Applying OCR Technology for Receipt Recognition

By **Ozhiganov Ivan** on April 7, 2016

The problem of optical character recognition (OCR) in various conditions remains as relevant today as it was in past years. Automated recognition of documents, credit cards, recognizing and translating signs on billboards – all of this could save time for collecting and processing data. With the development of convolutional neural networks (CNN) and methods of machine learning, the quality of text recognition is continually growing.

Having implemented a project for receipt recognition, we once again saw how effective convolutional neural networks are. For our research, we chose different receipts from a range of Russian stores with the text using both Cyrillic and Latin letters. The developed system can be easily adapted for recognizing receipts from other countries and text in other languages. Let's consider the project in detail in order to demonstrate the operating principle of the solution.

> ***The goal of our project is to develop an app using the client-server architecture for receipt recognition and extracting meaning from receipts.***

Project Overview

This task is broken into several stages:

1. Preprocessing
   - Finding a receipt on the image
   - Binarization
2. Finding the text
3. Recognition
4. Extracting meaning from receipts

Implementation

1. Preprocessing

The preprocessing stage consists of the following preliminary work with the image: finding a receipt in the image, rotating the image so that the receipt strings are located horizontally, and then making a binarization of the receipt.

1.1. Finding and turning a receipt in the image

In order to solve the problem of finding a receipt in the image, we used the following methods:
Adaptive binarization with a high threshold
Convolutional Neural Network
Haar cascade classifier

Finding a receipt via adaptive binarization with a high threshold

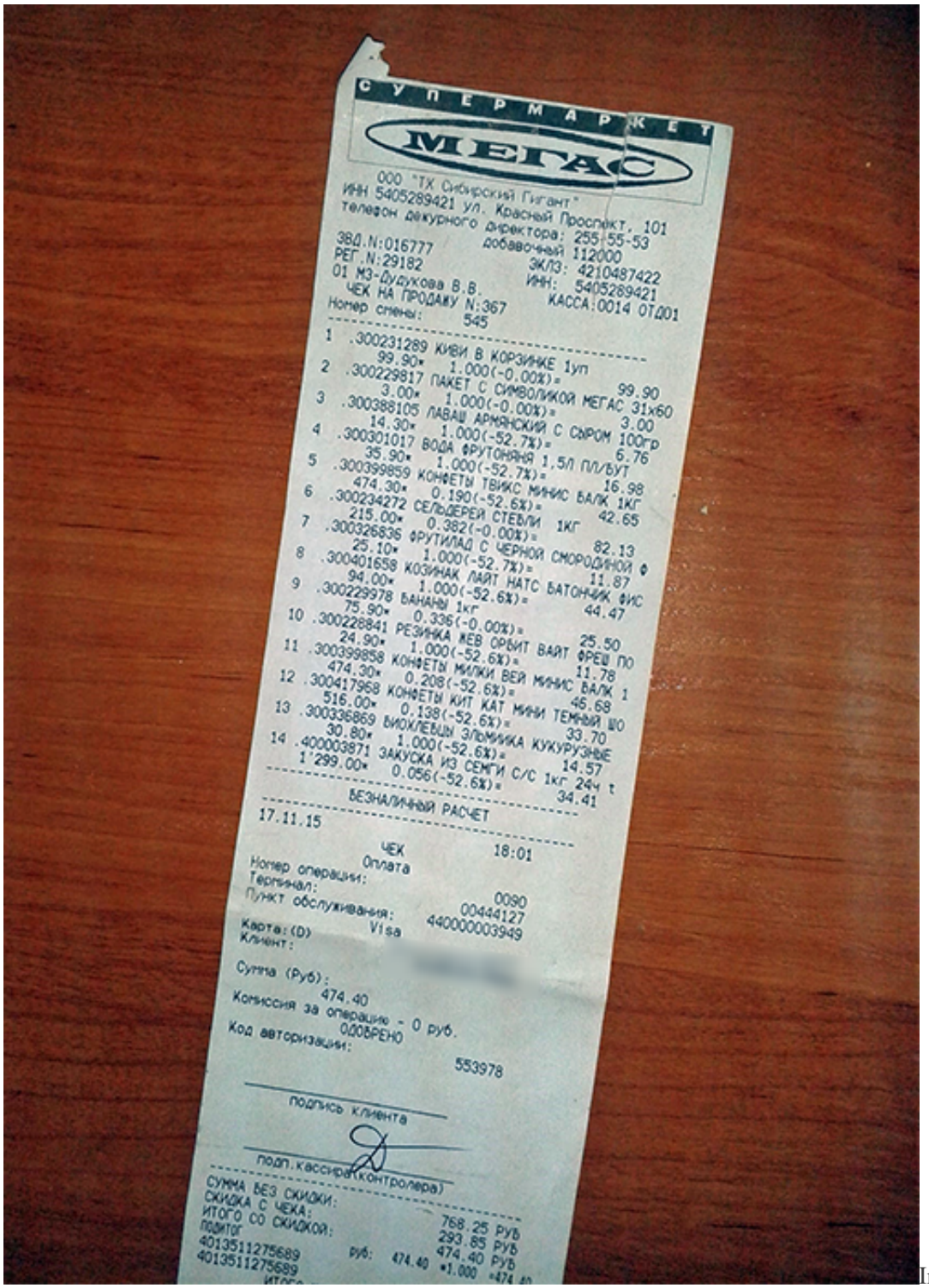


Image 1: Original receipt view

The problem here was reduced to finding an area in the image that contains a complete receipt and minimum background.
In order to make the finding process simpler, we first rotate the image so that each individual row is as close to horizontal as possible (Image 2). The turning algorithm is required to

maximize the variance of the brightness sum over the strings. The maximum is reached when the strings are located horizontally.

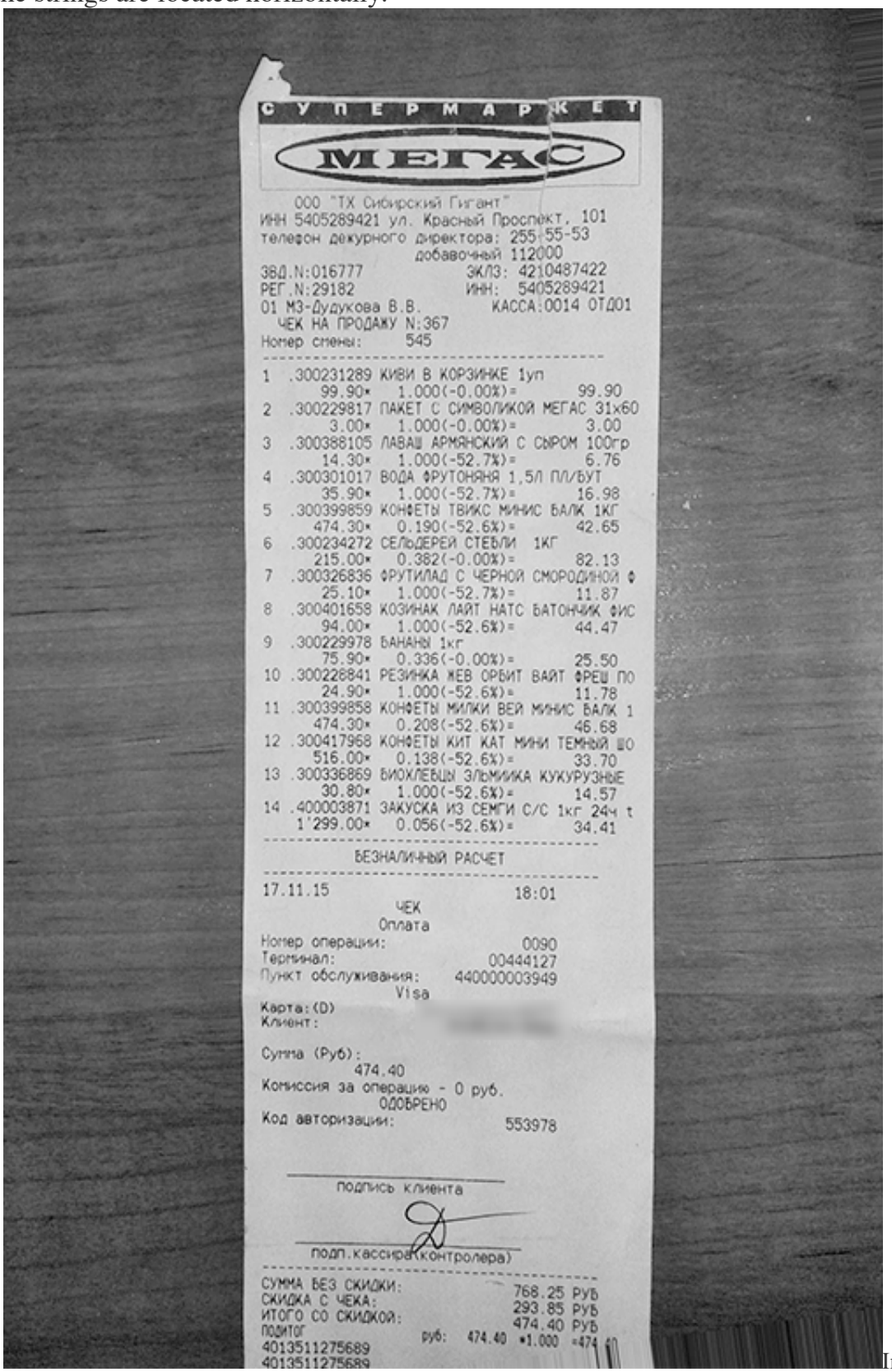
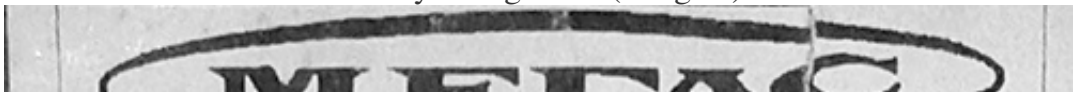


Image 2: Turn of the receipt

We used the *adaptive_threshold* function from the library *scikit-image* to find the receipt. This function does adaptive binarization with a high threshold and leaves white pixels in areas with a high gradient, whereas areas that are more homogeneous become black. Using this function, only a few white pixels remain with a homogeneous background and we look for them in the described rectangle. As a result, the derived rectangle includes the receipt area and a minimum of unnecessary background (Image 3).



ООО "ТХ Сибирский Гигант"
ИНН 5405289421 ул. Красный Проспект, 101
телефон дежурного директора: 255-55-53
добавочный 112000
ЗВД.N:016777 ЭКЛЗ: 4210487422
РЕГ.N:29182 ИНН: 5405289421
01 МЗ-Дудукова В.В. КАССА:0014 ОТД01
ЧЕК НА ПРОДАЖУ N:367
Номер смены: 545

---

1 .300231289 КИВИ В КОРЗИНКЕ 1уп
99.90* 1.000(-0.00%)= 99.90
2 .300229817 ПАКЕТ С СИМВОЛИКОЙ МЕГАС 31х60
3.00* 1.000(-0.00%)= 3.00
3 .300388105 ЛАВАШ АРМЯНСКИЙ С СЫРОМ 100гр
14.30* 1.000(-52.7%)= 6.76
4 .300301017 ВОДА ФРУТОНЯНЯ 1,5Л ПЛ/БУТ
35.90* 1.000(-52.7%)= 16.98
5 .300399859 КОНФЕТЫ ТВИКС МИНИС БАЛК 1КГ
474.30* 0.190(-52.6%)= 42.65
6 .300234272 СЕЛЬДЕРЕЙ СТЕБЛИ 1КГ
215.00* 0.382(-0.00%)= 82.13
7 .300326836 ФРУТИЛАД С ЧЕРНОЙ СМОРОДИНОЙ Ф
25.10* 1.000(-52.7%)= 11.87
8 .300401658 КОЗИНАК ЛАЙТ НАТС БАТОНЧИК ФИС
94.00* 1.000(-52.6%)= 44.47
9 .300229978 БАНАНЫ 1кг
75.90* 0.336(-0.00%)= 25.50
10 .300228841 РЕЗИНКА ЖЕВ ОРБИТ ВАЙТ ФРЕШ ПО
24.90* 1.000(-52.6%)= 11.78
11 .300399858 КОНФЕТЫ МИЛКИ ВЕЙ МИНИС БАЛК 1
474.30* 0.208(-52.6%)= 46.68
12 .300417968 КОНФЕТЫ КИТ КАТ МИНИ ТЕМНЫЙ ШО
516.00* 0.138(-52.6%)= 33.70
13 .300336869 БИОХЛЕБЦЫ ЭЛЬМИИКА КУКУРУЗНЫЕ
30.80* 1.000(-52.6%)= 14.57
14 .400003871 ЗАКУСКА ИЗ СЕМГИ С/С 1кг 24ч t
1'299.00* 0.056(-52.6%)= 34.41

БЕЗНАЛИЧНЫЙ РАСЧЕТ

17.11.15 18:01
ЧЕК
Оплата
Номер операции: 0090
Терминал: 00444127

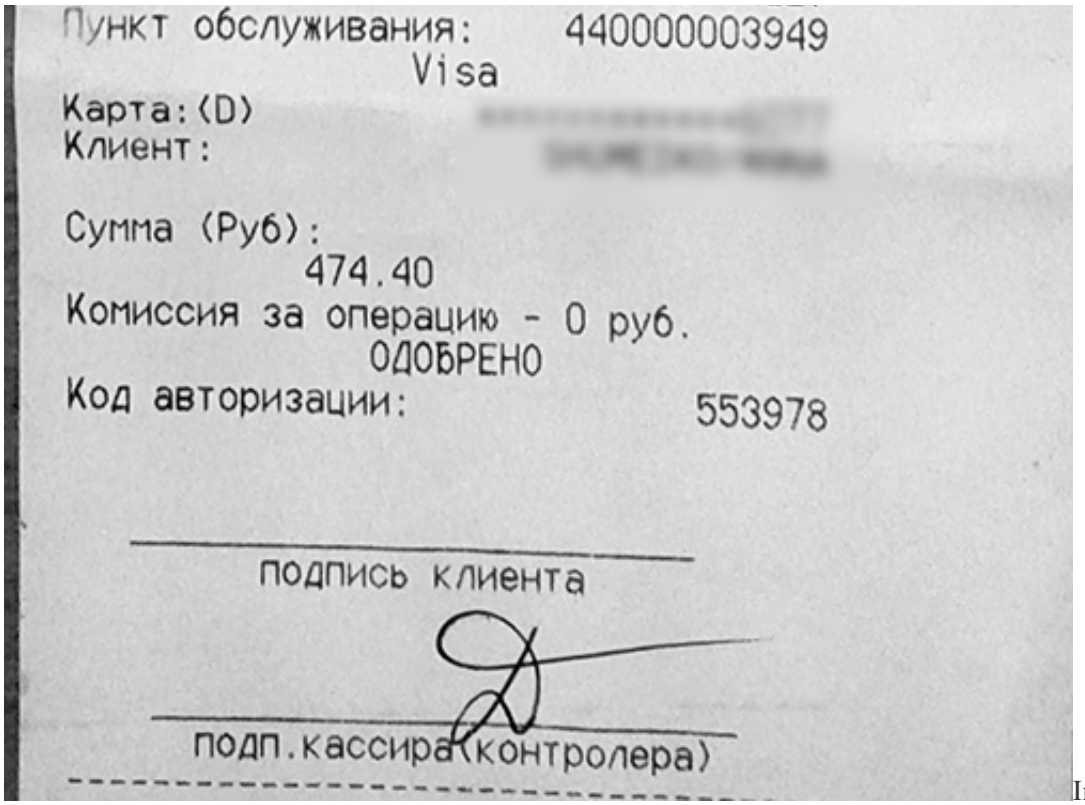


Image 3: Identified area with the receipt

Finding a receipt using a convolutional neural network

We decided to find keypoints of the receipt using a convolutional neural network as we did it before for the object detection project. We chose the receipt angles as keypoints. This method performed well, although it was not as efficient as adaptive binarization with a high threshold. The convolutional neural network showed a non-ideal result because it was trained for predicting the angle coordinates only relative to the found text. In addition to this the angles of the receipts meant that the text location varied from one receipt to another, and that's why the precision of the CNN model is not very high.

Here are the results that the convolutional neural network demonstrated:

Image 4: Examples of finding the receipt angles using CNN

Finding a receipt using a Haar cascade classifier

As an alternative to the described models, we decided to try a Haar cascade classifier. After a week of training the Haar cascade classifier and changing parameters of receipt recognition, we didn't get a satisfactory result. Even the CNN performed with higher quality.

Here are the examples of Haar cascade classifier work:


Image 5: Positive results of the Haar cascade classifier work

Image 6: False-negative and false errors of the Haar cascade classifier

1.2. Binarization

In the end we used the same *adaptive_threshold* method for binarization. The window is quite big so that it contains the text as well as the background (Image 7).

Image 7: Receipt binarization

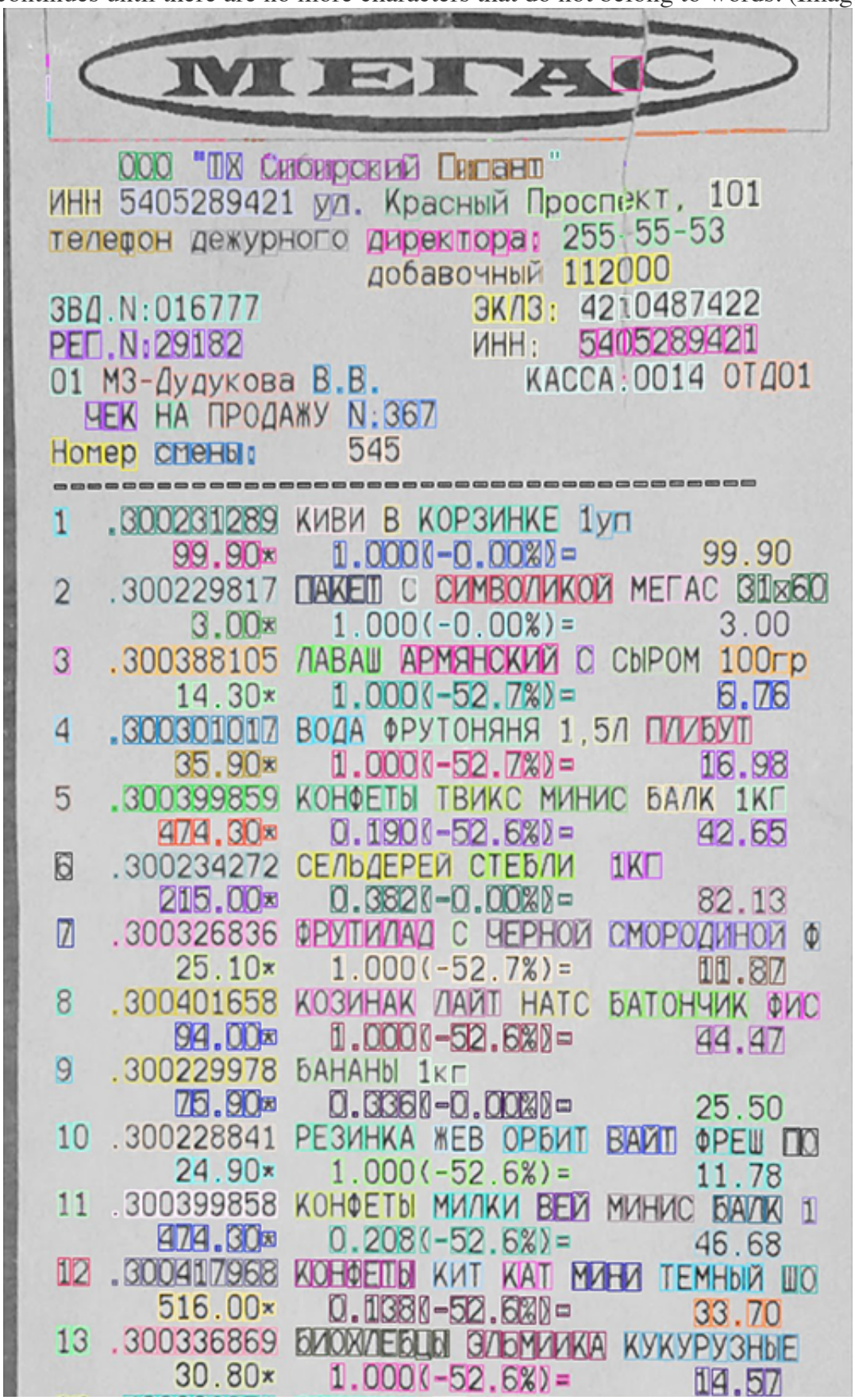
2. Finding the text

2.1. Finding the text using the method of connected components

The first stage of finding the text consists of finding the connected components. We did this

using the *findContours* function from OpenCV. The majority of connected components are real characters, but some of them are just noise fragments that are left after binarization. We eliminated them using filters across the maximal/minimal axis.

Then we applied the algorithm of combining connected components to the compound characters such as ":", "Й", "=". After this the characters are combined to form words by searching their closest neighbors. The principle of a closest neighbors search is the following: it's necessary to find the closest neighbors for every character, and then you can choose the most appropriate candidate for combination from the right and the left side. The algorithm continues until there are no more characters that do not belong to words. (Image 8).



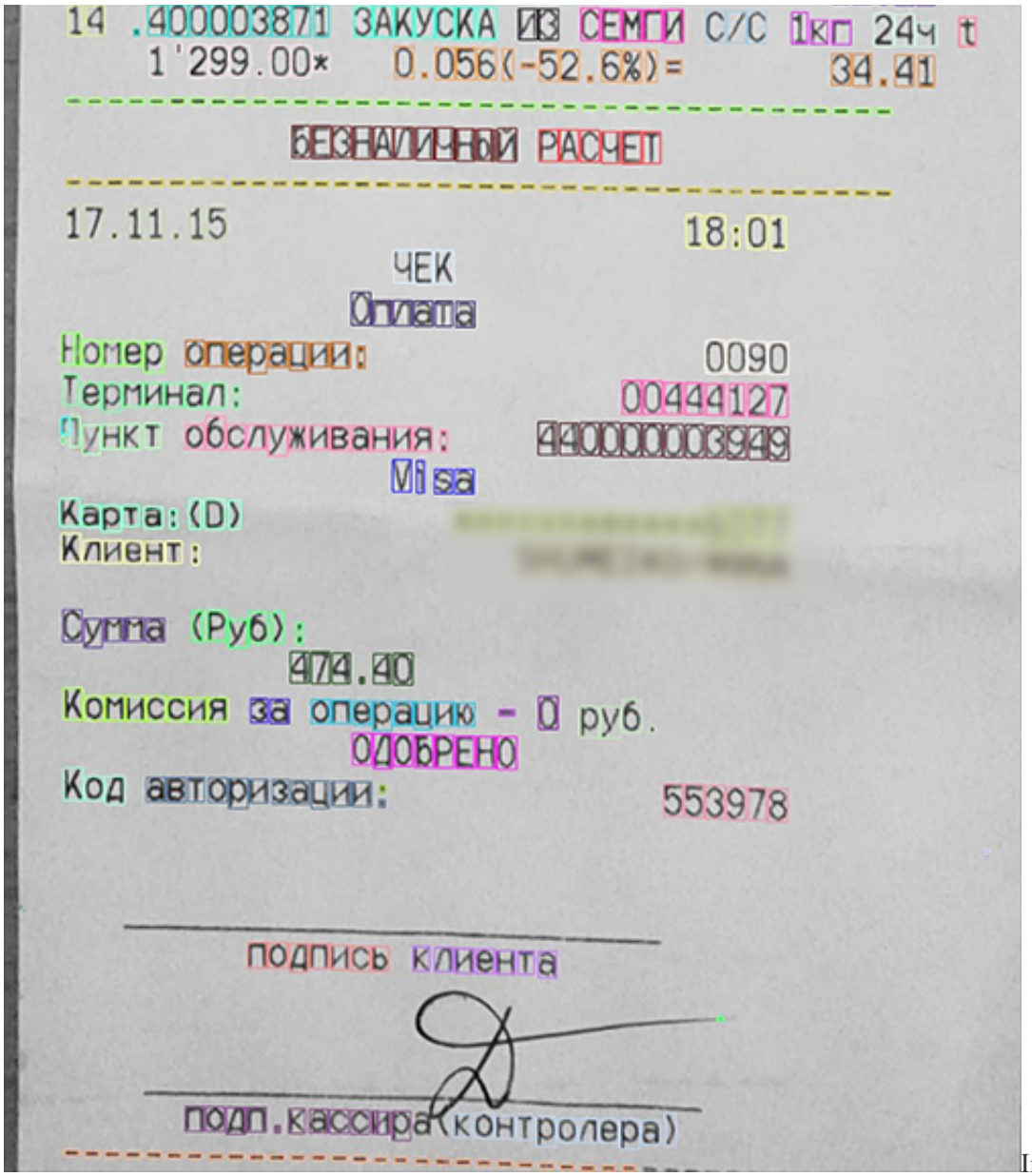
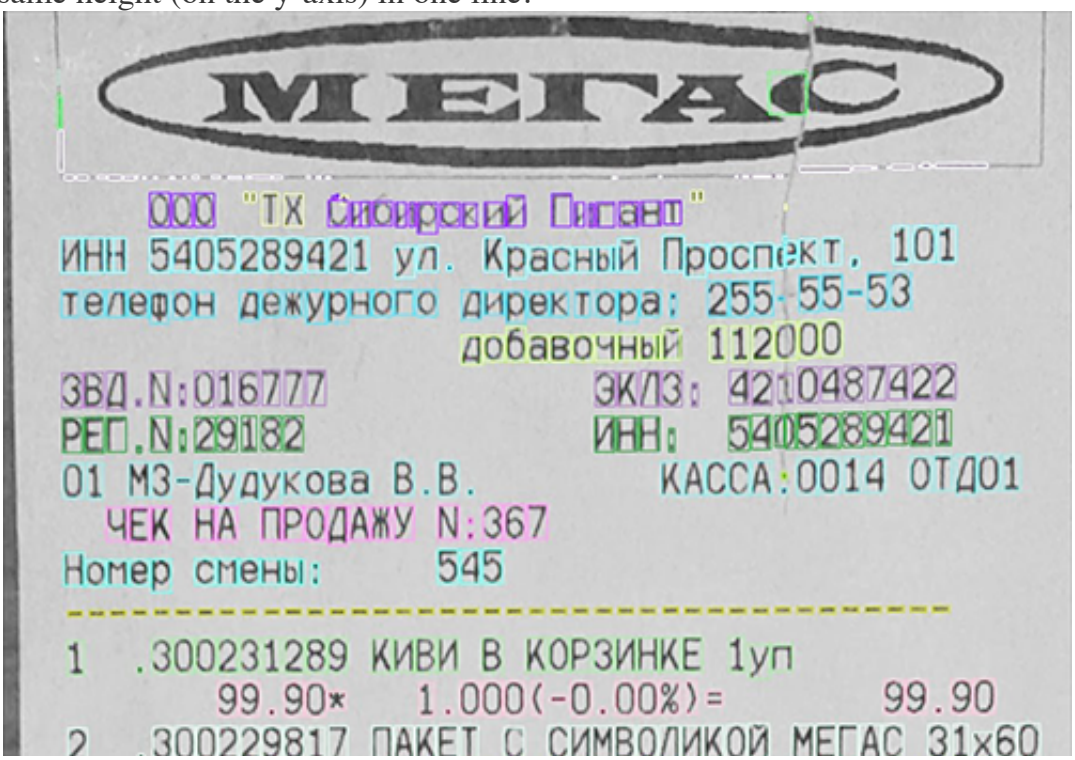


Image 8: Finding connected components and forming words (words are highlighted in one color)

Then lines are formed from the words. Here we use the theory that words are located at the same height (on the y-axis) in one line.



3.00* 1.000(-0.00%)= 3.00
3 .300388105 ЛАВАШ АРМЯНСКИЙ С СЫРОМ 100гр
14.30* 1.000(-52.7%)= 6.76
4 .300301017 ВОДА ФРУТОНЯНЯ 1,5Л ПЛ/БУТ
35.90* 1.000(-52.7%)= 16.98
5 .300399859 КОНФЕТЫ ТВИКС МИНИС БАЛК 1КГ
474.30* 0.190(-52.6%)= 42.65
6 .300234272 СЕЛЬДЕРЕЙ СТЕБЛИ 1КГ
215.00* 0.382(-0.00%)= 82.13
7 .300326836 ФРУТИЛАД С ЧЕРНОЙ СМОРОДИНОЙ Ф
25.10* 1.000(-52.7%)= 11.87
8 .300401658 КОЗИНАК ЛАЙТ НАТС БАТОНЧИК ФИС
94.00* 1.000(-52.6%)= 44.47
9 .300229978 БАНАНЫ 1кг
75.90* 0.336(-0.00%)= 25.50
10 .300228841 РЕЗИНКА ЖЕВ ОРБИТ ВАЙТ ФРЕШ ПО
24.90* 1.000(-52.6%)= 11.78
11 .300399858 КОНФЕТЫ МИЛКИ ВЕЙ МИНИС БАЛК 1
474.30* 0.208(-52.6%)= 46.68
12 .300417968 КОНФЕТЫ КИТ КАТ МИНИ ТЕМНЫЙ ШО
516.00* 0.138(-52.6%)= 33.70
13 .300336869 БИОХЛЕБЦЫ ЭЛЬМИИКА КУКУРУЗНЫЕ
30.80* 1.000(-52.6%)= 14.57
14 .400003871 ЗАКУСКА ИЗ СЕМГИ С/С 1кг 24ч t
1'299.00* 0.056(-52.6%)= 34.41

БЕЗНАЛИЧНЫЙ РАСЧЕТ
==
17.11.15 18:01
ЧЕК
Оплата
Номер операции: 0090
Терминал: 00444127
Пункт обслуживания: 440000003949
Visa
Карта:(D) [illegible]
Клиент: [illegible]

Сумма (Руб):
474.40
Комиссия за операцию - 0 руб.
ОДОБРЕНО
Код авторизации: 553978

подпись клиента

Image 9: Forming the lines (lines are highlighted in one color)

The disadvantage is that this algorithm cannot properly recognize words with connected or broken letters.

2.2. Finding the text with a grid

We found that almost all the receipts have monospaced text. This means that we can draw a grid on the receipt in a way that all the grid lines go between the characters:

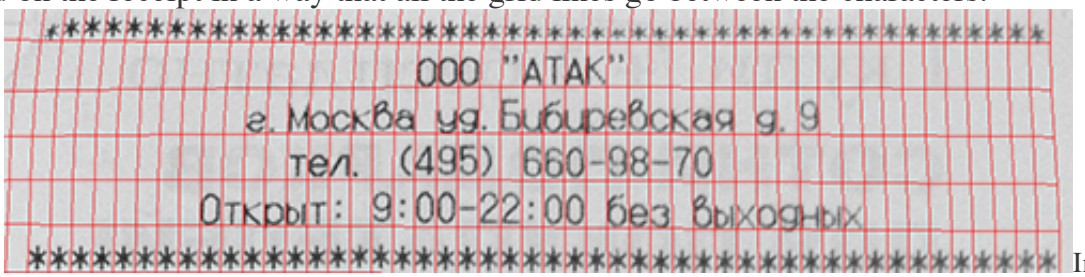


Image 10: Example of the grid

An automatic search algorithm through a receipt grid simplifies further receipt recognition. A neural network can be applied to every cell of the grid and every character can be recognized. There are no complications with connected and broken characters. Finally, the number of spaces that goes side by side in the string is defined precisely.

We tried the following algorithm to find the described grid. First, you need to find connected components on the binary image:

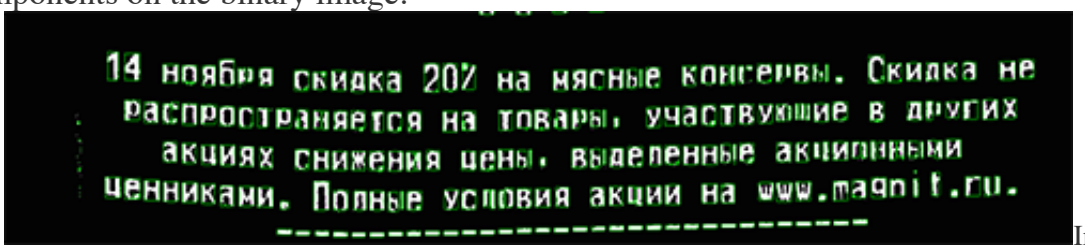


Image 11: Example of finding the connected components

Then you should take the lower-left angles of the green rectangles and get the set of points that are given by coordinates. In order to determine distortions, we developed the 2d periodic function:

$$f(x,y) = \frac{0.2}{1 - 0.4\left(\cos\left(\frac{x - x_0}{T_x}\right) + \cos\left(\frac{y - y_0}{T_y}\right)\right)}$$

The graph of the formula looks like this:

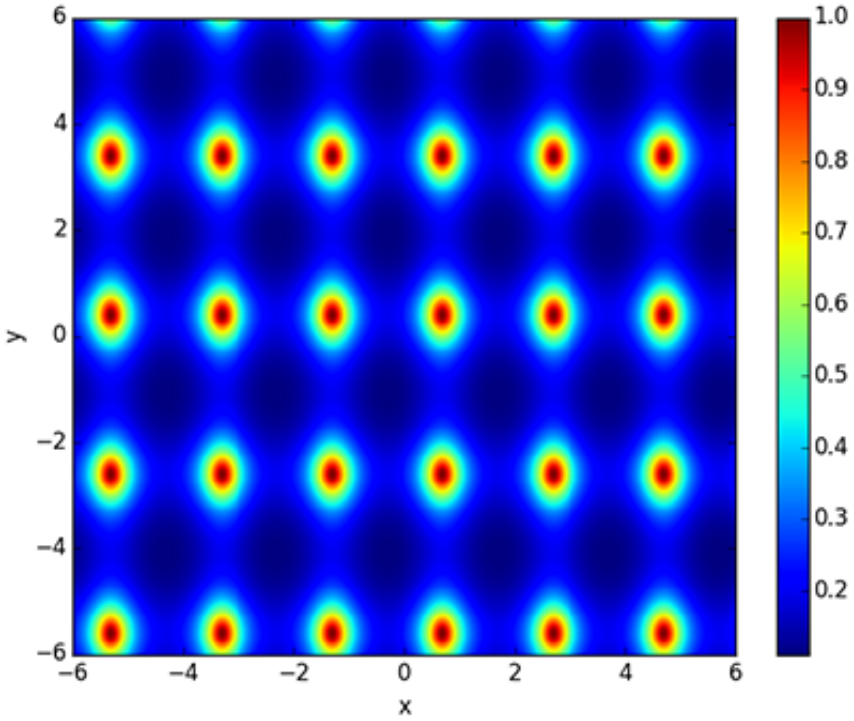


Image 12: Graph of the function in formula

The main idea behind the receipt grid is the finding of such nonlinear geometric distortions of the coordinates where the points are at the graph peaks. In other words, the problem is reformulated as a maximization problem of the function values sum. Taking this into account, it is necessary to find an optimal distortion. Geometric distortion was parametrized via the RectBivariateSpline function from the Scipy module in Python. Optimization was implemented using the minimize function from the Spicy module.

Here is an example of the correctly found grid:

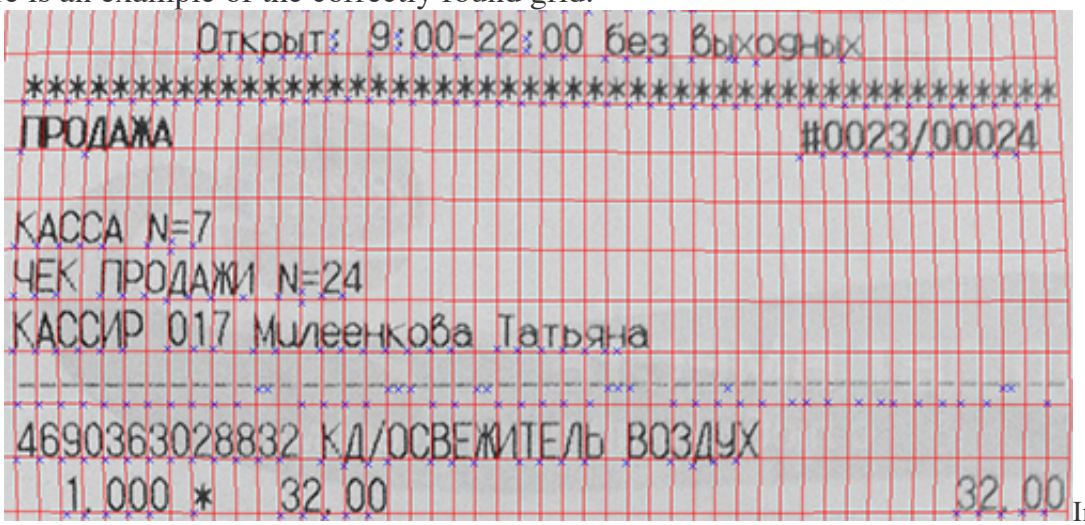


Image 13: Example of a correctly found grid

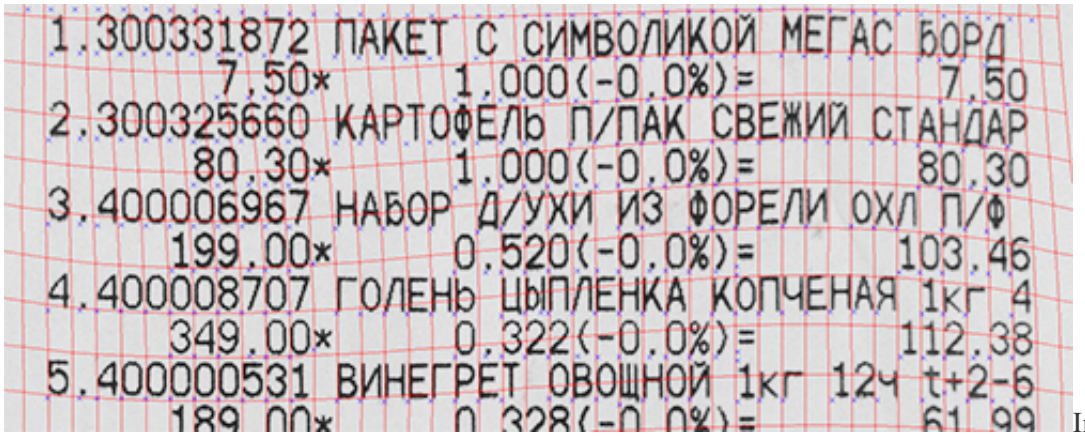


Image 14: Example of an incorrectly found grid

Finally, we decided to avoid using this method because it has a range of significant drawbacks, is unstable, and is slow.

3. Optical Character Recognition (OCR)

3.1. Recognizing text found by the method of connected components

OCR is implemented by a convolutional neural network that was trained to find fonts taken from receipts. At the output, we have probabilities for every character and take several initial options that give us a probability close to 1 (99%) of the total sum. Then we consider all the possible options of compiling the words from the received characters and check them using a dictionary. It helps to improve the accuracy of recognition and to eliminate mistakes from similar characters (for example, "З" and "Э", Cyrillic alphabet).

```
-
-

---------------- --- - - ---------.-
.'ТХ * .. -
ООО .-UOP1РСКИЙ ГР1ГАНТ
ИНЗ 5405289421 УЛ. КРАСНЬ:Й ПРОСП(. -)КТ. 101
ТЕЛЕФОН ДЕУРНОГО ДИРЕКТОРА= 255-55-53
ДОБАВОЧНЫЙ 112000
ЗВД.N:016777 ЭКЛЗ: 420487422
РЕГ.N:29182 ИНН: 54Г)5289421
.
01 МЗ-ДУДУКОВА В.В. КАССА.ОО14 ОТДО1
ЧЕК НА ПРОДАЖУ N.367 -
НОМЕР СМЕНЬ: 545
----------------------------------------
1 .300231289 КИВИ В КОРЗИНКЕ 1УП
99.90* 1.000(-0.00%)= 99.90
2 .300229817 ПАКЕТ С СИМВОЛИКОЙ МЕГАС 31Х60
3.00* 1.000(-0.00%)= 3-00
3 .300388105 ЛАВАШ АРМЯНСКИЙ С СЫРОМ 1ООГР
14.30* 1.000(-52.7%)= 6.76
4 .300301017 ВОДА ФРУТОНЯНЯ 1*5Л ПЛ/БУТ
35.90* 1.000(-52.7%)= 16.98
5 .300399859 КОНФЕТЫ ТВИКС МИНИС БАЛК 1КГ
474.30* 0.190(-52.6%)= 42.65
6 .300234272 СЕЛЬДЕРЕЙ СТЕБЛИ 1КГ
```

```
215.00* 0.382(-0.00%)= 82.13
7 .300326836 ФРУТИЛАД С ЧЕРНОЙ СМОРОД1НОЙ Ф
25.10* 1.000(-52.7%)= 11.87
8 .300401658 КОЗИНАК ЛАЙТ НАТС БАТОНЧИК ФИС
94.00* 1.000(-52.6%)= 44.47
9 .300229978 БАНАНЫ 1КГ
75.90* 0.336(-0.00%)= 25.50
10 .300228841 РЕЗИНКА ЖЕВ ОРБИТ ВАЙТ ФРЕШ ПО
24.90* 1.000(-52.6%)= 11.78
11 .300399858 КОНФЕТЬ МИЛКИ ВЕЙ МИНИС БАЛК 1
474.30* 0.208(-52.6%)= 46.68
12 .300417968 КОНФЕТЫ КИТ КАТ МИНИ ТЕМНЬТЙ ШО
516.00X 0.138(-52.6%)= 33.70
13 .300336869 БИОХЛЕБЦЬТ ЭЛЬМИИКА КУКУРУЗНЬТЕ
30.80* 1.000(-52.6%)-- 14.57
14 .400003871 ЗАКУСКА ИЗ СЕМГИ С/С 1КГ 24Ч Т
1.299.00* 0.056(-52.6%)= 34.41
----------------------------------------
БЕЗНАЛИЧНЬТЙ РАСЧЕТ
----------------------------------------
17.11.15 18:01
ЧЕК
ОПЛАТА
НОМЕР ОПЕРАЦИИ: 0090
ТЕРМИНАЛ: 00444127
Г -LУНКТ ОБСЛУЖИВАНИЯ: 440000003949
```

Unfortunately, this method performs stably only when characters are not broken or connected between each other.

3.2. Recognition of complete words

It is required to recognize complete words in complicated cases when characters are broken or are connected between each other. We solved this problem using two methods:
LSTM network
Uniform segmentation

LSTM network

We decided to use an LSTM neural network to recognize complete words in complex cases in accordance with the articles devoted to the reading text in deep convolutional sequences and using LSTM networks for language-independent OCR. For this purpose, we used the library OCRopus.
We used monospaced fonts and prepared an artificial sample for training (Image 15).

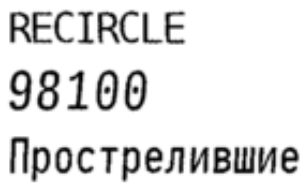
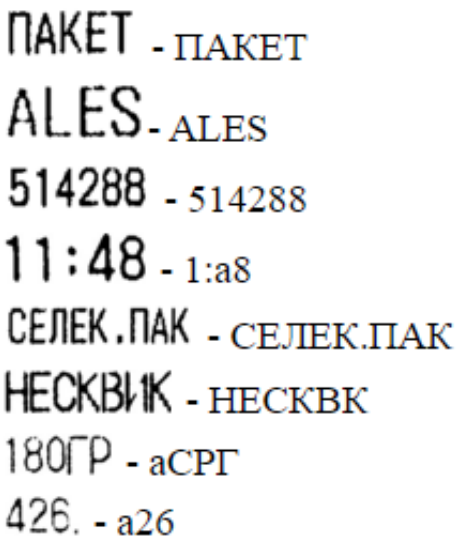


Image 15: Examples of the artificial set

After training the network we tested it with the validation set. Results of testing showed us that the network was trained well. Then we tested it using real receipts. Here are the results:



The trained neural network performed well on simple examples that we successfully recognized using other methods previously. As for complex cases, the network didn't work.
We decided to add various distortions to the training sample to approximate it to the words received in receipts (Image 16).

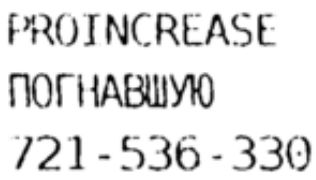
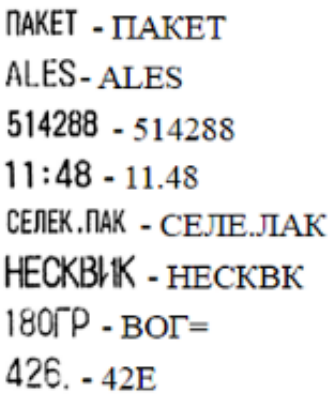


Image 16: Examples of the artificial set

In order to avoid overfitting the network, we stopped the training process periodically, prepared a new dataset and continued training the network with the new dataset. Finally, we got the following results:



The received neural network recognized complex words better but recognized simple words worse. As it was unstable, this model didn't satisfy us.
We think that if you have a single font and minor distortions, this neural network could work much better.

Uniform Segmentation

The font on the receipts is monospaced. For this reason, we decided to split the words into characters uniformly. We need to know the character width inside the word. Thus, the mode of the character width was estimated for every receipt. If we have bimodal distribution of the character width (Image 17), there are two modes chosen and the specific width is picked for every string.

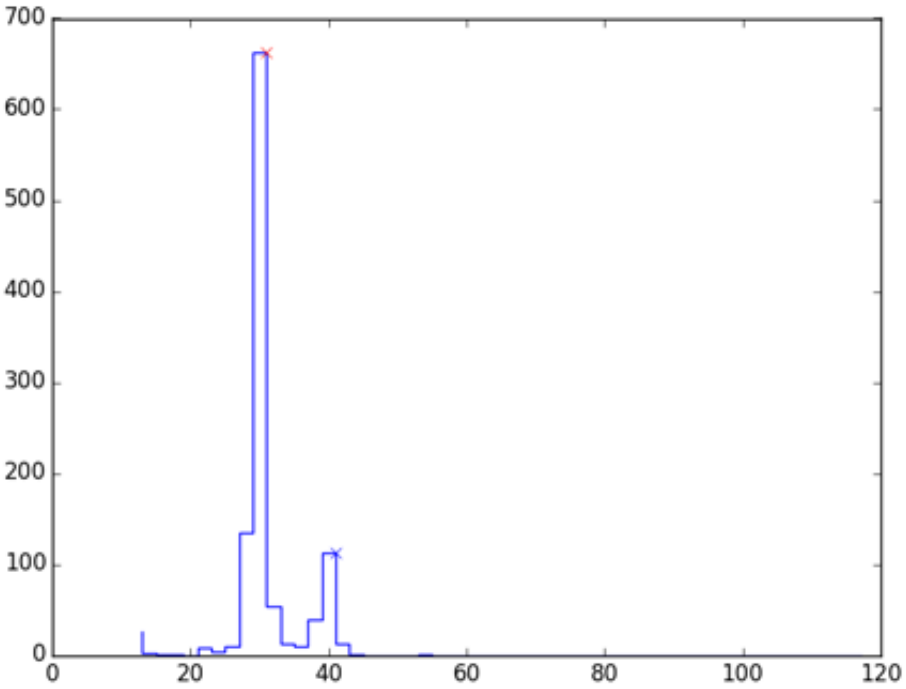


Image 17:

Example of the bimodal distribution of character widths in the receipt

When we get an approximate character width in the string, we divide the length of the word by the character width and get the approximate number of characters. We then divide the length of the word by approximate number of characters plus or minus one:

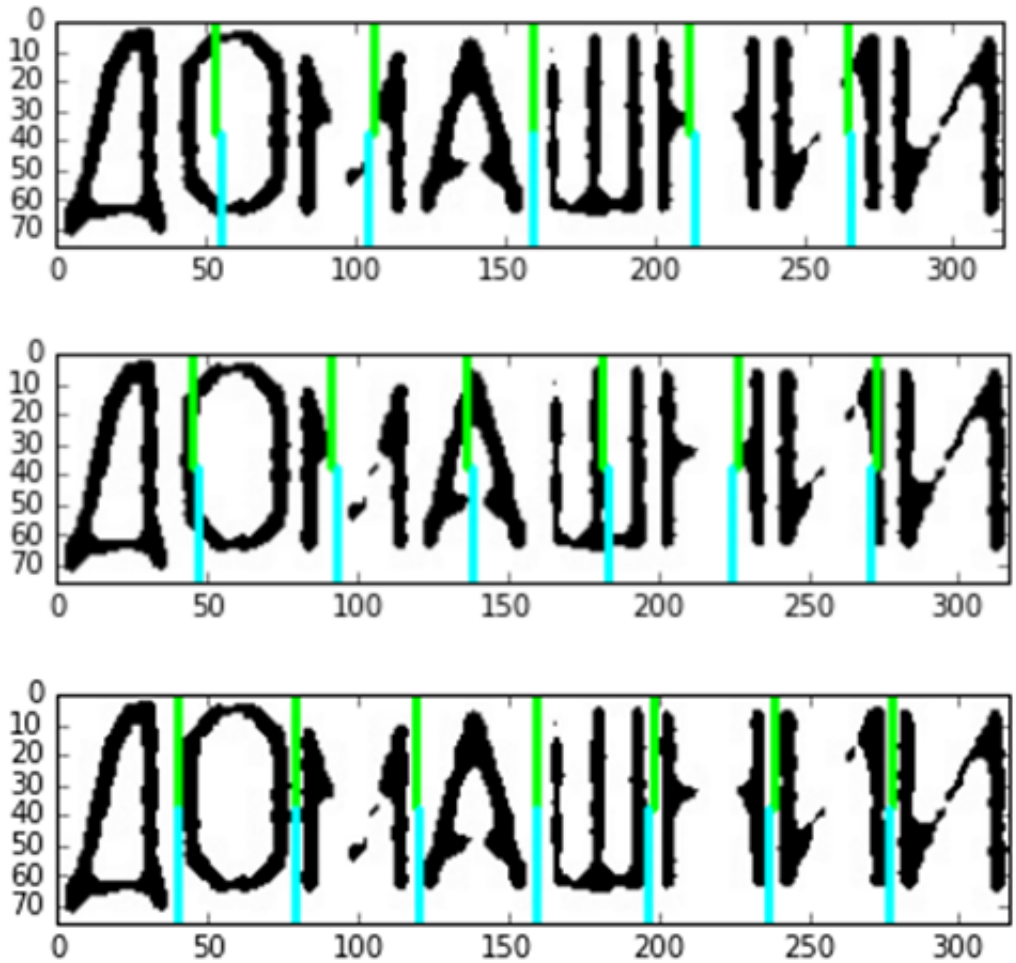


Image 18: The process of finding the optimal segmentation

Choosing the best option for division:

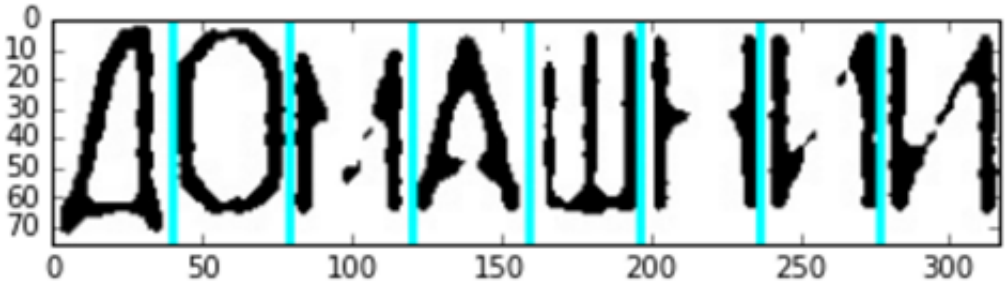


Image 19: Optimal segmentation

The accuracy of such segmentation is quite high.

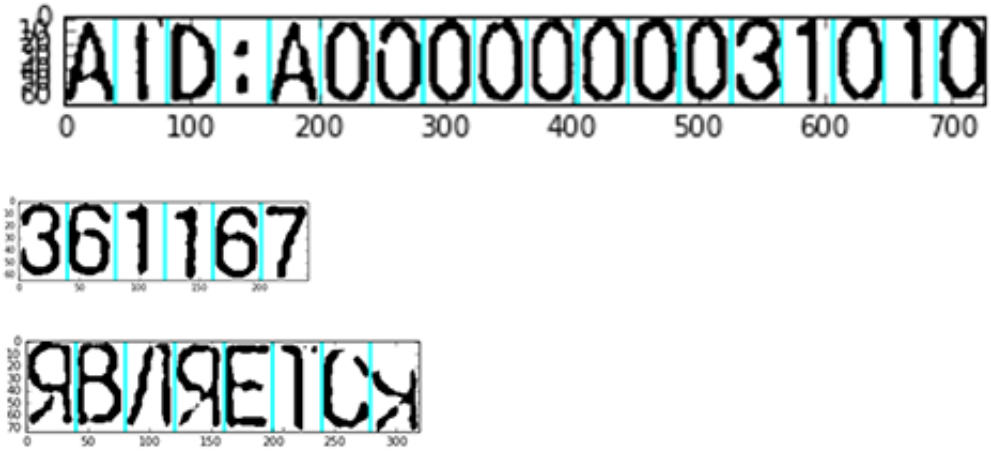


Image 20: Example of how the algorithm works correctly

However, sometimes we saw that the algorithm doesn't work correctly:

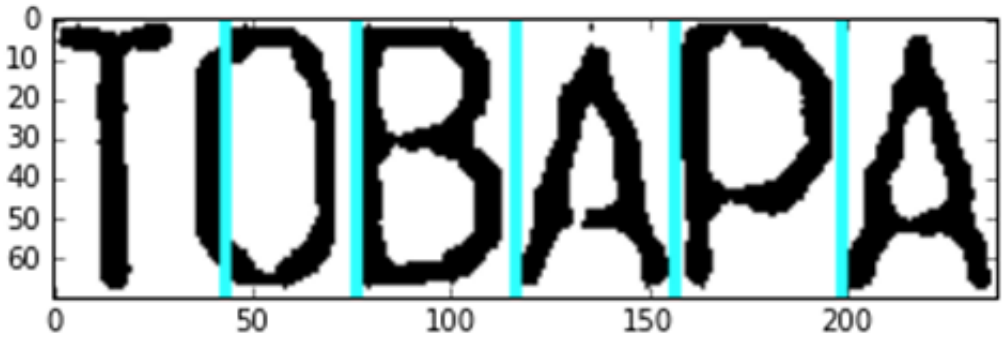


Image 21: Example of how the algorithm works incorrectly

After segmentation, every fragment goes to the convolutional neural network where it is recognized.

4. Extracting meaning from receipts

The search of the purchases in a receipt is implemented by regular expressions. There is one common feature for all the receipts: the price of purchases is written in the XX.XX format, where X is a number. Therefore, it's possible to extract the strings with purchases. The Individual Taxpayer Number can be found by 10 numbers and tested by the control sum. The Cardholder Name has the format NAME/SURNAME.

```
{
    "Cardholder": "NAME/SURNAME",
    "ИНН": "Individual Tax-payer Number",
    "Покупки": {
        "1": [
            "1 .300231289 КИВИ В КОРЗИНКЕ 1уп",
            99.9
        ],
        "2": [
            "2 .300229817 ПАКЕТ С СИМВОЛИКОЙ МЕГАС 31х60",
            3.0
        ],
        "3": [
            "3 .300388105 ЛАВАШ АРМЯНСКИЙ С СЫРОМ 100ГР",
            6.76
        ],
        "4": [
            "4 .300301017 ВОДА ФРУТОНЯНЯ 1*5Л ПЛ/БУТ",
            16.98
        ],
        "5": [
            "5 .300399859 КОНФЕТЫ ТВИКС МИНИС БАЛК 1КГ",
            42.65
        ],
        "6": [
            "6 .300234272 СЕЛЬДЕРЕЙ СТЕБЛИ 1КГ",
            82.13
        ],
        "7": [
            "7 .300326836 ФРУТИЛАД С ЧЕРНОЙ СМОРОД1НОЙ Ф",
            11.87
```


Image 22: Results of the extracting meaning from receipts

Conclusion

The problem of receipt recognition turned out to not be as simple as it seemed from first sight. While we looked for the solution, we faced many subproblems, that are fully or partially related to each other. Moreover, we understood that there is no silver bullet like we thought about LSTM. In practice, the majority of tools requires a lot of time for learning and don't always become useful.

Our work on this project continues. We are concentrated on the improvement of quality for every stage of recognition and on the optimization of concrete algorithms. Currently, the system recognizes the receipts with very high precision if there are no connected or broken characters. When a receipt contains the connected or broken characters, the results are slightly worse than we expect.