LAND COVER CLASSIFICATION OF HIGH RESOLUTION IMAGES FROM AN ECUADORIAN ANDEAN ZONE USING DEEP CONVOLUTIONAL NEURAL NETWORKS AND TRANSFER LEARNING

Santiago Gonzalez-Toral[*], Victor Saquicela, Lucia Lupercio
Universidad de Cuenca, Ecuador.

ABSTRACT
Different deep learning models have recently emerged as a popular method to apply machine learning in a variety of domains including remote sensing, where several approaches for the classification of land cover and use have been proposed. However, acquiring a suitably large data set with labelled samples for training such models is often a significant challenge to tackle, that leads to suboptimal models not being able to generalize well over different types of land cover. In this paper, we present an approach to perform land cover classification on a small dataset of high-resolution imagery from an area in the Andes of Ecuador using deep convolutional neural networks and techniques such as transfer learning, data augmentation, and some fine-tuning considerations. Results demonstrated that this method can achieve good classification accuracies if it is backed with good strategies to increase the number of samples in an imbalanced dataset.

KEYWORDS: Remote sensing, Transfer learning, Data augmentation

MSC: 93E35, 68T05 .

RESUMEN
Recientemente, ha surgido el uso de modelos de aprendizaje profundo o Deep Learning como un método popular para aplicar modelos de aprendizaje automático en una variedad de dominios, como la detección remota, donde se han propuesto varios enfoques para la clasificación de cobertura y uso de la tierra. Sin embargo, la adquisición de un conjunto de datos suficientemente grande con muestras etiquetadas dificulta el entrenamiento de dichos algoritmos, lo que conlleva a obtener modelos sub-óptimos que no pueden generalizarse bien en diferentes tipos de cobertura de la tierra. Este escenario se presenta a menudo por lo que es considerado como un desafío importante que debe abordarse. En este documento, presentamos un enfoque para realizar la clasificación de la cobertura terrestre en un pequeño conjunto de datos de imágenes de alta resolución perteneciente a una área en los Andes de Ecuador utilizando redes neuronales convolucionales profundas y técnicas como el aprendizaje por transferencia, el aumento de datos, entre otros ajustes a los parámetros del modelo. Los resultados

[*]hernan.gonzalezt@ucuenca.edu.ec

demostraron que este método es capaz de alcanzar una buena precisión de clasificación si está respaldado por buenas estrategias para aumentar el número de muestras en un conjunto de datos desequilibrado.
PALABRAS CLAVE: Teledetección, Aprendizaje por transferencia, Aumento de datos

1. INTRODUCTION

The application of Deep Learning (DL) techniques under different scenarios have started to become very popular in recent years, as they have been reported to tackle many pattern recognition and machine learning challenges that were considered difficult to solve with traditional models [5]. Moreover, their ability to learn and develop hierarchical feature representations from raw data have motivated remote sensing researchers to deliver more accurate state-of-the-art models. Kussul et. al, in [4] showed that a Convolutional Neural Network (CNN) model outperforms other methods, getting accuracies over 85% when classifying crop types using multi-spectral and multi-temporal remote sensing images. Castelluccio et. al in [1] demonstrated that a pre-trained CNN model can be fine-tuned using a satellite imagery dataset (a different domain and visual perspective), and it is able to obtain even better accuracy values (around 6% improvement) when performing semantic classification of remote sensing scenes than training a model from scratch.
In this work, we present an approach to perform land cover classification of high spatial resolution images from an area of the Andes of Ecuador using transfer learning and data augmentation techniques to train CNN models on a small dataset obtained from aerial photography, together with a subset of satellite imagery from Planet[1] to tackle the lack of imagery data in the area. The rest of the article is organized as follows. Section 2. describes the study area and datasets used under this work. Section 3. gives an introduction of deep CNNs for land cover classification, and describes useful techniques and fine-tuning considerations for training such models on small datasets. Section 4. provides an overview of the experimental setup to test the performance of our method. In Section 5. demonstrates and discusses the experimental results, whereas in Section 6. we provide some conclusions and future lines derived from this work.

2. STUDY AREA AND DATASETS

A land cover study was carried out in a local area from the Ecuadorian Andes with a geographic extension of $5.09Km^2$, altitude ranges between 2702.72m-3096.83m a.s.l., and mainly covered by agricultural lands with nearby edification. To create the *UC Dataset* we used an existing RGB *orthoimage* captured using an *aerophotogrametric* flight with a pixel resolution of 9cm. Then, we generated *png* tiles of 256x256 pixels each, while scene labelling was performed using a crowdsourcing session with university students and professionals in agricultural photo-interpretation with previous experience in the study zone. The final dataset contained 12 Level-II[2] standard land cover types, as shown in Table 1.

[1] https://www.planet.com
[2] Ecuador Ministry of Agriculture (MAGAP) states four standard levels of land cover classification. Documentation about the agreement can be found at http://sipa.agricultura.gob.ec/index.php/documentos

Table 1: Dataset summary of land cover types

UC dataset		**Planet dataset**	
Land cover class	No. samples	**Land cover class**	No. samples
Agricultural Land (TA)	1336	agriculture	12338
Pasture Land (P)	826		
Semi-permanent Cultivation (CLS)	15		
Permanent Cultivation (CLP)	3		
Native Forest (BN)	221	primary	37840
Plantation Forestry (PF)	168		
Shrub Vegetation (VA)	676		
Annual Crop Land (CLA)	120	cultivation	4477
Agricultural Mosaic (MA)	46		
Degraded Area (AD)	327	bare ground	859
Building & Infrastructure (IE)	66	habitation	3662
Roads (IV)	29	road	8076
Total	3833	**Total**	67252

Nevertheless, the generated dataset neither provides enough data samples to use it for training a deep neural network model, nor contains a balanced amount of samples per class, which can impact in a model's ability to be generalizable. In order to overcome the lack of data samples on the UC Dataset and still be able obtain a good model performance for discriminating land cover images we applied two strategies. First, we make use of a bigger public dataset published by Planet, which contains images of 3m pixel resolution and 13 types of land cover scenes from different Latin American Countries with Amazon territory including Ecuador, so prior using our dataset for training, we bootstrapped a CNN model using a subset of *jpg* tiles from this dataset. Table 1 shows an overview of the subset of Planet dataset we used and the underlying class mappings between both data sources. Finally, we made image copies of rare scenes in the UC Dataset to achieve a minimum ratio of 10:1 between classes.

3. DEEP LEARNING FOR LAND COVER CLASSIFICATION

The application of convolutional neural networks (CNNs) under different scenarios have started to become very popular in recent years. One of the most popular datasets for experimenting with deep learning models for image analysis was introduced in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [9], whose dataset consists of around 15M labelled images from 22K classes. Such big data data sources allowed deep architectures to characterize the diversity and variability of the training data and to leverage great power in normal image analysis tasks such as classification, object detection, and segmentation. However, there is still a lack of labelled remote sensing data to allow DL models to generalize well over different types of land cover around the world, and more specifically in areas from the Andean region.

3.1. Convolutional Neural Networks

Recent breakthrough developments in the field of deep learning for computer vision can be summarized in a handful of CNN architectures. The earlier *AlexNet* from 2012 [3], together with the *VGG* networks [10], follow the now archetypal layout of basic convolutional neural nets. Later, the *GoogLeNet* [12] stated the idea that CNN layers did not always have to be stacked up sequentially, but rather a horizontal growth of layers leads to wide networks with improved performance and efficiency. He et. al.[2] introduced the *ResNet* architecture that aims to learn residual functions with reference to the layer inputs, and thus distribute the gradient effectively when training deeper networks. Lastly, Xie et. al. [13] proposed a simpler design with a highly modularized architecture of stacked residual blocks called *ResNeXt*, that maintains the same complexity as ResNet, and is more effective than building deeper or wider neural networks.

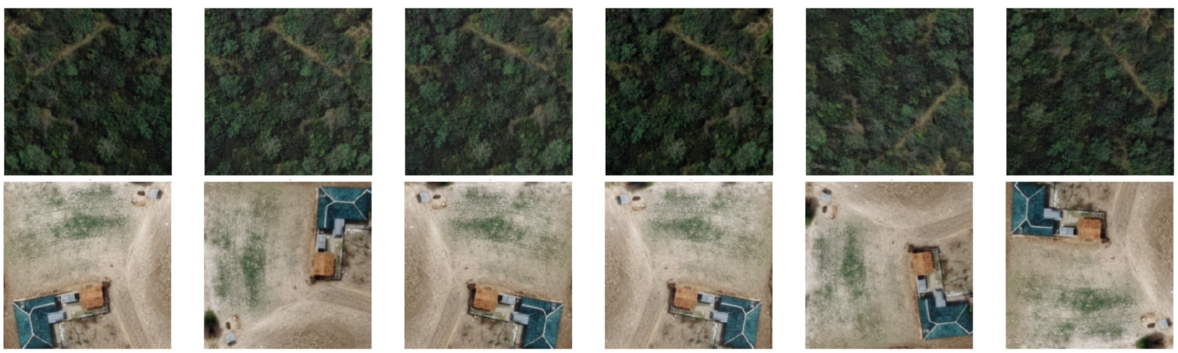

Figure 1: Illustrative image transformations generated during training

3.2. Transfer Learning

Training a Deep Neural Network (DNN) usually requires of a significant amount of training data to obtain a model that both achieve good classification accuracies and does not overfit. Nevertheless, when working on real world image classification problems, it is common to have small data sets, which makes it difficult to train a CNN from scratch. Instead, there already exist some pre-trained CNN models[3] on very large data sets that can be used either as an initialization network for fine-tuning or as a fixed feature extractor. This technique, called Transfer Learning (TL), aims to apply knowledge previously learned to another domain and still obtain good accuracy scores without the need of a large number of training samples. Pan et. al. presented in [7] a survey of how TL can save a significant amount of training effort. Penatti et. al. [8] applied transfer learning on remote sensing problems and concluded that pre-trained CNNs can generalize well even in domains considerably different from the ones they were trained for.

[3]https://pytorch.org/docs/master/torchvision/models

3.3. Data Augmentation

In addition to TL, it is also recommended to use a strategy to *augment* the limited amount of training samples on limited image datasets. Data augmentation is a technique that allows a model to make the most when there is only a few training examples by increasing them via a number of random transformations, so that a model would never see the exact same picture during training. Due to aerial imagery is generally captured from a viewpoint significantly high above the regions and/or objects of interest, it is possible to apply various types of transforms to the seed images. Figure 1 shows an example of augmented images that were randomly generated during training. In our experiments, we empirically applied the following set of image transformations:

1. *Random Scaling*: scale an image using the parameter $zoom_{max} = 1.05$;
2. *Random Rotation*: applies a rotation of 10 degrees with probability $\rho \geq 0.75$;
3. *Random Lighting*: uses *balance* $b = 0.05$ and *contrast* $c = 0.05$ parameters to randomly adjust the lighting of an image;
4. *Random Dihedral*[4]: rotates an image by random multiples of 90 degrees, and then randomly apply a reflection (or flip) in the left-to-right direction;
5. *Center Crop*: crops an image if it is not squared;
6. *Image Normalization*: normalizes the image pixels using ImageNet statistics.

3.4. Fine-Tuning Considerations

Learning rate (LR) is one of the most important and difficult hyper-parameters to tune when training a DNN model as it determines how quickly or slowly to update the network parameters, which can significantly affect a model performance. Smith [11] described a novel method for finding the optimal learning rate, called *cyclical learning rates* (CLR) that eliminates the need to use grid search for hyper-parameter tuning. This method, usually run for one iteration before model training, lets LR to cyclically vary between a reasonable boundary of values until the loss stops decreasing. Finally, optimal LR value is picked within a range where the loss curve decreases, allowing us to achieve better classification accuracies in fewer iterations.

Then, model optimization is performed using a technique called stochastic gradient descent with restarts (SGDR) [6], which gradually decreases the LR value as training progresses. This is considered helpful because as a model gets closer to the optimal weights, it should take smaller updates. However, there might be case when small changes to the weights may result in big changes in the loss. So, to encourage a model to find parts of the weight space that are both accurate and stable, from time to time this method increases the learning rate and forces the model to jump to a different part of the weight space in case it is falling under a local minima.

Finally, when fine-tuning a pre-trained DNN, it is common to use a smaller learning rate on those network layers that have already been trained (i.e. to recognize ImageNet features) in comparison to

[4]Based on Dihedral group definition: https://en.wikipedia.org/wiki/Dihedral_group

the last (fully connected) layers that find features associations and computes the class scores for the new dataset. Therefore, we will use different LR values over three equally-sized set of layers, so the earlier group of layers will get to 3x-10x lower LR than next. This technique is often referenced as *differential learning rate annealing*.

4. EXPERIMENTAL SETTINGS

With the aim to evaluate the effectiveness of transfer learning and data augmentation techniques when training a deep learning model for land cover classification, we design two experiments using different CNN architectures: a 50-layer ResNet and a 101-layer ResNeXt, both pre-trained on the ILSVRC dataset. Table 2 presents the hyper-parameter configuration used to train each model. While the LR value for each experiment was found using the CLR method described in Section 3.4., we empirically set the differential learning rates used to fine-tune the pre-trained models. Finally, batch size was chosen based on dataset size and memory availability.

Table 2: Hyper-parameter configuration for each experiment

Model (lr) SGDR	Dataset Diff. Learning Rate Dropout	Learning Rate Batch Size	Iter.	Cycle length for			
ResNet-50	Planet	0.2	[lr/1000, lr/100, lr]	64	3	x2	0.2
	UC		[lr/100, lr/10, lr]	18			
ResNeXt-101	Planet	0.05	[lr/9, lr/3, lr]	28			
	UC			18			

Due to the characteristics of the Planet dataset, we treated our learning algorithm as a multi-label classification problem. Therefore, the *softmax* activation function at the last layer was replaced with the *sigmod* to represent the probability of an image scene to have a certain type of land cover and atmospheric condition. Experiments were carried out on a computing node with an 8-core Intel Xeon 2.4Ghz CPU, 94GB RAM, and a 12GB NVIDIA Tesla K40m GPU. In general, our training process performed the following steps: (1) Load the Planet dataset and split it on 80% for training and 20% for validation; (2) Resize image tiles to 64x64 pixels; (3) Freeze all CNN layers except for the last fully connected layers (TL), and train the model using SGDR with optimal learning rate and data augmentation for 3 cycles. The number of epochs that will be run before LR is restarted is set by the cycle length multiplier (see Table 2), so after each iteration we doubled the number of epochs the model will be trained. Overall, this step runs 7 training iterations; (4) Unfreeze all layers and set differential learning rates on early layers; (5) Fine-tune the full neural network using SGDR; (6) Repeat steps 2 to 5 using different image sizes: 128x128 and 256x256 pixels. This process is done to avoid overfitting; and (7) Repeat steps 1 to 6 using the UC dataset. Overall, each experiment trained its model using 42 iterations in total.

Models were optimized using a dropout value of 0.2 and a multi-class log loss function, while classification accuracy was measured using the F_2 score, which weights recall higher than precision.

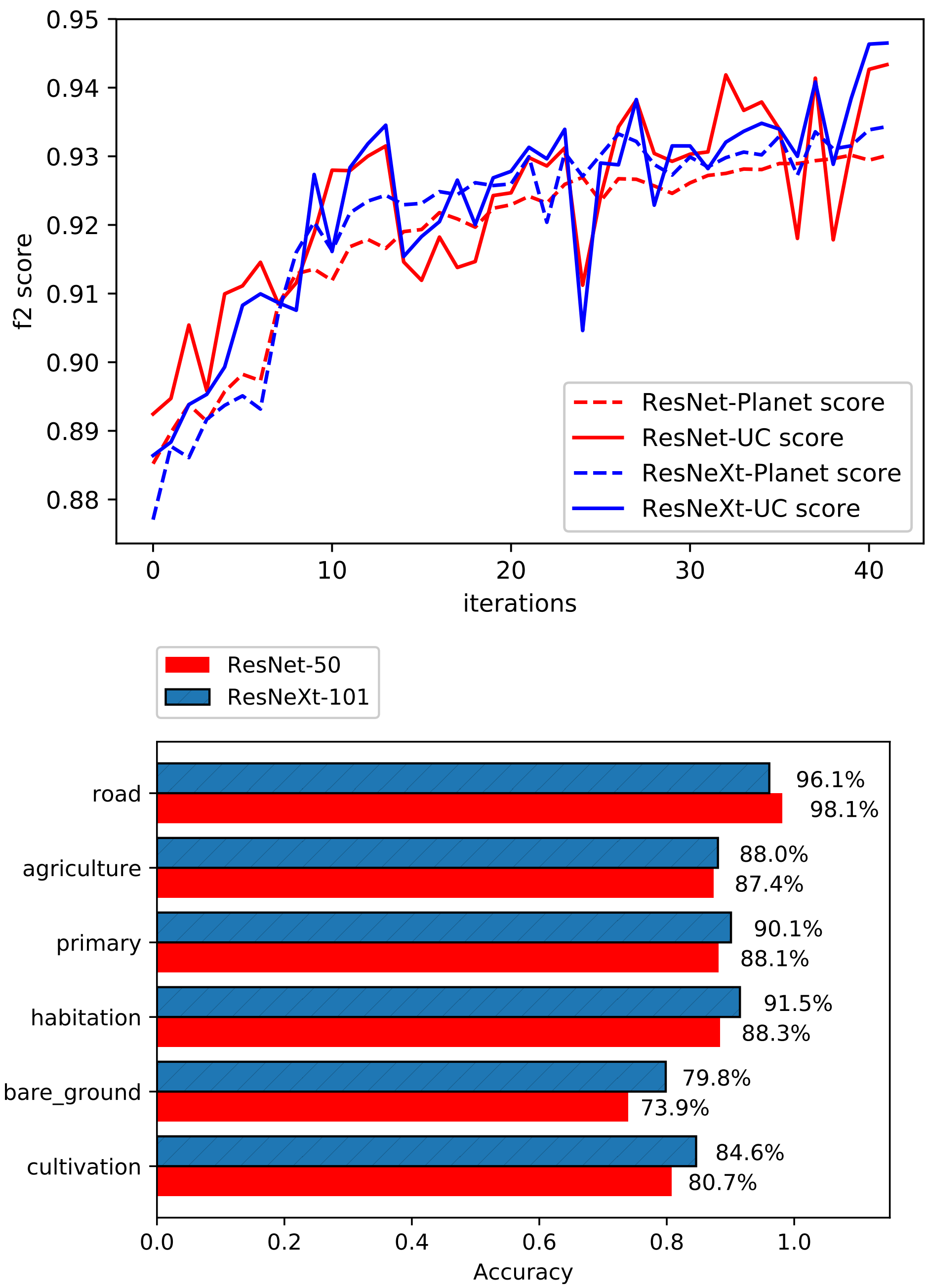

Figure 2: Land cover accuracy evaluation of the validation set. (Left) F_2 scores obtained during model training. (Right) Per-class land cover classification accuracies on the UC dataset

5. RESULTS

We evaluated the land cover classification performance of each setup using both Planet and UC datasets. Figure 2 shows the learning curve during training (left) and the per-class accuracy (right) obtained by each experiment. As can be seen, the ResNeXt model slightly outperformed the ResNet in terms of F_2 score on both the model trained using the Planet dataset only, and the subsequently fine-tuned CNN with the UC dataset. However, training behaviour was not very stable when using the latter mainly due to the lack of samples on certain types of land cover. An evaluation on the validation set was performed using a technique called *Test Time Augmentation* (TTA), which not only makes predictions on the original image scene, but also on four randomly augmented versions of it, to then take the average as the actual result. Table 3 tables summarizes the classification results for both datasets, where the ResNeXt-101 model achieved a maximum score of 94.15% on the UC dataset.

Table 3: Land cover classification accuracies (F_2 score) obtained by the ResNet-50 and ResNeXt-101 CNN architectures

Planet Dataset		**UC Dataset**	
ResNet-50	92.88%	ResNet-50	93.82%
ResNeXt-101	93.13%	ResNeXt-101	94.15%

On the other hand, the per-class breakdown analysis showed that both models achieved accuracy scores above 80% in almost every land cover type except for *bare ground*. ResNeXt produced better accuracy values (2.6% in average) in *habitation*, *primary*, *agriculture*, *cultivation*, and *bare ground* classes, but ResNet performed better when classifying *roads*. Finally, is it worth to mention that even if the ResNeXt has doubled the number of layers than the ResNet, both CNN architectures used almost the same amount of GPU memory during training, showing the effectiveness of ResNeXt to build deeper networks.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an approach to perform land cover classification of high-resolution images from a small area of the Andes of Ecuador using deep convolutional neural networks, together with transfer learning and data augmentation techniques. In order to overcome the lack of data samples in certain classes, we bootstrapped a pre-trained CNN model using a subset of satellite images from the Planet dataset. However, we had to sacrifice a level of detail in land cover classification in favour of model accuracy. Then, fine-tuning was performed using SGDR method and differential learning rate annealing to finally achieve promising accuracy scores of 94.15% and 93.82% with a ResNeXt-101 and ResNet-50 respectively.

As a future work, we plan to capture more imagery data with an additional colour infrared band from a larger area in order to obtain a more balanced dataset and be able to obtain a more detailed level of agricultural land cover classification. Additionally, we will experiment with other DNN models such as sparse autoencoders for semisupervised feature learning, as well as semantic image segmentation approaches to automate the generation of land cover maps.

RECEIVED: NOVEMBER, 2019.
REVISED: JANUARY, 2020.

REFERENCES

[1] CASTELLUCCIO, M., POGGI, G., SANSONE, C., and VERDOLIVA, L. (2015): Land use classification in remote sensing images by convolutional neural networks **arXiv preprint arXiv:1508.00092**.

[2] HE, K., ZHANG, X., REN, S., and SUN, J. (2016): Deep residual learning for image recognition In **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 29, 770–778.

[3] KRIZHEVSKY, A., SUTSKEVER, I., and HINTON, G. E. (2012): Imagenet classification with deep convolutional neural networks In **Advances in Neural Information Processing Systems**, 25, 1097–1105. Curran Associates, Inc., New York.

[4] KUSSUL, N., LAVRENIUK, M., SKAKUN, S., and SHELESTOV, A. (2017): Deep learning classification of land cover and crop types using remote sensing data **IEEE Geoscience and Remote Sensing Letters**, 14:778–782.

[5] LECUN, Y., BENGIO, Y., and HINTON, G. (2015): Deep learning **Nature**, 521:436–444.

[6] LOSHCHILOV, I. and HUTTER, F. (2016): Sgdr: Stochastic gradient descent with warm restarts **arXiv preprint arXiv:1608.03983**.

[7] PAN, S. J. and YANG, Q. (2010): A survey on transfer learning **IEEE Transactions on knowledge and data engineering**, 22:1345–1359.

[8] PENATTI, O. A., NOGUEIRA, K., and DOS SANTOS, J. A. (2015): Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In **Proceedings of the IEEE conference on computer vision and pattern recognition workshops (CVPRW)**, 28, 44–51.

[9] RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M., et al. (2015): Imagenet large scale visual recognition challenge **International Journal of Computer Vision**, 115:211–252.

[10] SIMONYAN, K. and ZISSERMAN, A. (2014): Very deep convolutional networks for large-scale image recognition **arXiv preprint arXiv:1409.1556**.

[11] SMITH, L. N. (2017): Cyclical learning rates for training neural networks In **Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV)**, 464–472.

[12] SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., and RABINOVICH, A. (2015): Going deeper with convolutions In **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**.

[13] XIE, S., GIRSHICK, R., DOLLÁR, P., TU, Z., and HE, K. (2017): Aggregated residual transformations for deep neural networks In **Proceedings of the 2017 IEEE conference on Computer Vision and Pattern Recognition (CVPR)**, 30, 1492–1500.