Remote Sensing Image Classification

Project Final Report

Atakan Serbes, 21200694
Department of Computer Science
Bilkent University
atakan.serbes@bilkent.edu.tr

***Abstract*—Remote sensing image analysis, with the advance of new technologies is becoming an important area of research. During recent years, after the introduction of UCMerced dataset in 2010, many new datasets have been made available to public. However, there are no extensive surveys, research or datasets in the field of remote sensing compared to natural scene image analysis. NWPU RESISC-45 is one of the most comprehensive datasets in the field of remote sensing and authors introduce their extensive dataset and apply deep learning methods to be a benchmark for the future studies. In this study, I apply VGGNet-16 architecture to study classification on NWPU RESISC-45 dataset and obtain good results when it is compared to what authors proposed in their own paper. Also it should be noted that I work with 12 classes in the dataset which also decreases the number of samples that is studied. Only the classes with objects inside them are selected and all the scene scenes like forest, sea, ice are discarded.**

***Index Terms*—Remote sensing, deep learning, scene classification, satellite images**

I. Problem Description

Nowadays, the amount of data collected by the aerial means, especially by the satellites, is increasing day by day with the advance of technology. Remote sensing image classification is one of the fundamental tasks of image processing in this field. Deep Convolutional Neural Networks (DCNNs) brought the state-of-the-art learning framework for the image recognition. DCNNs are data hungry and they are trained by many images with their labels, however it is hard to find adequate number of images or classes in the area of remote sensing. The first popular satellite scene classification dataset ”UC Merced Land Use Dataset” was introduced in 2010 [1] which shows that it is in a very different phase than nature scene image datasets. To show the limitations of remote sensing data even further, UCMerced dataset only has 21 class and 100 images for each class.

Aerial imaging are also valuable for efforts from urban planning, disaster evaluation, vegetation mapping, environment monitoring, natural hazards detection and economic analysis [2] to military and government agency efforts [3] .

I am going to lean on the problem of scene classification with very little remote sensing data. It is hard to train a model using small amount of data and this is why building a powerful image classification model using very little data is a good area to study on. Some studies may require us to work on limited number of training samples and it is important to be able to create a great model to be able to work on the specific study. This problem may be studying on just a few hundred or thousand pictures to be able to classify the images correctly or classify the images with near state-of-the-art results.

II. Related Work

With the increasing number of datasets to work remote sensing image classification increasing, more people are working in this field. In the paper that introduced the dataset NWPU-RESISC45 [3], some information about the current remote sensing image datasets are given. As can be seen from Figure 1, NWPU dataset is more comprehensive and with higher number of samples when compared with other publicly available datasets.

In the paper [3], authors try various state-of-the-art deep learning methods on their proposed NWPU-RESISC45 dataset. The popular deep networks such as AlexNet [4], VGGNet-16 and GoogLeNet [5] are fine-tuned and applied to dataset. These fine-tuned transfer learning methods for AlexNet, VGGNet-16 and GoogLeNet give overall accuracies of 85%, 90% and 86% respectively [3].

The not-so-high accuracies can be explained by the pretrained networks. Pretraining on ImageNet and then applying this to their dataset, they get the corresponding results. The images in ImageNet are not compatible with satellite images. ImageNet images are all nature scene images which are different in their nature to satellite images. Satellite images are bird-view and objects appear from their above whereas nature scene images are in upright position due to gravity.

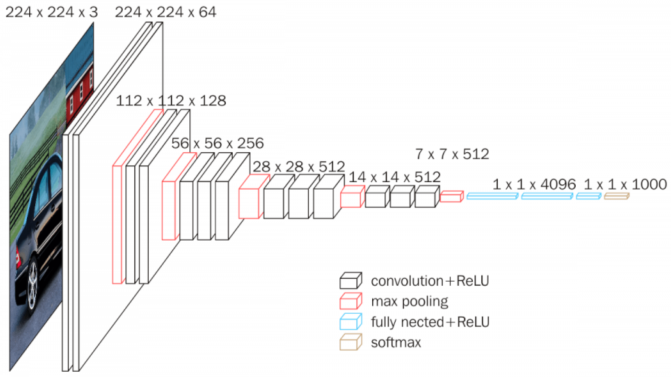
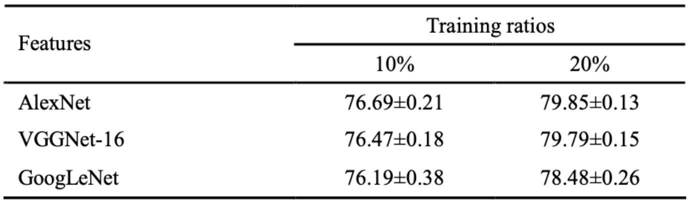


Fig. 1: VGGNet-16 architecture [6]

VGGNet was proposed [6] and has won localization and classification tasks of the ILVSRC-2014 competition. Authors of [3] use VGGNet-16 architecture instead of VGGNet-19 because of its simpler architecture and better performance.

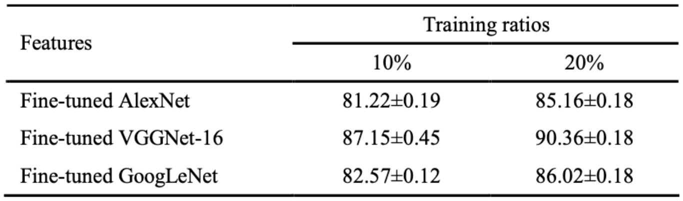
As explained previously, authors try three popular architectures and conclude that VGGNet-16 configuration performs better than AlexNet or GoogLeNet. Their results are given in the following figures.



Features	Training ratios	
	10%	20%
AlexNet	76.69±0.21	79.85±0.13
VGGNet-16	76.47±0.18	79.79±0.15
GoogLeNet	76.19±0.38	78.48±0.26

Fig. 2: Three kinds of deep learning SotA under training ratios of 10% and 20%

As it can be seen from figures 1 and 2, before fine-tuning the architectures, the accuracies of the best approaches are 76.69, 76.47 and 76.19 percent respectively for AlexNet, VGGNet-16 and GoogLeNet. The accuracy of VGGNet-16 increases substantially higher than the other two architectures after fine tuning.



Features	Training ratios	
	10%	20%
Fine-tuned AlexNet	81.22±0.19	85.16±0.18
Fine-tuned VGGNet-16	87.15±0.45	90.36±0.18
Fine-tuned GoogLeNet	82.57±0.12	86.02±0.18

Fig. 3: Fine-tuned SotA result under training ratios of 10% and 20%

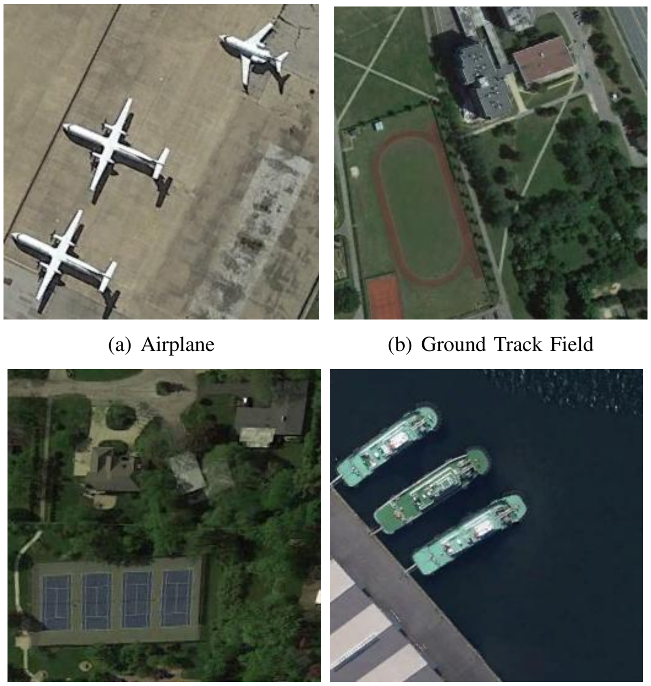
III. Dataset

Many datasets were examined for the project. And to study the problem of scene classification, I selected the NWPU-RESISC45 dataset [3]. The reason for this selection is, it has the highest number of total images compared to other publicly available satellite image datasets as it can be seen from Figure 5.

The dataset originally contains 31,500 images covering 45 scene classes with 700 images in each class. The image sizes are 256x256 and spatial resolution is higher than other datasets available. For my study I selected 12 classes to study inside this dataset and will implement my solution on the following classes,

- airplane
- baseball diamond
- basketball court
- bridge
- church
- ground track field
- harbor
- roundabout
- ship
- stadium
- storage tank
- tennis court

So we have 8400 images in total in which we will study the problem of image classification with little amount of data.



(a) Airplane

(b) Ground Track Field

(c) Tennis Court

(d) Ship

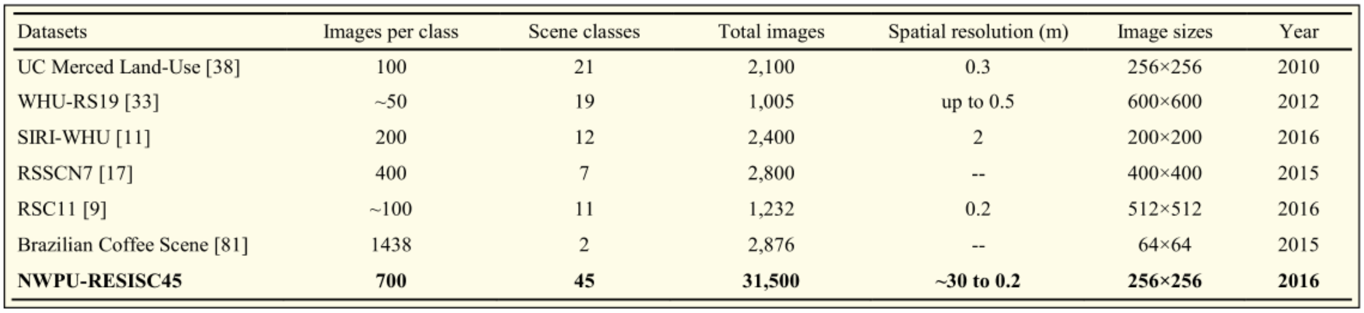
Fig. 4: Some examples from the NWPU-Resisc45 dataset

IV. Approach

I divided my train/validation/test ratio as 80/10/10 to study the classification. So the number of images ratio are as 560 / 70 / 70 per class for 12 class of images. The network was fed with batch sizes of 32 images and learning rate was set different to find the optimal rate while using the pretrained network. Also the number of fully connected layers at the end of the network was changed to see the effects on accuracy results.

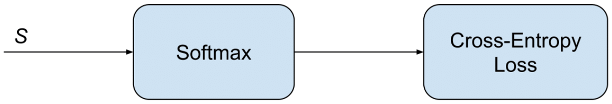
Network

I implemented transfer learning using a VGG-16 network pretrained on ImageNet data. The reason of selection for VGG-16 approach was its superior performance on this dataset compared to other approaches [3]. After extracting features with VGG16, I added fully connected layers and added softmax of 12 nodes for predicting 12 class. Since we have a multi-class classification problem, the loss used for this problem was categorical cross-entropy. (If this were a cat-dog classifier for example, or a building-no building classifier, we would use binary cross-entropy loss function.) The categorical cross-entropy loss is also called softmax loss. It is a softmax activation plus a cross-entropy loss. So it outputs a probability over the C classes for each image.



Datasets	Images per class	Scene classes	Total images	Spatial resolution (m)	Image sizes	Year
UC Merced Land-Use [38]	100	21	2,100	0.3	256×256	2010
WHU-RS19 [33]	~50	19	1,005	up to 0.5	600×600	2012
SIRI-WHU [11]	200	12	2,400	2	200×200	2016
RSSCN7 [17]	400	7	2,800	--	400×400	2015
RSC11 [9]	~100	11	1,232	0.2	512×512	2016
Brazilian Coffee Scene [81]	1438	2	2,876	--	64×64	2015
NWPU-RESISC45	**700**	**45**	**31,500**	**~30 to 0.2**	**256×256**	**2016**

Fig. 5: Comparison between NWPU-RESISC45 and some other publicly available datasets [3]



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \quad CE = -\sum_i^C t_i log(f(s)_i)$$

Fig. 6: Categorical cross-entropy loss

The network parameters are given as follows,

- VGG architecture
- 1 or 2 Fully connected layer
- 6720 training samples
- 840 / 840 validation and test samples
- batch size of 32
- 12 classes
- 50 epochs
- 0.008 learning rate
- 0.5 dropout rate

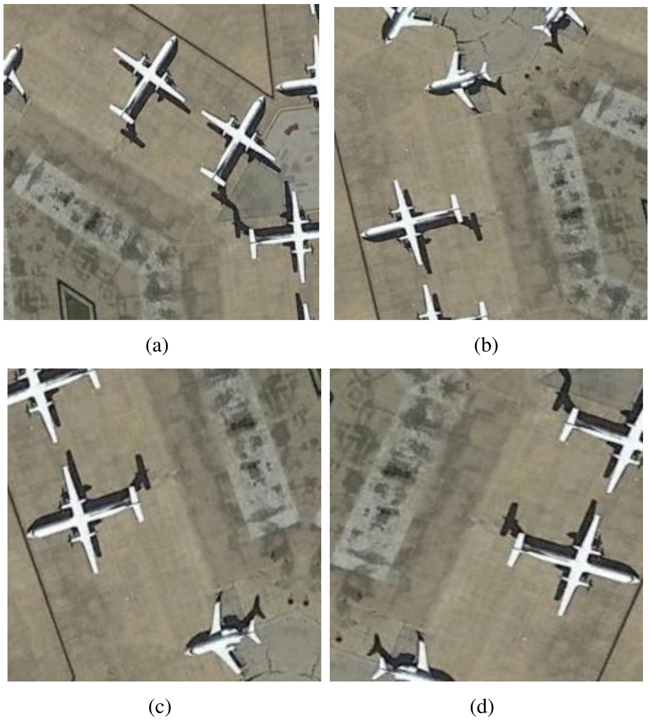
In the case of multi-class classification problem, the labels are in one-hot representation. (It is easily obtained by keras.utils.np_utils packages' to_categorical method) Only the positive class keeps its term in the loss.

Also as a side note, in a work by Facebook [7], they claim that categorical cross-entropy loss works better than binary cross-entropy loss in their multi-label classification problem.

Data Augmentation

After trying to implement the network with only 560 training samples, I used rotated images in 3 orientations other than the original to obtain 4 different images per image. This is logical because of the orientation of images in remote sensing. This would not be possible in nature scene images since rotating a nature scene image by 180 degrees would not give a meaningful result whereas it is very likely to meet a 180 degrees rotated version of a remote sensing image.

After using Python for rotating the images for augmentation, I encountered keras' ImageDataGenerator package in which you can freely augment the data using numerous type of options such as rotation, width shift, height shift, zoom, shear transformations, rescale, horizontal and vertical flips, filling newly created pixels etc. This is a powerful tool to obtain many different samples from one sample.



(a) (b)

(c) (d)

Fig. 7: Some examples from ImageDataGenerator

V. Results

The VGG pretrained on ImageNet data was run with different number of fully connected layers. The first one is has 1024 nodes in the last fully connected layer before the softmax function. And it should be noted that all of the activations used in here are ReLU activations.

The validation and training accuracy plots are given for the experiments done. The confusion matrices are also given for the results and we can see that circular/rectangular shaped objects have high accuracy. We can conclude from here that the features learned from ImageNet dataset may correspond to some features found in these classes (such as stadium, roundabout, basketball court)

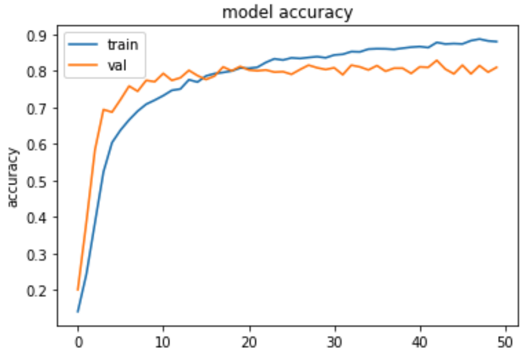


Fig. 8: Accuracy Plot for 1024 FC layer

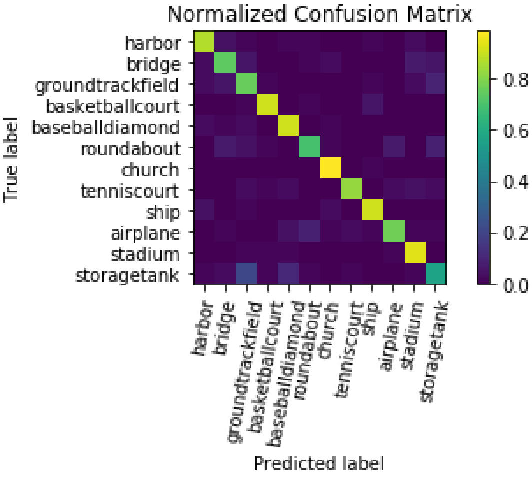


Fig. 9: Confusion Matrix for 1024 FC layer
(Church has a rectangular shape)

The best result is obtained with one 1024 fully connected layer after VGG architecture. Compared to state-of-the-art results given in the dataset proposal paper, it achieves 6% improved accuracy.

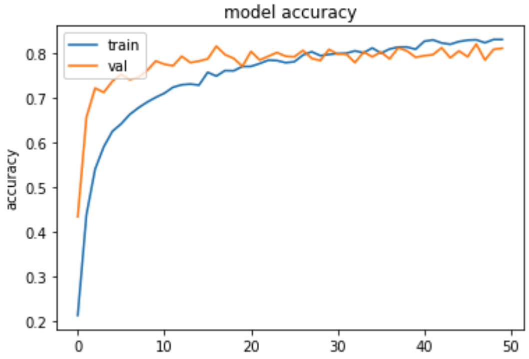


Fig. 10: Accuracy Plot for 512 FC layer

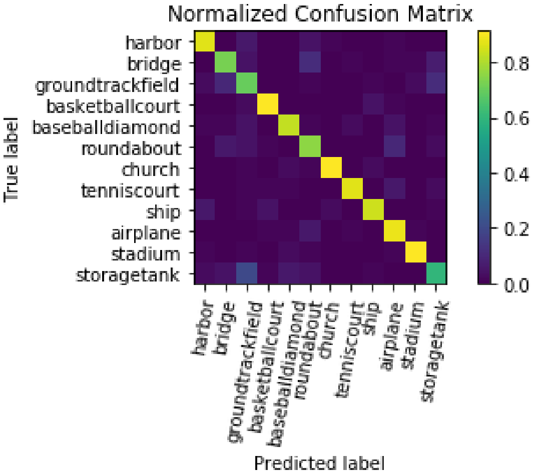


Fig. 11: Confusion Matrix for 512 FC layer

When more fully connected layers are added, the model overfits and the validation accuracy does not improve over the results of one 1024 or one 512 node fully connected layer models. The results and accuracies can be seen in below table. The test set and validation set has the same number of images and the images are randomly picked and selected.

Model	FC	Batch	Learn. Rate	Train Acc	Val Acc
vgg16	1024	32	0.008	87.77	82.86
vgg16	512	32	0.008	85.38	81.52

The results obtained by the models

VI. Conclusion and Discussion

I developed a remote sensing image classification model to classify the images of 12 of 45 classes in NWPU RESISC-45 dataset with very little data. During my experiments, I started with low number of images, (560 train / 70 val / 70 test) and tested transfer learning. It can be concluded that the experiments with VGG network that is pretrained on ImageNet dataset can only reach 90% accuracy after extensive fine-tuning. During my experiments, the accuracy values were between 70 and 82 percent for validation/test sets.

My results show better performance than what authors of [3] obtain when they only apply deep network architectures. Their state-of-the-art results for deep architectures can be found in Figure 2. We should check the training ratio of 10% which is the training/whole dataset ratio, it is different than my setup. It can be seen that my results exceed their results by 5 6% accuracy.

However, when they fine-tune their VGGNet approach, the state-of-the-art result for the model becomes 87.15% for 10% training/whole dataset ratio.

After trying with original dataset, I also try data augmentation and test my results with 57,000 training images instead of 6720 training images in the original dataset for 12 classes. However I only saw 0.69 validation and 0.68 train accuracy with augmented dataset. This shows that augmented dataset also learns as if it was not augmented. I did not use the augmented dataset for my further studies.

In this implementation, I saw that using transfer learning it is possible to learn images in another dimension (such as from nature scene images to remote sensing images) and obtain good classification results. However, the implementation can be further fine-tuned if we check the results of fine-tuned transfer learning on [3] . I also learned how to use weights, save weights of a model, use pretrained models, extensive data augmentation with python and keras and additional keras packages during my project.

We can conclude that the models are not very robust, they achieve a high accuracy (due to the good generalization of features learned from ImageNet dataset), but it is highly likely that the features learned in ImageNet dataset does not really correspond well to remote sensing imagery. The same model can be used with a binary cross-entropy loss on a cat-dog dataset and can achieve accuracies higher than 90% (Also the cat and dog classes are present in the ImageNet dataset).

VII. Code and Models

The code can be found at https://github.com/atakann/CS559-Deep-Learning

Models and weights can be also found at the same address.

VIII. Packages/Libraries Used

The main library used for this project was Keras. Keras is a Deep Learning library for Python. During my experiments I also used numpy, PIL, os, sys, matplotlib, sklearn libraries in python and
in ***keras***,

- utils
- applications
- optimizers

from ***keras.models***,

- Sequential
- model
- load_model

from ***keras.layers***,

- Dropout
- Dense
- Activation
- Flatten

from ***keras.layers.convolutional***,

- Convolution2D
- MaxPooling2D

from ***keras.utils.np_utils***,

- to_categorical

from ***keras.preprocessing.image***,

- ImageDataGenerator

References

[1] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*. ACM, 2010, pp. 270–279.

[2] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon, “Combining satellite imagery and machine learning to predict poverty,” *Science*, vol. 353, no. 6301, pp. 790–794, 2016.

[3] G. Cheng, J. Han, and X. Lu, “Remote sensing image scene classification: Benchmark and state of the art,” *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.

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

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[7] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten, “Exploring the limits of weakly supervised pretraining,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 181–196.