SCENIC IMAGE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

NAGA CHANDRA MOULI THAMMINENI- 5611158

RAJA RAJESHWARI PORUMALLA- 5609059

VIDEO LINK: VIDEO LINK

Abstract

- In this project, we developed a deep learning-based image classification model to categorize scenic images into six classes: mountain, street, glacier, buildings, sea, and forest. We explored various techniques, including a basic Convolutional Neural Network (CNN), feature extraction using VGG16, ensemble of neural networks followed by fine tuning image net. Our experiments showed that the ensemble model achieved the best performance, with an accuracy of 87.4% on the test set.

Datasets

- In this project, we are using data from Intel and Google, which contains around 25k images of size 150x150 distributed, which were further divided into six categories, such as buildings, forests, glaciers, mountains, sea, and streets. All the train, test, and prediction data are separated in each zip file. There are around 6k images in the training dataset, 3k in the test, and 3k in the prediction. The data can be accessed through a Python package.

INTRODUCTION

Image classification is a fundamental task in computer vision, with applications in various domains, such as autonomous driving, medical imaging, and surveillance. In this project, we focused on the problem of classifying scenic images into six categories: mountain, street, glacier, buildings, sea, and forest. Accurate classification of such images can have practical applications in areas like tourism, urban planning, and environmental monitoring.

The task of scenic image classification poses several challenges, including the diversity of visual features within each category, the presence of complex backgrounds, and the potential for overlapping characteristics between certain classes (e.g., buildings and streets). To address these challenges, we explored the use of CNN techniques, which have demonstrated remarkable performance in various image recognition tasks.

Convolutional Neural Network (CNN) Model:

- The CNN model was the first deep learning approach explored in the project for the scenic image classification task.
- The CNN model consisted of the following layers:
- Two convolutional layers with 32 filters, 3x3 kernel size, and ReLU activation
- Two max-pooling layers with 2x2 pool size
- A flattening layer
- Two dense layers with 128 units and 6 units (corresponding to the number of classes), respectively, with ReLU and softmax activation functions.

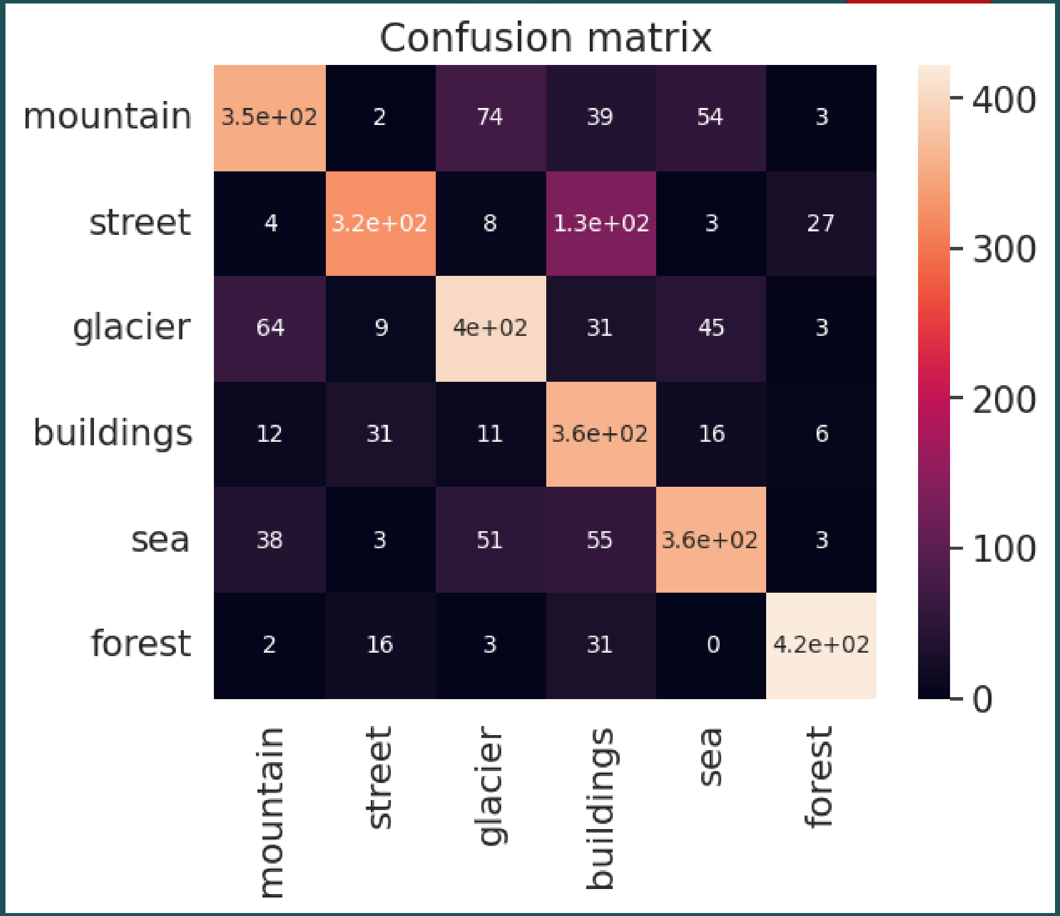
- Here, defining a CNN model for classification task, where the input images are expected to have dimensions of 150x150 pixels with 3 color channels (RGB), and the model outputs the probability distribution over 6 classes.

Steps Involved:

- Build the model,
- Compile the model,
- Train / fit the data to the model,
- Evaluate the model on the testing set,
- Perform an error analysis of our model.
- To Build the model composed of different layers such as: Conv2D, MaxPooling2D, Flatten, ReLu, and Softmax.
- The CNN model was compiled with the Adam optimizer and sparse categorical cross-entropy loss function.

CONFUSION MATRIX

- The confusion matrix provided valuable insights into the specific weaknesses of the CNN model, highlighting the areas where the model struggled the most and informing the team's approach to developing more robust solutions for the scenic image classification task.



VGG MODEL

VGG refers to a convolutional neural network (CNN) architecture developed by members of the Visual Geometry Group. The VGG network is widely used for image classification tasks.

The team explored using the pre-trained VGG16 architecture as a feature extractor for the scenic image classification task.

The VGG16 convolutional base was used to extract features from the input images, and these features were then used to train a simple one-layer neural network for the final classification.

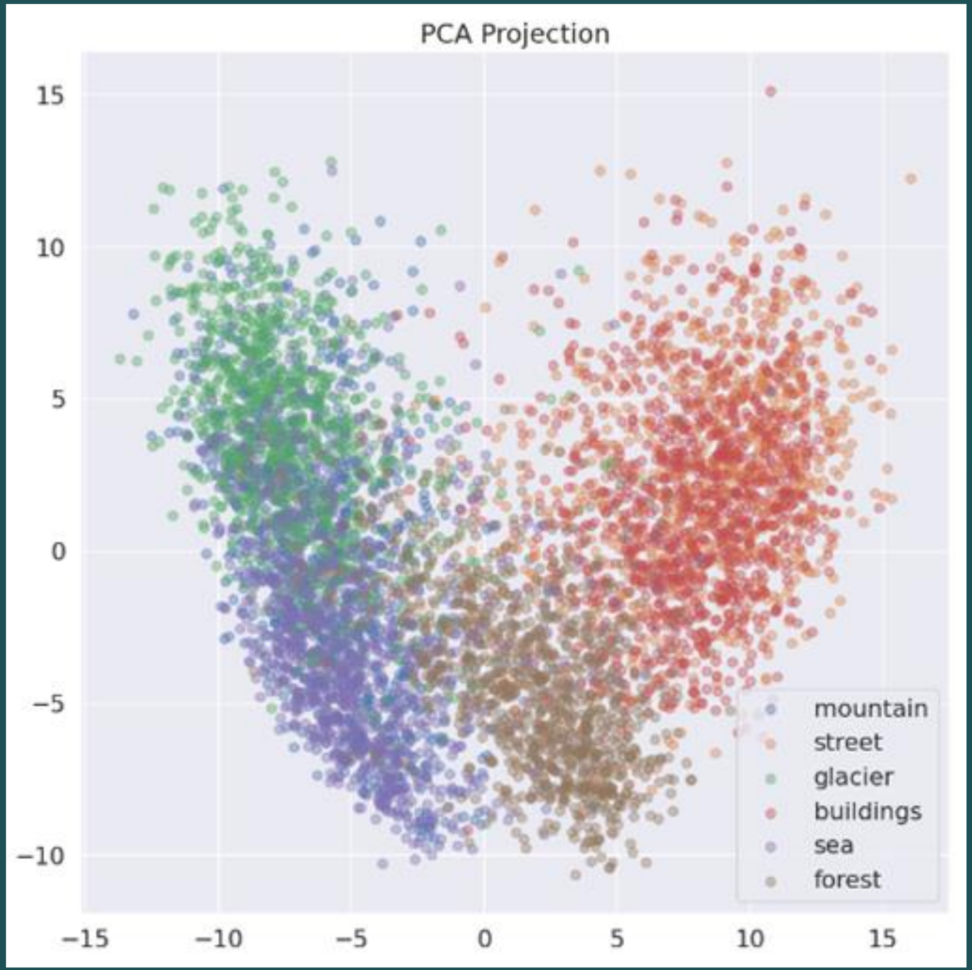
The VGG16-based approach achieved an accuracy of 86.67% on the test set, significantly outperforming the basic CNN model.

We implement the below steps to improve the results on our datasets.

- Feature extraction with VGG16 trained on ImageNet.
- Ensemble models of Neural Networks with the features extracted from VGG.
- Fine Tuning with VGG16 trained on ImageNet.

PCA

- The PCA visualization provided the team with valuable insights into the strengths and weaknesses of the VGG16 features, guiding them towards the development of more sophisticated models to address the remaining challenges in the scenic image classification task.



Training on top of VGG

We will train a simple one-layer Neural Network on the features extracted from VGG

```
model2 = tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape = (x, y, z)),
    tf.keras.layers.Dense(50, activation=tf.nn.relu),
    tf.keras.layers.Dense(6, activation=tf.nn.softmax)
])

model2.compile(optimizer = 'adam', loss =
'sparse_categorical_crossentropy', metrics=['accuracy'])

history2 = model2.fit(train_features, train_labels, batch_size
epochs=15, validation_split = 0.2)
```

Ensemble of Neural Networks

To further improve the classification performance, we implemented an ensemble of 10 neural networks.

Each individual model in the ensemble had a different number of units in the hidden layer, randomly selected between 50 and 100.

The models were trained on random subsets of the training data, containing 80% of the samples.

The final prediction was obtained by averaging the outputs of the ensemble, which achieved an accuracy of 87.4% on the test set.

Fine Tuning VGG ImageNet:

- Fine-tuning is often used when the target dataset is relatively small, as it allows the model to leverage the knowledge learned from the large pre-training dataset.

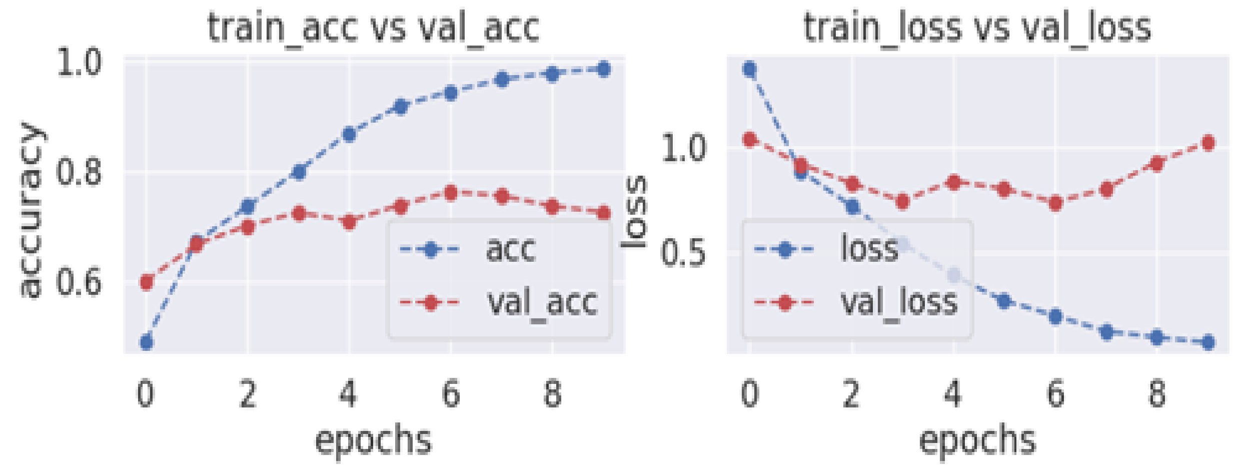
Model: "model_1"

Layer (type)	Output Shape	Param #
input_4 (InputLayer)	[(None, 9, 9, 512)]	0
block5_conv1 (Conv2D)	multiple	2359808
block5_conv2 (Conv2D)	multiple	2359808
block5_conv3 (Conv2D)	multiple	2359808
block5_pool (MaxPooling2D)	multiple	0
conv2d_2 (Conv2D)	(None, 2, 2, 64)	294976
max_pooling2d_2 (MaxPooling2D)	(None, 1, 1, 64)	0
flatten_12 (Flatten)	(None, 64)	0
dense_24 (Dense)	(None, 100)	6500
dense_25 (Dense)	(None, 6)	606

Total params: 7381506 (28.16 MB)
Trainable params: 7381506 (28.16 MB)
Non-trainable params: 0 (0.00 Byte)

RESULTS

ACCURACY FOR CNN MODEL

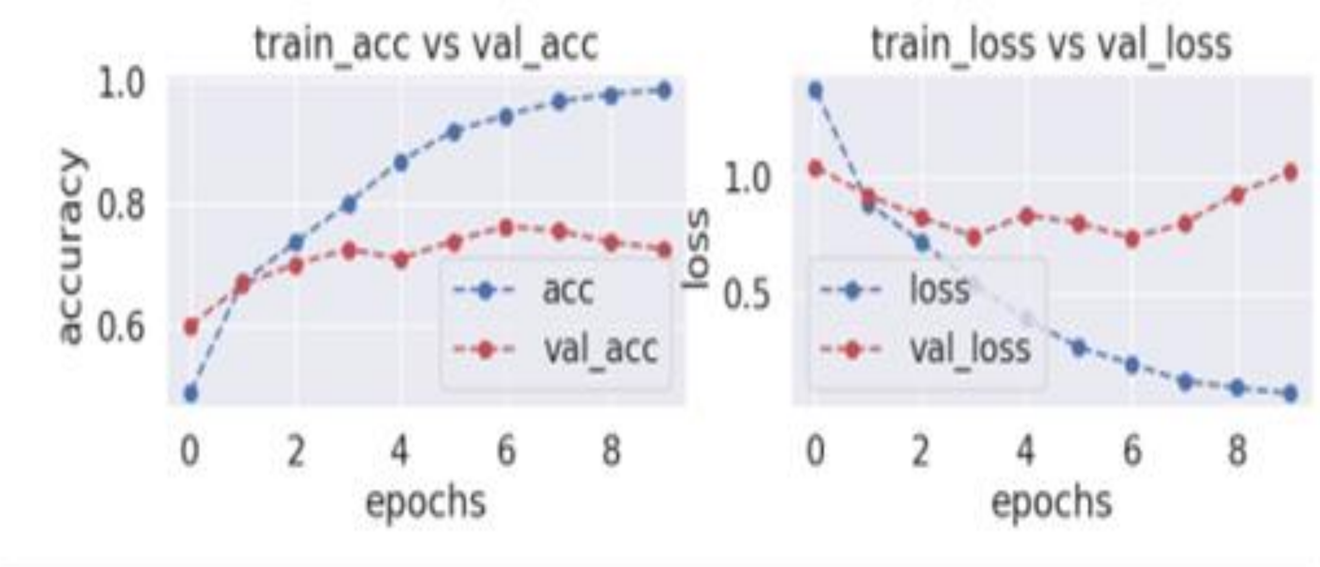


We should evaluate the model performance on test set

```
test_loss = model.evaluate(test_images, test_labels)

94/94 [==============================] - 1s 10ms/step - loss: 1.0048 - accuracy: 0.7407
```

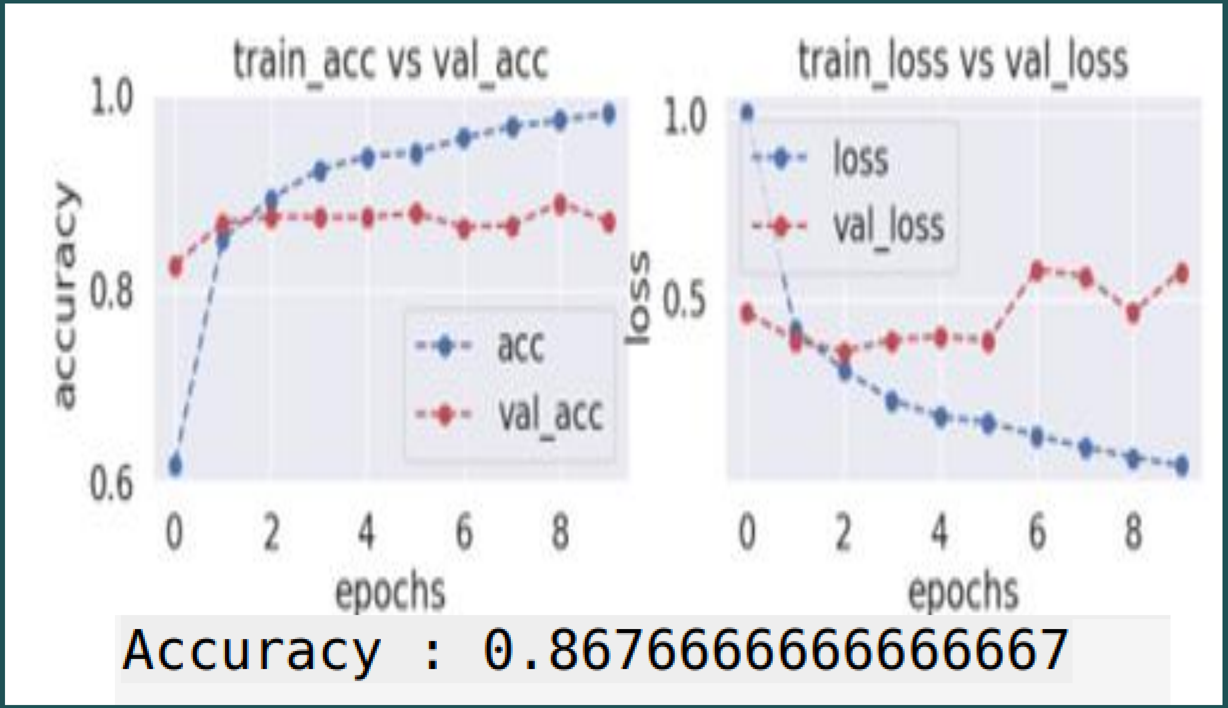
ACCURACY FOR VGG MODEL




```
test_loss = model2.evaluate(test_features, test_labels)

94/94 [==============================] - 0s 3ms/step - loss: 0.4382 -
accuracy: 0.8667
```

ACCURACY FOR FINE TUNING VGG IMAGENET



ACCURACY FOR ENSEMBLE NEURAL NETWORK:

- To further improve the classification performance, we implemented an ensemble of 10 neural networks. The ensemble model achieved the best performance, with an accuracy of 87.4% on the test set, demonstrating the advantages of combining multiple models.

```
from sklearn.metrics import accuracy_score
print("Accuracy : {}".format(accuracy_score(test_labels,
pred_labels)))

Accuracy : 0.874
```

ACCURACY COMPARISON FOR MODELS

Model	Accuracy (Test Set)
CNN	74.07%
VGG16 + One-layer NN	86.67%
Ensemble Neural Network	87.4%

MODEL PERFORMANCE

CNN Model:

- The basic Convolutional Neural Network (CNN) model achieved an accuracy of 74.07% on the test set.

VGG16-based Model:

- The VGG16-based approach, where the extracted features were used to train a simple one-layer neural network, achieved an accuracy of 86.67% on the test set.
- This VGG16-based model significantly outperformed the basic CNN model.

Ensemble of Neural Networks:

- The ensemble of 10 neural networks achieved an accuracy of 87.4% on the test set, outperforming both the basic CNN and the VGG16-based models.

CONCLUSION

- The study compared various deep learning techniques for scenic image classification. It found that a VGG16-based approach, along with an ensemble of 10 neural networks, achieved the highest accuracy of 87.4% on the test set. The analysis revealed challenges in distinguishing certain image classes due to overlapping visual characteristics. The research provides valuable insights and a robust ensemble model for improving scenic image classification tasks, benefiting researchers and practitioners in similar fields.

REFERENCES

- Krizhevsky, A., Sutskever, I.and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.
- Simonyan, K.and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE Conference on computer vision and Pattern Recognition Workshops (pp. 806-813).
- Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are featured in deep neural networks?. Advances in neural information processing systems, 27.
- Dietterich, T. G. (2000). Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1-15). Springer, Berlin, Heidelberg.
- Rokach, L. (2010). Ensemble-based classifiers. Artificial Intelligence Review, 33(1), 1-39.

THANK YOU!!

Video Link: Video Link