A Novel Machine Learning Based Approach for Retrieving Information from Receipt Images

RS *Roland Szabo*

Abstract In this paper we are approaching, from a machine learning perspective, the problem of performing optical character recognition on receipt images and then extracting structured information from the obtained text. Tools that have not been trained specifically for this kind of images do not handle them well usually, because receipts have custom fonts and, due to size constraints, many letters are close to each other. In this paper we adapt existing methods for doing OCR, in order to achieve better performance than off-the-shelf commercial OCR engines and to be able to extract the most accurate information from receipts. Document layout analysis is performed on the receipts, then lines are segmented into characters using Random Forests and finally they are classified using Linear Support Vector Machines. We provide an experimental evaluation of the proposed approach, as well as an analysis of the obtained results.

a computer to process, index and search the data much faster.

OCR is a difficult problem, even when done on straight papers, without creases, that are scanned, because first we must identify letters on the page and distinguish them from tables, figures and other objects that might be there, and because there are many kinds of fonts that have to be recognized. The problem of OCR on photographed documents is even more difficult. The illumination can vary, the document might be curved, there might be a skewed perspective and so on.

Information extraction is the process of taking raw, unstructured text and outputting information that is parsed and structured. This makes it easier to search and store the necessary data, because parts of the text that contain nothing useful can be discarded, while the rest is processed, brought to a standard form and is stored in a database.

The paper aims at investigating the use of Random Forests(Breiman 2001) and Support Vector Machines (SVM) (Cortes 1995) in developing an OCR engine that is tailored for receipts. By taking into consideration the constraints imposed by receipts, on both document layout and text font and spacing, we can improve performance compared to other OCR engines that are more general. The introduced approach is novel, since, to our knowledge, other papers for extracting information from receipts focus on improving either the image preprocessing step or on the parsing of text that was extracted using an off-the-shelf OCR engine.

The rest of the paper is structured as follows.

Section 2 describes the problem statement and the motivation behind developing such an OCR engine. Section 3 presents some of the related work in the field of character recognition using machine learning approaches. In Section 4 we present the approach used in developing our OCR engine. The experiments we have performed in order to experimentally evaluate our approach are described in Section 5 . Section 6 contains an analysis of our approach and a comparison to other OCR engines. Finally, we provide conclusions and pointers towards future work.

The first OCR engines used a set of handwritten rules to identify characters(Shepard 1971). These were hard to write and performed quite poorly on text written in new fonts. This kind of systems are fine when used on documents that are highly standardized and that always have the same font in the same place, such as passports, bank checks or credit cards.

More modern OCR engines use a machine learning algorithm to learn the rules by which to classify the characters(Smith 2007). They are more accurate and can easily learn to identify multiple fonts. While the rules are not handwritten, getting the labeled data for the machine learning algorithm is still a manual and tedious work.

While the general problem of object recognition is still a difficult one for computers, even though humans do it without a problem, recently there have been several breakthroughs in computer vision that give very good performance, in some cases even better than human, on simpler problems, such as character recognition.

2.2 OCR for Receipts. Motivation

Usually, an OCR engine has 2 main components: a document layout analysis part and a character recognition part.

In the case of receipts, the document layout is quite simple: all the text is on horizontal lines, there are no tables, and few figures, usually consisting of the logo of the shop. Identifying the lines can be done quite efficiently by looking at the color histogram of the receipt.

The character recognition is more complicated. While in a book the letters are almost always well separated and the lines have similar length, receipts are smaller, so they have less space available and everything is compressed as much as possible. This means that many letters end up touching, as show in figure 1, lines are of uneven length and have different justifications (left, right and center justifications alternate many times in a receipt). This means that before character recognition can be done, lines must be segmented into their composing letters.

In this paper we are researching the best ways to perform the character segmentation and then recognition, using only the data available from the image.

Fig. 1 Example of line where many characters are touching.

3 Literature Review

Optical character recognition has long been of interest in the computer industry. In 1976, Ray Kurzweil presented in (Schantz 1982) an omni-font OCR engine, that together with a speech synthesiser would read texts to blind people. It came together with its own special scanner, that was required for taking the images of pages.

R. Holley presents in a study(Holley 2009) done to evaluate OCR on australian newspapers that the accuracy of various commercial software on newspapers varies from 71% to 98%. A similar study done by E. Klijn on Dutch

(bilevel) images in this data set were initially size normalized to fit a 20x20 bounding box, while preserving aspect ratio. The resulting images were anti-aliased, which introduced grey levels. In the end, the images were centered in 28x28 image, translating the center of mass of the pixels to the center of the 28x28 image.

Initial results on the MNIST database, from the paper published by Y. LeCun and L. Bottou in 1998(Lecun 1998), are summarized in Table ???.

Table 1: Test error rates obtained on MNIST using various algorithms

Classifier	Test error rate
1-layer neural network	12.0 %
K-nearest-neighbors	5.0 %
PCA + quadratic classifier	3.3 %
SVM with Gaussian Kernel	1.4 %
2-layer neural network	4.7 %
3-layer neural network	2.45 %
Convolutional neural network	0.95 %

Many improvements have been published on the MNIST dataset. Some of the more recent ones include L. Deng and D. Yu(Deng 2011), obtaining an error rate of 0.83%, using a deep convex neural net with unsupervised pre-training. The best result obtained without using committees is D. Ciresan's deep neural network, trained on GPU devices(Cireşan 2010), that obtained 0.35% error rate. The current state of the art was obtained by the same group, using a committee of 35 convolutional neural networks, with an error rate of 0.23%, which beats human performance(Cireşan 2012).

For the character segmentation problem, one approach proposed by S. Lee and D. Lee(Lee) was based on vertical projections of the images. The projections were obtained by counting how many black pixels were in a column and then various criteria were used to determine which columns were candidates for segmentation, based on some threshold values. They obtained results ranging from 85% to 98% accuracy, depending on font and alphabet that was used.

F. Kahraman and B. Kurt used a nonlinear vector quantization to perform the character segmentation(Kahraman 2003) with an accuracy of 94.5% on a set of 2198 license plate characters.

F. Vojtěch and H. Václav presented in (Franc 2005) an approach using Hidden Markov Chains to model the problem of character segmentation in license plates. Without incorporating prior knowledge about the license plate, they got 37% incorrect segmentations, but by using that knowledge, they managed to get it down to 3.3%.

B. Janssen and E. Saund have worked on a system called Receipts2Go(Janssen 2012) for extracting information from small documents, including receipts. They presented two improvements to the image normalization process, but for the actual OCR part they used off-the-shelf commercial software.

4 Methodology

This section presents the background of the machine learning approaches we are using for the problem of OCR

Random forests have been introduced by Leo Breiman and Adele Cutler(Breiman 2001) as an ensemble of decision trees. When using only one decision tree to make a classification, one often runs into problems with high variance or high bias. Random forests present a mechanism to avoid these problems to make more accurate models, that generalize better.

When training the random forest, for each tree, n samples are taken with replacement from the training data (a bootstrap sample) and the tree is trained on these, using a slightly modified procedure. When splitting in a node, instead of choosing the best split across all features, as is done in the classical decision tree approach, here the best split is chosen from a random subset of the m features. This is done to avoid correlations between trees: otherwise good features would always get picked in all trees, so they would be correlated. As m is smaller, the correlation between the trees is smaller, but the strength of each tree (how well it can predict on its own) also goes down, so a balance must be found between these extremes.

Random forests are a popular algorithm in many machine learning competitions, because they are fast, they don't have many parameters to tune, yet still produce good predictions. Among their weaknesess is the fact that they can easily overfit a noisy dataset.

4.1.2 Support Vector Machines

Support vector machines(Cortes 1995) are discriminative classifiers formally defined by high-dimensional hyperplanes, which are used to distinguish between the classes to which data points belong. The hyperplane defined by an SVM maximizes the margin to the data points used in training, hoping that this leads to a better generalization of the classifier.

SVM can be used to project the original data to a much higher dimensional space (in some cases even infinte-dimensional space), so that they are linearly separable in that space. The mapping to a higher dimensional space is done using a kernel function.

One of the more popular kernels(Chang 2010) that can be used is the radial basis function (RBF) kernel, which is defined as:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||_2^2}{2\sigma^2}\right) \quad (1)$$

The value of the RBF value goes from zero (at infinity) to one (when $x = x'$), so it can be viewed as a similarity measure between the two samples. (Vert)

Because sometimes there is noise in the data, so it may not be possible to separate the data linearly, not even in a high dimensional space or, even if possible, this may not be desirable, because it would overfit to the data and not generalize well. In such cases, it is prefered to have a decision surfaces that makes some mistakes on the training data, but generalizes better and represents the noisy data more accurately. SVMs can be used as soft margin classifiers, allowing examples to be classified wrongly at training time, but penalizing them according to their distance to the other side of the hyperplane. (Russell 2009)

Because SVMs separate only two classes, when there are multiple classes to be distinguished, the "one-vs-all" approach can be used for classification, and is as accurate as any other approach for this problem(Rifkin 2004). In this case, one classifier is trained for each class, to distinguish it from all other classes. To make a prediction, all classifiers predict their value and the one that is used will be the one with the highest confidence score.

4.2 Model Design

4.2.1 The Random Forest Model

The random forest was used as a model for the character segmentation problem. The criterion for choosing the best feature to split a node is the information gain (entropy). Trees are grown to their full depth, no pruning or

4.2.2 The Support Vector Machine Model

The SVM was used for the character recognition problem. The performance of both linear and radial basis function kernels was evaluated.

The regularization parameter of the SVM was determined using cross-validation.

4.3 Document Layout Analysis

Before any characters can be recognized in a receipt, the image must first be preprocessed and normalized. This is done in several steps.

The preprocessing consists of binarizing the images, using Otsu's method(Otsu 1975), which adapts the threshold based on the histogram of the image. This step is done to remove any noise and speckles from the image.

The first step in normalization is to straighten the images. The receipts are assumed to be photographed with a mobile phone camera. Users will most often take photos that are slightly rotated. The orientation of the images is assumed to be vertical, so the software will not try to identify if the receipt is horizontal. To straighten the images, they are rotated from -10 to 10 degrees, with a 0.3 angle step, and a horizontal projection (summing the pixel values row-wise) is done for each resulting image. The straight image is assumed to be the one were are the most variations between the peaks and valleys of the histogram, because in the straight image there would be high peaks because of the lines and low valleys because of the space between lines.

The following step is removing the edges of the image, to keep only the receipt, removing any background. Due to variations in illumination, we cannot simply look for white patches to identify the receipt, because mobile cameras often use flash which gives receipts a blue tint while photos taken indoor close to a source of light have a yellow hue. The approach that was used was to look at the horizontal and vertical projections and to remove the section from the top and bottom that is over a threshold.

The last step is detecting the lines in the receipt. Because the images are already straight and without edges, all we have to do is identify the peaks in the horizontal histogram in the image.

The end result of the Document Layout Analysis is shown in figure 2, where we can see a receipt that is given as an input to the engine and the output of the Document Layout Analysis, the normalized image, with the detected lines highlighted.

S.C. ARTIMA S.A.
CLUJ NAPOCA, STR. BUCEGI, NR. 19
C.U.I. RO 11735628

RON

1,000 x 3,19
BATISTE NAZ.CLASIC3S 3,19 A
1,000 x 2,50
STICKLETTI CARTOF 80 2,50 A
0,390 x 7,99
MANDARINE 3,12 A
0,446 x 7,99
ROSII 3,56 A
1,000 x 11,20
SALAM CASA USCAT 290 11,20 A

SUBTOTAL 23,57

SUBTOTAL 23,57

TOTAL 23,57

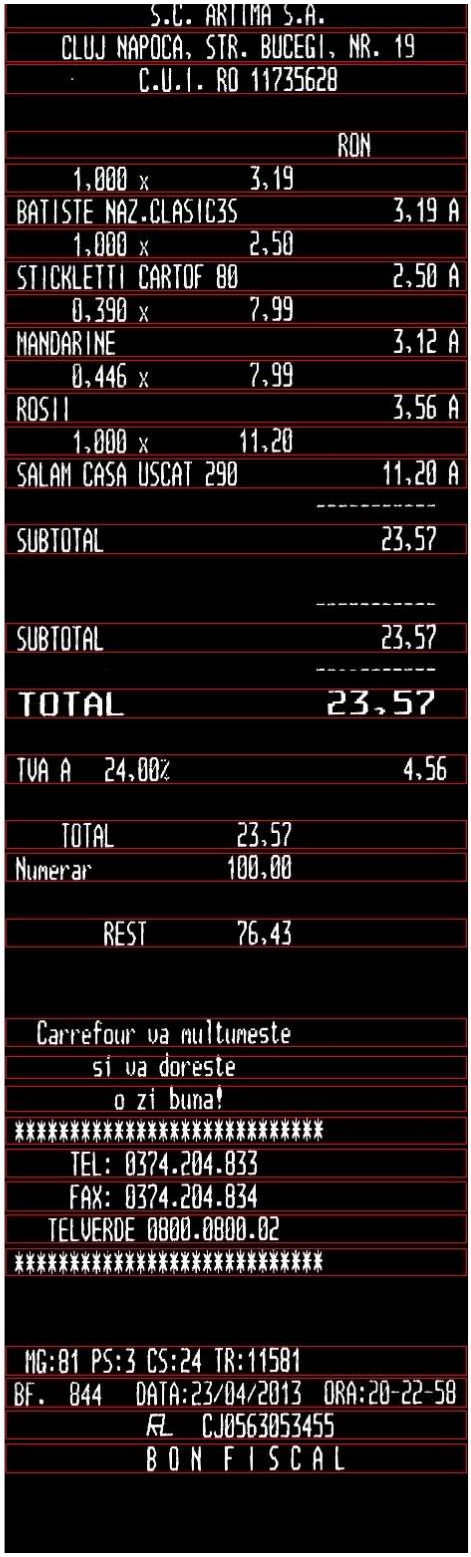
TVA A 24,00% 4,56

TOTAL 23,57
Numerar 100,00

REST 76,43

Carrefour va multumeste
si va doreste
o zi buna!
****
TEL: 0374.204.833
FAX: 0374.204.834
TELVERDE 0800.0800.02
****

MG:81 PS:3 CS:24 TR:11581
BF. 844 DATA:23/04/2013 ORA:20-22-58
RL CJ0563053455
B O N F I S C A L



S.C. ARTIMA S.A.
CLUJ NAPOCA, STR. BUCEGI, NR. 19
C.U.I. RO 11735628

RON
1,000 x 3,19
BATISTE NAZ.CLASIC3S 3,19 A
1,000 x 2,50
STICKLETTI CARTOF 80 2,50 A
0,390 x 7,99
MANDARINE 3,12 A
0,446 x 7,99
ROSII 3,56 A
1,000 x 11,20
SALAM CASA USCAT 290 11,20 A

SUBTOTAL 23,57

SUBTOTAL 23,57

TOTAL 23,57

TVA A 24,00% 4,56

TOTAL 23,57
Numerar 100,00

REST 76,43

Carrefour va multumeste
si va doreste
o zi buna!
****
TEL: 0374.204.833
FAX: 0374.204.834
TELVERDE 0800.0800.02
****

MG:81 PS:3 CS:24 TR:11581
BF. 844 DATA:23/04/2013 ORA:20-22-58
RL CJ0563053455
B O N F I S C A L

5 Experimental Evaluation

In this section we describe the experiments that we have done, starting with the data gathering process, the training of the models and the results of their evaluation.

5.1 Data Set and Processing

The data set was obtained from 20 receipts that were manually annotated with the position of each character in them. In total there are 7045 characters. There are 74 different characters, including digits, uppercase and lowercase letters and punctuation.

The bounding boxes of the characters were extracted from the images. The resulting patches were normalized to have a size of 30x30 pixels and were converted to grayscale. The small images that resulted after this processing were used as the data set for the character recognition problem.

For the character segmentation problem, positive and negative patches were extracted from the images, each containing 40 columns of pixels. The positive example were obtained by taking the leftmost and rightmost columns of the bounding boxes of characters, together with 19 previous columns and 20 columns that followed. The negative examples were obtained by sampling randomly from the middle of a character and taking 19 columns from before and 20 from after.

For the character recognition problem, the labels corresponding to each character were converted to a vector of 74 dimensions, with each dimension corresponding to one possible character value. The value of the dimension corresponding to the character of a data point was set to 1, while all the others were set to 0.

For the character segmentation problem, the labels were binary: 1 if a certain data point was were a segmentation should occur, 0 otherwise.

5.2 Training and Testing

The data set was shuffled and then split into two parts, one for training and one for testing. The splitting was done in a random way, because the data points are independent and order does not matter. The training set contained 80% of the data and the test set contained the remaining 20%.

All experiments were run multiple types, with the dataset being shuffled each time. In the case of the Random Forests, the multiple runs of the experiments are necessary because the splitting points for the trees and the dataset splits are chosen randomly across runs.

5.3 Results

For both tasks, the parameters for the algorithms were selected using cross-validation. In the case of the SVM, the search space was on logarithmic scale from 10^{-2} to 10^{4} for the regularization parameter. In the case of the random forest, the number of trees used ranged from 150 to 250, in steps of 50, and the number of features to be sampled at each point varied from using the square root, the base 2 logarithm, 10% or 30% of the total number of features.

Table ??? contains the average, maximum and minimum values obtained for the accuracy of the character recognition problem.

Table 2: The accuracy for the character recognition experiment

Kernel type	Regularization	Min	Max	Mean	Std. dev.
RBF	0.01	0.09141	0.09207	0.09165	0.00030
Linear	0.01	0.71585	0.71768	0.71672	0.00075

Linear	100	0.89695	0.90006	0.89860	0.00128
RBF	1000	0.90183	0.91341	0.90795	0.00475
Linear	1000	0.89207	0.89762	0.89453	0.00231
RBF	10000	0.90671	0.91220	0.90957	0.00225
Linear	10000	0.89146	0.89762	0.89494	0.00258

In this case, using a RBF kernel SVM resulted in a $91.018\% \pm 0.126$ accuracy in the best case, using a value of 100 for the regularization rate, while using a linear kernel yielded $90.490\% \pm 0.097$ as its best result, for the value of 1 for the regularization rate. A RBF kernel gives a slightly better results.

Figure 3 contains the confusion matrix for the best experiment on the character recognition problem.

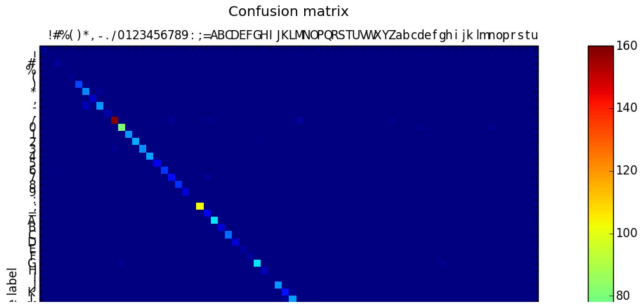


Fig. 3 Confusion matrix for the best model for the character recognition problem

Table 3: The F1 score for the character segmentation experiment

Number of trees	Number of features	Min	Max	Mean	Std. dev.
150	20	0.87318	0.88326	0.87735	0.00363
200	20	0.86975	0.88312	0.87657	0.00490
250	20	0.87258	0.88776	0.87773	0.00531
150	8	0.86894	0.88376	0.87569	0.00517
200	8	0.87227	0.88299	0.87675	0.00413
250	8	0.87178	0.88470	0.87717	0.00426
150	120	0.87262	0.88631	0.87816	0.00476
200	120	0.87122	0.88387	0.87724	0.00418
250	120	0.87367	0.88565	0.87885	0.00391
150	40	0.87073	0.88671	0.87822	0.00552
200	40	0.86980	0.88519	0.87845	0.00513
250	40	0.87358	0.88671	0.87936	0.00445

The confusion matrix for the best experiment on the character segmentation problem is presented in table ???.

Table 4: The confusion matrix for the character segmentation experiment

	No split	Split
Predicted no split	4556	363
Predicted split	255	2546

6 Discussion and Comparison to Related Work

This section analyzes our approaches and compares them to similar existing work.

6.1 Analysis of the Proposed Approach

As shown in section 5.3 , using an SVM with a RBF kernel seems to lead to slightly better results. One explanation for this would be that the data is already high dimensional (400 dimensions) and the data points lie in sufficiently different parts so that the separating hyperplanes work effectively, without needing a projection into the infinite dimensional space offered by the RBF kernel.

In the case of the character segmentation, the values of the parameter didn't matter that much. The difference between the best and the worst result was about 0.2. The number of features had the most influence, of 0.1%, on the F1 score, while the number of trees influenced it only by 0.01. The highest average was obtained for the values of 250 trees in the forest and 40 features chosen at each split, giving an F1 score of $87.936\% \pm 0.445$

somewhat sensitive to the number of features used in them.

If we look at the confusion matrix for the character recognition problem, computed for the whole data set, we can see that one of the most often made mistakes are:

- confusing , with . - 28 times
- confusing . with , - 12 times
- confusing O with 0 - 24 times
- confusing 0 with O - 15 times
- confusing 0 with o - 3 times
- confusing l with I - 6 times
- confusing I with T - 5 times
- confusing I with 1 - 4 times
- confusing 1 with I - 4 times

These mistakes are easy to make: a comma and a period are very similar, especially in a noisy, low resolution environment. The lower and upper case letters O are again very similar to the digit 0, on some receipts there being no distinction between O and 0, only the context in which they are used. For monospace fonts, which some receipts use, the lack of difference between l, I, 1 is a well known problem, which is why many design books and human computer interaction studies recommend using other fonts(Chaparro 2006).

If we look at the confusion matrix for the character segmentation problem, we see that it makes the mistake of not predicting a split more often than it does the opposite, so the model is more specific, rather then sensitive.

Specifity and sensitivity can be computed as(Fawcett 2006)

$$\text{sensitivity} = \frac{\text{true positives}}{\text{all positives}} = \frac{2546}{2546 + 363} = 87.52\% \tag{2}$$

$$\text{specificity} = \frac{\text{true negatives}}{\text{all negatives}} = \frac{4556}{4556 + 255} = 94.70\% \tag{3}$$

Another good evaluation measure is the *Area unde the ROC curve* (AUC)(Fawcett 2006). The *ROC* (Receiver Operating Characteristic) curve is a plot of *sensitivity* versus (1 - *specificity*). Usually they are constructed from classifiers that instead of a label, return a score that can be converted into a label by thresholding. In the case of classifiers that return the class directly, the ROC curve has only one point, from where the AUC measure can be computed.

In our case, the AUC is 0.911, as computed from the graph in figure 4.

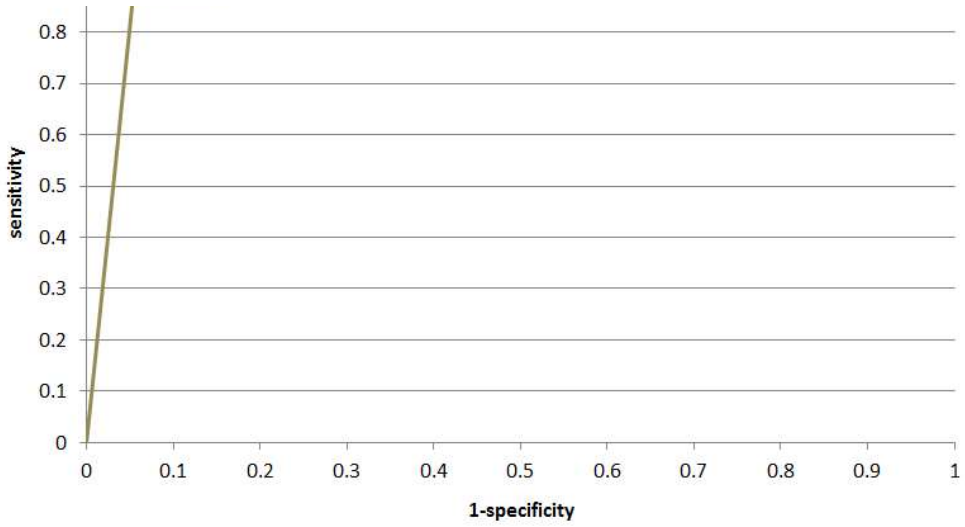


Fig. 4 ROC curve for the character segmentation problem

(corresponding to the ten digits), so it is much easier for a classifier to assign the correct class to a data point, compared to our dataset, which has 74 distinct classes, of which many are similar (the digit 0 and the letters o and O, the digit 1 and the letter l). Another factor that makes MNIST easier to solve is the fact that it has 70000 datapoints in total, while our dataset contains only 7045 data points. This almost ten fold difference has a great contribution to the ability of the classifier to generalize. Even on the MNIST dataset, adding more data is beneficial, as show by Ciresan et al.(Cireşan 2010), who obtained one of the best results on MNIST by augmenting the dataset with artificial data generated by transforming the existing digits.

The results obtained in (Kahraman 2003) and (Franc 2005) for the character segmentation problem are a bit better than our results: 94.5% and 96.7% accuracy, compared to our 91.9%. The difference between the two problems is that in their papers the number of characters in a licence plate is known and fixed, while lines in a receipt have varying lenghts. This prior knowledge used by them can explain the difference in results.

The paper by Janssen(Janssen 2012) deals with OCR on receipts, but they don't train the OCR engine for receipts, they just present a new approach for image normalization. In contrast, our approach improves the accuracy of the OCR engine by focusing on the character segmentation and recognition problem.

6.3 Application for the OCR Engine

The main reason for developing this OCR engine was to use it an application, called ReceiptBudget, which would enable users to take a photo of a receipt, the program would then extract the relevant information (date, items, store, total) from the photo, store it in a database, and then user could view an interactive dashboard of his expenses.

The motivation for using this method of data entry is that it is much faster than the alternative, which is much slower, tedious and users are prone to forgetting to do it.

One view of the dashboard would allow the user to view a heat map of his expenses, which can be seen in figure 5, either as a total or on a timeline. Another view would allow the user to filter expenses based on shop, month, day of week and date range, in order to notice patterns in his spending. This view can be seen in figure 6

The application was presented at the Imprezzio Software Contest in November 2013, where it won the first prize(Imprezzio Software Co...).

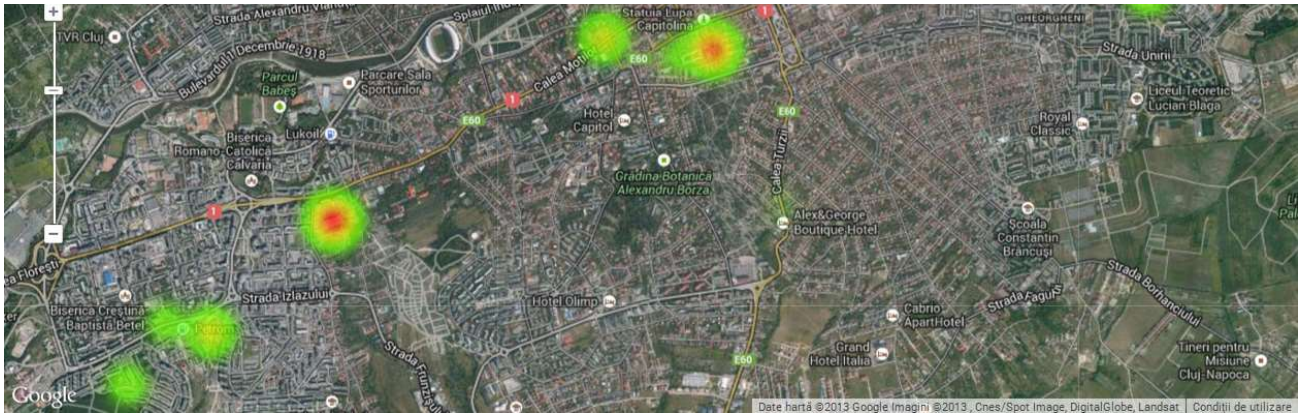


Fig. 5 The heatmap view of the dashboard in ReceiptBudget

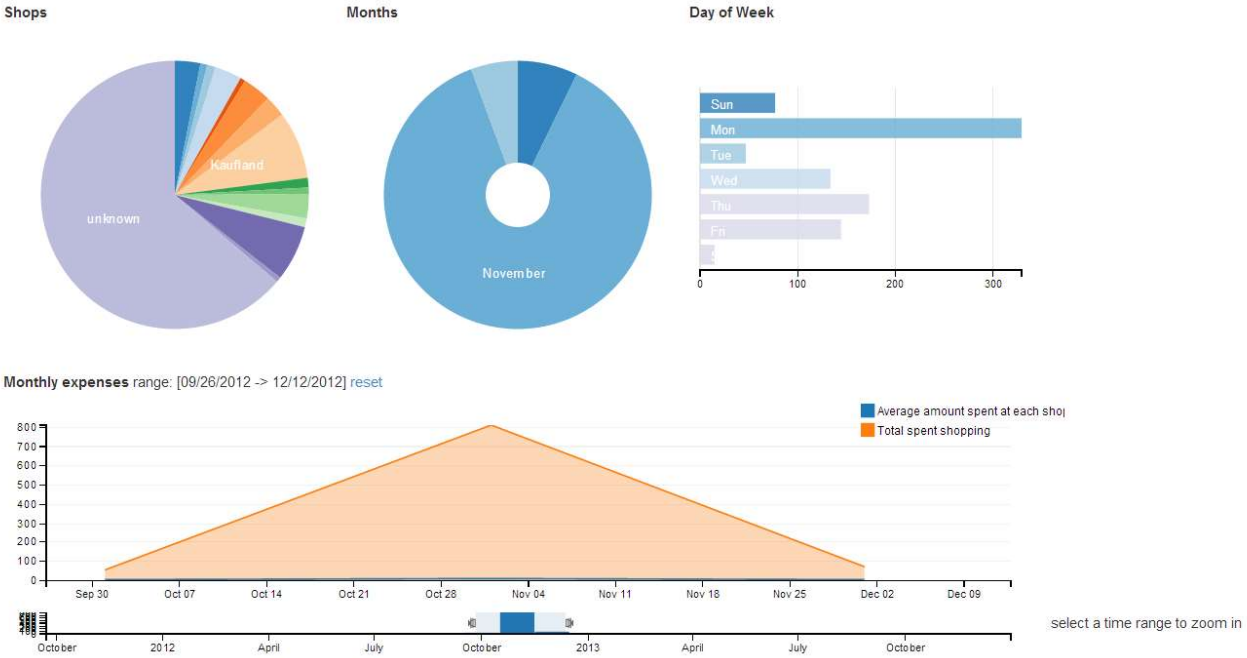


Fig. 6 The graphs and charts in the dashboard

advantages of our proposal with respect to similar existing approaches.

Future work will be made to improve the accuracy of the propsed models. One way to do this would be to gather more data. Almost all machine learning problems benefit from the addition of extra data(Halevy 2009).

Another improvement would be to try some other classifiers, such as artificial neural networks, which have been used to obtain the state of the art results on the MNIST digit classification dataset, so it seems reasonable to assume that they would perform well on this problem too.

For the character segmentation problem, it might help if the problem was presented differently. Now, for each column the classifier has to make a decision if it is a space between letters or not. Other approaches would be to instead predict the length of the current letter or to move to make a more global decision, to determine the best way to segment a line in a way that maximizes an energy function.

References

1. Herbert F Schantz. History of OCR, Optical Character Recognition. Recognition Technologies Users Association, 1982.
2. Leo Breiman. Random forests. *Machine learning* **45**, 5–32 Springer, 2001.
3. Corinna Cortes, Vladimir Vapnik. Support-vector networks. *Mach Learn* **20**, 273–297 Springer Science + Business Media, 1995. Link (http://dx.doi.org/10.1007/bf00994018)
4. David H Shepard, Howard W Silsby III. READING APPARATUS. (1971).
5. Ray Smith. An Overview of the Tesseract OCR Engine.. **7**, 629–633 In *ICDAR*. (2007).
6. Rose Holley. How Good Can It Get?. *D-Lib Magazine* **15** CNRI Acct, 2009. Link (http://dx.doi.org/10.1045/march2009-holley)
7. Edwin Klijn. The current state-of-art in newspaper digitization: A market perspective. *D-Lib Magazine* **14**, 5 Corporation for National Research Initiatives, 2008.
8. Yann LeCun, Corinna Cortes. MNIST handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist* (1998).
9. Y. Lecun, L. Bottou, Y. Bengio, P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**, 2278–2324 Institute of Electrical & Electronics Engineers (IEEE), 1998. Link (http://dx.doi.org/10.1109/5.726791)
10. Li Deng, Dong Yu. Deep convex net: A scalable architecture for speech pattern classification. In *Proceedings of the Interspeech*. (2011).
11. Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, Jürgen Schmidhuber. Deep, Big, Simple Neural Nets for Handwritten Digit Recognition. *Neural Computation* **22**, 3207–3220 MIT Press - Journals, 2010. Link (http://dx.doi.org/10.1162/neco_a_00052)
12. D. Cireşan, U. Meier, J. Schmidhuber. Multi-column Deep Neural Networks for Image Classification. *ArXiv e-prints* (2012).
13. Dong-June Lee, Seong-Whan Lee. A new methodology for gray-scale character segmentation and recognition. *Proceedings of 3rd International Conference on Document Analysis and Recognition* IEEE Comput. Soc. Press Link (http://dx.doi.org/10.1109/icdar.1995.599049)
14. Fatih Kahraman, Binnur Kurt, Muhittin Gökmen. License plate character segmentation based on the gabor transform and vector quantization. 381–388 In *Computer and Information Sciences-ISCIS 2003*. Springer, 2003.
15. Vojtěch Franc, Václav Hlaváč. License Plate Character Segmentation Using Hidden Markov Chains. *Lecture Notes in Computer Science* 385–392 Springer Science + Business Media, 2005. Link (http://dx.doi.org/10.1007/11550518_48)
16. [illegible]

(http://dl.acm.org/citation.cfm?id=1756006.1859899)

18. Jean-Philippe Vert. Kernel Methods in Genomics and Computational Biology. *Kernel Methods in Bioengineering, Signal and Image Processing* 42–63 IGI Global Link (http://dx.doi.org/10.4018/978-1-59904-042-4.ch002)
19. Stuart Russell. Artificial Intelligence: A Modern Approach. Prentice Hall, 2009.
20. Ryan Rifkin, Aldebaro Klautau. In defense of one-vs-all classification. *The Journal of Machine Learning Research* **5**, 101–141 JMLR. org, 2004.
21. Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica* **11**, 23–27 (1975).
22. Tom Fawcett. An introduction to ROC analysis. *Pattern recognition letters* **27**, 861–874 Elsevier, 2006.
23. B Chaparro, A Dawn Shaikh, Alex Chaparro. Examining the Legibility of Two New ClearType Fonts. *Usability News* **8** (2006).
24. Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters* **27**, 861–874 Elsevier BV, 2006. Link (http://dx.doi.org/10.1016/j.patrec.2005.10.010)
25. Imprezzio Software Contest Wraps Up in Romania; Gathers Steam in US.
26. Alon Halevy, Peter Norvig, Fernando Pereira. The unreasonable effectiveness of data. *Intelligent Systems, IEEE* **24**, 8–12 IEEE, 2009.