performance, with no signs of overfitting. The classification report confirms well-balanced learning across all classes, and the confusion matrix reflects accurate predictions. The ROC plot also shows excellent separability, with all values between 0.97 and 1.00.

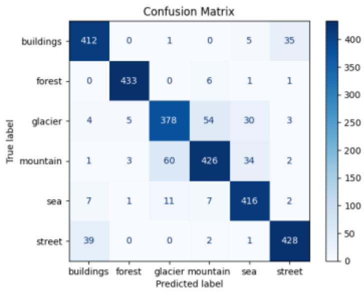

Fig. 9. MobileNetV2 with ReLU activation confusion matrix

	precision	recall	f1-score	support
buildings	0.89	0.91	0.90	453.00
forest	0.98	0.98	0.98	441.00
glacier	0.84	0.80	0.82	474.00
mountain	0.86	0.81	0.83	526.00
sea	0.85	0.94	0.89	444.00
street	0.91	0.91	0.91	470.00

Fig.10. MobileNetV2 with ReLU activation classification report

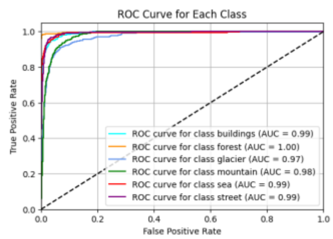

Fig.11. MobileNetV2 with ReLU ROC plot

- Tanh Activation Function
  MobileNetV2 with Tanh, maintained highly consistent performance, ending with a test accuracy of 90.06% and validation accuracy above 90% for the final epochs. The model showed smooth and efficient convergence with no signs of overfitting, as seen in the loss curves. The classification report highlights balance across all classes and particularly strong results for "forest" class.

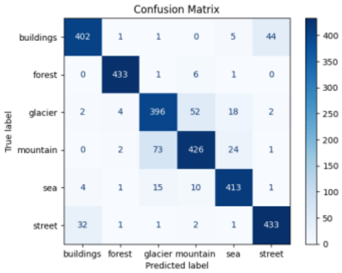

Fig. 12. MobileNetV2 with Tanh confusion matrix

	precision	recall	f1-score	support
buildings	0.91	0.89	0.90	453.00
forest	0.98	0.98	0.98	441.00
glacier	0.81	0.84	0.82	474.00
mountain	0.86	0.81	0.83	526.00
sea	0.89	0.93	0.91	444.00
street	0.90	0.92	0.91	470.00

Fig.13. MobileNetV2 with Tanh classification report

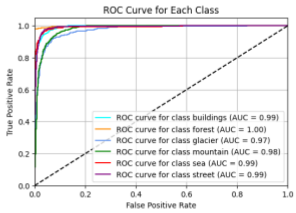

Fig.14. MobileNetV2 with Tanh ROC plot

- Softsign Activation Function
  MobileNetV2 model with Softsign activation achieved the highest test accuracy of 89.57% with stable validation accuracy around 90% and consistently decreasing loss curves throughout training. The model performed robustly across all classes and standout performance in "forest". The confusion matrix shows minimal misclassification, especially in dominant classes. ROC curves for all classes reflect excellent separability, with AUC values >= 0.97.

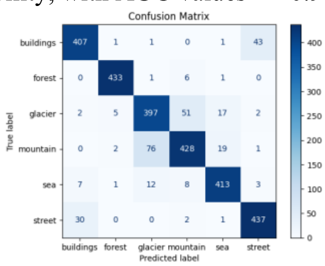

Fig. 15.. MobileNetV2 with Softsign confusion matrix

	precision	recall	f1-score	support
buildings	0.91	0.90	0.91	453.0
forest	0.98	0.98	0.98	441.0
glacier	0.82	0.84	0.83	474.0
mountain	0.86	0.81	0.84	526.0
sea	0.91	0.93	0.92	444.0
street	0.90	0.93	0.91	470.0

Fig.16. MobileNetV2 with Softsign classification report

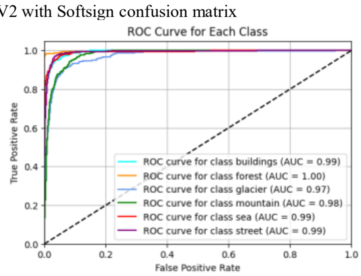

Fig.17. MobileNetV2 with Softsign ROC plot

All three MobileNetV2 models using ReLU, Tanh, and Softsign achieved strong performance with test accuracies around 89-90%. Among them, Tanh achieved the highest test accuracy (90.06%) and Softsign delivered the best F1-scores and class balance. Overall, Tanh and Softsign slightly outperformed ReLU, making them more effective choices for this classification problem.

V. Experiments and results

A. Device Specification

All experiments were conducted on a Kaggle Notebook environment with Python 3.11.11, running on a virtualised Intel(R) Xeon(R) CPU with 4 cores and 31 GB RAM. A Tesla T4 GPU with 15 GB VRAM was used weights were loaded manually from disk.

B. Performance Evaluation and Comparison

All models were trained using the Adam optimiser with a learning rate of 0.0001 and cross-entropy loss on a consistent batch size of 32.

The Custom CNN shows a gradual decrease in both training and validation loss over the 15 epochs, indicating stable learning and improved generalisation. However, the overall loss values still remain relatively high throughout training.

Parameter/Model	Custom CNN	MobileNetV2 (A)	MobileNetV2 (B)	MobileNetV2 (C)
Layers	10 Conv + 2 FC	53 from pretrained MobileNetV2 + 1 transfer learning classifier	53 from pretrained MobileNetV2 + 1 transfer learning classifier	53 from pretrained MobileNetV2 + 1 transfer learning classifier
Batch Size	32	32	32	32
Pretrained	No	Yes	Yes	Yes
Optimiser	Adam	Adam	Adam	Adam
Learning Rate	0.0001	0.0001	0.0001	0.0001
Loss Function	CrossEntropyLoss	CrossEntropyLoss	CrossEntropyLoss	CrossEntropyLoss
Output Classes	6	6	6	6
Activation Function	ReLU	ReLU	Tanh	Softsign
Training Time (s)	599.54 s	507.97 s	521.05 s	508.87 s

Table 1. Hyparameters used for model training

In contrast, all MobileNetV2 models exhibit a sharp decline in loss within the first few epochs, thanks to the pretrained backbone. Their final training and validation losses settle much lower, demonstrating faster convergence and better overall fit.

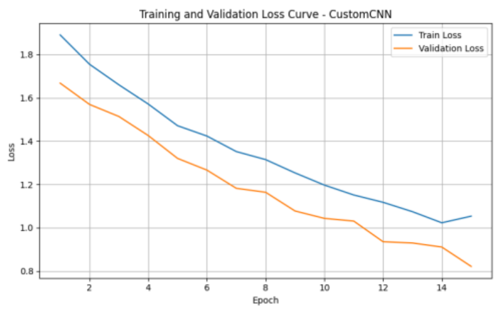

Fig. 18. Custom CNN loss curve

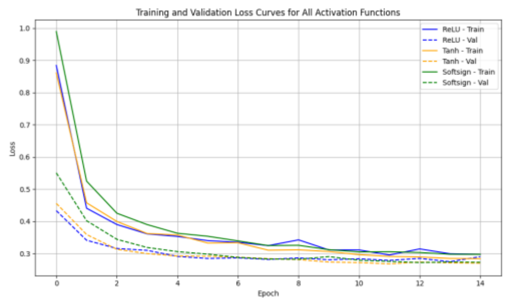

Fig. 19. MobileNetV2 loss curve across different activation functions

V. CONCLUSION

This project explored the performance of convolutional neural networks and transfer learning using MobileNetV2 with different activation functions for a multi-class image classification problem. The custom CNN model trained from scratch showed steady improvement, achieving a final test accuracy of 73.79%, demonstrating solid learning ability despite its simpler structure. In contrast, all MobileNetV2 variants achieved significantly higher performance, with test accuracies close or above 89%. Among them, the MobileNetV2 model with Tanh activation function showed the highest test accuracy while the MobileNetV2 model with Softsign activation function showed the best class balance. These results highlight the effectiveness of transfer learning and suggest that selecting the right activation function can lead to enhancing the model performance.

REFERENCCES

[1] G. K. Pandey and S. Srivastava, "ResNet-18 comparative analysis of various activation functions for image classification," in Proc. 2023 Int. Conf. Inventive Comput. Technol. (ICICT), Lalitpur, Nepal, 2023, pp. 595–601, doi: 10.1109/ICICT57646.2023.10134464.

[2] A. Kaur, V. Kukreja, N. Thapliyal, M. Aeri, R. Sharma, and S. Hariharan, "Fine-tuned EfficientNet and MobileNetV2 Models for Intel Images Classification," in Proc. 2024 3rd Int. Conf. Innov. Technol. (INOCON), Bangalore, India, 2024, pp. 1–5, doi: 10.1109/INOCON60754.2024.10512279.

[3] K. Agasti, K. G., M. Thalapilly, P. M., and V. Panicker J., "Experimentally Enhancing ResNet50 Performance on the Intel Dataset Through Architectural Modifications," in Proc. 6th Int. Conf. Smart Syst. Inventive Technol. (ICSSIT), Kalpa Publications in Computing, vol. 19, pp. 445–454, 2024, doi: 10.29007/ls2m.

[4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4510–4520, doi: 10.48550/arXiv.1801.04381.

[5] B. Zia, R. Illikkal, and B. Rogers, "Use Transfer Learning for Efficient Deep Learning Training on Intel® Xeon® Processors," Intel White Paper, 2018.

[6] K. Dong, C. Zhou, Y. Ruan and Y. Li, "MobileNetV2 Model for Image Classification," in Proc. 2020 2nd Int. Conf. Inf. Technol. Comput. Appl. (ITCA), Guangzhou, China, 2020, pp. 476–480, doi: 10.1109/ITCA52113.2020.00106.