Building Recognition at The University of Virginia Using Deep Learning Based Models

Soneya Binta Hossain, Will Leeson
Department of Computer Science, University of Virginia, Charlottesville, VA 22904
[sh7hv, wel2vw]@virginia.edu

Abstract

In this work, we studied deep learning methods to classify 20 different buildings at the University of Virginia. Due to the lack of an existing training and validation dataset, we built a dataset of captured images of the buildings. We built two different datasets with varying numbers of total training and validation images per class. Our first dataset is slightly unbalanced and contains 507 images in total, whereas the second dataset is more unbalanced and contains 769 images in total. We trained ResNet50, MobileNet_v2, and Alexnet on both of our datasets to observe how training data size affects model accuracy. We also fine-tuned versions of these models that were trained on ImageNet.

We performed several experiments to compare accuracies of different CNN models without pretraining and pretrained on Imagenet on two different datasets, and presented our experimental results. Accuracy of models trained and fine tuned on our second dataset (616 training and 153 validation image) performed better than first dataset (415 training images and 92 validation images) which is smaller than second dataset. With our second dataset (616 training and 153 validation image), we achieved 93.13% on fined tuned MobileNet model, and 71.88% accuracy on the ResNet50 architecture.

Index Terms—Deep Learning, CNN, Building Recognition

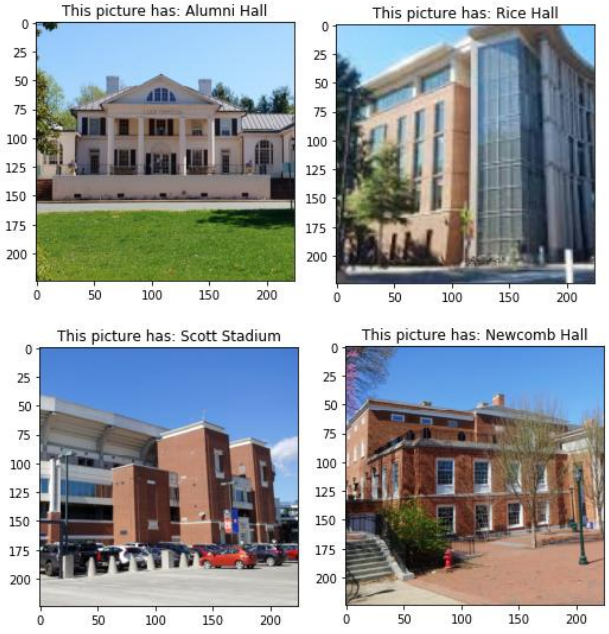

Figure 1. Here is a sample of images that make up our data set. The majority of each image contains the labelled building. These images were collected from pictures the authors took and images found online. Our goal is to predict for any of the 20 buildings we chose to look at in this project.

1. Introduction

Machine leaning models, specially deep learning models, are being used in a variety of fields for various purposes. To name a few, deep learning models are used for steering angle prediction in self driving cars, object detection, semantic segmentation, instance segmentation, and image captioning. Indoor scene recognition is an open and challenging problem in computer vision. Classification of building as an apartment building, industrial building or other type of building is also an interesting and challenging classification problem in computer vision[7].

The task of object recognition and classification is one of the most important tasks in the field of computer vision. As self driving cars become more and more popular, it will allow them to recognize and react on the fly to objects that may appear in the road at a moments notice. It can also help with the recognition of people in photos to identify where someone was based on the location of the camera that captured the photo.

As of late, the primary tool used for object recognition has been convolutional neural networks (CNNs). These networks pass images through their many layers where a se-

ries of linear and activation functions make calculations and then propose a possible label for the image. These labels can classify what objects are in the image, answer whether or not a object in question is in the image, or identify where certain objects are in the image. The complexity of the state of the art networks have grown over time, from 8 layers to over 100[4][5].

Building recognition has become a popular computer vision problem. Given outdoor views of building, a deep learning model can predict the building name. Like, indoor scene recognition, build recognition with outdoor images of buildings is very challenging. Since photos can be taken at a variety of angles at different time of day with different lighting, a model must be able to recognize buildings under various conditions.

In this project, we developed a building recognition system for the University of Virginia using convolutional neural networks. Due to the lack of an existing dataset, we built our own dataset. We captured images of different buildings at the University of Virginia serving purposes such as academic buildings, recreation centers, libraries, or athletic stadiums. Our datasets consist of 769 images capturing one of the 20 buildings we chose. We trained ResNet50, MobileNet_v2, and Alexnet on our dataset. We also fine tuned pre-trained ResNet50, MobileNet_v2, and Alexnet models. To observe how model accuracy varies with training data size we built two different data sets with varying number of images. The maximum accuracy we achieved is 93.75%.

Therefore, our contributions in this project are:
1. We built a dataset that can be used for future building recognition tasks.
2. Trained and fine tuned ResNet50, MobileNet_v2, and Alexnet networks at the task of identifying buildings at the University of Virginia.
3. Performed several experiments to compare accuracy of different models.

This paper is structured as follow: Section II will describe related works, Section III will discuss our datasets, Section IV will discuss implementation and training, Section V describe the experiments of this study, Section VI will discuss the experimental results, and we conclude our paper in Section VII.

2. Related Work

Bezak et al. [3] applied a similar deep learning approach to recognize buildings in historical building photographs of the town Trnava. Like our project, they had to create their own data set of training and validation images. For their experiment, they used the LeNet model, which was the least memory consuming. Their model was based on the data set of 460 training images and 140 validation images. Each image was a color jpg of dimension $28 * 28$ pixels. Furthermore, there are a lot of other works for indoor scene recognition with deep learning, which is a challenging open problem in high level vision.

Ayatar et al. [2] have shown the advantages of taking a model that has been pre-trained on a large data set and fine-tuning it to a new data set. For one, this process reduces the time required to train a new model for a new task. It can also reduce the amount of data needed for this new task. Most importantly, it can result in a boost in accuracy.

Lomio et al. [7] used machine learning models related to image recognition task to to automatically recognize the type of the building based on its Building Information Model (BIM) virtual representation. They used classical machine learning methods, pre-trained deep learning networks, and networks designed from scratch to separate BIM data in three different categories: apartment building, industrial building and others.

Table 1. Images Per Category in slightly unbalanced Data Set

Building Name	Images	Building Name	Images
Alderman Library	23	Alumni Hall	16
Aquatic Fitness Center	27	Clay Hall	31
Chemical Engineering Building	24	John Paul Jones Arena	25
Mechanical Engineering Building	25	Memorial Gym	20
Monroe Hall	24	Newcomb Hall	24
North Grounds Gym	24	Old Cabell Hall	28
Olsson Hall	28	Physics Building	29
Rice Hall	34	The Rotunda	29
Scott Stadium	21	Slaughter Recreation	25
Thornton Hall	25	Wilsdorf Hall	25

Table 2. Images Per Category in more unbalanced Data Set

Building Name	Images	Building Name	Images
Alderman Library	23	Alumni Hall	16
Aquatic Fitness Center	70	Clay Hall	60
Chemical Engineering Building	34	John Paul Jones Arena	79
Mechanical Engineering Building	61	Memorial Gym	20
Monroe Hall	24	Newcomb Hall	24
North Grounds Gym	24	Old Cabell Hall	28
Olsson Hall	50	Physics Building	62
Rice Hall	34	The Rotunda	29
Scott Stadium	21	Slaughter Recreation	25
Thornton Hall	30	Wilsdorf Hall	56

3. Data Set

The data set used in this work consists of 20 buildings at the University of Virginia's main campus and law school campus. These buildings serve a variety of purposes such as lecture halls, administrative offices, sports complexes, and fitness centers. Most of the images were captured by the authors, a small portion were collected from internet.

Our dataset consists of images of the following buildings: Rice Hall, Wilsdorf Hall, Thornton Hall, the Physics Building, Olsson Hall, the Mechanical Engineering Building, Clay Hall, John Paul Jones Arena, Chemical Engineering Building, Aquatic & Fitness Center, Scott Stadium, Alderman Library, Newcomb Hall, Old Cabell Hall, The Rotunda, Monroe Hall, North Grounds Gym, Slaughter Recreation Center, Alumni Hall, Memorial Gym.

We build two datasets of different size with a different distribution of image labels to understand how models' accuracies vary with these variables.

DATASET-1: This dataset is a subset of DATASET-2. This dataset consists of 414 training images and 93 validation image. As shown in 1, number of images per category is not equal, therefore, this dataset is slightly unbalanced. Most buildings have around 25 images, however, we have only 16 images of Alumni Hall, and 34 images for Rice Hall.

DATASET-2: This dataset consists of 616 training images and 153 validation images. As shown in Table 2, the maximum number of image per category is 79 and the minimum number of image is 16. Therefore, this dataset is even more skewed.

4. Implementation

We implemented our data set and models using the python library pytorch.

4.1. Data loader

We built our training and validation dataset using PyTorch's **torch.utils.data.Dataset** module, and implemented _len_ and _getitem_ method. In a CSV file, we stored the image file names, images' true category and category names. We also used **torch.utils.data.DataLoader** module to create an enumerable training and validation set. We resized each image to 320 * 320. Detail implementation of dataloader can be found in our project repository[6].

4.2. Training Networks

In order to train our networks, we created a train_model function which takes in several variables: a network, a loss function, the batch size, the training data set, the validation data set, a pytorch optimizer, and the number of training epochs. This function is based on the train_model function from assignment 4 and the function found in the pytorch classifier training tutorial [1].

5. Experiments

We trained AlexNet, MobileNet v_2, and ResNet50 on both of our datasets: DATASET-1 and DATASET-2. We ran 30 training epochs with a learning rate 5e-4, and a momentum of 0.9 for fresh networks and 5 training epochs with a learning rate 5e-4, a momentum 0.9 and weight decay of 1e-5 for fine tuned networks.

Table 3. Accuracy of Models

	Fresh Network		Fine Tuned Network	
Architecture	Dataset-1	Dataset-2	Dataset-1	Dataset-2
AlexNet	25.00%	27.50%	83.33%	82.05%
MobileNet	52.08%	71.25%	92.71%	93.13%
ResNet	41.67%	71.88%	90.62%	91.87%

5.1. Training Fresh Networks

First, we attempted training three well known networks, AlexNet, MobileNet v_2, and ResNet50, with our initial dataset. For each network, we used cross entropy loss as our loss function and pytorch's stochastic gradient descent optimizer. We trained pytorch's implementation of each network to varying results.

5.2. Fine-Tuning Networks

Next, we attempted fine-tuning the same three networks. Each network was trained using ImageNet and came from pytorch. We once again used cross entropy loss and pytorch's stochastic gradient descent optimizer.

5.3. Using Unbalanced Dataset

To understand how accuracy of our models using AlexNet, MobileNet v_2, and ResNet50 network vary with training dataset size, we performed separate experiments using our two different size datasets: DATASET-1 and DATASET-2. We trained all three networks and fine tuned them on both datasets. We also applied weighted cross entropy loss to observe how this loss affects models accuracy. However, in our experiments, we did not observe any significant change on accuracy after using weighted cross entropy loss. We used different formulas for calculating weights. However, the accuracies remain almost same with and without weights. We only observe that increasing the dataset's size increases model's accuracy. Models trained on DATASET-2 performed better that models trained on DATASET-1. We show the detail results in Section VI.

6. Experiment Discussion

6.1. Training Fresh Networks

The accuracy results from our first experiment with DATASET-1 were rather poor. For AlexNet, we achieved 25.00% accuracy. This is better than random, but that is a low bar for results in this day and age. As we can see from Figure 2, the network predicted five labels for most images, so it is biased towards those images.

For MobileNet and ResNet, we achieved an accuracy of 52.08% and 41.67% respectively. This is by no means

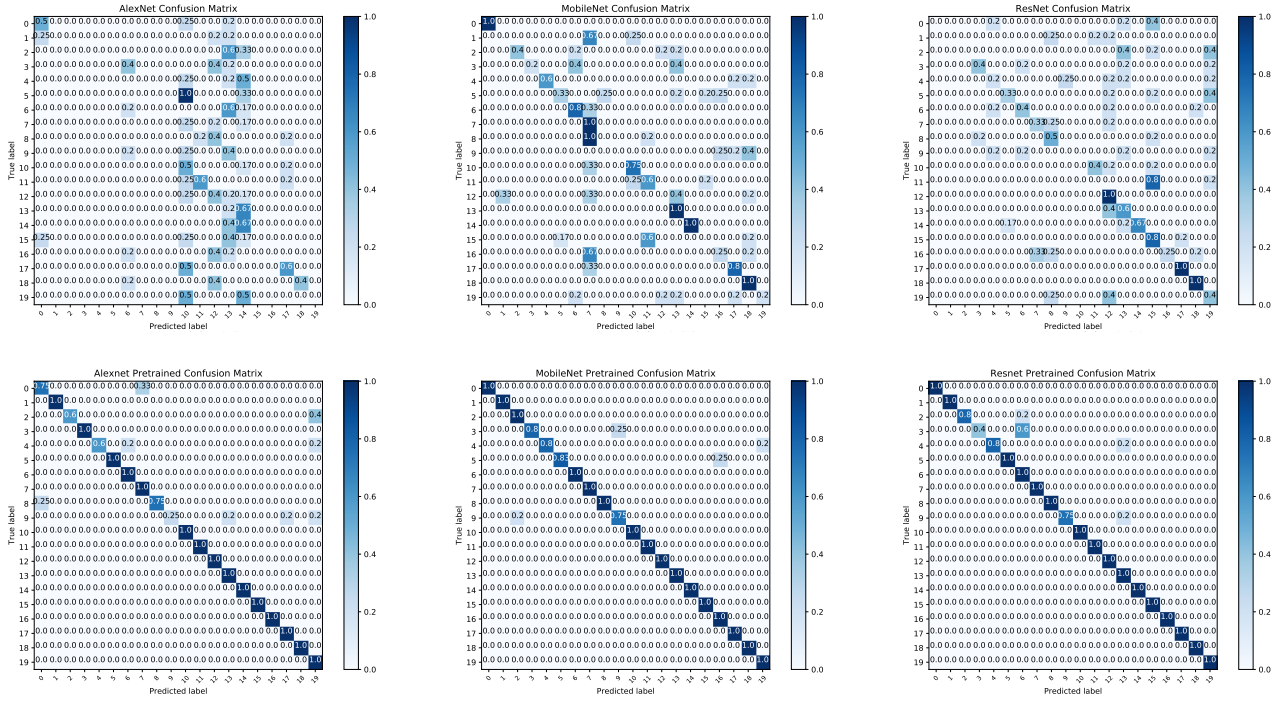

Figure 2. Confusion Matrices for models trained on DATASET-1

fantastic accuracy, but it is a significant improvement over AlexNet. The confusion matrices for these networks in Figure 2 show a stronger diagonal. This means that these networks were better at correctly predicting images over more labels. Thus, these networks are more general purpose since they can identify more buildings.

The results from this experiment show the issue with having too little data. The lack of data prevented these models from being able to learn to better solve this task. We are able to see the power of having deeper networks. Both MobileNet and ResNet are significantly more robust networks than AlexNet and this shows in the accuracy these networks were able to achieve on such a small data set and their ability to learn more about more types of buildings.

6.2. Fine-Tuning Networks

The accuracy results for our second experiment with DATASET-1 were far superior to our first experiments. For AlexNet, we achieved an accuracy of 83.33%. This is not state of the art, but it is around what is to be expected of AlexNet. On ImageNet, AlexNet has been recorded to obtain around 85% accuracy [5]. Since ImageNet consists of millions of images, a 2% decrease in accuracy is reasonable for a dataset consisting of around 500 images.

Similar to the first experiment, MobileNet and ResNet outperformed AlexNet. Both of these models were able to obtain accuracy around 91%. ResNet has been recorded at achieving an accuracy of around 97%[4]. As we pointed out, our data set is significantly smaller than ImageNet.

Figure 2 shows a stark difference between the fine-tuned and fresh networks. The fine-tuned networks have a clear diagonal on their confusion matrices with a few blips. These networks perform very well across all labels and do not favor one label over another. The fresh networks either have no clear diagonal, or they have many areas not across the diagonal which are filled in.

This experiment speaks to the strength of transfer learning. A data set of 500 images is minuscule, especially compared to the large crowd sourced data sets that exist in the world now. As shown in experiment one, the state of the art networks such as ResNet and MobileNet achieve poor accuracy when trained alone on such a small data set. However, when they are first trained on a large data set such as ImageNet, they are able to perform near state of the art. This shows that these networks are gaining some sort of knowledge that can be targeted at solving a new task better than they could without this background knowledge.

6.3. Using Unbalanced Datasets

We repeated these experiments in Section 6.1 and 6.1 using our DATASET-2. We did this to see the effects of additional data. We also normalized for the skew of the label distribution. This prevents the model from predicting a label more simply because it occurs more in the data set.

From this experiment, we found a slight bump in accuracy for fine-tuned models and a significant bump in fresh models. This speaks to the power of more data. Even though both data sets are only 500 and 700 images, this increase of around 40% to the size of the data set has an effect.

7. Conclusion

Building a large data set for a deep learning problem is a nontrivial task. For many problems, this process is achieved from scraping the internet, crowd sourcing, and a large manual effort. When the authors approached this problem, they knew this would be an issue. Since many of these buildings are obscure to anyone who is not a member of the University of Virginia community, they do not have a large presence on the Internet. This meant the authors would have to capture the images themselves. Since they could not dedicate all of the hours of this project to building a data set, a data set of 700 images would have to suffice.

This project showed the power of data when creating deep learning models and transfer learning. The difference between 200 images in the training stage resulted in a difference of 20% in the accuracy. Unfortunately, there are diminishing returns on these increases in data, so it would take significant effort to continue to improve the model to state of the art. This is where transfer learning comes into play. Transfer learning allows for using a much larger data set to teach the model some latent knowledge that can then be targeted at a new task with some light training. This fine-tuning allowed the authors to create a model with accuracy near the state of the art with a data set far smaller than a state of the art data set.

References

[1] Training a classifier, apr 2017.

[2] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *2011 International Conference on Computer Vision*, pages 2252–2259, 2011.

[3] P. Bezak. Building recognition system based on deep learning. In *2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR)*, pages 1–5. IEEE, 2016.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] W. lesson and S. Hossain. Building recognition deep model. `https://github.com/soneyahossain/deep_learning_project`, 2020.

[7] F. Lomio, R. Farinha, M. Laasonen, and H. Huttunen. Classification of building information model (bim) structures with deep learning. In *2018 7th European Workshop on Visual Information Processing (EUVIP)*, pages 1–6. IEEE, 2018.