Making Intelligent Document Processing Smarter

Part 1 – Figuring out the gaps

Akshay Kumar, Dhiraj Hasija, Abhijeet Yadav, Shiyam Cumar, Namit Bhardwaj, Aniruddh Rawat, Vijendra Jain, Sigmoid Analytics, Bangalore
https://www.sigmoid.com/
August 2022

1. Introduction:

OCR i.e., Optical Character Recognition has been a significant tool to improve productivity and to automate the business processes. Some of the examples are automating the license/ number plate detection, invoice processing, customer onboarding process in insurance companies etc.

Intelligent Document Processing (IDP) is an extension of the OCR where scanned documents are processed and text is extracted from them.

As per research firm Verified Market Research (VMR), Global market for IDP (Intelligent Document Processing) is expected to reach 6,324 million dollars by 2028 with a CAGR of 35.35%. This article focuses on exploring existing OCR APIs and attempts to determine the gaps or areas of improvement in those APIs. Essentially, two major players in this segment: Amazon's API Textract and Google's Vision API along with open source API Tesseract have been tested using various datasets to list out the strengths and limitations of these APIs. Our hypothesis is that accuracy of these OCR APIs might suffer due to various noises present in the scanned documents like Blurs, Watermarks, Faded text, Distortions etc.

This article attempts to measure the effect of such noises on the performance of various APIs, In order to establish, "Is there a scope to make Intelligent Document Processing smarter?"

This research is a two part series wherein, first part identifies the need to denoise a document before feeding it to an OCR API and the second part explores denoising methods to enhance the API's performance. Part 2 is in progress and this article summarises the first part.

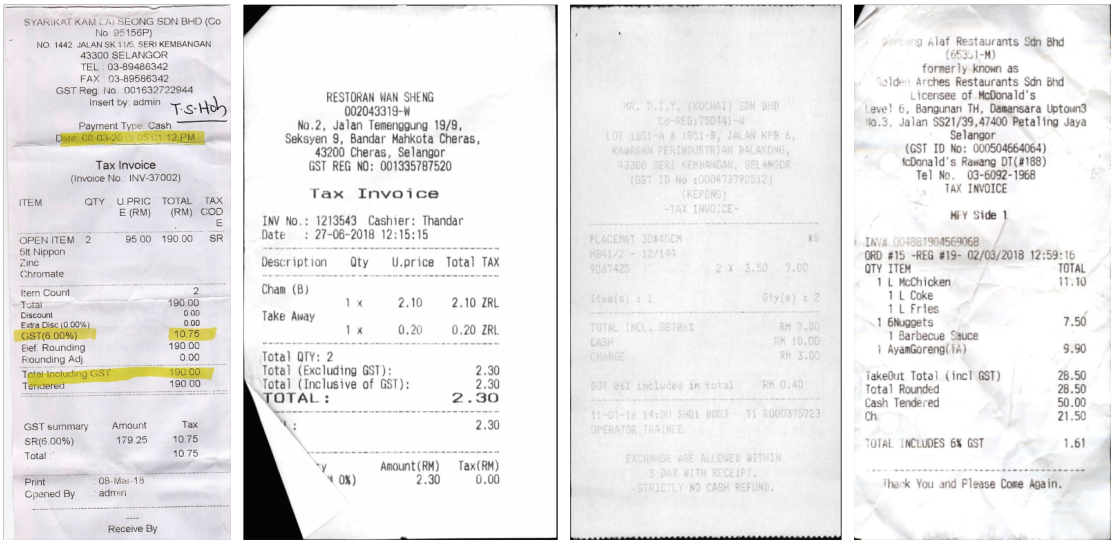
2. Types of Noises in Documents

There are various types of noises in the documents which can lead to poor accuracy of OCR. These noises can be divided into two categories:

2.1 Noises due to the document quality

- Paper Distortion - Crumbled Paper, Wrinkled Paper, Torn Paper
- Stains - Coffee Stains, Liquid spill, ink spill
- Watermark, stamp
- Background Text
- Special Fonts

Below are some of the examples of these noises:

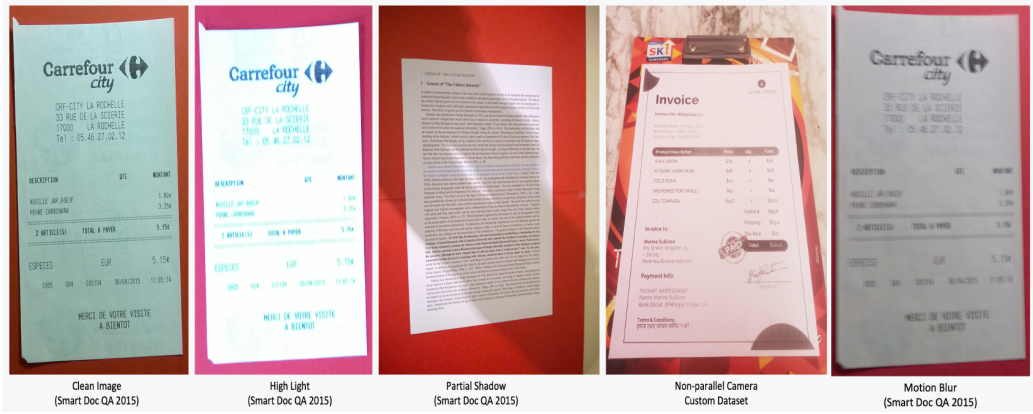


Folded Sheet & Highlighting (SROIE 2019) | Missing Words due to Folding (SROIE 2019) | Faded Font (SROIE 2019) | Wrinkle (SROIE 2019)

Fig 2.1 Noises due to the document quality

2.2 Noises due to image capturing process:

1. Skewness - Warpage, Non-parallel camera
2. Blur - Out of Focus Blur, Motion Blur
3. Lighting Conditions - Low Light (Underexposed), High Light (Overexposed), Partial Shadow



Clean Image (Smart Doc QA 2015) | High Light (Smart Doc QA 2015) | Partial Shadow (Smart Doc QA 2015) | Non-parallel Camera Custom Dataset | Motion Blur (Smart Doc QA 2015)

Fig 2.2 Noises related to image capturing process

Because of the presence of these noises, images need pre-processing/ cleaning before being fed to an IDP/OCR pipeline. Some OCR engines have built-in pre-processing tools which can handle most of these noises. Our aim is to test the APIs with variety of noises in order to determine which noises the OCR APIs can handle and which noises they cannot.

3. Metrics to measure performance of the API:

To measure the performance of an OCR engine, ground truth (actual text) is compared with the OCR output or the text detected by the API. If the text detected by the API is exactly same as the ground truth, that means accuracy is 100% for that document. But this is a very ideal case. In real world, detected text will differ from the ground truth because of the noises present in the document. This difference between ground truth and detected text is measured using various metrics.

Following table lists the metrics that we have considered to measure the performance of the APIs. Except the first metric (Mean Confidence Score), rest all the metrics compare the detected text with the ground truth.

S. No.	**Metric**	**Type**	**Brief Description**
1	Mean Confidence Score	Given by API	Confidence score indicates the degree to which the OCR API is certain that it has recognized the text component correctly. Mean confidence score is the average of all the word level confidence scores.
2	Character Error Rate (CER)	Error Rate	The CER compares the total number of characters (including spaces) in the ground truth, to the minimum number of insertions, deletions and substitution of characters that are required in the OCR output to obtain the ground truth result. $CER = \frac{\text{(Substitutions + Insertions + Deletions) in OCR Output}}{\text{Number of Characters in Ground Truth}}$
3	Word Error Rate (WER)	Error Rate	It is similar to CER but the only difference is that WER operates at word level instead of characters. $WER = \frac{\text{(Substitutions + Insertions + Deletions) in OCR Output}}{\text{Number of Words in Ground Truth}}$
4	Cosine Similarity	Similarity	If x is mathematical vector representation of the ground truth text and y is mathematical vector representation of the OCR output, the cosine similarity is defined as below: $Cos(x, y) = x \cdot y \ / \ \|\|x\|\| * \|\|y\|\|$
5	Jaccard Index	Similarity	If A is set of all the words from Ground Truth and B is set of all the words in OCR output, the Jaccard Index is defined as below: $J = \frac{\|A \cap B\|}{\|A \cup B\|}$

Note that WER and CER are affected by the order of the text whereas Cosine Similarity, Jaccard Index and Mean Confidence Score are independent of the text order. Consider a case where an OCR API detects all the words correctly, but if the order of the detected words is different from the Ground Truth, then WER/CER will be very poor (high error i.e. poor performance) whereas Cosine Similarity will be very good (high similarity, i.e. good performance). Hence it is important to see all the metrics together to get the clear idea of the OCR API's performance.

4. Data Sets Explored

We have explored some standard datasets available in the literature and we also created some custom datasets using the real world invoices and dummy invoices. Summary of these datasets is given below:

S. No.	Dataset	Brief Description	Total Documents	Noises / Variations
1	Noisy Office 2007	Real: Text documents printed, noised and scanned. Simulated: Noise added on the text images digitally.	360	Coffee Stains, Footprints, Folding, Wrinkle, Font Type and Size
2	Smart Doc QA 2015	Text documents (admin docs/contemporary/invoices) printed and scanned under various noisy conditions.	4260	Lighting Variations, Partial Shadow, Skewness, blur
3	SROIE 2019	Real receipts/bills	1000	Real world Noises like Stamp, blur, font, wrinkle, warpage/skewness, poor paper quality, poor printing quality
4	Custom Dataset 1 (Real Invoices)	Real receipts/bills scanned under various noise conditions.	56	Lighting, Skewness, warpage, blur, watermark
5	Custom Dataset 2 (Invoices Alpha Foods)	Dummy invoices printed and scanned under various noise conditions	120	Lighting, Skewness, warpage, low light
6	Custom Dataset 3 (Real Invoices)	Real receipts/bills scanned under various noise conditions	48	Lighting, Skewness, warpage

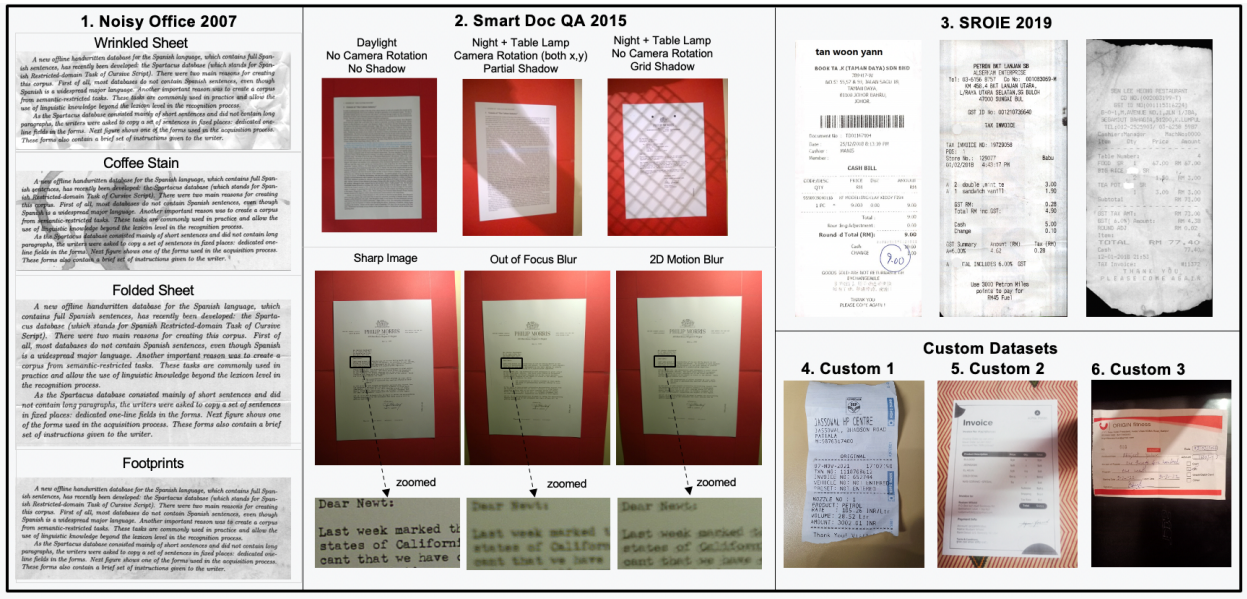


Fig 4.1 – Sample images from various datasets explored. Left box: Noisy Office; Middle Box: Smart Doc QA; Right Top Box: SROIE Dataset; Right Bottom Box: Custom Datasets.

5. Results Summary

As mentioned earlier, we tested three APIs Vision, Textract and Tesseract on the mentioned datasets and calculated the performance metrics. We observed that Tesseract's performance is significantly poor than Vision and Textract in almost all the cases, hence, in the result summary, Tesseract is excluded. Our result summary is divided into two parts, first based on the dataset and second based on the noise type.

5.1 Results Summary (Based on the Dataset):

We have classified the metrics into two sets, first set includes error rates (WER & CER) and second set includes similarity metrics (Cosine Similarity & Jaccard index). APIs are compared using means of these metrics for their respective sets. This is used to develop a rating system ranging from 1 to 10. Here a rating of 1 represents that the mean performance metric for that dataset is between 0-10% (Worst Performance), whereas a rating of 10 represents it is between 91-100% (Best Performance).

Sl. No.	Data Set	Major Noise on which API performs poorly		API Performance Relative Rating (1: Worst, 10: Best)			
				Based on Error Rate		Based on Similarity Metrics	
		Vision	Textract	Vision	Textract	Vision	Textract
1	Noisy Office 2007	Both APIs work well on all the noises of Noisy office dataset		10	10	10	10

2	Smart Doc QA 2015	Motion Blur, Out of Focus Blur, Invoice Type Documents	7	9	9	8
3	SROIE 2019	Dot Printer Font, Stamp	6	9	10	10
4	Custom Dataset 1	Blur and watermark	6	9	10	9
5	Custom Dataset 2 (Alpha Foods)	Both APIs work good on all the noises	5	7	10	9
6	Custom Dataset 3	Without solid background (when there is a text present on both the sides)	3	8	10	9

An important point here is that, Vision's CER and WER error rates are generally higher than that of Textract. But the Cosine Similarity and Jaccard Index are similar for both the APIs. This is because of the order of the words or sorting method used by APIs. Our finding is that although both Vision and Textract are detecting texts with almost equal performance, but because of the different ordering in Vision's output, its error rates are higher than that of Textract. Hence Vision shows poor performance based on the error rate.

5.2 Results Summary (Based on Noise):

Here we provide a subjective evaluation of the API based on their observed performance. Right tick (✓) represents that the API can generally handle that particular noise and cross (X) represents that the API generally performs poorly with that particular noise. For example we observed that Textract cannot detect a vertical text in a document.

S. No.	Noise / Variation	Google's Vision API	Amazon's Textract API	Observation
1	Light Variation (Day Light, Night Light, Partial Shadow, Grid Shadow, Low Light)	✓	✓	Both Vision and Textract APIs can handle these kind of noises
2	Nonparallel camera (x, y, x-y)	✓	✓	
3	Uneven Surface	✓	✓	
4	2x Zoom In	✓	✓	
5	Vertical Text	✓	X	Limitation of the Amazon API
6	Without solid background	X	X	Both Vision and Textract APIs tend to perform poor in these kind of noises
7	Watermark	X	X	
8	Blur (Out of Focus)	X	X	
9	Blur (Motion Blur)	X	X	
10	Dot Printer Font	X	X	

Some of the examples are given below:

Fig 5.2 (a): SmartDocQA - Out of focus blur: Vision and Textract text output comparison. Left image is the input and middle one is Vision output where yellow boxes are word level bounding boxes and the right image is Textract output where blue boxes are word level bounding boxes. Red boxes indicate the words without bounding boxes, i.e. the words that have not been detected by the API.

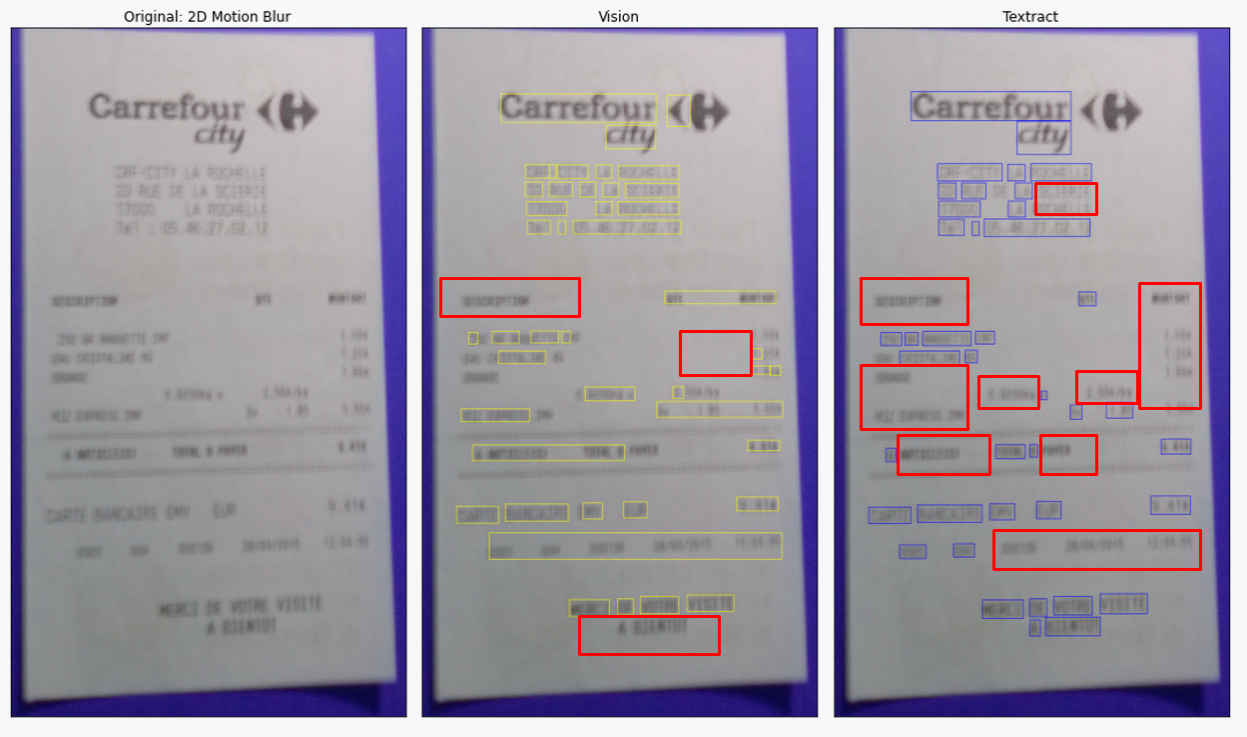


Fig 5.2 (b): SmartDocQA - 2D Motion Blur: Vision and Textract text output comparison. Red boxes indicate the texts that are not recognized by the APIs.

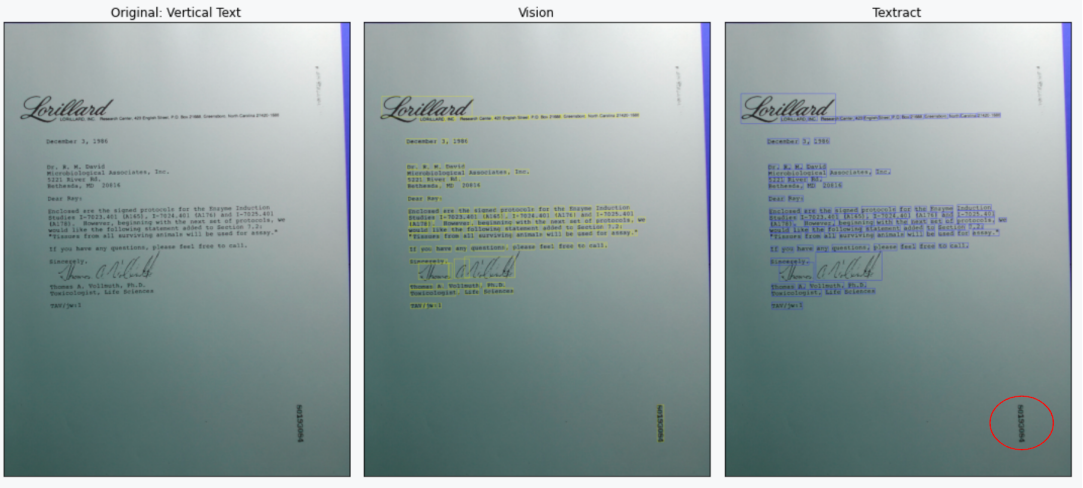


Fig 5.2 (c): SmartDocQA - Vertical text: Vision and Textract text output comparison. Red circle indicates that Textract API is not able to detect the vertical text in the image.

6. Document Denoising:

It is now established that some noises do affect API's text recognition capabilities.
Hence we tried various methods to clean the images before feeding to the API and checked whether API performance improved or not. We have provided links in the reference section where these methods can be understood. Below is the summary of the observations:

S. No.	Noise / Variation	Cleaning Method	Total Samples	Observation	
				Vision	Textract
1	Blur (Out of Focus)	Kernel Sharpening	1	Degraded	No Effect
		Custom Pre-processing	3	1: Improved 1: Slightly Improved 1: Slightly Degraded	1: Improved 2: No Effect
2	Blur (2D Motion Blur)	Blurring (Average/Median)	2	2: Improved	1: Slightly Degraded 1: Improved
		Kernel Sharpening	2	1: No effect 1: Degraded	1: Degraded 1: Improved
		Custom Pre-processing	1	No effect	1: No effect
3	Horizontal Motion Blur)	Blurring (Average/Median)	2	1: Degraded 1: Improved	1: Slightly Improved 1: Degraded

		Custom Pre-processing	1	Slightly Improved	No Effect
4	Watermark	Morphological Filtering	2	1: Improved 1: Degraded	1: Improved 1: Degraded

As seen from the table, these cleaning methods do not work on all the images and in fact sometimes the API performance degrades after applying these cleaning methods. Hence there is a need of a unified solution which can work on all kind of noises.

7. Conclusion

Below is our summary of our experiments:

- In this study, we tested various datasets (Noisy Office, Smart Doc QA, SROIE and custom datasets) to evaluate and compare the performance of OCR APIs (Tesseract, Vision and Textract).
- Our studies and experiments show that OCR output gets affected by the noises present in the documents. There is indeed a scope of improvement in the existing OCR APIs. The inbuilt denoiser or pre-processor is not sufficient to handle most of the noises (example: motion blur, watermark etc.)
- If the document images are denoised, the OCR output can improve significantly.
- The noises in the documents are diversified and we tried various non-model methods to clean the images. Different methods work for different kind of noises. Also Currently there is no unified option available which can handle all kinds of noises or at least major noises.
- Hence there is a scope to make Intelligent Document Processing smarter. There is a need of a unified ("one model fitting all") solution which can denoise the document before inputting to the OCR API to improve the performance.

This article is a summary of our research done to explore the limitation of the OCR text extraction APIs. Complete report is can be referred here.

8. References

1) F. Zamora-Martinez, S. España-Boquera and M. J. Castro-Bleda, Behaviour-based Clustering of Neural Networks applied to Document Enhancement, in: Computational and Ambient Intelligence, pages 144-151, Springer, 2007.
   UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/NoisyOffice]
2) Castro-Bleda, MJ.; España Boquera, S.; Pastor Pellicer, J.; Zamora Martínez, FJ. (2020). The NoisyOffice Database: A Corpus To Train Supervised Machine Learning Filters For Image Processing. The Computer Journal. 63(11):1658-1667. https://doi.org/10.1093/comjnl/bxz098
3) Nibal Nayef, Muhammad Muzzamil Luqman, Sophea Prum, Sebastien Eskenazi, Joseph Chazalon, Jean-Marc Ogier: "SmartDoc-QA: A Dataset for Quality Assessment of

Smartphone Captured Document Images - Single and Multiple Distortions", Proceedings of the sixth international workshop on Camera Based Document Analysis and Recognition (CBDAR), 2015.

4) Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shjian Lu, C. V. Jawahar , ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction, (SROIE) 2021 [arXiv:2103.10213v1]
5) https://cloud.google.com/vision
6) https://aws.amazon.com/textract/
7) https://docs.opencv.org/3.4/d4/d13/tutorial_py_filtering.html
8) https://scikit-image.org/docs/stable/auto_examples/applications/plot_morphology.html
9) https://pyimagesearch.com/2014/09/01/build-kick-ass-mobile-document-scanner-just-5-minutes/