Automatic detection of natural slicks in Lake Geneva from a ground-based optical imagery package

ML4Science project, Machine Learning CS-433
École Polytechnique Fédérale de Lausanne

Students: Radenko Pejić, Gojko Čutura, Vuk Vuković
Mentor: Mehrshad Foroughan
Hosted by: Prof. D. Andrew Barry (Ecological Engineering Laboratory EPFL)
December, 2021

Abstract: Slicks are smooth surface areas that often appear in lakes due to various reasons. Ecological Engineering Laboratory (ECOL) acquires an extensive database of RGB images taken by a ground-based imagery system installed on the northern shore of Lake Geneva. An essential task for further research into the slick formation, morphology, and kinematics is their automated detection in a large corpus of taken images. We created a dataset of cropped, resized, and clean images from the raw data. We propose two approaches for solving this problem: image processing with basic machine learning models and a CNN-based image classifier. *Random Forest Classifier* achieved a categorical accuracy of 78.86%, while the *Densenet121* architecture reached a categorical accuracy of 91.73% on the test set.

Keywords: slick, detection, lake, classification, CNN.

1 Introduction

Slicks (smooth surface areas, Figure 1) are ubiquitous in lakes and coastal waters and are known to host higher concentrations of various substances, from living organisms to microplastics [1]. They also induce significant horizontal variability in air-water exchanges of heat, momentum, and CO_2 [2], [3]. Yet, the underlying mechanisms of slick formation and their kinematics are not well understood. Ecological Engineering Laboratory (ECOL) acquires an extensive database of RGB images taken by a ground-based imagery system installed on the northern shore of Lake Geneva. Slicks appear in these images frequently; however, the photos must be categorized in terms of slicks' occurrence before performing any detailed analysis. Such categorization would allow for future research into the effects of large-scale lake hydrodynamics and meteorological conditions on slick formation, morphology, and kinematics.

Figure 1. The distribution of slicks on lake surface water. The image was taken from the ECOL imagery package on the northern shore. Slicks (smooth areas) are marked by a light color appearance on this image.

2 Data cleaning and labelling

A ground-based imagery system is installed on the northern shore of Lake Geneva and takes pictures each minute during the day. However, for this project, images were sampled at the 10-minute interval during the period of 1 year (September 2020 - September 2021). The original dataset contained 27538 raw images in NEF (Nikon Electronic Format) weighing more than 700 GB in total. The data is annotated with additional files containing information about the camera pan and tilt settings for each day.

The data cleaning stage has been done in several steps.

1. **Converting images to JPG (compression)** Although certain Python libraries (e.g. Rawpy) provide an option to convert raw NEF images to JPG, Nikon camera settings are not perfectly inferred resulting in images with bad color settings. Therefore, we used *Apple Preview* and

Nikon Capture NX-D applications for semi-automatic conversion of these images in batches.

2. **Filtering images with appropriate metadata**
   Sometimes camera configuration (pan) changes multiple times during the day. Sadly, we do not have information at which time of the did the change occur. In order to further process the data automatically, images taken on the days with multiple pan values and images without pan values are removed from the dataset.

3. **Cropping**
   Based on the camera pan values of each image, the images were cropped (their height) in order to remove irrelevant parts (trees, hills, house roof, farthest part of the lake).

4. **Removing night images**
   Although the camera is set up to take images during the day, it happens that a few night images without proper lightning end up being taken. They do not contain any useful information being completely black and such images were removed from the dataset using the mean value of the image color and the appropriate threshold.

5. **Resizing**
   All of the images were resized from the width of 6000px to the width of 600px, keeping the aspect ratio. Resizing was performed for the sake of easier data handling since larger images were not beneficial for the models.

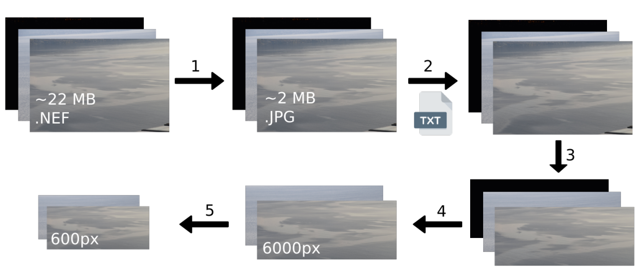

Figure 2. Data cleaning steps

At this stage, the dataset consists of 21090 cropped and resized JPG images weighing a little bit less than 300 MB in total.

After applying and evaluating different unsupervised approaches (clustering, autoencoders) without much success, we have decided to label a portion of the dataset into two classes (slicks and no slicks) and move forward with supervised approaches. A total of 8 months have been labelled which corresponds to 11038 images without slicks and 1968 images with slicks.

3 Machine learning approach

Our first approach applies image processing techniques for generating a few hand-crafted features. These features are then used as an input for basic ML models.

OpenCV [4], *skimage* [5] and *scipy* [6] libraries were used to make the image processing easier. Images were firstly converted to **greyscale** and the **logarithmic correction** was performed. Secondly, **the Hessian matrix** was calculated. In order to extract the patches of the image (which usually appear as brighter regions), the pixels with intensity higher than the threshold (mean of the image) were marked as 1, and 0 otherwise. As a result of applying the Hessian filter, two specific lines always appeared at the top and the bottom of the image. Such **lines were discarded** (the image was additionally cropped). Finally, we apply the ***ndimage.label*** to detect and number all of the connected components (patches). In order to **reduce the noise**, patches with an area smaller than the defined threshold were discarded.

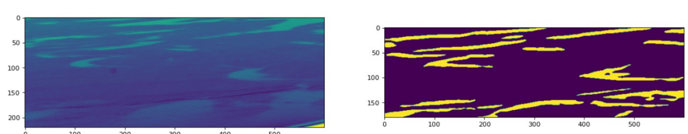

Figure 3. Greyscale image on the left and its final result after image processing on the right

Five features were extracted from each processed image.

1. **Patch number**
   The total number of patches identified in the image.

2. **Total patch area**
   Total (sum) area of all of the patches identified in the image.

3. **Mean diagonal length of the patch's circumscribed rectangle**
   The circumscribed rectangle was calculated for each identified patch in the image. The feature was created as a mean length of all of the circumscribed rectangles' diagonals.

4. **The ratio of the total patch area and the mean diagonal length**

5. **Mean difference of circumscribed circle area and path area**

Logistic Regression, *Random Forest Classifier*, *Support Vector Classifier* and *Gaussian Naive Bayes Classifier* models from *scikit-learn* [7] library were trained for image classification using the aforementioned features. Dataset was balanced to include 50% of images with slicks and 50% of images without slicks. Train-test dataset split was performed in the 70%/30% ratio. Default training parameters provided by *scikit-learn* models were used. Evaluation results of each model are displayed in table 1.

Model	Accuracy	Precision	Recall
LR	77.13%	76.02%	77.45%
RFC	**78.86%**	77.05%	**80.58%**
GNBC	77.13%	**78.73%**	72.65%
SVC	78.56%	77.46%	78.91%

Table 1. Basic ML approach models evaluation

We can conclude that all of the evaluated models perform similarly, with the best achieved categorical accuracy of 78.86%.

4 Deep learning approach

For a more advanced and promising approach to solving this problem, we decided to employ several state-of-the-art image classification models which are based on Convolutional Neural Networks. The models which we applied and evaluated from the *PyTorch* [8] library are *AlexNet*, *VGG16*, *ResNet18* and *Densenet121*. More specifically, we used these pre-trained models and fine-tuned them for our use case by replacing the last, classification, layer with a linear layer with a single output. The output of this layer is interpreted as a probability of an image containing the slick. Average binary cross-entropy loss function and Adam optimizer were used in the training.

Since the pre-trained models were utilized, training lasted only a few epochs. It seems as this problem is ”too easy” for such large and deep convolutional neural networks. In order to avoid overfitting, we performed **early stopping** - if two epochs occur, with average validation loss values lower than the best validatoin loss value until that point, we end the training phase.

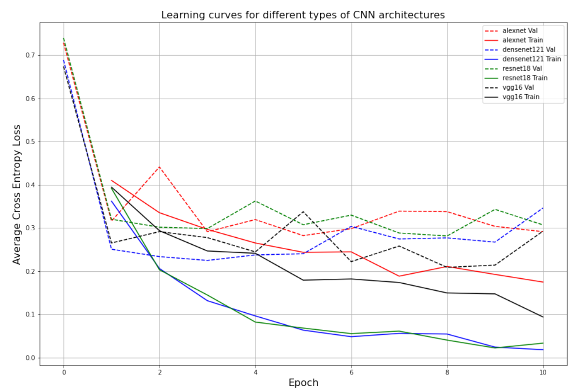

Figure 4. CNN learning curves

Figure 4 provides learning curves of different models throughout epochs. The initial, epoch zero, is actually the evaluation of the pre-trained model, as a sanity checks. In these examples, early stopping was not used, for the sole purpose of generating the complete learning curves.

Dataset was balanced to include 50% of images with slicks and 50% of images without slicks. We performed a 70%/20%/10% train-val-test split. Evaluation results of each model on the test set are displayed in table 2.

Model	Accuracy	Precision	Recall
AlexNet	88.04%	91.18%	84.22%
VGG16	86.39%	79.67%	**97.71%**
ResNet18	90.97%	**91.28%**	90.58%
DenseNet121	**91.73%**	90.20%	93.64%

Table 2. Deep CNN models evaluation

5 Discussion

As expected, deep CNN-based models performed significantly better than the basic ML classification models using the handcrafted features.

In the Error space analysis, a few specific groups can be observed from the error space analysis of both approaches.

- **Boat passing**
  Boat passing leaves a trail behind which can be falsy detected as a slick.

- **Inconclusive examples**
  Some of the examples which are not conclusive can be observed in both groups of false positives and false negatives. Those are usually the examples that even we, during labelling, were not sure whether it is a slick, wave, reflection or something else.

- **Waves**
  Wave formations can create a surface that can be mixed for the slicks.

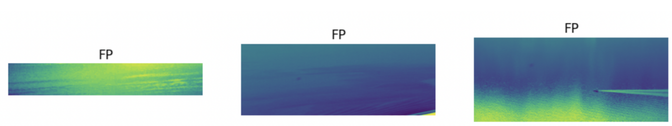

Figure 5. Examples from error space analysis

Figure 6 depicts the distribution of decision confidence for *ResNet18*. As we can see from the histograms, our model is very confident when it comes to classifying samples into our two classes. We infer this by the fact that the distrubutions of the true positives and true negatives are both largely skewed towards 1 and 0, respectively.

6 Conclusion

We have successfully performed data cleaning and labelling getting a dataset that can be utilized for further training and research. We implemeneted and evaluated two approaches to solving slick detection problem, hand crafted feature extraction with basic ML classifiers and a more advanced, deep learning, CNN based image classification. Respectively from each group, the best models achieved categorical accuracies of 78% and 91%.

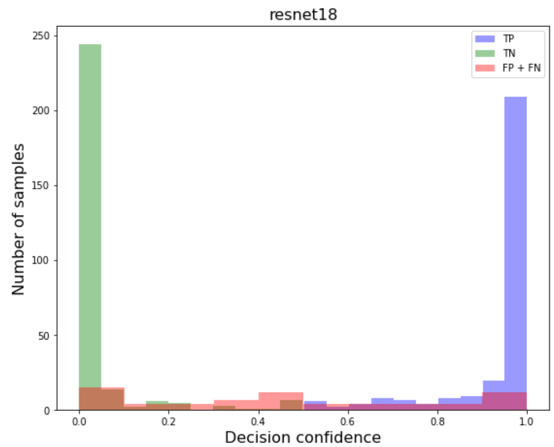

Figure 6. Distribution of decision confidence

Suggestions for further research

Apart from detecting whether slicks exist in the image or not, detected slicks can be classified based on their morphological properties into several groups: randomly distributed patches, streaks, filaments, eddies, aggregates, etc. We have tried to accomplish this using an unsupervised approach by creating histogram-like features from the patch areas and diagonals (e.g. one feature represents one histogram bin). Sadly, after performing k-means clustering together with Elbow and Silhouette methods, we were not able to identify clusters' representatives with any specific group. The supervised approach is also expected to have a better performance for solving this problem.

All of the models that we proposed are time-invariant. Final results may be improved by taking time into the account. For example, one of the possible false positive causes is the boat that passes by leaving a mark behind. In case that the images from the previous hour and the next do not contain slicks, but only a single image in between, this can be an indication that the image was not classified correctly.

The labelled dataset which we created can be used to develop a method of scaling up to large quantities of new unlabelled images, following the semi-supervised learning paradigm.

Thermal images which are also obtained can be further utilized for slick detection and classification since there are possible thermal factors having an influence on their emergence and dynamics.

References

[1] J. M. Gove, J. L. Whitney, M. A. McManus, *et al.*, "Prey-size plastics are invading larval fish nurseries," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, no. 48, pp. 24 143–24 149, 2019, ISSN: 10916490. DOI: 10.1073/pnas.1907496116. pmid: 31712423.

[2] N. M. Frew, E. J. Bock, U. Schimpf, *et al.*, "Air-sea gas transfer: Its dependence on wind stress, small-scale roughness, and surface films," *Journal of Geophysical Research*, vol. 109, C08S17, Aug. 2004, ISSN: 01480227. DOI: 10.1029/2003JC002131. [Online]. Available: https://doi.org/10.1029/2003JC002131.

[3] R. Pereira, I. Ashton, B. Sabbaghzadeh, J. D. Shutler, and R. C. Upstill-Goddard, "Reduced air-sea CO_2 exchange in the atlantic ocean due to biological surfactants," *Nature Geoscience*, vol. 11, no. 7, pp. 492–496, 2018, ISSN: 1752-0908. DOI: 10.1038/s41561-018-0136-2. [Online]. Available: https://doi.org/10.1038/s41561-018-0136-2.

[4] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[5] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, *et al.*, "Scikit-image: Image processing in python," *PeerJ*, vol. 2, e453, 2014.

[6] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[8] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

7 Appendix

Dataset, labelled data, and the trained models are available with approval.

- Dataset
- Labelled data
- Trained models