Scene Classification using a Transfer Learning Approach

# Abdullah McDonald

Department of Computer Science, University of the Western Cape, South Africa [3708949@myuwc.ac.za](mailto:3708949@myuwc.ac.za)

***Abstract*—Scene classification means to determine which scene category the contents of an image belongs to. Convolutional Neural Networks, specifically Residual Networks, have proved to be quite usual for the task of image classification. In this paper, we make use of a pre-trained Residual Network to do scene classification. We carry out three experiments in particular; investigate whether data augmentation techniques can improve classification accuracy, whether decreasing the resolution of an image can help prevent object detectors and whether pooling methods like Average Pooling and Max Pooling can improve classification accuracy and also whether they have the ability to preserve spatial information. The results of these experiments showed that data augmentation techniques can improve classification accuracy, decreasing the resolution of an image that does not remove object detectors and that average pooling can improve performance whereas max pooling does not. We also conclude that average pooling can preserve some spatial information on scene images.**

***Index Terms*—Scene Classification, Residual Network, Transfer Learning, Data Augmentation, Image Resolution, Pooling Methods**

1. INTRODUCTION

A scene compromises a place, environment, and more commonly, a variety of objects [1]. Scene classification means to determine which scene category the contents of an image belongs to [2]. For the task of image classification, Convolutional Neural Networks (CNN) have been widely used. Scene classification has many real-world applications.

In robotics, for a robot’s positioning and orientation to be determined, a system for both map construction and positioning is needed. Scene classification plays a vital role in these systems [2]. Scene understanding is an important aspect of self-driving classification [3].

Another application of the scene involves land-use and land cover (LULC). This classification task identifies the land- resource management and urban planning of a specified area [4].

Convolutional Neural Networks have been widely used for image classifications tasks and thus it will be used for the task of scene classification in this paper.

1. PRELIMINARIES
2. *Scene Classification*

Scene classification aims to detect which scene category a specific image belongs to. Fig 1 above shows images of scenes and their respective categories.

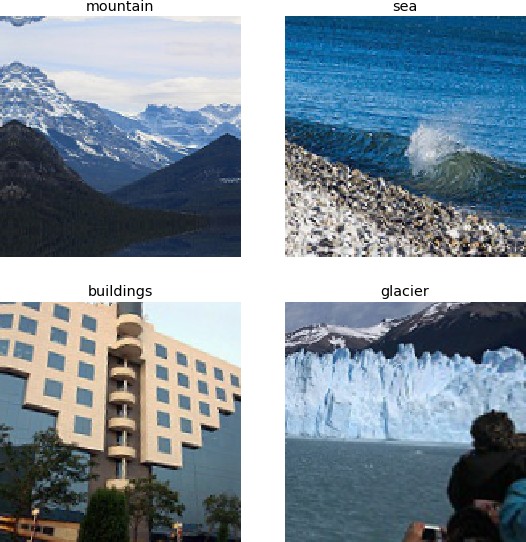


Fig. 1: Scene Categories

Scene classification involves a variety of sub-tasks with object detection and semantic segmentation being some of them. Each sub-task requires a specific amount of attention to achieve optimum performance.

1. *Convolutional Neural Networks*

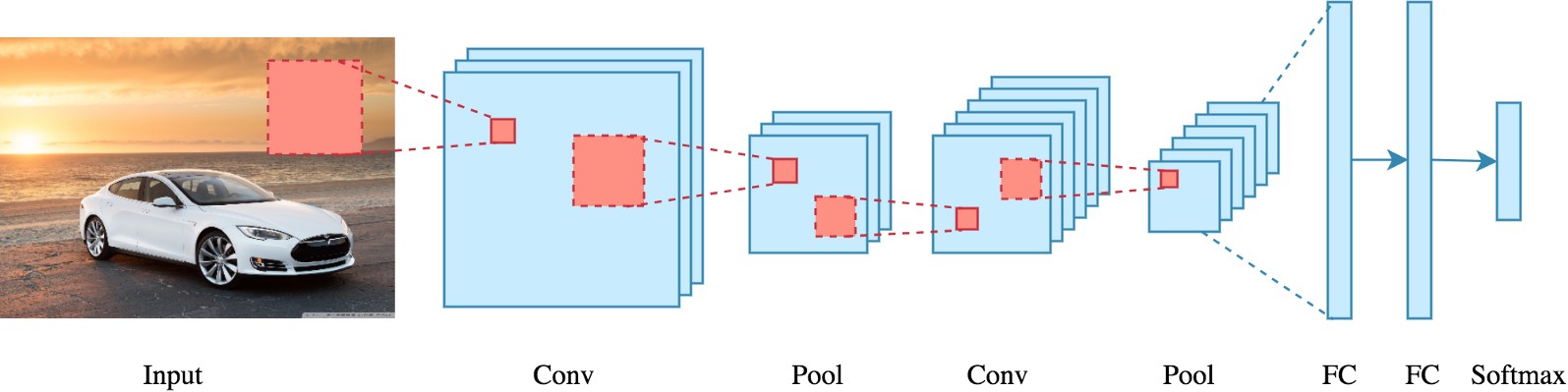


Fig. 2: Basic Architecture of a Convolutional Neural Network

Convolutional Neural Networks were inspired by the multilayer perceptron. It is mainly made up of an input layer, a convolutional layer, and a full connection layer. It might also consist of a pooling layer [5]. The convolutional layer has an advantage over the multi-layer perceptron of reducing the number of features in the overall network. Fig 2 above shows the basic architecture of a convolutional neural network.

The convolution layer takes the input data, convolves it using a specified kernel. The output feature maps of this layer are produced after an activation function is applied [5].

The pooling layer is used to reduce the size of the feature maps and also has the advantage of speeding up computation. Types of pooling methods include Average Pooling, Max Pooling, and Spatial Pooling.

The function of the full connection layer is the same as that in a multi-layer perceptron network where the inputs of the previous are connected to every activation unit is the next layer.

1. *Residual Networks*

When starting with a simple CNN, you would think that by adding more layers to your network, it will perform better. This is not always the case. As CNN's get deeper and deeper, they experience the problem of vanishing gradients. Residual Networks, or ResNets for short, is a type of CNN that effectively solves this problem. ResNets make use of residual blocks.

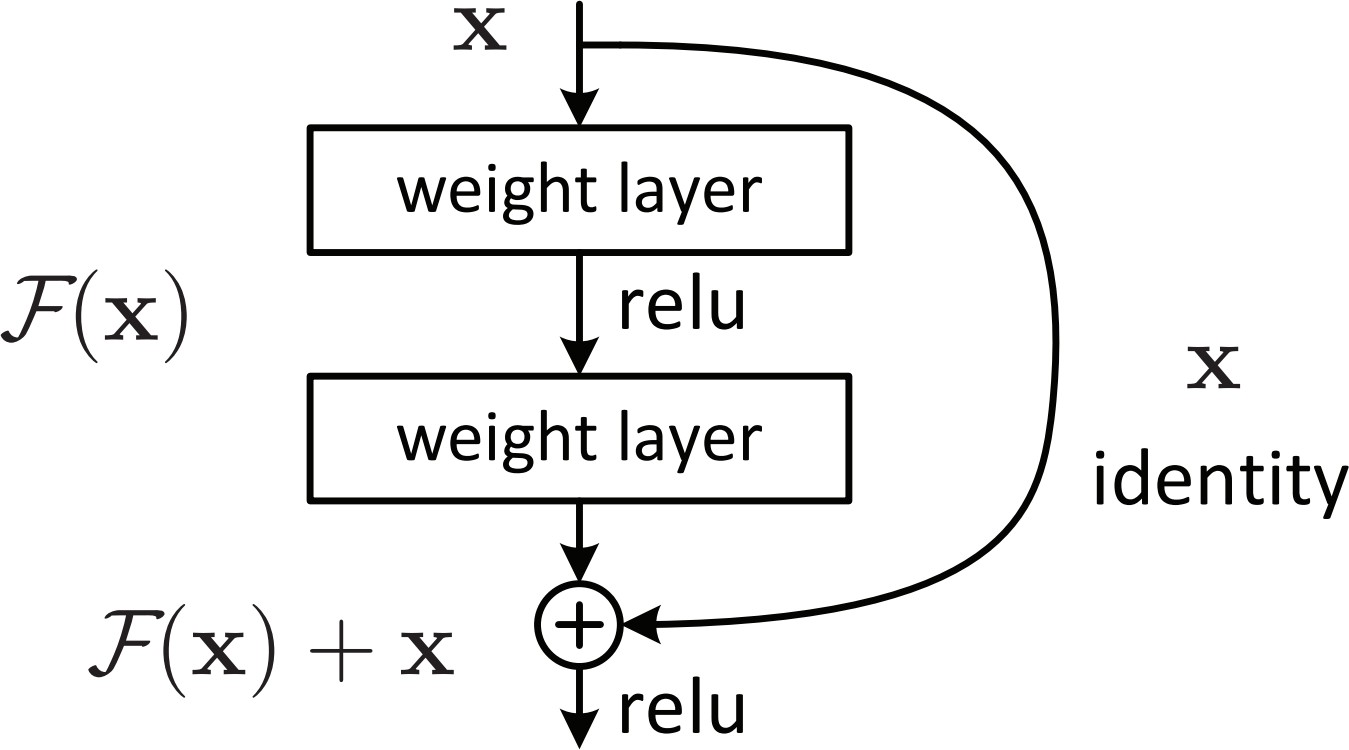


Fig. 3: Residual block.

As depicted in Figure 3, a residual block passes feature maps from one layer to another layer deeper in the network. By doing this, rather than deeper layers learning complex functions, the worst function that the deeper layers can learn is the identity function. By implementing these residual blocks, ResNets has the advantage of converging faster, increasing accuracy and decreasing loss.

This paper is divided as follows, section III briefly mentions previous research for the scene classification task; section IV explores the dataset, explains the preprocessing, gives an overview of how transfer learning is used and also explains the evaluation processes used for each experiment; section V explains the model used for the classification task; section VII explains the results of our experiments and finally, section VIII details how improvements can be made on the approach of this paper.

1. LITERATURE REVIEW
2. *Research*
   1. *Data Augmentation:* Data augmentation is a method for increasing a dataset’s diversity without acquiring new data [6].

It involves applying image transformation techniques to the images in the current dataset to generate new images. Some of the image transformation techniques are rescaling, rotation, flipping, and cropping. In [7], Liu et al. took a transfer learning approach to scene recognition. The baseline transfer learning model was ResNet. The authors worked with a small dataset and thus used data augmentation to further expand their dataset and potentially increase the accuracy of their proposed model. Data filtering was also used to ensure that the newly generated images from the data augmentation techniques belong to the same scene category as the original image. The results from Liu et al. paper were promising.

* 1. *Pooling Methods:* Pooling methods that have been investigated in the field of scene classification have mostly been spatial pooling methods. Spatial pooling is a method for creating spatial invariance in the data and works by grouping local features together based on their spatial adjacent pixels in images. In [11], Koskela et al. focus on using a spatial pyramid pooling approach for feature extraction. They trained several CNNs using this spatial pyramid approach on four different datasets and compared their observations to various other feature extraction methods. Results showed that there proposed approach performed better than previous methods.

Yang et al. recognized that without spatial layout information, the performance of deep CNNs suffers. Spatial information is really important for describing and discriminating scene images. The authors take full advantage of this and proposed a randomized spatial pooling layer and a maxout objective function for deep CNNs. What the randomized spatial pooling layer does is that it defines a pool of random partition patterns by which the feature map generated by the convolutional layers is partitioned into sub-regions with numerous shapes and sizes. Each sub-region is then further divided into a specific number of cells and the pooling process is performed over each cell. The maxout objective function perfectly complements the randomized spatial pooling layer by selectively choosing the optimal partition pattern to characterize the image layout. The results that Yang et al. produced showed an increase in performance [12].

* 1. *Model Bias and Dataset Problems:* Model bias is when a model performs extremely well on the training set but not so much on the validation and testing sets. Problems with datasets in scene classification tasks were that the datasets never consisted of an equal amount of images per scene category and it either never had enough density or diversity. These problems were solved by either ensuring high density and diversity in the dataset or making sure that all scene categories contain an equal amount of images. Zhou et al. built a database of scene images that overcomes the issue of model bias. This was done by making sure all image categories have a high density and high diversity. A high density was achieved by collecting a large number of images for each category. While for the issue of high diversity, they proposed a variation of the Simpson index of diversity [13].

In [14], Xiao et al. realized when comparing scene classification to object classification that one limiting factor of scene

classification was the inadequate amount of scene images and scene categories. The largest dataset of scene images contained only 15 categories. They built a database of scene categories that contained 899 scene categories and 130519 images to solve one of the problems mentioned above.

We intend to solve both of these problems by ensuring that each scene category contains an equal amount of images.

* 1. *Object Detectors:* Objects are quite prevalent in every scene and thus it would make sense for CNNs to classify scenes according to the objects in them. However, in [21], it was showed that instead of CNN's classifying scenes in the task of scene classification, it would classify objects. So even though it would make sense for CNNs to pick up these objects since they contribute to the overall ”sceneness” of a scene image, they also cause a problem. We intend to investigate how we can reduce this problem.

1. *Proposal*

Motivated by the research above, this paper will conduct three experiments using a Residual Network. They are:

* 1. Investigate how data augmentation can affect classification accuracy
  2. Investigate whether decreasing the resolution/size of the scene images can result in better classification accuracy and potentially decrease the emergence of object detectors
  3. Investigate how pooling methods such as Average Pooling and Max Pooling can affect classification accuracy

1. *Motivation for Proposal*

Below we motivate why we would investigate the outcome of each experiment.

## Experiment 1:

As shown in [7] data augmentation can improve performance, we would like to see if we can improve the performance of the dataset being used, using data augmentation techniques like flipping, rescaling, and cropping.

## Experiment 2:

From [21] and various other papers, it was shown that object detectors are most likely to emerge when doing scene classification. We would like to investigate whether decreasing the image resolution would make objects less detectable in scene images and therefore resulting in proper scene detectors.

## Experiment 3:

In [11] and [12], Spatial Pooling methods are used and is shown to not only improve performance but also preserve spatial information in images. We would like to investigate whether conventional pooling methods like Average Pooling and Max Pooling can also improve performance for the task of scene classification and as a result of this, can preserve spatial information.

1. METHODOLOGY
2. *Data Exploration*

The dataset used for this study is the Intel Scene Classification dataset and it was used in the Intel Scene Classification

Challenge. This dataset contains over 17 000 scene images and also contains 6 different scene categories, namely, building, forest, glacier, mountain, sea, and street.

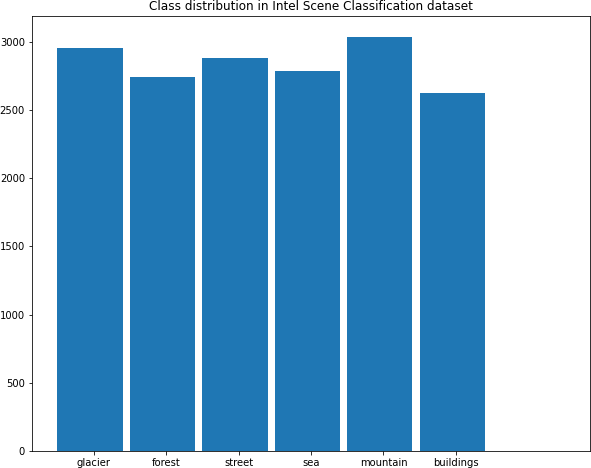


Fig. 4: Distribution of scene categories

The distribution of the scene categories is depicted in Figure

4. Each image is a color image with a size of 150x150 pixels. Figure 5 shows numerous images for each scene category to indicate the diversity of images in the dataset.

1. *Preprocessing*

As shown in Figure 4, the distribution of scene categories in the dataset is quite similar. However, to make sure that none of these experiments are affected by factors such as model bias, we record each scene category with the lowest amount of images and down-sample the rest of the scene categories to match this number. This results in each scene category having the same number of images. After down-sampling the categories we end up with 15768 images in total and 2628 image per scene category.

For our experiments, we end up splitting our dataset into 13146 images for the training dataset, with 2191 images for each scene category, and 2622 images for the testing dataset, with 437 images for each scene category.

Experiment 1 and 2 require different setups and is thus detailed below. No preprocessing is done for experiment 3 as the images are read in with a resolution of 100x100.

**Experiment 1:** Data augmentation is helpful when working with a small dataset [7]. It can drastically increase the number of images in the dataset with techniques such as flip, zoom, crop, and rotation of images. However, data augmentation also can destroy the spatial information of scene images [7]. The techniques such as zoom and crop can extract information from scene images that perhaps misrepresent the scene. Thus to make sure spatial information isn’t destroyed, only the flip technique, specifically the horizontal flip, is used. This technique mirrors the image along a central vertical line. An interpretation of this technique on a scene image is shown

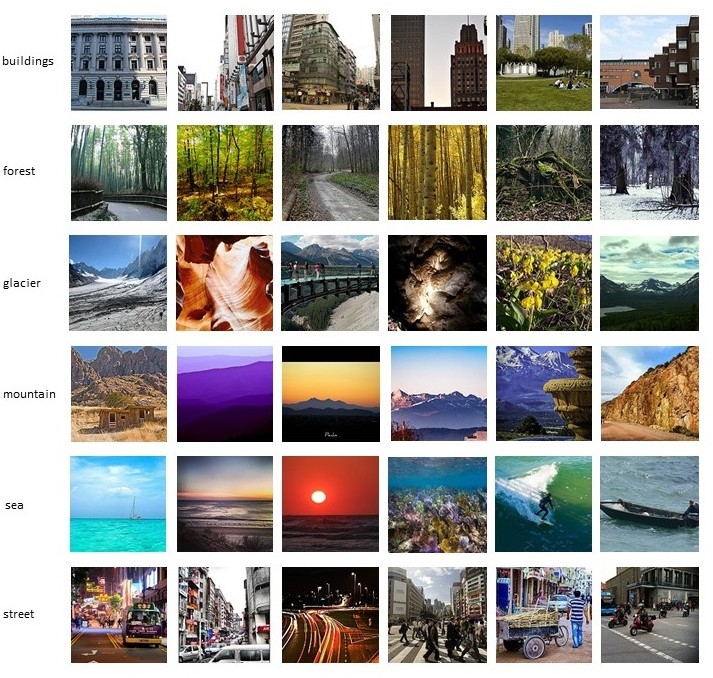


Fig. 5: This shows the diversity of images for each scene category in the Intel Scene Classification dataset

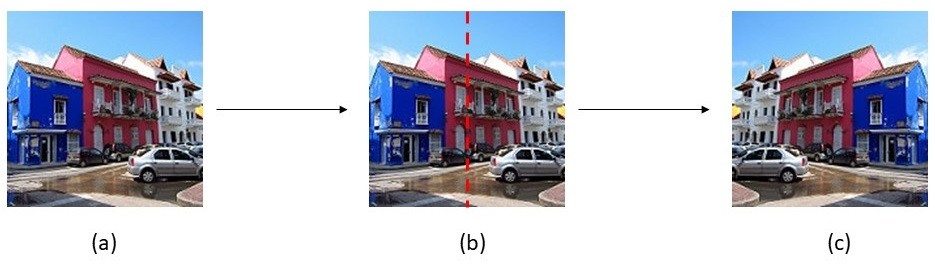


Fig. 6: This figure shows how the horizontal flip is performed. (a) Shows the original image. (b) Shows the vertical line upon which the image is mirrored. (c) Shows the result of the horizontal flip of the original image.

in Figure 6. Once this technique is applied to the dataset, it increases the overall number of images in the training set to 24894, with 4149 images per scene category.

**Experiment 2:** In this experiment, we investigate if the size of an image plays a role in the classification accuracy. Preprocessing for this experiment involves the resizing of the images. We resize the images to the 75x75, 100x100, 125x125, and 150x150 resolutions respectively.

1. *Transfer Learning*

Transfer learning is a type of learning method where a model that was used for one type of problem is used as a starting point for another problem. Transfer learning has gained quite a few tractions in the field of deep learning in the last couple of years. These transfer learning models have been trained for several hours on a very general dataset, making it easier for the next person to use these pre-trained models when the problem they are trying to solve falls in the same domain. With transfer learning, advantages such as a higher

starting point accuracy, better overall accuracy, and quicker model convergence can all be achieved.

Due to the limited computational resources, a transfer learning approach was used for this study. The Keras API comes with various pre-trained models. The pre-trained ResNet50 model which contains 50 layers was used in this work. This model was pre-trained on the ImageNet dataset which consists of over 1.2 million images of 1000 different classes. The ImageNet dataset is a general dataset that contains everyday things and therefore it made sense to use the ResNet50 model pre-trained with this dataset as a starting point.

1. *Evaluation Process*

For this paper, we carried out 3 different experiments. The evaluation process for each experiment is listed below:

## Experiment 1:

The aim of Experiment 1 is to see how data augmentation techniques affect the classification accuracy of the ResNet model. The specific data augmentation technique investigated is the horizontal flip. The first step in this experiment is to train the model on the original Intel Scene Classification dataset and then on the augmented Intel Scene Classification dataset. For each dataset, the same image size is used to ensure that this does not play a role in affecting the classification accuracy. By plotting the accuracy of the training and validation set for the original dataset and the augmented dataset against each other, we will be able to compare their results. The test set’s accuracy for each dataset will also be compared. By doing this thorough comparison we can safely conclude whether the classification accuracy achieved by the augmented dataset outperformed that of the original dataset and whether data augmentation techniques affect the classification accuracy.

## Experiment 2:

The aim of this experiment if to see if the size of the images that the ResNet model is trained on affects the classification accuracy and then whether it can reduce the emergence of object detectors. There are 4 different image resolutions that we investigate, they are: 75x75, 100x100, 125x125, and 150x150. The first step in this experiment is to create 4 separate datasets for the 4 different image resolutions. The ResNet model is then trained on each of the 4 datasets separately. For each image resolution, the same hyperparameters are used for the ResNet model so that we make sure no external factors can affect the classification accuracy. We then record the training and validation accuracy for the model trained on each of the 4 datasets. We also record the testing accuracy for the model trained on the 4 different datasets. By viewing the training, validation, and testing accuracy for each of the 4 different datasets, we can safely conclude which is better and whether the resolution of the images plays a role in the classification accuracy and whether the decreasing the image resolution can lessen the effect of object detectors.

## Experiment 3:

The aim of experiment 3 is to determine whether a specific pooling method affects the classification accuracy of the

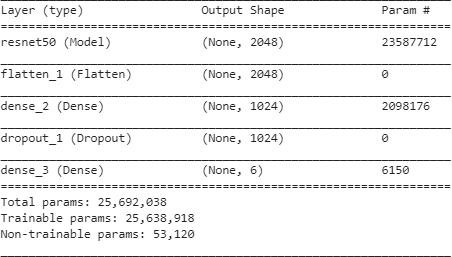
ResNet model. Two different pooling methods are investigated, Average Pooling and Max Pooling and we compare it to the ResNet model with no pooling method applied. To ensure that external factors do not affect the classification accuracy, all hyper-parameters of our ResNet model are kept constant. Also, we train the ResNet model on the same dataset. The sizes of the images for the dataset are 100x100. After constructing the ResNet model with the different pooling methods and the one without the pooling method, it is trained for 10 epochs. We record the training and validation accuracy for each approach. We also record their testing accuracy. By comparing the training, validation, and testing accuracies achieved by the different approaches, we can conclude whether or not a specific pooling method affects the classification accuracy.

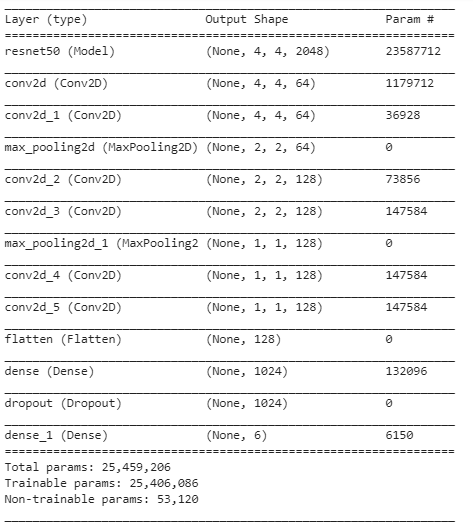
1. ARCHITECTURE

For this paper, we use 3 different architectures for the 3 different experiments. For experiment 1, we start by making use of the ResNet50 model from the Keras API. The weights used for this model are obtained from ImageNet. We set the weights to ImageNet for initialization purposes. Next, we add 6 convolution layers and 2 max pooling layers on top of the ResNet50 model. It is then followed by 1 Flatten, 1 Dropout, and 2 Dense layers. The last dense layer has a softmax activation. The point for adding all these layers on top of the ResNet50 model was to decrease the number of trainable parameters for the network. The architecture for experiment 1 is summarized in Figure7.

The model was compiled using the RMSprop optimization technique, with a learning rate set to 0.0001. This learning rate was obtained using a random search approach. The learning rate of 0.0001 made sure that the ResNet converged efficiently. Due to the limited computational power at hand, a batch size of 16 was used and the number of epochs was only 10. In the future, it would be advisable to train the network for longer as it might be able to improve performance. The type of loss function used was the Sparse Categorical Crossentropy loss function. The reason for using this loss function was because the target scene category labels were read in as integers instead of being one-hot encoded.

For experiment 2, a similar architecture was used with only the max pooling layers being removed. The max pooling layers were removed due to the reduced image resolutions. The rest of the hyper-parameters was the same as for experiment 1. The architecture for experiment 2 is summarized in Figure8. Experiment 3 used the same ResNet model from the Keras API coupled with 1 Flatten layer, 1 Dropout layer, and 2 Dense layers, with the last Dense layer providing the softmax output. The architecture for experiment 3 is summarized in Figure9. The model was compiled using the same optimization function and learning rate as in experiments 1 and 2, RMSprop, and 0.0001. Due to limited computation power, the batch size for this model was set to 8 and the number of epochs was set to 10. The same loss function was used as in experiments 1 and 2, Sparse Categorical Crossentropy.



Fig. 9: Architecture for experiment 3

The metric used to evaluate all experiments was overall accuracy.

Fig. 7: Architecture for experiment 1

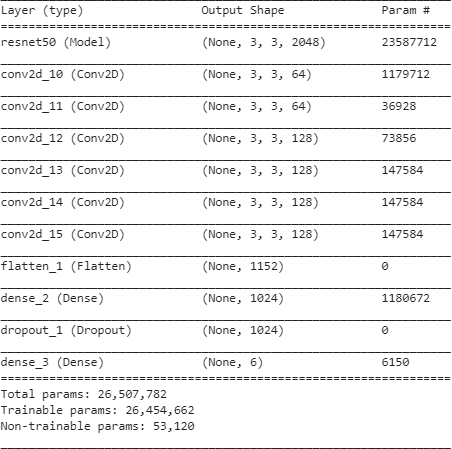


Fig. 8: Architecture for experiment 2

1. EXPERIMENTAL SETTING

All experiments were run in the Google Colaboratory environment. The Python libraries that were used are:

* + Numpy
  + Matplotlib
  + Random
  + Pathlib
  + Keras
  + Tensorflow

1. RESULTS AND ANALYSIS
2. *Experiment 1*

From Figures 10 and 11 we can see that the augmented dataset does unquestionably better than the original dataset. The ResNet model trained on the augmented dataset that achieves better training and validation accuracies than the ResNet model trained on the original dataset for all epochs. The model trained on the augmented dataset also converges much quicker than the model trained on the original dataset. By comparing the testing accuracies on the models as shown in Table I, we see that these statements are further enhanced. Hence, we conclude that data augmentation techniques like horizontal flipping can improve the classification accuracy of a model on this dataset.

TABLE I: Table showing testing accuracies for experiment 1

|  |  |
| --- | --- |
| Experiment 1 | |
| Technique | Testing Accuracy |
| No Data Augmentation | 87.30% |
| Data Augmentation | 89.93% |

TABLE II: Table showing testing accuracies for experiment 2

|  |  |
| --- | --- |
| Experiment 2 | |
| Image Resolution | Testing Accuracy |
| 75x75 | 86.80% |
| 100x100 | 81.05% |
| 125x125 | 88.18% |
| 150x150 | 90.20 % |

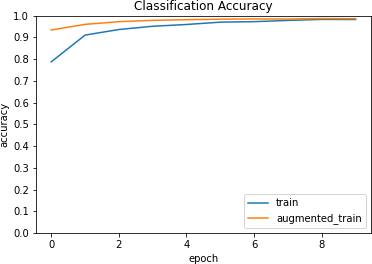


Fig. 10: Training accuracy for experiment 1

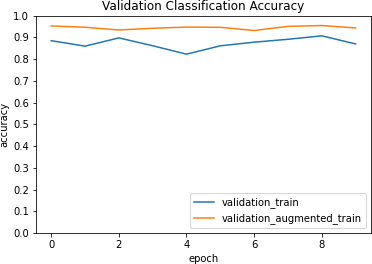


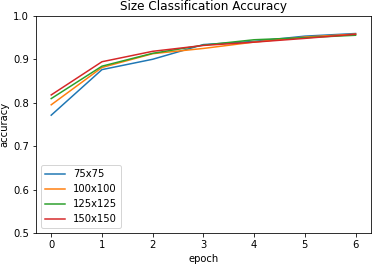
Fig. 11: Validation accuracy for experiment 1

TABLE III: Table showing testing accuracies for experiment 3

|  |  |
| --- | --- |
| Experiment 3 | |
| Pooling Method | Testing Accuracy |
| No Pooling Methods | 85.74% |
| Average Pooling | 89.09% |
| Max Pooling | 81.46 % |

1. *Experiment 2*

For experiment 2, we tested whether decreasing the image resolution result in better classification accuracy and whether it can potentially decrease the presence of object detectors. From Figure 12 we get confusing results as it seems that there is no clear image resolution that outperforms the rest. How- ever, looking at Figure 13 where we can see the average accuracy over all epochs of training and validation sets as well as the testing accuracy for each image resolution, we get a much better answer. The image resolution of 75x75 achieved the worst average validation accuracy while the image resolution of 150x150 achieved the best average validation accuracy. It seems that decreasing the image resolution decreases the classification accuracy. This is further noted by looking

Fig. 12: Training accuracy for experiment 2

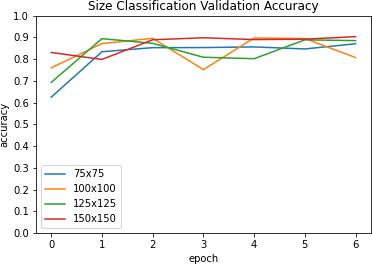


Fig. 13: Validation accuracy for experiment 2

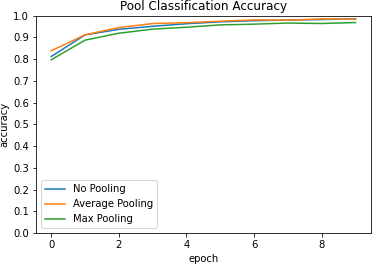


Fig. 14: Training accuracy for experiment 3

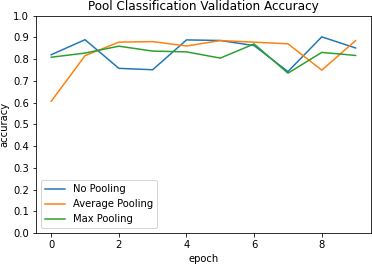


Fig. 15: Validation accuracy for experiment 3

at the testing accuracies in Table II. Even though the 75x75 image resolution achieved better accuracy than the 100x100 image resolution, it achieved the worst classification accuracy compared to the image resolutions of 125x125 and 150x150. This allows us to conclude that even though a decrease in image resolution decreases the classification accuracy and does not decrease the presence of object detectors.

1. *Experiment 3*

From Figures 14 and 15 we can see that average pooling had a positive effect on classification accuracy whereas max pooling had a slightly negative effect on classification accuracy. From Table III these results are further emphasized. The reason for average pooling doing better than max pooling is due to average pooling being able to extract a portion of each pixel in an image whereas max pooling only extracts the most important pixels or features from the image. By extracting a portion of each pixel or feature, it might be that average pooling preserves more spatial information than max pooling and therefore performs better.

1. DISCUSSION AND OUTLOOK

The main issue experienced when carrying out with paper was the limited amount of resources that were at hand. For future approaches on scene classification, it would be advisable to run the experiments on a computer that has a sufficient amount of RAM. Issues experienced for experiments 1 and 2 includes:

**Experiment 1:** We were only able to use one type of data augmentation technique, horizontal flipping, and due to the limited amount of RAM at hand. Other data augmentation techniques like rescaling and cropping might be able to increase the classification accuracy even further and is recommended for future work.

**Experiment 2:** We were not able to evaluate our ResNet model on more than 4 different sizes of image resolutions. It would be advisable to use at least 10 different image resolutions to see the effect on this experiment.

For future work, Weighted Average Pooling could be investigated for the task of scene classification as it can preserve even more spatial information than Average Pooling.

REFERENCES

[1]Z. Yi and H. Zhang, ”Multi-Scale Convolutional Neural Networks for Scene Recognition,” 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Zhangjiajie, China, 2019, pp. 2389-2394.

[2]M. Jiafa, W. Weifeng, H. Yahong and S. Weiguo, ”A scene recognition algorithm based on deep residual network”, in Systems Science & Control Engineering, vol. 7, no. 1, pp. 243–251, 2019.

[3]L. Chen, W. Zhan, W. Tian, Y. He and Q. Zou, ”Deep Integration: A Multi-Label Architecture for Road Scene Recognition,” in IEEE Transactions on Image Processing, vol. 28, no. 10, pp. 4883-4898, Oct. 2019.

[4]B. Li et al., ”Further Exploring Convolutional Neural Networks’ Poten- tial for Land-Use Scene Classification,” in IEEE Geoscience and Remote Sensing Letters.

[5]J. Gao, J. Yang, J. Zhang and M. Li, ”Natural scene recognition based on Convolutional Neural Networks and Deep Boltzmannn Machines,” 2015 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, 2015, pp. 2369-2374.

[6]H. Daniel, E. Liang and R. Liaw, ”1000x Faster Data Augmentation,” 2019. [Online]. Available: https://bair.berkeley.edu/blog/2019/06/07/data aug/. [Accessed 26 May 2020]

[7]S. Liu, G. Tian, Y. Xu, ”A novel scene classification model combining resnet based transfer learning and data augmentation with a filter”, Neurocomputing, vol. 338, pp.191-206, 2019.

[8]H. Seong, J. Hyun, and E. Kim. ”Fosnet: An end-to-end trainable deep neural network for scene recognition”, arXiv preprint arXiv:1907.07570, 2019.

[9]S. H. Khan, M. Hayat, M. Bennamoun, R. Togneri and F. A. Sohel, ”A Discriminative Representation of Convolutional Features for Indoor Scene Recognition,” in IEEE Transactions on Image Processing, vol. 25, no. 7, pp. 3372-3383, July 2016.

[10]S. Guo, W. Huang, L. Wang and Y. Qiao, ”Locally Supervised Deep Hybrid Model for Scene Recognition,” in IEEE Transactions on Image Processing, vol. 26, no. 2, pp. 808-820, Feb. 2017.

[11]M. Koskela and J. Laaksonen, “Convolutional network features for scene recognition,” in ACM, 2014, pp. 1169–1172.

[12]M. Yang, B. Li, H. Fan and Y. Jiang, ”Randomized spatial pooling in deep convolutional networks for scene recognition,” 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, 2015, pp. 402-406.

[13]B. Zhou, A. Lapedriza, A. Khosla, A. Oliva and A. Torralba, ”Places: A 10 Million Image Database for Scene Recognition,” in IEEE Transac- tions on Pattern Analysis and Machine Intelligence, vol. 40, no. 6, pp. 1452-1464, 1 June 2018.

[14]J. Xiao, J. Hays, K. A. Ehinger, A. Oliva and A. Torralba, ”SUN database: Large-scale scene recognition from abbey to zoo,” 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010, pp. 3485-3492.

[15]L. Herranz, S. Jiang and X. Li, ”Scene Recognition with CNNs: Objects, Scales and Dataset Bias,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 571-579.

[16]M. Oeljeklaus, F. Hoffmann and T. Bertram, ”A combined recognition and segmentation model for urban traffic scene understanding,” 2017 IEEE 20th International Conference on Intelligent Transportation Sys- tems (ITSC), Yokohama, 2017, pp. 1-6.

[17]B. Oh and J. Lee, ”A case study on scene recognition using an ensemble convolution neural network,” 2018 20th International Conference on Ad- vanced Communication Technology (ICACT), Chuncheon-si Gangwon- do, Korea (South), 2018, pp. 351-353.

[18]B. Yuan, S. Li, and N. Li, “Multiscale deep features learning for landuse scene recognition,” J. Appl. Remote Sens., vol. 12, no. 1, p. 015010, 2018.

[19]L. Wang, S. Guo, W. Huang, and Y. Qiao, “Places205-vggnet models for scene recognition,” CoRR, vol. abs/1508.01667, 2015.

[20]D. N. Tran, L. H. Pham, H. M. Tran and S. V. Ha, ”Scene recognition in traffic surveillance system using Neural Network and probabilistic model,” 2017 International Conference on System Science and Engi- neering (ICSSE), Ho Chi Minh City, 2017, pp. 226-230.

[21]B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Object detectors emerge in deep scene cnns,” International Conference on Learning Representations, 2015.