

Assignment 1 Report

Süleyman Yolcu
2210765016

1 Introduction

Document image rectification is the process of transforming a photographed or distorted document into a flat, fronto-parallel view as if it were scanned. This is important for improving readability and enabling accurate optical character recognition (OCR) on documents captured by camera. In this assignment, we implement a pipeline that automatically detects the boundaries of a document in an image and corrects for geometric distortions (such as perspective skew or page curvature) to produce a flattened output. The approach leverages classical computer vision techniques: first detecting edges in the image, then using the Hough Transform and RANSAC to identify the document's outline, and finally applying a perspective transformation (homography) to warp the image into a rectangular, flat view. We evaluate the quality of rectification by comparing the output image against a ground-truth flattened image using the Structural Similarity Index (SSIM), which provides a quantitative measure of image similarity. Six categories of test images (curved, fold, incomplete, perspective, random, rotate) covering various distortions are used to assess performance. In this report, we detail the methodology and discuss the results for each category, highlighting strengths, challenges, and example outcomes.

2 Methodology

2.1 Preprocessing and Edge Detection

The input color document image is first preprocessed to simplify feature detection. Preprocessing involves converting the image to grayscale and reducing noise that might interfere with subsequent edge detection. In our experiments, a Gaussian blur with a 17×17 kernel was found to be the most effective; this strong blur smooths the image sufficiently so that the Canny edge detector, using threshold values of 30 and 90, reliably finds the edges of the paper without being distracted by minor textures or noise. After smoothing, we apply the Canny edge detection algorithm to produce a binary edge map. This edge map, which captures the most prominent intensity gradients, serves as the foundation for detecting the straight lines corresponding to the document's borders.

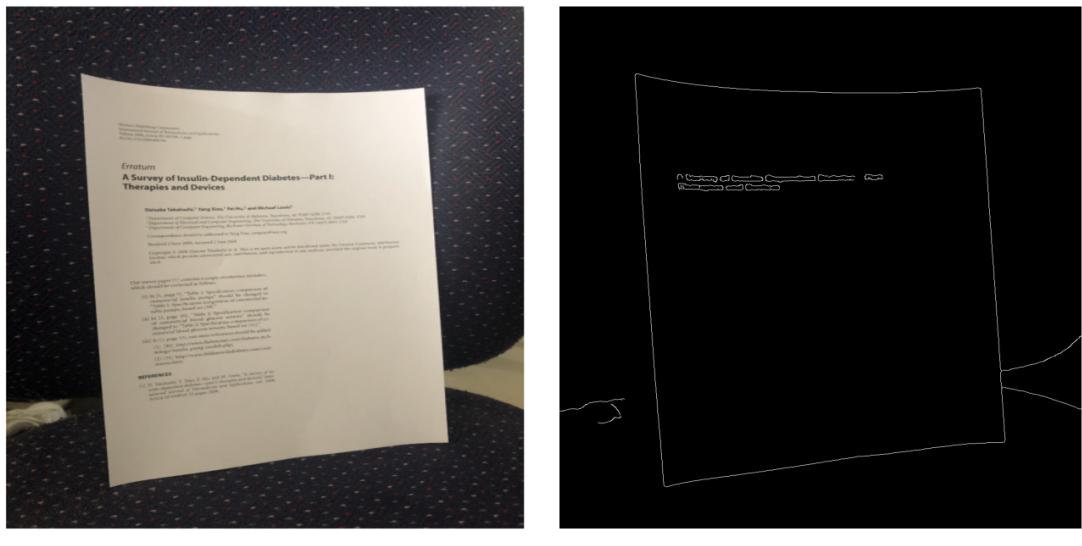


Figure 1: Preprocessing and Edge Detection

2.2 Line Detection with Hough Transform

Given the edge map, the next step is to detect straight lines that could form the borders of the document. We use the Hough Line Transform, a voting-based technique for line detection. The Hough Transform works by transforming edge points into a parameter space (defined by line angle θ and distance ρ from the origin) and accumulating votes for all possible lines that pass through each edge point. Peaks in the Hough accumulator space correspond to lines where many edge points align. By setting a vote threshold, we extract the most significant lines in the image. The reason for using the Hough Transform is its robustness in finding globally significant lines even if the edges are not perfectly continuous; it can bridge gaps in the edge and tolerate some noise. In the context of document images, the Hough Transform should ideally detect up to four dominant lines outlining the document (top, bottom, left, right edges of the page). It often produces multiple lines (including duplicates or spurious lines from text or background), but among these the true page borders should have high votes due to their length and continuity. The output of this stage is a set of line equations or segments identified in the image (e.g., represented by their ρ and θ parameters or end-point coordinates).

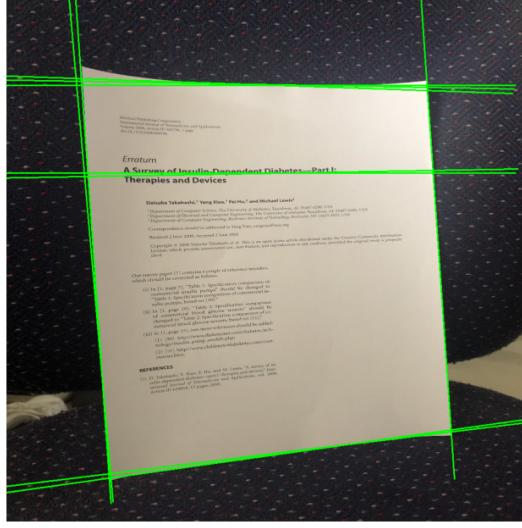


Figure 2: Hough Transform

2.3 Robust Line Refinement with RANSAC

To refine the line detection and ensure we identify the correct document outline, the pipeline employs RANSAC (Random Sample Consensus) in conjunction with the Hough Transform. RANSAC is a robust model fitting algorithm that iteratively selects random subsets of data points to hypothesize a model and then checks how many points from the full dataset agree with that model (considered inliers). In this assignment, RANSAC is used to improve line detection results by filtering out outliers and spurious edges. For example, given a cluster of edge points along what should be a straight border, RANSAC can fit a line through random pairs of those points and quickly converge on the best-fitting line by maximizing inliers (edge points that lie close to that line). This helps in cases where the edge might be partially occluded or not perfectly straight. By using RANSAC, we ensure that each detected border line is supported by a consensus of edge pixels, reducing the chance that a stray line (like a fold line or a margin line) is mistaken for the true document boundary. In our implementation, we set the number of iterations to 100 and used a distance threshold of 2 pixels—parameters that were determined to be the most effective for this task. These settings allow the algorithm to robustly filter out outliers and converge on the best-fitting line for each document edge. In practice, the algorithm can iterate to find multiple lines: find one dominant line via RANSAC (maximizing inliers), remove those inliers, then find the next line, and so on, until four candidate lines are obtained. Using Hough Transform and RANSAC together takes advantage of both global voting and random sampling: Hough provides a list of probable line candidates, and RANSAC fine-tunes the selection and parameters of those lines in a robust way. This two-step approach increases reliability in detecting the correct outline, especially under challenging conditions (e.g., uneven lighting or clutter, where Hough might detect extra lines, or when edges are slightly curved, where RANSAC can still approximate a straight line through the scattered edge points).

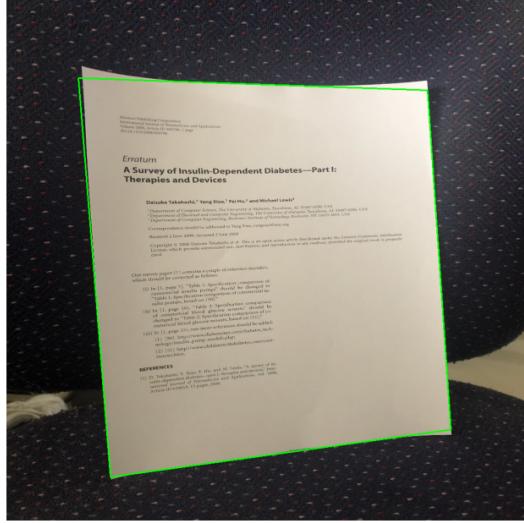


Figure 3: RANSAC

2.4 Corner Detection and Document Boundary Identification

Once we have a set of candidate lines (ideally the four sides of the document), we determine the corner points of the document by computing the intersections of these lines. Each pair of adjacent border lines (e.g., top with left, left with bottom, bottom with right, and right with top) should intersect at a point corresponding to a document corner. Because Hough Transform can sometimes detect multiple nearly overlapping lines for one edge, or lines that extend beyond the true intersection, we take care to choose the correct line pair intersections. Techniques such as filtering line intersections by angle can be used: the document corners should form roughly 90° angles for a rectangular page, so we can discard intersection points where the lines meet at very acute or obtuse angles not characteristic of a rectangle. We also prioritize combinations of four corners that form a convex quadrilateral with large area, since the document is likely the largest rectangular object in the scene. The four corner points are then ordered (for example, by sorting by $x+y$ coordinate sums to identify top-left vs bottom-right, etc.) to consistently label them as top-left, top-right, bottom-right, bottom-left. This ordering is important for the next step to correctly map corners to the output image's corners.

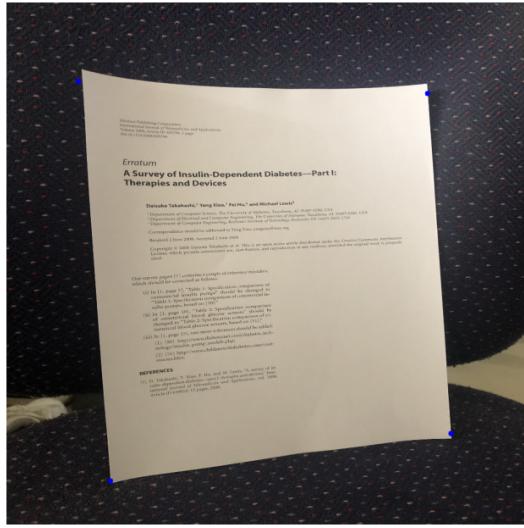


Figure 4: Detected Corners

2.5 Perspective Transformation (Homography Warp)

With the four document corner points identified in the input image, we can compute a perspective transform (homography) to "unwarp" the document. A perspective transformation is a geometric operation that maps points from one plane to another, correcting for projective distortion. Using the corner correspondences, we define a mapping from the quadrilateral in the input image to a perfect rectangle representing the output document. In our assignment these geometric transformations are used: **translation, rotation, scaling and projective warping**. For instance, if the detected corners in the original image are (x_i, y_i) for $i = 1..4$, and we want the output to be a straight rectangle of width W and height H , we assign those input corners to target coordinates $(0, 0), (W, 0), (W, H), (0, H)$ (assuming a desired output size that matches the document's aspect ratio). A 3×3 homography matrix is then computed that transforms each input corner to the corresponding output corner. We apply this homography to the entire image using warping, which moves every pixel to a new location such that the document appears frontal and rectangular in the result. This step corrects the perspective skew and any rotation — the document in the output image is now aligned with the image axes, as if scanned. It's worth noting that this perspective transformation assumes the document is planar. It will perfectly rectify affine and projective distortions caused by camera angle or in-plane rotation. However, if the page was curved or wavy (non-planar), a single homography cannot fully remove all distortions (it cannot undo true 3D warping like page curvature, though it might still flatten the overall page outline). Despite that limitation, perspective warping will generally improve the appearance of the document, making it more rectangular and easier to read.

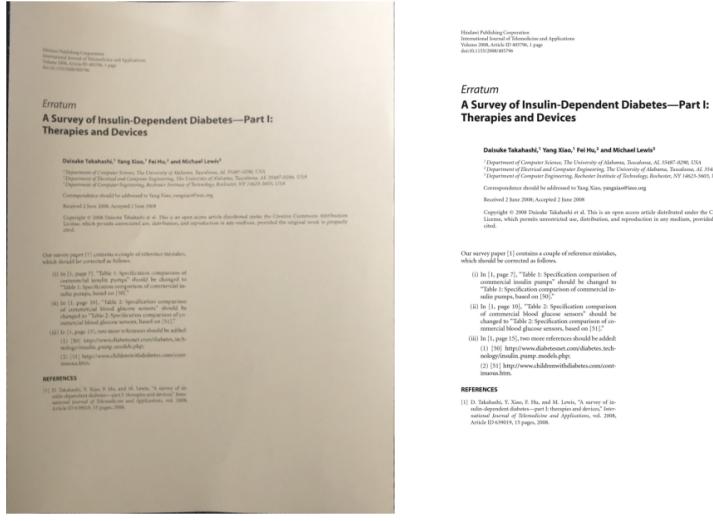


Figure 5: Final Output and Ground Truth Image

2.6 Quality Evaluation with SSIM

After obtaining the rectified document image via perspective transform, the final step is to evaluate how close this result is to an ideal "ground truth" flat document. The ground truth in this context is typically a perfectly scanned image or a manually corrected version of the same document for reference. We use the Structural Similarity Index (SSIM) to quantify the similarity between the rectified output and the ground truth image. SSIM is an image quality metric that compares two images in terms of luminance, contrast, and structural information on a scale from -1 to 1 (where 1 indicates two images are identical). In practice, SSIM values are usually reported between 0 and 1 for non-negative image data, with higher values indicating greater similarity. Unlike a simple pixel-wise error (such as mean squared error or peak signal-to-noise ratio), SSIM is designed to be more perceptually meaningful, focusing on the preservation of image structure (for example, the continuity of text lines and edges in the document). In our evaluation, we compute the SSIM for each rectified image against its ground truth counterpart. To ensure robust evaluation across diverse conditions, 50 images from each folder were used to compute the average SSIM score for each distortion category. This large sample size helps in obtaining a statistically meaningful assessment of the algorithm's performance. A higher SSIM means the algorithm successfully recovered the document with minimal distortions or quality loss (the text and layout match the reference closely), whereas a low SSIM means the output still differs significantly (due to remaining distortions, blurring, cropping errors, or other artifacts). We report average SSIM values for each category of distortion in the test set and also note the best and worst performing cases in each category to analyze where the algorithm succeeds or fails.

3 Results

After applying the above rectification pipeline to a set of document images, we organized the results into six categories based on the type of distortion or challenge each image presented: curved, fold, incomplete, perspective, random, and rotate. For each category, we calculated the average SSIM across all images in that group, as well as identified specific examples of best-case and worst-case outputs (with their SSIM scores) to illustrate the range of performance. Below is a summary of the performance in each category:

Table 1: SSIM Scores

Category	Average SSIM score	Best SSIM Score	Worst SSIM Score
curved	0.4760	0.8805	0.1318
fold	0.5154	0.7521	0.1031
incomplete	0.4795	0.7130	0.0724
perspective	0.5491	0.8050	0.2053
random	0.4829	0.7840	0.0441
rotate	0.4505	0.8097	0.0121

A comparative analysis of these scores shows that the perspective category achieved the highest average SSIM, suggesting that the pipeline performs best when correcting pure perspective distortions of a flat document. In contrast, the rotate and curved categories show lower scores, indicating that extreme rotations or non-planar (curved) distortions pose greater challenges.

3.1 Curved

Average SSIM = 0.4759. This category contains images where the document page is curved (such as pages from a book that were not flat when photographed). The average SSIM is relatively low, indicating that on average the rectified outputs only partially resemble the ground truth. However, there are notable successes: the best result is for image **0002** with SSIM **0.8804**, and another good result is image **0037** with SSIM **0.7644**. In these best cases, the algorithm managed to flatten the page to a great extent, yielding an output very close to the reference.

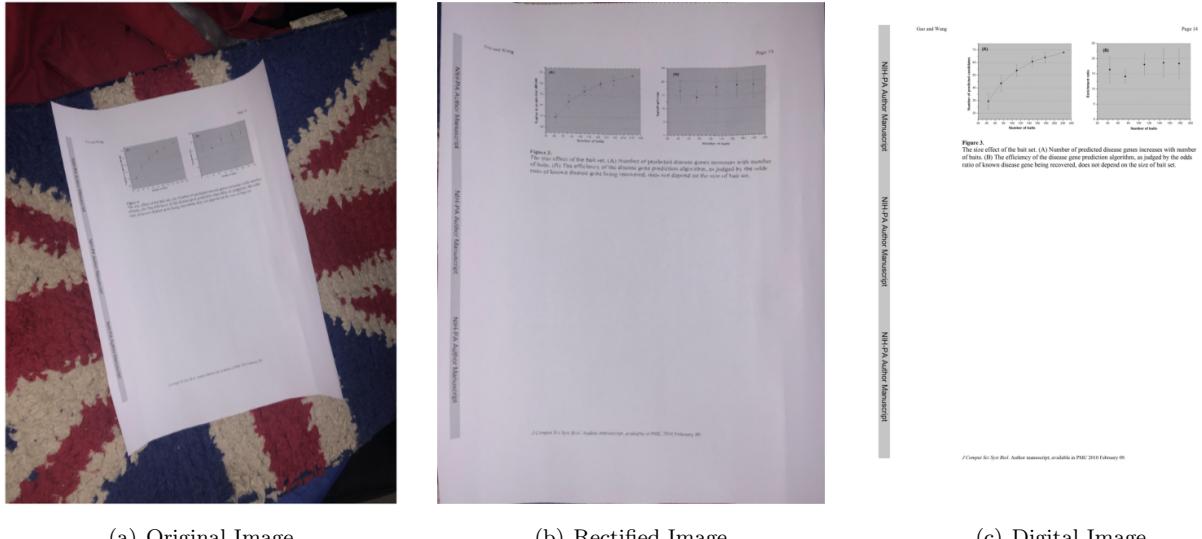


Figure 6: Curved Category Example (SSIM 0.88)

By contrast, the worst outputs in this category come from image **0024** (SSIM **0.1318**) and image **0005** (SSIM **0.1454**). These very low SSIM values mean the rectification largely failed – the output image still has strong distortions or deviates from the true layout.

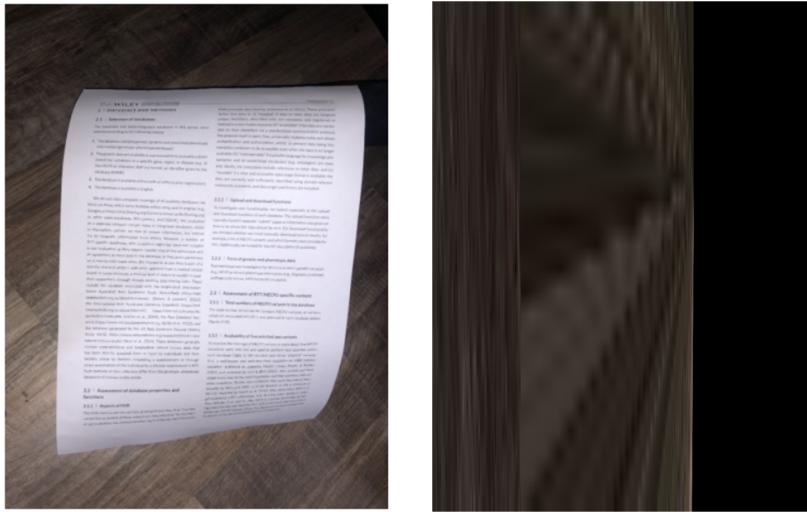


Figure 7: Curved Category Example (SSIM 0.14)

In the worst cases, the page curvature likely caused the border detection to be inaccurate, and since a single perspective warp cannot fully remove warping of a bent page, the resulting image remains distorted. The contrast between the best and worst in *curved* suggests that the pipeline can handle mild curvature (approximating it as a flat quadrilateral) but struggles with severe curves where the assumption of planarity breaks down.

3.2 Fold

Average SSIM = 0.5154. Images in the *fold* category have documents with folds or creases, meaning the page might have sharp bent lines or partially occluded content along the fold. The average SSIM is slightly above that of curved pages, indicating moderate success overall. The best-performing outputs were for image **0005** (SSIM **0.7521**) and image **0028** (SSIM **0.7453**), which show that despite folds, the algorithm could identify the correct page outline and flatten it reasonably well in some instances.

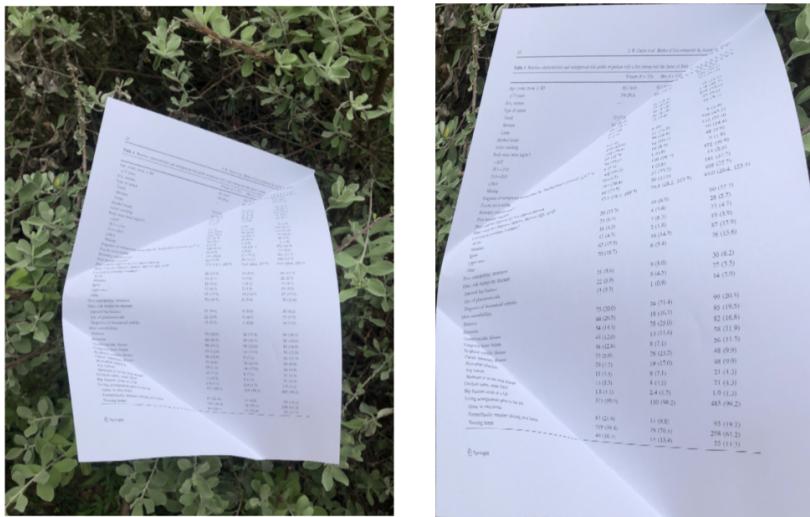
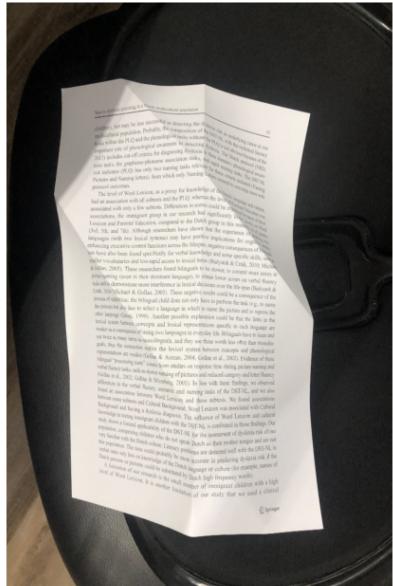


Figure 8: Fold Category Example (SSIM 0.75)

On the other hand, the worst cases in *fold* are image **0034** (SSIM **0.1031**) and image **0048** (SSIM **0.1572**). Such low SSIM values suggest that the presence of folds might have misled the pipeline.

Table 1		Baseline characteristics and outcomes risk profile of patients with a first nonspastic, high-grade stroke		
	Women (n = 77)	Men (n = 112)	Total (N = 189)	
Age, years, mean \pm SD	82.0 \pm 6.6	83.5 \pm 7.0	83.2 \pm 6.7	
< 75 years	339 (96.0)	406 (90.9)	439 (96.1)	
Sex	77 (100.0)	112 (100.0)	189 (100.0)	
Type of stroke				
Ischaemic	67 (87.0)	26 (23.2)	93 (48.8)	
Haemorrhage	10 (13.0)	73 (76.8)	83 (41.2)	
Large	20 (26.3)	49 (43.4)	69 (36.8)	
Small	24 (31.5)	34 (30.4)	58 (31.0)	
Other	23 (30.2)	10 (9.1)	33 (17.2)	
Atrial stroke	7 (9.1)	11 (9.8)	18 (9.5)	
Body mass index (kg/m^2)				
< 18.5	8 (10.5)	9 (8.2)	17 (9.0)	
18.5–25.0	99 (42.4)	51 (45.5)	210 (48.1)	
25.1–30.0	113 (45.3)	54 (46.4)	167 (35.9)	
> 30.0	56 (35.5)	12 (10.7)	70 (14.6)	
Unknown	1 (1.3)	1 (0.9)	2 (1.1)	
Diagnosis of entrepreneurship established by domain score (F -score \leq 5)				
Free from available	8 (10.5)	1 (0.9)	9 (4.8)	
Incomplete	92 (96.0)	118 (98.2)	210 (96.0)	
Unknown	2 (2.5)	0 (0.0)	2 (1.0)	
Prior year for stroke				
No	544 (94.0)	37 (33.3)	581 (98.3)	
Yes	28 (5.0)	8 (7.2)	36 (6.7)	
Time since first stroke, months, median (Q1–Q3)	84 (21.3) (10.8)	78 (29.3) (26.3)	84 (20.4) (12.8)	
Location of previous frequent stroke				
Head	90 (53.3)	10 (8.9)	60 (31.3)	
Shoulder	24 (16.4)	4 (3.6)	28 (15.7)	
Arm	10 (6.8)	1 (0.9)	11 (6.0)	
Upper arm	17 (4.8)	2 (1.8)	19 (5.9)	
Elbow	47 (17.6)	16 (14.3)	67 (37.0)	
Wrist or hand	70 (25.3)	20 (17.8)	90 (48.0)	
Other risk factors for stroke				
Previous stroke	21 (5.6)	9 (4.6)	30 (8.0)	
Use of risk-modifiers	22 (5.9)	5 (4.5)	27 (5.5)	
Diagnosis of cardiovascular disease	13 (3.5)	1 (0.9)	14 (2.9)	
Habits:				
Tobacco	75 (20.0)	24 (21.4)	99 (26.5)	
Alcohol	44 (11.5)	10 (9.0)	54 (14.3)	
Cocaine/heroin	54 (14.0)	28 (25.0)	82 (18.4)	
Congestive heart failure	45 (12.0)	13 (11.6)	58 (13.0)	
Diabetes mellitus	40 (10.5)	10 (9.0)	50 (13.3)	
Chronic pulmonary disease	22 (5.9)	26 (23.2)	48 (9.9)	
Hyperlipidemia	29 (7.7)	10 (9.0)	39 (8.5)	
Any tumor	13 (3.5)	8 (7.1)	21 (6.3)	
Mild or severe renal disease	13 (3.5)	8 (7.1)	21 (4.3)	
Hypertension	134 (34.5)	101 (90.5)	135 (71.4)	
Hypertension rate of a fall				
Stroke rate per 100	37 (39.9)	106 (98.2)	483 (99.2)	
Alone is one				
Yes	82 (21.5)	11 (9.8)	93 (19.1)	
Particularly someone sharing own home	219 (59.0)	79 (70.5)	268 (62.6)	



(a) Original Image



(b) Rectified Image

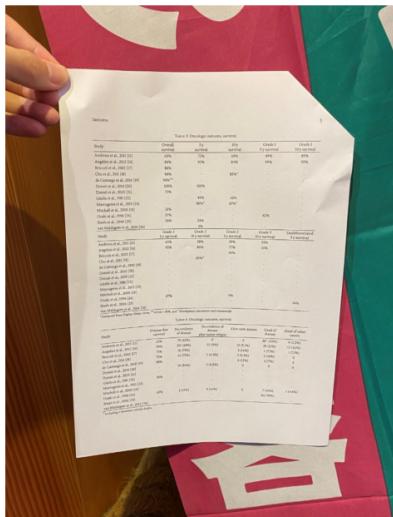


Figure 9: Fold Category Example (SSIM 0.10)

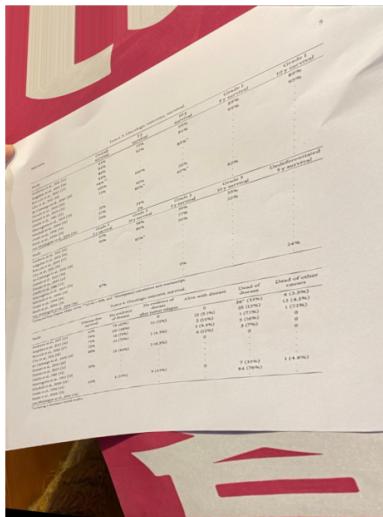
Overall, the presence of folds adds extraneous edges that make the Hough/RANSAC detection less reliable unless the algorithm correctly distinguishes the actual page boundary from the fold lines.

3.3 Incomplete

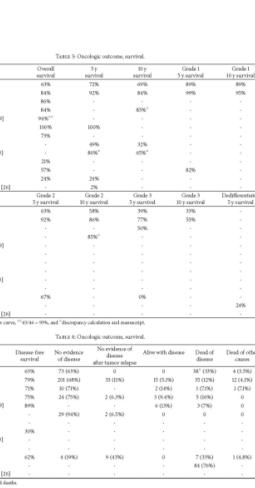
Average SSIM = 0.4785. The *incomplete* category refers to images where the document is not fully in frame or parts of its border are obscured, resulting in an incomplete outline. The pipeline finds these particularly challenging, as reflected by a fairly low average SSIM. When part of the page edge is missing, the algorithm has to extrapolate or might pick a wrong edge. Nevertheless, some images were handled decently: image **0001** achieved SSIM **0.7130** and image **0021** reached **0.6715**, indicating that even with missing sections, the algorithm sometimes inferred the document shape correctly and produced a reasonably good rectification.



(a) Original Image



(b) Rectified Image



(c) Digital Image

Figure 10: Incomplete Category Example (SSIM 0.67)

In these better cases, perhaps the majority of the page was visible and the detected lines could be extended to meet each other, correctly locating the corners. In contrast, the worst results (image **0025**, SSIM **0.0724** and image **0020**, SSIM **0.1216**) show scenarios where the approach failed to recover the document.

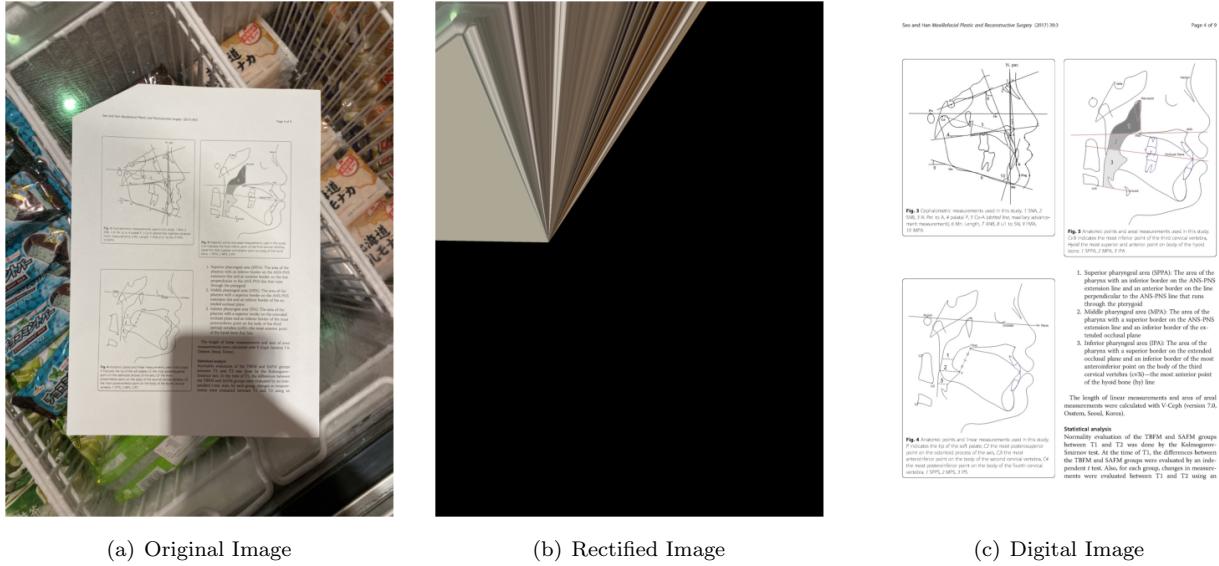


Figure 11: Incomplete Category Example (SSIM 0.12)

Possibly, a large portion of the border was missing or the algorithm locked onto incorrect features as a border. The output might then be a severely skewed or wrongly cropped image that does not match the ground truth.

3.4 Perspective

Average SSIM = 0.5499. This category includes images where the main distortion is perspective skew (the camera was at an angle relative to the document). Here the document is flat and fully visible, but appears trapezoidal due to perspective. This is the scenario the algorithm is fundamentally designed to handle, and accordingly, this category has the highest average SSIM of all. The best outputs are from image **0049** (SSIM **0.8050**) and image **0008** (SSIM **0.7545**), showing that in many cases of pure perspective distortion the pipeline successfully detects all four edges and applies an accurate homography to yield a nearly perfect front-view.

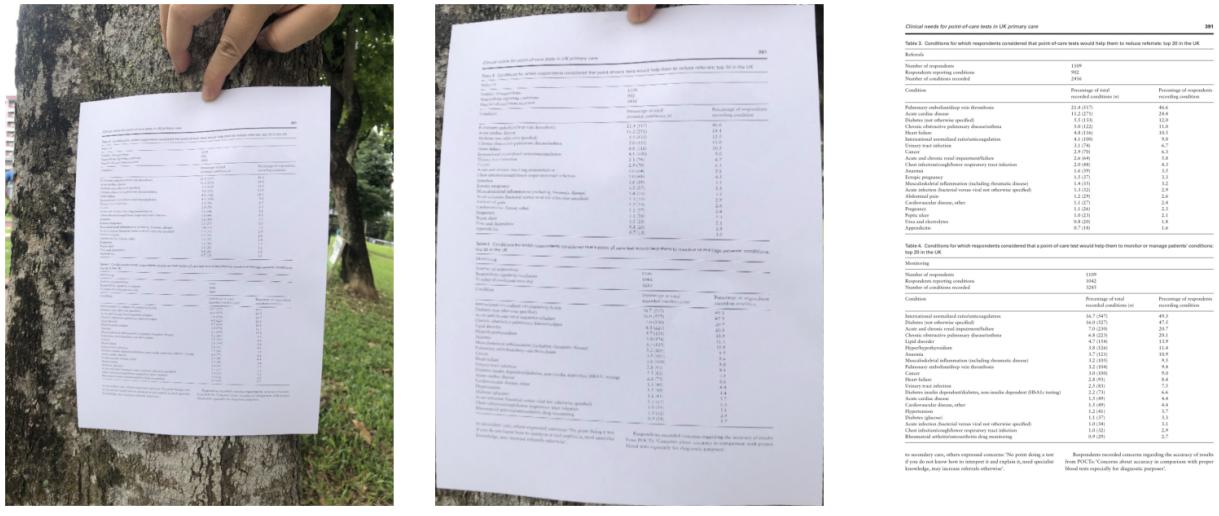


Figure 12: Perspective Category Example (SSIM 0.75)

The text and layout in these outputs align very well with the ground truth. The worst cases in *perspective* (image **0041**, SSIM **0.2031** and image **0025**, SSIM **0.2433**) are lower, but still not as extreme as some other categories.

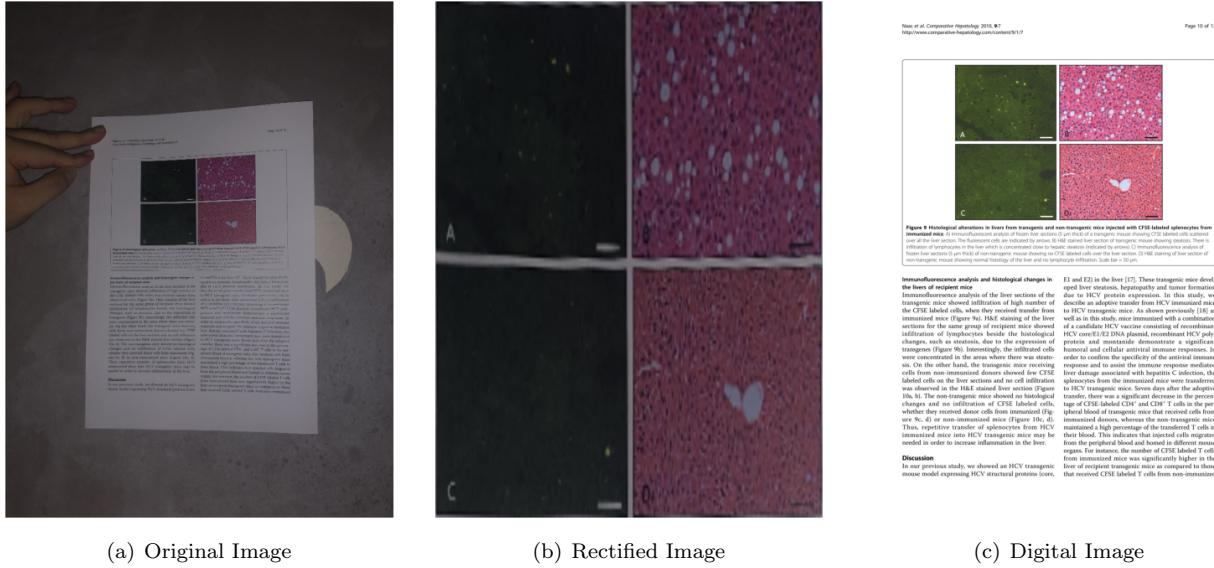


Figure 13: Perspective Category Example (SSIM 0.24)

A low SSIM in this straightforward case suggests the algorithm might have failed to find the correct corners due to issues such as unusual lighting or low contrast.

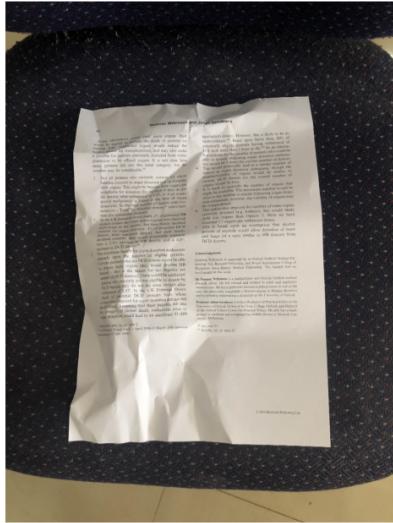
3.5 Random

Average SSIM = 0.4828. The *random* category consists of images with miscellaneous distortions or challenging conditions that do not fall into one specific type, possibly a combination of issues. The average performance is similar to the curved and incomplete categories, suggesting moderate difficulty. The best examples of rectification in this group are image **0024** (SSIM **0.7839**) and image **0049** (SSIM **0.7495**), where the algorithm managed to correct the document well despite various challenges.

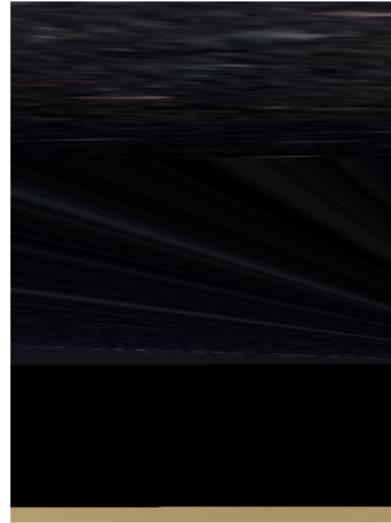


Figure 14: Random Category Example (SSIM 0.78)

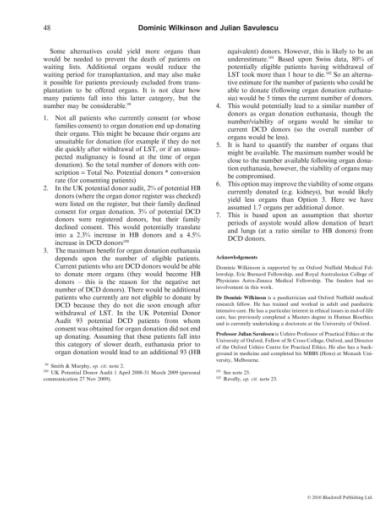
The worst cases (image **0041**, SSIM **0.0441** and image **0003**, SSIM **0.0789**) demonstrate failure modes where the document was likely not correctly identified.



(a) Original Image



(b) Rectified Image

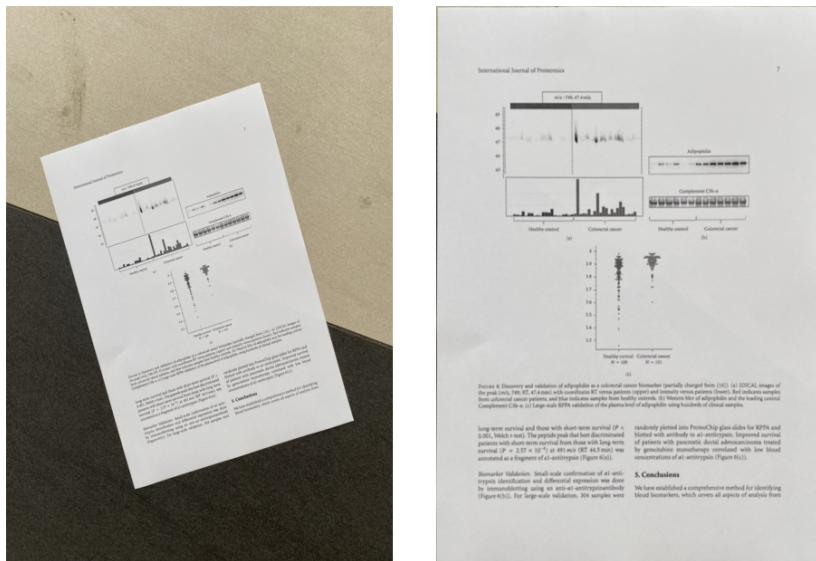


(c) Digital Image

Figure 15: Random Category Example (SSIM 0.07)

3.6 Rotate

Average SSIM = 0.4505. This category includes images where the document is primarily rotated in the plane (and possibly also slightly skewed). A rotated document still forms a proper rectangle, just not aligned with the camera's view. In theory, this should not pose a big problem for the algorithm — the edges are all there, just at different angles. Indeed, in some cases the pipeline works very well: the best result is image **0006** with SSIM **0.8098**, and another good one is image **0022** with SSIM **0.6858**.



(a) Original Image

(b) Rectified Image

(c) Digital Image

Figure 16: Rotate Category Example (SSIM 0.80)

However, the low average SSIM and extremely poor worst-case results indicate inconsistency. The worst outputs are from image **0001** (SSIM **0.0121**) and image **0043** (SSIM **0.0422**), which are practically failed outcomes.



Figure 17: Rotate Category Example (SSIM 0.04)

3.7 Discussion

Low scores typically indicate a failure in corner detection or incorrect homography mapping, resulting in a severely distorted output. Low scores also happen when the algorithm orders the corners incorrectly or detects the wrong quadrilateral. For example, if one corner of the document is hard to detect due to rotation, the pipeline might connect the wrong edges. This error could produce a result where the page is flipped or only a tiny, skewed portion is cropped as the “document.”

The challenges with rotated images suggest that while simple rotation is manageable, certain angles may confuse the corner-detection process. When the top and side edges look similar, the algorithm might swap them. This highlights the need for an extra check to ensure that the detected quadrilateral represents a plausible document in the correct orientation.

Across all distortion types, the average SSIM values range from roughly 0.45 to 0.55, showing a moderate similarity to the ground truth. The method performs best on standard perspective distortions, achieving an average SSIM of around 0.55. In contrast, the rotation category scores around 0.45, indicating that severe orientation issues lead to more failures. Other categories, such as curved, folded, incomplete, or random distortions, show similar moderate success, while the best-case outputs reach SSIM values between 0.75 and 0.88. In these successful cases, the rectified images are nearly identical to the original flat documents, with straight text lines and correct proportions.

4 Conclusion

This project explored a complete pipeline for automatic document image rectification using classic computer vision techniques. The process began with edge detection to identify candidate features, followed by the use of Hough Transform and RANSAC to robustly detect document borders. Finally, a perspective transformation was applied to produce a flat view of the document.

The approach successfully transformed many distorted images into a scan-like format. The best results, with SSIM scores up to 0.88, show that when the algorithm works, the rectified image is nearly indistinguishable from the original document. However, the overall average SSIM of around 0.5 indicates there is room for improvement.

Documents with curved pages, heavy folds, or confusing features remain particularly challenging. These issues often cause the algorithm to select incorrect edges, leading to a failed rectification. The pipeline’s performance is highly dependent on accurately detecting all four corners, and any mistake can lead to significant errors.

Future work could focus on improving performance by using non-linear dewarping algorithms for curved pages, incorporating additional clues from text layout or color segmentation, or applying machine learning to enhance corner detection. In summary, the project demonstrates the effectiveness of a classical computer vision approach for document rectification while also highlighting clear areas for further development.

Of note, Sherman’s screw design remained the “gold standard” in orthopedics until the introduction of the AO screw half a century later [5]. Revision and wear introduced by the use of cortical screws led to the popularity of bone screws [5]. In the 1940s, the Belgian surgeon Jean Guillet developed the “Guillet screw system,” further modified screw designs to apply torque, and introduced the first implants implementing the following three technical features [6]:

1. A change of the ratio from the exterior screw diameter to the interior screw diameter from 4:3 in industry and medicine to 3:2 in orthopedics.
2. A reduction of thread surface area to 1/3, based on the finding that the load-bearing force of bone screws have a lower pitch than conventional cortical screws.
3. A transition from classic industrial V-shaped thread design to bottom threads (Fig. 3), based on the calculated increased pull-off resistance of bottom threads.

Robert Danis’ pioneering work on internal fixation, including improved screw design and plate resection, was presented at the First International Congress of the ‘Deutsche Gesellschaft für Osteosyntheseforschung’ in 1958 in Bonn, Germany [7]. The work was a major achievement of the AO who set the global standardization of surgical principles and techniques, and the introduction of the AO design for orthopedic implants and instruments [5].

Principles of screw threads:

Some screw threads are designed to distribute initial contact and surface area, dissipate and distribute stress forces at the screw–bone interface, and transfer these forces to load [8]. The basic principle of screw thread geometry includes shear, face angle, pitch, pitch depth, and width. Thread pitch depth is the distance variable among the available orthopedic screws on the market for example, condition screws have an increased

thread depth compared to cortical screws, with the intent of increasing the surface area for improved screw purchase in lower quality bone [9]. Thread pitch refers to the distance between two consecutive threads along the face of the thread in a spiral or around 36° (Fig. 1a). The angle of the thread is called helix angle. Improved screw designs is reflected by the shortening of bending moment and the increasing of the shear force due to increased shear strength and improved unidirectional shear transfer [10–12]. However, orthopedic screws are typically not challenged by axial loading forces from torsion or bending moments, as the majority of orthopedic screws remain at a significant risk of failure when subjected to axial loading forces [13]. Thus, an attempt to address the physiological multiaxial loading conditions of bone screws has been made, which has been able to reduce the risk of implant failure; nevertheless, the use of axial loading force in orthopedic surgery relies on the benefit of a fixed-angle connection that does not rely on friction and compression forces because of the potential for slippage [14]. Some orthopedic screws have been shown to have their own set of shortcomings such as increased cost, increased complexity, increased weight, and increased cost, hence research continues towards the development of new screw designs to reduce the risks to failures at the bone-implant interface.

In essence, until present, the history of orthopedic screw designs, despite significant biomechanical shortcomings associated with high failure rates [13–15],

