Enhancing Supervised Machine Learning Output Using Image Processing Techniques

Răzvan DUȚESCU
National Institute of Research & Development in Informatics ICI Bucharest
razvan.dutescu@ici.ro

For the past 20 years, deforestation has been a major issue in Romania. While there have been reforestation attempts, it is still hard to get a clear picture of how the forest situation has changed over the years. This paper explores a possible solution to finding out how Romanian forests have evolved from the year 2000 to 2019 by using geospatial data in order to see where trees were cut down or where an effort was made to replant them. This is achieved by using a decision trees machine learning model and by using clear pictures of the ground as well as some ground variables to determine where a particular forest is. Furthermore, additional steps were taken in an effort to improve the result.
Keywords: *Decision Forest, Geospatial data, Dilation, Erosion, Google Earth Engine*
DOI: 10.24818/issn14531305/25.3.2021.02

1 Introduction

The aim of this paper is to explore the possibility of classifying geospatial data and distinguishing between forests and non-forests. This process can be seen as the training of a classifier to distinguish between 2 classes of pixels based on RGB colors but there are a lot of variables that need to be taken into consideration. At face value, finding a forest on a map could be as easy as finding the darker spots on a map, as these represent trees as opposed to light green which is usually grass or shrubbery. Another important aspect is to distinguish between dark green forest pixels and a dark green pixel that could be potentially different kinds of vegetation such as a farmer's crop. Moreover, there is a difference between a cluster of trees which represent a forest and some trees in someone's back garden. A series of steps have been taken in order to mitigate these problems, from data selection to result manipulation.

There have been multiple attempts to classify forests in the past. These include both supervised and unsupervised methods. Clustering algorithms have been shown to produce poorer results than traditional supervised models [1]. Out of all standard supervised machine learning techniques, Artificial Neural Networks are the weakest out of all, with Random Forests and Support Vector Machines coming out on top in terms of accuracy [2]. People have also experimented with using a combination of multiple data sources when training their classifiers to great effect [3].

2 Tools, environment and maps

2.1 Google Earth Engine

Google Earth Engine [4] is a platform developed by Google used for data science and analysis. It features a variety of satellite images and geospatial datasets that are available for free for anyone to use. It also has its own API developed for both JavaScript and Python that can be used to access and manipulate the data available. This API uses Google's cloud in order to perform computations, meaning that every single operation applied on Earth Engine's maps has to go through their servers. Because of this, problems that arise on the server side are harder to fix. The 2 main problems that arise are computation timeout errors and exceeding memory limit errors.

2.2 Jupyter

Jupyter is a project developed with the goal of providing open-source and interactive computing across multiple programming languages. The core programming languages supported are Julia, R and most importantly for this paper, Python. Google offers their own free online version of Jupyter called Google Collaboratory. This environment runs in the cloud and stores its notebooks on Google Drive. It

also provides seamless integration with the Google Earth Engine API, making it a better candidate to be used for developing this machine learning model over other Jupyter Notebooks offered by other companies.

2.3 Training data

In order to train a classifier, a dataset must first be created. Because the aim is to classify forests, it stands to reason that the first element that needs to be included is a photographic map of the ground. To that end, the LANDSAT 7 dataset was selected as it fits this need. While LANDSAT 7 is not the newest map that is readily available, with LANDSAT 8 having been launched in 2013, it is nonetheless still being updated and has a much longer timeframe, having been launched in 1999. As with every picture, the main components of this map are the RGB (red, green and blue) channels. In addition to those, there are 4 more channels present in the image. The 4th band is near infrared with a wavelength between 0.77-0.90 μm. The 5th and 7th bands are short infrared with wavelengths between 1.55-1.75 μm and 2.08-2.35 μm. The last band is the 6th band which is the brightness temperature. This band is unique because it was initially collected at a resolution of 120m, having been resampled at 30m afterwards. As mentioned before, it's really difficult to classify vegetation based solely on an image taken from a satellite. To that end, further data needs to be analyzed. A robust dataset which contains numerous surface variables is required since there are a lot of factors that change with forests. Furthermore, this dataset needs to include historical data as far back as the year 2000, in order to cover the timeframe of the LANDSAT 7 dataset. With all these factors in mind, the "ERA5-Land" dataset fits precisely these criteria, as it satisfies all requirements mentioned before. This map is provided by the Climate Data Store, and data collection started in 1981, which is well within the time frame established for the dataset required to train the classifier. In total there are 50 surface variables in this dataset, although not all are useful. For example, since the classifier uses data from the summer, it wouldn't be advantageous to use variables related to snow, such as snowfall or snow albedo, so these variables can be discarded without them affecting the final result in a negative way. Furthermore, variables that have to do with lakes or bodies of water can also be discarded since the main focus is on dry land. The one drawback of this dataset is its pixel size. The LANDSAT 7 dataset has a pixel size of 30 m while ERA5-Land has a pixel resolution of 11 km. Because approximately a third of Romania's surface is covered by mountains and most forests are in these areas, another important aspect that needs to be included is elevation and slopes of the terrain. The Shuttle Radar Topography Mission (STRM) provides an elevation dataset which can be also used to calculate slopes. While this dataset was formed based on data from the year 2000, it will not be an issue since altitude doesn't change over the years.

2.4 Labels

Lastly, a set of labels needs to be formed in order to associate them with the training data. Because the aim is to classify forests, a dataset which depicts them is required. Thus, the COPERNICUS CORINE LAND COVER dataset provides precisely that. This dataset includes numerous number-coded types of land types, including 3 types of forests. The codes are intuitively designed in such a way that it's easy to extract the relevant data and the resolution of the map is ideal. At 100 meter per pixel, this is the same resolution as the LANDSAT 7 maps. The one drawback of this dataset is its fragmentation. The 5 time periods covered by it are called assets and each one covers a different number of years, as shown in Table 1.

Table 1. Time frame of assets

Asset name	Time period covered
1990	1989 - 1998
2000	1999 - 2001
2006	2005 - 2007
2012	2011 - 2012
2018	2017 - 2018

The first CORINE LAND COVER inventory was performed in 1990 and was subsequently

updated in the year 2000. Now it has an update cycle every 6 years, which is why there is a discrepancy in the time passed between the first asset and subsequent ones.

3 Classifier

As mentioned before, this analysis of forested areas is achieved using the Google Earth Engine API. As such, the classifier being used is entirely dependent on the classifying algorithms provided by the library. In total, there are 8 classification algorithms, with 5 of them being a variation on a decision tree. When choosing a classifier, not only the scope of the problem but also the limitations of the API have to be considered. The first three algorithms are Naïve Bayes classifier [5], GMO maximum entropy model [6], and Support Vector Machines [7]. The Naïve Bayes classifier has the underlying assumption that the features are independent, but for this scenario it is not the case. The GMO maximum entropy model is not feasible since it requires significant computing resources, which as mentioned before, can be a problem considering that the processing is done server-side. Lastly, while a powerful algorithm, SVMs limit this problem to a binary classification problem, meaning that it cannot be expanded in the future to work for multiple kinds of vegetation. For the reasons outlined above, the classifier used will be based on a decision tree. The workflow of a standard machine learning application is detailed in Figure 1. Having a single decision tree [8] as the sole classifier poses the problem of overfitting or underfitting it. With so many features, finding the right length of a tree can be very difficult. Too many layers and the tree will only be able to handle the scenarios found in the training data while having few layers makes the classifier too broad. However, a collection of shallow trees which vote on the result can overcome this limitation, as shown in Figure 2. This algorithm of multiple weak trees voting on a result is called a decision forest [9]. There are 3 such algorithms found in the Google Earth Engine library.

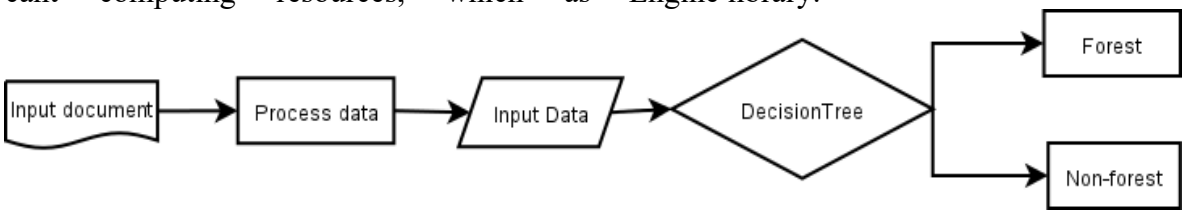

Fig. 1. Basic machine learning pipeline. In this case, "Input document" represents the 3 maps used to extract the data and "Process data" represents the extraction of the relevant features.

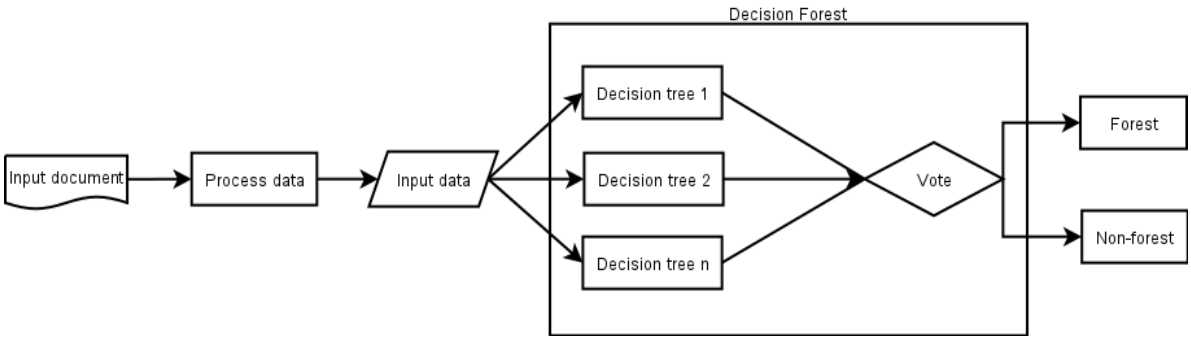

Fig. 2. Machine learning pipeline with a decision forest. Note that instead of each tree classifier deciding, the results are tallied and based on that a decision is taken.

Just classifying the data may not be enough to get an accurate picture of the situation. The main goal is to have smooth blobs that can be further analyzed based on their shape and volume. However, misclassifications can lead to blobs that have bald spots within their borders. There are multiple reasons that might cause this, potentially appearing due to missing data either because of the picture that was taken or as a result of the cloud removal process, or simply because the classifier misclassified that pixel. Not all these spots are not

necessarily wrong as larger spots can also be more open areas typically found within a forest such as meadows or open plateaus on top of the mountains. In image processing, the operation which fills out gaps within a blob is called ***dilation***. Dilation is the process by which pixels are added to the boundary of an object, in this case the forest. These borders don't always have to be external, with enclaves within the object also having pixels added. The number of pixels being added to the object is dependent on the shape and size of the structure going around, called a kernel. The opposite of dilation is called ***erosion*** and it's a process which removes pixels from a blob in a binary image. When these two simple morphological operators are applied in succession, the resulting operation is called ***closing*** [10]. This ensures that the object is returned as close as possible to the original shape while filling in gaps inside of it. Figure 3 shows 2 classification scenarios, one which is ideal and one which is a more likely result due to misclassifications.

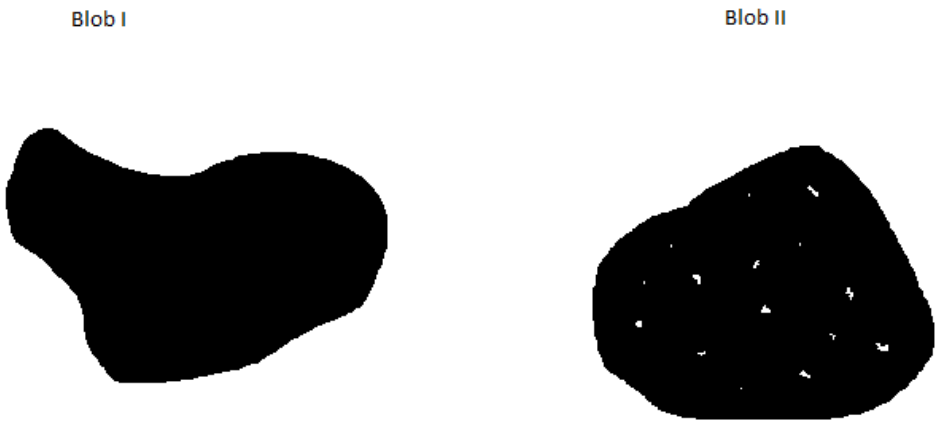

Fig. 3. The 2 possible results. Blob I shows an ideal scenario of a classification. Blob II shows the likely result with bald spots within.

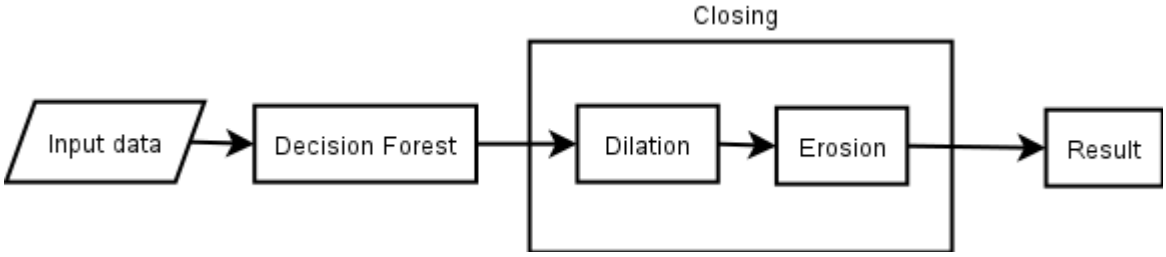

Fig. 4. Final working pipeline of the classifier

Figure 4 illustrates the complete model, from extracting the necessary data to the final result with closing applied on the output of the decision forest. The one drawback of this approach is the distortion of the blob's borders. Following the closing function, fine details around the border will disappear, the blob having a smoother shape all around. There needs to be a balanced approach to this method as well. Selecting a kernel too big or performing multiple iterations may also close those larger spots that were indeed not a forest, as well as distorting the original shape of the result as mentioned before. However, having a kernel that is too small or not performing enough iterations can lead to not closing the maximum number of bald spots within the figure.

4 Testing model components

4.1 Classifier selection

As mentioned above in *section 3*, there are 5 sets of labels that can be used to create the training dataset. Each of the 5 correspond to a dataset and in turn can be used to train 5 different classifiers. Considering that the period

that is of interest is between the year 2000 and 2019, the 1990 dataset can be discarded as it is outside this range while all the others can be considered for this task. Intuitively, the classifier to be chosen is the one with the best training and testing accuracy and precision. However, it does not provide a clear picture of its performance over the whole time period. To get a better understanding of this, each of the 4 remaining classifiers need to be tested against each testing set. The hypothesis is that each classifier will have a strong performance when classifying data from its own time period but the accuracy and precision will drop the further away the dataset is from the original year. This does not include the morphological function applied to the result of the classifier, as we are only interested in the former.

Table 2. Accuracy and precision levels

	2000 Label set	2006 Label set	2012 Label set	2018 Label set
2000 Classifier	**Accuracy:0.9328 Precision:0.8682**	**Accuracy:0.9330 Precision:0.8582**	**Accuracy:0.9295 Precision:0.8539**	**Accuracy:0.9279 Precision:0.8536**
2006 Classifier	**Accuracy:0.9217 Precision:0.8755**	**Accuracy:0.9273 Precision:0.8738**	**Accuracy:0.9248 Precision:0.8710**	**Accuracy:0.9236 Precision:0.8704**
2012 Classifier	**Accuracy:0.9246 Precision:0.8761**	**Accuracy:0.9292 Precision:0.8732**	**Accuracy:0.9313 Precision:0.8484**	**Accuracy:0.9303 Precision:0.8793**
2018 Classifier	**Accuracy:0.9171 Precision:0.8780**	**Accuracy:0.9223 Precision:0.8761**	**Accuracy:0.9232 Precision:0.8793**	**Accuracy:0.9236 Precision:0.8825**

Table 2 shows the testing accuracy and precision of each classifier with each dataset. The training of these classifiers was done over their respective labels set, with 50000 randomly sampled data points. For testing, 10000 entries were randomly extracted from the set. These number of training and testing points were selected in a way to avoid timeout errors from the Google Earth Engine servers. Looking at the table, the original hypothesis is rejected with most accuracies sitting at around 0.93 and precision at 0.87. This means a different method to select the classifier. The 2012 classifier will be used since it is closest to the middle of the selected time frame

4.2 Hyperparameters selection

The random forest classifier has a variety of hyperparameters that need to be tuned in order to achieve maximum efficiency. The 6 variables that go into training such an algorithm are the number of trees inside the forest, the number of variables per split, the minimum leaf population, the fraction of input per tree, the maximum number of nodes, and the randomization seed.

Setting any of these parameters too high or too low can lead to overfitting or underfitting. The first one of these that needs to be chosen is the number of trees. While decision forests are impossible to overfit by increasing the number of trees, having too many can potentially lead to Google Earth Engine server problems. It has been argued [11] that increasing the number of trees over 128 does not yield significant improvements, so that will be the number of trees in the classifier. The randomization seed is only used to generate random numbers when training the classifier and has no influence over the final result, so the default value will be used in this instance.

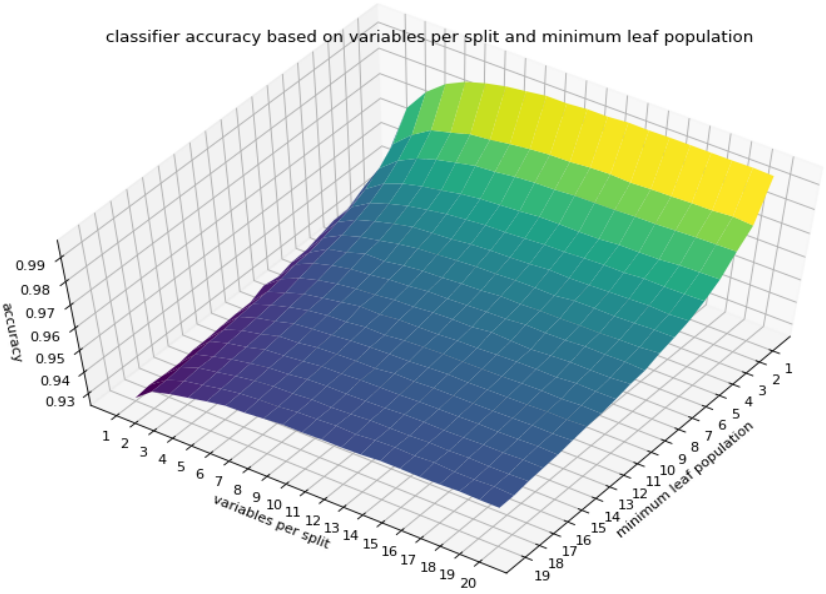

Fig. 5. Evolution of accuracy based on the variation of variables per split and minimum leaf population

Figure 5 depicts the training accuracy of the classifier based on the variation of variables per split and the minimum leaf population. From this plot it can be inferred that the minimum leaf population needs to be minimized for maximum efficiency while the number of variables per split stops influencing the accuracy in a significant way over 11. While with the increase of the latter the accuracy keeps going up, it's only on the 3rd decimal as seen in Table 3, having a small statistical influence overall. However, the increase in the number of variables per split does affect the runtime of the algorithm, making it significantly slower. For this reason, the final algorithm will be trained with the number of variables per split set to 11 and the minimum leaf population set to 1.

Table 3. Accuracy increase with more than eleven variables per split

Variables per split and min leaf population values	**Accuracy**
variables per split = 11 and min leaf population = 1	0.9945275510
variables per split = 12 and min leaf population = 1	0.9945628506
variables per split = 13 and min leaf population = 1	0.9952048473
variables per split = 14 and min leaf population = 1	0.9951045303
variables per split = 15 and min leaf population = 1	0.9953051643
variables per split = 16 and min leaf population = 1	0.9953452911
variables per split = 17 and min leaf population = 1	0.9953452911
variables per split = 18and min leaf population = 1	0.9953452911
variables per split = 19 and min leaf population = 1	0.9955258617
variables per split = 20 and min leaf population = 1	0.9952851009

4.3 Validation of the model

After selecting the parameters of the decision forests, the whole model including the morphological operator needs to be validated. As mentioned in section 2, the model is very sensitive to the size of the kernel and the number of iterations performed on the output. There are multiple possible shapes of a kernel, including a cross shape, square shape and circle shape. Furthermore, there are 2 different options when setting the size of the kernel. The first one is size by pixels and the second one is size by meters. In this case, the latter was chosen since the size of the pixel is 100 meters

and incrementing it would be too much. The aim is to consider accuracy, as well as precision and recall, since the objective is to minimize the false negatives rate, but also make sure the false positive doesn't spike too much.

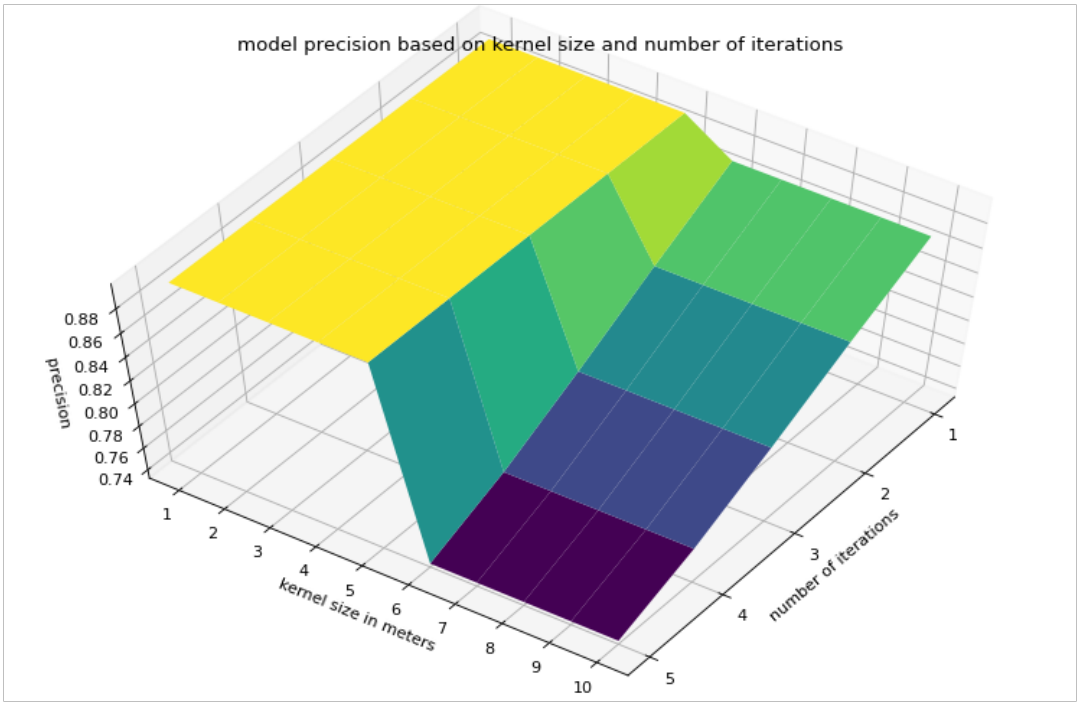

Fig. 6. Precision of the model based on the number of iterations and kernel size

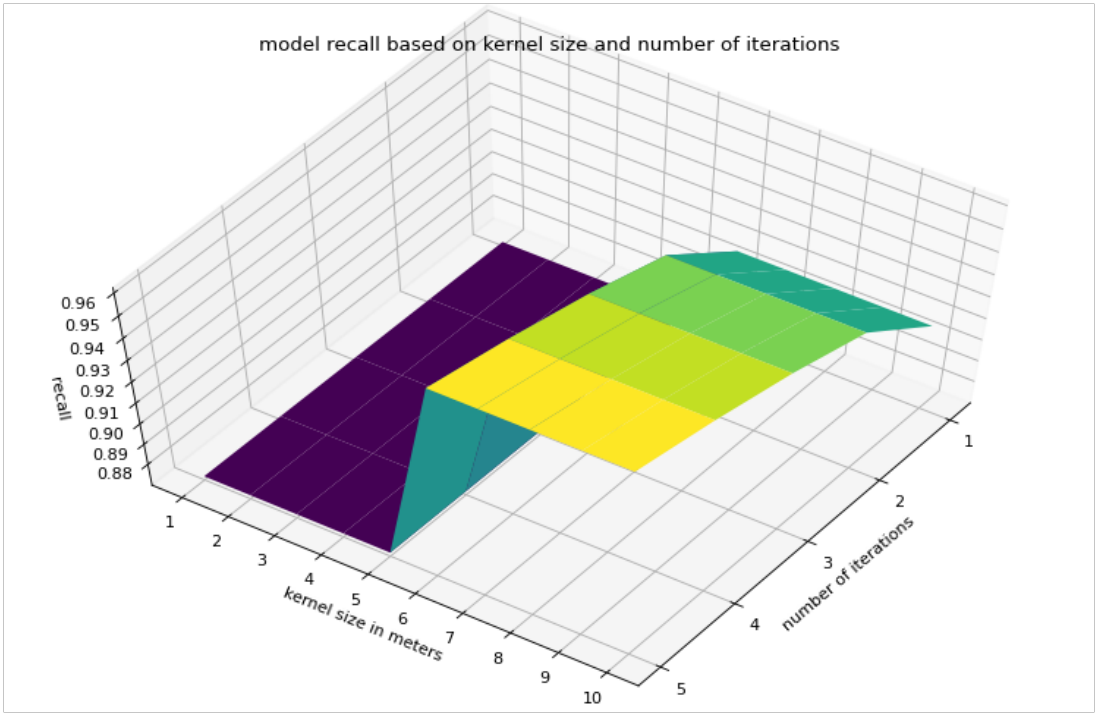

Fig. 7. Model recall based on number of iterations and kernel size

Figures 6 and 7 show, as the precision of the model decreases as the recall increases. This means that the number of iterations and the size of the kernel must be chosen in such a way that both these values are maximized.

The optimal number of iterations is 1 with the size of the kernel between 60 and 100 meters.

5 Analysis of results

With the model validated, it can now be applied to the whole 20 years long dataset. Since the result for each year is a binary image, a 2 bins histogram can be created for each one. With the total land area of Romania being approximately 240000 square kilometers and the size of one pixel set to 100 square meters, the resulting histogram would have over 2 billion elements each. That is way more than Google Earth Engine can handle, so the result would have to be randomly sampled.

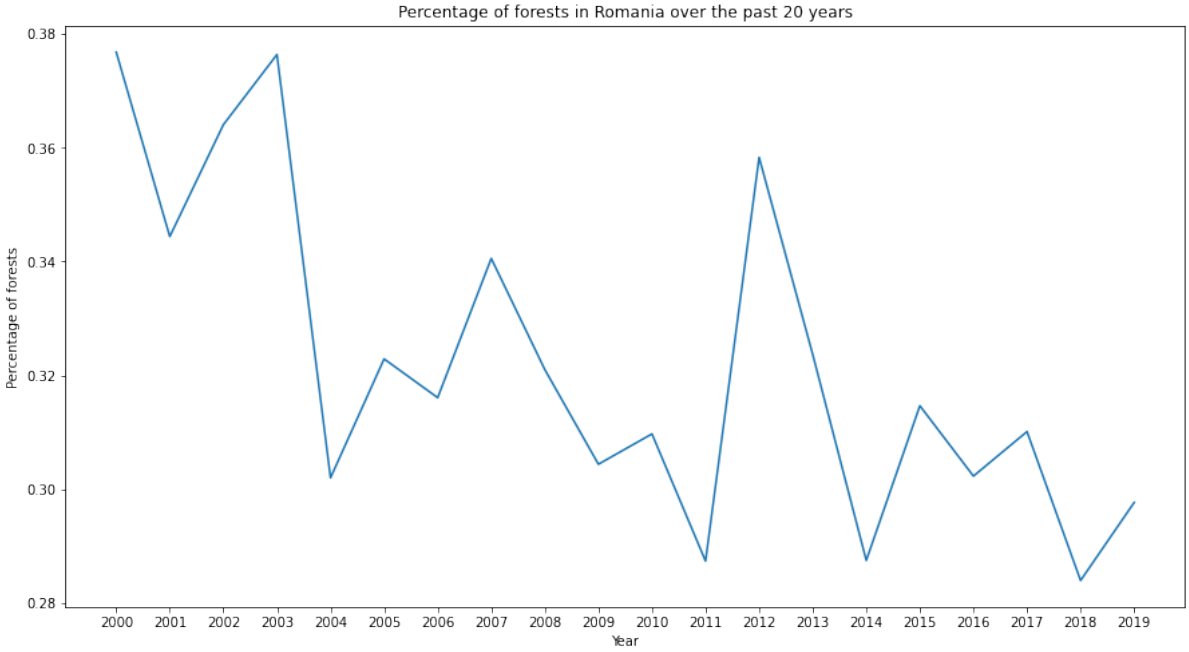

Fig. 8. Evolution of forested areas in Romania from 2000 to 2019

Figure 8 presents the evolution of forested areas in Romania from 2000 to 2019. The graph shows that there is a decrease in the total forested area, going from 0.37 in the year 2000 to 0.27 in 2019. While there are spikes going up and down, the general trend over the past 20 years is that reforestation has not outpaced deforestation. In 2016, the total surface area covered by trees was around 27% [12], with Figure 8 showing that at 32%. With this we can conclude that there is at least a 5% discrepancy between the results of the model and reality. Multiple factors can contribute to this fluctuation in the results such as the random sampling from the output image, the relatively small number of results when compared to the actual number of pixels or the variation in sampling size present within the 3 datasets used. A definitive conclusion cannot be given as to the reason for this inconsistent performance because this matter requires further investigation.

6 Conclusions

To conclude with, we have created a pipeline to classify forested areas on a map using satellite images and ground variables. However, this analysis does not have to be limited to just forests, as it can be extended to find any kind of vegetation, provided that some minor changes are made to the features. Furthermore, while it is a powerful tool, we have seen some of the limitations of Google Earth Engine. The fact that all operations have to be done on their servers instead of local machines means that if a function takes a long time to return a result or runs out of memory, a simple hardware upgrade is not enough to overcome it. Even so, the variety of datasets that it provides is very diverse, making it a powerful source of data despite its shortcomings.

One unique property that this type of problem has is that the position of the pixel on the map in relation to other pixels can also be an indicator to the class it belongs to. For example, a single pixel marked as a non-forested area inside a forest blob is most likely a misclassification and vice versa. This can't be represented in the data to aid with classification so applying a morphological operation to the result of the classifier is a workaround unique to this classification problem.

The results show a downward trend in regards to Romanian forests, even if the variation from year to year is too great to get an accurate picture of the situation.

Acknowledgement

The findings of this article are part of the broader research project PN 19 37 06 01 "Advanced applications of Artificial Intelligence and Big Data" by the National Institute for Research & Development in Informatics – ICI Bucharest.

References

[1] J. Glatthorn,, E. Feldmann, V. Tabaku, *et al.* (2018) *Classifying development stages of primeval European beech forests: is clustering a useful tool?*. *BMC Ecol* 18, 47.

[2] A. Lapini, S. Pettinato, E. Santi, S. Paloscia, G. Fontanelli, A. Garzelli. (2020) *Comparison of Machine Learning Methods Applied to SAR Images for Forest Classification in Mediterranean Areas*. Remote Sens. *12*, 369.

[3] S. Attarchi,, R. Gloaguen. (2014). *Classifying Complex Mountainous Forests with L-Band SAR and Landsat Data Integration: A Comparison among Different Machine Learning Methods in the Hyrcanian Forest.* Remote Sens. *6*, 3624-3647.

[4] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, R. Moore. (2017). *Google Earth Engine: Planetary-scale geospatial analysis for everyone*, Remote Sensing of Environment, Volume 202, Pages 18-27

[5] Hand, D. J. & Yu, K. (2001). *Idiot's Bayes- not so stupid after all?*. International statistical review, 69(3), 385-298

[6] G. Mann, R. McDonald, M. Mohri, N. Silberman, Nathan, D. Walker. (2009). *Efficient Large-Scale Distributed Training of Conditional Maximum Entropy Models*. Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference. 22. 1231-1239.

[7] N. Cristianini and E. Ricci. (2008) *Support Vector Machines*. In: Kao MY. (eds) Encyclopedia of Algorithms. Springer, Boston, MA.

[8] J.R. Quinlan.(1986). *Induction of decision trees*. Mach Learn 1**,** 81–106.

[9] L. Breiman. (2001). *Random Forests*. Machine Learning 45**,** 5–32 (2001)

[10] K.A. Mat Said, A. Jambek, N. Sulaiman. (2016). *A study of image processing using morphological opening and closing processes*. International Journal of Control Theory and Applications. 9. 15-21.

[11] T. Oshiro, P. Perez, J. Baranauskas. (2012). *How Many Trees in a Random Forest?*. Lecture notes in computer science. 7376.

[12] L. Mammadzada, U. Abuzarli, T. Karimov. (2017) *Fondul forestier al României – încotro? Managementul resurselor naturale în România*. In: Competitivitatea şi inovarea în economia cunoaşterii. Vol.1, 22-23 septembrie 2017

Răzvan DUȚESCU graduated from the University of Manchester with a BSc in Computer Science and completed his MSc in Artificial Intelligence at the University of Aberdeen in 2019. His area of expertise includes image processing and unsupervised machine learning. He currently works for the National Institute of Research & Development in Informatics- ICI Bucharest.