[For abstract .. we may highlight our product's exclusive features for invoice detection and extraction. General OCR engines is not optimized for invoices / bills.]

This module helps to extract relevant content from invoices of various categories automatically ,efficiently and accurately. So that the automation reduces human efforts and human mistakes while typing. Hence reducing the turnaround time of invoice processing and giving a better user experience.

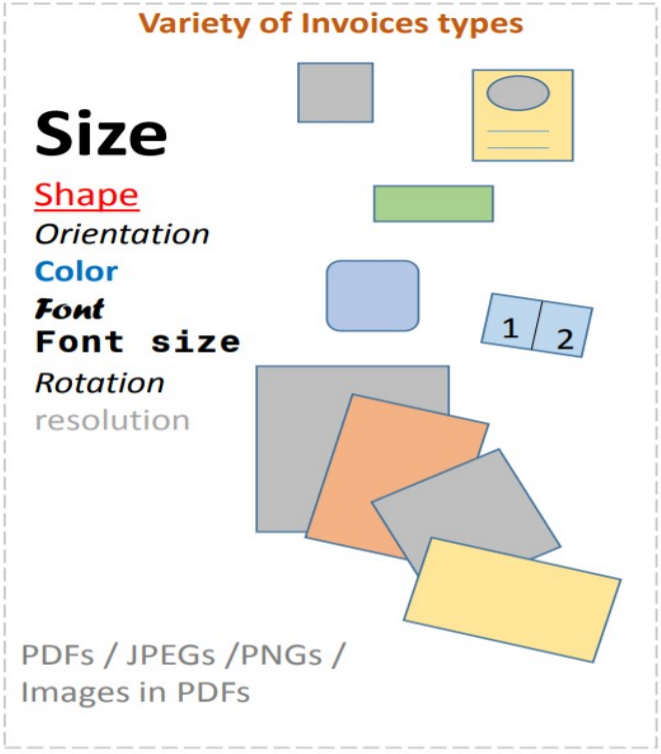
As there is diversity of the type of services, there are many fold diversity in the bills/invoices produced for the same. We could find invoices/bills of various size or dimension, orientation, paper color, font types, font size, font color, skewness, rotation, resolution or clarity.

If we even try to store the physical copies of the same, it will be a daunting task. Now searching on the bills will make the complexity many folds. Our solution hence helps to simplify the same and provide effective and fast solution by making it digital automatically.

Intelligence comes in two parts. First, we are making the physical invoices digital. Secondly, we able to extract relevant information from the unstructured contents.

Future scope could be to train for more fonts present in invoices, train with extracted contents to learn the final layer of LSTM which could help it understand common words in the invoices and even could handle small spelling mistakes to an extent.

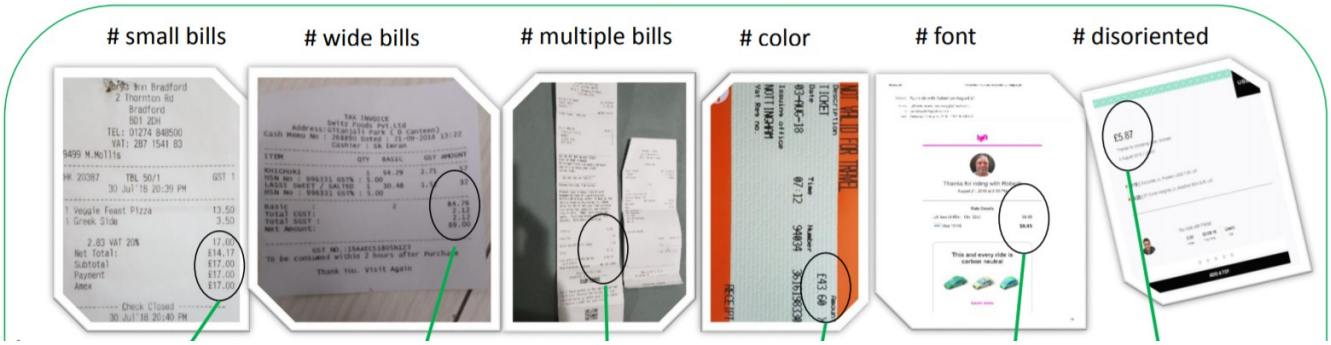
Invoice and bill has same meaning/context.

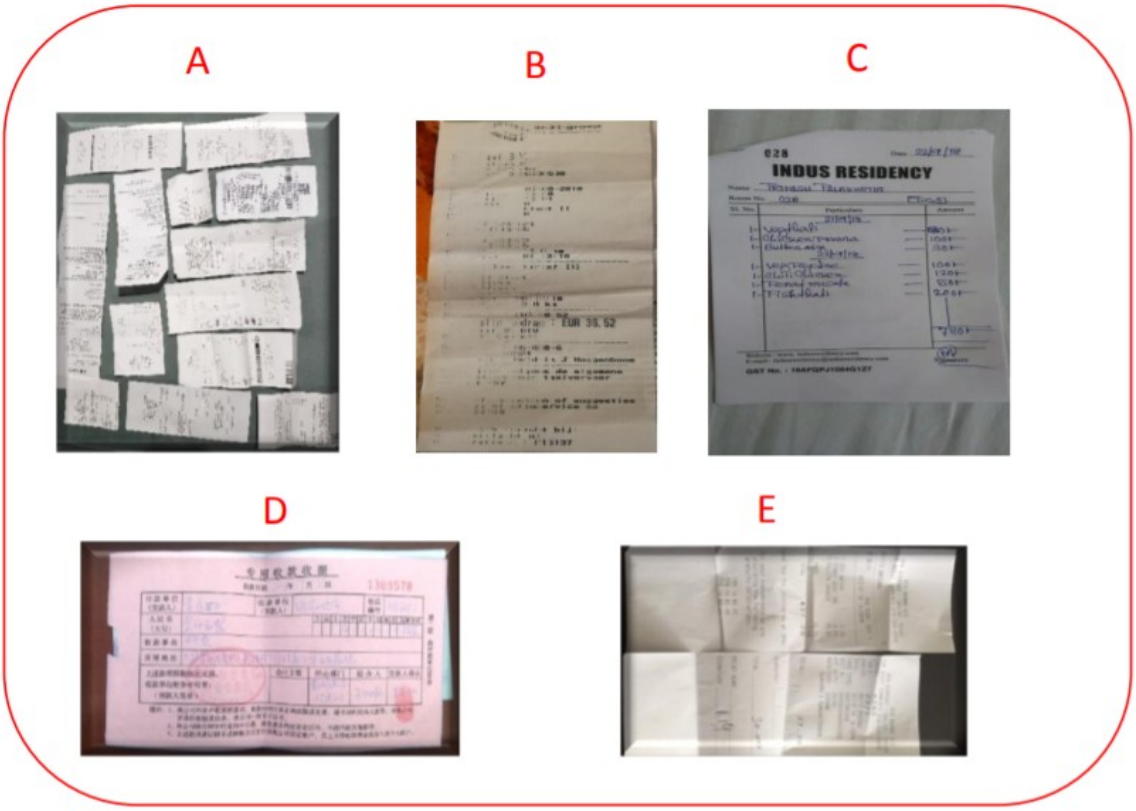


@@ We are not supposed to use .. TCS images .. we may need to download sample images / or create similar ones.

Examples of what types of bills can be there.

Some small , some large, restaurant bills, travel – air / bus / cab , telephone bills etc ..





[At first we could describe, what tesseract does.]

** Tesseract is an OCR (optical character recognition) extraction tool/api. It can handle basic image processing, image correction, OCR recognition and LSTM based word detection.

So most of the clear and straight images can be handled by basic version of tesseract.

Case 1: Perfect images

[have to provide what accuracy is there with basic raw default tesseract module]

Now when we fine tune tesserct detection with following parameters, accuracy can further be increased.

```
C:\Users\vish\Desktop>tesseract.exe
Usage:tesseract.exe imagename outputbase [-l lang] [-psm pagesegmode] [configfile...]

pagesegmode values are:
0 = Orientation and script detection (OSD) only.
1 = Automatic page segmentation with OSD.
2 = Automatic page segmentation, but no OSD, or OCR
3 = Fully automatic page segmentation, but no OSD. (Default)
4 = Assume a single column of text of variable sizes.
5 = Assume a single uniform block of vertically aligned text.
6 = Assume a single uniform block of text.
7 = Treat the image as a single text line.
8 = Treat the image as a single word.
9 = Treat the image as a single word in a circle.
10 = Treat the image as a single character.
-l lang and/or -psm pagesegmode must occur before anyconfigfile.

Single options:
   -v --version: version info
   --list-langs: list available languages for tesseract engine
```

[we could show examples [psm] from the commands, we have run. Present in the code]

But still tesseract detection has a limit. When the image is not clear or has skewness or is tilted or rotated, tesseract fails to detect correct OCR. Now when we realign manually, tessercat happens to work better.

Case 2 : Single images with some defects

Now we can discuss, where tesseract fails. And our next image processing pipeline helps exactly in solving that. This module helps to realign the images , make auto thersholding to make the input to the tesseract better. Hence the over all accuracy is improved.

[An example of how we handle :

- tilting
- skewness
- improper lighting / unclear
- etc

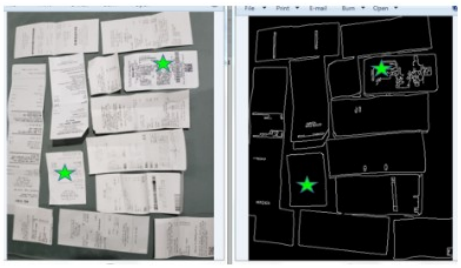
[also examples to show , on the same test batch how much accuracy gain did it make.]

Case 3 : Multiple images and multiple defects

Deskewing

Handling multiple bills and skewness

Handling multiple bills



Extracting multiple bills

Now when there is multiple images, basic tesseract fails to recognize any. Even google api, can’t find multiple bills and its corresponding entities. In general, invoices does have such cases.

[a few togh example.. 2 bills , 3+ bills, having different ways of tilted images in a single page , rotated bills]

Here , our module can find the edges of bills and extract each individual bills and its content.

[insert a few examples]

……..

Further enhancements / tweaks:

An improvement we tried was to segment the bills based on the white spaces and some other criteria. Now each segment is passed on to tesseract and the outputs are merged before sending to the entity module.