Road Segmentation on Satellite Images Using Convolutional Neural Networks

Danya Li, Shichao Jia, Zhuoyue Wang
École Polytechnique Fédérale de Lausanne, Switzerland

***Abstract*—In the project, we propose three convolutional neural networks (CNNs) to segment roads from backgrounds on satellite images. We achieve the best results with a CNN model based on ResNet. Image augmentation is implemented to diversify the data set, and regularization measures such as dropout layers as well as weight decay are configured to prevent overfitting. The performance of the model is assessed by the F_1 score, catering for the metric used by the challenge on AIcrowd. The model is able to give decent predictions and some representative results are showcased.**

I. Introduction

In recent years, CNNs have been proved to outperform a number of existing methods in visual recognition tasks, and they have been extensively applied in the fields of image classification, semantic segmentation, object detection, etc. In this project, we are given a set of satellite images and each of them is paired with a ground truth image which is labelled patch-wise as "road" and "background". We are expected to establish and train machine learning models based on the labelled images and make predictions on a set of unlabelled test images.

In this report, we start from a baseline CNN model whose drawbacks give us an idea of how to complicate and improve it. Afterwards, we explain how the image augmentation enriches the data set. Then, we propose three CNN models and demonstrate the best results. In the end, we conclude this project by summarizing our work and giving an outlook on further improvements.

II. Baseline

A. Strategy

The aim of this road segmentation task is to train a classifier to assign a label {road=1, background=0} to each patch of size 16×16. The information contained in one patch is not enough to support training directly over patches since intuitively the width should be at least wider than the road width. By expanding small patches we obtain training patches with applicable size. A patch-size of 72×72 is adopted after several tests. Every 72×72 training patch corresponds to one 16×16 small patch which is located in the center of the training one. The training label is generated from every such small central patch.

B. Baseline Model

For the purpose of evaluating the performance of our models later on, LeNet is introduced as our baseline model for it is small and easy to understand yet able to provide good results. As shown in Table I, it mainly consists of two parts including two convolution layers and three fully connected layers. Trained on the original dataset, as a baseline model, LeNet obtains a validation F1-score of 0.873.

Layer	Parameters
Input	$72 \times 72 \times 3$
Convolution + ReLU	6 filters 5×5
Max-Pool	2×2
Convolution + ReLU	16 filters 5×5
Max-Pool	2×2
Softmax + ReLU	$16 \times 15 \times 15$ to 120
Softmax + ReLU	120 to 84
Softmax	84 to 2

Table I: Structure of LeNet

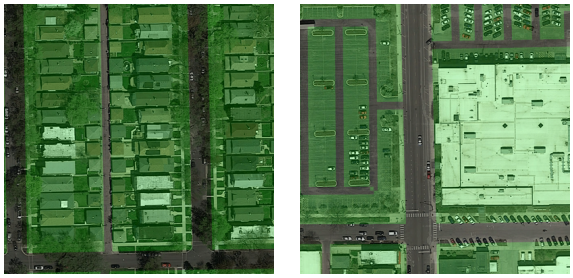
This accuracy shows some drawbacks of this model. LeNet was first designed for handwritten digit recognition which is relatively easy to hit a high accuracy with such a small network. However, in this task, a total of 22 filters may not be able to capture all the features that help identify roads against others with more complicated background, thus LeNet may not achieve very good results. Therefore, in the following parts, improvements on the baseline model and other more advanced models will be introduced.

III. Image Augmentation

A. Exploratory Data Analysis

The training set is composed of 100 satellite images (400 by 400 pixels) of urban areas and their corresponding ground-truth masks where white pixels represent roads while black pixels represent the others.

In accordance with the task to identify patches of 16×16 pixels as roads or not, the ground truths of patches are relabelled as 1 if more than 25% pixels in the original patch are white, 0 otherwise. According to the statistics, their distribution are as follows: 25.9% roads and 74.1% background. Therefore, a model obtaining an accuracy less than 74.1%, which can be realized by marking pixels all black, can be considered as inefficient.



(a) (b)

Figure 1: (a) Roads are blocked by trees (b) Parking lots are not recognised as roads (the backgrounds are marked in green)

Besides, four noteworthy characteristics are found by putting these images under scrutiny:

- Many roads are partially blocked by trees or their shades (see Fig. 1a);
- Roads like sidewalks and parking lots are not labelled as roads (see Fig. 1b);
- For the shape of roads, most of them are criss-cross and parallel or perpendicular to horizontal lines while few of them are tilted;
- For the type of roads, most of them are secondary main roads or access roads while only a few of them are wide arterial roads.

The first two factors may confuse our models to some extent and the last two would cause unbalance in the data set. So further processing of the data set is needed.

B. Augmentation methods

The given data set merely includes 100 pairs of satellite images and their ground truths, the number of which is far from enough for convolutional neural networks, and therefore image augmentation is required to increase the diversity of the data at hand and meanwhile prevent overfitting. We choose geometric transformation as our main approach since it usually introduces a higher degree of variability than altering pixel values [1]. Listed below are three different transformations that we implement,

- `rotation` Rotate the images counterclockwise by 45° and extend the portion beyond boundaries by reflection;
- `flip` Flip the images horizontally and vertically, respectively;
- `shift` Shift the images horizontally and vertically in the same time by 25% and extend the portion beyond boundaries by reflection.

The transformations above are chosen in light of the data set characteristics as mentioned in III-A. `Rotation` diversifies the road orientations, while `flip` and `shift` balance the positional occurrence of some specific road/background elements. With these three transformations implemented, the original data set can be enlarged fourfold. In reality, such a huge data set is too large to handle for the RAM accessible to us, and therefore only 1/4 of these training images (randomly selected) are flipped and shifted.

IV. Models and Methodology

A. Traditional Deep Convolution Neural Networks

1) AlexNet: Compared with LeNet, which is designed for character recognition, AlexNet aims to recognize objects in much larger images, and thus it utilizes larger convolution kernels to capture information.

- **Layers and channels** As shown by Fig. 2, AlexNet contains eight layers including five convolutional layers, two fully connected hidden layers and one fully connected output layer. The number of channels are extended from 3 at the input all the way up to 384 at the 4^{th} convolutional layer. Unlike the original AlexNet, the one that we are using has two output channels, catering for our binary segmentation task.
- **Activation function** The activation function used here is the ReLU function, which makes it easier to train the model comparing with the sigmoid function.
- **Regularization** To alleviate overfitting, dropout layers are placed between fully connected layers with a dropout probability $p = 0.5$.

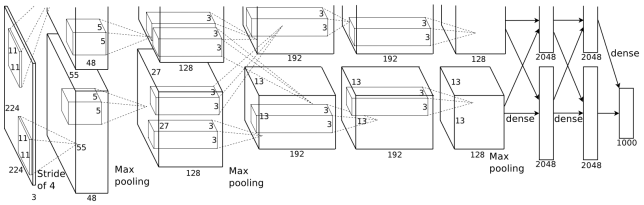


Figure 2: Structure of AlexNet [2]

2) Modified LeNet: Until this moment, two models have been introduced to perform this classification task. One was the very shallow LeNet and another one was the too deep AlexNet, neither of which leads to satisfactory prediction results. Hence, we expect to make a compromise and derive a model between them, thus properly capturing different features. Following are a number of modifications were implemented on LeNet to fulfill this expectation as follows.

- **Layers and Channels** The number of layers and the number of channels in each layer are important parameters that control the topology of network. Based on different trials, one additional layer is added to LeNet and channels substantially increased as well. Number of filters in the each layer increases to 64, 128, and 256, respectively, with a kernel size of 5×5 in the first layer and 3×3 in the rest.

- **Activation Function** For its non-linearity and ability to recover dead filters, Leaky ReLU was added after each convolution layer and the fully connected layer. It is defined as $ReLU(x) = max(\alpha x, x)$, with $\alpha = 0.1$ in our case.
- **Regularization** To mitigate over-fitting, two regularization techniques are applied in our model. The first method is dropout which randomly discards neurons in hide layers then keep the model from being overly dependent on any one of them. It was added after each max-pooling layer with drop probability $p = 0.25$ and after fully connected layer with $p = 0.5$. Moreover, weight decay, also known as L_2-regularization, was also used for weights with $\lambda = 0.01$. These methods prove to be efficient ways to avoid over-fitting for our model but the convergence rate is significantly slowed down as a consequence.

B. Res-Net

In the training process of deep neural network. it is hard to reduce validation error simply by adding more layers. This is the well-known degradation problem. Adding layers does not only make the network more expressive but also changes it sometimes in unpredictable ways. A more efficient structure of neural network is needed to improve performance in classification task. Our final model is based on ResNet which can be trained more effectively by having residual blocks pass through cross-layer data channels [3].

1) Structure: ResNet can achieve better performance on classification task mainly because it introduces the residual block as a basic component in neural network structure. The difference between a residual block and a regular block is shown in Fig. 3. Residual blocks allow for a parametrization relative to the identity function. In practice, the residual mapping is often easier to optimize which leads to more effective training. Using different numbers of channels and residual blocks in the module, different ResNet models can be easily created. After trading off between model complexity and expressiveness, we choose ResNet-34 which contains 34 layers in total.

2) Fine Tuning: The training set given is small which limits classification performance. In this case transfer learning is used to migrate the knowledge learned from the other bigger representative data set to our data set. We implement transfer learning from ImageNet database due to its representativeness and richness. For classification tasks, the same model well-trained on ImageNet can extract general image features that help identify edges, textures, shapes and object composition. Hence, we can use fine tuning technique to train a model towards our data set based on these general knowledge learned from ImageNet. We inherit all model designs and their parameters on the source model, except the output layer, and fine-tunes these parameters based on the target dataset. However, the output layer of the target model needs to be trained from scratch. In our model, every patch is normalized and resized to fit the input format of ResNet trained on ImageNet dataset. The learning rate set for the output layer is 10 times larger than the others.

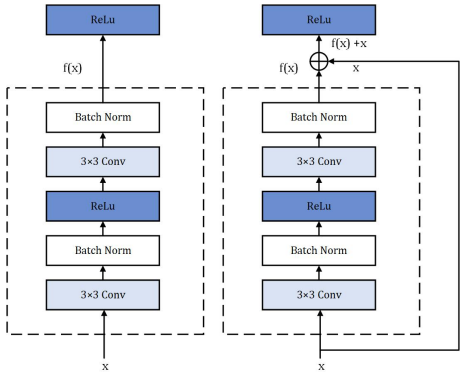


Figure 3: Differences between a regular block (left) and a residual block (right).

V. Metrics & Optimization Methods

A. Accuracy

The model performance is first quantified simply by the precision, i.e., the proportion of classifications that it predicts right. Additionally, another metric called the F_1 score is introduced to the model to assess the classification accuracy as well as the occurrence of misclassification. The F_1 score is expressed as below [4],

$$F_1 \ score = \frac{2PR}{P+R}, \tag{1}$$

where P and R represent *precision* and *recall*, respectively. The F_1 score is within the interval [0,1], and a higher score indicates a higher accuracy.

B. Loss function

We adopt the binary cross-entropy loss as our loss function as it is appropriate for classification problems.

C. Optimizer

We choose `Adam` as the optimizer as it is an adaptive learning rate optimization algorithm tailored for training deep neural networks [5].

D. Validation

Prior to image augmentation, the original data set is randomly partitioned into a training data set (80%) and a validation data set (20%). At the end of each epoch, the optimal weights are tested on the validation data set to assess accuracy. Thus, overfitting shall be prevented and the model generality shall be enhanced.

VI. Results

To draw a comparison among the respective performances of the four aforementioned networks, Tab. II lists the results of these models after training. It is worth mentioning that F_1 **score** shown here is the highest F_1 score that has been achieved on the validation data set during the training. AlexNet, despite its more complex structure, does not show much of an improvement from our baseline LeNet. Since AlexNet was originally designed to perform image classification tasks, the number of channels (up to 4096) in its fully connected layers are probably beyond what is needed for our current binary image segmentation task. Such a surplus in channels may have induced overfitting. Comparatively, the modified LeNet manages to raise the score to 0.919, implying that regularization measures takes effect.

Network	**LeNet**	**AlexNet**	**Modified LeNet**
F_1 score	0.873	0.876	0.919

Table II: F_1 scores of different networks

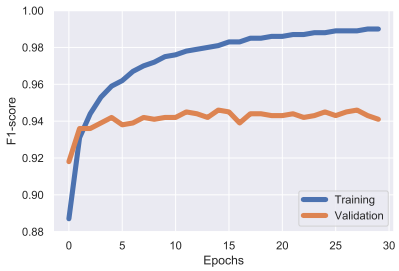
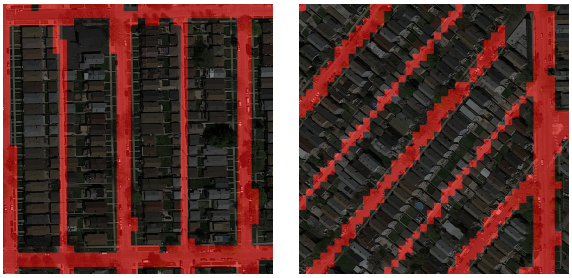


Figure 4: F_1 scores achieved by ResNet on the training and the validation data sets, respectively

Fig. 4 shows the F_1 scores obtained by training ResNet. Our final model, ResNet, shows its superiority over the others by achieving an F_1 score as high as **0.946** on the validation data set with a cross-entropy loss of 0.0258. Our prediction scores **0.891** in the AIcrowd challenge. Such a remarkable improvement can be attributed to the fact that the parameters of the model have been optimized based on the ImageNet data set and that fine tuning should suffice to adjust it for our task.

As shown in Figs. 5a & b, our model gives decent segmentation results either on the satellite image of longitudinal roads or the one with tilted roads. It is noteworthy that the recognition of roads are almost unsusceptible of the cars or the trees on the road. As mentioned in II-A, we expand the training patch size from 16×16 to 72×72, and the additional adjacent information encompassed in the larger patch probably accounts for the robustness.



(a) (b)

Figure 5: Prediction results from ResNet on (a) longitudinal and (b) tilted roads (the predicted roads are marked in red)

VII. Conclusion

In this project, we have build a CNN model based on ResNet which is able to segment roads from backgrounds in satellite images. Our model manages to achieve an F_1 score as high as **0.946** on the validation data set during training, and its prediction scores **0.891** in the AIcrowd challenge. Image augmentation as well as regularization measures such as dropout layers and weight decay are exploited to enhance the generality of the model and mitigate overfitting. Only part of the data set is augmented given the limited hardware resources accessible to us, and a fully augmented set of images will definitely further elevate the performance. Besides, more advanced CNNs designed for pixelwise segmentation can be adopted to realize finer labelling.

References

[1] J. Muñoz-Bulnes, C. Fernandez, I. Parra, D. Fernández-Llorca, and M. A. Sotelo, "Deep fully convolutional networks with random data augmentation for enhanced generalization in road detection," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct 2017, pp. 366–371.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] L. Derczynski, "Complementarity, f-score, and NLP evaluation," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016.

[5] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.