2020 Fall COMP4471 Project Final Report
Google Landmark Recognition

LAM Man Hei
HKUST
mhlamaf@ust.hk

LIAO Yi Han
HKUST
yliaoag@ust.hk

Abstract

Deep convolution neural network has already gotten remarkable results in image classification tasks in recent years. Therefore, in this project, we will attempt to construct a landmark recognition model with deep convolution neural network to recognize the specific landmark in an image. ResNet[5] and EfficientNet[9] with label-distribution-aware margin loss [3], which is used to mitigate to effect of imbalanced dataset, have been tested and evaluated. Also, multiple approaches are applied, including transfer learning and mixed-precision training, in order to reduce the training and inference time. From the experiment result, although both models perform similarly, we decided EfficientNet is a better recognition model over ResNet due to parameter size difference.

1. Introduction

Landmark recognition can be very useful in our daily life. For instance, Google Image Search allows users to search similar images by uploading an image instead of normal text search, but it can only search images with similar style or context. The search engine cannot recognize the landmark in the image and then provide similar images from the same landmark. This also happens in photo management service, such as Google Photo, which can only identify general objects in photos, but not the specific landmark in the photo. If the service can recognize the landmark in the uploaded photos, users can better manage their photo collection by categorizing their photos by locations. It is especially helpful for photos without location information.

In this project, we are going to investigate landmark recognition with deep convolution neural network. We will attempt to use ResNet[5] and EfficientNet[9] to construct a landmark recognition model. The expected output of the model is a column vector of the probability of each label. The label with the highest probability would be the predicted label of that image. Both model performance will be evaluated with some basic metrics, such as accuracy, F1-score and confusion matrix, and the best model will be elected as the final landmark recognition model.

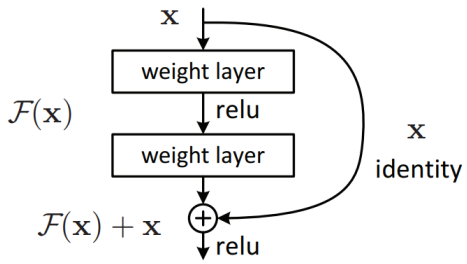

Figure 1. ResNet Building Block [5]

After experiments, ResNet and EfficientNet both obtain similar performance with the accuracy of 67.70% and 68.19%, and average F1 score of 0.60 and 0.61 respectively. The majority classes mostly achieve great performance, but many minority classes do not perform well.

2. Related Work

ResNet. ResNet [5] is a image classification model first proposed in 2015, and won the first prize of ILSVRC 2015 [8]. ResNet is built with a CNN architecture shown in figure 1, which is based on the concept of residual learning. Shortcut connections are added in between convolution layers in the ResNet building block. This architecture resolves the degradation problem of very deep convolution neural network [5], and thus ResNet can achieve high accuracy by stacking over 100 convolution layers.

EfficientNet. EfficientNet [9] provide a new model scaling method for CNN architectures which uses simple yet effective compound coefficient to scale up CNNs. It uniformly scales each dimension (such as width, depth, and resolution) with a fixed set of scaling coefficients. The baseline network of EfficientNet uses mobile inverted bottleneck convolution (MBConv) shown in figure 2. EfficientNet model can achieve high accuracy and better efficiency

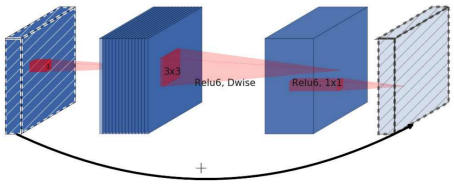

Figure 2. EfficientNet Mobile Inverted Bottleneck Conv [9]

over other CNN models with reduced parameter size.

Dataset Resampling. Data imbalance is a common issue for the real-world dataset, yet deep neural network performs very bad in imbalanced dataset [2] and have difficulties in learning representation of minority class. Thus, resampling is used to convert an imbalanced dataset into a balanced dataset. Random oversampling and undersampling are two of the common resampling techniques. However, random oversampling can lead to overfitting problem [4] since the algorithm just duplicates the samples of minority classes. SMOTE [4] is a more advanced oversampling algorithm to overcome the overfitting issue, which creates new samples by interpolating neighboring samples. Although undersampling is preferable to oversampling in some cases, undersampling has a significant drawback that a large portion of majority class samples will be discarded. In general, resampling is not suitable to construct a balanced dataset from a severely imbalanced dataset.

LDAM Loss. Label-distribution-aware margin loss [3] is another method to resolve data imbalance issue, which aims to have the optimal trade-off between margins of each class, which encourages the minority classes to have larger margins like figure 3, as a large margin can be regarded as regularization. It proposes to improve the generalization error of minority classes via regularizing the minority classes more strongly and at the same time keep the model's ability to fit the frequent classes.

3. Data

The dataset used in this project is retrieved from Google Landmark Recognition 2020[1], which is a Kaggle competition. Since the competition is closed and no longer accept submission, the ground truth of the test dataset cannot be retrieved. Thus, only the train dataset will be used in this project. The entire dataset consists of 81313 landmarks classes with 1580470 images.

3.1. Exploratory Data Analysis

In exploratory data analysis (EDA), we inspect the dataset and summarize the following characteristics of the dataset.

[1] www.kaggle.com/c/landmark-recognition-2020

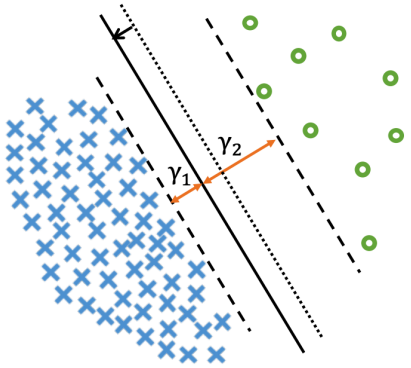

Figure 3. LDAM Customized Margin[]

Figure 4. Frequency graph of each landmark class. The classes are sorted in descending frequency order.

First, the dataset is severely imbalanced. The largest landmark class has a sample size of 6272 while the smallest landmark class only has a sample size of 2. Almost 97.5% of the landmark classes are with sample size under 100. The frequency graph in figure 4 can clearly show that there are very few classes having a large sample size.

Besides, the quality of images in the dataset is not very high. There are some quite misleading images even for human beings to recognize. Also, the features in the images from the same landmark class can be very distinct from our visual inspection. Figure 5 is one of the examples.

3.2. Custom Dataset Creation

We further create three datasets from this Google landmark dataset. The top 12 and top 20 landmark dataset, which include 12 and 20 landmarks classes with the largest sample size in the original dataset, are first created for preliminary testing. The top 12 landmark dataset has 19999 images while the top 20 landmark dataset has 26779 im-

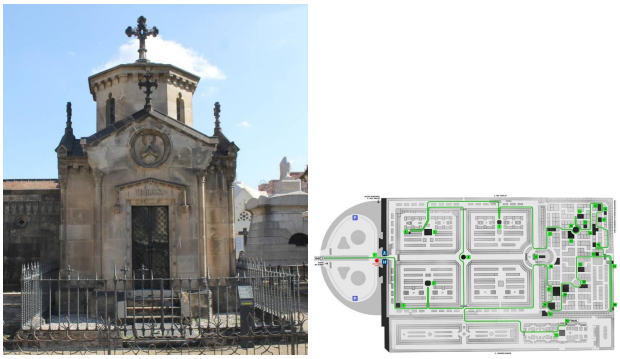

Figure 5. Example images of the same class in the dataset. The left image can clearly see the landmark where the right image is just a floor plan.

ages. Then, a final dataset is created by discarding all landmark classes with a sample size less than 6. The final dataset has 57042 landmarks classes with 1490153 images in total. All of the datasets are split into training, validation, and test set with the ratio of 0.72, 0.18, 0.1 respectively.

4. Methods

4.1. Preprocessing

In pre-processing, the images are first resize to 224 x 224 in order to fit into the models (ResNet and EfficientNet). Then, apply normalization to the image data. To increase the number of data samples for those landmark classes with extremely limited samples, using data augmentation to enlarge their sample size. It also helps reduce the overfitting problem.

4.2. Landmark Recognition Model

In this project, two deep convolution neural networks, ResNet and EfficientNet are tested and evaluated. Since the dataset is very large, it is time-consuming to train a brand new network due to the limited hardware situation. Therefore, transfer learning technique is applied in order to shorten the training time. The final fully connected layer of the models is replaced with a new fully connected layer with the output size the same as the number of landmarks. Then, the model will be trained for a few epochs.

To further increase the speed of training, both models are trained under mixed precision mode. With proper technique, mixed precision training can speed up the training speed significantly without sacrificing the final model accuracy [6], especially with GPU that has tensor core [1]. For instance, fully connected layers and CNN layers can operate under half-precision (float16) while batch normalization layers can operate in single-precision (float32) in order to reduce the effect of accumulated floating point errors. In addition, mixed precision training also reduce memory usage. Therefore, a larger batch size can be used during training compare with single-precision training with the same size of GPU memory, which also increase the training speed slightly.

In the training progress, two loss functions, cross entropy loss and label-distribution-aware margin (LDAM) Loss[3] are tested. Cross entropy loss is a very common loss function used in classification problem. It can be considered as a baseline loss function. However, the dataset may have data imbalance issue, which cannot be solved by cross entropy loss. Using this loss function with a extreme imbalance dataset may cause low precision/recall in minority class. To get a better representation on minor class, LDAM loss can be use. The loss function is formulated as such:

$$\mathcal{L}_{LDAM}((x,y);f) = -\log \frac{e^{z_y - \Delta_y}}{e^{z_y - \Delta_y} + \sum_{j \neq y} e^{z_j}} \tag{1}$$

$$, \Delta_j = \frac{C}{n_j^{1/4}} \textbf{ for } j \in [1,k] \tag{2}$$

where C is a hyperparameter, n_j is the number of sample of j-th class, and z_j is the model prediction output of j-th class. By using LDAM loss, it attempts to increase the margin distance of the minority classes from the decision boundary.

In addition, hyperparameters tuning is required in order to obtain an optimized model. Grid search or Bayesian optimization, which is a more efficient technique, is applied to tune model hyperparameters.

4.3. Evaluation

To evaluate the performance of the deep learning model, accuracy, precision, recall and F1 score would be used.

$$\textbf{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\textbf{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$\textbf{F1} = 2 \times \frac{\textbf{Precision} \times \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}} \tag{5}$$

where TP, TN, FP, FN is the number of true positive, true negative, false positive, false negative prediction of the class respectively. Since the dataset is imbalance, F1 score is consider as a more important evaluation metrics than accuracy. Moreover, confusion matrix would be a visual evaluation metric.

5. Experiments

5.1. Experiment Environment and Implementation

The computer we used for the model training has an 8-core CPU with Nvidia GeForce RTX 2070 Super. The operating system is Ubuntu 18.04 LTS.

Model	Training Acc	Testing Acc
ResNet50	98.02%	96.65%
EfficientNet-B0	97.69%	98.05%

Table 1. Model performance of top 12 landmark classes dataset

All codes are implemented in Python 3.7.9, and the models are implemented with PyTorch [7] along with some common python packages, such as Numpy, Pandas, and scikit-learn. The ResNet50 pretrained model are retrieved from torchvision library, and the EfficientNet-B0 pretrained model are retrieved from a python package EfficientNet_PyTorch[2]. Also, python package Ax are used for hyperparameter optimizaiton.

5.2. Experiment

Before implementing the model on the whole dataset, top 12 and top 20 landmark classes is first selected to do some experiments. Each sub dataset is split into three parts: training, validation, and testing with the ratio of 0.72, 0.18, 0.1. For both datasets, using ResNet50 and EfficientNet-B0 as training models and cross-entropy as loss function.

From the performance of these two sub-datasets(table 1 and table 2), we can observe that as more landmark classes are in the dataset, the accuracy will decrease. The main reason is that the minority class has relatively low precision and recall. As shown in the confusion matrix in figure 6, the images of landmarks with fewer samples are misclassified as some other landmarks with much more sample. This is a common issue in model training with an imbalanced dataset. Therefore, to cope with the dataset with more landmarks and more severe imbalance issues, LDAM is implemented to resolve this issue.

After implementing the LDAM as loss function for the top 20 landmark classes dataset, the model performance has improved as both training and testing accuracy increase as shown in table 2. Therefore, the LDAM loss function will be applied to the later big dataset.

From the original training dataset with 81313 landmark classes, discard those classes with a number of images less than six. After that, there are finally 57042 landmark classes with 1490153 images in total. This dataset is also split into three parts: training, validation, and testing with the ratio of 0.72, 0.18, 0.1. Data augmentations with some random flipping and color jittering are used on the training data in order to mitigate the effect of the imbalanced dataset issue. Because the dataset is quite large (over one million images for training dataset), methods to speed up the training process under limited resources are in need. First of all, resize all the images to store in the size of 224x224. Due to the limited RAM, each batch of data is loaded while training the model instead of loading all the data beforehand. Mixed

[2]https://github.com/lukemelas/EfficientNet-PyTorch

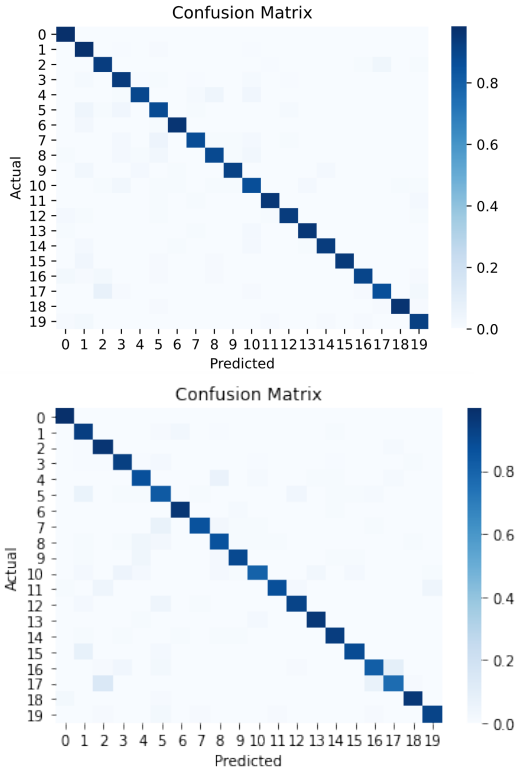

Figure 6. Test set confusion matrix of top-20 landmark classification with ResNet50(top) and EfficientNet-B0(bottom). The larger the landmark id, the fewer the sample that landmark have

precision training is applied in order to reduce the networks' runtime and memory footprint, since some operations, like linear layers and convolutions, can be much faster in float16 (half) datatype. Also, add gradient scaling to prevent gradients with small magnitudes from vanishing under mixed precision training.

The final training results are shown in table 3. The testing performances of the two networks drop a lot as the number of landmark classes is extremely large. Analyzing the f1 scores of each network, both models have over 20000 classes with f1 score over 0.9 and around 15000 classes with f1 score lower than 0.1 as shown in the F1 score histograms in figure 7. Most of the landmark classes with f1 score lower than 0.1 are minority classes with just a few samples in the test set, and hence the support of these f1 score is very small. Having a further look into the f1 scores, we found that there are some landmarks classes with relatively large size datasets getting very low f1 scores. For instance, landmark 177870 has 783 samples for training, however, in both EfficientNet and ResNet model, its f1 scores are 0.215 and 0.3968 respectively. From figure 8, it quite obvious that the architectural styles of these four buildings are very different, which indicates the dataset quality of this class is not good and it seems that the prediction performance is also in-

Model	Training Accuracy	Testing Accuracy	Testing Average F1
ResNet50	96.05%	94%	0.92
ResNet50+LDAM	97.88%	94.18%	0.95
EfficientNet-B0	92.40%	92.72%	0.91
EfficientNet-B0+LDAM	99.48%	98.02%	0.97

Table 2. Model performance of top 20 landmark classes dataset

Model	Training Accuracy	Testing Accuracy	Testing Average F1
ResNet50+LDAM	80.26%	67.70%	0.60
EfficientNet-B0+LDAM	76.98%	68.19%	0.61

Table 3. Model performance of 57042 landmark classes dataset

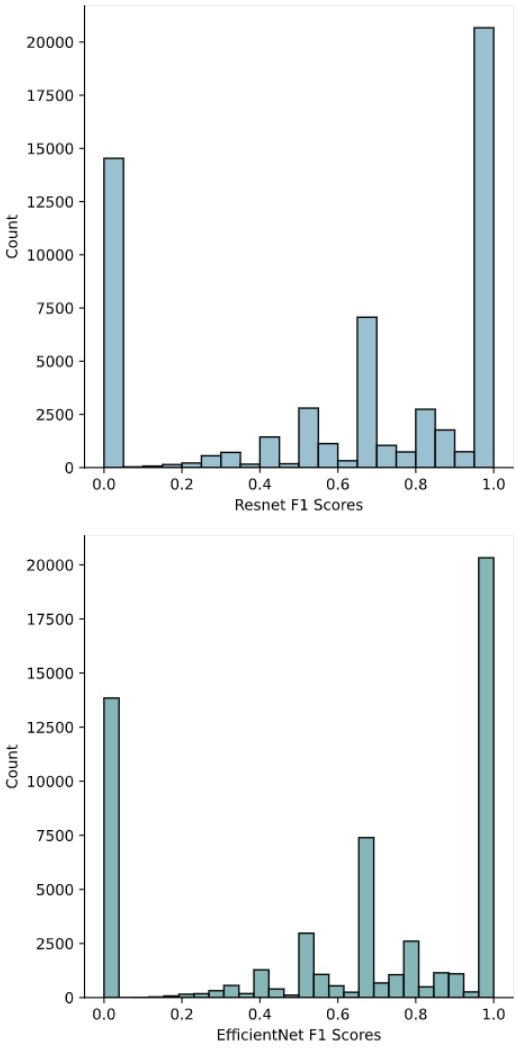

Figure 7. F1 score histogram of ResNet50(top) and EfficientNet-B0(bottom)

fluenced by this low quality of the dataset. As for landmark class 169630 which has only 4 images for training, but the testing f1 scores of both models are 1. From (figure 9), it

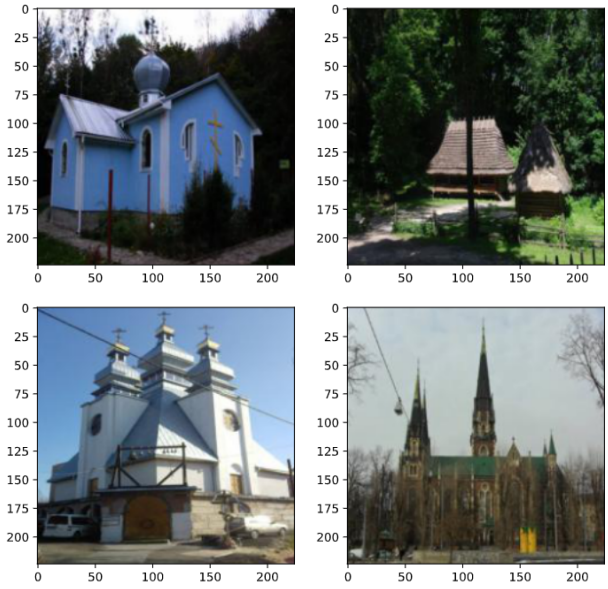

Figure 8. Example images of the class 177870 in the dataset.

can be easily observed that these four images are the same landmark as the feature of this class is presented very well, which can be inferred that the quality of this class samples is quite high.

6. Conclusion

From the experiment result, both ResNet50 and EfficientNet-B0 achieve accuracy of over 65% and average F1 score over 0.50. LDAM loss indeed improves the performance of both models. However, since the dataset is extremely imbalanced, the effect of LDAM loss on the model performance is limited and models still perform poorly in minority classes.

Both models have similar performance in terms of accuracy and F1 score. However, the parameter size of the two models is very different. ResNet50 has over 25 million parameters while EfficientNet-B0 only has about 5 million parameters. EfficientNet-B0 can achieve similar performance

Figure 9. Example images of the class 169630 in the dataset.

with way fewer parameters than ResNet50. Thus, we consider EfficientNet-B0 as the best model for landmark recognition.

References

[1] Training With Mixed Precision :: NVIDIA Deep Learning Performance Documentation. Archive Location: Training.

[2] M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *arXiv:1710.05381 [cs, stat]*, Oct. 2018. arXiv: 1710.05381.

[3] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 1567–1578. Curran Associates, Inc., 2019.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002. arXiv: 1106.1813.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, Dec. 2015. arXiv: 1512.03385.

[6] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu. Mixed Precision Training. Oct. 2017.

[7] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv:1912.01703 [cs, stat]*, Dec. 2019. arXiv: 1912.01703.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[9] M. Tan and Q. V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv:1905.11946 [cs, stat]*, Sept. 2020. arXiv: 1905.11946.