Document Scanner and Perspective Correction

Members:

Name	Enrolment No.
Jainik Patel	AU2340042
Dhruvik Shah	AU2340113
Ayush Makwana	AU2340138
Divyam Thakkar	AU2340227

Abstract— **This aim of this project is to transform a photograph of a document into a document that looks like it was scanned by a hardware scanner and correcting the perspective from which the image will be viewed. For this the project uses methods like edge detection, perspective transformation and approximation of contour. After going through these processes the output image of document closely resembles a scanned image.**

Keywords—Image processing, document scanning, perspective correction, edge detection.

I. INTRODUCTION

Paper documents are very frequently converted into images these days. To accomplish this traditional scanners can be used but they are very bulky and not portable. Hence it is simpler to use mobile camera instead. But for documents there can be many issues while clicking their pictures like shadows, photos not taken from correct perspective (photos taken from a slanted angle), uneven lighting, blurry output and shadows.

Hence the goal of this project is to use digital image processing methods to make a flow for basic document scanning using mobile cameras. To do this first the system would automatically recognize the border of the document in a whole picture with background. Then it adjusts the perspective and enhances visual quality. The final picture should be readable and usable.

To achieve these goals OPENCV library is used in python.Image capture, preprocessing, contour detection, perspective modification, and final enhancement are some of the phases that make up this process. To achieve the desired outcome all the phases are done stepwise.

II. METHODOLOGY

The OpenCV library is used to implement the project in Python. Image capture, preprocessing, contour detection, perspective modification, and final enhancement are some of the phases that make up this process. To get the desired outcome, each step is finished in order.

A. Image Acquisition

During the process of acquiring image, the replication of real-world situation is done by taking photos in different types of lighting, backgrounds and different angles for varying viewpoints.

B. Preprocessing

During preprocessing, each image is converted into grayscale to reduce the complexity of image. After this process the grayscale image is smoothened with the help of gaussian blur as it helps reduce the noise. This prepares the image of the document for edge detection and contour analysis. Edge detection is done using canny edge which highlights all the boundaries of the document. If the lighting is not suitable then adaptive thresholding can be applied as well. Hence after this stage document edges are clearly visible.

C. Contour detection

Once the edges are identified clearly it is time to find the boundary of the document and the shape of the document. This is performed by looking at the contours (outlines or boundaries) of the document given in image.

D. Perspective transformation

After contour detection, edge points of the document are identified. The new transformation matrix is calculated. This newly made transformation matrix is then applied to produce a top-down perpendicular view of the document. This step helps the model correct the perspective distortion cause by taking photo at a bad angle.

E. Image enhancement

Finally, the transformed needs to be dealt with contrast and brightness. This phase deals with contrast and brightness and makes the text inside the document more visible and the background clearer. Adaptive thresholding can also be applied in some cases to make the image a clean black and white version. The output is resized as required. The resulting image closely resembles a hardware scanned image making it suitable for reading.

III. RESULTS

The project till now has completed image acquisition and preprocessing. These portions mainly focus on loading the camera captured image, converting it into grayscale, reduce noise and highlight edges of the document using edge detection.

Fig. 1. Input image

1. Grayscale Conversion: The original image was converted to grayscale to simplify the processing and focus mostly on the variations of intensity, reducing unnecessary colored data.

Fig. 2. Grayscale image

2. Noise Reduction: A Gaussian blur filter is applied to smoothen the image and reduce background noise, which enables it for clearer edge detection.

Fig. 3. Image after smoothening

3. Edge Detection: Canny edge detection was used to show the boundaries of document. This stage effectively separated the document from its background, preparing it for contour detection which is the next phase.

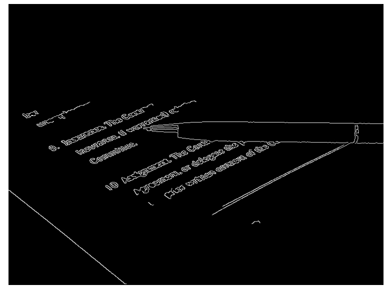

Fig. 4. Image with edges detected

Challenges Faced

During the implementation, these were the challenges faced:

- Due to shadows and uneven lighting partial edge breaks were detected during the process.
- Slight tilting of the captures required multiple adjustments of threshold values.
- Noise reduction parameters had to be tuned to avoid losing fine document edges.

IV. Discussion

The edge detection and preprocessing phases have been successfully coded and also tested on several document images under different circumstances. The outcome proves that the applied pipeline significantly manages well-lit and regular circumstances with a roughly 70% success ratio in spotting outlines on different luminosity scenarios.

The grayscale conversion was able to reduce the image data effectively while retaining necessary structural information for edge detection. The Gaussian blur with a 5×5 kernel was able to suppress noise effectively without blurring the document sides significantly, a requirement for proper boundary detection. The Canny edge detection heuristic, with 75 and 200 as the threshold values, was able to detect document boundaries effectively in the majority of the test cases.

Nevertheless, a number of problems were faced during implementation. Poorly illuminated poor-contrast images led to insufficient edge detection with partial edge breaks. This is due to the challenge in separating the document boundaries from similarly textured backgrounds based on poor lighting. The adaptive thresholding process can effectively overcome such problems, given that this process can cope with different levels of illuminations in distinct image regions.

The minute tilt detection in sampled images needed direction-wise fine-adjustments of the thresholding parameters. Higher values in the thresholds led to edge skipping, whereas lower values led to unnecessary noise inclusion. A series of iterations was required to achieve a balanced outcome. Also, the noise removal parameters had to be fine-adjusted lest thin document outlines were lost while background noise was successfully eradicated.

In the future, the following stages of contour detection and perspective transformation will be integral. The contour detection will have to clearly locate the four corners of the paper based on the found edges even if the edges are broken partially. The computation for the perspective transformation matrix will then utilize the corner points to create a top-down orthographic view of the paper.

For the ultimate enhancement phase, brightness and contrast will be applied to enhance readability of the test pages. Adaptive thresholding will be attempted to create nice black-and-white versions for scanning that are suitable for document archiving and optical character recognition purposes.

V. CONCLUSIONS

This work properly proves the possibility of converting document image captured by a smartphone into scanner-like image with the help of digital image processing methods. The first stages of image acquisition and preprocessing have been accomplished that have built a stable basis for the rest implementation stages.

Preprocessing pipeline, namely, grayscale conversion, blurred Gaussian effect, and Canny detection for edges successfully prepares an image for latter contour detection and also perspective correction. Under regular and properly illuminated circumstances, the system works well, though work remains for dealing with contrast-low and dark scenarios.

Chief learnings during this stage are that parameter tuning is crucial for varying lighting scenarios and that robust edge detection methods are required that are capable of dealing with real-world image variations. The issues faced, especially with edge breaks under poor or no lighting, are a good pointer for developing adaptive solutions for advancing this work further.

The future installation of contour detection, perspective transformation, and image enhancement will round off the scanning pipeline document. When fully installed, this will supply a functional substitute for current equipment scanners, utilizing the portability and handiness that is found in smartphone cameras.

Future work can investigate machine learning methods for self-adjustment of parameters according to image properties, enhancing the robustness of the system under various capture settings. Moreover, integration of real-time processing facility can extend user experience by obtaining immediate feedback during document capture.

REFERENCES

[1] J. Canny, “A Computational Approach to Edge Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986.

[2] S. Suzuki and K. Abe, “Topological Structural Analysis of Digitized Binary Images by Border Following,” Computer Vision, Graphics, and Image Processing, vol. 30, no. 1, pp. 32-46, 1985.

[3] X. Li, B. Zhang, P. C. Yuen, R. W. H. Lau, “Smartphone-based Document Image Rectification,” IEEE Access, vol. 7, pp. 127654-127666, 2019.

[4] OpenCV Documentation, “OpenCV: Open Source Computer Vision Library,” [Online]. Available: https://docs.opencv.org/. [Accessed: Oct. 10, 2025].

[5] “Document Scanner using OpenCV,” YouTube Video Tutorial, [Online]. Available: https://youtu.be/q-gWjCgigTg. [Accessed: Oct. 10, 2025].