Interactive Learning with Convolutional Neural Networks for Image Labeling

Martin Längkvist, Marjan Alirezaie, Andrey Kiselev and Amy Loutfi
Applied Autonomous Sensor Systems, Örebro University, Fakultetsgatan 1, 701 82 Örebro, Sweden
{martin.langkvist, marjan.alirezaie, andrey.kiselev, amy.loutfi}@oru.se

Abstract

Recently, deep learning models, such as Convolutional Neural Networks, have shown to give good performance for various computer vision tasks. A pre-requisite for such models is to have access to lots of labeled data since the most successful ones are trained with supervised learning. The process of labeling data is expensive, time-consuming, tedious, and sometimes subjective, which can result in falsely labeled data, which has a negative effect on both the training and the validation. In this work, we propose a human-in-the-loop intelligent system that allows the agent and the human to collaborate to simultaneously solve the problem of labeling data and at the same time perform scene labeling of an unlabeled image data set with minimal guidance by a human teacher. We evaluate the proposed interactive learning system by comparing the labeled data set from the system to the human-provided labels. The results show that the learning system is capable of almost completely label an entire image data set starting from a few labeled examples provided by the human teacher.

1 Introduction

Scene parsing (or scene labeling) is an important task within computer vision where the task is to label each pixel in an image. A major challenge in scene parsing is to find solutions that are robust to changes in viewpoint and illumination. Recently, several techniques have emerged to automate this process. One recently emerging technique is Convolutional Neural Networks (CNNs) [LeCun *et al.*, 1998] that has been used for scene parsing [Farabet *et al.*, 2012; Pinheiro and Collobert, 2014; Mohan, 2014; Längkvist *et al.*, 2016; Zhou *et al.*, 2015]. The model parameters in a CNNs uses supervised feature learning to learn the filters and do not rely on human-designed features, which makes them adaptable to the input data. CNNs, and in particular deep CNNs, which consist of several layers of stacked CNNs, have shown to give good performance for various computer vision tasks. However, they have a large number of model parameters to learn and therefore require a large labeled data set. This prevents researchers to try out CNNs on their own unlabeled data sets since they would have to first label their data set or resort to transfer learning [Raina *et al.*, 2007]. In this work, we propose a human-in-the-loop intelligent system that allows the CNN model and the human to collaborate to simultaneously solve the problem of labeling data and perform scene labeling of an entire unlabeled image data set from scratch.

The proposed process involves the human to partially label a few pixels in one or more images from the data set to be used to train a small CNN model. The trained model is then used to assist the human to label more pixels based on the confidence of the model. The human provides feedback during the learning process to guide the model to learn the desired objects. The process is repeated iteratively as more and more pixels are labeled from the human and system to improve the performance of the model in order to correctly label more pixels. This allows the system to improve over time while at the same time the data set is being labeled with minimal guidance by a human teacher.

One advantage of interactive machine learning over traditional supervised learning is that the learning gradually increases in difficulty. Most pre-labeled data sets contain examples that vary in difficulty and may even be subjectively or falsely labeled. In supervised learning all examples are treated equally and considered the ground truth and falsely labeled examples affect both the training and the validation. In many data sets, and particularly for images, a human teacher is capable of pointing out true examples for each class of interest. The manual selection of a few training examples that is used in the proposed framework provides a small but reliable set of easy training examples. The difficult examples are left for later. This process mimics how we humans learn from mistakes by first learning a small amount of knowledge that is then examined and corrected by a teacher. More advanced knowledge is then learned based on previous experiences.

The idea of expanding an intermediate representation via sequential discoveries has previously been done by focusing on the easiest instances first [Lee and Grauman, 2011]. Our work is related to this idea, except that the ranking on the complexity of the instances comes from the human instead of an automatic sorting algorithm. Other previous work on human-in-the-loop for training, classifying, and correcting the classification for scene labeling have mostly focused on using fast algorithms for efficient interaction. The work by [Fails and Olsen, 2003] uses Decision Trees with a subsampling technique to improve training times. Neural networks were also considered for their efficiency but were considered not feasible for interactive learning. In this work, we focus more on the classification accuracy than the training time. However, the structure of CNNs allow for fast per-pixel classification of full images and the use of GPUs can reduce the training time [Strigl *et al.*, 2010] to a feasible level to make them a potential powerful and fast algorithm for interactive learning. Another advantage of using deep models is that they can grow dynamically during learning as the size of the labeled data sets is increased without restarting the learning. This work is, to the authors knowledge, the first work that explores the idea of using CNNs for interactive learning.

We evaluate the proposed interactive learning system by measuring the difference between the finished labeled images that was labeled in collaboration between the human and the interactive learning system and the human-provided labeled ground truth.

The rest of the paper is structured as follows. An introduction to the CNN model and interactive framework is given in Section 2. The evaluation of the framework is presented in Section 3. The conclusions is given in Section 4.

2 Material and methods

2.1 Interactive learning for Scene Parsing

The proposed framework consists of a collaboration and interaction between the human and the agent during the learning process. One iteration consists of the following steps:

1. The user manually labels a few examples from each category.
2. A CNN is trained on the labeled examples.
3. The trained CNN performs a full per-pixel classification of the image.
4. The user decides on a threshold to filter out predictions with a low classification certainty.
5. The filtered predictions are added to the labels.

In more detail, the user starts with a completely unlabeled image that contains several categories. The user labels some pixels in the image. The number of labeled pixels will be few but have a high certainty of being correct. A CNN is then trained on this initially very limited training set. The trained CNN is then used to classify each pixel in the whole image. Due to the small initial training set, most of the pixels will be misclassified. However, pixels that are close to the labeled pixels have a higher classification certainty than pixels further away. Those pixels that have a classification certainty below a certain threshold are re-labeled as unknown since they have a higher probability of being misclassified. The filtered pixels with a high classification certainty are then added to the labels for the next iteration. The user has a chance to correct any misclassifications or label more unlabeled pixels before training the CNN on the new labeled data.

2.2 Interactive learning GUI

The GUI that is used for the interaction between the user and the system is shown in Figure 1. For this demonstrative example we have chosen to use a satellite image. The axes to the left shows the current image with the manually labeled pixels masked on top. The size of the paintbrush and the label is selected in the middle panels. To label pixels in the image the user click-and-drags on the left image. Bottom right axes shows a clean version of the current input image without the user-painted labels. The current image can be changed with the left and right buttons above this axes. The text between the buttons displays the number of the current image and the total number of images in the data set. When the user has finished labeling some pixels on one or multiple images the "‘Train CNN"’ button is pressed and the model parameters of the CNN are updated with training data randomly drawn with an equal class distribution from the labeled pixels. When the model is finished training, the user can click the "‘Inferrence and filter"’ button to perform a full classification of each pixel in the current image. The predictions are filtered with the threshold given by the adjustable slider. The classification result of the current selected image is shown in the top right axes. Finally, the user can get assistance with the labeling process from the trained CNN by pressing the "‘Add predictions"’ to add all filtered classified pixels and the process is repeated.

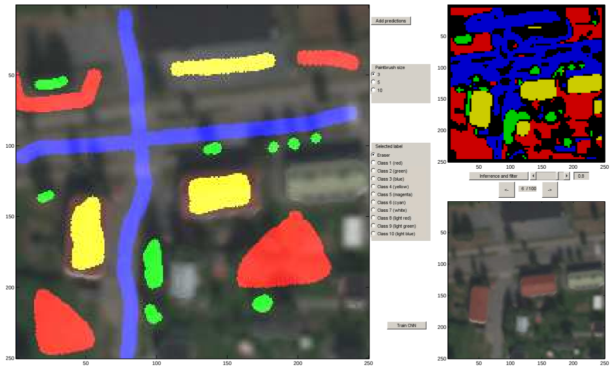

Figure 1: The user has labeled some parts of the image (red = ground, green = tree, blue = road, yellow = building) and trains a CNN using the human-labeled training data and classifies each pixel in the image. The user then decides the threshold for the classification accuracy and adds the CNN generated predictions to the labels.

2.3 Scene labeling using a CNN

In this work, we use a CNN to extract meaningful representations from the raw input image patches. CNNs uses local connections and tied weights to learn feature representations that are slightly translational and rotational invariant, which makes them powerful models for natural structured data (such as image, speech, and video). The process of using a CNN for classifying a single pixel can be seen in Figure 2. The input to the CNN is an image patch x^c, where c is the number of bands in the input (for RBG images, $c = 3$), of size $m \times m$ with the pixel to be classified located at the center and the contextual area. A CNN performs three steps: convolution, non-linear activation, and pooling. The convolution step involves convolving k number of feature maps w^k with size $n \times n$ over the input x^c. The convolutional layer f^k is calculated as:

$$f_{ij}^k = \sigma\left(\sum_c \sum_{a=0}^{n-1}\sum_{b=0}^{n-1} w_{abc}^k x_{i+a,j+b}^c\right) \quad (1)$$

where σ is a non-linear activation function, for example, the sigmoid activation function $\sigma(x) = 1/(1+exp(-x))$ or Rectified linear units (ReLU) $\sigma(x) = max(0, x)$. The pooling step computes the maximum (or the mean) value of the feature maps over each local non-overlapping region. The pooling layer g^k is calculated as:

$$\begin{aligned} g_{ij}^k = \max(&f_{1+p(i-1),1+p(j-1)}^k, \dots, f_{pi,1+p(j-1)}^k, \dots, \\ &f_{1+p(i-1),pj}^k, \dots, f_{pi,pj}^k) \end{aligned} \quad (2)$$

where p is the pooling dimension and $1 \leq i, j \leq (m - n + 1)/p$. The output of the pooling layer is transformed to an input vector to a fully-connected layer. Finally, a softmax classifier is attached to the last fully-connected layer to classify the middle pixel. The process is repeated for each pixel in the image. This can be speed-up by convolving the filters on the whole image and then assign the convolved image to each individual pixel.

3 Experimental results and analysis

This section evaluates the use of interactive learning for the task of scene labeling on three different data sets: human face images, satellite images, and outdoor images.

Human face images This data set is the Yale Face database [Belhumeur *et al.*, 1997], which contains 164 grayscale images of size 243 × 320 of 15 individuals with different facial expressions. The data set was partially labeled manually using the GUI.

Satellite images This data set contain 2500 satellite images of size 250 × 250 pixels with 14 spectral bands and a digital surface model of a small city in Sweden. The data set was reduced to 200 images for class-balance and an optimal subset of 5 spectral bands and the digital surface model is used to reduce the dimensionality. The RGB channels are displayed to the user during the labeling process. The map is fully labeled into five categories (vegetation, ground, road, building, and water). A more detailed description of the data and the process of selecting the optimal spectral bands can be found in [Längkvist *et al.*, 2016].

Outdoor images For images of outdoor scenes we use the Stanford Background Dataset [Gould *et al.*, 2009]. This data set contains 715 outdoor images of approximately size 320 × 240. All pixels in the images are labeled into 8 categories (sky, tree, road, grass, water, building, mountain, and foreground object).

3.1 Experimental setup

For each data set, 10 images are randomly drawn and manually partially labeled by the user. The training set is drawn from the labeled pixels with an equal class distribution to train a CNN. The CNN then classifies all images in each data set and labels those pixels that have above a certain classification accuracy. A new training set is randomly drawn from the new larger labeled data set to further train the CNN. The process is repeated for 10 iterations and the result is evaluated. Each experiment is first run without any interaction between the agent and the human except for the initial labeling. The threshold is set fixed at 95% for this mode. Then the experiment is run with a simulation of human interaction after each iteration. The interaction consisted of removing falsely classified pixels.

The same CNN structure is used for all three experiments. The layers of the CNN model consist of: one input layer of size 25 × 25 pixels; one convolutional layer with 20 filters, filter dimension 11 × 11 pixels, and sigmoid activation function; one pooling layer with pooling dimension 5; 2 fully-connected layers with 100 hidden units each and using sigmoid activation function; and finally a softmax layer using cross-entropy loss. The input size, filter dimension, and pooling dimension were chosen so that the size of the output from the pooling layer is a patch of size 3 × 3, where the middle value is a representation of the surrounding area close to the pixel to be classified and the outer values represent the context area around the pixel to be classified. Training of the model parameters is done with supervised backpropagation of the whole network for 3 training epochs using minibatch stochastic gradient descent (SGD) with 128 training examples per minibatch and momentum.

3.2 Results

Figure 3 shows the result on one image from each of the data sets without any interaction from the human except for the initial labeling. Top-left figure shows the learning of 6 categories (mouth, eyes, nose, hair, face, and background). The second column shows the initial labeling done by the user. For this face image, only the left part of the face is labeled. During each iteration, the CNN fills in the missing labels. It can be seen that the CNN first learns about the easy background class and then the rest of the classes. After 10 iterations most of the image has been labeled. There is some misclassification between hair and face on the right side which is probably caused by shadows and the illumination.

The second row shows the classification of a satellite image. After 10 iterations almost the full image is labeled. Notice how the building with the white roof in the middle-right eventually gets labeled even though the user only labeled buildings with red roofs.

The third row shows the result from one image from the Stanford background data set. This illustrates well how the CNN confidently fills in unlabeled pixels near previously labeled pixels and saves the more difficult cases (e.g. object borders) for later.

Figure 4 show the classification accuracy and the percentage of classified pixels on the testing set for the Stanford data set after each iterations without interaction. After 10 iterations, 89.83% of the data set is labeled with a 81.14% accuracy. This is a promising result, given that only 10 images where partially labeled by the user and no intervention was done during learning. The result could be improved by more

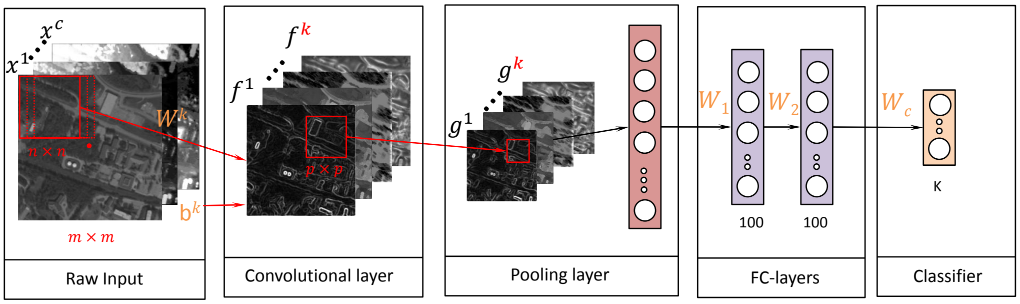

Figure 2: Overview of the process of using a CNN for per-pixel classification. The model parameters to be learned are marked in orange. The CNN architecture parameters are shown in red.

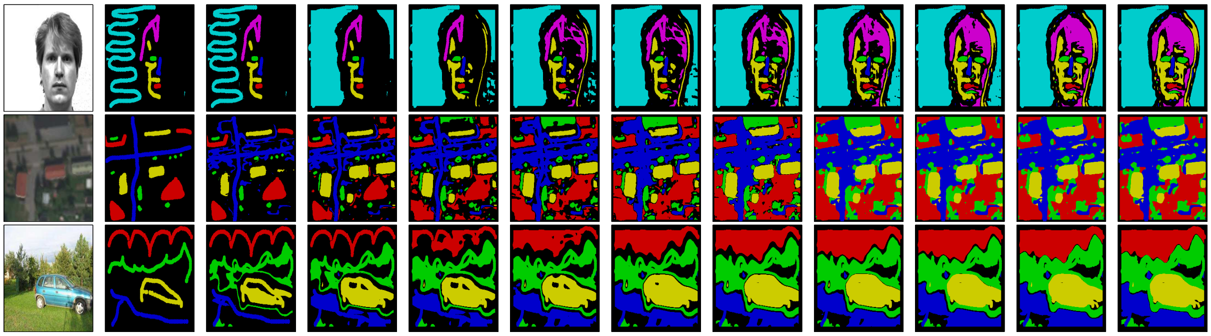

Figure 3: First column show the input image from three different data sets in each row. Second column show the human-labeled initial labeling. The rest of the columns show the labeled image after each iteration after pixels above 95% classification accuracy have been added.

human interaction during learning and possibly with more experimentation with the CNN architecture parameters.

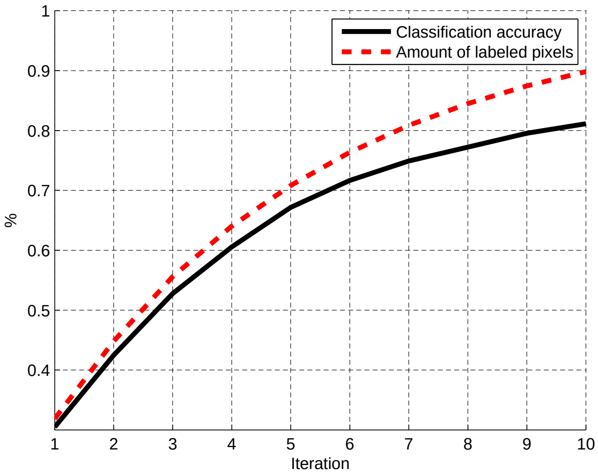

Figure 4: The classification accuracy (solid black line) and percentage of labeled pixels (dotted red line) for the Stanford data set after each iteration.

To evaluate the influence of interactivity, we simulate a situation where the human user removes all the falsely predicted pixels. This is done by utilizing the known labels by comparing the predictions and the ground truth and only add the correct predictions. For a real scenario, where the ground truth is unknown, this feedback is replaced by a human. Table 1 shows the final classification accuracy after 10 iterations on the testing sets for the three data sets with and without using the simulated feedback from the human. For all data sets, the accuracy is increased slightly with interaction. We suspect that the lower accuracy for the satellite image data set is due to a high amount of mislabeled pixels caused by annotation ambiguity. The increase of accuracy for the human face image data set is very low probably due to the images were not fully annotated.

4 Conclusions

In this work we have shown how a CNN can be used for interactive learning for the task of image labeling and scene parsing. The proposed method offers an intuitive collaboration between a human and the agent where the learning process is supervised by the human teacher and allows for bi-directional feedback; both from the model on what classes it struggles with and from the user by guiding the model to correctly learn

Method	Without interaction	With interaction
Human face images	75.5	76.3
Satellite images	64.2	66.5
Outdoor images	81.1	84.7

Table 1: Classification accuracy [%] after 10 iterations from an initial labeling from a human and with no interaction (left column) and with interaction between the human and the classifier after each iteration (right column).

the desired classes.

Future work include reducing the training time with code optimization, GPU implementation, and introducing an online-learning component so that the CNN model is constantly training in the background while the user is labeling in order to properly evaluate if the proposed method can reduce user-fatigue during labeling.

Another future direction is to look at dynamically changing the size of the CNN model (adding more filters and layers) as the size of the training set is increased.

Acknowledgments

We would like to thank Vricon [Vricon,] for providing the satellite data. This work has been supported by the Swedish Knowledge Foundation under the research profile on Semantic Robots, contract number 20140033.

References

[Belhumeur *et al.*, 1997] Peter N Belhumeur, João P Hespanha, and David J Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997.

[Fails and Olsen, 2003] Jerry Alan Fails and Dan R. Olsen, Jr. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pages 39–45, New York, NY, USA, 2003. ACM.

[Farabet *et al.*, 2012] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, volume 1, pages 575–582, 2012.

[Gould *et al.*, 2009] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1–8. IEEE, 2009.

[Längkvist *et al.*, 2016] Martin Längkvist, Andrey Kiselev, Marjan Alirezaie, and Amy Loutfi. Classification and segmentation of satellite orthoimagery using convolutional neural networks. *Remote Sensing*, 8(4):329, 2016.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Lee and Grauman, 2011] Y. J. Lee and K. Grauman. Learning the easy things first: Self-paced visual category discovery. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1721–1728, June 2011.

[Mohan, 2014] Rahul Mohan. Deep deconvolutional networks for scene parsing. *arXiv preprint arXiv:1411.4101*, 2014.

[Pinheiro and Collobert, 2014] P.O. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *31st International Conference on Machine Learning, ICML*, volume 1, pages 151–159, 2014.

[Raina *et al.*, 2007] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the Twenty-fourth International Conference on Machine Learning*, 2007.

[Strigl *et al.*, 2010] Daniel Strigl, Klaus Kofler, and Stefan Podlipnig. Performance and scalability of gpu-based convolutional neural networks. *2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, 0:317–324, 2010.

[Vricon,] Vricon, homepage. `http://www.vricon.com`. Accessed: 2016-01-03.

[Zhou *et al.*, 2015] Yisu Zhou, Xiaolin Hu, and Bo Zhang. *Advances in Neural Networks – ISNN 2015: 12th International Symposium on Neural Networks, ISNN 2015, Jeju, South Korea, October 15-18, 2015, Proceedings*, chapter Interlinked Convolutional Neural Networks for Face Parsing, pages 222–231. Springer International Publishing, Cham, 2015.