A coordinate-geometry based approach for document deskewing in maritime digital KYC processes

Narayanan Arvind (Yetirajan*)[a]

[a]IN-D by Emulya Technologies Pvt. Ltd.

Abstract

ID documents submitted for Maritime digital KYC processes can be skewed due to the environment in which the photograph is taken or due to user preferences and/or errors. The skewed image results in a low accuracy in downstream image processing tasks like optical character recognition (OCR). ID document deskewing has been typically approached using deep learning (MaskRCNN), regression, projection plans, Hough transforms, Fourier transforms and other computer vision techniques. The aim of this study is to build a robust document deskewing system based on keyword detection and coordinate geometry. The research is carried out by analyzing skewed Indian PAN cards available with IN-D. The database has 50 Indian PAN card images. These images are augmented to generate 150 images, with 50 images for each of the +90, -90 and 180 degree skew cases. Google Vision API is used as the OCR engine for finding the coordinates of the keyword in our study. The research employs Numpy, Pandas and OpenCV open-source libraries for Python. The accuracy of the reported model is 95.33 %. The accuracy of our present approach surpasses the accuracy of all the models available in literature.

Key words: Deskewing, Maritime KYC, digital KYC, Computer vision, Hough transform, Fourier transform, Python, Google Vision, OCR, optical character recognition, OpenCV, coordinate-geometry

[a] Corresponding author, Email: arvind.narayanan@in-d.ai, Tel: +91-8349232657

B.Tech. / M.Tech. IIT Kharagpur (Dept. of Ocean Engineering and Naval Architecture)

**meaning king of the ascetics*

1. Introduction

KYC or “know your customer” check is performed by shipping operators, logistics companies and various other shipping industry segments to verify the identity of an individual. These checks can be performed at the time of onboarding or at other regular intervals of time for identity verification. As economies go paperless, the ID documents are being submitted in a digital format. The digital copies of the ID documents can be skewed due to the environment in which the photograph is taken or due to user preferences / errors. The skewed image results in low accuracy in downstream image processing tasks like optical character recognition (OCR) for intelligent document processing. Typical solutions for handling ID document skewing in the industry involve semantic/instance segmentation using a deep learning model like MaskRCNN [8] followed by a four point perspective transformation. A challenge of this method is the higher number of samples required for training and a relatively higher latency as compared to the present method. A yet another challenge of this method is the difficulty in handling 180-degree skewed images. UI based deskewing where the user provided boundary points on an ID card are used for deskewing the ID card is another relevant technology. However, end-to-end automation using this technology can be challenging. Open source libraries like Pytesseract [1] can be used to find the skew angle of a document followed by a custom computer vision function to deskew the image. However, presently the accuracy of this method is not high. Another approach for deskewing is based on using computer vision to detect the angle of skew of an entire block (blob) of text in the image, converted to a binary image [3]. Another similar computer vision technique involves rotating the image by certain angle increments. The row-wise sum of rotated image pixels is stored in an array. We find the angle of rotation by considering the angle which causes the maximum sum of mean squared error between the pairs of neighboring elements of the array starting with the first two elements and ending at the last two elements. A machine learning approach to deskewing can be to label a multitude of skewed images with their respective skew angles and train a regression model to detect the skew. Challenges of this method are the human effort required in labeling the samples and the large number of samples required to train the model. Other models include Projection profiles (which are limited to -10 to 10 degrees) and Hough & Fourier transforms, which are mathematically challenging [2]. Our model is simple, robust, has lower latency and guarantees higher accuracy compared to the aforementioned models. In the present study, we apply our deskewing approach to +90, -90 and 180 degree skewed images. Our document deskewing model can be extended to other angles easily with a high accuracy. To our knowledge, this model provides the highest accuracy on document deskewing tasks and is the most robust model available in literature.

2. Coordinate-Geometry for Image deskewing

2.1 Preprocessing

We work with 50 Indian PAN card images available with IN-D. The dataset is augmented to generate 50 documents each for the three skew angles considered in this study: +90, -90, and 180 degree skew angles. Only image documents are considered in this study namely *.jpg, *.jpeg, *.png, *.tiff document types. The model can be easily extended to other document types by using suitable document conversion techniques. We start by the extraction of text from the deskewed images using the Google Vision API. The Google Vision API is used as a component of the proprietary IN-D OCR package developed at IN-D.

2.2 Solution methodology

We find the skew angle of the image by first calculating the coordinates of the keyword "INCOME" in the Driving License sample. Due to the possibility of OCR errors, we check for a fuzzy match with the keyword "INCOME". The coordinates of the keyword are returned by the Google Vision API. we then calculate the slope of the keyword by using the formula:

$$tan\,\theta = \frac{(y_2 - y_1)}{(x_2 - x_1)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Where (x1,y1) and (x2,y2) are the top-left and top-right coordinates of the keyword respectively.

x1,y1 INCOME x2,y2

Figure 1. Keyword "INCOME" and its coordinates.

For example, the slope of the keyword "INCOME" in the above image is

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{0}{x_2 - x_1} = 0$$

We find the skew angle from the slope using the *arctan* method on the slope. We convert the found skew angle from radians to degrees. A custom deskewing code can be used to then rotate the image. This custom function ensures that images are not cropped when they are deskewed.

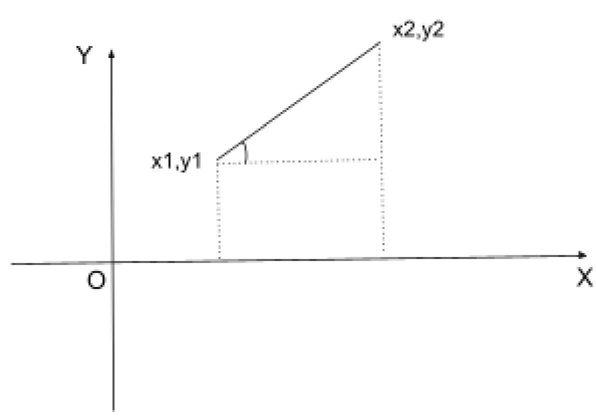

Figure 2. A schematic for calculating the slope of a line using coordinate geometry.

3. Experiments

In this section we propose the methodology for carrying out our experimental studies. We use 50 Indian PAN card images available with IN-D. Python is the chosen language for programming. OpenCV, Numpy, Pandas and Pytesseract OCR engine are the major open-source libraries used. Google Vision API is a Commercial off-the-shelf (COTS) product used in the current study.

3.1 Dataset

The Indian PAN card samples are available with IN-D. The keyword “INCOME” is present in each sample. A unique keyword can be found on various other ID cards for carrying out studies on these other IDs as an extension of the present deskewing method. We use 50 Indian PAN cards for our study. We manually rotate the images and generate 50 samples in each category for +90 (clockwise rotation), -90 and 180 degree skew cases.

3.2 Experimental protocol

We find the keyword coordinates using the Google Vision API. We calculate the slope of the keyword using the modified version of equation (1) as given in equation (2). As we are dealing with only +90, -90 and 180 degree skew cases, the *abs* method ensures that all angles within +/-5 degrees of multiples of 90 degrees (positive and negative slopes) are rounded-off to the nearest multiple of 90. This formula accounts for +90 and -90 degree skews. Next we calculate if the found angle is less than 15 degrees (y2 is close to y1) and if x2 < x1 (as we have taken *abs* of x2-x1 in equation (2)). If this condition is satisfied, then we mark the skew angle as 180 degrees.

$$tan\,\theta = \frac{(y_2 - y_1)}{abs(x_2 - x_1)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

3.3 Results and discussions

Our model achieves an accuracy of 95.33 %. This means, on an average our model predicts the skew angles of 95 images correctly out of 100 given images. Results from other studies are also mentioned for comparison. Our model's accuracy surpasses the accuracy numbers for all the models mentioned in the literature (Table 1). Our model is particularly accurate for +90 and -90 degree cases, commonly found in the KYC processes. For calculating accuracy, we round-off +/-5 degrees deviation from multiples of 90 degrees to the nearest multiple of 90.

S num	Author / Source	Method	+90^0 Acc	-90^0 Acc	180^0 Acc	Total Acc
1	Narayanan et. al. [present]	Coordinate-geometry	100 %	100 %	86 %	95.33 %
2	R Smith [1]	Pytesseract	58 %	60 %	68 %	62 %
3	PyImagesearch [3]	OpenCV minAreaRect	0 % **	62 %	0 % *	20.67 %
4	9SphereAI [2]	Hough transform	16 %	76 %	0 %	30.67 %
5	9SphereAI [2]	Fourier transform	0 % **	96 %	0 %	32 %
6	Stackoverflow [7]	Projection profile	0 % **	100 %	96 %	65.33 %

Table 1. Comparison of accuracy for different methods on the Indian PAN card dataset.

** The model fails to distinguish between +90 and -90 degree skew cases

* The model is limited to angles between [-90,0], hence the zero accuracy.

4. Conclusions

A robust coordinate-geometry based deskewing approach was presented for Intelligent Document Processing (IDP) in maritime digital KYC processes in this paper. Indian PAN card samples, available with IN-D, were used for our experiments. A known keyword and its coordinates are extracted from the documents using Google Vision API which are then used for document deskewing. Our results surpass the performance of all the models available in the literature.

Acknowledgement

This work has been supported by the R&D labs at IN-D by Emulya Technologies Pvt. Ltd.. The OCR engines used are Google vision API and Pytesseract OCR. The author would also like to thank the open source community for providing various libraries to complete this project.

References

[1] An overview of the Tesseract OCR engine, Ray Smith, Google Inc.

[2] 9Sphere AI, Machine Vision Recipes: Deskewing document images, Medium Aug 23 2020 https://medium.com/@9sphere/machine-vision-recipes-deskewing-document-images-e17827894c34

[3] Text skew correction with OpenCV and Python, Pyimagesearch Feb 2017 https://pyimagesearch.com/2017/02/20/text-skew-correction-opencv-python/

[4] OpenCV python https://opencv.org/

[5] Numpy https://numpy.org/

[6] Pandas https://pandas.pydata.org/

[7] Projection profile method, Sep 16 2019 https://stackoverflow.com/questions/57964634/python-opencv-skew-correction-for-ocr

[8] Mask R-CNN Kaiming He et.al., Mar 20, 2017 https://arxiv.org/abs/1703.06870