DSCI552 Project Report

Group 23: Kayvan Shah, Zuqi Li, Qianyou Wang

April 2023

Abstract

This project addresses the difficulty of landmark image classification by predicting the category and landmark names of given images. The provided dataset is organized into six categories, with five landmarks in each category. Due to the small dataset size, transfer learning utilizing pre-trained models, EfficientNetB0 or VGG16, is explored. To handle the classification problem of landmark categories, a single multi-output classifier is proposed. Multi-task learning technology is also considered to incorporate landmark and category classification into one model. EfficientNetB0 is chosen as the pre-trained model due to its higher Top-1 & Top-3 accuracy than VGG16.[1]

Due to the small size of the dataset, we used data augmentation techniques to expand the data and prevent overfitting. We utilized the Adam optimizer with a learning rate of 0.001 and categorical cross-entropy as the loss function, which is suitable for multi-class classification problems. F1-score was monitored during training as a metric. After fitting the model, we analyzed its behavior by visualizing the learned features and evaluated its performance on other test datasets to ensure its generalization ability.

1 Introduction

Image classification has always been an important topic in machine learning that involves assigning a label to an image based on its content. In this project, we aim to address the difficulty of landmark image classification, where the task is to predict the category and landmark names of given images.

The dataset provided is organized into a two-level hierarchy structure. It is categorized into six categories: Gothic, Modern, Mughal, Neoclassical, Pagodas, and Pyramids, and for each category, there are five landmarks for a total of 30 landmarks, with each landmark having 14 images. The small dataset size makes it difficult to avoid overfitting with deep learning models. Therefore, we should explore transfer learning utilizing a pre-trained model instead of training conditional neural networks from scratch. However, achieving high accuracy on small datasets is challenging, and choosing a suitable pre-trained model is crucial. Due to resource usage limitations, we can only apply EfficientNetB0 or VGG16 to complete the classification task.

Despite the limitations, there are still challenges when classifying the landmark categories. The task involves classifying six categories, and a decision must be made whether to use one 6 binary classifiers. Based on research, a single multi-classifier is proposed for this classification problem.

Firstly, landmark categories often belong to several categories, and there are correlations between them. A single multi-classifier can capture these correlations more accurately and improve classification accuracy. Secondly, the categories of landmark buildings are usually balanced, and the difference in numbers between them could be better, so multiple binary classifiers are unnecessary to deal with the class imbalance problem. Thirdly, landmark categories usually need to consider the overall characteristics, shape, structure, and other aspects of the building. Using a single multi-classifier can process all these features at once and be more efficient in data processing during training and testing. In conclusion, a single multi-classifier is suitable for handling the classification problem of landmark categories, as it can capture correlations, manage class imbalance, and efficiently process data during training and testing.

Another challenge we need to address is whether the landmark classification task can benefit from knowing the output of the category classification task. The dataset has a hierarchical structure where landmarks and categories are strongly associated. Hence, the completion of landmark classification implies the completion of category classification, making the two tasks related. Therefore, we can consider multi-task learning (MTL) technology to incorporate the two tasks into one model for training and optimization. MTL is a machine learning technique that aims to improve model performance by simultaneously learning multiple related tasks. In traditional single-task learning, the model is only trained for a specific job. In MTL, the model must learn multiple tasks simultaneously, and there is some association or dependency between tasks.[4]

Before constructing the model, we must choose the pre-trained model between EfficientNetB0 and VGG16. According to research, EfficientNetB0 requires more computing resources than VGG16 because it has more network layers and parameters. However, the accuracy of EfficientNetB0 is higher than that of VGG16 based on the ImageNet Benchmark. This is because EfficientNetB0 adopts a method based on network scaling, which can improve the accuracy and generalization ability of the model while reducing model parameters and computation. At the same time, EfficientNetB0 also uses some effective techniques to optimize the model, such as using the Swish activation function and adaptive learning rate. Therefore, EfficientNetB0 is also expected to have higher accuracy than VGG16 on our given dataset.

Based on the challenges we face and the discussion above, we will develop and evaluate transfer learning models with EfficientNetB0 and investigate the effect of different hyperparameters on model performance, such as learning rate, batch size, and optimizer. Furthermore, we will analyze the model's behavior by visualizing the learned features and identifying which parts of the images the model focuses on for classification.

2 Model Building

Initially, we process the raw data of 420 landmarks by importing it into a data frame and creating categorical labels for each category and landmark. We then use stratified split to divide the dataset into the train, validation, and test sets with an equal number of images per label. First, we allocate 10% of the data to the test set, followed by a split of 20% of the remaining data for the validation set, resulting in 302 samples for training, 76 for validation, and 42 for testing.

Subsequently, we convert these data frames into TensorFlow datasets for efficient preprocessing and integration into a training pipeline with better memory utilization. TensorFlow datasets also offer a variety of pre-processing functions, such as shuffling, batching, and mapping, which can be applied to the data on the fly during training. Additionally, the datasets are optimized for performance, making it faster to iterate over the data, and can be cached in memory for faster retrieval during training.

The model is built using the EfficientNetB0 pre-trained model and fine-tuned for category and landmark classification tasks. The input layer is defined with the input shape of the images, which is then passed through an image augmentation layer. The EfficientNetB0 model is then loaded as a Keras layer with pre-trained weights and fine-tuned. Two separate branches are created for category and landmark classification, each having a dense layer with a swish activation function, batch normalization, dropout, and a dense layer with a softmax

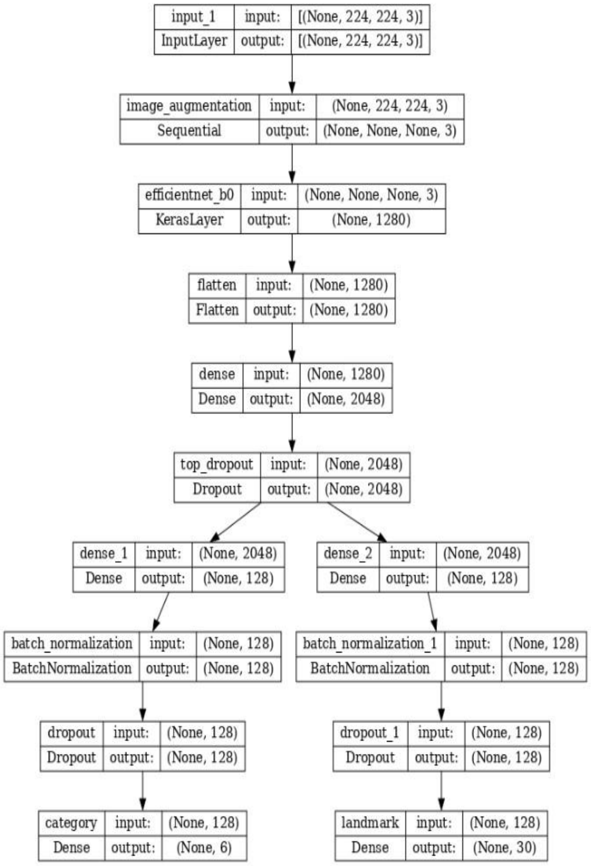
input_1 | input: | [(None, 224, 224, 3)]
InputLayer | output: | [(None, 224, 224, 3)]
image_augmentation | input: | (None, 224, 224, 3)
Sequential | output: | (None, None, None, 3)
efficientnet_b0 | input: | (None, None, None, 3)
KerasLayer | output: | (None, 1280)
flatten | input: | (None, 1280)
Flatten | output: | (None, 1280)
dense | input: | (None, 1280)
Dense | output: | (None, 2048)
top_dropout | input: | (None, 2048)
Dropout | output: | (None, 2048)
dense_1 | input: | (None, 2048)
Dense | output: | (None, 128)
dense_2 | input: | (None, 2048)
Dense | output: | (None, 128)
batch_normalization | input: | (None, 128)
BatchNormalization | output: | (None, 128)
batch_normalization_1 | input: | (None, 128)
BatchNormalization | output: | (None, 128)
dropout | input: | (None, 128)
Dropout | output: | (None, 128)
dropout_1 | input: | (None, 128)
Dropout | output: | (None, 128)
category | input: | (None, 128)
Dense | output: | (None, 6)
landmark | input: | (None, 128)
Dense | output: | (None, 30)

Figure 1: Category & Landmark Classification Model

activation function. The two branches are combined and compiled with a loss function of categorical cross-entropy and metrics of F1 scores. The model architecture can be found in the figure to the right.

3 Results

The results of the model training are shown in Table 1. The model was trained for 65 epochs with early stopping based on validation loss with a patience of 8. The table shows the F1-score and loss for both the category and landmark branches of the model on the training, validation, and test datasets.

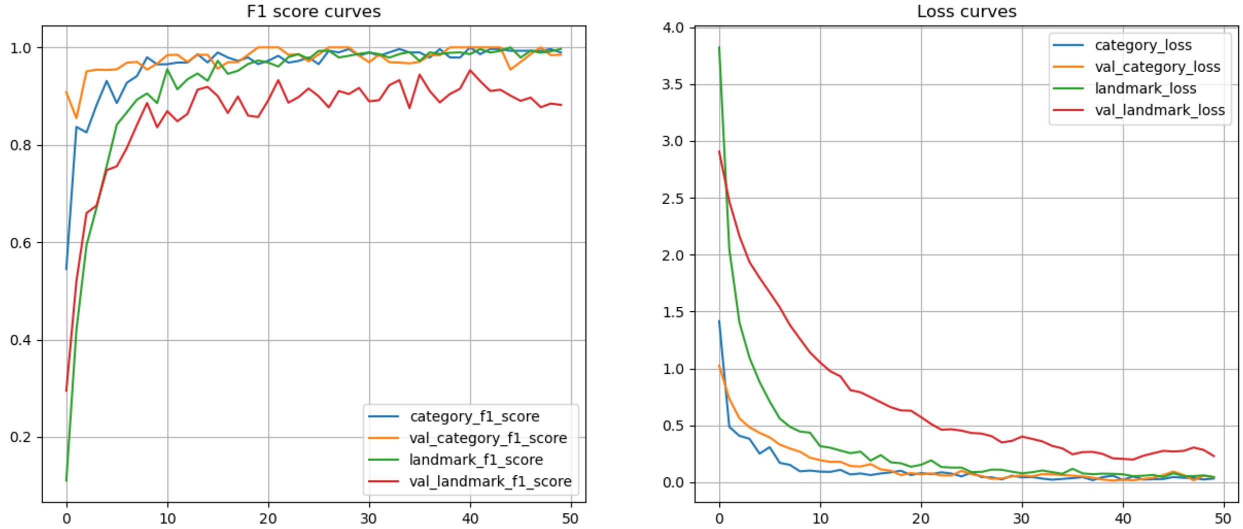

Figure 2: Plot showing learning curves for train and validation datasets.

The training dataset achieved high F1-scores of 0.9895 for category and 0.9968 for landmark, with relatively low losses of 0.0317 and 0.0450, respectively. The validation dataset also had high F1-scores of 0.9848 for category and 0.8819 for landmark, with slightly higher losses of 0.0409 and 0.2294, respectively.

Table 1: F1 Score & Loss for Category & Landmark Classification on Train, Validation and Test Dataset

	Category		Landmark	
	F1-score	Loss	F1-score	Loss
Training	0.9895	0.0317	0.9968	0.0450
Validation	0.9848	0.0409	0.8819	0.2294
Test	0.9515	0.1090	0.9600	0.2791

The test dataset achieved F1-scores of 0.9515 for category and 0.9600 for landmark, with higher losses of 0.1090 and 0.2791, respectively. These results suggest that the model has good performance on both the training and validation datasets, but slightly lower performance on the test dataset, which may indicate some overfitting to the training data.

Table 2: Average Accuracy on Test Dataset

	Category	Landmark
Test Avg. Accuracy	0.95	0.95

Table 2 summarizes the average accuracy results of the model on the test dataset for both category and landmark classification. The model achieved an average accuracy of 0.95 for both category and landmark classification tasks.

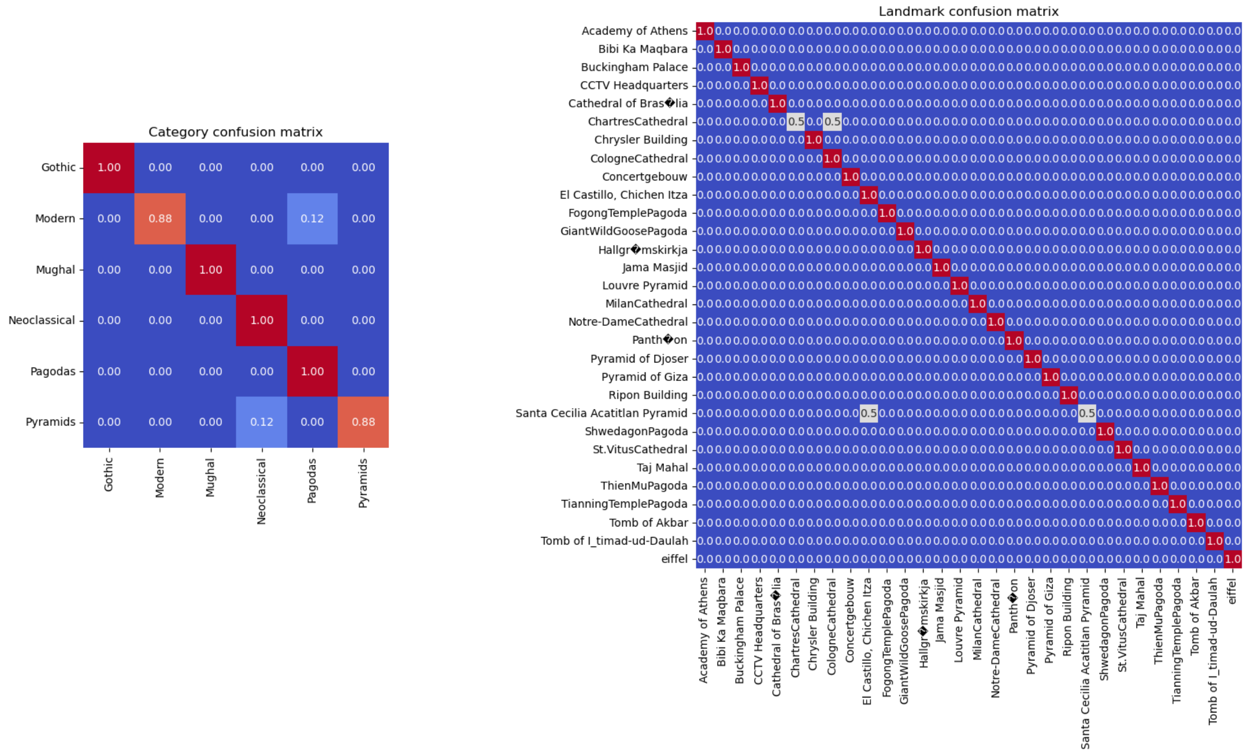

Figure 3: Confusion Matrix for Category & Landmark Classification

Additionally, the confusion matrix shows that the model has distinguished between the majority of the categories and landmarks with reasonable accuracy, although there are some errors in cases where two landmarks or categories share similar features.

4 Ablation Studies

The ablation study is a technique to analyze the effectiveness of each component in a machine learning model by removing or disabling one or more components and observing the impact on performance metrics. In this study, each row of the table, a component that was missing in the previous row is added, and the performance of the model is evaluated after adding those components: pre-trained weights, data augmentation, fine-tuning, and hyperparameter tuning.

From the results, we can see that all components play an important role in improving the model's performance. Without pre-trained weights, the model's performance decreases significantly, indicating that pre-trained weights help the model to learn better features. Data augmentation also helps to improve the model's performance by preventing overfitting. Fine-tuning and hyperparameter tuning also have a positive impact on the model's performance. The highest performance is achieved when all components are used together.

Table 3: Results of the ablation study for the classification of categories and landmarks on Test dataset

	Category			Landmark		
Model	F1 Score	Accuracy	Loss	F1 Score	Accuracy	Loss
Without pre-trained weights	0.9002	0.9614	0.4435	0.8351	0.8106	0.6451
Without data augmentation	0.9368	0.9315	0.5147	0.5282	0.4675	1.6445
Without finetuning	0.9108	0.9000	0.3448	0.6336	0.6900	1.2431
Without hyperparameter tuning	0.9262	0.7400	0.4218	0.7333	0.9300	1.7124
With all	0.9515	0.9500	0.1090	0.9600	0.9500	0.2791

5 Future Scope

The future scope of the project includes the possibility of extending the dataset by incorporating the Google Landmarks v2 dataset, which contains a much larger number of images and landmark categories. This would provide a more diverse set of images for training the model, potentially improving its accuracy and ability to recognize a wider range of landmarks. Additionally, there is scope for exploring other techniques for data augmentation and transfer learning, as well as experimenting with different neural network architectures and hyperparameters to further improve the model's performance.

In addition to extending the dataset using Google Landmarks v2, another future scope of the project could be to explore other deep learning models such as Convolutional Neural Networks (CNNs) and Residual Networks (ResNets) to see if they can achieve better performance compared to the current model. Moreover, the project could also be extended to include additional features such as image segmentation and object detection to make it more robust and versatile in handling diverse real-world scenarios.

6 Conclusion

In conclusion, we have presented a deep learning approach to classify landmarks using convolutional neural networks. We have performed various experiments and conducted an ablation study to evaluate the impact of different components in our model. Our results demonstrate that the use of pre-trained weights, data augmentation, fine-tuning, and hyperparameter tuning significantly improves the performance of our model in terms of accuracy, F1 score, and loss.

Moreover, we have discussed the limitations of our approach and proposed future directions for improvement. One such direction includes extending our dataset using Google Landmarks v2 data to increase the diversity and number of landmark categories in our model. Another possible direction is to explore the use of advanced architectures such as attention-based models and graph convolutional networks to capture the spatial relationships and dependencies between different parts of the image.

Overall, our study contributes to the field of computer vision and has practical applications in the field of tourism, cultural heritage, and geographical information systems. We hope that our work will inspire further research and development in this area and lead to the creation of more accurate and robust landmark recognition systems.

References

[1] Benchmark, I. (2022). Papers with code. *Accedido el*, 2(04):2022.

[2] Francois Chollet (2022). Basic classification: Classify images of clothing. https://www.tensorflow.org/tutorials/keras/classification. Last Accessed on Apr 24, 2023.

[3] Perez, L. and Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

[4] Savchenko, A. V. (2021). Facial expression and attributes recognition based on multi-task learning of lightweight neural networks. In *2021 IEEE 19th International Symposium on Intelligent Systems and Informatics (SISY)*, pages 119–124. IEEE.

[5] Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.