International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

Enhance Photo Document for software Integration

Nisreen Nizar Raouf[1], Mohammad A. Taha Aldabbagh [2]
[1,2] *College of Computer Science and Mathematics,University of Mosul, Mosul, Iraq*

Corresponding author: First A. Author (nisreen.21csp9@student.uomosul.edu.iq).

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: Data integration is a critical component of Software Engineering (SWE) since it guarantees that various software systems, applications, and datasets work together seamlessly. This connection allows for the aggregation of data from different sources, improving data consistency and precision. This paper presents a comprehensive methodology for auto-enhancing document images of University of Mosul data centers and their quality assessment through data integration with a cloud platform. Our approach has two stages: integration and cloud platform. The university's data center is seamlessly connected to a cloud platform using a FastAPI during integration, allowing efficient data exchange while maintaining data protection. The cloud platform stage receives picture document representations and enhances them with tools and algorithms developed for image enhancement. The approaches employed in this study encompass image scaling, similarity evaluation utilising the Histogram of Oriented Gradients (HOG) algorithm, image warping employing the FLANN (Fast Library for Approximate Nearest Neighbours) algorithm, and image quality enhancement by the application of a Laplacian sharpening filter. Furthermore, the study includes many evaluations of feature extraction techniques, document enhancement algorithms, image similarity algorithms, and their practical implementation outcomes in cloud systems.. The proposed integrated cloud-based document enhancement has shown exceptional efficiency in data sharing and precise analysis and enhancement of picture documents within the university's data center.

Keywords: Cloud Data Integration, Document Enhancement, Software Engineering Integration

1. INTRODUCTION

Data integration has significantly transformed educational decision-making and efficacy initiatives over the past few decades. The vast quantity, rapid flux, and variety of sources of big data in the digital age have altered the way in which we monitor and comprehend education. The proliferation of digital technology has generated an accumulation of data that is beyond all comparison, necessitating enormous computational resources and creative analysis [1]. As educational institutions have integrated comprehensive student administration systems in the era of digitalization, a wealth of data has been amassed. Unfortunately, the vast volume of data gives rise to a number of concerns, including the possibility of errors and inaccuracies in decision-making processes and the complexity of data storage, particularly as a result of image distortion and document quality degradation during data processing and storage [2].

Data integration harmonizes data from many sources to improve quality, analysis, warehousing, migration, and synchronization. Heterogeneity, inconsistency, duplication, and security issues create problems. These issues can be addressed using many software technologies. Data integration platforms include Oracle Data Integration Platform Cloud, Microsoft Azure Data Factory, Google Cloud Data Fusion, Pentaho, SSIS, and Talend; ETL tools include Stitch, Fivetran, Matillion, and AWS Glue. These technologies enable seamless data movement across many systems, applications, and platforms, making them crucial to university data architecture. These tools help universities collect, organize, and integrate data from multiple sources to create a single and consistent data environment [3]. APIs are used to link data from different systems in modern technology. Data is often transmitted via a cloud platform for speed, scalability, and security. It ensures data integrity and simplifies analysis, reporting, and decision-making in the cloud. After integration, data can be used, enriched, reported, and analyzed to gain insights that boost academic institutions' competitiveness and operational efficiency in the quickly changing digital world [4].

Cloud computing allows Internet-based application and service consumption, reimagining IT solutions. The technology reduces the need for personal computers and local servers by processing, distributing, and storing huge amounts of data on remote servers [3]. This technology

E-mail:author's email

eliminates institutional and physical hurdles to solve processing power and storage issues and transforms businesses. Companies can use public, private, community, and hybrid clouds to access infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). It saves money, gives firms flexibility, and improves operations [5].

Efficient data integration is essential for seamlessly implementing algorithms and approaches to enhance documents stored in a university system data center. This becomes particularly important when dealing with common picture distortions that can occur during the upload process [1]. This technique of document enhancement includes identifying, correcting mistakes and inconsistencies within system data. Document improvement at educational institutions includes a wide range of tasks such as resolving text recognition errors in scanned documents, repairing picture distortions, and standardizing data formats. Preserving the quality of papers, particularly those comprising photographs and handwritten content, is a significant difficulty. By automating these procedures, educational institutions unlock the full potential of their data, allowing data-driven decision-making, trend detection, and targeted interventions to improve educational system performance [6].

The main contributions of this paper include the following: (1) Integrate the university's data center with a cloud platform via a FAST API for the purpose of fetching image documents. (2) Design two servers hosted on a cloud platform, where the first server represents a user interface that interacts (as demo which replace with when tools work in university data center) with the third-party user for the purpose of selecting the image and sending it to the other server, while the second server functions as the main server responsible for image document checking and enhancement procedures. (3) Apply algorithms specialized in tasks of image similarity assessment, image warping, and image enhancer. Subsequently, a comprehensive comparison and evaluation of these algorithms are conducted against other algorithms applied to these tasks.

2. DATA INTEGRATION

Many innovative ideas have been explored on the subject of data integration to harness the potential of cloud computing, scalable infrastructure, and sophisticated technologies to handle different difficulties across many domains. In 2019, presented an IoT-cloud platform that established a connection between IoT devices and cloud resources through the Restful HTTP protocol. The platform's performance was assessed using JMeter tests with varied numbers of IoT devices (150, 100, 50) and measured using HTTP server throughput and response time matrices. The results consistently showed that as the number of devices increased, response time increased, while device data reception throughput decreased [7]. While in 2020, proposed a cloud-based middleware for smart healthcare wearables that consists of three essential parts: a device for Bluetooth Low Energy (BLE) sensor data collection, a messaging service using MQTT for effective data transmission, and a data access interface with options for both Restful HTTP and MQTT websocket access. The middleware showed acceptable batch data access speeds, decreased stream data latency by up to 52%, and could manage 172 requests per second in settings with concurrent users [8]. Moreover, in 2020, they presented a generalized architecture (IoTaaS) for integrating IoT with cloud computing, emphasizing security with a protected gateway, data encryption, and robust authentication. It assigned all database tasks to a server administrator and controller, ensuring authenticated operations. The system supported real-time experiences through AI and context awareness, allowing for data processing modifications. It also emphasized quality of service (QoS) and service level agreements (SLA) to improve performance through the use of context information [9]. Furthermore, in 2021, presented an integrated IoT platform designed to improve elderly healthcare. Wearable sensors such as Fitbit were used to collect real-time health data, which was then stored on a cloud server using the OAuth protocol and accessed by smartphones. In addition, a web-based database application was developed to manage elderly information, facilitate remote health monitoring by doctors, and provide caregivers with emergency alerts. The experimental results underscored the platform's assets in terms of accessibility, security, and cost-efficiency [10]. While, in 2021, introduced, i2Factory a cloud-based integration platform for enterprise applications, utilizing model-driven engineering. The platform facilitates the creation, implementation, and management of integration solutions. They also established a platform-specific modeling language based on the Guaraná integration framework. I2Factory proved useful in developing, implementing, monitoring, and managing integration solutions systematically with cost-efficiency and a high level of abstraction in real-world testing [11]. Finally, in 2023, the INTUITIONS platform was created to facilitate epilepsy research by integrating electronic health records, neuroimaging, and EEG data. It's built with Django's Model-View-Template (MVT) architecture, a PostgreSQL database, and a variety of libraries and frameworks for data processing and user interface design. This system offered powerful data storage, retrieval, and visualization capabilities, as well as sophisticated functions for complicated searches, cohort formation, and user access control [12].to summary up Table (1) contain data integration techniques most related works.

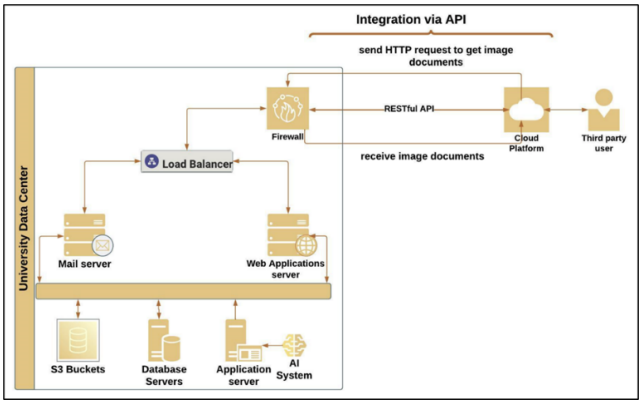
3. DOCUMENT ENHANCEMENT

A wide range of innovative techniques and technologies have been developed by researchers in the field of

document enhancement to address complex problems in document processing and correction. In 2020, presented the Adversarial Gated Unwarping Network, a deep neural network for rectifying distorted document images collected in natural situations. The approach uses three gated-based modules to capture the document's structural information. Furthermore, the model's accuracy and visual realism were improved via an adversarial method. Their approach exceeded the cutting-edge techniques in terms of visual quality and OCR rates [13]. While, in 2020, they proposed a fully convolutional network model for background removal and distortion correction of distorted document pictures. The FCN was utilized to estimate pixel-wise displacements and classify the document picture as either foreground or background. The document image was regarded as a displacement flow estimation. This method showed good learning capacities even when trained on virtual datasets that were not exact replicas of real-world data [14]. Furthermore in 2021, presented an end-to-end piece-wise unwarping method for document correction, addressing local and global deformations. It utilized three interconnected networks: the Shape Network for 3D shape estimation, the Piece-wise Unwarping Network for local unwarping maps, and the Global Stitching Network to combine these maps into a global unwarping map. The method significantly improved unwarping quality, achieving a PSNR value of 0.487 dB for both local and global unwarping tasks [15]. Moreover, in 2021, they Introduced DP-LinkNet for improving the binarization of degraded historical document images. DP-LinkNet follows the LinkNet and D-LinkNet architectures with the addition of two important modules: the Hybrid Dilated Convolution module for expanding the perceptual field and combining multiscale features, and the Spatial Pyramid Pooling module for encoding HDC outputs with multicore pooling. DP-LinkNet outperformed existing models, achieving FM ranging from 87.67% to 97.47% across various DIBCO and H-DIBCO dataset versions [16]. Another study in 2021, Proposed a page layout segmentation and text extraction baseline model for analyzing scanned Arabic book pages utilizing a fine-tuned Faster R-CNN structure (FFRA). The model included convolutional layers for extracting feature maps, region proposal networks and ROI pooling for suggesting potential regions and converting them into fixed-size regions, and a fully connected layer. Their approach achieved a classification accuracy of 99.4% for text verses non-text blocks classification [17]. Additionally in 2021, Suggested a technique based on the checkboard pattern and geometric transformation for the enhancement of document images. Their method involved aligning a warped document image with a ground truth checkboard that was not warped, then selecting specific control points to provide a mapping function for the dewarping process. The method outperformed state-of-the-art approaches, achieving a 5% OCR error rate when evaluated on the CBDAR 2007 dataset [18]. More, in 2023, a cutting-edge deep neural network called CGU-Net was introduced for document unwarping. This advanced technology aims to efficiently correct any warping in documents. It utilizes a dual-head architecture to predict a 3D grid mesh representing the document's shape and a 2D unwarping grid from a single RGB image. This approach implicitly encoded the relationship between the 2D and 3D grids, facilitating accurate document unwarping. CGU-Net achieved state-of-the-art performance on document unwarping tasks while maintaining a smaller network size for efficiency [19]. Finally in 2023, presented an automated algorithm for control point detection from simple to complex geometrical distortions in camera-captured document images. The method involved preprocessing, automated boundary detection, isometric mesh-based dewarping, noise reduction, and background removal. The method was tested on a variety of datasets, both handwritten and typewritten, and achieved outstanding dewarping results, even for severely distorted images [20].

4. THE PROPOSED METHODOLOGY

The proposed methodology encompasses two sequential stages: the integration stage and the cloud platform stage, depicted in Figure (1). The integration stage involves the seamless connection of the university's data center to the cloud platform through an application programming interface (API), enabling the exchange of data between the two entities. Subsequently, the cloud platform stage operates as a server hosted in the cloud, under the control of an external user. This stage is dedicated to receiving image document representations of data and executing enhancement procedures through specialized tools and



algorithms tailored for image enhancement. Following these enhancement processes, the enhanced data is

returned and securely stored within the university data center.

Figure 1: The Proposed Methodology.

A. integration stage

At this stage, the university's data center has integrated with a cloud platform through an API of type RESTful. The integration using the RESTful API has been extremely beneficial in exchanging image documents quickly and simply between two entities without causing any damage to the image documents during the exchange.

The process of fetching image documents from the university data center involves a sequence of steps, as shown in **Figure (2).** It begins with the cloud platform sending an HTTP request at the request of the third-party user to the data center via the Fast API integrated between them. As the request traverses the data center, it encounters a firewall designed to secure the internal network by filtering network traffic, thwarting unauthorized access, and mitigating potential cyber threats. Upon passing through the firewall, the request reaches a load balancer responsible for distributing network or application traffic across multiple servers. This strategic distribution ensures that no single server becomes overburdened. The load balancer then directs the request to the appropriate server, which could be a web applications server handling web-based requests or directly to database servers where the image documents are stored. If the image documents are stored in S3 buckets, the web applications server or another service within the data center retrieves the images from these storage buckets. Once the image documents are fetched, they traverse back through the data center infrastructure, exit through the firewall, and return to the cloud platform. Finally, the cloud platform delivers the image documents to the third-party user device that initiated the request.

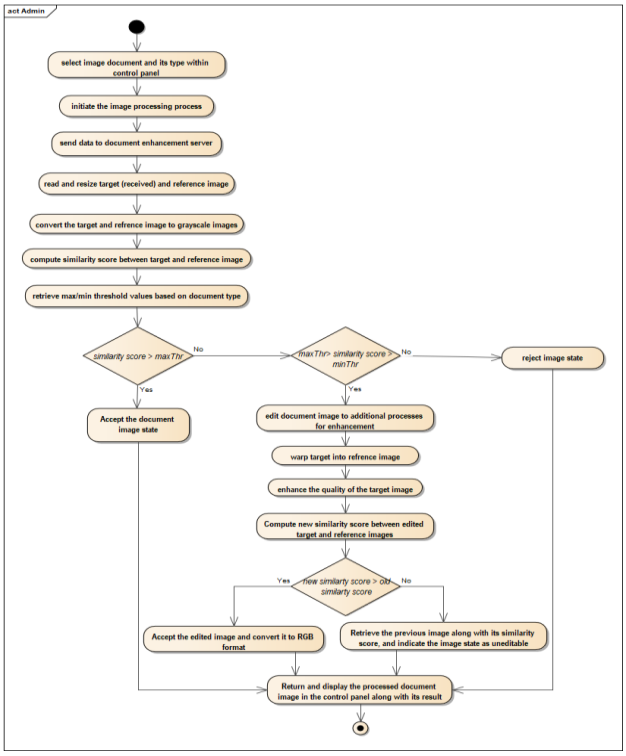


Figure 2: Diagram for the process of fetching image documents from the data center via FAST API between the cloud platform and data center.

B. Cloud platform stage

This stage represents a cloud platform controlled by external users via the Internet. It hosts two servers that are designed for the purpose of document enhancement. The first server (API Services Drive) acts as a user interface, which interacts with external users to handle document data. It exposes API services linked to a second server that external users can call to send and receive document-related information. The second server (document enhancement server), acts as a service to enhance the received documents through a variety of improvement tools such as warp image, image similarity, and image enhancer. In addition, it provides results on how accurately a document resembles the base document and gives a state of the document as to whether it is accepted, edited, or rejected. Once document enhancement operations are done, the server sends the enhanced document (image) along with the results to the first server and displays it to the user.

C. API Services Drive server

The API Services Drive server is a user interface constructed using Flask due to its perceived advantages

in simplicity and rapid data transfer to an associated server. Since the server is a user interface represented as a page designed with HTML, CSS, and JAVASCRIPT, a third-party user can interact with the page. Within this interface, third-party users can select an image and specify the type of document for examination and enhancement. Once the user presses the process button that appears on the page after selecting the image and type of document, the server transmits the image and document type to the document enhancement server. Once the checking and enhancement procedures on the document image are completed in the document enhancement server, the enhanced image document, along with the corresponding results, is returned to the API Services Drive server. The server then renders and displays this enhanced document on the page for the user's review.

D. Document Enhancement server

Document Enhancement Server serves as the main server dedicated to performing various enhancement operations for image documents. It is designed using the Fast API framework, chosen for its ease of implementation in image processing and efficient communication with other servers. The server's workflow involves receiving an image document along with its document type from the first server as input. Subsequently, it directs the received data to image enhancement processes, which include resizing, image similarity, image warp, and image enhancer tasks. Figure (3) illustrates the use case diagram for the document enhancement server and its role in processing image documents.

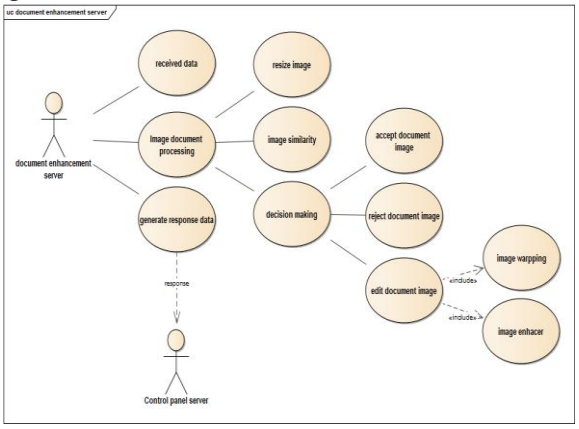


Figure 3: The use case diagram of the document enhancement server

E. Image resizes as target

In this process, the target image (the received document) and the reference image are resized to fit the required size based on the document type. The documents are specific to the Iraqi government and include five types of documents, each with its own unique size, as shown in Table (I). Therefore, the dimensions of both the received document image and the reference image are changed to depending on the document type.

TABLE I. **:** TYPES OF DOCUMENTS AND THEIR DIMENSIONS

Document Type	dimensions
Ration card	188, 755
Residence card	321, 204
Nationality cert	321, 204
Civil card	321, 204
National card	321, 204

F. Image similarity

After the image resizing process, the similarity of the target image to the reference image is measured. The similarity measurement process is done by extracting features using the HOG algorithm for both images and then measuring how similar the features of the target image are to the features of the reference image using the correlation coefficient measure.

The process of extracting features from an image using HOG [21] involves dividing the image into cells, and the size of each cell is 24* 24. For each pixel within the cell, the gradients Gx and Gy are calculated using the Sobel filters shown in equations (1) and (2) [21]. Then the magnitude (M) and angle value (θ) of each pixel are calculated as shown in equations (3) and (4) [21]. Where Gx , Gy represent the directional gradients of pixel intensities, indicating the rate of intensity changes along the horizontal and vertical directions in an image, magnitude (M) represents the overall intensity change or strength of the gradient at a particular pixel, and the angle value (θ) represents the direction of the gradient vector.

$$Gx(x,y) = P(x+1,y) - P(x-1,y) \quad (1)$$

$$Gy(x,y) = P(x,y+1) - P(x,y-1) \quad (2)$$

$$M(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \quad (3)$$

$$\theta(x,y) = arctan(\frac{Gy(x,y)}{Gx(x,y)}) \quad (4)$$

Then the histogram of each cell is computed by accumulating information about gradient orientations within the cell. Commonly, 9 orientation bins are used for HOG descriptors. Each orientation bin covers a specific range of gradient orientations, and the magnitudes of gradients within the cell contribute to these bins. Therefore, for one cell with 9 orientation bins, the number of histogram bins for that cell would be 9. Afterward, the formed histograms within the 24x24 cells are grouped into larger blocks, with each block comprising a 4x4 arrangement of cells. The histograms within each block are then normalized using L2 normalization to handle variations in gradient magnitude and provide a consistent representation of local orientation patterns across different blocks. Finally, the normalized block histograms are combined to create the final feature vector that represents the HOG feature for the entire image.

ALGORITHM 1: Image Similarity Task

INPUT: Target image, Reference image
OUTPUT: Convergence percentage between target image and reference image
Begin
// HOG algorithm process.
1: Divide images into cells of size 24x24.
2: **For** 1 **to** number of cells **do**
3: Calculate directional gradients (Gx and Gy) using Sobel filters.
4: Compute magnitude (M) and angle (θ) for each pixel within the cell.
5: Form histograms of gradient orientations within cell (9 orientation bins).
6: **End For**
7: Group histograms into larger blocks with each block comprising a 4x4 arrangement of cells.
8: **For** 1 **to** number of blocks **do**
9: Normalize histograms using L2 normalization.
10: **End For**
11: Combine the normalized histograms to create final feature vectors representing HOG features for both images.
// calculate similarity score and make decisions
12: Find the similarity score using the correlation coefficient (ρ) between feature vectors of target images and reference images.
13: Convert correlation coefficient score to a percentage value.
14: **If** Convergence percentage >= maxThr **then**
15: Accept the document image without further processing.
16: **Else If** Convergence percentage >= manThr **then**
17: The target image proceeds into image warp and image enhancer tasks.
18: **Else**
19: Reject the document image.
20: **End If**
End

Once the HOG features of both the received and reference images are captured, the similarity between the features is measured using a correlation coefficient [22] measure, which is calculated based on the covariance and standard deviations of these vectors described in Equation (5)[22]. Where $\bar{X}$, $\bar{Y}$ mean the averages of brightness for vector X and Y, respectively.

$$\rho(X,Y) = \frac{\sum_{n=0}^{N}(X_n-\bar{X})(Y_n-\bar{Y})}{\sqrt{\sum_{n}^{N-1}(X_n-\bar{X})^2}\sqrt{\sum_{n}^{N-1}(Y_n-\bar{Y})^2}} \qquad (5)$$

The correlation coefficient determines the degree of similarity between 1 and -1 for two vectors, where a value close to 1 indicates strong similarity, a value close to -1 indicates no similarity, and a value close to 0 indicates moderate correlation. This correlation coefficient value is then converted into a percentage value, reflecting the convergence percentage between the vectors. The obtained convergence percentage is subsequently compared to predetermined threshold values based on the document type, as outlined in Table (II). If the convergence percentage is equal to or exceeds the maximum threshold, the document image is accepted and forwarded back to the initial server without undergoing further enhancement processes. Conversely, if the convergence percentage is equal to or surpasses the minimum threshold, it proceeds to enhancement operations, which are image warping and enhancer tasks. In cases where the convergence percentage falls below both the maximum and minimum thresholds, the document image is deemed unsuitable for enhancement. Consequently, it is rejected and returned to the first server without undergoing any enhancement operations. Algorithm (1) represents a series of steps to implement the image similarity task.

TABLE II. **:** THRESHOLD PERCENTAGES ALLOWED FOR ACCEPTANCE OR MODIFICATION OF DOCUMENT TYPES.

Document Type	Threshold (max, min)
Ration card	(43, 23.0)
Front Residence card	(43, 23)
Back Residence card	(50, 31)
Nationality cert	(45, 25)
Front Civil card	(51, 27)
Back Civil card	(56, 29)
Front National card	(72, 38)
Back National card	(67, 40)

A. Image warping

At this stage, the target image is warped to match the warping of the reference image using the FLANN [23] algorithm with homograph transformation [24]. The process begins by extracting keypoints and descriptors from both the target image and the reference image using the SIFT [25] algorithm. Following the extraction of key points and descriptors from the reference and target images, the FLANN algorithm takes descriptors as input and employs the KD tree structure to represent these descriptors hierarchically, where each node within the KD tree encapsulates a descriptor. Once both KD trees are constructed for the reference and target image features, FLANN independently conducts a k-nearest neighbor search within each tree to obtain matches between the descriptors of target and reference image. It finds the two nearest neighbors for each descriptor in the target image within the set of descriptors in the reference image. After obtaining matches using FLANN, the Euclidean distance (L2) [26] metric is used to calculate the distance between descriptors of the target image and their corresponding nearest neighbors in the reference image. Given a set of descriptors in the reference image *R* and a set of descriptors in the target image *T*, each represented as a descriptor D, the Euclidean distance is obtained by equation (6) [26]. Where D_i^R is the descriptor for feature *i* in the reference image, D_j^T is the descriptor for feature *j* in the target image, and *n* is the dimensionality of the descriptor.

$$D_{L2}\left(D_i^R, D_j^T\right) = \sqrt{\sum_{i=1}^{n}(D_{i,k}^R - D_{j,k}^T)^2} \quad (6)$$

Lowe's ratio test is then used to filter out matches based on distance ratios. This test keeps only matches whose distance is less than 0.7 times that of the second-best match. By removing unclear matches, this methodical filtering procedure makes sure that the matches that are chosen have a higher probability of correctly corresponding to characteristics in the reference and target pictures.

The matching descriptions were then associated with the target and reference images. After that, source and destination points were created by reshaping the keypoints and matching matched descriptors from the target and reference images into arrays. These points were then used as input to the homography[24] algorithm. Based on these related points, the homography method computes the homography matrix (M), which captures the transformation between the source (target) and destination (reference) points as stated in equation (7) [24]. Where, the coordinates of a point in the destination image (reference image) are represented by the first matrix on the left, the homography matrix (M) by the second matrix, and the coordinates of the corresponding point in the source image (target image) by the last matrix.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (7)$$

Furthermore, the RANSAC method with a threshold of 0.5 was used in the homography computation. This algorithm ensures the accurate estimation of the transformation, particularly in the presence of outliers or incorrectly matched key points, enhancing the reliability of the homography calculation. The calculated homography matrix is then applied to warp the target image onto the reference image. Finally, the resulting warped images are passed to the image enhancer task. Algorithm (2) represents a series of steps to implement the image warping task.

ALGORITHM 2: Image warping Task

INPUT: Target image, Reference image
OUTPUT: Warped target image
Begin

1: Use SIFT algorithm to extract keypoints and descriptors from R and T images.
//FLANN algorithm process.
2: **For** 1 **to** number of descriptors **in** target image **do**
3: Construct KD tree for descriptors of target image using FLANN algorithm.
4: **End For**
5: **For** 1 **to** number of descriptors **in** reference image **do**
6: Construct KD tree for descriptors of reference image using FLANN algorithm.
7: **End For**
8: FLANN Perform a k-nearest neighbor search within each tree to obtain matches between descriptors of the target and reference image.
9: Calculate distances between descriptors of target image and their nearest neighbors in reference image using Euclidean distance metric to obtain distance matches.
// Lowe's ratio test
10: **For** match (m, n) in number of matches **do:**
11: **If** distance of the first match (m) < 0.7 * distance of the second match (n) **Then:**
12: Append the first match (m) to the list of good matches
13: **End If**
14: **End For**
15: Reshape key points and corresponding matched descriptors of target and reference image into arrays, forming source (target) and destination (reference) points.
// homography algorithm process.
16: Use source and destination points as input for homography algorithm.
17: Calculate homography matrix (M) encapsulating transformation between source (T) and destination (R) points.
18: Employ RANSAC algorithm with threshold set to 5 during homography process for accurate estimation.
19: Apply found homography matrix to warp target image onto reference image.
20: Pass the warped target image to image enhancer task
End

G. Image enhancer

At this stage, a Laplacian sharpening [27] filter is applied to highlight features and improve the visual clarity of the target image. The process is done by applying a Laplacian operator to the image, which is described in equation (8) [27]. This results in an image where edges and fine details are enhanced, while smooth regions are suppressed.

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (8)$$

Afterward, the Laplacian-transformed image is multiplied by a constant factor, set to 0.3, to control the degree of sharpening. The scaled Laplacian is then subtracted from the original image to obtain the final sharpened image as described in equation (9). The resulting sharpened image may contain pixel values outside the standard 8-bit intensity range (0 to 255). To ensure the image remains within this valid intensity range, negative values are set to zero, and values exceeding 255 are capped at 255.

$$I_{sharpened} = I_{org} - 0.3 * I_L \quad (9)$$

After the target image undergoes warping and quality enhancement, its features are once again compared to those of a reference image to determine their convergence percentage. Subsequently, the original and updated convergence percentages are compared. If the updated convergence percentage is greater than the original, the document image is considered edited and then returned to the first server. However, if the updated convergence percentage is less than the original, suggesting that no significant improvement has been achieved, the image is deemed unimprovable. In this case, the original state of the image is preserved and returned to the first server as a document image that cannot be edited.

5. EXPERIMENTAL RESULTS AND ANALYSIS

The document enhancement servers hosted on the cloud platform were tested using three documentary images of the national document type. These images encompassed varying levels of quality, with the first image being of excellent quality, the second exhibiting good quality, and the third demonstrating poor quality. The images were sourced by submitting a request to the university's data center through an API that is linked to the cloud platform. Following this, a comparison was conducted between the selected algorithms for image similarity, image warping, and image enhancement tasks with other algorithms commonly employed in these tasks.

A. Document enhancement servers

In the first experiment in examining the image documents enhancement server, the selected **excellent-quality image document** demonstrated a convergence percentage of 98.19 when compared for similarity to a reference image, as shown in **figure(4-a).** Therefore, the

image document was accepted without the need for modification or enhancement. While, in the second experiment, the selected **moderate-quality image** document showed a convergence percentage of approximately 40.22 when compared with a reference image before editing the image document in additional enhancement processes. Given this low convergence percentage, additional enhancement processes including image warping and image enhancer, were applied to the document image. Following these modifications, the document's similarity was reevaluated against the reference image, resulting in an improved convergence percentage of 60.49, as depicted in **figure (4-b).** Finally, in the last experiment, the selected **bad-quality image** document showed a convergence percentage of 4.21 when compared to the similarity with a reference image,

as shown in **figure (4-c).** Consequently, due to this very low convergence percentage in comparison to the threshold required for acceptance or editing of the national document, the image document was rejected.

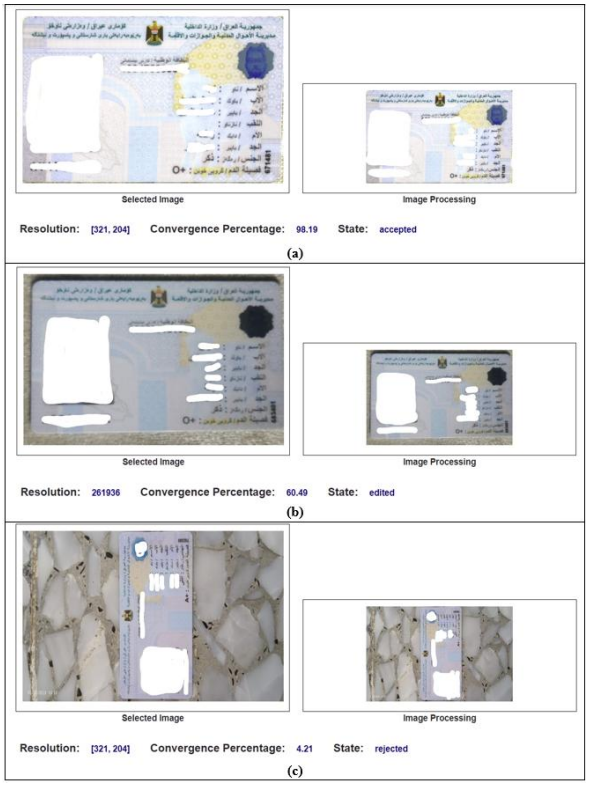


Figure 4: Experiment selecting a) an excellent-quality image document, b) moderate-quality image document, c) a bad-quality image document.

B. Comparison Image Similarity algorithms

In the image similarity task, a comparison was conducted between the selected HOG [21] algorithm and other widely employed algorithms, namely FAST [28], ORB [29], and SIFT [25]. The HOG algorithm is distinguished by its methodology of identifying features through the representation of local intensity gradients, thereby capturing pertinent patterns associated with edges and shapes within the image. In contrast, the ORB [29] algorithm seamlessly integrates the efficiency of FAST for key-point detection with the binary feature descriptors of BRIEF. The SIFT algorithm identifies distinctive key points within an image, describing them through histograms of gradient orientations in their respective surrounding regions. Finally, the FAST algorithm identifies interest points or corners within an image by scrutinizing pixel intensities in the vicinity of a specified pixel. The experiment to measure the similarity between two ration card images, as shown in **figure** (5), showed the superiority of the HOG algorithm in showing an acceptable convergence percentage using a correlation coefficient measure compared to other algorithms, as illustrated in table (III), which showed a low convergence percentage despite the presence of similarity details between two images.

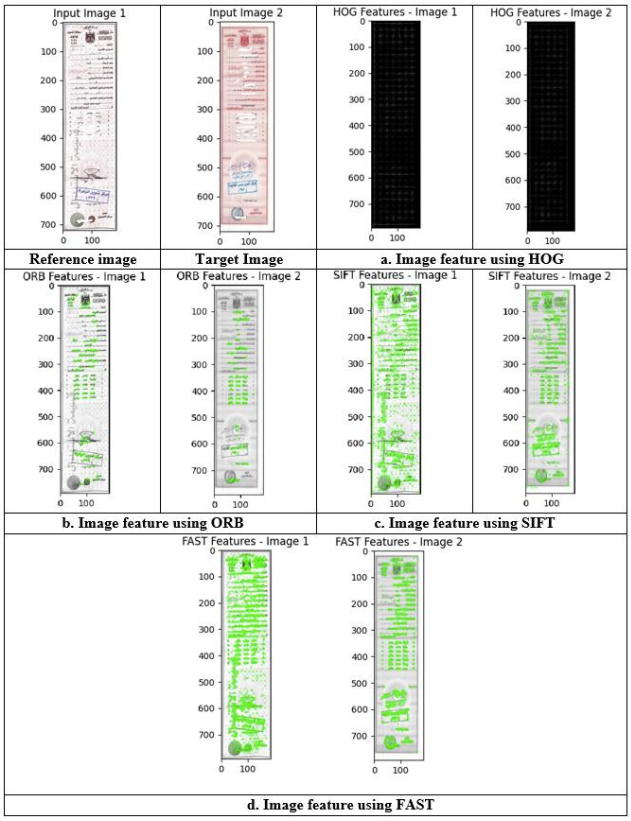


Figure 5: Comparison HOG feature extraction with other algorithms.

We additionally compared the correlation coefficient measure with the cosine measure in measuring the similarity between the features of the two images extracted by the algorithms shown in the figure. As shown in table (III), it was observed that the chosen correlation coefficient measure was superior in the accuracy of measuring similarity and in reasonable proportions compared to the cosine measure. This preference is attributed to the inherent capability of the correlation coefficient measure to capture nuanced relationships between features, thereby enhancing its accuracy in quantifying similarity.

TABLE III. . COMPARISON BETWEEN THE CORRELATION COEFFICIENT MEASURE AND THE COSINE MEASURE

Similarity metrics	**HOG**	**ORB**	**SIFT**	**FAST**
correlation coefficient similarity	44.61%	10.41%	21.26%	4.77%
Cosine similarity	19.10%	17.92%	52.52%	60.13%

C. Comparison Image warping algorithms

In the image warping task, a comparative analysis was undertaken between SIFT [25] with the FLANN [23] matcher and ORB [29] with the brute force matcher [30]. Experiments were systematically executed across four different images, as illustrated in Figure (6). The discernible observations from these experiments indicated a superior performance of the FLANN matcher over the brute force matcher. This superiority is attributed to FLANN's utilization of the KD-trees algorithm, which enables an efficient search for approximate nearest neighbors rather than exact matches between features of images. Consequently, this approach yielded accurate and lucid image warping across diverse images. In contrast, the brute force matcher, oriented towards precise feature or point matching between images, exhibited perceptible errors in the warping of certain images.

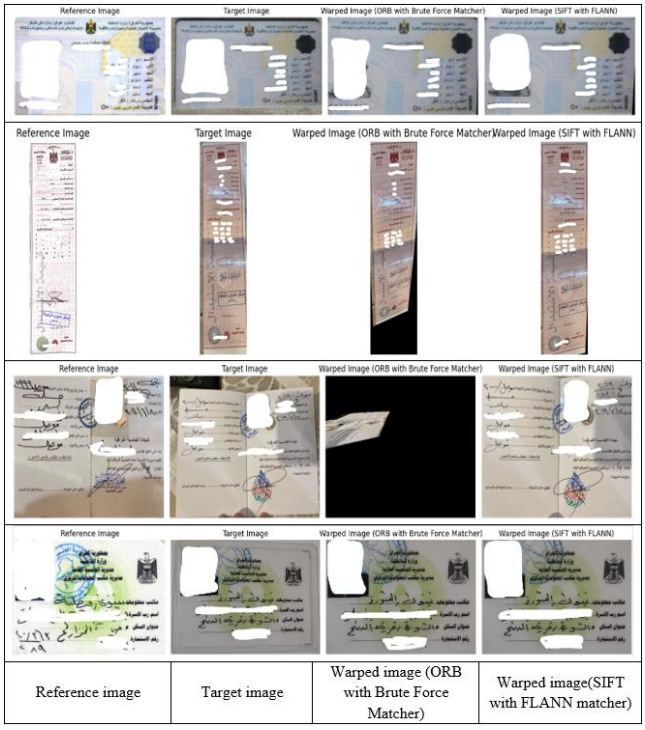


Figure 6: Comparison between SIFT with FLANN matcher and ORB with Brute Force Matcher.

D. Comparison Image enhancer algorithms

In the image enhancer task, a comparison was made between the selected Laplacian sharpening filter and the Gaussian smoothing filter. As **Figure** (7) shows, the studies were conducted methodically on four distinct images. The results of these tests clearly showed that, in comparison to the Gaussian smoothing filter, the

Laplacian sharpening filter is more effective in boosting image quality and text clarity. The Laplacian sharpening filter possesses the inherent capability to effectively highlight high-frequency features and edges inside images. On the other hand, the Gaussian smoothing filter, originally developed for the purpose of noise reduction and blurring, has a more restricted capacity to preserve and improve intricate visual features.

Figure 7: A Comparison of Gaussian Smoothing Filter and Laplacian Sharpening Filter.

6. CONCLUSIONS

In summary, the proposed cloud-based document enhancement system presents a feasible option for universities and other entities responsible for overseeing extensive document repositories. By employing contemporary algorithmic methodologies and a secure cloud infrastructure, the system may significantly enhance the lucidity and excellence of scanned documents. Furthermore, the effectiveness of the proposed methodology was demonstrated by empirical findings on the document enhancement servers hosted on the cloud infrastructure. The methodology was showcased through a sequence of experiments utilizing images of varying degrees of quality, showcasing its ability to accurately ascertain document similarity, perform necessary enhancements, and make educated decisions regarding document acceptance or rejection. Furthermore, the chosen approaches were found to be superior when compared to other regularly used algorithms for image processing applications. The algorithms that were chosen consistently outperformed their competitors in terms of accuracy, efficiency, and their ability to preserve and enhance visual characteristics in various tasks such as evaluating picture similarity, warping images, and enhancing images

ACKNOWLEDGMENT

I would like to extend my heartfelt thanks and deep appreciation to the programming team at the Computer Centre of Mosul University and the Software Department for their important guidance throughout the process of design and developing the suggested system. Your assistance was vital in our achievement. Thank you.

REFERENCES

[1] C. Fischer et al., “Mining big data in education: Affordances and challenges,” Review of Research in Education, vol. 44, no. 1, pp. 130–160, 2020.

[2] P. Kaushal, “Integrated Information Systems in Higher Education: Systematic Review and Research Opportunities,” Anwesh, vol. 4, no. 2, p. 28, 2019.

[3] N. A. H. M. Rodzi, M. S. Othman, and L. M. Yusuf, “Significance of data integration and ETL in business intelligence framework for higher education,” in 2015 International Conference on Science in Information Technology (ICSITech), IEEE, 2015, pp. 181–186.

[4] [P. Groth, A. Loizou, A. J. G. Gray, C. Goble, L. Harland, and S. Pettifer, “API-centric linked data integration: The open PHACTS discovery platform case study,” Journal of web semantics, vol. 29, pp. 12–18, 2014.

[5] B. Kwofie, “Cloud computing opportunities, risks and challenges with regard to Information Security in the context of developing countries: A case study of Ghana.” 2013.

[6] D. L. Goodhue, M. D. Wybo, and L. J. Kirsch, “The impact of data integration on the costs and benefits of information systems,” MIS quarterly, pp. 293–311, 1992.

[7] A. Bhawiyuga, D. P. Kartikasari, K. Amron, O. B. Pratama, and M. W. Habibi, “Architectural design of IoT-cloud computing integration platform,” TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 17, no. 3, pp. 1399–1408, 2019.

[8] A. Bhawiyuga, S. A. Kharisma, B. J. Santoso, D. P. Kartikasari, and A. P. Kirana, “Cloud-based middleware for supporting batch and stream access over smart healthcare wearable device,” Bulletin of Electrical Engineering and Informatics, vol. 9, no. 5, pp. 1990–1997, 2020.

[9] S. A. Alsaidy, A. M. Al-Chalabi, A. H. Alnooh and M. A. Sahib, "Multi-resource Power Efficient Virtual Machine Placement in Cloud Computing," 2021 International Conference on Computing and Communications Applications and Technologies (I3CAT), Ipswich, United Kingdom, 2021, pp. 93-97, doi: 10.1109/I3CAT53310.2021.9629421.

[10] M. Awadalla, F. Kausar, and R. Ahshan, “Developing an IoT platform for the elderly health care,” International Journal of Advanced Computer Science and Applications, vol. 12, no. 4, 2021.

[11] R. Z. Frantz, R. Corchuelo, V. Basto-Fernandes, F. Rosa-Sequeira, F. Roos-Frantz, and J. L. Arjona, “A cloud-based integration platform for enterprise application integration: A

Model-Driven Engineering approach," Softw Pract Exp, vol. 51, no. 4, pp. 824–847, 2021.

[12] B. Maharathi, F. Mir, K. Hosur, and J. A. Loeb, "INTUITION: a data platform to integrate human epilepsy clinical care and support for discovery," Front Digit Health, vol. 5, p. 1091508, 2023.

[13] X. Liu, G. Meng, B. Fan, S. Xiang, and C. Pan, "Geometric rectification of document images using adversarial gated unwarping network," Pattern Recognit, vol. 108, p. 107576, 2020.

[14] G.-W. Xie, F. Yin, X.-Y. Zhang, and C.-L. Liu, "Dewarping document image by displacement flow estimation with fully convolutional network," in Document Analysis Systems: 14th IAPR International Workshop, DAS 2020, Wuhan, China, July 26–29, 2020, Proceedings 14, Springer, 2020, pp. 131–144.

[15] S. Das et al., "End-to-end piece-wise unwarping of document images," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4268–4277.

[16] W. Xiong, X. Jia, D. Yang, M. Ai, L. Li, and S. Wang, "DP-LinkNet: A convolutional network for historical document image binarization," KSII Transactions on Internet and Information Systems (TIIS), vol. 15, no. 5, pp. 1778–1797, 2021.

[17] R. Elanwar, W. Qin, M. Betke, and D. Wijaya, "Extracting text from scanned Arabic books: a large-scale benchmark dataset and a fine-tuned Faster-R-CNN model," International Journal on Document Analysis and Recognition (IJDAR), vol. 24, no. 4, pp. 349–362, 2021.

[18] M. W. Labib and K. M. Amin, "Warped document image correction based on checkboard pattern and geometric transformation," IJCI. International Journal of Computers and Information, vol. 8, no. 1, pp. 30–54, 2021.

[19] [19] F. Verhoeven, T. Magne, and O. Sorkine-Hornung, "Neural Document Unwarping using Coupled Grids," arXiv preprint arXiv:2302.02887, 2023.

[20] C. H. Nachappa, N. S. Rani, P. B. Pati, and M. Gokulnath, "Adaptive dewarping of severely warped camera-captured document images based on document map generation," International Journal on Document Analysis and Recognition (IJDAR), vol. 26, no. 2, pp. 149–169, 2023.

[21] S. Ghaffari, P. Soleimani, K. F. Li, and D. W. Capson, "Analysis and comparison of FPGA-based histogram of oriented gradients implementations," IEEE Access, vol. 8, pp. 79920–79934, 2020.

[22] S. Kaneko, Y. Satoh, and S. Igarashi, "Using selective correlation coefficient for robust image registration," Pattern Recognit, vol. 36, no. 5, pp. 1165–1173, 2003.

[23] S.-Y. Hwang, J. H. Lee, C. S. Ha, M. Yang, and J. H. Choi, "Real-Time 2D Orthomosaic Mapping from Drone-Captured Images Using Feature-Based Sequential Image Registration," 2023.

[24] S. Wen, X. Wang, W. Zhang, G. Wang, M. Huang, and B. Yu, "Structure Preservation and Seam Optimization for Parallax-Tolerant Image Stitching," IEEE Access, vol. 10, pp. 78713–78725, 2022.

[25] Y. Gao, "Article identification for inventory list in a warehouse environment." 2014.

[26] K. Belattar and S. Mostefai, "Similarity measures for content-based dermoscopic image retrieval: A comparative study," in 2015 First international conference on new technologies of information and communication (NTIC), IEEE, 2015, pp. 1–6.

[27] N. Patel, A. Shah, M. Mistry, and K. Dangarwala, "A study of digital image filtering techniques in spatial image processing," in Proceedings of the 2014 International Conference on Convergence of Technology (I2CT), 2014, pp. 1–6.

[28] Y. Li, W. Zheng, X. Liu, Y. Mou, L. Yin, and B. Yang, "Research and improvement of feature detection algorithm based on FAST," Rend Lincei Sci Fis Nat, vol. 32, no. 4, pp. 775–789, 2021.

[29] Z. N. Sultani and B. N. Dhannoon, "Modified Bag of Visual Words Model for Image Classification," Al-Nahrain Journal of Science, vol. 24, no. 2, pp. 78–86, Jun. 2021, doi: 10.22401/ANJS.24.2.11.

[30] A. Jakubović and J. Velagić, "Image feature matching and object detection using brute-force matchers," in 2018 International Symposium ELMAR, IEEE, 2018, pp. 83–86.

Nisreen Nizar Raouf
Master's student in the College of Computer Science and Mathematics, Software Department, Mosul University

Dr. Mohammad A. Taha Aldabbagh
Assistant professor in software engineering. director of the Information Technology Center at the University of Mosul. Interested in developing software systems and managing software development projects.