 doi:10.20944/preprints202007.0686.v1

Research Article

CleanPage: Fast and Clean Document and Whiteboard Capture

Jane Courtney[1]

[1] Technological University Dublin, Ireland; jane.courtney@tudublin.ie

Abstract: The move from paper to online is not only necessary for remote working, it is also significantly more sustainable. This trend has seen a rising need for high-quality digitization of content from pages and whiteboards to sharable online material. But capturing this information is not always easy, nor are the results always satisfactory. Available scanning apps vary in their usability and do not always produce clean results, retaining surface imperfections from the page or whiteboard in their output images. CleanPage, a novel smartphone-based document and whiteboard scanning system, is presented. CleanPage requires one button-tap to capture, identify, crop and clean an image of a page or whiteboard. Unlike equivalent systems, no user intervention is required during processing and the result is a high-contrast, low-noise image with a clean homogenous background. Results are presented for a selection of scenarios showing the versatility of the design. CleanPage is compared with two market leader scanning apps using two testing approaches: real paper scans and ground-truth comparisons. These comparisons are achieved by a new testing methodology that allows scans to be compared to unscanned counterparts, by using synthesized images. Real paper scans are tested using image quality measures. An evaluation of standard image quality assessments is included in this work and a novel quality measure for scanned images is proposed and validated. The user experience for each scanning app is assessed, showing CleanPage to be fast and easier to use.

Keywords: document scanning; whiteboard capture; image enhancement; image alignment; image registration; image quality assessment, NR-IQAs

1. Introduction

In the worldwide COVID-19 pandemic 2020, many faced a sudden thrust into an online-only working environment, making paper document sharing and physical whiteboard usage impossible. Besides this move to an online-only environment, many documents still exist only in paper form and must be scanned to digitize them. Many users also enjoy the 'chalk-and-talk' appeal of whiteboards but are left with no record of their work on erasure. With the ubiquity of smartphones and a variety of scanning apps available, digitization is becoming a simpler task.

However, in most scanning apps, the output quality of the scans is highly dependent on the quality of the one image captured and on the performance of the user in the capture process. As the camera is handheld, issues such as motion, focus, distance, lighting variations and perspective distortion are to be expected and have significant impacts on the success of the scan. To overcome these, most scanning apps offer additional manual intervention steps, such as corner adjustment or lighting correction. This has an impact on the user experience and slows scanning time considerably. The results are also affected by the quality of the paper or whiteboard surface and by clutter in the background of the image. In all scanning apps tested, the background surface was not corrected, leading to paper or whiteboard imperfections appearing in the final output.

Document scanning now focuses on detection of documents in images [1–3] and content recognition [4,5] – typically Optical Character Recognition (OCR) for text – but the problem of high-quality capture has not been fully solved. Available apps such as Microsoft Office Lens, Google PhotoScan, Cam Scanner, Adobe Scan, Scanbot and others, produce high-quality results but are

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

sensitive to capture conditions, retain surface imperfections, require manual intervention and often have slow or difficult to use user interfaces.

For input image capture, many scanning approaches use image stitching or mosaicking [6,7] which requires moving the camera around the document or whiteboard in a controlled way [8]. Others use deep learning techniques to find the object in the image in order to limit the capture area to content [9,10]. Improvements on these methods often focus on estimating text direction [11], limiting their usability to text-only images.

In the area of whiteboard capture, surprisingly little work has been done on whiteboard image enhancement, despite poor performance of existing methods on dirty whiteboards [12,13]. Though early work in the area has developed into high quality apps, e.g. Zhang & He's work [14] which ultimately contributed to the development of Microsoft Office Lens, these solutions still suffer from the same dependency on high-quality input images and therefore on user performance [15]. A review of existing whiteboard enhancement methods can be found in Assefa et al [16].

When it comes to document scanning, document image enhancement is still an open field of research, with most methods focusing on document binarization [17–20]. The problem of document binarization, i.e. moving towards a binary or black-and-white-only image, is still unsolved and a comprehensive review of the challenges involved can be found in Sulaiman et al [21]. By 2015, the document scanning problem was still not fully solved and the SmartDoc competition [22] was established to encourage development in the area and to evaluate the performance of document imaging methods. In the 2017 version [23], it was recognized that the solution may lie in video-based scanning. However, it was assumed any solution would use stitching or mosaicking, and so the videos in the provided dataset were captured using the usual input method, i.e. by moving the camera around the document in a controlled way.

Some multi-image or video-based approaches to scanning have been offered for the problem. Zhukovsky et al [24] use a segment graph-based approach to find the document boundaries while Chen et al [25] use video as a guide to the user before static image capture. Jiang et al [26] use key image extraction and image stitching for capture and Retinex processing [27] for enhancement. Luqman et al [28] combine mosaicking with data from the phone's own accelerometer to improve results from the usual moving-camera input method.

To date, no solution has been found which retains a fast and simple user interface while addressing the limitations of existing scanning apps, namely: surface imperfections, background clutter and sensitivity to both capture conditions and user performance.

2. Materials and Methods

In CleanPage, the user interface (UI) is simplified to a single button. Instead of capturing a single image, it records a short sequence of images, similar to the operation of High Dynamic Range (HDR) photography. However, unlike HDR, this system includes an alignment phase to overcome shortcomings in user capture and an enhancement phase to remove background surface imperfections. The system works by calculating inter-image motion vectors and homography to align and stack the images before merging these to a single output. This is then processed to enhance contrast and remove any background clutter by cropping to content. The output is a high-contrast, low-noise image with an ideal homogenous background surface. Because the input is a set of images instead of a single image, the system is less reliant on user performance and high-quality results are consistent across a variety of scan types. CleanPage's design results in improved background homogeneity, improved content retention and a simple and fast UI.

2.1. CleanPage Design

This system was designed in Python 3.7 with OpenCV 4.3 and ported to an Android app developed in Kivy using Buildozer. Testing was done on a PC using MATLAB 2020b for image quality tests.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

2.1.1. Image Capture

In CleanPage, images are acquired as a short sequence with one button-tap by the user. The raw RGB data is split into its three color channels and each is handled separately during the merging phase. The original image is also converted to a grayscale image, I (the intensity or average of the three channels) for use in the alignment phase.

2.1.2. Alignment

Once the images are acquired, these are aligned before merging. Alignment is achieved by first extracting distinct features from each image. These features are then tracked to the next image by determining the direction and magnitude of motion from image to image. The motion vectors extracted allow the homography between images to be computed. This is used to warp the images to an aligned position.

To find distinct features in an image, it is important that the method is scale, rotation and illumination invariant as hand motion is expected through the image sequence. There are many feature descriptors available, ranging from SIFT-based keypoints (SIFT, SURF, KAZE, ORB, etc. [29]) to simple corner detectors [30]. Here, Shi-Tomasi features are used as they are quick to compute and suitable for tracking, while remaining robust to scale, rotation and illumination changes [31].

Shi-Tomasi features are determined from the autocorrelation matrix:

$$A = \sum_{\mathrm{i}} \sum_{j} g(i,j) \begin{pmatrix} I_X^2 & I_X I_Y \\ I_X I_Y & I_Y^2 \end{pmatrix} \tag{1}$$

$g(i,j)$ is a set of Gaussian weights defined over a neighborhood region and I_X and I_Y are the horizontal and vertical gradients respectively of the intensity image, I.

The feature quality measure is then the smallest eigenvalue of A:

$$F = \min(\lambda_1, \lambda_2) \tag{2}$$

F is the feature measure and λ_1 and λ_2 are the eigenvalues of A. The Shi-Tomasi algorithm also includes non-maximal suppression to avoid 'clumping' of features and allows other parameters to be set, such as the minimum distance between features and the maximum number of features.

As the motion between images is small, simple optical flow can be used to determine the motion vectors. Here Lucas-Kanade optical flow is used [32]. This is similar to the feature extraction method outlined above but uses the brightness constancy assumption to incorporate the time dimension. The brightness constancy assumption states that the appearance of a feature does not change significantly from image to image when the motion is small. This means that the motion can be determined by a pair of horizontal and vertical vectors, (u, v), such that:

$$I(x, y, t) \approx I(x + u, y + v, t + 1) \tag{3}$$

These motion vectors can then be extracted using the autocorrelation matrix from before:

$$\begin{pmatrix} u \\ v \end{pmatrix} = A^{-1} \begin{pmatrix} \sum_i \sum_j g(i,j) I_X I_T \\ \sum_i \sum_j g(i,j) I_Y I_T \end{pmatrix} \tag{4}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

I_T is the intensity gradient in the time direction (from image to image), while $g(i,j)$ represents the parameters of a 2D window of gaussian weights and I_X and I_Y are the horizontal and vertical gradients respectively of the intensity image, I, as before.

Using the calculated motion vectors, the location of corresponding features from each image can be found in the next image. This creates a set of (x_t, y_t) locations of the features in each image, where t denotes the image number.

As the camera is handheld, it moves slightly during capture. This creates a different image plane for each image in the sequence. In order to re-align the images, the set of corresponding feature locations are used to compute the 2D homography between the images:

$$\begin{pmatrix} x_t \\ y_t \\ 1 \end{pmatrix} = H \begin{pmatrix} x_{t+1} \\ y_{t+1} \\ 1 \end{pmatrix} \tag{5}$$

The transformation between images as a result of camera movement is not generally considered a homography. But in this case, the feature points correspond to the same planar surface (the page or whiteboard), so the relationship holds true. This homography is calculated using the robust estimation method, RANSAC (RAndom SAmple Consensus) [33]. Once the homography, H, has been estimated, it can be used to affine warp the entire image to align with a reference image, so that all pixels align with their corresponding pixel in each image.

2.1.3. Merging

After the images have been aligned, they are then merged into a single image, M. This is done on each color channel separately by calculating the median of each pixel's color value across the entire image sequence:

$$M(i,j) = \left(\tilde{R}(i,j), \tilde{G}(i,j), \tilde{B}(i,j)\right) \tag{6}$$

where $\tilde{R}$, $\tilde{G}$ & $\tilde{B}$ are the medians of the red, green and blue color channels of pixel (i,j) respectively across the image sequence. This method has the added advantage of minimizing any noise in the images as each pixel is averaged over a set of images of the same content.

2.1.4. Border Removal

As a result of the affine warping, some empty space is introduced near the borders of the image where no corresponding pixels exist. This is to be expected and appears in the aligned images as a black border. This removes some of the background clutter and provides a starting point for finding the background surface (paper or whiteboard).

In order to identify regions of background surface in the image, the intensity is first scaled to enhance the background surface color. To do this, the Yen threshold [34], y, is calculated and the image intensity is scaled from range $[0, y]$ to the full range $[0,255]$. This is then thresholded to extract regions of background surface. The resultant image is a binary mask, B, with the surface represented by a zero value.

To remove elements of background clutter introduced by the user's inaccurate capture, a search box is initiated at the edge of the black border produced by the alignment phase. Each side of this search box is contracted, and the sum of its pixels from the binary surface mask, B, is calculated at each contraction. When this sum reaches a minimum value, it is determined that the location is entirely on the background surface, and the image is then cropped to this point. The locations of each side are found by calculating the sum of binary pixels at each row and each column for every nth contraction:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

$$R(n) = \sum_{j=l}^{j=r} B(n,j) \; \forall \; n \; \in \{0 \dots h\}$$
$$C(n) = \sum_{i=t}^{i=b} B(i,n) \; \forall \; n \; \in \{0 \dots w\} \tag{7}$$

where $R(n)$ is the pixel sum at the nth row and $C(n)$ is the pixel sum at the nth column from the edge, $t, b, l \,\&\, r$ are the top, bottom, left and right locations, updated recursively, and $w \,\&\, h$ are the width and height of the image respectively. The minimum value of n that gives a minimum value for R is found to be the topmost surface row, t, while the maximum value is the bottommost surface row, b. Similarly, the leftmost and rightmost locations, $l \,\&\, r$, are found from the column sums.

$$\begin{aligned} t &= \min\left(\underset{n}{\operatorname{argmin}}(R)\right) \\ b &= \max\left(\underset{n}{\operatorname{argmin}}(R)\right) \\ l &= \min\left(\underset{n}{\operatorname{argmin}}(C)\right) \\ r &= \max\left(\underset{n}{\operatorname{argmin}}(C)\right) \end{aligned} \tag{8}$$

In other border removal designs, contours are generally used to find the edges of the surface. However, this requires the entire surface to be visible in the image and it must have a strong contrast with the background. By using the method outlined here, the system is not limited by these constraints.

2.1.5. Image Enhancement

In the final stage of the design, the image is enhanced using two improvements: a sharpening kernel to remove blurring in the content area and intensity scaling to improve contrast in the background surface.

The merging phase causes significant reduction in noise, but this can also cause some blurring, and so sharpening is applied after merging to correct this. To avoid reintroduction of noise, the sharpening is applied only in the region of content by using a dilated version of the binary mask, B, calculated previously in the border removal phase. The following sharpening kernel is used:

$$k = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix} \tag{9}$$

To enhance the color of the background surface, the Yen threshold [34], y, is recalculated and the image intensity is rescaled in this region. This recalculation is done to improve performance of the enhancement phase since the absence of background clutter will improve the yen threshold calculation.

The outputs of each stage of CleanPage's design can be seen in **Figure 1**, showing the process from the original images to the final cropped and enhanced image scan.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

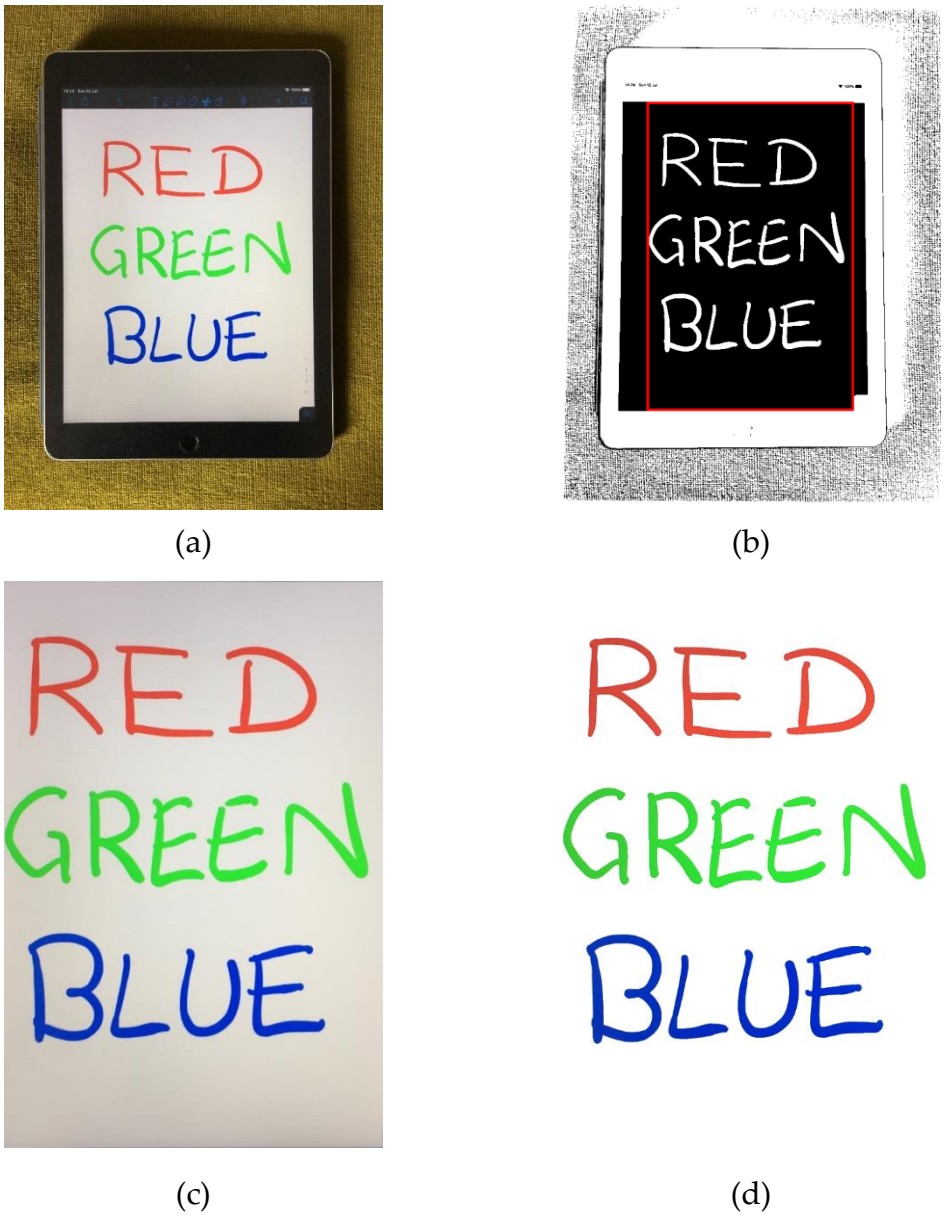

Figure 1. Stages of CleanPage. (a) a sample image from the original image sequence; (b) the binary surface mask showing the border search box; (c) the merged image after border removal; (d) after enhancement.

2.2. Data Inputs

To test the versatility and quality of the output, four scan styles were used: synthesized color test images, synthesized text test images, text on paper pages and content on crumpled paper. Samples of these can be seen in **Figure 2**. The synthesized test images (color and text) are high resolution, uncompressed, simple images and were displayed on a tablet and scanned with a smartphone. This novel testing method allows the original images to be used as ground truths. The real paper images were scanned in the normal way, using a smartphone. As these are on real paper, no reference will be available for comparison, so image quality measures were investigated for testing performance.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

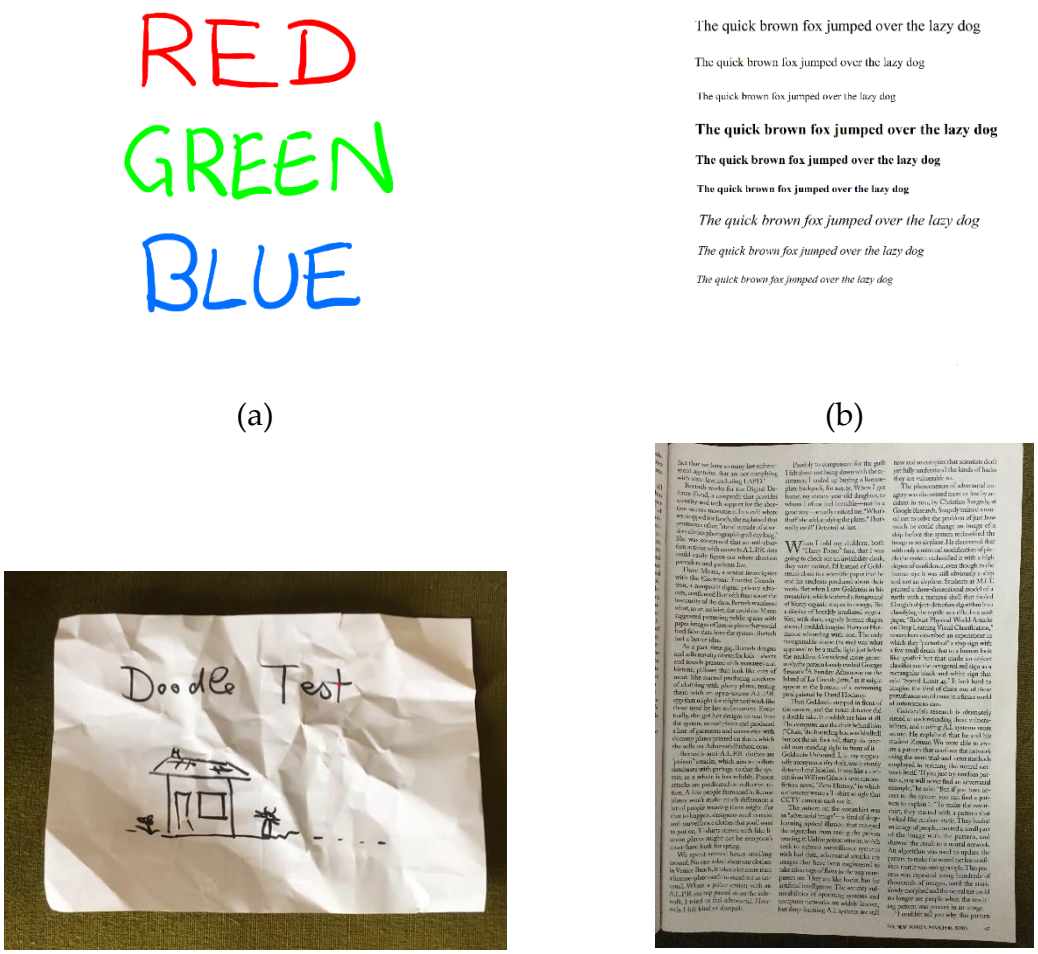

Figure 2. Sample original test images. (Note: real paper images are shown as photographs here). (a) a synthesized color reference image; (b) a synthesized text reference image; (c) a sketch on crumpled paper; (d) a sample of text on a page.

In all cases, images were scanned using CleanPage as well as two of the leading scanning apps available: Microsoft Office Lens and Google PhotoScan. Other scanning apps were trialed, but it was found that they fall into the two categories of design represented by these two apps: single shot designs (Microsoft Office Lens) and stitching approaches (Google PhotoScan). All images were scanned in the same lighting conditions and background environment and all were captured at the smartphone camera's maximum resolution (12 megapixels).

2.3. Image Quality Measures

For the synthesized test images, the originals are used as ground truths and a Structural Similarity score (SSIM) is calculated for each of the scanning methods compared with these images.

For the paper images, there is no ground truth for comparison, so a No-Reference Image Quality Assessment (NR-IQA) is needed. The most commonly used NR-IQAs were investigated: BRISQUE (Blind Referenceless Image Spatial Quality Evaluator [35]), NIQE (Natural Image Quality Evaluator [36]) and PIQE (psychovisually-based image quality evaluator [37]). However, in testing, it was found that these measures are not suitable for scanned images. Since they are designed to work with natural images, most NR-IQAs are in fact likely to favor the more natural, but less clean and accurate, photographic-style images over the ideal, homogenous background which is the goal of a high-contrast scan.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

Instead, for assessing the success of the scan, a measure of homogeneity is needed to test the consistency of the surface background. For this, the discrete entropy (DE) is used:

$$DE = \sum^{N} p_I \log_2 p_I \tag{10}$$

In a natural image, DE is seen as a measure of quality [38], as it represents the heterogeneity of the image. However, in a document or whiteboard image, it is expected that the background surface should be a large homogenous area (paper/whiteboard) so it is expected that in a high-quality image of this type, DE should be low.

This has been shown previously [39] where natural images and text images were compared under increasing distortion levels. The natural image DE decreases with distortion, while the text image DE increases significantly. Entropy has also been used previously to improve binarization methods for text recognition for this reason [40].

3. Results

3.1. Assessment of Image Quality Measures

To test potential NR-IQAs, the synthesized reference images were included in the assessment. As these are not scanned, they should represent higher quality and should give better scores in any image quality measure than scanned versions, regardless of the scanning app used. Standard NR-IQAs (BRISQUE, NIQE and PIQE) were assessed alongside the proposed DE measure. The standard NR-IQAs were shown to fail at assessing image quality in the context of scanning, as the scanned images show a better score than their unscanned counterparts (see **Table 1**). DE, on the other hand, consistently resulted in better scores for the unscanned ground truth images compared to the scanned versions, as predicted. This shows DE to be a valid and reliable measure of scanned-image quality.

Table 1. Comparison of image quality measures using original unscanned reference images as ground truths (Note: higher values denote lower quality). Highlighted in red is where the measure erroneously gives a worse value to the better image (the original unscanned version).

Scoring Metric	Synthesized Color Images		Synthesized Text Images	
	Scanned	Original	Scanned	Original
BRISQUE	45.1108	**44.773**	**41.7858**	46.322
NIQE	**4.2445**	11.8021	**3.6031**	14.68
PIQE	**68.7442**	90.9064	**45.8493**	81.4116
DE	1.5917	**0.5978**	5.8915	**0.2743**

3.2. Comparative Results

To assess the quality of output images from each scanning app (CleanPage, Microsoft Office Lens and Google PhotoScan), a variety of scan scenarios were tested. To allow comparisons to reference images, color and text images were synthesized and scanned. To test real world conditions, text on paper was scanned. Finally, to demonstrate the robustness to surface imperfections, scans were taken of content on crumpled paper. Sample results from each scan type, using each scanning app, are presented here.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

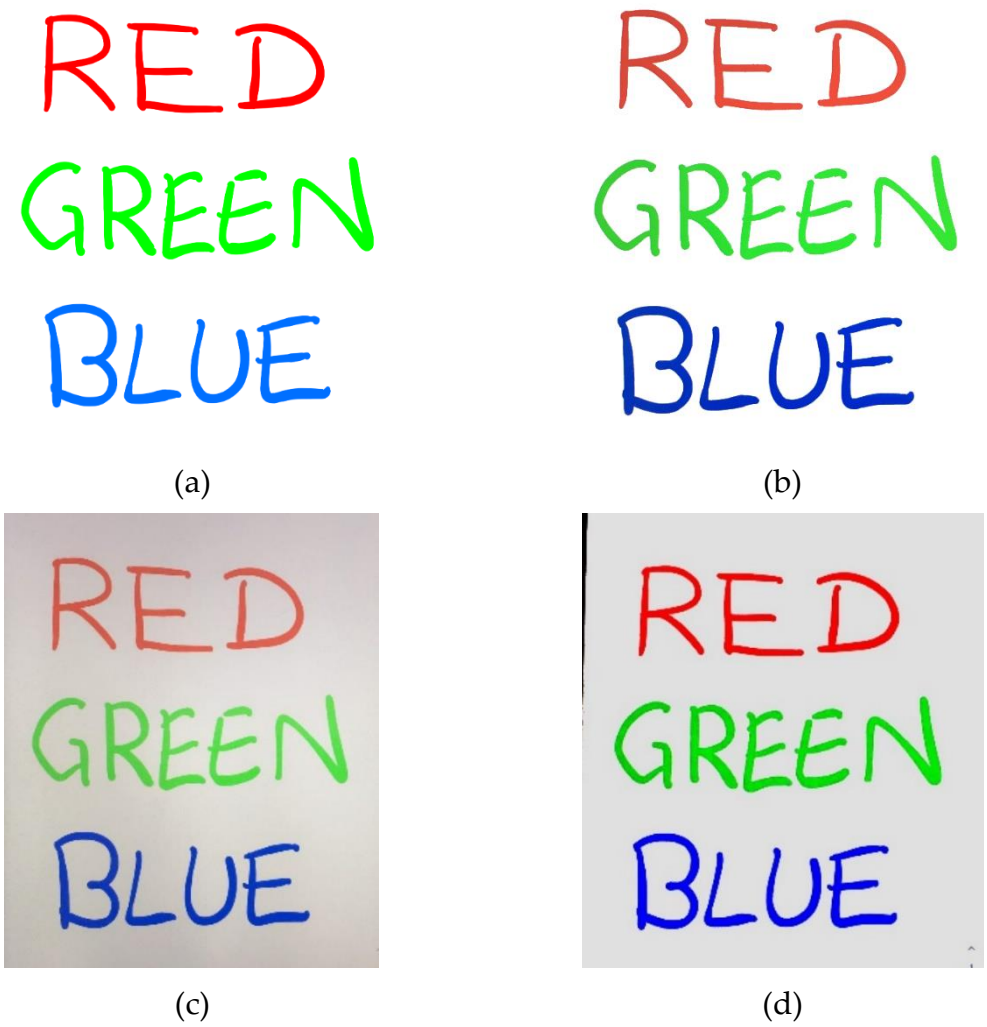

Figure 3. Sample results of a synthesized color image scanned using a smartphone with the image showing on a tablet screen. (a) Original Image; (b) CleanPage; (c) Google PhotoScan; (d) Microsoft Office Lens.

Figure 4. Zoomed results extracted from the smallest line of a synthesized text image, scanned using a smartphone with the image showing on a tablet screen. (a) Original Image; (b) CleanPage; (c) Google PhotoScan; (d) Microsoft Office Lens.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

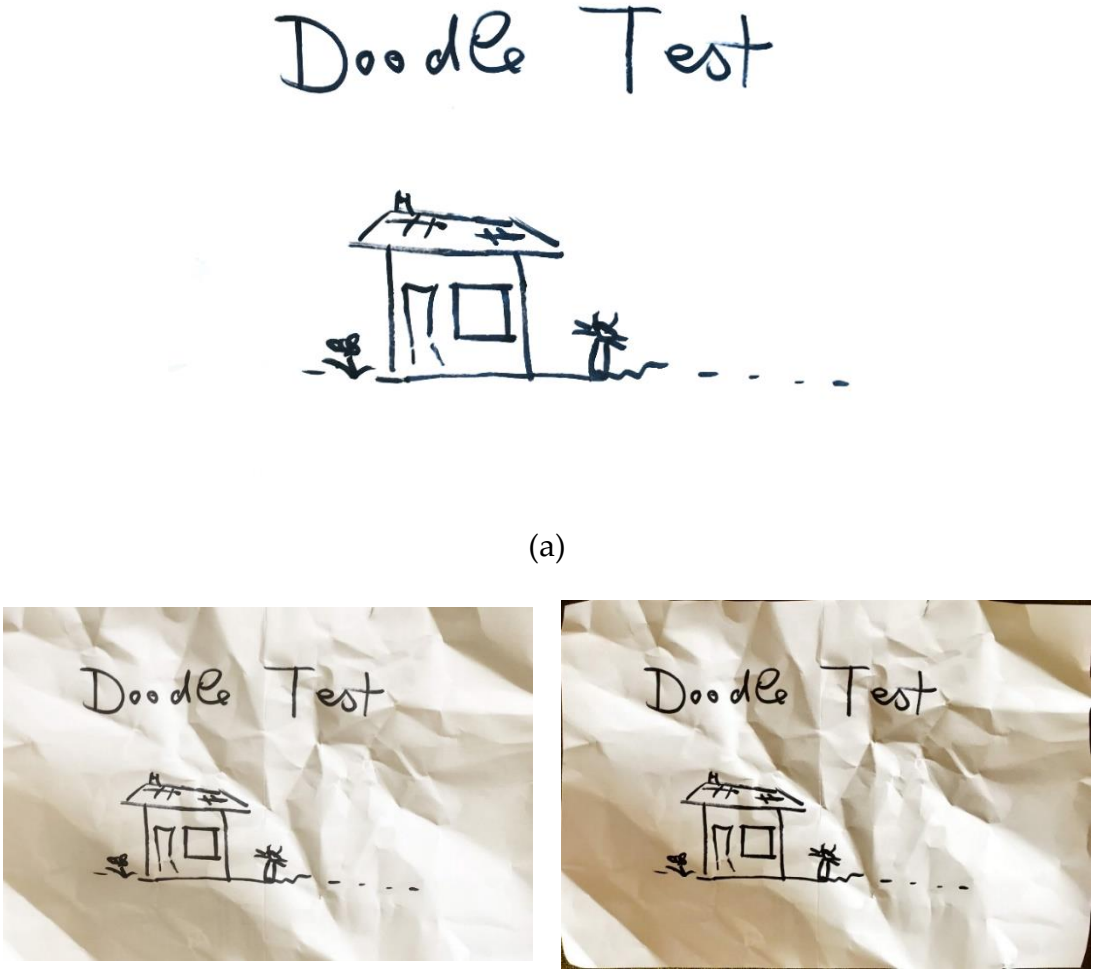

Figure 5. Sample results from scans of content on crumpled paper. (a) CleanPage; (b) Google PhotoScan; (c) Microsoft Office Lens.

CleanPage creates a homogenous background, crops to background surface automatically, and preserves image content while erasing surface imperfections. This creates a cleaner, more accurate reproduction of the original and therefore a better scan, while the other methods make for better photographs.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

(a) (b) (c)

Figure 6. Sample results from a magazine page scan showing a zoomed paragraph. (a) CleanPage; (b) Google PhotoScan; (c) Microsoft Office Lens.

3.3. Image Quality Assessment

All scans were scored using DE and SSIM. The original unscanned versions of the synthesized color and text images were also included in the tests for reference. While visual results may be subjective, DE and SSIM offer objective ways of assessing the scans. **Table 2** shows the DE scores for the three methods with the original, unscanned versions showing the best score as expected. The effects of real paper on the quality of the output can be seen clearly in the DE scores for the paper tests.

Table 2. Comparison of scanning methods using DE as a quality measure. A lower DE score suggests better quality.

	Original	CleanPage	Microsoft Office Lens	Google PhotoScan
Synthesized Color	**0.5978**	0.8941	1.5917	6.2870
Synthesized Text	**0.2743**	0.5319	5.8915	6.7118
Crumpled paper	-	**0.5604**	7.2129	6.7611
Text on paper	-	**1.7964**	6.7725	6.2016

SSIM is a measure of similarity but is normally used for measuring effects of noise, processing, distortion, etc. on an existing image, and so a ground truth of the same size, aspect ratio and scale is assumed. This is not the case for scanned images, as each scanning method can suffer from alignment and resizing issues. SSIM has also been shown to fail at identifying color-based distortions [41], such as those introduced by scanning. As a result, SSIM is not a reliable indicator of the percentage similarity in this case. However, as a comparative measure, it is still useful for ranking the images in order of similarity to a reference, as it will still give the highest score to the most similar image. **Table 3** shows the SSIM scores when each method's scans are compared to the original unscanned images.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

Table 3. Comparison of scanning methods using SSIM to compare each scan to synthesized reference images. Higher scores indicate greater similarity to the original unscanned image.

	CleanPage	Microsoft Office Lens	Google PhotoScan
Synthesized Color	**0.8332**	0.7873	0.6170
Synthesized Text	**0.8602**	0.6963	0.7490

3.4. User Experience

In the field of User Experience (UX), the golden rule is 'the simpler the better'. Previous work in the area provides insight into the document scanning problem from a UX perspective, focusing on hand-grip diversity and showing how different users approach the problem [42]. Using this insight, the simplest possible UI was designed: one button.

On tapping CleanPage's one button, all phases are performed automatically, and an output image is produced. Because a sequence of images is captured for processing, any imperfections in user performance (jitter, defocusing, inconsistent distance, etc.) have less impact on the result so capture does not require a particularly steady hand. CleanPage does not need a distinction to be made between image types as it works equivalently for all capture.

In the other methods tested – Microsoft Office Lens, Google PhotoScan and others – the result is highly dependent on the quality of the one image captured.

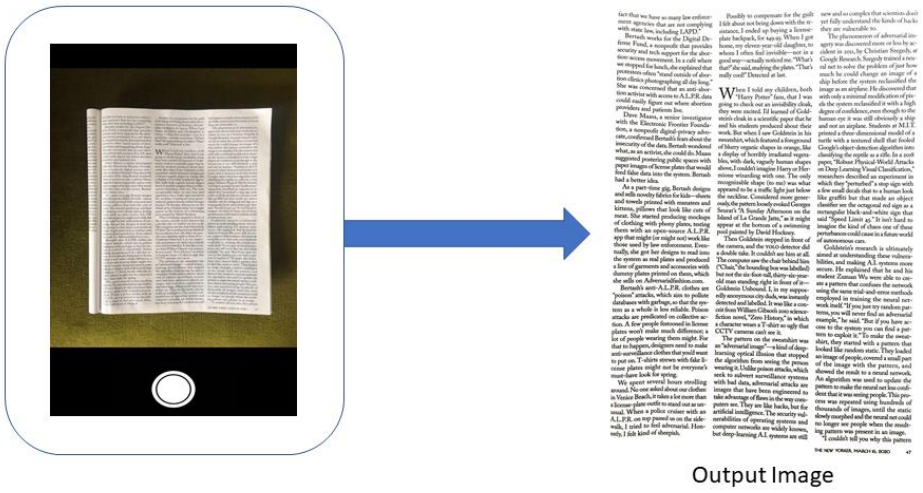

Figure 7. CleanPage's simple UI. One button-tap results in the output image.

Microsoft Office Lens has the next simplest UI, requiring just a steady capture to produce the output image, after first selecting the input type (document, whiteboard, etc.)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

Figure 8. Microsoft Office Lens UI. An optional manual corner correction phase is also offered but not shown, as it is often not necessary and still yields good cropping results.

Google's scanner proved the trickiest to use. Once the capture button is pressed, four stitching points appear, and the camera must be moved to each point in order to capture the subject. This often leads to tilt warnings, as it is required that the camera be held at the same angle throughout. The user performance at this stage has a significant impact on the quality of results and both stitching errors (see **Figure 4 (c)**) and cropping errors (see **Figure 6 (b)**) are frequent.

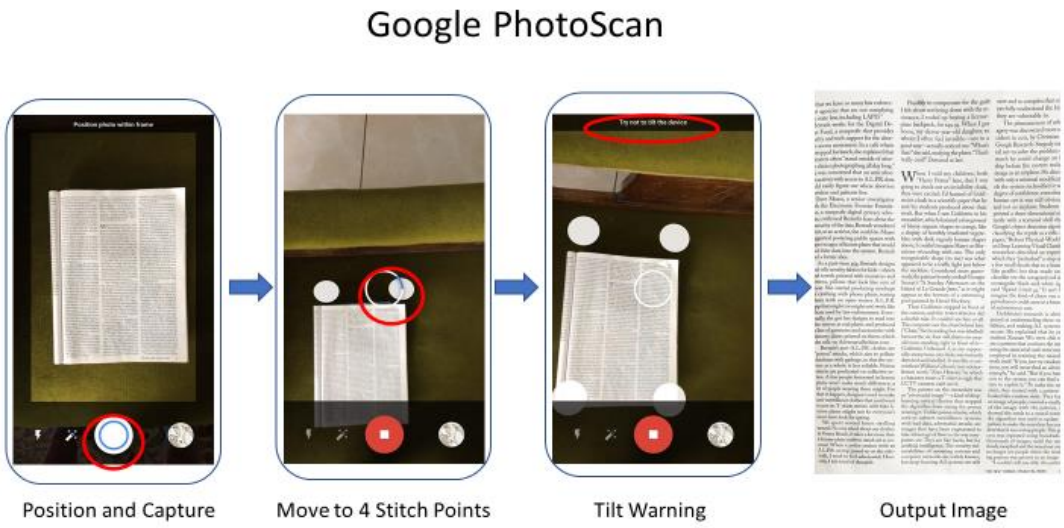

Figure 9. Google PhotoScan UI. An optional manual corner correction phase is also offered and is not shown, though is often necessary as automatic cropping is frequently incorrect.

3.4.1. Processing Time

Although scanning time depends highly on phone quality, a comparative analysis was made by using the same android smartphone with all three scanning apps. Time was measured from the first step of the process in each case (See **Figure 7-9**) to the appearance of the output image onscreen.

With CleanPage, a short sequence of images is captured (taking less than 1 second) and the simple methods used in each phase of the design mean processing is fast. The total capture time, though dependent on phone quality, is short. As there is no need to wait for a steady capture, the process is the same for every capture, so the processing time also remains more consistent.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2020 doi:10.20944/preprints202007.0686.v1

Without manual intervention, Microsoft Office Lens' scanning times were comparable, though capture time increases in poor lighting, where the bounding box can struggle to attach to the subject. Even without manual intervention, Google PhotoScan was the slowest as four images need to be carefully captured. It also frequently required manual correction or repeated attempts, as cropping or stitching errors were common. This further increased the overall capture time, but these manual interventions were not included in the results in **Table 4** for fair comparison.

Table 4. Comparison of scanning times over several trials. Time was measured from the first step of the process to the appearance of the output image onscreen.

	CleanPage	**Microsoft Office Lens**	**Google PhotoScan**
Scanning Time	**8-11 seconds**	8-15 seconds	18-23 seconds

4. Discussion

4.1 Scanning Performance Assessment

While scanning is becoming a day-to-day necessity as we move from physical workspaces to online environments, there is still much room for improvement in the technologies available and in the means of assessing these technologies. For example, testing of accepted NR-IQAs highlighted their limitations when applied to scanned images, revealing that they are not suitable for this application.

Along with a new high-quality scanning technology, two novel performance assessments are presented here: the discrete entropy as a quality measure and a new technique for testing using synthesized reference images. The discrete entropy is validated as it gives consistently better scores for unscanned ground-truths than for their scanned counterparts. The new method of testing scanning performance, by using synthesized images displayed on a tablet, allows comparisons to be made to these ground-truth images using the standard reference-based quality measure, SSIM.

4.2 CleanPage

The quality of CleanPage's outputs can be seen in both the visual results and in the qualitative and quantitative comparisons presented here with the scanning apps provided by software giants Microsoft and Google. Its fast and simple UX, without the need for manual intervention, makes it easier to use than other apps, particularly those using well-established stitching and mosaicking approaches. The design also reduces reliance on the user's careful, steady hand to produce quality outputs, as it is relatively robust to motion. As seen in the presented results, it works on a variety of scan types without the need for user input, making it consistently fast across numerous scanning trials. The performance of the system is not only comparable to the leading scanning apps but outperforms them in removing surface imperfections to produce a clean scan, scored using DE, and in content retention, scored using SSIM with synthesized ground truth images as reference.

In future work, a more thorough quality measure will be designed to capture contrast, smoothness, sharpness, etc. This quality measure may be used as an input to the system outlined here, to select the best images in the sequence, remove poor quality images and create a weighted average in the merging phase. While the focus here is on acquisition, the outputs from CleanPage could be combined with an OCR algorithm to develop a full scan-to-text design. Although initial testing shows consistency in OCR results, achieving typically less than 5% word error rate, OCR was not included here, as CleanPage is more broadly applicable to both document and whiteboard capture as well as to both text and images.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Acknowledgements: I would like to thank Claire Chambers for useful discussions and valuable insights. I hope that these sentences are sufficiently succinct.

References

1. Ôn Vũ Ngoc, M.; Fabrizio, J.; Géraud, T. Document Detection in Videos Captured by Smartphones using a Saliency-Based Method. In Proceedings of the 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW); 2019; Vol. 4, pp. 19–24.
2. Leal, L.R.; Bezerra, B.L. Smartphone camera document detection via Geodesic Object Proposals. In Proceedings of the 2016 IEEE Latin American Conference on Computational Intelligence (LA-CCI); IEEE, 2016; pp. 1–6.
3. Puybareau, É.; Géraud, T. Real-time document detection in smartphone videos. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP); IEEE, 2018; pp. 1498–1502.
4. Attivissimo, F.; Giaquinto, N.; Scarpetta, M.; Spadavecchia, M. An Automatic Reader of Identity Documents. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC); 2019; pp. 3525–3530.
5. Aggarwal, V.; Jajoria, S.; Sood, A. Text Retrieval from Scanned Forms Using Optical Character Recognition. In *Sensors and Image Processing*; Springer, 2018; pp. 207–216.
6. Saha, M.; Chakraborty, M.; Biswas, T. An Improved Approach for Document Image Mosaicing. *International Journal* 2016, *6*.
7. Eum, S. Image and Video Analytics for Document Processing and Event Recognition. PhD Thesis, 2017.
8. Liang, J.; DeMenthon, D.; Doermann, D. Mosaicing of camera-captured document images. *Computer Vision and Image Understanding* 2009, *113*, 572–579.
9. Javed, K.; Shafait, F. Real-time document localization in natural images by recursive application of a cnn. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR); IEEE, 2017; Vol. 1, pp. 105–110.
10. Yi, X.; Gao, L.; Liao, Y.; Zhang, X.; Liu, R.; Jiang, Z. CNN based page object detection in document images. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR); IEEE, 2017; Vol. 1, pp. 230–235.
11. Tang, Y.; Wu, X. Scene text detection and segmentation based on cascaded convolution neural networks. *IEEE Transactions on Image Processing* 2017, *26*, 1509–1520.
12. Dickson, P.E.; Kondrat, C.; Adrion, W.R.; Richards, T.D.; Szeto, R.B. Improved Whiteboard Processing for Lecture Capture. In Proceedings of the 2016 IEEE International Symposium on Multimedia (ISM); 2016; pp. 649–654.
13. Ha, T.Q.; Nguyen, Q.H.; Nguyen, B.T.; Tran, S.V.; Phuong, N.T.; Trinh, L.V. A Novel Method for Automatic Detection of Basic Shapes on Whiteboard Images using Faster RCNN. In Proceedings of the 2019 6th NAFOSTED Conference on Information and Computer Science (NICS); 2019; pp. 466–470.
14. Zhang, Z.; He, L.-W. Whiteboard scanning and image enhancement. *Digital Signal Processing* 2007, *17*, 414–432, doi:10.1016/j.dsp.2006.05.006.
15. Office Lens Is a Snap. *Microsoft Research* 2014.
16. Assefa, M.; Yngve Hardeberg, J. Evaluating the Naturalness and Legibility of Whiteboard Image Enhancements. *Electronic Imaging* 2019, *2019*, 86-1-86–9, doi:10.2352/ISSN.2470-1173.2019.14.COLOR-086.
17. Gupta, D.; Bag, S. A Local-to-Global Approach for Document Image Binarization. In *Computational Intelligence in Pattern Recognition*; Springer, 2020; pp. 693–702.
18. Calvo-Zaragoza, J.; Gallego, A.-J. A selectional auto-encoder approach for document image binarization. *Pattern Recognition* 2019, *86*, 37–47, doi:10.1016/j.patcog.2018.08.011.
19. Feng, S. A novel variational model for noise robust document image binarization. *Neurocomputing* 2019, *325*, 288–302.
20. Mondal, T.; Coustaty, M.; Gomez-Krämer, P.; Ogier, J.-M. Learning Free Document Image Binarization Based on Fast Fuzzy C-Means Clustering. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR); IEEE, 2019; pp. 1384–1389.
21. Sulaiman, A.; Omar, K.; Nasrudin, M.F. Degraded Historical Document Binarization: A Review on Issues, Challenges, Techniques, and Future Directions. *Journal of Imaging* 2019, *5*, 48, doi:10.3390/jimaging5040048.
22. Burie, J.-C.; Chazalon, J.; Coustaty, M.; Eskenazi, S.; Luqman, M.M.; Mehri, M.; Nayef, N.; Ogier, J.-M.; Prum, S.; Rusiñol, M. ICDAR2015 competition on smartphone document capture and OCR (SmartDoc). In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR); IEEE, 2015; pp. 1161–1165.

23. Chazalon, J.; Gomez-Krämer, P.; Burie, J.-C.; Coustaty, M.; Eskenazi, S.; Luqman, M.; Nayef, N.; Rusiñol, M.; Sidère, N.; Ogier, J.-M. SmartDoc 2017 Video Capture: Mobile Document Acquisition in Video Mode. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR); 2017; Vol. 04, pp. 11–16.
24. Zhukovsky, A.E.; Arlazarov, V.V.; Postnikov, V.V.; Krivtsov, V.E. Segments graph-based approach for smartphone document capture. In Proceedings of the Eighth International Conference on Machine Vision (ICMV 2015); International Society for Optics and Photonics, 2015; Vol. 9875, p. 98750P.
25. Chen, F.; Carter, S.; Denoue, L.; Kumar, J. SmartDCap: semi-automatic capture of higher quality document images from a smartphone. In Proceedings of the 2013 international conference on Intelligent user interfaces; Association for Computing Machinery: Santa Monica, California, USA, 2013; pp. 287–296.
26. Jiang, B.; Liu, S.; Xia, S.; Yu, X.; Ding, M.; Hou, X.; Gao, Y. Video-based document image scanning using a mobile device. In Proceedings of the 2015 International Conference on Image and Vision Computing New Zealand (IVCNZ); 2015; pp. 1–6.
27. Land, E.H.; McCann, J.J. Lightness and retinex theory. *Josa* 1971, *61*, 1–11.
28. Luqman, M.M.; Gomez-Krämer, P.; Ogier, J.-M. Mobile phone camera-based video scanning of paper documents. In Proceedings of the International Workshop on Camera-Based Document Analysis and Recognition; Springer, 2013; pp. 164–178.
29. Tareen, S.A.K.; Saleem, Z. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In Proceedings of the 2018 International conference on computing, mathematics and engineering technologies (iCoMET); IEEE, 2018; pp. 1–10.
30. Mokhtarian, F.; Mohanna, F. Performance evaluation of corner detectors using consistency and accuracy measures. *Computer Vision and Image Understanding* 2006, *102*, 81–94.
31. Shi, J. Good features to track. In Proceedings of the 1994 Proceedings of IEEE conference on computer vision and pattern recognition; IEEE, 1994; pp. 593–600.
32. Lucas, B.; Kanade, T. Performance of Optical Flow Techniques. *Proceedings of the DARPA Image Understanding Workshop* 1981, 121–130.
33. Fischler, M.A.; Bolles, R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 1981, *24*, 381–395, doi:10.1145/358669.358692.
34. Jui-Cheng Yen; Fu-Juay Chang; Shyang Chang A new criterion for automatic multilevel thresholding. *IEEE Transactions on Image Processing* 1995, *4*, 370–378, doi:10.1109/83.366472.
35. Mittal, A.; Moorthy, A.K.; Bovik, A.C. No-Reference Image Quality Assessment in the Spatial Domain. *IEEE Transactions on Image Processing* 2012, *21*, 4695–4708, doi:10.1109/TIP.2012.2214050.
36. Mittal, A.; Soundararajan, R.; Bovik, A.C. Making a "Completely Blind" Image Quality Analyzer. *IEEE Signal Processing Letters* 2013, *20*, 209–212, doi:10.1109/LSP.2012.2227726.
37. Chan, R.W.; Goldsmith, P.B. A psychovisually-based image quality evaluator for JPEG images. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics; 2000; Vol. 2, pp. 1541–1546 vol.2.
38. Chen, S.-D.; Al-Najja, Y.; Azami, N.H.; Beh, K.S. Measuring Image Quality for Assessment of Contrast Enhancement Techniques. *Australian Journal of Basic and Applied Sciences* 2013, 11.
39. Zheng, L.; Shen, L.; Chen, J.; An, P.; Luo, J. No-Reference Quality Assessment for Screen Content Images Based on Hybrid Region Features Fusion. *IEEE Transactions on Multimedia* 2019, *21*, 2057–2070, doi:10.1109/TMM.2019.2894939.
40. Michalak, H.; Okarma, K. Improvement of Image Binarization Methods Using Image Preprocessing with Local Entropy Filtering for Alphanumerical Character Recognition Purposes. *Entropy* 2019, *21*, 562, doi:10.3390/e21060562.
41. Hassan, M.A.; Bashraheel, M.S. Color-based structural similarity image quality assessment. In Proceedings of the 2017 8th International Conference on Information Technology (ICIT); 2017; pp. 691–696.
42. Oh, J.; Kim, J.; Kim, M.; Choi, W.; Lee, S.; Lee, U. Understanding mobile document capture and correcting orientation errors. *International Journal of Human-Computer Studies* 2017, *104*, 64–79, doi:10.1016/j.ijhcs.2017.03.004.