Machine Learning CS-433: Project 2
Road Segmentation

Diego Fiori , Paolo Colusso , Valerio Volpe
EPFL
Instructors: Martin Jaggi, and Ruediger Urbanke
(Dated: December 20, 2018)

Abstract. The work applies selected machine learning algorithms to classify roads from background on a set of satellite images. Our approach involves a set of distinct models, logistic regression, ridge regression and neural nets, which share some preprocessing steps. For regression, post-processing turns out to be essential and some heuristics are applied to increase accuracy. Emphasis is placed on feature augmentation, achieved by adding noise, filters, clustering and rotations and by taking polynomials, as well as on post-processing, based on road completion and the study of connected components. For neural networks, different models are experimented: one based on the bootstrapping principle, a simple version of the U-Net and two convolutional nets of different depth. Results vary across the methods proposed, as we find neural networks over-perform regression.

I. INTRODUCTION

This work aims at finding a model which classifies road from background on a set of satellite/aerial images acquired from GoogleMaps. The dataset also includes ground-truth images with labels 1 for road and 0 for background. Machine Learning offers promising approaches which can be deployed to come up with novel solutions and we experiment with several distinct models. A first solution concepts exploits regression, while a second approach makes use of neural networks. Both of them make extensive use of preprocessing steps, used to augment the space of features, and post-processing, which tries to solve for unlikely patterns in the output of the classification algorithm.

II. METHODS

A. Data Exploration

The training dataset consists of 100 images as well as their ground-truth versions. Patches of 16x16 pixels were extracted and observations were obtained as the mean and variance of the channels on each patch. The target variable is set to `road=1` if more than 25% of the pixels are 1 and `background=0` otherwise.

B. Preprocessing

We extend the dataset by rotating and flipping the images, increasing the number of images available by 8 times.
As for the feature construction, starting from the three channels available (Red, Green and Blue), further channels are obtained by means of the following:

- image conversion to grayscale;
- Sobel filter, used to sharpen the edges of the image;
- Gaussian-Laplacian filter, which is used for edge detection on the blurred version of the image;
- segmented image, which creates a binary, histogram-based segmented version of the image;
- clustered image, where colours are limited to a specified number based on K-mean clustering.

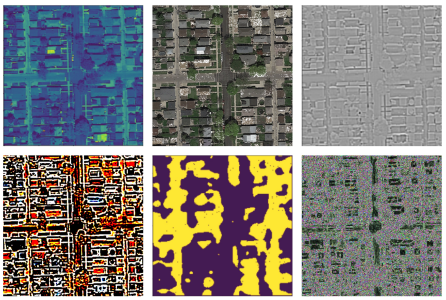

FIG. 1. *Examples of filters applied in feature augmentation. From top left, clockwise: grayscale, clustering, Gaussian and Laplacian filter in grey scale and coloured, histogram-based segmentation and Sobel filter.*

Not all of the filters were shown to perform equally well so that, due to memory issues, some of them were excluded when selecting features.

C. Methods

1. *Logistic regression*

We perform a regularised logistic regression, where the regularising parameter λ is set by a K-fold cross validation.
After obtaining channels of the image as described in the previous section, features are constructed for each patch

by computing the mean and variance of the channel values on the patch. To further expand the feature space, we take polynomials of up to degree two and interactions of the regressors.

2. Ridge regression

Ridge regression performs the same preprocessing steps as logistic regression, but cross validation is here performed on the two hyper-parameters: the threshold, which determines whether the prediction is mapped to 0 or 1, and the regularising parameter λ.

3. Neural Networks

Convolutional neural networks have been adopted in image classification and recognition problems, as they allow to reduce the number of parameters to be estimated with respect to standard neural nets, due to their local approach. Following some intuitions in signal processing, outputs are given by the convolution of inputs with filters. In particular, applying multiple filters leads to smaller output in size, but to more channels as we get deeper into the net, in a pyramidal fashion.
We implement four different neural nets: a simple convolutional neural network, a more complex convolutional neural network with bootstrapping, a U-Net and a deeper CNN.

- The **simple net** involves two convolutional layers with 8 output channels and a ReLU activation function followed by two fully connected layers. Finally, a sigmoid activation is applied in order to obtain an output in $(0, 1)$.

- The **CNN with bootstrapping** produces 10 different models by bootstrapping on the sample images, thus obtaining 10 different predictions. Models are derived from the previous simple net and the final prediction is determined by a majority voting on the models.
  The goal of this attempt is to reduce variance by resampling and hence avoid overfitting.

- The **U-Net** involves a sequence of convolutional layers interspersed with transpose 2D convolutions and normalisation and pooling steps. Specifically, after two convolutional steps, the nodes are stored in a separate variable x_1 and a further step of convolution is performed, before storing the output in a new variable x_2. After an additional convolution step on the output, a transpose 2D convolution (denoted as `convUp`) is applied. This is done twice: after each step the output is concatenated to x_1 and x_2 respectively, taking the smallest of the dimensions. Two final convolutional layers and the sigmoid activation function conclude the net.
  The idea in this case is that copies of partial outputs are stored as the net becomes deeper: this approach aims at targeting the problem of the loss of information which arises with the `convUp`. At the same time, also the issue of the vanishing gradient should be dealt with by incorporating the intermediate results as new channels.
  Table I and Figure 2 illustrates the structure of our version of the U-Net.

- The **Deep Net** consists of a series of 2D convolutions and max pooling and ends with two fully connected layers. With respect to the simple U-Net implemented, the *deep net* features a higher number of channels, but a simpler structure.
  Its structure is presented in Table II.

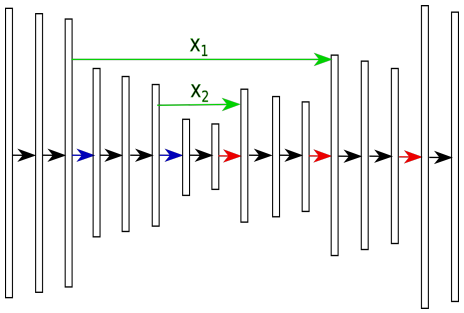

FIG. 2. *Structure of the U-Net: the black arrows represent convolutions, the blue ones stand for max-pool and red ones are transpose convolutions.*

Layer	Channel		Kernel Size	Stride
conv 2D	10	8	3	\
conv 2D	8	8	3	\
Max-Pool	8	8	2	\
conv 2d	8	16	3	\
conv 2d	16	16	3	\
Max-Pool	16	16	2	\
conv 2d	16	16	4	\
Transpose Conv	16	16	2	2
conv 2d	32	16	3	\
conv 2d	16	16	3	\
Transpose Conv	16	8	2	2
conv 2d	16	8	3	\
conv 2d	8	8	3	\
Transpose Conv	8	4	39	\
conv 2d	4	1	3	\

TABLE I. Structure of the U-Net.

Dropout is performed in the last two models as a tool to avoid overfitting other than the one proposed in the bootstrap-like solution concept. This technique involves the selection of a subset of nodes at each training step

Layer	Channel/ Hidden Layer		Kernel Size
conv 2D	3	32	3
Max-Pool	32	32	2
conv 2d	32	64	4
Max-Pool	64	64	2
conv 2d	64	128	3
Max-Pool	128	128	2
conv 2d	128	256	3
Max-Pool	256	256	3
Fully-connected	256	256	\
Fully-connected	256	1	\

TABLE II. Structure of the Deep Net.

and also leads to model averaging. The selected value for the probability of keeping a node was set to $p = 0.5$.

D. Post-processing

After the predictions we try to detect and solve macroscopic errors produced by our methods by implementing some functions in a heuristic way.
Starting from the prediction of the patches' labels we propose the following steps:

1. if most of the patches in a row or column are classified as a road, then the row or column probably has to be classified as a road in its entirety;
2. since a road is supposed to be a large connected component, we remove small connected components;
3. since we have observed that some roads (mostly entire row/column) are noisy, we try to remove that noise (see the images and the codes for a better understanding);
4. we split the image using lines entirely classified as road and we repeat the idea in the first point.

There is not a standard way to adopt these techniques. However, we can infer that the first and the fourth options transform a non-road into a road, so that they work better for methods which tend to predict a small number of roads. The other two are instead better suited for methods which classify a large number of patches as road.
An example of the post processing operations implemented is shown in Figure 3.

III. RESULTS

We find that neural nets over-perform regression methods.
Ridge regression is shown to heavily depend on the

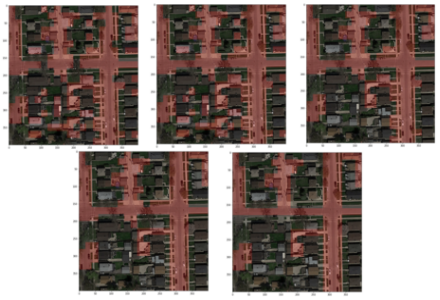

FIG. 3. *Post-processing operations (1), (2), (3) and (3) on the output of a regression. The last two steps act on the horizontal and vertical dimensions respectively.*

choice of the threshold, and less significantly on the λ. Results for different values of λ are presented in Figure 4 and show that the optimal threshold for determining 0 or 1 lies at about 0.28. Regularised logistic regression

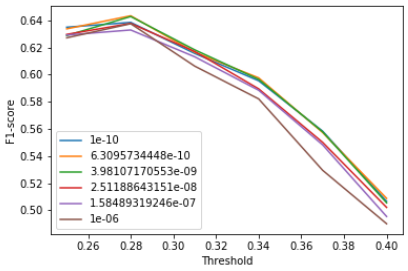

FIG. 4. *Cross-validated F1-score for different values of λ in ridge regression.*

performs slightly worse than ridge regression and its computational time is noticeably higher. Conversely, ridge regression requires cross validation for more parameters.
In both cases, memory and computational power turn out to be crucial as the dimensions of the feature space prevent exploitation of all the constructed features.

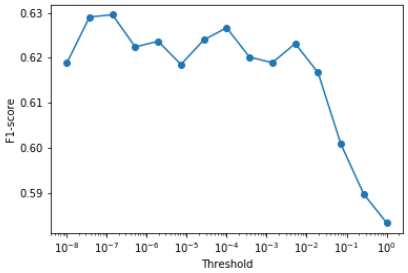

FIG. 5. *Cross-validated F1-scores for different values of λ and threshold in logistic regression.*

Table III sums up results for regression methods

Method	F1	Hyperparameters
Ridge	0.64	$t = 0.28$, $\lambda = 6.3e-10$
Logistic	0.63	$\lambda = 1e-4$

TABLE III. Comparison of regression methods on cross validation.

Method	Accuracy
Simple Net	68.2%
CNN with Bootstrapping	70.0%
U-Net	72.2%
Deep Net	82%

TABLE IV. Comparison of the F1-score for different neural nets.

and corresponding values for the hyper-parameters of interest.

Neural networks tend to perform better than regression methods (see Table IV).
Although computing power limited our choice in terms of depth of the network, we tried to experiment with several solution concepts. The simple net acted as a baseline for further results: this model alone does not achieve results which are distant from regularised regressions, but it can lead to better outcomes if used in a bootstrapping-like setting.
The U-Net in turn allows the F1 score to improve, but is longer to train and is more demanding in terms of computing power.
A better trade-off in terms of resources vs. results seems to be offered by the fourth model, which we called *deep net*.

As far as post-processing operations are concerned, they turn out to be more useful for regression methods, which in general suffer from lower F1 scores. However, a couple of percent points are usually also achieved when applying post-processing steps to neural nets.
An example of the classification results is shown in Figure 6.

IV. CONCLUSIONS

We proposed multiple approaches to the problem of road segmentation, obtaining different results. Regression methods require extensive pre-processing steps: we combined image processing (filters) and machine learning (K-Means) to obtain new channels for the images. Ridge regression performed better than regularised logistic in terms of results and speed. Neural networks obtained higher values for the F1 score, but training them turned out to be more challenging in the case of deeper nets. The model denoted as the *deep net* provided a good compromise between performance and results: while the simple net was not complex enough, the U-Net proved harder to train. The bootstrapping-like model allowed us to explore a possible alternative with a view to reduce variance, but performed slightly worse than the U-Net.
Further developments could allow us to invest more resources (in terms of computing power) in training more complex models: this would be especially more promising in the case of the U-Net, where we were limited in the number of channels to make use of.

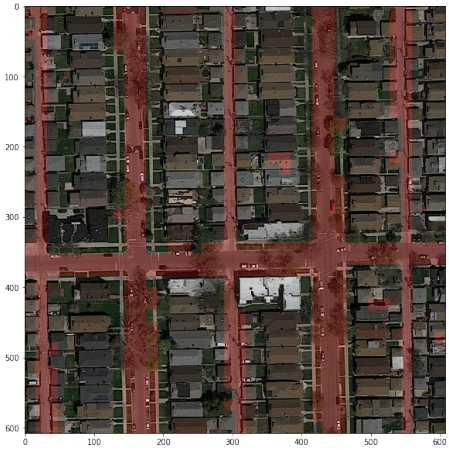

FIG. 6. *Classification performed by the deep net on a sample image before applying the post-processing steps.*

[1] Burger, W., Burge, M. J., *Digital Image Processing. An Algorithmic Introduction Using Java*, New York: Springer, 2016.
[2] James, G., Witten, D., Hastie, T., & Tibshirani, R.. *An introduction to statistical learning with applications in R.*, New York: Springer, 2017.
[3] Ronneberger, O., Fischer, P., and Brox, T., *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 2015.