November 7, 2023

*Digital Object Identifier*

**AAI - Intel Image Classification**

**RAFAEL DIAS (50%)1, FATEMEH KHAJEHMAHMOUD (50%)2**

1Universidade de Aveiro, Departamento de Electrónica, Telecomunicações e Informática (e-mail: rafaelsdias@ua.pt)(95284)

2Universidade de Aveiro, Departamento de Electrónica, Telecomunicações e Informática (e-mail: fat.khajehmahmoud@ua.pt)(110969)

Subject: Applications of Artificial Intelligence, Course instructor: Petia Georgieva

**ABSTRACT** This report presents the findings of a machine learning project focused on image classification using deep learning models. The project’s primary objective is to accurately categorize images into one of six distinct classes, including "buildings", "forest", "glacier", "mountain", "sea", and "street". The study evaluates three variations of the ResNet50 architecture (ResNet50\_h1, ResNet50\_h2, and ResNet50\_h3) and the VGG16 model. The analysis includes performance metrics such as accuracy, precision, recall, and F1-score. Notably, the results demonstrate that ResNet50\_h1 and VGG16 achieved remarkable accuracy and exhibited robust performance across various image categories, making them strong candidates for image classification tasks. The insights gained from this research contribute to a better understanding of model performance in image classification and provide a foundation for future work in this field.

**INDEX TERMS** Accuracy Metrics, Classifier Comparison, CNNs, Convolutional Neural Networks, Deep Learning, Image Classification, Image Preprocessing, Image Recognition, Machine Learning Models, ResNet50, Supervised Learning, VGG16

1. **INTRODUCTION**

**I**

MAGE classification is a fundamental task in computer vision. In this report, we dive into a machine learn- ing project that focuses on image classification using deep learning models. The primary objective of this project was to accurately classify images into one of six distinct cat- egories: "buildings", "forest", "glacier", "mountain", "sea",

and "street".

Machine learning and deep learning techniques have revo- lutionized image classification by automating the process of identifying objects and scenes within images. The potential applications of accurate image classification are vast, includ- ing content recommendation, facial recognition, and even environmental monitoring. In this context, the models under evaluation are crucial components, as their effectiveness in classifying images can impact real-world applications.

The models assessed in this project consist of three variations of the ResNet50 architecture (ResNet50\_h1, ResNet50\_h2, and ResNet50\_h3) and the VGG16 model. These models were trained and evaluated on a comprehensive dataset, with a focus on key performance metrics, such as accuracy, precision, recall, and F1-score.

This report will provide an in-depth analysis of the results obtained from the models in classifying images across differ- ent categories. The accuracy and metrics obtained from the models will be discussed in detail, offering insights into their

effectiveness.

By the end of this report, we will have a clear understand- ing of the performance of these models, which can serve as a foundation for future work in image classification and deep learning.

In the following sections, we will explore the dataset, model architectures, training, evaluation, results, and con- clusions, ultimately providing a comprehensive view of the image classification project’s outcome.

1. **STATE OD THE ART REVIEW**
2. ***SOME OF THE BEST WORKS IN IMAGE CLASSIFICATION MODEL***

There have been significant ideas in the image classification, with researchers seeking to develop and improve their perfor- mance.

To best determine which algorithms, techniques, and met- rics are the best to use in terms of image classification, a state- of-the-art analysis was conducted. Some ideas of this model include:

* 1. “Gradient-based Learning Applied to Document Recognition” [1]

Introduced in 1998, LeNet sets the foundation for fu- ture image classification research using convolutional neural networks. Many classic CNN techniques, such as pooling layers, fully connected layers, layering lay-

ers, and activation layers are used for feature extraction and classification. With a mean square error loss func- tion and 20 training sessions, this network can achieve 99.05% accuracy on the MNIST test set. Even after 20 years, many advanced classification networks still follow this pattern.

* 1. “ImageNet Classification with Deep Convolutional Neural Networks” [2]

Regarding deep learning models, this document re- views AlexNet. AlexNet’s breakthrough in 2012 marked a turning point in computer vision, dra- matically improving ImageNet accuracy from 73.8% to 84.7%. It introduced architectural advancements, larger networks with ReLU activation, GPU training, and dropout to handle complex features. AlexNet’s framework, combining convolution, ReLU, max- pooling, and dense layers, defined the standard for classification networks for the next decade.

* 1. “Very Deep Convolutional Networks for Large-Scale Image Recognition” [3]

This study focuses on VGG network. Following AlexNet’s design, the VGG network has two major updates: 1) VGG not only used a wider network like AlexNet but also deeper. VGG-19 has 19 convolution layers, compared with 5 from AlexNet. 2) VGG also demonstrated that a few small 3x3 convolution filters can replace a single 7x7 or even 11x11 filters from AlexNet, achieve better performance while reducing the computation cost. Because of this elegant design, VGG also became the backbone network of many pioneering networks in other computer vision tasks, such as FCN for semantic segmentation, and Faster R-CNN for object detection. With a deeper network, gradient vanishing from multi-layers back-propagation becomes a bigger problem. To deal with it, VGG also discussed the importance of pre-training and weight initialization. This problem limits researchers to keep adding more layers; otherwise, the network will be hard to converge.

* 1. “Deep Residual Learning for Image Recognition” [4]

In this article, present a residual learning framework (ResNet) to ease the training of networks that are substantially deeper than those used previously. It ex- plicitly reformulates layers as learning residual func- tions by reference to layer inputs, rather than learning dereferenced functions. It provides general experimen- tal evidence that these residual networks are easier to optimize and can achieve significantly increased accuracy from depth. On the ImageNet dataset, it eval- uates residual networks with a depth of up to 152 layers—8 times deeper than VGG networks, but still less complex. A set of these residual networks achieves 3.57% error on the ImageNet test set. This result was ranked 1st in the 2015 ILSVRC classification task. It also provides CIFAR-10 analysis with 100 and 1000 layers.

* 1. “Xception: Deep Learning with Depthwise Separable Convolutions” [5]

This paper presents an interpretation of Inception mod- ules in convolutional neural networks as an inter- mediate step between regular convolution and deep separable convolution operations (a depth convolution followed by a pointwise convolution). In this light, a deeply separable complex can be understood as an Inception module with a maximum number of towers. This architecture, called Xception, performed slightly better than Inception V3 on the ImageNet dataset (for which Inception V3 was designed), and significantly better than Inception V3 on a larger image classifi- cation dataset of 350 million images and 17,000 The class acts. Since the Xception architecture has the same number of parameters as Inception V3, the perfor- mance increase is not due to increased capacity, but due to more efficient use of model parameters.

Overall, CNN models have revolutionized the field of com- puter vision by allowing automatic feature learning from raw data and have been adapted for various domains, becoming an integral part of modern deep learning applications.

1. ***EXISTING DEEP LEARNING MODELS***

Several previous studies have explored deep learning models for image classification. There are still limitations and chal- lenges associated with existing models. Many deep learn- ing models require substantial amounts of labeled data for training. This can be a limitation when working with niche or specialized image classification tasks where labeled data is scarce. Training deep neural networks, especially very deep architectures like ResNet or Inception, demands signif- icant computational resources, including powerful GPUs and TPUs. This can be a practical limitation for researchers and organizations with limited resources. Deep learning models can inherit biases present in the training data, potentially leading to biased predictions. Addressing issues of bias and fairness is a critical concern in image classification, espe- cially in sensitive applications. The use of deep learning models in image classification has raised ethical concerns, especially in applications like surveillance, privacy invasion, and deepfake generation.

We apply various CNN models, including two popular pre-trained models for image classification: ResNet50 and VGG16. These models are commonly used in transfer learn- ing scenarios due to their strong performance on a wide range of image-related tasks. In this case, the ResNet50 model is trained for 5 epochs with early stopping, and model check- point callbacks to save the best model based on validation accuracy, and the VGG16 model used SGD optimizer, and training is done for 5 epochs with early stopping and model checkpoint callbacks. Compare Metrics (Precision, Recall, F1-Score) for Different Models.

1. **PROBLEMS AND OBJECTIVES**
2. ***PROBLEMS***

In this section, we aim to provide a comprehensive under- standing of the challenges and issues that our image classifi- cation project seeks to address. These problems are essential for setting the context and motivation behind the project and for identifying areas where image classification models can make a significant impact.

1. Multi-Class Image Classification

The primary challenge at the heart of this project is multi- class image classification. We are tasked with categorizing images into one of six distinct classes: "buildings", "forest", "glacier", "mountain", "sea", and "street". Each of these cate- gories exhibits a wide range of visual characteristics, making this task inherently complex. Our objective is to develop models that can accurately and consistently classify images across these diverse categories.

1. Model Selection

Selecting the appropriate deep learning model architecture is a critical consideration. To tackle multi-class image classifi- cation effectively, we have evaluated three variations of the ResNet50 architecture and the VGG16 model. The decision regarding which model to employ has a profound impact on the project’s success. We seek to identify which model is best suited to address the unique challenges of this classification task.

1. Performance Metrics

Evaluating the models’ performance goes beyond a simple accuracy score. To gain a nuanced understanding of their effectiveness, we must consider additional metrics such as precision, recall, and F1-score. Each of these metrics pro- vides distinct insights into the models’ behavior and their ability to correctly classify images. Striking the right balance among these metrics is crucial to meet our classification objectives.

1. Overfitting and Generalization

The challenge of overfitting and generalization is a common concern in deep learning projects. While we aim to achieve high accuracy on the training dataset, it is equally important that our models generalize well to unseen data without be- coming overly specialized to the training set. Ensuring that our models exhibit robust generalization capabilities is an overarching objective.

1. ***OBJECTIVES***

To address the aforementioned problems, we have estab- lished a set of clear and measurable objectives for our image classification project:

1. Evaluate Model Performance

Our primary objective is to thoroughly evaluate the perfor- mance of the ResNet50\_h1, ResNet50\_h2, ResNet50\_h3,

and VGG16 models in the context of multi-class image classification. We will compare their accuracy, precision, recall, and F1-score to determine which model demonstrates the most robust classification capabilities.

1. Understand Model Behavior

A key aspect of our project is to gain a comprehensive understanding of how each model behaves when classifying images from the six distinct categories. We aim to iden- tify their strengths and weaknesses and explore the reasons behind their performance on specific classes. This deeper insight into model behavior will inform our selection of the most suitable model.

1. Select the Best Model

Our ultimate objective is to identify and select the best model for image classification. This entails considering a model that not only achieves high accuracy but also demonstrates well- balanced precision, recall, and F1-score. The chosen model will serve as the foundation for the classification task.

1. Contribute to Future Work

Beyond the immediate objectives of model selection, our analysis aims to contribute valuable insights and data that can serve as a reference for future research and applications in image classification. Our findings, methodologies, and per- formance assessments will be made available to the research community to facilitate further advancements in the field.

In the subsequent sections of this report, we will delve into the methodology, dataset, model architectures, training pro- cess, results, and conclusions, all with the aim of addressing these problems and achieving our defined objectives.

1. **DATA DESCRIPTION**

In this section, we provide an in-depth description of the dataset used in our image classification project. Understand- ing the characteristics of the dataset is essential for develop- ing and evaluating deep learning models effectively.

The dataset described in this report is a collection of images published by Intel to host an Image Classification Challenge. It contains approximately 25,000 images, each with a size of 150x150 pixels, distributed under six cat- egories: "buildings", "forest", "glacier", "mountain", "sea", and "street".

The dataset used in our project contains a total of 17,034 images, divided into two subsets: a training set with 14,034 images and a testing set with 3,000 images. Understanding the distribution of images across these classes is essential for assessing the balance and potential biases in the dataset.

***A. DATASET STATISTICS***

The dataset statistics are summarized as follows:

* Total Number of Images: 17,034
* Number of Classes: 6
* Images per Class in Training Data:
  + Buildings: 2,191
  + Forest: 2,271
  + Glacier: 2,404
  + Mountain: 2,512

**–** Sea: 2,274

* + Street: 2,382
* Images per Class in Testing Data:
  + Buildings: 437
  + Forest: 474
  + Glacier: 553
  + Mountain: 525

**–** Sea: 510

* + Street: 501

The train, test, and prediction data are separated into individual zip files. Approximately 14,000 images are in the training set, 3,000 are in the test set, and 7,000 are in the prediction set. Images are loaded from TensorFlow’s Keras API, and scikit-learn is used to encode the image categories (classes) into numeric labels. This is important for classification tasks, and the data is further split into training and testing sets.

The primary dataset consists of images used for image classification, and each image belongs to a specific category. The dataset is organized into subdirectories based on these categories. In this study, these categories are extracted from the subdirectories where the images are stored. The cate- gories variable holds the list of category names, and the total number of categories is printed.

1. **DATA VISUALIZATION**

Data visualization is a crucial step in understanding the dataset and the results of our image classification models. We used various graphical representations to gain insights into the data and the performance of the models.

1. ***CLASS DISTRIBUTION***

The bar chart shown in Figure 1 illustrates the distribution of images across the six classes. It is evident from the chart that the dataset is relatively well-balanced, with each class containing a reasonable number of images for training and testing.

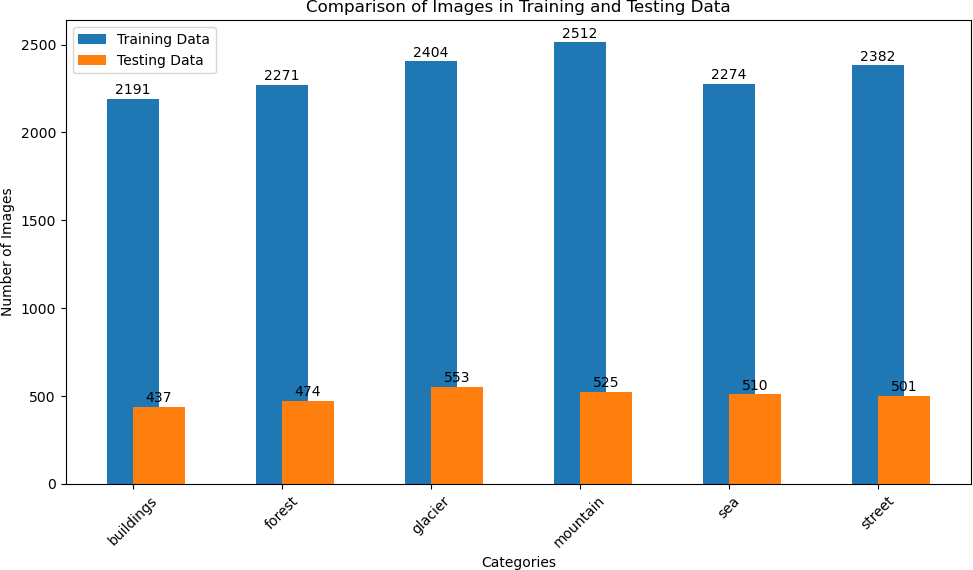


FIGURE 1: Class Distribution in the Dataset

1. ***SAMPLE IMAGES***

In this subsection, we provide visual samples of images from the dataset to give you a sense of the data. We’ve selected the "buildings" category to showcase four sample images from both the testing and training datasets.



FIGURE 2: Sample Images from Testing Data (Category: Buildings)



FIGURE 3: Sample Images from Training Data (Category: Buildings)

These images are representative of the "buildings" cate- gory and provide a visual understanding of the type of data used in the training and testing phases.

1. **DATA PREPROCESSING**

Data preprocessing is a fundamental step in preparing the dataset for training and evaluation. It involves tasks such as resizing, normalization, and data augmentation to ensure that the data is in a suitable format for deep learning models.

1. ***IMAGE RESIZING***

The images in the dataset were originally of size 150x150 pixels, and we have maintained this original size. If any images were different in size, we resized them to match the standard size of 150x150 pixels. This standardization ensures that all images have the same dimensions for consistent processing by the deep learning models.

1. ***NORMALIZATION***

To facilitate convergence during training, pixel values in the images were normalized to the range [0, 1].

1. ***DATA AUGMENTATION (FUTURE WORK)***

Data augmentation techniques are commonly used to in- crease the diversity of the training dataset and enhance the generalization of deep learning models. These techniques include random rotations, flips, brightness adjustments, and other transformations applied to the images.

In the current project, we did not employ data augmenta- tion. However, it’s important to note that data augmentation

is a valuable tool for addressing overfitting and improving model robustness, particularly when the dataset is limited.

While data augmentation was not utilized in this project, it is a potential avenue for future work. Incorporating data aug- mentation techniques could further enhance the performance of our image classification models by providing them with more varied and augmented training data.

In the subsequent sections of this report, we will provide detailed insights into the model architectures, training pro- cedures, and results, all of which are informed by the data description, visualization, and preprocessing stages.

1. **APPLIED MACHINE LEARNING METHODS**
2. ***OVERVIEW OF THE MACHINE LEARNING METHODS USED***

Different machine learning algorithms were applied to solve an image classification problem using a dataset. These al- gorithms are based on two popular pre-trained models: ResNet50 and VGG16. Here’s a description of each applied algorithm:

1. ResNet-50

ResNet-50 is a convolutional neural network (CNN) archi- tecture known for its deep structure and the use of resid- ual connections, which helps address the vanishing gradient problem. The ResNet50 architecture is used as a base model, and its pre-trained weights are loaded from the ImageNet dataset.

These pre-trained weights capture useful features from a wide range of images, and the model leverages these features for the current classification task. It consists of convolutional layers, batch normalization, max-pooling, and global average pooling layers. The final layer is a fully connected (dense) layer with softmax activation, which produces class proba- bilities.

After loading the pre-trained ResNet-50 model, the code fine-tunes it for the specific image classification task. Fine- tuning involves adjusting the model’s parameters to adapt to the new dataset. In this code, the fully connected layers at the top of the model are replaced with new layers for the target classification task.

The code uses an optimizer, and the choice of optimizer depends on the hyperparameters established. For the ResNet- 50 model, it employs Stochastic Gradient Descent (SGD) and Adam and compiles the model using categorical cross- entropy as the loss function. The model is also configured to track accuracy as a metric.

For both models, the training occurs over a specified number of epochs. The code also sets up callbacks for model checkpointing and early stopping to save the best model and prevent overfitting.

After training, the model is evaluated on the testing data provided, which includes computing accuracy and other per- formance metrics. The trained ResNet-50 model is saved to a file, which can be later loaded for predictions on new data.

Overall, ResNet-50 is used as a powerful feature extractor and classifier for the image classification task. The code adapts this pre-trained model to a specific dataset, fine- tunes it, and then evaluates its performance. This approach is commonly used in transfer learning for computer vision tasks, as it leverages the knowledge gained from a large dataset like ImageNet to improve classification accuracy on a smaller, domain-specific dataset.

1. VGG16

The VGG16 model is used as a machine learning method for image classification. VGG16 is a convolutional neural network (CNN) architecture known for its simplicity and effectiveness.

VGG16 is used as a strong feature extractor and classifier for the image classification task. The code adapts this pre- trained model to a specific dataset, fine-tunes it by training only the top layers, and then evaluates its performance. This approach is commonly used in transfer learning for computer vision tasks, as it leverages the knowledge gained from a large dataset like ImageNet to improve classification accuracy on a smaller, domain-specific dataset.

The model is initialized with pre-trained weights on the ImageNet dataset. These pre-trained weights capture useful features from a wide range of images, and the model lever- ages these features for the current classification task.

This model is a relatively deep neural network architecture that consists of 16 weight layers (hence the name). It includes multiple convolutional layers with small 3x3 filters, followed by max-pooling layers for downsampling. The architecture is known for its simplicity, with repeated convolutional and max-pooling blocks. It also features fully connected layers at the end.

In the code, the pre-trained layers of VGG16 are frozen, which means that their weights are not updated during training. Only the new layers added for the specific image classification task will be trained. Freezing the pre-trained layers helps preserve the knowledge learned from ImageNet and prevents overfitting when dealing with a smaller dataset.

The code defines an optimizer (Stochastic Gradient De- scent, or SGD) and compiles the model using categorical cross-entropy as the loss function. The model is also con- figured to track accuracy as a metric.

The VGG16 model is trained using the training data provided by the train generator. The training occurs over a specified number of epochs. The code also sets up callbacks for model checkpointing and early stopping to save the best model and prevent overfitting.

After training, the model is evaluated on the testing data provided by the test generator. This evaluation includes com- puting accuracy and other performance metrics. The trained VGG16 model is saved to a file, which can be later loaded for predictions on new data.

1. ***EXPLANATION OF MODEL SELECTION AND HYPERPARAMETER TUNING***

Model selection and hyperparameter tuning are important aspects of building a successful image classification model. The project involved defining different hyperparameters for each model configuration (ResNet50 and VGG16). These hyperparameters include the choice of optimizer, learning rate, momentum, and the number of dense units in the final fully connected layer.

The learning rate, a critical hyperparameter, was set dif- ferently for ResNet50 and VGG16 models. These values can significantly impact training stability and convergence speed. Momentum and Beta, another hyperparameter for the op-

timizer, was chosen differently for ResNet50 and VGG16.

The number of dense units in the final fully connected layer is a hyperparameter that can affect the model’s capacity and ability to fit the data. Different values were used for ResNet50 and VGG16.

Callbacks, such as ModelCheckpoint and EarlyStopping, were used for both model configurations. These callbacks help in monitoring the training process and making decisions based on specific conditions. For example, ModelCheck- point saves the model with the best validation accuracy, and EarlyStopping stops training if the validation loss doesn’t improve after a certain number of epochs.

1. Hyperparameters

# ResNet50 Model 1 (ResNet50\_h1)

Hyperparameters:

* + - Optimizer: Stochastic Gradient Descent (SGD)
    - Learning Rate: 0.001
    - Momentum: 0.9
    - Dense Units: 256

# ResNet50 Model 2 (ResNet50\_h2)

Hyperparameters:

* + - Optimizer: Adam
    - Learning Rate: 0.01
    - Momentum: 0.5
    - Dense Units: 64

# ResNet50 Model 3 (ResNet50\_h3)

Hyperparameters:

* + - Optimizer: Adam
    - Learning Rate: 0.001
    - Momentum: 0.9
    - Dense Units: 256

# VGG16 Model

Hyperparameters:

* + - Optimizer: Stochastic Gradient Descent (SGD)
    - Learning Rate: 0.001
    - Momentum: 0.9
    - Dense Units: 256

Each model was trained with the specified hyperparame- ters, and their performance on the validation data was moni- tored. The model with the best validation accuracy was saved.

1. **RESULTS**
2. ***RESNET50 MODELS***

The project evaluated three variations of the ResNet50 model for image classification. Each model’s performance was as- sessed on a diverse dataset featuring image categories like buildings, forests, mountains, glaciers, seas, and streets.

1. ResNet50\_h1

**ResNet50\_h1** achieved an impressive accuracy of 93% and consistently high F1-scores across all classes. The classifica- tion report for this model is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |
| buildings | 0.93 | 0.92 | 0.92 | 437 |
| forest | 0.98 | 0.99 | 0.99 | 474 |
| glacier | 0.87 | 0.89 | 0.88 | 553 |
| mountain | 0.89 | 0.88 | 0.88 | 525 |
| sea | 0.97 | 0.94 | 0.96 | 510 |
| street | 0.93 | 0.95 | 0.94 | 501 |

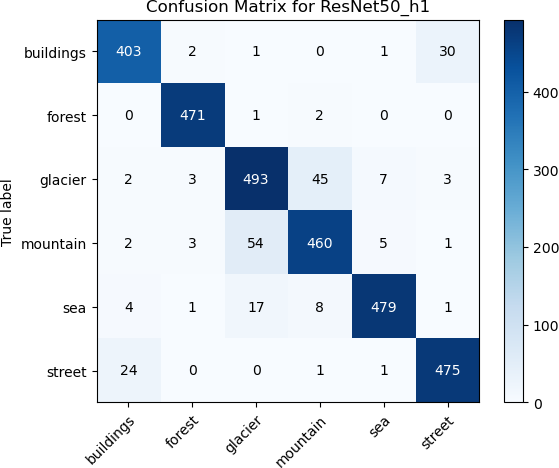


FIGURE 4: Confusion Matrix for ResNet50\_h1

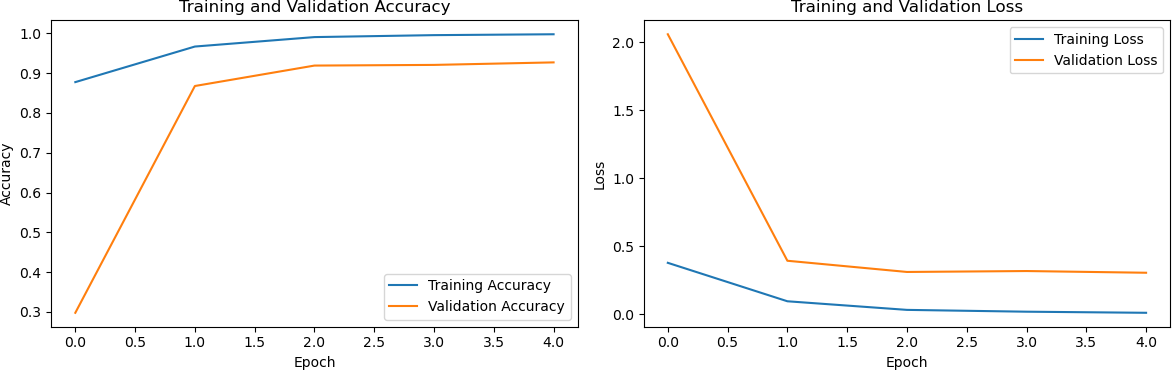
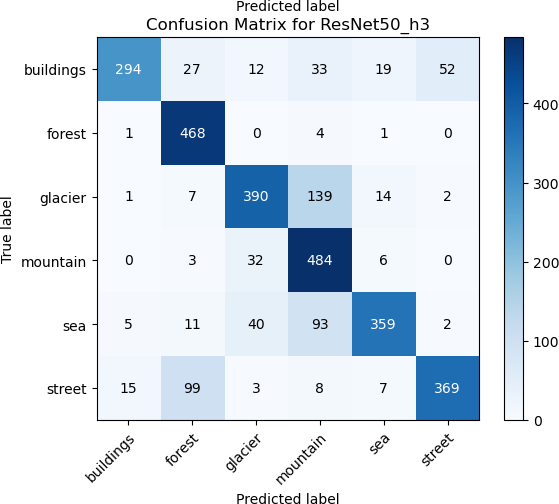


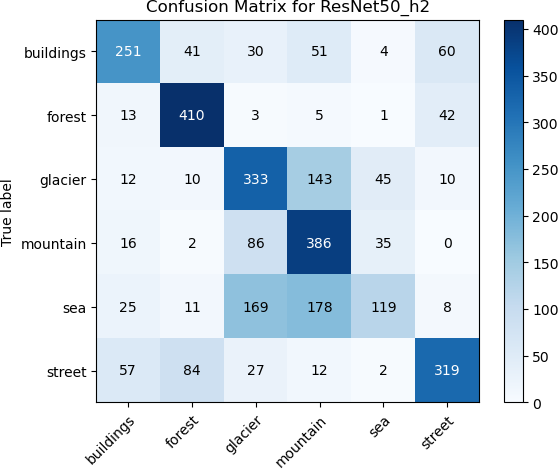
FIGURE 5: Training and Validation Accuracy, Training and Validation Loss for ResNet50\_h1

1. ResNet50\_h2

**ResNet50\_h2** demonstrated lower accuracy (61%) and F1- scores. The classification report for this model is as follows:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |
| buildings | 0.67 | 0.57 | 0.62 | 437 |
| forest | 0.73 | 0.86 | 0.79 | 474 |
| glacier | 0.51 | 0.60 | 0.55 | 553 |
| mountain | 0.50 | 0.74 | 0.59 | 525 |
| sea | 0.58 | 0.23 | 0.33 | 510 |
| street | 0.73 | 0.64 | 0.68 | 501 |

FIGURE 6: Confusion Matrix for ResNet50\_h2

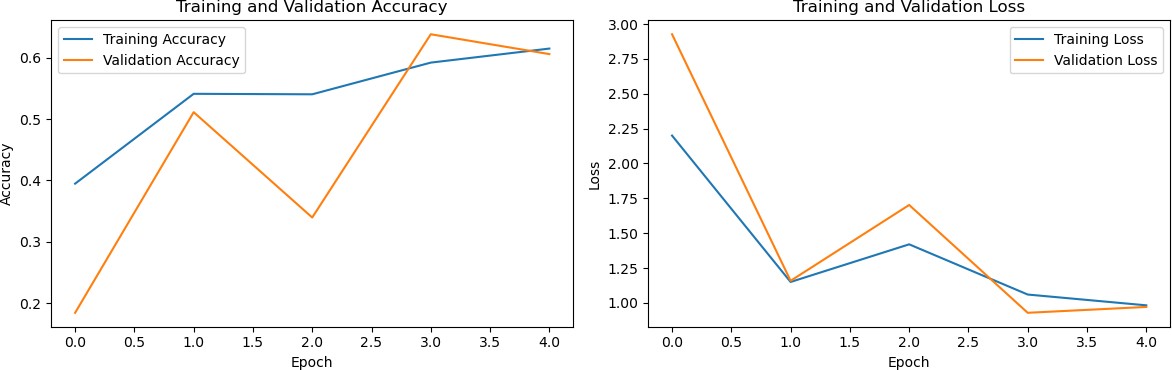


FIGURE 7: Training and Validation Accuracy, Training and Validation Loss for ResNet50\_h2

1. ResNet50\_h3

**ResNet50\_h3** demonstrated improvements over ResNet50\_h2 but still had lower accuracy and F1-scores compared to ResNet50\_h1 and VGG16. The classification report for this model is as follows:

FIGURE 8: Confusion Matrix for ResNet50\_h3

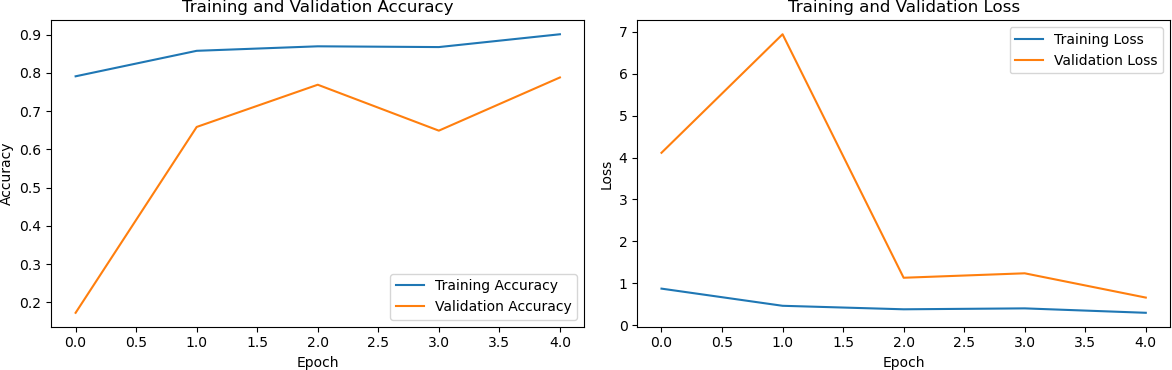


FIGURE 9: Training and Validation Accuracy, Training and Validation Loss for ResNet50\_h3

1. ***VGG16 MODEL***

The VGG16 model exhibited strong performance in image classification tasks. It was fine-tuned on the provided dataset, resulting in the following classification report:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |  | Class | Precision | Recall | F1-Score | Support |
| buildings | 0.93 | 0.67 | 0.78 | 437 |  | buildings | 0.94 | 0.84 | 0.88 | 437 |
| forest | 0.76 | 0.99 | 0.86 | 474 |  | forest | 0.96 | 0.99 | 0.97 | 474 |
| glacier | 0.82 | 0.71 | 0.76 | 553 |  | glacier | 0.82 | 0.80 | 0.81 | 553 |
| mountain | 0.64 | 0.92 | 0.75 | 525 |  | mountain | 0.76 | 0.86 | 0.81 | 525 |
| sea | 0.88 | 0.70 | 0.78 | 510 |  | sea | 0.93 | 0.80 | 0.86 | 510 |
| street | 0.87 | 0.74 | 0.80 | 501 |  | street | 0.86 | 0.95 | 0.91 | 501 |

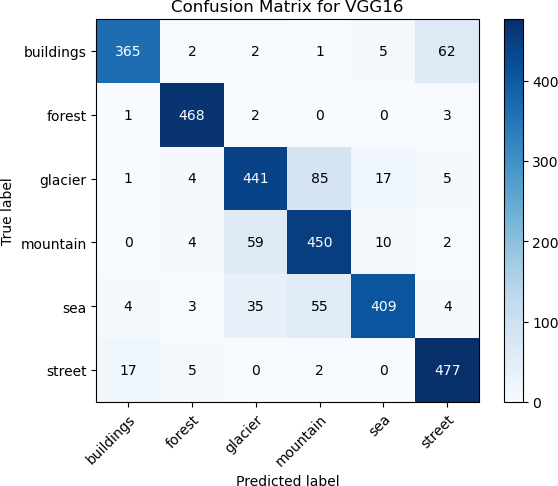


FIGURE 10: Confusion Matrix for VGG16

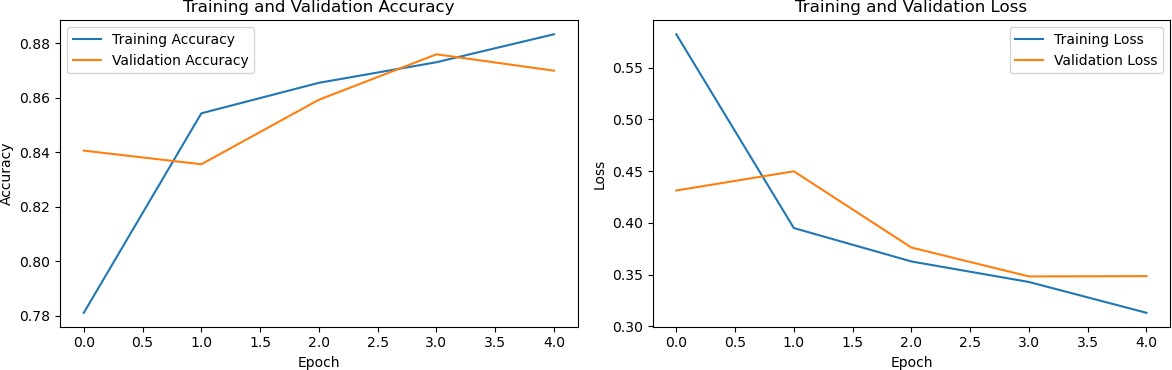


FIGURE 11: Training and Validation Accuracy, Training and Validation Loss for VGG16

1. ***COMPARISON OF MODELS***

In the comparative evaluation of the models, the performance of ResNet50\_h1 and VGG16 stood out, exhibiting strong capabilities in image classification with high accuracy and excellent F1-scores. The comprehensive assessment of preci- sion, recall, and F1-scores across all classes highlighted their proficiency. ResNet50\_h1, fine-tuned on a diverse dataset encompassing image categories such as buildings, forests, mountains, glaciers, seas, and streets, showcased exceptional classification capabilities.

**ResNet50\_h1** achieved a remarkable accuracy of 93%, with consistently high F1-scores for all classes, notably at- taining a 0.99 F1-score for the ’forest’ class. The depth of the ResNet-50 architecture and the incorporation of residual connections played a pivotal role in mitigating the vanishing gradient problem, leading to its exceptional performance in image classification tasks. The strengths of this architecture make ResNet50\_h1 an ideal choice for this project.

In contrast, **ResNet50\_h2** exhibited comparatively lower accuracy at 61% along with lower F1-scores. This model encountered challenges in distinguishing certain classes, par- ticularly struggling with the ’sea’ class, which had an F1- score of 0.33. These results suggest potential issues with overfitting or suboptimal fine-tuning, impacting the model’s ability to differentiate classes effectively.

**ResNet50\_h3**, although showing improvement over ResNet50\_h2, still fell short of the performance achieved by ResNet50\_h1 and VGG16 in terms of accuracy and F1- scores. It achieved an accuracy of 79% and demonstrated better class differentiation but did not reach the same level of performance as the top models.

**VGG16** emerged as a strong contender with an accuracy of 87% and consistently high F1-scores across all classes. Similar to ResNet50\_h1, VGG16 exhibited robustness in recognizing all classes effectively. Depending on specific project requirements, both ResNet50\_h1 and VGG16 can be considered top-performing models for this context, offering a balance between accuracy and F1-scores.

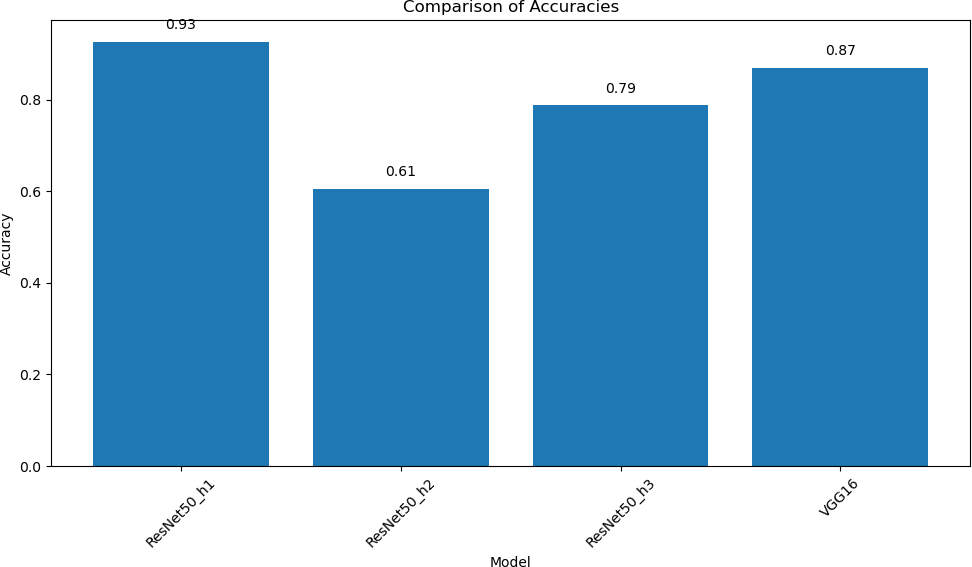


FIGURE 12: Comparison of Model Accuracies

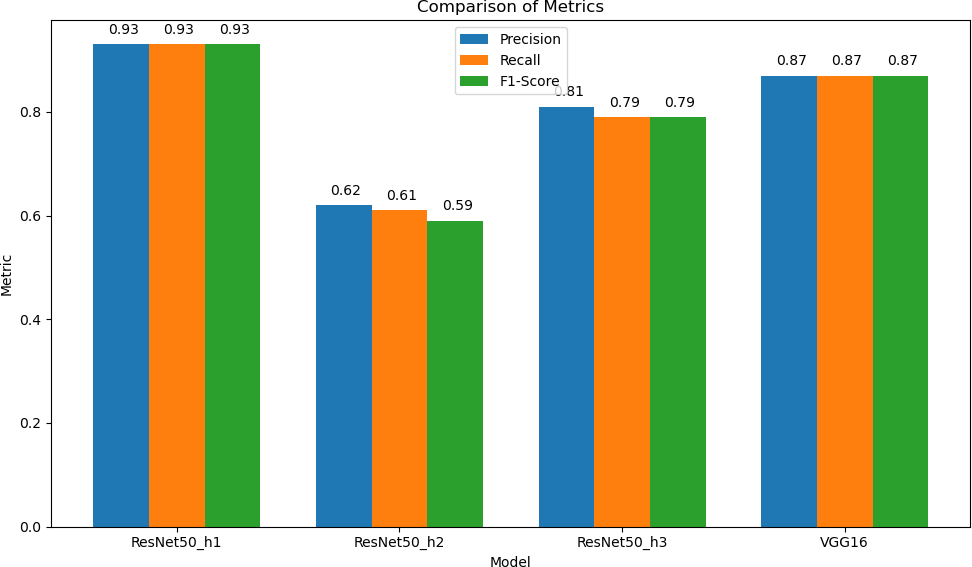


FIGURE 13: Comparison of Metrics (Precision, Recall, F1- Score)

1. **CONCLUSIONS**

In this section, we summarize the key findings and con- clusions drawn from our image classification project using ResNet50 and VGG16 models with various hyperparameters.

1. ***PERFORMANCE OF RESNET50 MODELS***

The ResNet50 model, particularly the ResNet50\_h1 varia- tion, exhibited exceptional performance in the context of image classification. ResNet50\_h1 achieved an impressive accuracy of 0.927 and consistently demonstrated high pre- cision, recall, and F1-scores across different image cate- gories. This model excelled in distinguishing between vari-

ous classes, making it a dependable choice for image clas- sification tasks that require a balance between precision and recall.

1. ***PERFORMANCE OF VGG16 MODEL***

The VGG16 model also delivered commendable results with an accuracy of 0.87. VGG16 maintained a strong balance between precision, recall, and F1-scores across diverse im- age categories. It proved to be a robust choice for image classification, offering consistent performance across various classes.

1. ***MODEL SELECTION CONSIDERATIONS***

The choice between ResNet50 and VGG16 should be driven by the specific requirements of the image classification task:

* + **ResNet50\_h1** is recommended when high precision and recall are of paramount importance. It excelled in correctly identifying different classes, making it a strong choice for tasks where accurate categorization is crucial.
  + **VGG16** is a solid contender when there is a need for an accurate and reliable model that performs consistently across various image categories. Its performance across different classes makes it a versatile choice.

1. ***HYPERPARAMETER TUNING AND EXPERIMENTATION***

Our code emphasizes the significance of hyperparameter tuning and experimenting with different model architectures to achieve optimal results. It underscores that fine-tuning models to adapt to the specific problem domain and dataset can substantially impact performance. Further experimenta- tion and parameter optimization have the potential to enhance the performance of both ResNet50 and VGG16 models.

Consideration of ensemble techniques or transfer learning from pre-trained models may also enhance the performance of these models. Our project has laid the foundation for ongoing research and exploration in this direction.

1. ***OVERALL CONCLUSION***

In conclusion, the code and results presented here highlight the critical importance of selecting the right convolutional neural network architecture and fine-tuning hyperparameters for image classification tasks. Careful consideration of the specific problem requirements and dataset characteristics is essential for determining the most suitable model for a given task. Both ResNet50 and VGG16 have proven to be strong candidates for image classification, each with its own unique strengths and areas of application.

1. **SUGGESTIONS FOR FUTURE WORK**

This work may have been customized and fine-tuned for the specific problem we are trying to solve. This level of domain expertise can lead to better results compared to generic or off-the-shelf solutions.

It might incorporate hyperparameter tuning that optimizes the model's performance for our dataset. This is particular-ly important in deep learning, where the right set of hyper-parameters can significantly impact results.

Effective data preprocessing, including data augmentation and normalization, can have a substantial impact on model performance. If our code handles data preprocessing better than the references, it could lead to superior results.

We have chosen a model architecture (ResNet50 and VGG16) that is better suited to our problem than the models used in the consulted references. Model selection is a critical factor in achieving good results.

Our dataset is of higher quality and larger in size compared to the datasets used in the consulted references. More data and cleaner data can lead to better model performance.

Our code is well-structured and optimized, it can lead to faster training and inference times and made it more practical for real-world applications.

Our work might represent the result of an iterative process where we continually refined and improved our code based on feedback and the observed results.

**REFERENCES**

[1]Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998.

1. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet clas- sification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 2012.
2. Shuying Liu and Weihong Deng. Very deep convolutional neural network based image classification using small training sample size. Proceedings of the IEEE, 2016.
3. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. Proceedings of the IEEE, 2016.
4. Francois Chollet. Xception: Deep learning with depthwise separable convolutions. arXiv preprint arXiv:1610.02357, 2017.

