Image Classification Using VGG19 Transfer Learning

Introduction

This report details the process of building and evaluating an image classification model using transfer learning with the pre-trained VGG19 model. The task involves classifying images into multiple categories related to disaster types. The classes were initially defined as fine-grained but later combined into broader categories to handle class imbalance.

Classes

The original classes were divided into **broad classes** and **fine classes**. Due to class imbalance, the fine classes were combined into their respective broad categories:

Broad Classes:

Damaged_Infrastructure | Fire_Disaster |Human_Damage |Land_Disaster |Non_DamageWater_Disaster

Fine Classes:

- **Damaged_Infrastructure**
  - Damaged_Infrastructure_Earthquake
  - Damaged_Infrastructure_Infrastructure
- **Fire_Disaster**
  - Fire_Disaster_Urban_Fire
  - Fire_Disaster_Wild_Fire
- **Human_Damage**
- **Land_Disaster**
  - Land_Disaster_Drought
  - Land_Disaster_Land_Slide
- **Non_Damage**
  - Non_Damage_Buildings_Street
  - Non_Damage_Wildlife_Forest
  - Non_Damage_Human
  - Non_Damage_Sea
- **Water_Disaster**

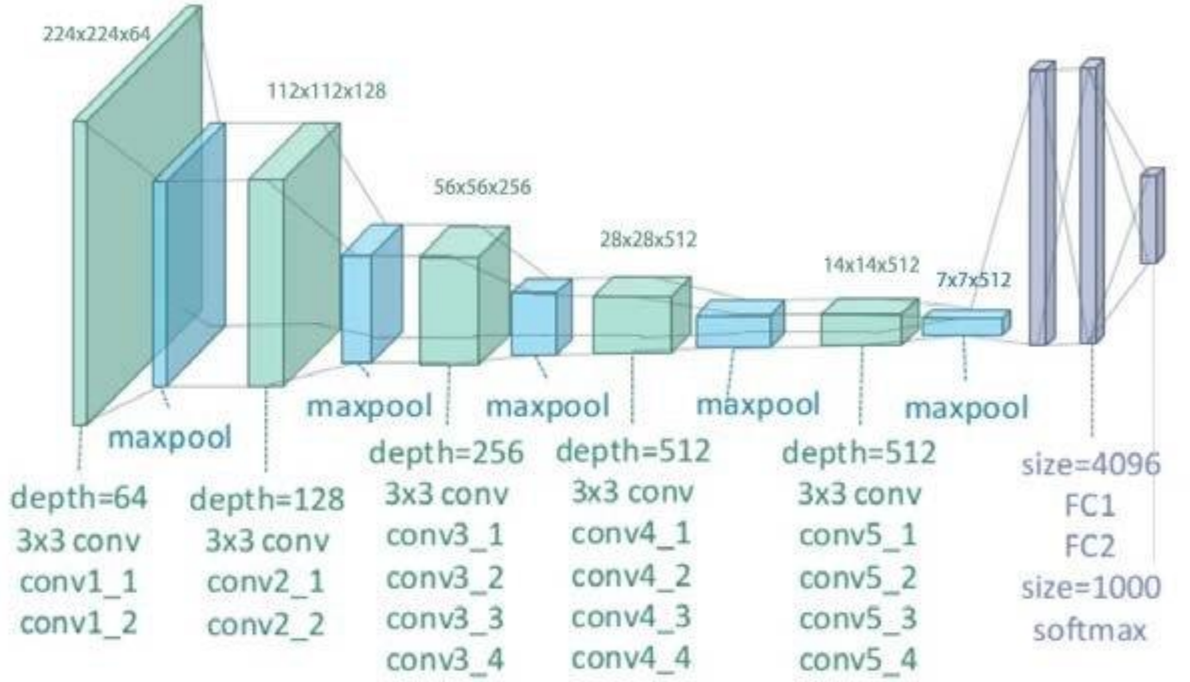
Dataset Preparation and Split

The dataset was split into train, validation, and test sets with a ratio of 70:15:15. The data was shuffled before splitting to ensure randomness. Any corrupted files were removed before training to ensure a clean dataset.

Model Architecture

For image classification, the VGG19 model was employed as a base model. The top layers of VGG19 were excluded, and custom fully connected layers were added:

- **Base model**: VGG19 (pretrained on ImageNet, with frozen layers).
- **Custom layers**: Flatten layer, Dense layer (128 units with ReLU), Output layer with softmax activation for multi-class classification.



Data Augmentation

For training, data augmentation included:

- Rotation (up to 40 degrees)
- Horizontal flipping
- Width/height shifts (up to 20%)
- Shear/zoom transformations
- Rescaling pixel values to [0, 1]

Validation and test sets were only rescaled to maintain original data distribution.

Training Process

- **Loss function**: Categorical cross-entropy
- **Optimizer**: Adam
- **Metrics**: Accuracy

Callbacks used:

- **ModelCheckpoint**
- **EarlyStopping**
- **ReduceLROnPlateau**
- **LearningRateScheduler**
- **CSVLogger**
- **TensorBoard**
- **TerminateOnNaN**

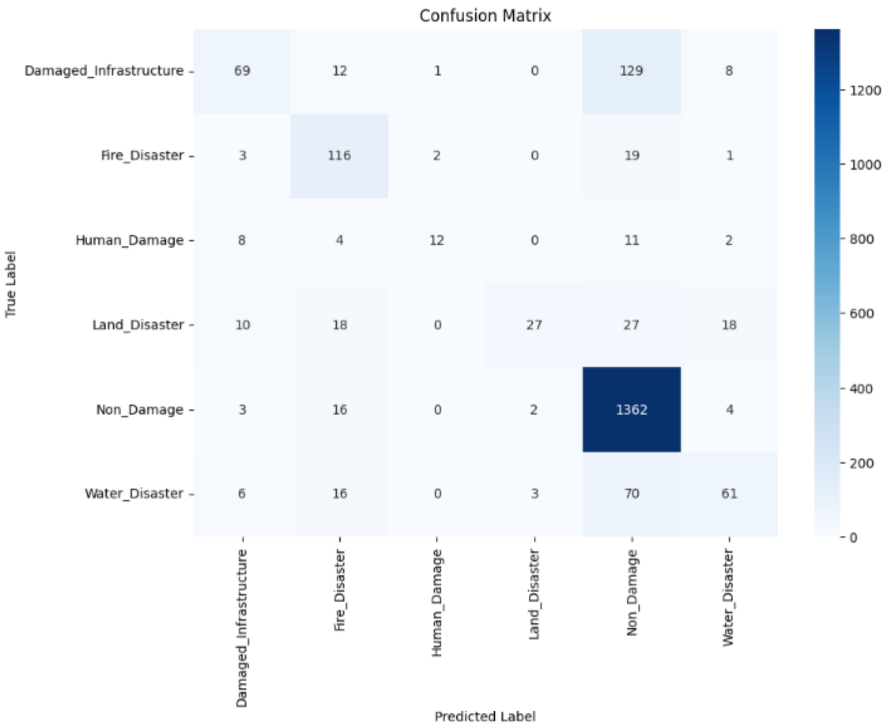
The model was trained for 20 epochs with real-time monitoring and logging.

Model Evaluation

The model was evaluated on the test set, with the following metrics:

- **Accuracy**: 0.81
- **Precision (macro-average)**: 0.74
- **Recall (macro-average)**: 0.52
- **F1-Score (macro-average)**: 0.57
- **Specificity (True Negative Rate)**: 0.85

Confusion Matrix:



Classification Report:

	precision	recall	f1-score	support
0	0.70	0.32	0.43	219
1	0.64	0.82	0.72	141
2	0.80	0.32	0.46	37
3	0.84	0.27	0.41	100
4	0.84	0.98	0.91	1387
5	0.65	0.39	0.49	156
accuracy			0.81	2040
macro avg	0.74	0.52	0.57	2040
weighted avg	0.80	0.81	0.78	2040

Analysis of Evaluation Metrics

- **Accuracy (0.81)**: The model demonstrated a strong overall accuracy, indicating that a high proportion of images were correctly classified. However, this metric alone may not fully reflect the model's performance due to class imbalance.
- **Precision**: The macro-average precision of 0.74 suggests that the model generally performed well in minimizing false positives. However, individual class precision varied significantly, particularly for classes 0 and 5, which had lower precision values (0.70 and 0.65, respectively). This indicates that while the model is good at identifying certain classes, it struggles with others.
- **Recall (0.52)**: The relatively low recall indicates that the model missed many actual positive instances, particularly in class 3 (Human Damage), which had a recall of only 0.27. This suggests the need for further training or tuning to improve the model's sensitivity to underrepresented classes.
- **F1-Score (0.57)**: The macro-average F1-Score of 0.57 reinforces the challenges faced by the model, balancing precision and recall across all classes. While some classes (like 4, Non-Damage) achieved a high F1-Score (0.91), others did not perform as well, indicating potential areas for improvement.
- **Specificity (0.85)**: A high specificity score suggests that the model effectively identifies negative instances. This can be advantageous for real-world applications where the cost of false positives is significant.

Conclusion

This report details the development of an image classification model using VGG19 transfer learning to categorize disaster-related images. While the model achieved an accuracy of 0.81, challenges related to class imbalance were evident, particularly in recall and F1-Score for certain classes. Future improvements, such as hyperparameter tuning, class balancing techniques, and ensemble learning, will be essential to enhance the model's performance and sensitivity to underrepresented categories. Overall, this project demonstrates the potential of transfer learning in effectively addressing image classification tasks in disaster management.

Future Improvements

- **Hyperparameter Tuning:** Refining learning rates, batch sizes, and neuron counts in dense layers can help improve performance.
- **Class Balancing:** Techniques such as class weighting or oversampling the minority classes can be employed to enhance the model's sensitivity to underrepresented classes.
- **Ensemble Learning:** Combining different models like ResNet and EfficientNet could lead to more robust generalization.
- **Data Augmentation:** Increasing the diversity of training data through augmentation techniques can help mitigate class imbalance and improve model robustness.
- **Collect More Data:** Acquiring additional data for underrepresented classes can significantly enhance the model's ability to learn and generalize better across all categories.