Handwritten Grocery List to Bill Conversion

Chatak Vitthal Shinde

Date: 21/06/2022

Abstract

Handwritten Grocery List to Bill Conversion is unique idea. Handwritten text recognition system is very basic application of Artificial Intelligence and Machine Learning. I utilized this idea for specific domain which is local grocery shops. Every customer wants to save their time on grocery shop and also want fairness in rates. This need is fulfilled by my idea which is briefly covered in this prototype report. As per the customers and client need, I tried to mention all-pros of implementing this idea at local grocery shop level. Entire idea about this prototype is in the concept generation and problem clarification section. This report covers introduction, customer need assessment, Revised Needs Statement and Target Specifications, External Search and Consept generation, Concept development and final product, Pseudo Code Implementation for Final Product and conclusion

1.0 Introduction

In local grocery shops, customer comes with handwritten list and vendor need to check each item manually and vendor requires to arrange all items accordingly. This process consumes a time and if all workers in the shop are busy then other customers need to wait until their turn. Even if list is taken by vendor they can’t keep on hold for previous customers. If AI can fix this problem, then why should customer wait?

Fig. 1.1: Indian Local Grocery Shop

We can save time by just taking snap on my developed application and it will be connected to server (computer in grocery shop). It will recognize product name and quantity and it will generate bill accordingly. This will save time of calculation of all items in the list. If my client has many (more than one) branches, then we can connect two servers together so that owner will get daily summary of sales at home. Another advantage is that we will get real time data of sales. Using this data, we can also use recommendation system. We can program such a way that as per recognized list vendor can recommend few more groceries. These can be done by few algorithms like APIORI, FP-Growth and ECLAT. These three algorithms are specially used for product recommendations. Every user will get same rate for same product or groceries. I can also set a program such that it will alert to vendor and owner for shortage of any product.

2.0 Customer Needs Assessment

Owner: As owner of grocery shop is our client they require centralized data of sales of all stores or franchise. They require time to time alert in advance to avoid shortage of any product. They want to apply new technologies like AI to increase their sales for analysing customer's behaviour and to determine general trends in sales. Owner want daily summary of sales in every stores. Fig. 2.1 shows grocery shop owner analysing summary on mobile phone

Fig. 2.1 Grocery Shop Owner

Customer: In my prototype customers are the people visiting grocery shops. They want fairness in rate and their requirements is to reduce time required for entire process till finalising the bill along with fairness in rates. Fig. 2.2 shows Customers waiting in front of Grocery Shop to finalize the bills.

Fig. 2.2 Customers waiting in front of Grocery Shop

3.0 Revised Needs Statement and Target Specifications

3.1 Revised Needs

- To make sales data centralised at local stores
- To make ease for final billing from handwritten list.
- To keep tack on sale of each and every product.
- To analyse the sales data and prepare strategies accordingly.
- Provide same rate for all customers.
- To save time of customers.

3.2 Target Specifications

- Local Language Handwriting Recognition
- Auto Correction
- Bill Finalization
- Data Storing
- Data Pre-Processing
- Data Analysis
- Summary and Predictions

4.0 External Search

4.1 Applicable Papers and Mobile Applications

- Obaid, Ahmed & El-Bakry, Hazem & Eldosuky, M.A. & Shehab, Abdulaziz. (2016). Handwritten Text Recognition System based on Neural Network. International Journal of Advanced Research in Computer Science & Technology. 4. 72-77.
- Dutta, A., Garai, A., Biswas, S. et al. Segmentation of text lines using multi-scale CNN from warped printed and handwritten document images. IJDAR 24, 299–313 (2021). https://doi.org/10.1007/s10032-021-00370-8
- Nurseitov D, Bostanbekov K, Kanatov M, Alimova A, Abdallah A, Abdimanap G. Classification of handwritten names of cities and handwritten text recognition using various deep learning models. arXiv preprint arXiv:2102.04816. 2021 Feb 9.
- "Evernote" Application for Handwriting to Text Conversion
- “Pen to Print” Mobile Applicaion

4.2 Applicable Constraints

- Customer may write numbers in English and contents in local language.
- Customer may write few items in short form.
- Vendor need to update costs of products regularly.
- Bad handwriting may result in wrong text recognition

5.0 Concept Generation

5.1 Problem Clarification

There are various applications are available which recognises handwritten texts but My idea not only recognises the handwritten text but it utilises for the specific application of grocery list to bill conversion.

When customer comes to shops he will give list of groceries items to the vendor. Vendor will collect all items. After collecting all items instead of calculating the total cost he will scan the list using developed mobile application. To scan the list vendor, need to choose the language first. Due to region wise language constraints in idea we will allow them to choose the language in which the grocery's items list is created. The mobile application will recognise the list as per trained model based on CNN (Convolutional Neural Network) or RNN (Recurrent Neural Network) or RNN with LSTM (Long Shot Term Memory) or any other more effective than above mentioned method.

After that recognized data will be sent to computer as the server system which will process the data and it will extract the required information. About item and quantity. It will generate the bill of the all items and computer contains the data of availability of all products so that if any item is not available it will simply put NA in the place of cost cell. After final printing of bill, it will subtract count of availably of that product accordingly. So that it will automatically alert to vendor and vendor about future shortage well in advance. The message about shortage will be sent to registers mobile application user and they will take actions of orders accordingly.

A programme will be developed which will create summary of sale from data present on server and it will be sent to the mobile application of shop owner. Similarly, at the end of the week it will send weekly summary of the sales to the shop owner. Similarly, for monthly then quarterly and yearly.

5.2 Concept Generation

- **Basic Steps using python**

1. Image Capturing
2. Pass the image to the pytesseract to convert the image to data item.
3. View the key using below command

```
keys = list(data.keys())
print(keys)
```

4. Check the extracted text data using line print(data['text'])
5. Extract the coordinates for the required text using the following command.

```
if re.match(accountIdReg, d['text'][i].lower()):
    (x, y, w, h) = (d['left'][i], d['top'][i], d['width'][i], d['height'][i])
```

6. Repeat the same process for all the required text fields.
7. Use the extracted coordinates to crop the image for the required field and save the cropped image to a specific location.

```
idImageBox = (x + w + 65, y - 20, x, y + h + 5)
# (1662, 2342)
idImage = img.crop(idImageBox)
# accountIdImage1.show()
idImage.save(cropped_dir + '\\id_Extracted_' + fileName + '.png')
```

Note- At this point, we are going to use Handprint, a Python package to convert handwritten images into digital data which later can be saved to CSV data or a database. It also shows annotated images with the text recognized. Install Python’s Handprint application on your server and configuration is also needed to set up Google Cloud Vision and set up credentials to access the API.

8. Download the credentials file after setting up the Cloud Vision account on Google.

```
handprint -a SERVICENAME CREDENTIALSFILE.json
os.system('handprint /a google C:\\Users\\ocr-keras\\cred.json')
```

9. Process these cropped images using Google’s Cloud Vision to extract handwritten values.

```
images = glob.glob(cropped_dir + "\\" + "*.png")
```

```
for image in images:
  os.system('handprint /s google /e ' + image)
```

Fig 5.1 shows the annotated images with the extracted information

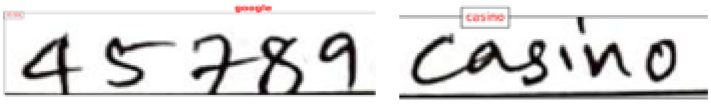


Fig. 5.1 annotated images with the extracted information.

With this, we can confirm that extracted data is accurate. The process above also generates Json files with the extracted data.

10. Then extract the required data from the Json files

```
dictFields = {}
for field in fileds:
   jsonFiles = glob.glob(cropped_dir + "\\" + field +
"*.json")
   for jsonFile in jsonFiles:
       with open(jsonFile) as f:
           temp = json.load(f)
           distros_dict = json.loads(temp)
           print(distros_dict['text_annotations'][0]['
description'])
           dictFields[field] =
distros_dict['text_annotations'][0]['description']
```

11. We have created a Python dictionary to store the extracted information. Convert the dictionary to dataframe for easy processing.

```
dataFrame = pd.DataFrame([dictFields])
print(dataFrame)
```

Below are the dataframe contents.

```
Id        business

45789   casino
```

12. Save this dataframe to CSV format using the following.

```
dataFrame.to_csv(path + '\\' + fileName + '.csv', index=False)
```

This example deals with only two images, the instructions did not include many validations. For a real-world application, analysts should validate the extracted data based on confidence levels and filter out the accurate data.

6.0 Concept Development & Final Product

Initially, we must recognize Handwritten list of item for grocery shop. For that purpose, we can use CRNN (Convolutional-Recurrent Neural Network) for feature extraction from the text image.

After extracting the characteristics, they are employed independently to train a BLSTM-CTC (Bidirectional Long Short-Term Memory - Connectionist temporal classification) network. The output of the CNN layer is sent to the BLSTM, which is used for sequence dependency and time-sequence operations. The network, like alphabets to words, turns a stream of unsegmented data into a one dimensional output vector.

Then CTC LOSS is used to train the RNN which reduces the problem of alignment in Handwritten as every student has different handwriting alignment. When we feed it what is written in the image (Ground Truth Text) and BLSTM output, then it calculates loss simply as log("gtText"); aim to minimize the negative maximum likelihood path and recognizing the text.

Now, that the handwriting is recognized, there is need of processing the data, we will apply stemming, lemmatization, tokenization for reducing the data.

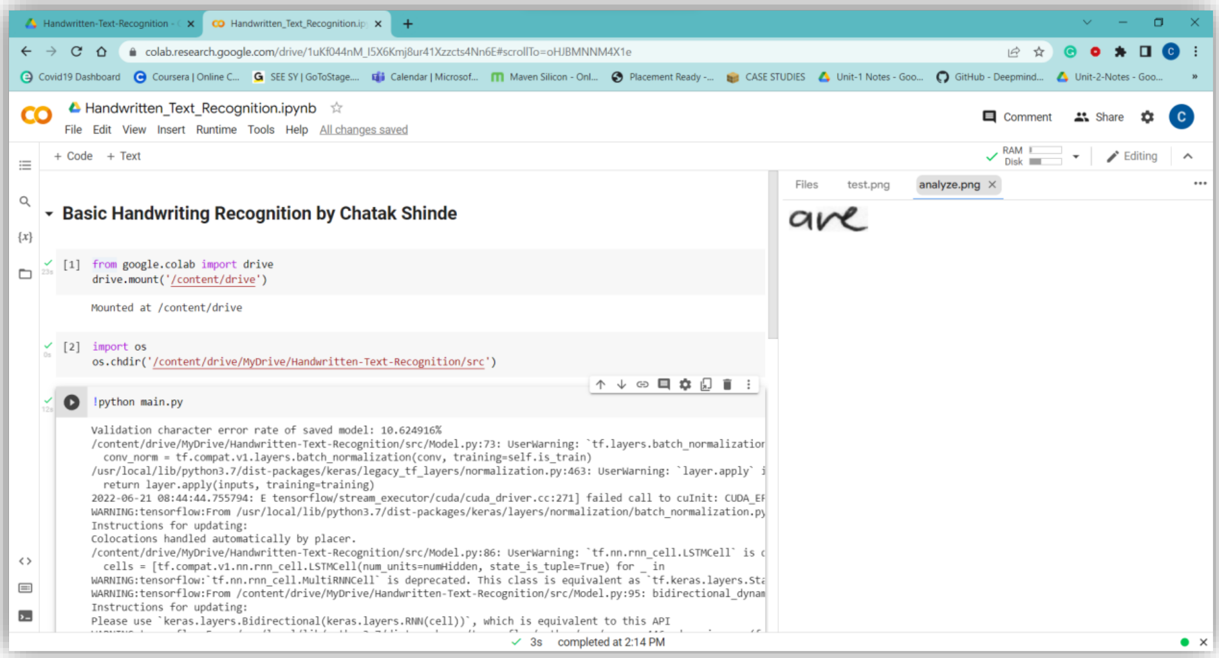
Tokenization is the process of breaking down large pieces of text into smaller ones. Tokenization divides the raw text into words and sentences, which are referred to as tokens.

Stemming is the process of reducing the word to its root form by removing the suffixes and prefixes, even if the word (called stem) has no meaning and is not a valid word.

Lemmatization is the process of extracting the meaningful root word called “Lemma” from the original word. Unlike to stemming this has lemma has a proper dictionary meaning.

Once the data is reduced, System will detect grocery item names and after chaking its availability it will all item name, quantity and cost for each item one by one.

7.0 Pseudo Code Implementation for Final Product



Handwritten_Text_Recognition.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Files test.png analyze.png
are
3s completed at 2:14 PM

Basic Handwriting Recognition by Chatak Shinde

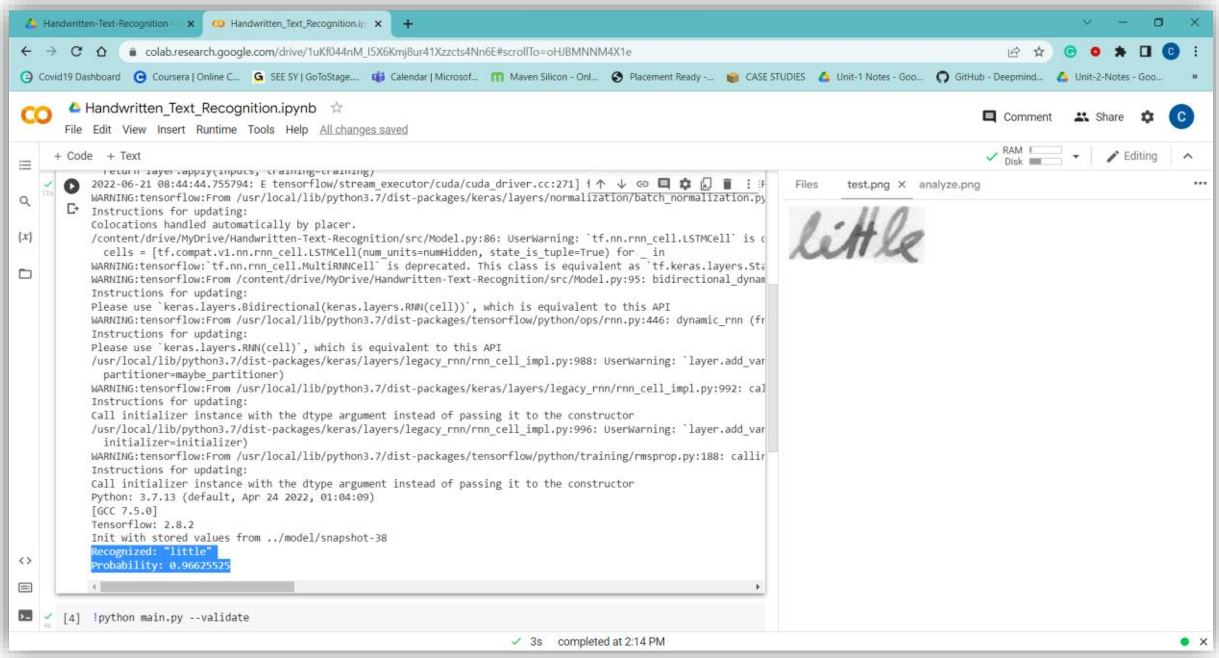
```
[1] from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive

[2] import os
    os.chdir('/content/drive/MyDrive/Handwritten-Text-Recognition/src')

    !python main.py

    Validation character error rate of saved model: 10.624916%
    /content/drive/MyDrive/Handwritten-Text-Recognition/src/Model.py:73: UserWarning: `tf.layers.batch_normalizatior
      conv_norm = tf.compat.v1.layers.batch_normalization(conv, training=self.is_train)
    /usr/local/lib/python3.7/dist-packages/keras/legacy_tf_layers/normalization.py:463: UserWarning: `layer.apply` i
      return layer.apply(inputs, training=training)
    2022-06-21 08:44:44.755794: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] failed call to cuInit: CUDA_EF
    WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/keras/layers/normalization/batch_normalization.py
    Instructions for updating:
    Colocations handled automatically by placer.
    /content/drive/MyDrive/Handwritten-Text-Recognition/src/Model.py:86: UserWarning: `tf.nn.rnn_cell.LSTMCell` is c
      cells = [tf.compat.v1.nn.rnn_cell.LSTMCell(num_units=numHidden, state_is_tuple=True) for _ in
    WARNING:tensorflow:`tf.nn.rnn_cell.MultiRNNCell` is deprecated. This class is equivalent as `tf.keras.layers.Sta
    WARNING:tensorflow:From /content/drive/MyDrive/Handwritten-Text-Recognition/src/Model.py:95: bidirectional_dynam
    Instructions for updating:
    Please use `keras.layers.Bidirectional(keras.layers.RNN(cell))`, which is equivalent to this API
```



Handwritten_Text_Recognition.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Files test.png analyze.png
little
3s completed at 2:14 PM

```
    2022-06-21 08:44:44.755794: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] f
    WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/keras/layers/normalization/batch_normalization.py
    Instructions for updating:
    Colocations handled automatically by placer.
    /content/drive/MyDrive/Handwritten-Text-Recognition/src/Model.py:86: UserWarning: `tf.nn.rnn_cell.LSTMCell` is c
      cells = [tf.compat.v1.nn.rnn_cell.LSTMCell(num_units=numHidden, state_is_tuple=True) for _ in
    WARNING:tensorflow:`tf.nn.rnn_cell.MultiRNNCell` is deprecated. This class is equivalent as `tf.keras.layers.Sta
    WARNING:tensorflow:From /content/drive/MyDrive/Handwritten-Text-Recognition/src/Model.py:95: bidirectional_dynam
    Instructions for updating:
    Please use `keras.layers.Bidirectional(keras.layers.RNN(cell))`, which is equivalent to this API
    WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/ops/rnn.py:446: dynamic_rnn (fr
    Instructions for updating:
    Please use `keras.layers.RNN(cell)`, which is equivalent to this API
    /usr/local/lib/python3.7/dist-packages/keras/layers/legacy_rnn/rnn_cell_impl.py:988: UserWarning: `layer.add_var
      partitioner=maybe_partitioner)
    WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/keras/layers/legacy_rnn/rnn_cell_impl.py:992: cal
    Instructions for updating:
    Call initializer instance with the dtype argument instead of passing it to the constructor
    /usr/local/lib/python3.7/dist-packages/keras/layers/legacy_rnn/rnn_cell_impl.py:996: UserWarning: `layer.add_var
      initializer=initializer)
    WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/training/rmsprop.py:188: callir
    Instructions for updating:
    Call initializer instance with the dtype argument instead of passing it to the constructor
    Python: 3.7.13 (default, Apr 24 2022, 01:04:09)
    [GCC 7.5.0]
    Tensorflow: 2.8.2
    Init with stored values from ../model/snapshot-38
    Recognized: "little"
    Probability: 0.96625525

[4] !python main.py --validate
```

Above screenshots shows that given word is accurately detected and converted into the text. Here, I got same text as written in the image and with probability of 96.62%.

8.0 Conclusion

This prototype will surely help local grocery shops to improve their sales, business and number of daily customers. In this prototype, grocery shop vendor will scan the handwritten list and it will be recognized by application and text will be sent to server which will generate the bill. Data of customers will be stored in server will be utilised for further business strategies like customer trends, sales trends, brand trends etc. Shop owner can access all data and time to time summary of sales any time though their mobile phone.

I have shown basic implementation of handwriting recognition. To make this idea possible we need to work on it for implementing it on mobile application, choice of different languages, auto correction system and server implementation or programing for data analysis.

9. References

- Dutta, A., Garai, A., Biswas, S. et al. Segmentation of text lines using multi-scale CNN from warped printed and handwritten document images. IJDAR 24, 299–313 (2021). https://doi.org/10.1007/s10032-021-00370-8
- Nurseitov D, Bostanbekov K, Kanatov M, Alimova A, Abdallah A, Abdimanap G. Classification of handwritten names of cities and handwritten text recognition using various deep learning models. arXiv preprint arXiv:2102.04816. 2021 Feb 9.
- Roy, Akash. AKHCRNet: Bengali Handwritten Character Recognition Using Deep Learning (2020).
- https://blog.devgenius.io/handwritten-text-recognition-using-convolutional-neural-networks-cnn-714d69d28c9a
- Manchala, Sri & Kinthali, Jayaram & Kotha, Kowshik & Kumar, Jagilinki. (2020). Handwritten Text Recognition using Deep Learning with TensorFlow. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS050534.
- https://towardsdatascience.com/build-a-handwritten-text-recognition-system-using-tensorflow-2326a3487cd5
- https://blog.hexstream.com/process-handwritten-text-using-python-and-cloud-vision