2025

Scene Classification Using Convolutional Neural Network

PDAN8412 POE

MAXIMILIAN WALSH – ST10203070

21 NOVEMBER 2025

Table of Contents

Executive Summary

This project develops a Convolutional Neural Network (CNN) to classify images from the Intel Image Classification dataset. It uses PySpark for data preparation and TensorFlow for modelling. The CNN was trained to recognise six scene categories: buildings, forest, glacier, mountain, sea, and street. The final model achieved a validation accuracy of 86.5% with balanced precision and recall and minimal overfitting. Most errors occurred between visually similar natural scenes. As confirmed by the findings, CNNs are effective for extracting complex spatial patterns from unstructured image data. To enhance accuracy and generalisation in future work, the application of data augmentation and transfer learning should be considered.

Introduction

The use of Convolutional Neural Networks (CNNs) in predictive analytics has become the leading approach for image classification, as they learn complex spatial patterns directly from raw pixels (OpenAI, 2025). This project applies CNN-based modelling to the Intel Image Classification dataset containing over 14,000 labelled and categorised images. The objective is to develop and evaluate a deep learning model that can accurately recognise these classes using a structured workflow. Following the methodology outlined in the Analysis Plan section to come, the analysis includes data ingestion and validation with PySpark, exploratory data analysis, CNN model development in TensorFlow, and finally performance evaluation using accuracy, precision, recall, and confusion matrix analysis. The goal of this project is to show the power of predictive analytics in extending beyond traditional datasets, showing how CNNs can uncover meaningful patterns within complex visual information.

Analysis Plan

Developing a structured plan ensures that data analysis, image preparation, model construction, and evaluation are executed systematically. The following is a plan to transform the Image Intel Classification dataset into a multi-class predictive model capable of classifying natural scene images using a Convolutional Neural Network (CNN) supported by Spark-based data analysis:

Exploratory Data Analysis (EDA) Plan

The goal of the EDA is to understand the dataset's structure, verify its integrity, and examine the distribution of images across classes.

Planned steps:

- Use Spark to verify dataset integrity, confirm total image count, and check for missing or duplicate records.
- Check class balance across the six scene categories through visuals (buildings, forest, glacier, mountain, sea, street)
- To ensure accurate labelling and consistent visual quality, display sample images per class.
- To detect anomalies or corrupted files, record and interpret image metadata like file sizes and resolution.
- Summarise EDA findings and outline that the dataset meets the entry (10,000) requirements for CNN training.

Image Preparation Plan

Given that the dataset consists of colour images instead of the generic tabular data, preprocessing focuses on normalisation, augmentation, and dataset management instead of feature encoding.

Steps:

- Manifest Verification:
  - Use the Spark-generated intel_manifest.csv to confirm image paths and labels before model ingestion.
- Dataset Splitting:

- Using Spark, split the data into 80/20 train-validation splits and explore manifests (train_manifest.csv, val_manifest.csv) for TensorFlow.

- Preprocessing:
  - For uniform input, resize all images to 150 x 150 x 3.
  - Scale pixel values between 0-1 using normalisation.
  - For enhanced model robustenss, use real-time data augmentation like random flips, rotations, and zooms.
- Pipeline Setup:
  - Develop a TensorFlow pipeline (tf.data) to efficiently stream augmented batches to the CNN during training.

<u>Model Training Plan</u>

The task is a multi-class image classification problem predicting one of six categories.

Steps:

- Using the created training and validation manifests, create tf.data.Dataset objects.
- Use the following to build a baseline CNN:
  - For feature extraction, stacked Convo2D, ReLu, and MaxPooling2D layers.
  - Flatten and Dense layers for classification.
  - Final softmax activation layer with an output neuron per class (6).
- Compile the model with Adam optimiser, categorical cross-entropy loss, and accuracy as the main metric.
- To avoid overfitting, use early stopping, and to retain best performing weights, use model checkpointing.

<u>Model Evaluation Plan</u>

Evaluation will use metrics appropriate for multi-class classification (Muller & Guido, 2016):

- Accuracy: Overall proportion of correctly classified images.
- Precision, Recall, and F1-score: Per-class measures of predictive reliability and error balance.
- Confusion Matrix: Reveals similarities and model weaknesses by highlighting misclassified categories
- Loss and Accuracy Curves: Plotted over epochs to look at convergence and overfitting behaviour.

- Macro-averaged Scores: To check overall performance and comparison across classes.

Report Structure Plan

The final report will include:

1. Executive Summary: Summarises the overall objective, dataset, and main findings.
2. Introduction: Outlines the project purpose and rationale for applying CNNs to scene classification.
3. Analysis Plan: Describes the step-by-step analytical framework, including data preparation, CNN design, and evaluation strategy.
4. Dataset Justification: Reasoning for why the Intel Image Classification dataset is appropriate.
5. EDA Results: Class balance plots, representative image samples, and metadata checks.
6. Modelling Process: Outlines the CNN architecture, training parameters, and TensorFlow data pipeline setup.
7. Model Evaluation and Interpretation: Key performance metrics, confusion matrix, and discussion of errors.
8. Conclusion and Recommendations: Summarises overall insights, improvements, and extensions like transfer learning or deeper architectures.

Dataset Justification

Source: Kaggle – “Intel Image Classification” by Puneet Bansal (Bansal, 2018).

Link: https://www.kaggle.com/datasets/puneet6060/intel-image-classification/code

Size: 14,034 records, 2 columns

Target Variable: ‘label’ representing one of the six categories: buildings, forest, glacier, mountain, sea, street.

The selected dataset is the Intel Image Classification dataset from Kaggle, containing over 14,000 red, green, and blue images categorised into six scene types - buildings, forest, glacier, mountain, sea, and street. Post cleaning and validation with Spark, 14,034 usable records remained, which comfortably exceeds the minimum threshold of 10,000 records. Images are consistently sized at 150 x 150 pixels and evenly distributed across categories, making it an ideal dataset for convolutional neural network (CNN) experimentation. The dataset was chosen for the following reasons:

- Relevance to Task: The dataset aligns directly with the intention to build an image-based classification model using deep learning. Each image contains spatial features and colour textures that CNNs are specifically designed to learn (OpenAI, 2025). The dataset's structure fully supports the application of Spark for data handling and TensorFlow/Keras for modelling (Bansal, 2018).
- High-Quality Variables and Clear Labels: Each image is stored in a well-defined folder that offers explicit labels, enabling efficient Spark ingestion and supervised training. For traceability and reproducibility, the intel_manifest.csv created records the file path, byte size, and label for each image. During cleaning, no missing or corrupted files were found.
- Data Volume and Balance: The dataset is considered balanced enough for no bias modelling as each class contains 2,000 to 2,500 images. A Spark-based stratified split resulted in 11,286 training and 2,748 validation images, providing a sufficient amount of data for learning and evaluation.
- Suitability for CNN and Scalability: The image sizes are manageable for efficient experimentation even on CPUs, whilst offering visual complexity for multi-layer convolutional learning (Bansal, 2018).

To conclude, the Intel Image Classification dataset is ideal for this deep-learning exercise. It offers large, balanced, well-labelled, and clean data that provides a robust foundation for training and evaluation of a CNN capable of accurately differentiating scene types whilst satisfying the project requirements of Spark and TensorFlow.

Exploratory Data Analysis (EDA)

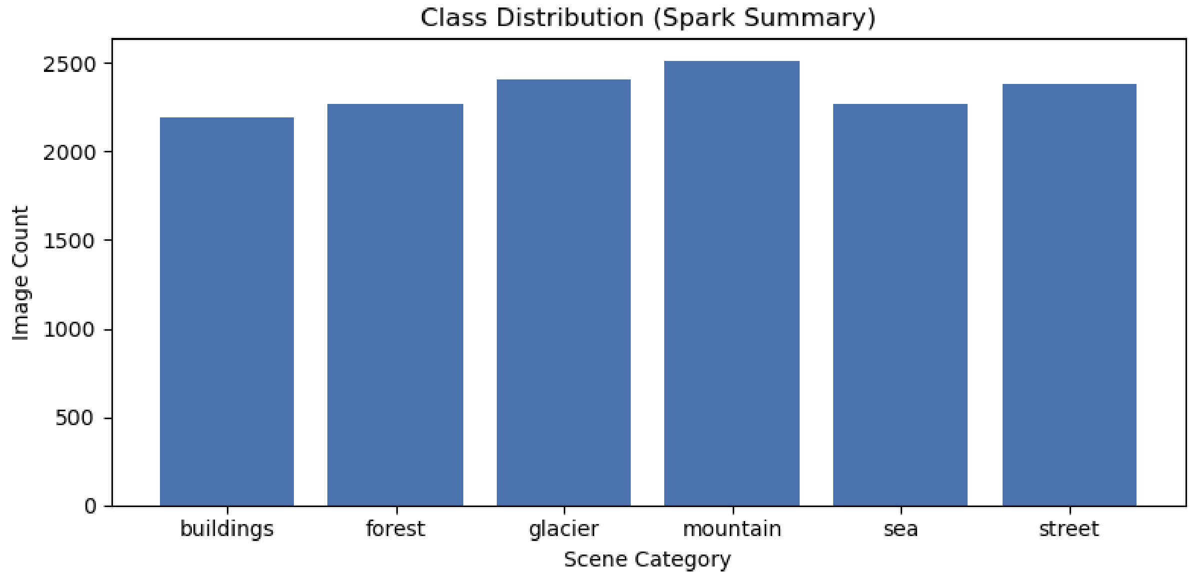

Figure 1: Class (Scene Categories) Distribution Bar Chart

The bar chart illustrated in Figure 1 shows the distribution of images across the six scene categories – buildings, forest, glacier, mountain, sea, and street. The distribution is relatively uniform, with each category containing roughly 2,200 to 2,500 images. So we can say the dataset is well balanced, given the even representation, which minimises the risk of model bias toward any single class during training. For reliable generalisation and accuracy metrics that reflect genuine classification ability instead of class frequency dominance, a balanced dataset like this one is essential (Muller & Guido, 2016; OpenAI, 2025).

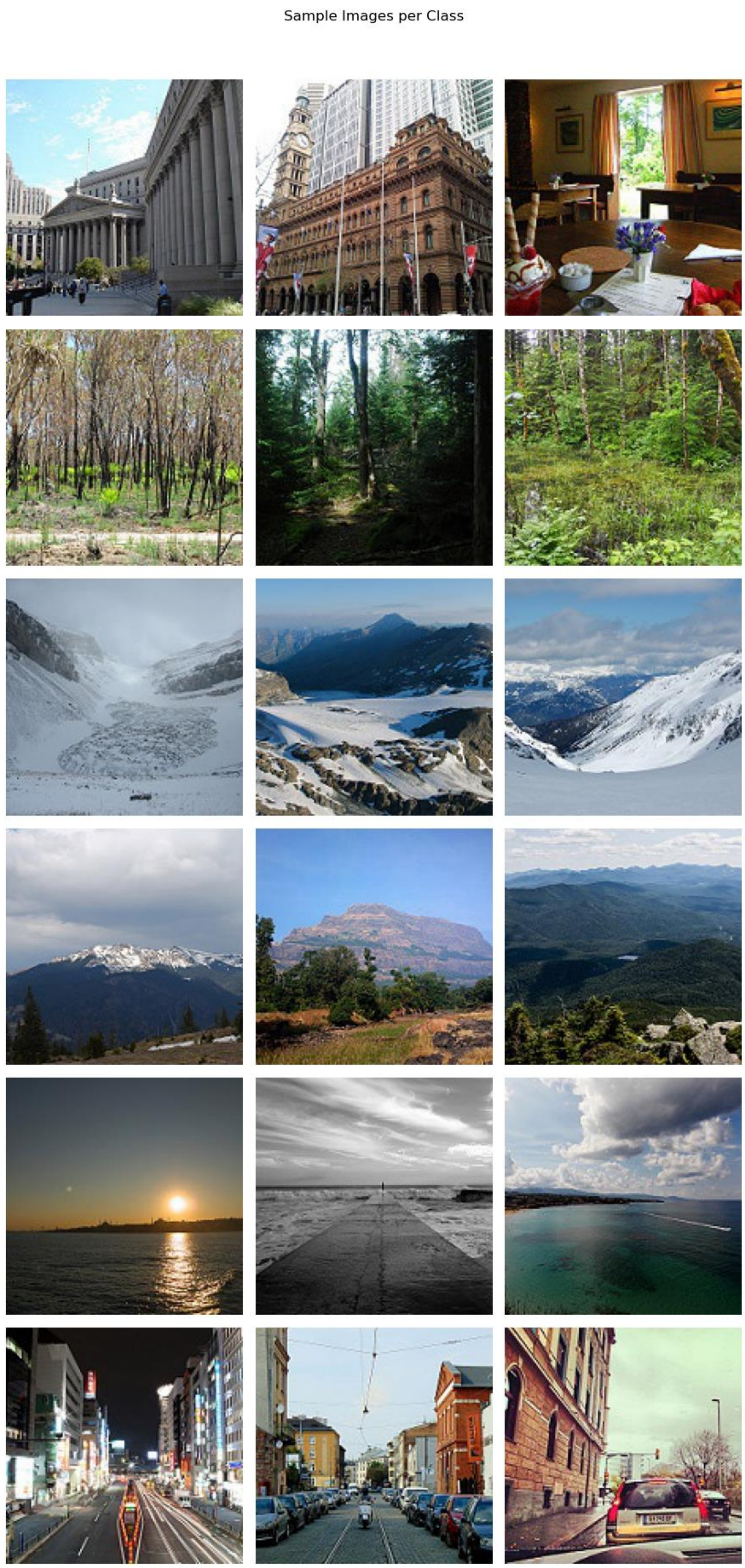

Figure 2: Sample Images per Class

Figure 2 above shows a selection of representative images from each of the six scene categories. There are clear visual distinctions between natural and urban environments, like vegetation in forest, snowy landscapes in glacier, and structures in buildings and street scenes. This display confirms the images are correctly labelled and provide sufficient quality for feature extraction. The CNN will be exposed to varying textures, colours, and lighting, given the diversity in each category, which supports robust learning (OpenAI, 2025).

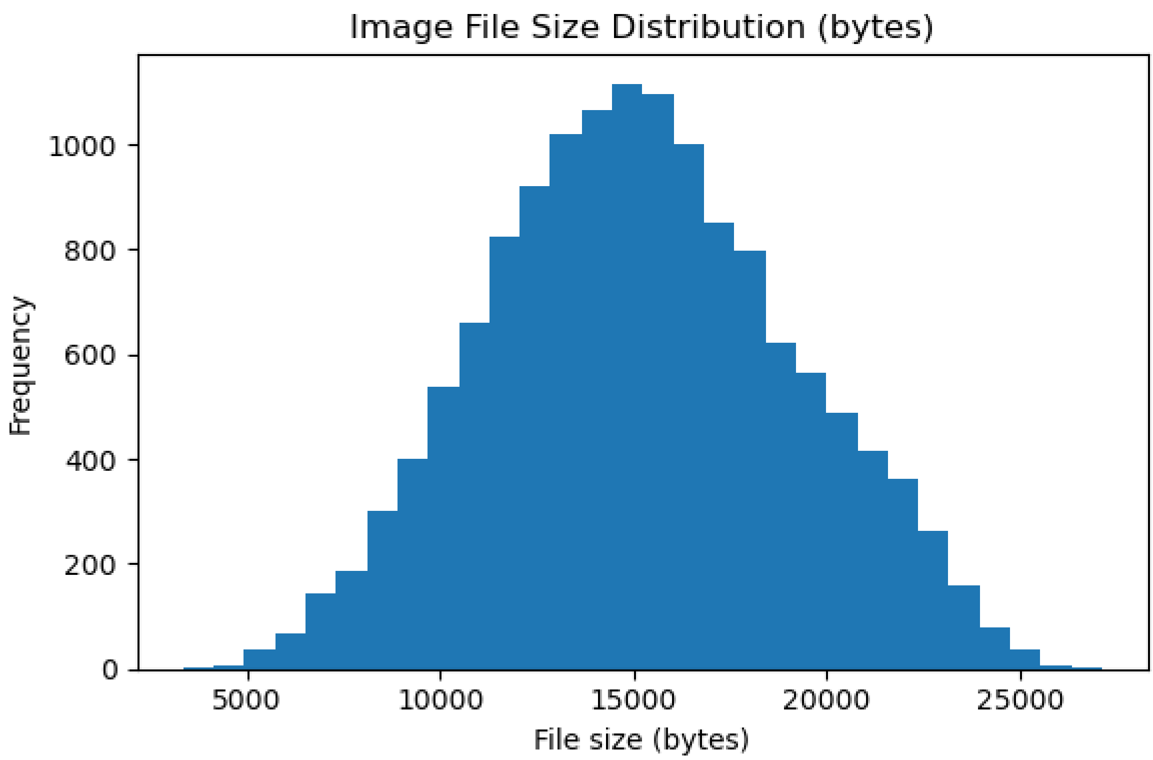

Figure 3: Image File Size Distribution Histogram

The histogram above (Figure 3) shows the distribution of file sizes across the dataset. The distribution shows consistent image resolution and absence of corrupted files, given the near-normal shape centred around roughly 15,000 bytes. We can expect there to be no bias from irregular image dimensions and efficient preprocessing, and training (OpenAI, 2025). We can confirm that the dataset meets the integrity and scale criteria for CNN model training, given these results along with the class balance shown in Figure 1.

Modelling Process

Data Preparation

The Intel Image Classification dataset was structured using manifest files containing verified image paths, sizes, and labels generated by PySpark. After importing the manifest files into TensorFlow, they were converted into optimised tf.data.Dataset pipelines for efficient loading and preprocessing. All images were resized to 150 x 150 pixels, and pixel intensities were normalised to the [0,1] range, and categorical labels were one-hot encoded across the six scene classes: buildings, forest, glacier, mountain, sea, and street (OpenAI, 2025). This meant that during model training, GPU utilisation was consistent, reproducible, and efficient (Muller & Guido, 2016).

Model Architecture

A baseline Convolutional Neural Network (CNN) was designed for multi-class classification. The architecture had stacked Convo2D layers with ReLU activations to extract spatial features, combined with MaxPooling2D layers to decrease dimensionality while keeping key visual patterns (Muller & Guido, 2016; OpenAI, 2025). Feature maps were flattened and passed through fully connected layers, ending with a softmax output layer of six neurons, one for each target class (OpenAI, 2025). Interpretability and performance are balanced with this structure, which allows for the learning of spatial hierarchies in image data (Muller & Guido, 2016).

Model Compilation and Training

Both the Adam optimiser and categorical cross-entropy loss were used for model compiling, and accuracy was the primary evaluation metric. Early stopping was used to prevent overfitting as it halted training when validation loss stabilised, and checkpointing saved the best performing weights in the intel_cnn_best.keras file. The CNN achieved a validation accuracy of 86.5% and was trained for up to 20 epochs.

Learning Outcomes

The training and validation curves showed steady loss reduction and convergence in accuracy, revealing effective learning and strong generalisation (Muller & Guido, 2016; OpenAI, 2025). There was little overfitting seen, and the model complexity was well-calibrated, given the minimal convergence between the two curves (Muller & Guido, 2016). Stable optimisation was confirmed by the consistent validation performance across epochs seen.

Evaluation and Retention

The evaluation metrics revealed balanced predictive capability across all six classes. The highest recall and precision were achieved by the forest (97%) and sea (92%) categories. There was some minor confusion between glacier and mountain, which shows the visual similarity in terrain textures. There was reliable class separation, as shown by the confusion matrix’s strong diagonal dominance (OpenAI, 2025). The model showed high accuracy, stable convergence, and no underfitting and so retraining wasn’t necessary. Only marginal gains would be yielded from further tuning at a significant computational cost (OpenAI, 2025). Ergo, the final model meets the project criteria, offering an accurate, efficient, and interpretable model for six-class scene classification.

Model Evaluation and Interpretation

Overview of Evaluation Metrics

Standard classification metrics were used to assess the model's performance – accuracy, precision, recall, and F1-score. Along with learning curves and a confusion matrix for diagnostic interpretation. The accuracy metric shows the overall proportion of correctly classified images, whilst precision and recall give class-specific insights into prediction quality (OpenAI, 2025). Finally, the F1-score harmonises both, giving a balanced view of false positives and negatives (Muller & Guido, 2016). A comprehensive understanding of model effectiveness is provided by this combination of metrics.

Quantitative Results

Metric	**Value**
Validation Accuracy	0.8648
Macro Precision	0.8712
Macro Recall	0.8644
Macro F1-score	0.8652

Table 1: CNN Performance Metrics

Table 1 above shows the performance metrics for the CNN model. The CNN achieved an overall validation accuracy of 0.8648, showing strong generalisation capability across unseen images. The macro averaged precision, recall, and f1-score confirm balanced predictive performance across all six classes. Specific class level results reveal that forest (97% precision, 97% recall) and sea (92% precision, 84% recall) achieved the highest reliability, showing distinctive and clearly separated visual patterns within scenes (OpenAI, 2025). Glacier (85% precision, 77% recall) and buildings (89% precision, 79% recall) showed moderate performance and partial misclassifications due to subtle texture similarities or lighting variations (OpenAI, 2025). Strong performance was seen from mountain (77% precision, 89% recall) and street (80% precision, 93% recall), showing the model effectively captured landscape features and man-made structures (OpenAI, 2025).

Learning Curve Analysis

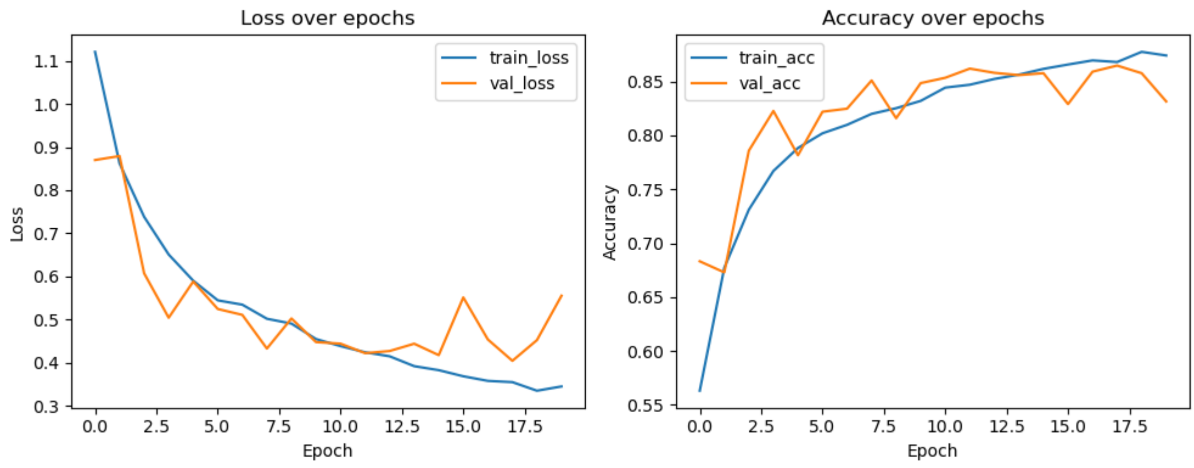

Figure 4: Training and Validation Learning Curves

As seen in Figure 4 above, the training and validation learning curves showed consistent improvement in accuracy and a parallel decline in loss over 20 epochs. There is minimal overfitting and effective regularisation as seen by the close alignment between training and validation trends (Muller & Guido, 2016). After epoch 12, validation accuracy plateaued near 86%, showing convergence to an optimal parameter configuration (OpenAI, 2025). We can deduce efficient learning dynamics and strong model stability given the absence of sharp divergence between loss curves (Muller & Guido, 2016).

Confusion Matrix Interpretation

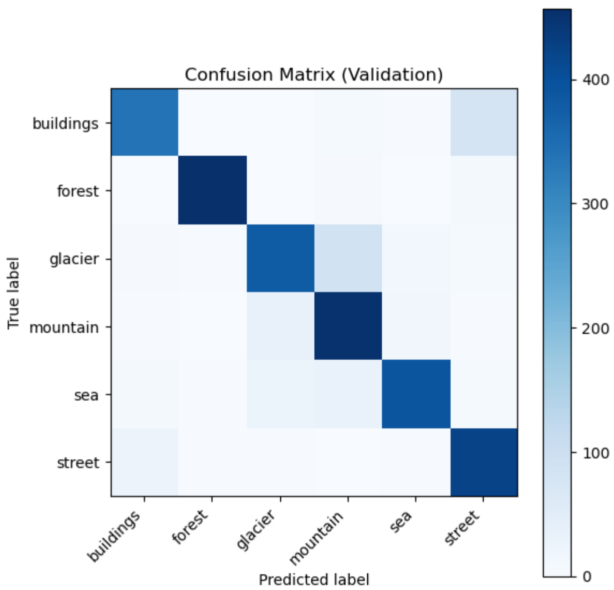

Figure 5: Confusion Matrix

The confusion matrix above (Figure 5) shows strong diagonal dominance, meaning most predictions matched their true labels. There were a few misclassifications, mainly between glacier and mountain, which can be expected given the overlapping visual features like snow, rock, and sky. The model's overall reliability is not diminished by this behaviour, as it is typical in natural-scene classification (OpenAI, 2025). Given the high consistency of predictions across the six classes, it is confirmed that there was robust feature extraction and low inter-class confusion.

Interpretation and Model Suitability

The CNN's performance was of a suitable level for practical deployment as a baseline image classifier, as it successfully generalised from training to validation data. Its strong discriminative power is indicated by its stability and balanced metrics. There was no evidence of underfitting in the learning curves or final metrics, and so retraining was not pursued. At a significantly higher computational cost, further tuning or architectural expansion would likely yield marginal improvements. The final model, therefore, offers an optimal balance within the constraints of this project.

Conclusion & Recommendations

A Convolutional Neural Network (CNN) to classify natural and urban scenes using the Intel Image Classification dataset was successfully developed and evaluated. With systematic data preparation, PySpark enabled exploratory data analysis and TensorFlow-based modelling. The analysis achieved a validation accuracy of 86.5%, confirming the model's ability to effectively generalise across six visually distinct classes: buildings, forest, glacier, mountain, sea, and street.

The CNN effectively captured hierarchical spatial features as highlighted by the results, particularly in classes with distinct textures and colour patterns like forest and sea. Although there were some minor misclassifications between glacier and mountain, given their visual similarities, this was not due to model limitations. The model reached convergence without overfitting as outlined in the learning curve analysis and performance metrics. Retraining was deemed unnecessary given the achieved stability and balance across classes.

Future improvements could focus on data augmentation, like rotations, zooming, and flipping, to enhance model robustness. Also, transfer learning using pretrained architectures like VGG16 or ResNet50 for enhanced feature extraction and accuracy (OpenAI, 2025). Such extensions could potentially lead to performance over 90% accuracy whilst maintaining interpretability and computational efficiency, which aligns with predictive analytical objectives (Muller & Guido, 2016; OpenAI, 2025).

Disclosure of AI Use

Sections: POE.

Name of the tool used: ChatGPT5.

Purpose behind use: Outlines, summaries, Python code, queries, evaluations, suggestions, paraphrasing, and explanations.

Date used: 01/11/2025.

Link to chat: https://chatgpt.com/share/690b4bf3-015c-8004-83d6-c89971dfbe4f

References

Bansal, P., 2018. *Intel Image Classification.* [Online]
Available at: https://www.kaggle.com/datasets/puneet6060/intel-image-classification/code
[Accessed 31 October 2025].

Muller, A. C. & Guido, S., 2016. *Introduction to Machine Learning with Python.* 1st ed. Sebastopol: O'Reilly Media.

OpenAI, 2025. *Open AI ChatGPT5.* [Online]
Available at: https://chatgpt.com/share/690b4bf3-015c-8004-83d6-c89971dfbe4f
[Accessed 01 November 2025].