# AI-Driven Document Intelligence: A Comprehensive Approach to Classification, Q&A Handling, and Fraud Detection

Kanika Kanojia[1*], Mehak Aggarwal[1], Princy Singhal[1], Shambhavi Rai[1], Sonanshi Goel[1], Poonam Bansal[1], Ritika Kumari[1,2]

[1]Department of Artificial Intelligence and Data Sciences, IGDTUW, Delhi, 110006, India.
[2]USICT, Guru Gobind Singh Indraprastha University, Dwarka, New Delhi, India.
[3]Department, Organization, Street, City, 610101, State, Country.

Contributing authors: kanikakj07@gmail.com; mehakagg1313@gmail.com; princysingla11@gmail.com; shambhavi.rai1604@gmail.com; sonanshig02@gmail.com; poonambansal@igdtuw.ac.in; ritikakumari@igdtuw.ac.in;

**Abstract**

With the growth in dependency on electronic verification systems used both by government and private organizations, there has been an increase in the usage of digitized manuscripts. This demands user-friendly and efficient document interaction tools. The optimization of the document management process has become AI-driven with features like identity document classification, authenticity checking algorithms, and interactive question answering. If we effectively identify the document type, we can streamline workflows and enable secure verification processes. The project employs deep learning techniques to classify documents like PAN cards, Aadhar cards, etc. Also, we employ a similar document template matching algorithm that can seamlessly extract features, match the template with a real document, and thereby detect fraud. Additionally, LLMs have been leveraged to enable interactive question-answering over documents. Experimental findings in our study showed the effectiveness of our work in improving efficiency, accuracy, and usability in document management.

**Keywords:** CNNs for Document classification, LLMs for Question Answering, LangChain, Similar Document Template Matching Algorithm, VGG16, Fraud Detection, ORB

## 1 Introduction

Recognizing duplicates of each image is a difficult task in image processing and pattern recognition, which is also known as a "template" in an image. When we look for a small template image in the scene image, it is called template matching. These resulting algorithms are generalized, which is referred to as template matching algorithms [1]. Some problems, like robust matching and searching, may occur in image processing. Rotation, distortion, dimension changes, partially obstructed, and light exposure alterations [2–7] are some of the problems that arise. Existing techniques like one characteristic or intricate characteristic groupings are used to tackle such problems [8]. However, it is difficult to distinguish similar items and correlate the backdrop complexity substantially and precisely. Due to the heavy reliability of algorithms on their processing time, template matching algorithms are presented [9] to lower the amount of computing power needed.

At its heart, AI-Driven Document Intelligence uses AI methods that facilitate smart document categorization, thus enabling quick identification and extraction of vital content segments. Classification and categorization of identity documents like Aadhaar cards, PAN cards, etc. facilitate more effective

document management. The spatial layout and organization of text blocks, graphics, and tables tell more about the documents. These descriptions are then used for classification [3, 10] or in computing similarities [4, 5]. The second type of method relies on text. These ways generate a descriptive text of the text content (extracted with an OCR in the case of scanned documents), such as a bag of words or Word2Vec, which is fed into classifiers [6].
A large number of people use large language models (LLMs) in real-world applications like chatbots, search engines, and coding assistants. LLMs have also greatly expedited the field of Natural Language Processing (NLP) [11]. Additionally, document management would be easier to build using chatbots and scalable AI or LLM applications with the LangChain framework. The LLM Model is a massive language model that may be used to write text, translate across languages, and respond to your questions in a useful manner [12].
The rest of the paper is structured as follows: Section 2 discusses related work. Section 3 presents materials and methods. In Section 4, the proposed methodology is presented. In Section 5, experimental results are discussed. Section 6 concludes the research.

# 2 Related Work

Recent advances in machine learning have led to notable improvements in domains like image recognition, query processing, and validation of language models. Using cutting-edge methods to improve model performance while lowering manual labor is a major trend. Large language models (LLMs) require automated validation; Huo et al. [13] addressed this need by integrating LLMs with LangChain to construct a chatbot that answers questions from PDF documents. Similarly, to increase the effectiveness of information retrieval and answering queries tasks, Xiaoxi et al. [14] presented Uni-Gen, a unified generative framework that combines query handling and retrieval into a single model. Both methods highlight a developing trend that optimizes complexity and performance by combining disparate tasks into unified models.
In visual recognition, Gabriella et al. [15] proposed a computationally efficient method using the bag-of-keypoints approach with vector quantization of affine invariant descriptors. By employing Naïve Bayes and SVM classifiers, their study balances simplicity with computational power. This theme is reinforced by Jia et al. [16], who presented Image-Net, a massive, annotated image database organized by WordNet's semantic hierarchy. Image-Net's impact on object identification, classification, and clustering highlights the importance of large-scale datasets in improving model training and accuracy across various tasks.
Hybrid approaches have also shown great promise in defect detection and image matching. Okubo et al. [17] introduced TM-CNN, a method that combines template matching with a convolutional neural network (CNN) to detect small defects in periodic structures, achieving a high F1 score of 0.988. This technique significantly reduces the need for manual annotations. Similarly, Xinwei Qi and Ligang Miao [18] developed an algorithm for multi-scale and rotated image template matching, using ring projection vectors and normalized cross-correlation to improve the accuracy of determining matching positions. Both studies demonstrate the effectiveness of hybrid methods in tackling specialized challenges with high precision.
In summary, these studies reflect trends in unifying tasks, leveraging large datasets, and adopting hybrid approaches to optimize model performance. While notable progress has been made, challenges remain, particularly in fully automating validation processes and integrating multimodal data for more complex tasks.

# 3 MATERIALS AND METHODS

In this section we discuss the materials and methods of our study.

## 3.1 Models/Frameworks used

This section demonstrates the four fundamental models used in our study. Section 3.1.1 shows the custom CNN model architecture properties that were built from scratch. Section 3.1.2 presents our VGG16 model and its modified architecture. Section 3.1.3 depicts the Pytesseract model as the OCR engine. Lastly, Section 3.1.4 refers to the LangChain framework.

### 3.1.1 Custom CNN Model:

Our custom CNN model is specifically applied to classify documents from PDFs or images, including more than one document such as Aadhar, PAN, driving license, passport, and voter ID, arranging individual pages into different document types. The model has been trained rigorously, and hyper

parameter tuning has been performed to accurately identify and retrieve various types of documents, leading to progress in document processing and information retrieval tasks.


**Fig. 1**: CNN Model Layers

The custom CNN model begins with an input layer that takes the raw input image (224x224x3) of the user uploaded ID card document image to prepare it for feature extraction.Following this, the model includes five convolutional layers that uses 32, 64, 128, 256, and 512 filter size each utilizing a 3x3 kernel. .This increasing filter count progression enables the extraction of increasingly complex features as the image is passed through the network. After each convolutional layer, a Max Pooling layer with a 2x2 kernel is used to down sample the feature map while retaining the crucial features. To prevent overfitting, dropout layers are introduced between crucial layers, with rates varying between 0.10 and 0.07. Model also includes a flatten layer that reshapes the output from the previous layers into a 1D vector, allowing it to connect to fully connected (dense) layers for final classification.Once the Conv2D layers are flattened, five dense(fully connected) layers, each with descending values of filter sizes (512, 256, 128, 64) are employed.The final output layer consists of 5 neurons, corresponding to the document types or classes being predicted.Softmax activation returns the class probabilities. For regularization, ReLU activation is applied throughout the convolutional and dense layers, with intermediate Dropout.

### 3.1.2 VGG16 Model:

We utilize the architecture of a deep CNN for retrieving features from document images by employing the pre-trained VGG16 model which was originally trained on a massive ImageNet dataset. This enhances the performance of our document classification model by enabling productive feature extraction from document images.

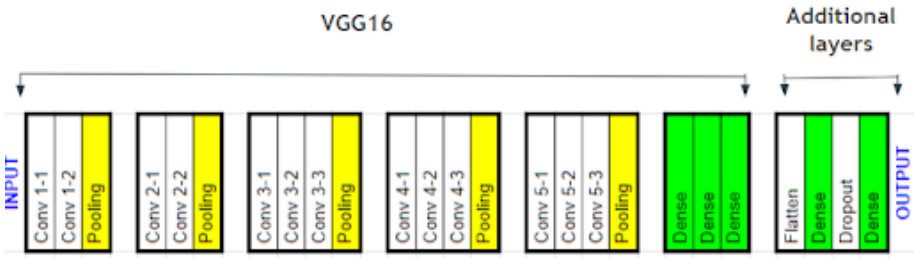


**Fig. 2**: VGG16 Model with custom layers

### 3.1.3 OCR for Information Extraction:

Before using Optical Character Recognition (OCR) with Pytesseract to retrieve text from an image, images have to be pre-processed. For example, this can involve transforming, resizing, sharpening, or converting them to grayscale. The retrieved text will then be refined using document-specific keywords to extract relevant information, which will be stored in an Excel file for further evaluation.

### 3.1.4 LangChain for Question Answering:

OpenAI Language Models utilize the LangChain framework for question answering. This feature enables the integration of task generation and language understanding within a unified system. LangChain is instrumental in implementing a question-answering system in a Language Model, as it supports the matching of inquiries with document-based insights to generate accurate and appropriate answers.

### 3.2 Dataset Properties

This section details the dataset used to train and validate the custom CNN model as well as the fine-tuned variations of the VGG-16 and VGG-19 models. Table 1 demonstrates each document class's dataset size used in classification model training. The sources of the gathering are paper-based manual scanning, Google Images, Kaggle repositories etc. To enhance the dataset vertical and horizontal data augmentation is employed. Image alterations such as flipping; rotation; translation are used to present disparities and make the model generalized better. First, using one-hot encoding to changed the categorical labels into numerical values which is followed by the rearranging of data to avoid bias.

**Table 1**: Dataset Size

| Document Type | Aadhar Card | Pan Card | Driving License | Voter Id | Passport |
|---|---|---|---|---|---|
| No. of Documents | 129 | 45 | 64 | 76 | 37 |

### 3.3 Performance Metrics

This section outlines the performance metrics used for evaluating and validating the models. The Confidence Score generates zero-centered log-likelihood ratios, where a higher score indicates a greater likelihood that a hypothesized word is correct, and a lower score suggests a higher probability of error [19]. Similarly, the Similarity Score measures the ratio of high-quality matches to the total number of matches, with values ranging between 0 and 1, and further details provided in Section 4.2. The SSIM Score in image-based assessments, (Structural Similarity Index) analyzes resemblance between images based on brightness, structural information, and contrast, In -1 to 1 ranging, where higher scores denote greater resemblance [1]. In the model training background, Train Accuracy measures the percentage of correctly categorized instances in the training dataset, delivering as a key metric for performance assesses during training [2], while Train Loss evaluates the discrepancy between the actual targets and predicted outputs within the training set [3]. To evaluates Validation Accuracy, generalization reflects the proportion of correctly designated instances in the validation set, indicating on unseen data how well the model performs [2], whereas Validation Loss captures the comparison in predicted outputs and actual targets in the validation set. A lower validation loss proposes proper generalization, whereas higher values could suggest overfitting [3].

## 4 PROPOSED METHODOLOGY

In this section we discuss the methodology of our study as shown in Figure 3.

### 4.1 Interactive Question Answering

Figure 4 shows the procedural mechanism of question answering system. To begin with, these PDF files are exported into a readable format. This text is then broken down according to the Character-TextSplitter so that normalcy is maintained and the material can be used for subsequent analysis. To comprehend this semantic meaning, the aforementioned text was converted into vector forms or embeddings through the usage of an OpenAI GPT.webdriver LLM model. For purposes of obtaining similar documents, the cosine similarity of the text section and the cosine similarity of the query vector are established. This has the advantage of being able to perform smarter information retrieval, where the user's query and the content of the document are what is being targeted and not the presence or absence of particular keywords.

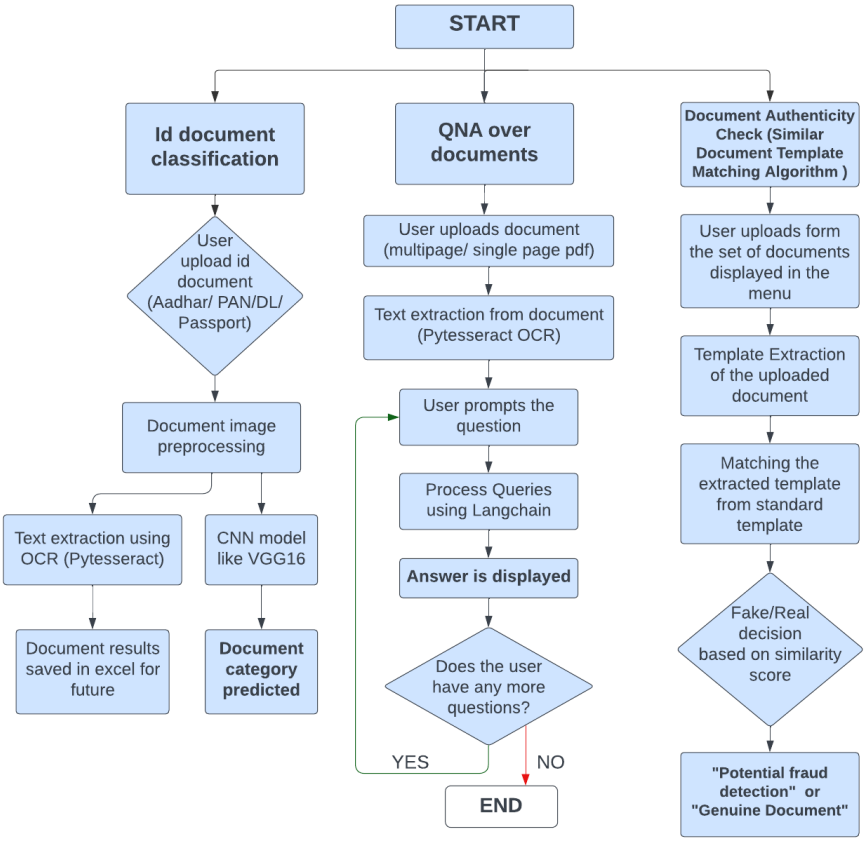


**Fig. 3**: User Interaction Flowchart

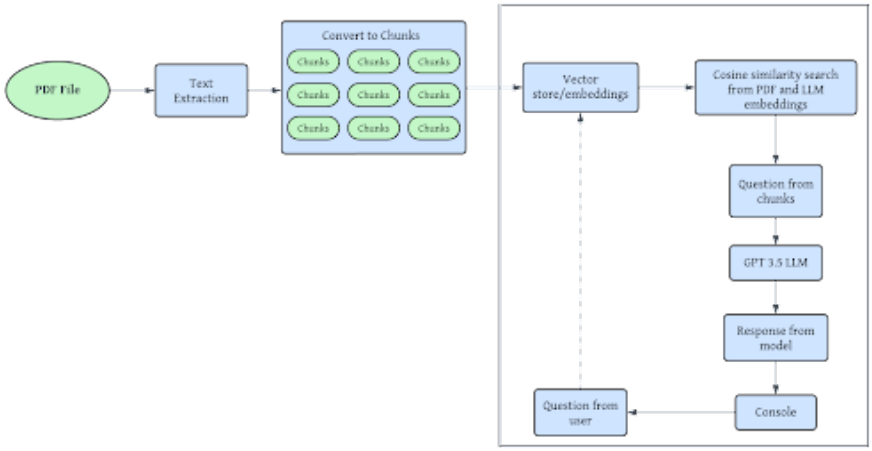


**Fig. 4**: System Architecture of Interactive Question Answering

## 4.2 Similar Document Template Matching

A combination of classical approaches and contemporary methods has been employed to develop the algorithm. The difficulty lies in correctly identifying the fake documents, particularly those where there are only slight differences, such as differences in logo or text styles, while still allowing some reasonable difference between compared authentic documents.[20].

### 4.2.1 Template Extraction

A combination of classical approaches and contemporary methods has been employed to develop the algorithm. The difficulty lies in correctly identifying the fake documents, particularly those where

there are only slight differences, such as differences in logo or text styles, while still allowing some reasonable difference between compared authentic documents.
Relevant features from the image were captured through some procedures with multiple steps. Regional of images was taken out based on the documents first. This is carried out to make it easy to detect edges small images by converting the images to grayscale. Then, burr free edge detection technique was used and followed by contour detection [21] . In order to eliminate undocumented variations in image quality, a dynamic thresholding
algorithm was specially designed for this purpose which improved edge detection and reduced noise drastically. Binary thresholding was used in order to remove all the areas to be removed where the pictures were captured in order to extract the ROIs features after which tones were enhanced.[22]

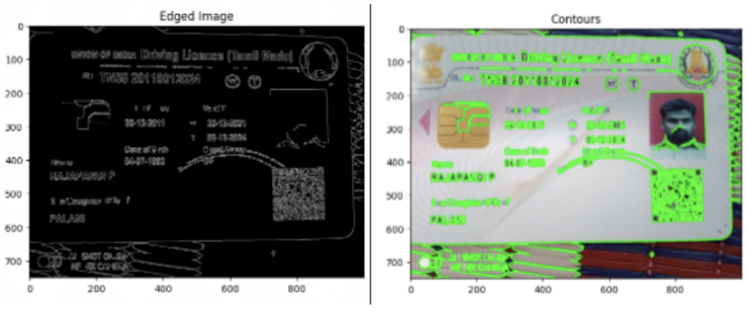


**Fig. 5**: Edge detection and contrast enhancement

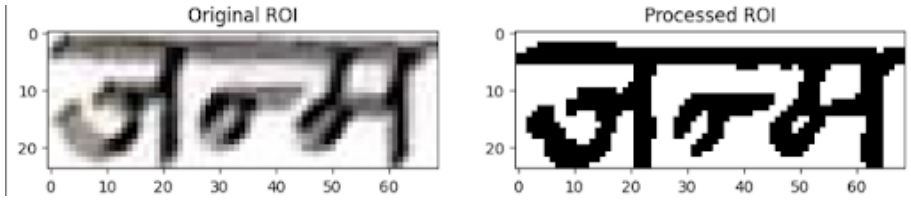


**Fig. 6**: Processed ROI of input image

### 4.2.2 Template Matching & Comparison

The features obtained in the previous step in form of key points and descriptors are passed to the function (match features) which finds two things : Total close matches and good matches. The most closely comparable characteristics between the test document and a predetermined authentic document are called total close matches. This is obtained using a brute-force or BF matcher with the hamming distance[23]. Further, out of total close matches, good matches were identified using the nearest neighbors ratio.

### 4.2.3 Fraud Detection

The criteria for identifying fraud are based on two key factors, Similarity Scoring and Decision Making. The algorithm's success in distinguishing between authentic and falsified documents is measured by the Similarity Score and SSIM Score, which evaluate the degree of similarity between documents. In Decision Making, binary outputs such as "probably genuine" or "potentially fake" are determined using decision-making parameters like the match threshold and similarity threshold. These thresholds help in classifying documents as either legitimate or suspicious based on the evaluated scores.

### 4.2.4 Iterative Development

Hyper parameter Optimization was employed to reduce high false positive rates that was one of the shortcomings discovered through initial analyses on an experimental dataset. These were fixed by improving the preliminary processing phases and adjusting parameters like blur kernel size and Canny thresholds in ROI extraction, Adaptive threshold and improved Gaussian blur parameters in preprocessing steps, K-value and matching ratio were adjusted and lastly match and similarity thresholds in make decision were modified.

# 5 Experimental Results

In this section we present the experimental results of our study. All the code implementation and experiments have been conducted on Google Colab version 1.0.0. In the aspect of document classification, our custom model achieved a notable accuracy of 83.33% after continuous monitoring as mentioned in Table 2, while VGG16 reached 87.69% accuracy with extra layers being added to its existing architecture. Interactive question answering also showed promising results, with confidence scores reaching as high as 98.43% as shown in Table 4. For matching document templates, we were able to accurately classify five out of six documents as shown in Table 5. However, we did encounter some occasional false positives due to variations in brightness and distortion.

## 5.1 Document Classification

Table 2 represents the various parameters evaluated after hyperparameter tuning of CNN Model.

**Table 2**: Hyperparameter Tuning Results for CNN model

| Hyperparameters Tuned | Train Loss | Train Accuracy | Validation Loss | Validation Accuracy |
|---|---|---|---|---|
| Kernel size (3,3) | 68.01 | 69.44 | 67.23 | 83.33 |
| Activation function (Sigmoid) | 59.14 | 69.44 | 53.38 | 83.33 |
| Dropout (0.05) | 1.09 | 51.90 | 96.05 | 64.15 |
| Dropout (0.07) | 57.31 | 73.42 | 74.99 | 77.36 |
| Batch size (64) | 1.34 | 40.00 | 1.42 | 25.00 |
| Batch size (128) | 1.43 | 33.85 | 1.44 | 34.09 |

Figure 7 and figure 8 depict the model accuracy and loss respectively for custom CNN Model while figure 9 and figure 10 depict training and validation accuracy and loss respectively for VGG16 Model for each epoch.

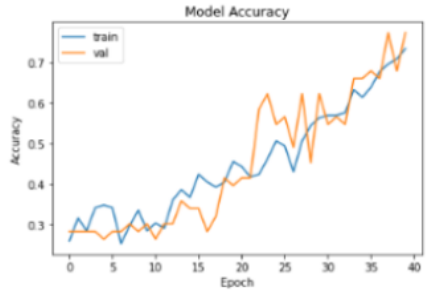


**Fig. 7**: CNN Model Train and Validation Accuracy

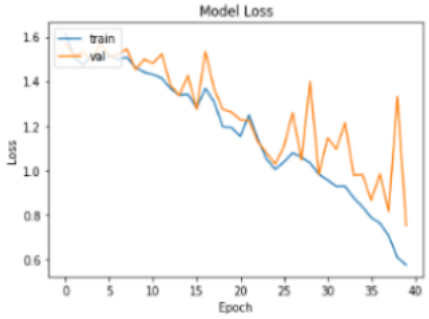


**Fig. 8**: CNN Model Train and Validate Loss

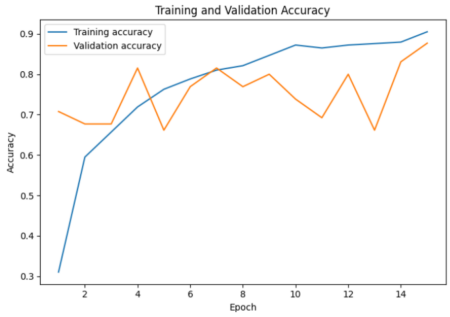


**Fig. 9**: Training and Validation Accuracy (VGG16)


**Fig. 10**: Training and Validation Loss (VGG16)

Table 3 illustrates the comparative analysis of different models used.

**Table 3**: Comparative Analysis

| Model | Number of layers | Epochs | Parameter | Train Accuracy (in %) | Validation Accuracy (in %) | Observation |
|---|---|---|---|---|---|---|
| CNN | 9 | 40 | Dropout = 0.07 | 69.44 | 83.33 | Limited data augmentation and a smaller network architecture |
| VGG16 | Additional 5 | 40 | Learning rate = 1e-5 | 90.51 | 87.69 | Better generalisation leading to improved accuracy |

## 5.2 Interactive Question Answering

Table 4 depicts a comparative analysis between LLM and BERT models, showcasing best results in LLM with less page PDF. The possible reason for nuanced and accurate response to the user prompts may be due deeper context understanding shown by LLMs on small PDF.

**Table 4**: Comparison between LLM and BERT

| Length of PDF (in pages) | Confidence Score (LLM Model) | Confidence Score (BERT Transformer) |
|---|---|---|
| 1 | 98.43 | 80.04 |
| 4 | 94.46 | 56.51 |
| 12 | 92.85 | 55.46 |

## 5.3 Document Template Matching

An overview of the experimental outcomes from our document verification algorithm is displayed in the Table 5 below. Hyper parameters were k-value of 2 for the k-nearest neighbors technique, nearest neighbor ratio of 0.75 and Similarity Threshold of 0.47.

**Table 5**: Analysis of Document Authenticity through Feature Matching and SSIM Evaluation

| Image | Key features | SSIM | Decision Obtained | Expected Decision | Observation |
|---|---|---|---|---|---|
| aadhar_img1 | With display picture variation, no qr code, missing name in Hindi | 0.29 | Potential fraud detected | Fraud | Significant changes detected very well |
| driver_img2 | Black and white, rotated | 0.54 | Document is likely genuine | Real | Color invariant |
| passport_img3 | Face hidden | 0.20 | Fraud | Fraud | Recognises inconsistencies with Display image in the id |
| aadhar_img4 | Date font size changes and logo missing | 0.20 | Potential fraud detected | Fraud | Worked on subtle variation like text font and size |
| aadhar_img5 | Colored, well aligned and illuminated | 0.21 | Potential fraud detected | Real | False positives are high. Here additional checks can be employed like re-uploading etc |
| pan_img6 | Colored, well aligned and illuminated | 0.51 | Document is likely genuine | Real | Ideal document |

# 6 Conclusion and Future Work

Different CNN designs have been evaluated and each displayed unique performance traits. Our custom CNN, with only minimal data enhancement, struck a compromise between efficiency and intricacy. Additional layers improved the ability to generalize performance of VGG16. Furthermore, our interactive Q&A system yielded reliable outcomes with high confidence levels. To ascertain document genuineness with most possible accuracy, we have fused up-to-date image processing techniques with the established techniques of comparing template shapes. Our project to develop AI-powered document intelligence has made certain that its templates are matched, answering interactive questions

and document classification. Subsequent development stages of AI-driven document intelligence will major on enhancing the accuracy levels of our models by training them on wider data sets as well as applying more advanced classifiers. In order to ensure smooth navigation, AI-Driven Document Intelligence will use document template matching so as to quickly identify document in clusters.

# 7 Conflict of Interest Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

# References


[1] Brunelli, R.: Template Matching Techniques in Computer Vision: Theory and Practice. John Wiley & Sons, ??? (2009)

[2] Jiang, X., Ma, J., Xiao, G., Shao, Z., Guo, X.: A review of multimodal image matching: Methods and applications. Information Fusion **73**, 22–71 (2021)

[3] Liu, B., Shu, X., Wu, X.: Fast screening algorithm for rotation invariant template matching. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 3708–3712 (2018). IEEE

[4] Lee, H., Kwon, H., Robinson, R.M., Nothwang, W.D.: Dtm: Deformable template matching. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1966–1970 (2016). IEEE

[5] Lan, X., Zhu, X., Gong, S.: Person search by multi-scale matching. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 536–552 (2018)

[6] McLaughlin, N., Ming, J., Crookes, D.: Largest matching areas for illumination and occlusion robust face recognition. IEEE transactions on cybernetics **47**(3), 796–808 (2016)

[7] Mudunuri, S.P., Biswas, S.: Low resolution face recognition across variations in pose and illumination. IEEE transactions on pattern analysis and machine intelligence **38**(5), 1034–1040 (2015)

[8] Ma, J., Jiang, X., Fan, A., Jiang, J., Yan, J.: Image matching from handcrafted to deep features: A survey. International Journal of Computer Vision **129**(1), 23–79 (2021)

[9] Kawanishi, T., Kurozumi, T., Kashino, K., Takagi, S.: A fast template matching algorithm with adaptive skipping using inner-subtemplates' distances. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., vol. 3, pp. 654–657 (2004). IEEE

[10] Kumar, J., Doermann, D.: Unsupervised classification of structurally similar document images. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1225–1229 (2013). IEEE

[11] Kim, J., Nam, J., Mo, S., Park, J., Lee, S.-W., Seo, M., Ha, J.-W., Shin, J.: Sure: Improving open-domain question answering of llms via summarized retrieval. In: The Twelfth International Conference on Learning Representations (2023)

[12] Pesaru, A., Gill, T.S., Tangella, A.R.: Ai assistant for document management using lang chain and pinecone. International Research Journal of Modernization in Engineering Technology and Science (2023)

[13] Deng, J.: A large-scale hierarchical image database. Proc. of IEEE Computer Vision and Pattern Recognition, 2009 (2009)

[14] Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1717–1724 (2014)

[15] Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5297–5307 (2016)

[16] Cimpoi, M., Maji, S., Vedaldi, A.: Deep filter banks for texture recognition and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3828–3836 (2015)

[17] Eglin, V., Bres, S.: Document page similarity based on layout visual saliency: application to query by example and document classification. In: Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings., pp. 1208–1212 (2003). Citeseer

[18] Xing, C., Wang, D., Zhang, X., Liu, C.: Document classification with distributions of word vectors. In: Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-pacific, pp. 1–5 (2014). IEEE

[19] Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29 (2015)

[20] Yenigalla, H., Reddy, B.R.S., Rahul, B.V., Raju, N.H.: Similar document template matching algorithm. arXiv preprint arXiv:2311.12663 (2023)

[21] Hossain, F., Asaduzzaman, M., Yousuf, M.A., Rahman, M.A.: Dynamic thresholding based adaptive canny edge detection. Int. J. Comput. Appl **975**, 37–41 (2016)

[22] Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International Conference on Computer Vision, pp. 2564–2571 (2011)

[23] Jakubović, A., Velagić, J.: Image feature matching and object detection using brute-force matchers. In: 2018 International Symposium ELMAR, pp. 83–86 (2018)