Title: Natural Scene Classification using CNN and Transfer learning

Introduction

In the rapidly evolving field of artificial intelligence (AI), one of the most significant advancements has been the ability of machines to understand and interpret visual data . Image classification has emerged as a strong research area in computer vision over recent years, forming the foundation for different visual recognition applications (Chen et al., 2021).

Scene recognition is essential for machines to comprehend their environment, enabling them to make informed decisions and take appropriate actions. This technology holds great promise in various domains, such as aiding visually impaired individuals, enhancing surveillance systems, and augmenting virtual and reality experiences (Shabbir et al., 2021). In scene recognition, objects are identified based on their arrangement within the image and the context of the background, as opposed to object classification, which primarily focuses on categorizing prominent objects in the foreground (Soudy et al., 2022). Natural scene classification has remained an active area of research for several years and it is very important because it has a diverse range of applications such as robot navigation, disaster detection, content-based image retrieval and intelligent surveillance (Zeng et al., 2021).

In this research, the intel dataset used comprises of images categorized into six classes: Building, Sea, Forest, Glacier, Street, and Mountain. The training set consists of 14,000 images, while the test set contains 3,000 images. A customized CNN model, alongside three pre-trained models (VGG16, ResNet50, and Inception V3), will be used to classify the image categories. Following training using these various models, the metrics of accuracy, recall, and precision will be utilized to evaluate and compare the performance of each model. The goal of the research is to classify scene images to one of the predefined scene categories (Building, Sea, Forest, Glacier, Street, and Mountain) so as to contribute to the development of scene recognition models thereby advancing the capabilities of Artificial Intelligence system to be able to understand outdoor visual content. The expected outcome of this research is to identify the most suitable CNN model architecture for scene recognition tasks.

Aim

To design and implement a customized CNN model and evaluate its performance against pretrained architectures such as VGG16, ResNet50 and InceptionV3 using the Intel Image Scene dataset.

Objectives

- To develop a custom-designed CNN architecture with better accuracy for natural scene using the intel image scene dataset
- To compare the performance of the implemented custom-designed CNN architecture with pretrained models, such as VGG16, ResNet50 and InceptionV3.

Research Question

How does the performance of the custom-designed CNN architecture compared to that of the pretrained models (VGG16, ResNet50 and InceptionV3) using the Intel Image Scene dataset?

Related Work

Image analysis and classification has witnessed a lot of advancements in recent years, largely driven by the emergence of deep learning techniques, particularly convolutional neural networks (CNNs). Convolutional Neural networks have showed remarkable success in its various applications which include remote sensing, scene classification, and image recognition. Scene classification involves the classification of various scenes depicted in images and achieving high accuracy in this task often involves the utilization of objects or visual descriptors. Transfer learning, a method where a pre-trained model is adapted to a new task, has been a cornerstone in achieving high accuracy and efficiency in scene classification tasks, as it leverages uses learned from large datasets.

Several studies have delved into the realm of image classification using transfer learning based CNNs. Baral & Aryal (2023) present an experimental overview of transfer learning approaches in remote sensing image classification, showcasing the potential benefits of pre-trained models in this domain. Their research aims to enhance the accuracy and efficiency of image classification in the context of remote sensing by leveraging the knowledge acquired from pre-trained models through transfer learning techniques. Similarly, Pires De Lima & Marfurt (2019), contributes valuable insights into the use of transfer learning-based CNNs for remote sensing scene classification. Its focus on the effectiveness of transfer learning and the potential for performance improvement aligns well with the objectives of this project, by providing a relevant context for the investigation and comparison of different image analysis architectures. The need for customized convolutional neural network architectures has also been addressed. Soudy et al. (2022) propose a novel architecture called RepConv which uses repetitive

convolutional blocks as a key component for image scene classification, showcasing the potential for innovation in CNN design. The RepConv model which contains five convolutional layers and one fully connected layer, was compared with the ResNet architecture using the Intel dataset and stochastic gradient descent as the optimizer. This methodology shares similarities with the approach in this research, as both uses the same dataset and ResNet-50 as a pretrained model. However, there is difference in the custom CNN architecture employed and this research uses more pretrained models like VGG16 and Inception v3 and Adam as the optimizer which is a popular optimization algorithm for training neural networks.

Surendran et al. (2022) focus on the application of transfer learning techniques for natural scene classification, particularly with the goal of improving its understanding by the intelligent machines. The authors explore how pre-trained models and knowledge from one domain can be used to enhance the performance of machines in comprehending and classifying natural scenes. The paper uses two variants of the ResNet model (ResNet-50 and ResNet-101) on a dataset of natural scenes and obtain a notable accuracy level.

Ali et al. (2021) emphasize the importance of evaluating novel approaches against existing techniques. Their research proposes a CNN architecture comprising 5 convolutional layers, 3 activation layers, 3 max-pooling layers, and 2 fully connected layers, employing Adam as the optimizer. The model was used for concrete crack detection analysis. The model was compared with pre-trained model such as ResNet50, VGG 16 and Inception V3. Our research aligns with theirs in methodology, though the dataset that was used for the analysis is different.

All these related works highlight the growing significance of transfer learning and custom-designed CNN architectures in image analysis and scene classification and this research involves comparing the performance of the custom-designed CNN architecture with well-established pretrained models, including VGG16, ResNet50, and InceptionV3. Finally, this work will contribute to the existing body of research by offering insights into the effectiveness of custom designed CNN architecture and its performance in relation to established transfer learning models on the intel image scene dataset.

Methodology

Dataset

The dataset consists of around 25,000 natural scenes images. The images are categorized into groups such as mountain, forests, glaciers, buildings, street, and sea. The dataset is divided into

three sets: training data (14,000 images), testing data (3,000 images), and prediction (7,000 images). The data was initially published on https://datahack.analyticsvidhya.com for the image classification challenge hosted by Intel. This dataset was gotten from Kaggle (https://www.kaggle.com/datasets/puneet6060/intel-image-classification)

Models

The performance of four models (customized CNN, VGG16, ResNet50 and Inceptionv3) will be evaluated on the intel image dataset and each model are explained in detail below.

VGG16

VGG16 is a convolutional neural network (CNN) introduced by Simonyan & Zisserman (2014). It's well-known for its deep architecture with 16 weight layers. VGG16 is an object detection and classification algorithm that can classify 1000 classes with 92.7% accuracy, and it is one of the top models from the ILSVRC-2014 competition. VGG16 has thirteen convolutional layers, five layers of Max Pooling, and three Dense layers resulting to 21 layers, However, the model possesses only sixteen weight layers, representing the learnable parameters layer. The convolutional layer has 3x3 filter with stride 1 using the same padding and max-pool layer of 2x2 filter of stride 2. The architecture has three fully connected layers which have 4096 channels each and the third layer contains 1000 channels, one for every class. This model achieved a great performance on ImageNet dataset which is the benchmark for image classification task. The model is simple to implement yet it is powerful, and it has been used in variety for variety of image classification.

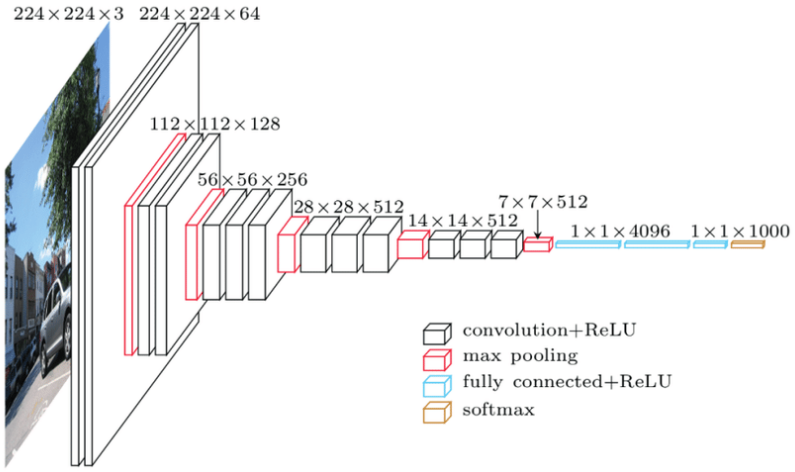

Figure 1 - VGG 16 Model Architecture (*VGGNet-architecture-19.ppm (850×498)*, no date)

ResNet50

This is a CNN architecture with 50 layers. It was introduced by He et al. (2015) at Microsoft Research. it is designed to address the problem of vanishing gradient often encountered in very Convolutional neural networks.

ResNet-50 is built on the concept of residual learning, which involves using skip connections to bypass one or more layers. These skip connections retain information from preceding layers, which aids the network to learn better representations of the input data.

The architecture of ResNet-50 is divided into blocks, in which each blocks contains a set of residual blocks. These residual blocks consist of multiple convolutional layers and skip connections. By using residual blocks, the model can learn more meaningful and efficient representations of input data. One of the major accomplishments of ResNet-50 was its remarkable performance on the ImageNet dataset, achieving a very low error rate of 3.57%.

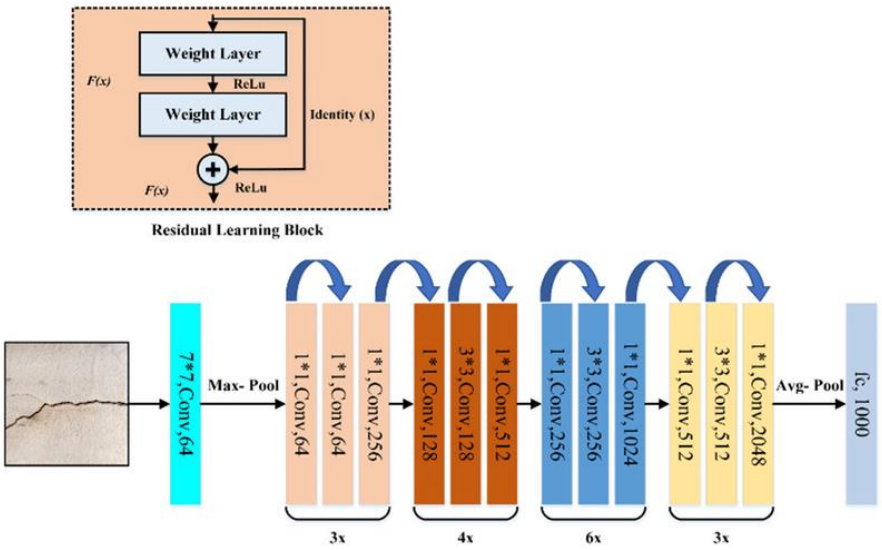

Figure 2 - ResNet 50 Model Architecture (Ali et al., 2021)

Inception V3

This is a pre-trained model with 48 deep layers, and it can classify images into 1000 categories. This model was based on the paper: "Rethinking the Inception Architecture for Computer Vision" by Szegedy et al. (2016). The model demonstrates an ability to achieve an accuracy

surpassing 78.1% on the ImageNet dataset. The model consist of factorized convolutions to enhance computational efficiency, smaller convolutions for faster training, asymmetric convolutions, auxiliary classifiers which act as a regularizers, and a grid search reduction technique. The loss is computed using the Softmax function. Inception V3 is distinguished by its computational efficiency, reducing the number of network parameters generated and minimizing economic costs.

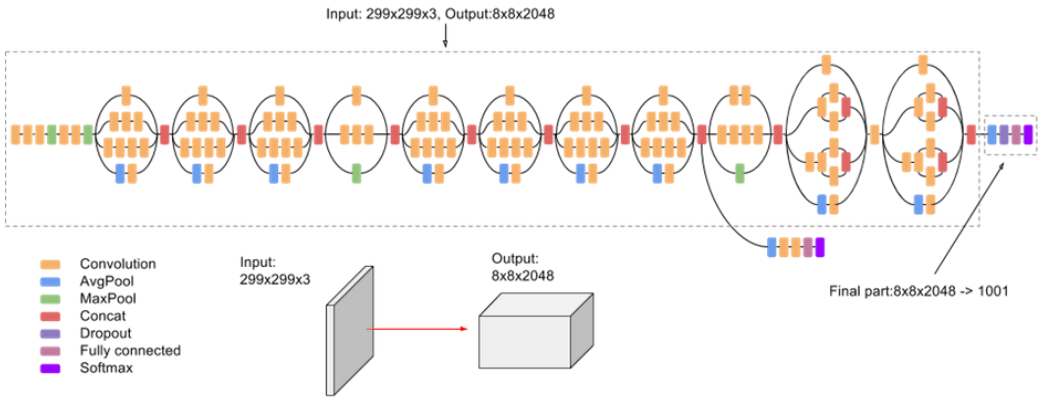

Figure 3 - Inception V3 Architecture (*Inception V3 Model Architecture*, 2021)

Customized CNN Model Architecture

The customized model architecture was constructed from scratch using several hyper parameters tuning like number of convolutional layers, types of optimizers and learning rates as seen in Table 1.

Table 1 - Different customized CNN models

CNN Model	Optimizer	Number of Convolutional layers	Accuracy	Loss	No of Epoch before early stopping
Model 1	SGD	4	0.76	0.6648	21
Model 2	RMSprop	4	0.79	0.6548	8
Model 3	Adam (0.001)	4	0.83	0.4483	19
Model 4	Adam (0.0001)	4	0.82	0.5044	20
Model 5	Adam (0.0001)	5	0.71	0.8310	28

The best model architecture achieved with Adam as the optimizer and learning rate of 0.001 with the accuracy of 83% consists of an input layer which takes the image as input, three hidden layers which are the main building blocks that carries the main portion of the network computational load which is used to extract relevant features. Rectified linear unit (ReLU) is used as the activation function in the hidden layers and this help the convolutional neural network to capture non-linearities. The ReLU returns zero if the input is negative and returns same value if the input is positive. Each convolutional layer is followed by a max-pooling layer which helps to reduce the dimensionality of the representation and the complexity of the model (O'Shea & Nash, 2015). The padding in the convolutional layers is set to 'same' so that the output can be the same shape as the input which helps to preserve the border information of the input data. Following the convolutional layer, flattening occurs before integrating two fully connected layers, which helps to map the representation between the input and the output. Softmax function is used as the last activation function, and this can handle multiple classes and calculate the probability distribution of the events.

Experiment

The experiments were conducted using Jupiter notebook on the DAIM lab system which is Dell desktop system with a Dell core i9-12900 CPU running at 2.40 GHz, 32 GB RAM and a NVIDIA GeForce RTX 3090 GPU. When working with CNN models, it requires alot of data and to address this, we use the Keras ImageDataGenerator to implement data augmentation which expand the size of our dataset. The images were rescaled to 150x150 pixel. Throughout the augmentation process, data normalization was performed, bringing values within the range of 0 to 1. The models were compiled using sparse categorical cross-entropy as the loss function which is the difference between what the model is predicting and the target. Adam optimizer is used as the optimizer the evaluation metrics are Accuracy, Precision and Recall. The training procedure encompassed 30 epochs with a batch size set at 32. The Callbacks were used in the training process, encompassing several components such as model checkpoint which was used to save the model weight during training whenever the monitored quantity is optimum when compared to the last epoch, early stopping which is an optimization technique which is used to reduce overfitting and Learning rate scheduler which adjust learning rate during training. The customized CNN models and baseline models (VGG16, ResNet50 and Inception V3) will trained and compared.

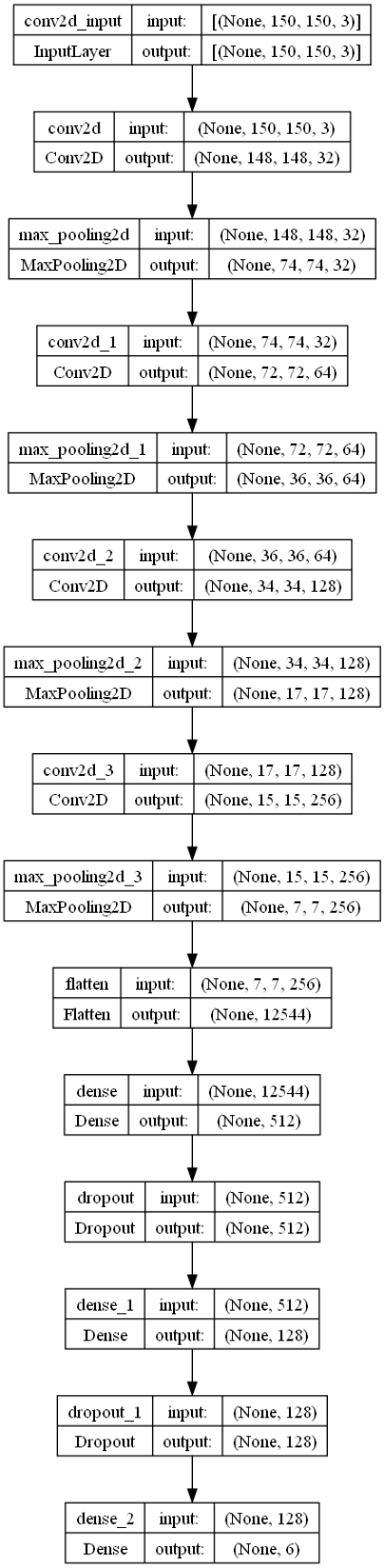

Figure 4 - Customized CNN Model architecture

Evaluation metrics

In this research, the evaluation metrics used include accuracy, precision, and recall. Accuracy represents the percentage of accurate predictions made for the test dataset. It is calculated as the total of True positive and True Negative divided by the overall number of samples or prediction.

$$Accuracy = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{Overall\ number\ of\ predictions}$$

Where overall number of predictions = TP +TN+FP+FN

True Positive (TP) occurs when an observation is predicted to belong to a specific category, and it indeed belongs to that category.

True Negative (TN) is when an observation is predicted not to belong to a certain category, and it does not actually belong to that category.

False Positive (FP) takes place when an observation is predicted to belong to a category, but the observation does not truly belong to that category; this is referred to as a Type I error.

False Negative (FN) arises when an observation is predicted not to belong to a category, yet it genuinely belongs to the category; this is Known as Type II error.

Precision is the ratio of the accurately positive outcomes to the total number of positive outcomes predicted by the model. Out of the positive prediction how many did the model get correctly.

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)}$$

Recall is the ratio of the accurately positive outcomes to the total number of outcomes that should be identified as positive. Recall is out of the positive class; how many did the model get correctly. It is also known as sensitivity of True positive rate.

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)}$$

Results and Discussions

Table 2 - Results of the customized CNN model and the pretrained models

CNN Model	Number of Convolutional layers	Accuracy	Precision	Recall	Loss
VGG16	16	0.85	0.86	0.86	0.3823
ResNet50	50	0.60	0.60	0.60	1.0181
Inception V3	48	0.89	0.90	0.90	0.2745
Customized CNN Model	4	0.83	0.85	0.84	0.4483

From the result in Table 2, it is evident that the customized CNN model achieves a competitive accuracy (83%), precision (85%) and recall (84%) when compared to the pretrained models which indicates that the customized model can learn relevant features from the dataset resulting to a balanced performance on the evaluation metrics. The inception V3 achieved the highest accuracy of 89% and shows a better precision (90%), recall (90%) and lower loss (0.2745) which implies that the model is efficient in minimizing loss during training when compared to the other models. However, the ResNet-50 exhibit the lowest performance in terms of accuracy (60%) and other evaluation metrics which indicates that ResNet-50 may not be a better choice of model for this specific dataset possibly due to its model architecture. The VGG-16 also shows a better performance with the accuracy, precision and recall of 85%, 86% and 85% respectively.

Comparing the results, the objectives of this research were partially met as the custom-designed CNN architecture demonstrated competitive performance but didn't surpass the pretrained models except for ResNet-50. The choice of pretrained models like Inception V3 and VGG-16, which are specifically designed for image classification tasks, poses a strong challenge to the custom model.

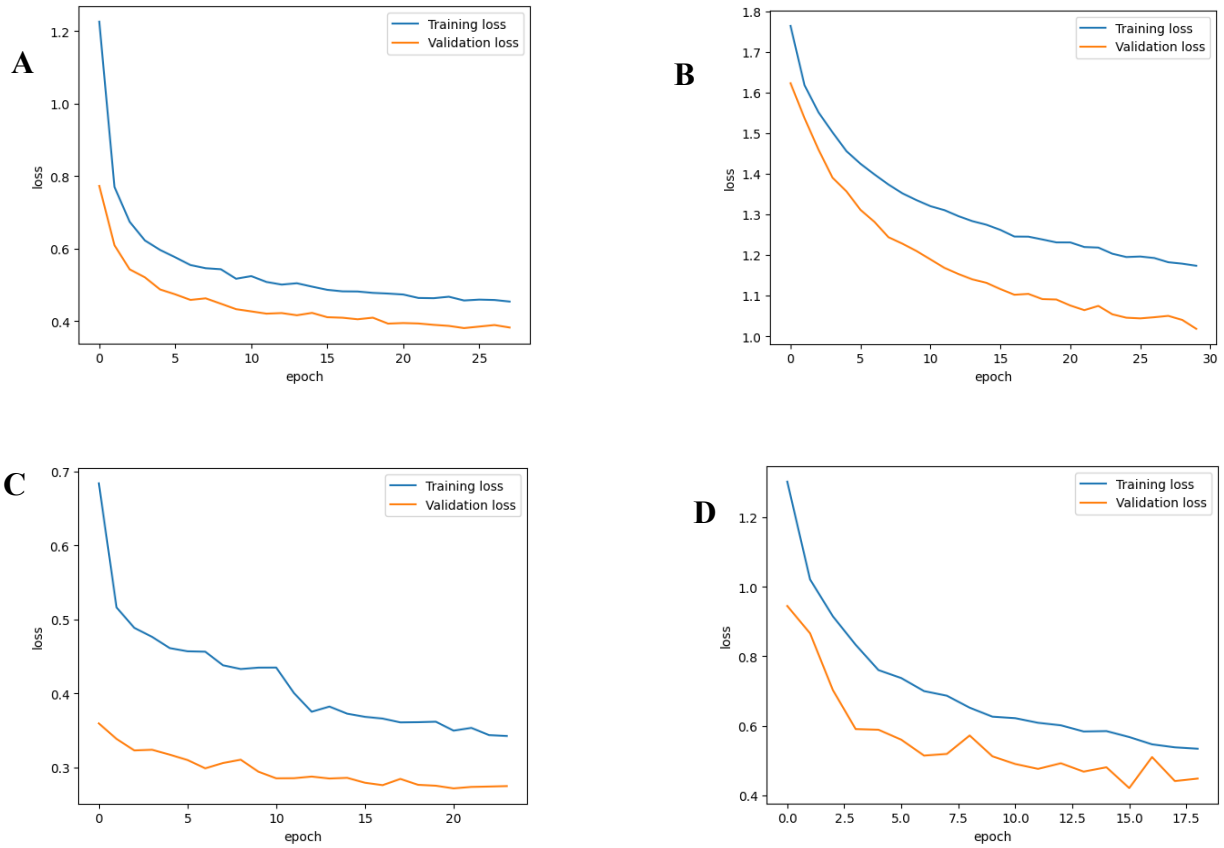

Figure 5: Training and validation loss of the models A: Loss on VGG-16 B Loss on ResNet-50 C: Loss on Inception V3 D: Loss on Customized CNN Model.

From Figure 5, we can see that the model performances converge well and learn pattern from the training data. The plot indicates that the models generalizes well because the training loss is higher than the validation loss and there is no overfitting of the model at any point.

Comparing the results of the model performance from this research with that of Soudy et al., (2022) can give a valuable insight into the preformance and contribution of custom CNN architecture and pretrained models for scene classification. The RepConv architecture used by Soudy et al. (2022) gives 93.55% and 75.54% accuracies for training and validation respectively on multi-class classsifiaction while our customized architecture gives 83% accuracy both on training and validation respectively. The fact that our customized architecture achieves a similar accuracy on both training and validation sets suggests a better balance between model complexity and generalization. While RepConv's impressive accuracy on training data is appealing, the comparatively lower validation accuracy might indicate potential overfitting concerns. It's crucial to strike a balance between training and validation accuracies to ensure that the model generalizes well to unseen data.

The most interesting thing about this research is the poor performance of ResNet-50 model when compared to other models. Further investigation would need to be done to understand why ResNet50 underperformed relative to other models. Hyperparameters tuning of the model such as exploring different learning rates or considering variations in optimizers which could provide insights for improvement of the model needs to be investigated.

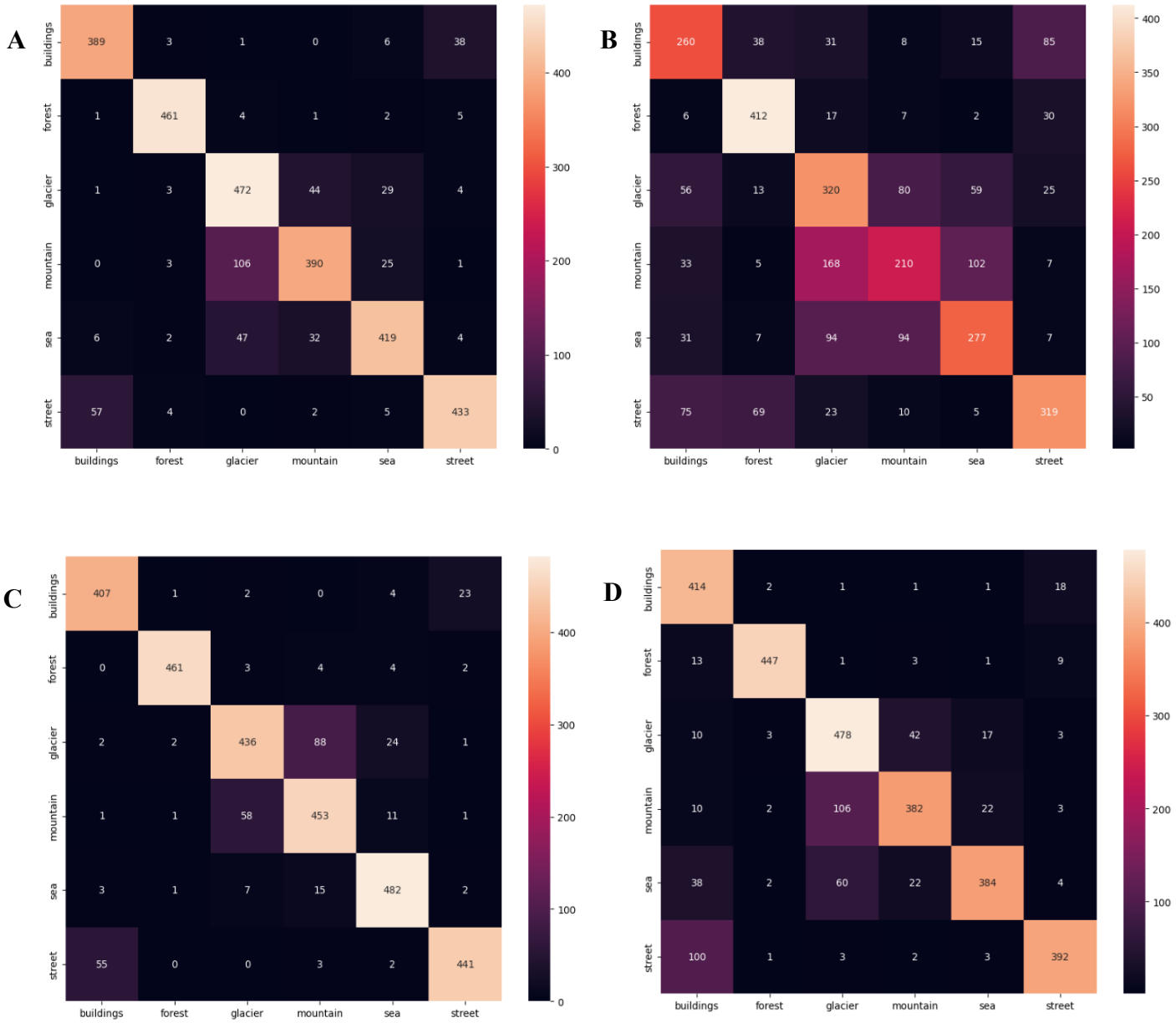

Figure 6 - Confusion matrix of the models A: VGG-16 Model. B ResNet-50 Model. C: Inception V3 Model. D: Customized CNN Model.

Conclusion

In artificial intelligence and computer vision, image classification stands as the basis for machine to learn and understand visual data. In this research, performance of four CNN models including the customized model were evaluated for natural scene classification using the intel image dataset which was gotten from Kaggle.com. Various hyper-parameter tuning such as different optimizers, learning rate etc. was carried out to fine-tune the customized model, ultimately achieving an accuracy of 83%. The customized model was then compared with the pretrained model such as VGG-16, Inception V3 and ResNet-50. VGG-16, Inception V3 and the customized architecture performed so well on the dataset with their accuracy of above 80% while the ResNet-50 model performance was poor with accuracy of 60%. Finally, Inception V3 is the best model for natural scene classification because it gives high accuracy, recall and precision while minimizing loss. With its notably lower loss value, Inception V3 demonstrated its capability to understand and classify natural scenes efficiently. For future work, alternative pretrained model like DenseNet, EfficientNet and MobileNet would be explore and the best performing models will be implemented in real time application such as mobile apps or web apps to test their performance in real world scenarios.

References

Ali, L., Alnajjar, F., Jassmi, H.A., Gocho, M., Khan, W. & Serhani, M.A. (2021) Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures. *Sensors*, 21(5), 1688. Available online: https://doi.org/10.3390/s21051688.

Baral, S. & Aryal, J. (2023) *Remote Sensing Image Classification Using Transfer Learning Based Convolutional Neural Networks: An Experimental Overview*. preprint. Available online: https://doi.org/10.36227/techrxiv.22581457.v1.

Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S. & Miao, Y. (2021) Review of Image Classification Algorithms Based on Convolutional Neural Networks. *Remote Sensing*, 13(22), 4712. Available online: https://doi.org/10.3390/rs13224712.

He, K., Zhang, X., Ren, S. & Sun, J. (2015) Deep Residual Learning for Image Recognition. Available online: https://doi.org/10.48550/ARXIV.1512.03385.

Inception V3 Model Architecture (2021) *OpenGenus IQ: Computing Expertise & Legacy*. Available online: https://iq.opengenus.org/inception-v3-model-architecture/ [Accessed 18/08/2023].

O'Shea, K. & Nash, R. (2015) An Introduction to Convolutional Neural Networks. arXiv. Available online: http://arxiv.org/abs/1511.08458 [Accessed 20/08/2023].

Pires De Lima, R. & Marfurt, K. (2019) Convolutional Neural Network for Remote-Sensing Scene Classification: Transfer Learning Analysis. *Remote Sensing*, 12(1), 86. Available online: https://doi.org/10.3390/rs12010086.

Shabbir, A., Ali, N., Ahmed, J., Zafar, B., Rasheed, A., Sajid, M., Ahmed, A. & Dar, S.H. (2021) Satellite and Scene Image Classification Based on Transfer Learning and Fine Tuning of ResNet50. *Mathematical Problems in Engineering*. Edited by M. Maqsood, 2021, 1–18. Available online: https://doi.org/10.1155/2021/5843816.

Soudy, M., Afify, Y. & Badr, N. (2022) RepConv: A novel architecture for image scene classification on Intel scenes dataset. *International Journal of Intelligent Computing and Information Sciences*, 0(0), 1–11. Available online: https://doi.org/10.21608/ijicis.2022.118834.1163.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016) Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, 2818–2826. Available online: https://doi.org/10.1109/CVPR.2016.308.

VGGNet-architecture-19.ppm (850×498) (no date). Available online: https://www.researchgate.net/profile/Timea-Bezdan/publication/333242381/figure/fig2/AS:760979981860866@1558443174380/VGGNet-architecture-19.ppm [Accessed 18/08/2023].

Zeng, D., Liao, M., Tavakolian, M., Guo, Y., Zhou, B., Hu, D., Pietikäinen, M. & Liu, L. (2021) Deep Learning for Scene Classification: A Survey. arXiv. Available online: http://arxiv.org/abs/2101.10531 [Accessed 22/08/2023].