Recognition of natural landscape images

David Biard
david.biard@u-psud.fr

Robin Duraz
robin.duraz@u-psud.fr

Samuel Berrien
samuel.berrien@u-psud.fr

Hao Liu
hao.liu@u-psud.fr

Trung Vu-Thanh
trung.vu-thanh@u-psud.fr

Théo Cornille
theocornille3@gmail.com

Areal Team
areal@chalearn.org

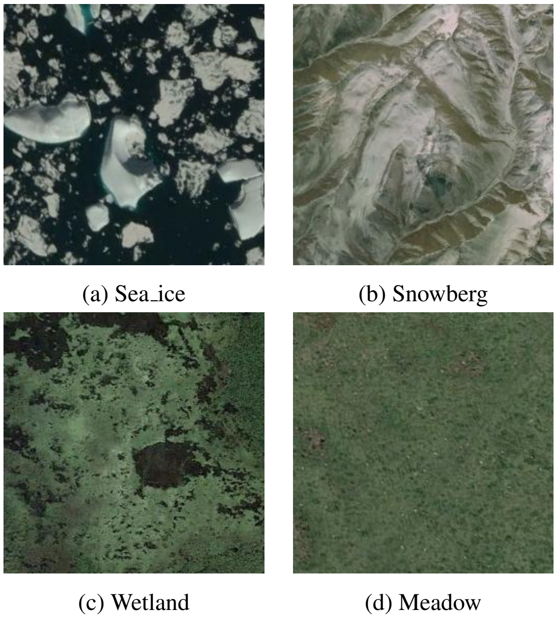
Team's github
https://github.com/ArealTeamM2AIC/

URL raw data challenge: `https://codalab.lri.fr/competitions/379?secret_key=559bdb5c-9517-443d-a71f-791fec9c33a5`
Link raw data: `https://drive.google.com/open?id=1gQYYjmSYsBIAkIKg8q_pgypyKqWwkaXs`
URL preprocessed data challenge: `https://codalab.lri.fr/competitions/378?secret_key=31601eb6-32a8-4e25-82bb-ae91f0f4ffca`
Link preprocessed data: `https://drive.google.com/open?id=1m_dVvuO4tSWKvTH4r-_gOYtCuS51p-Bq`

The goal of our project is to create a challenge about recognizing different natural environments. The dataset NWPU-RESISC45 we use part of, is a new dataset (in 2016), bigger than most in its number of representatives for each class. We decided to make two different challenges, one requiring to work on real images, but being harder, and one requiring to work on preprocessed data, also being a lot easier. From the dataset containing different classes, most of them being human constructions, we only kept 13 classes representing : beach, chaparral, cloud, desert, forest, island, lake, meadow, mountain, river, sea, snowberg, and wetland.
In this report, we describe how we created the two challenges and the methods we used to try solving them.

1 Background

Since aerial imagery services and high resolution appeared, aerial imagery has become of the most important components of various industries. Energy, mining, military situation, disaster management, urban planning and more industries as well as other organizations in emergency situations can make use of aerial images to enhance their productivity and quality of work. There exists many applications in which classifying aerial images can prove to be useful. For example, detecting objects like icebergs or even ships in the sea.
Recent breakthroughs in image understanding techniques using deep learning methods and improvements in hardware like GPUs have opened the way for people to experiment with different approaches and techniques.
Most recent approaches make use of deep learning and convolutional neural networks (CNN) to directly classify or to learn new representations of features, allowing a simpler classification task. Those CNN will try to use imagery techniques like convolutions and pooling to transform the image. The main problem using this approach is that the smaller the dataset, the worse the performance and the less informative the new representation of the features.
We chose to use only part of the original data, for simplicity purposes. We kept 13 classes, but with all available images for the classes. Each class represents a kind of natural scenery, which even for a knowledgeable human, may not necessarily be easy to classify.
Our project tackles the issue of multi-class classification with uncommon classes like wetland, meadow or snowberg as can be seen in 1.
Being able to recognize different landscapes can have many applications, going from following the retreat of glaciers in the arctic to finding how much of an habitat, in which some specific animal species can survive, exists.
In the next section, we will present the data and how we processed it in order to create the dataset for the second challenge. We will also describe methods we implemented in order to resolve the two challenges.



(a) Sea_ice

(b) Snowberg

(c) Wetland

(d) Meadow

Fig. 1: Examples of 4 different classes

2 Related Work

Image classification has always been one of the most challenging problems in computer vision. During the last few years, great success has been made in the task. The ImageNet [14] Large Scale Visual Recognition Challenge is a benchmark for classification and detection on millions of images belonging to hundreds of categories. The challenge has been run annually since 2010, attracting participations from more than fifty institutions. In 2012, ImageNet was the world's largest public academic image classification dataset. The history [7] began with AlexNet [2] which is often considered to be the beginning of the deep learning revolution. Pre-trained on Imagenet(15 million images and 22000 categories), and later trained on ILSVRC-2012 dataset, the network [2] achieves an error rate of 16.4% and won the challenge by a huge margin, since the second team had an error rate of 26.2%. The next year, some small modification was made on AlexNet [2] to make another CNN, ZFnet [15], which won the challenge with an error rate of 11.7%. After that, VGG [3] and GoogLeNet [4], both CNNs, were proposed by Oxford and Google respectively in 2014. They had a very similar performance on ILSVRC-2014. GoogLeNet won the challenge on object classification with an error rate of 6.7%, however VGG won the challenge on object localisation. In ILSVRC-2015, Kaiming He and his team at Microsoft published Deep Residual Networks [5], and defeated all the others with an error rate of 3.57%. Such a result is quite astonishing given that human error is about 5.1% on ImageNet. ResNet [5] was for the first time, a deep learning neural network, which surpassed human level performance. All these winners of ImageNet gave us very good architectures which have already been proven very helpful in all kinds of image classification tasks. In our project, we use some of these classic CNN architectures by doing transfer learning to achieve our objective, as well as creating the preprocessed dataset.

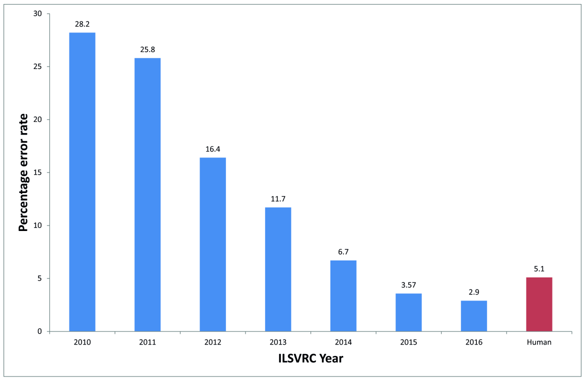


Fig. 2: Accuracy of winners on ImageNet ILSVRC challenge from 2010 to 2016

3 Material & Method

3.1 Original dataset - First challenge

In order to create those projects, we used the NWPU-RESISC45 dataset, which has 45 classes of 700 images per class, each image being a 256x256 RGB image, such as in Figure1. As said before, we selected 13 classes in those 45 classes in order to not have too many classes, and also eliminate all classes related to human constructions.
We kept those remaining 13 classes for the first challenge. The first challenge is thus a classical image classification challenge, most probably requiring the use of deep learning methods in order to achieve satisfying results.

3.1.1 AlexNet

AlexNet is one of the state of the art in deep convolutional neural network. It contains at first five convolution layers and then three fully-connected layers. With the convolution layers, this model is able to learn complex image patterns and has proved its efficient in image classification. Moreover with the used of ReLU activations it prevents the gradient vanishing - problem that often appears in deep neural network - and improves the training performance. As we need to recognize forms or patterns in our data set to deal with classification, this model stands to be appropriate for this tasks. The most relevant application done with this model was on ImageNet and AlexNet has proved that it can achieve powerful results by reaching the top of leader-board. Hence we decided to initialize the model with pre-trained weights excepts the last layer who needs to be re-trained.

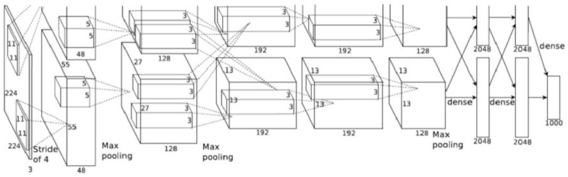


Fig. 3: Alexnet neural network architecture

3.1.2 VGG net

VGG16 model owes its name to its 16 layers. Its architecture is from VGG (Visual Geometry Group) group. It has only 3x3 convolutions layers, 2x2 max pooling layers and fully connected layers at the end. All hidden layers are equipped with ReLU. The multiple non-linear layers increases the depth of the network which enables it to learn more complex features and at a low cost due to their small size. This model exists also for 19 layers and is called VGG19.

3.1.3 Inception V3

Today, there are several high-performance convolutional networks for image classification. Before the birth of inception networks, most popular CNNs just stacked convolution layers deeper and deeper, hoping to get better performance. One of the main motivations leading to the creation of inception networks has been to tackle the problem of kernel size. Indeed, salient parts in the image can have very large variations in size. For instance, if you want to recognize dog pictures, a dog in an image can take a large part of the image, or conversely, only a small part. In any case, it's always a dog picture. Consequently, a large kernel is preferred for information that is distributed more globally, and a smaller kernel for information that is distributed more locally. The idea of inception modules is to have filters with multiple sizes which could operate on the same level. The network would get a bit wider than deeper (remember that the deeper a network is, the greater the risk of over-fitting).

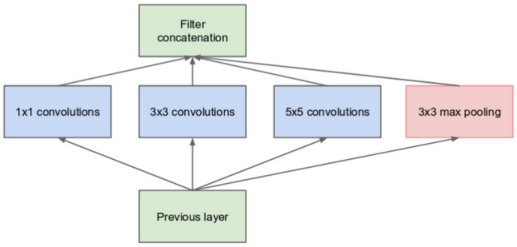


Fig. 4: Inception module

The figure above is what is called an inception module, and that is essentially what the Inception V3 is composed of. Moreover, there are within the network, layers of auxiliary outputs to improve the training phase. Basically, Inception V3 is an optimized version of the firsts inceptions networks, with the addition of batch normalization and some other different tricks to improve the computational complexity (factoring of convolutions).

3.1.4 Deep Residual Network

Deep Residual Learning for Image Recognition [5], written by Kaiming He and his team at Microsoft, is one of the most cited paper in deep learning. Kaiming He and his team observed that simply stacking more convolutional layers in a CNN architecture couldn't improve the result anymore because of gradient vanishing. It can even hurt the performance. The basic intuition of ResNet is shown in Figure 5. ResNet use this kind of shortcut connections (directly connecting input of nth layer to some (n+x)th layer. It has proven that training this form of networks is easier than training simple deep convolutional neural networks and the problem of degrading accuracy is resolved.

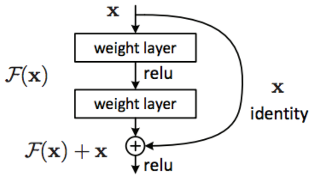

x
weight layer
relu
$\mathcal{F}(\mathbf{x})$
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

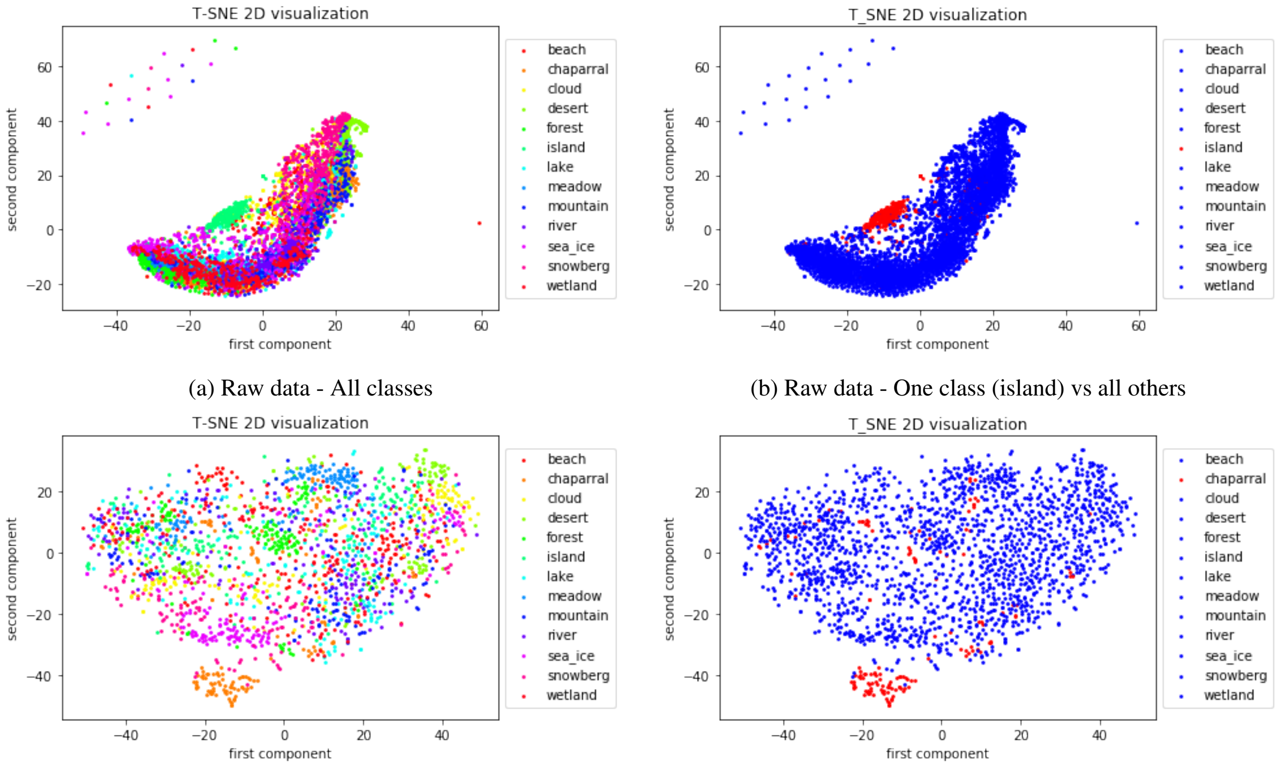
Fig. 5: Residual block

We used a pre-trained ResNet50 from torchvision.modelsn which is a 50 layer Residual Network. There are also other variants like ResNet101 and ResNet152.

3.2 Preprocessed dataset - Second challenge

In order to make the second challenge, using preprocessed data, which is easier to work on for people less knowledgeable on deep learning methods, we used a known pre-trained CNN (AlexNet). This CNN is often used in image classification, and was a good solution for the biggest imagery classification challenge, ImageNet. Its architecture is visible in figure 3. We kept all its original layers, which we didn't train and kept their weights, but replaced the last layer in order to better fit to our data. We then trained this last layer with the aim to learn with our data how to better represent original images.

In order to extract features, we used the representation of the image at the last layer (the one we trained), just before classification. We then created a new dataset from that, with each image having a new representation. Data in this form doesn't visibly have any of the properties existing in images useful for classification like correlations between close pixels, but maybe the information about those correlations still exist in the new features. In that case, CNN and the like aren't better anymore and we can use simpler machine learning models. Results may be a bit worse but results over 60% are much



(a) Raw data - All classes

(b) Raw data - One class (island) vs all others

(c) Preprocessed data - All classes

(d) Preprocessed data - One class (chaparral) vs all others

Fig. 6: T-SNE 2D representation of raw data (a & b) and preprocessed data (c & d)

easier to obtain and don't require as much knowledge.

In the end, we will have two different datasets, one being the original images, and the other being their new representation. The size of the original dataset is about 360MB, whereas the second dataset's size is close to 50MB, for their compressed size.
To divide our dataset in train, validation and test sets, we decided to make it different for both challenges. Deep learning methods make use of large amounts of data so we decided to keep 5200 images as train (400 per class), and 1950 images each for validation and test(150 per class). For the challenge using preprocessed data, we kept 1950 samples for train as well as validation and the rest, i.e. 5200 samples, for test.

3.3 Evaluation metric

Considering that we have balanced datasets, we chose a simple metric, which is accuracy, to determine the classification's quality. It is computed as Number of true positive(correctly classified) in class divided by Number in class. This kind of metric is at the same time simple and informative on the performance for classification tasks where classes are balanced.

4 Results and discussion

We used four classic CNN architectures for the first challenge. They are AlexNet [2], VGG [3], Inception V3 [4], and ResNet50 [5]. The results we got are shown in Figure 7 and in Figure 8 for train set and validation set respectively.

We first tried to visualize our data for both challenges to see if some classes were easily separable. To see that, we did some reductions of dimensions with t-SNE to represent them in two dimensions. Using t-SNE for visualization some classes can possibly be separated from others, but there are nevertheless no clear distinction between classes. We can still make out that data appear much more packed for raw images which can tell that classifying for raw data would probably be harder than classifying preprocessed data. Results can be seen in figure 6. We have done different comparison with one versus all other coloring in order to underline the class representation mixing / detaching.

4.1 Challenge on original dataset

To create the second challenge, we implemented as baseline the model AlexNet. Hence, to solve the problem from raw data, we had to test state of the art models for Deep Learning. We thus compared AlexNet, Inception V3, VGG16 and ResNet on their accuracy for this challenge for a 15 epochs training (figure 7 and 8).

ResNet outperforms others chosen models, reaching 92% in accuracy. To reach this accuracy, Stochastic Gradient Descent was used and was set with a learning rate of 1e-3 and a momentum of 0.09. Once the model trained, we defroze the last layer of ResNet base and train it again with a really low learning rate (1e-6) in order to fine-tune the model.

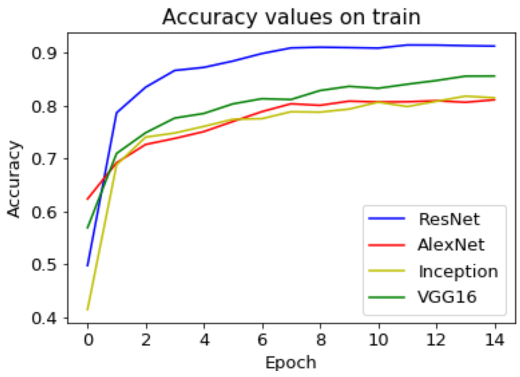


Fig. 7: Accuracy values of different network architectures on train

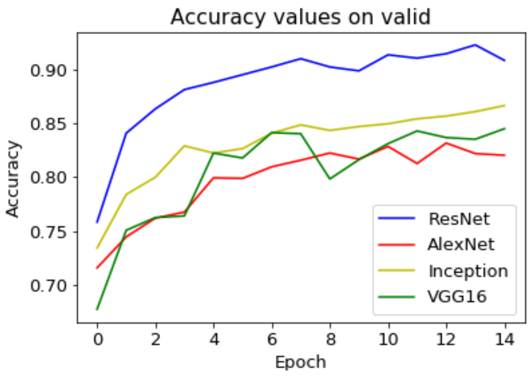


Fig. 8: Accuracy values of different network architectures on validation

4.2 Challenge on preprocessed data

Since we used a CNN to extract this data, we considered that good informations such as correlations between pixels were kept in this new representation and hence, tried solving this challenge using classical machine learning algorithms.
Since we had no idea what kind of algorithm would be better on this kind of data, we tried several algorithms and kept four of them to make a comparison. One of them is the baseline model that we provide with the challenge and was kept as is, to better see the range of performance.
The other algorithms we kept were a Random Forest, a Multi-Layer Perceptron, and a Light Gradient Boosting Machine.
For those three algorithms, we performed Grid search cross validation to fine-tune their parameters and used the best parameters we found to make the comparison. The results we obtained can be seen in Figure 9. As we can see in the figure, even though we optimized by performing cross validation grid search, all classifiers tend to overfit. We can also see that the three models we chose are better than the baseline model by about 25 to 30%, their results all quite close with the MLP classifier nonetheless a bit better. Unlike the baseline model, they achieve the same range of results on valid and test sets. Note that those results are quite lower than the CNN trained on raw data (see figures 9 and 8) this can be explained with the over-fitting on train set.

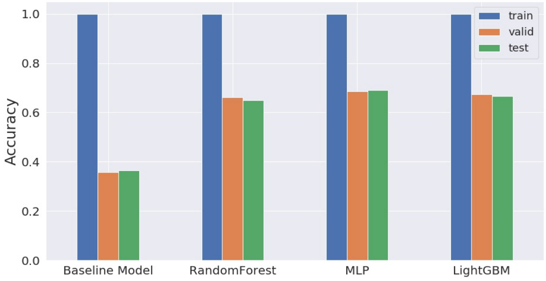


Fig. 9: Accuracy of different models for the preprocessed challenge

5 Conclusion

The goal of the project was to propose, analyze, create and test a data challenge. We chose a multiclass classification problem on aerial natural scenery imagery taken from the NWPU-RESISC [1] dataset. We used a pretrained CNN to extract features and create a preprocessed challenge, easier to solve for students. To solve our own challenge starting from raw data, we experimented many state of the art models in Deep Learning: AlexNet [2], InceptionNet [4], ResNet [5]. In general, we are quite satisfied with the results. The performances of models are just as we expected. ResNet outperformed all the other models with an error rate of 8.4%, but it requires a lot of computing resources. It also takes a longer time to do the prediction. AlexNet and Inception V3 have very similar performance. Both of them achieved roughly about 81% accuracy. Since AlexNet is a much less complicated architecture but still giving a satisfying result. In the situation that computing resources is limited, priority should be given to AlexNet. As for the challenge with preprocessed data, we reached a poorer result, which can also be explained by the fact that we also introduced a bias, and potentially errors while extracting features to create the dataset.

References

[1] Gong Cheng, Junwei Han, and Xiaoqiang Lu, Remote Sensing Image Scene Classification: Benchmark and State of the Art. IEEE International.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ”Imagenet classification with deep convolutional neural networks,” in Proc. Conf. Adv. Neural Inform. Process. Syst., 2012, pp. 1097-1105.

[3] K. Simonyan and A. Zisserman, ”Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Represent., 2015, pp. 1-13.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, ”Going deeper with convolutions,” in Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit., 2015, pp. 1-9.

[5] K. He, X. Zhang, S. Ren, and J. Sun, ”Deep residual learning for image recognition,” in Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit., 2016, pp. 770-778.

[6] Simon Kornblith, Jonathon Shlens, Quoc V. Le Do Better ImageNet Models Transfer Better?, arXiv:1805.08974

[7] Md Zahangir Alom, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S. Awwal, Vijayan K. Asari The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches, arXiv:1803.01164

[8] Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson How transferable are features in deep neural networks? arXiv:1411.1792

[9] Learning Transferable Architectures for Scalable Image Recognition Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V. Le, arXiv:1707.07012

[10] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In ECCV, 2016.

[11] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. CoRR, abs/1506.02640, 2015.

[12] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. CoRR, abs/1506.01497, 2015.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, Mask R-CNN. ICCV, 2017

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei
ImageNet Large Scale Visual Recognition Challenge, arXiv:1409.0575

[15] Visualizing and Understanding Convolutional Networks Matthew D Zeiler, Rob Fergus, arXiv:1311.2901