KEC Conference 2025

ISSN: 3021-940X (Print) DOI: https://doi.org/10.3126/injet.v2i2.78620

SmartKYC

Narayan Poudel[1], Saugat Neupane[2], Smarika Shrestha[3], Prof. Dr. Subarna Shakya[4], Suyasha Nepal[5, *]

[1]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, paudelnarayan434@gmail.com*

[2]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, saugatneupane50@gmail.com*

[3]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, smareeka.shrestha @gmail.com*

[4]*Department of Electronics and Computer Engineering, IOE Pulchowk Campus, Pulchowk, Lalitpur, Nepal, drss@ioe.edu.np*

[5]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, nepalsuyasha@gmail.com*

Abstract

This Paper demonstrates an automated system to streamline the KYC (Know Your Customer) process using machine learning. It integrates document validation, image pre-processing, text extraction, automated form filling, face detection, liveliness detection, and verification. Tesseract, utilizing LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network), extracts text from smart driving licenses to auto-fill digital KYC forms, reducing manual entry errors. MTCNN (Multi-task Cascaded Convolutional Networks) handles face detection, OpenCV ensures liveliness, and FaceNet (Inception-ResNet-v1) verifies the selfie against the ID photo for security. Successful verification grants confirmation; otherwise, users resolve issues. The system achieved 86.43% training and 86.25% validation accuracy, improving efficiency, accuracy, and user experience.

Keywords: KYC, Text Extraction, Face Verification, Liveliness Detection, Machine Learning.

1. Introduction

The KYC process is vital for verifying client identities but remains mainly manual, making it time-consuming and prone to errors. Traditional KYC procedures involve collecting, verifying, and analyzing documents, which reduces efficiency while increasing costs and delays. As organizations seek faster, more reliable solutions, automated KYC systems have emerged as a promising alternative. Advancements in digital technologies, particularly OCR and identity verification techniques, offer significant improvements in streamlining KYC processes. OCR means converting handwritten, typed, or printed text into machine-readable text (Patel, 2021). Intelligent integration of OCR ensures a seamless user experience while maintaining compliance standards. This dynamic synergy enables organizations to perform robust identity verification with unprecedented efficiency (Bhushan, et al., 2024). KYC-OCR integration allows seamless text extraction from ID documents, eliminating manual data entry and accelerating verification. However, challenges such as ensuring precise text extraction across diverse document formats and maintaining high-security standards remain critical considerations. SmartKYC addresses these challenges by developing an advanced automated system integrating multiple technologies for document validation, image pre-processing, text extraction, and identity verification. The objectives of this research include developing an automated system using OCR to extract text from smart driving licenses and automatically populate digital KYC forms. Additionally, it aims to integrate liveliness detection and facial recognition to compare the user's face with a recent selfie, ensuring enhanced security in the identity verification process. It utilizes Tesseract with LSTM and CNN for OCR, MTCNN for face detection, and FaceNet for identity verification. Additionally, OpenCV-based liveliness detection ensures that uploaded selfies are live captures. SmartKYC automates key aspects of the KYC process, minimizing manual effort, reducing errors, and improving efficiency while ensuring strong security and compliance.

**Corresponding Author*

Related Works

Automating Know Your Customer (KYC) processes presents a transformative opportunity to enhance customer experience, streamline operations, and bolster security across various sectors. By leveraging advancements in machine learning and computer vision, particularly in OCR technologies, face detection and verification, and liveliness detection, the potential for creating a more efficient and reliable KYC system is substantial. Understanding the current landscape of these technologies is crucial, as they underpin the functionality and success of automated KYC systems. Numerous studies and projects have explored these areas, providing valuable insights into the challenges and solutions associated with automating identity verification and regulatory compliance.

The model for text recognition leverages Tesseract-OCR, an optical character recognition engine, to convert handwritten, typed, or printed text into machine-readable text. Using Pytesseract, a versatile wrapper compatible with multiple programming languages and frameworks, the model interfaces seamlessly with Tesseract-OCR. OpenCV enhances the model's capabilities by enabling character and word detection, digit detection, handwritten text conversion, and multilingual text recognition. Tesseract's advanced architecture employs Long Short-Term Memory (LSTM) networks, which are particularly effective for training on long sequences with numerous classes. The architecture includes several stages: Line and Word Finding, where it preserves image quality without deskewing while identifying gaps between the baseline and mean line to detect words; Word Recognition, which involves separating joined characters and associating broken ones to enhance recognition; and the Character Classifier stage, which uses prototype features for accurate character classification. The classifier is trained on a substantial dataset of 60,160 samples across various fonts. Tesseract's robust framework, featuring comprehensive line and word recognition and accurate character classification, open-source availability and compatibility with numerous programming environments, makes it an ideal choice for OCR applications (Patel, 2021).

An innovative method for scene and text recognition was proposed, combining Bidirectional LSTM and CNN to address challenges such as blurred backgrounds, varying font sizes, and different image orientations (Kantipudi, et al., 2021). The methodology begins with contour detection, which identifies meaningful regions in the image, improving focus and speeding up recognition. Next, features are extracted using CNN to locate text regions, create bounding boxes, and classify individual characters. Finally, Bi-LSTM processes sequential features and captures contextual information, enhancing accurate character prediction. The method was tested on diverse datasets, including MSRATD 500, SVHN, UFPR-ALPR, SVT, and random datasets. These results demonstrate that integrating Bi-LSTM and CNN provides a robust framework for scene text recognition, outperforming traditional methods in terms of precision, recall, and F1 score. This approach is particularly practical in real-world scenarios involving complex backgrounds and varying text attributes, making it a valuable contribution to text recognition.

The ORB (Oriented FAST and Rotated BRIEF algorithm was introduced as an efficient and robust alternative to traditional feature detection and description methods like SIFT and SURF (Rublee, et al., 2011). The algorithm combines the FAST keypoint detector and BRIEF descriptor with significant enhancements to achieve rotation invariance and high computational efficiency. By incorporating an orientation operator using the intensity centroid method and aligning descriptors to the computed orientation, ORB effectively addresses the lack of rotation invariance in FAST and BRIEF. Furthermore, the algorithm employs a learning-based approach to reduce correlation and increase variance in binary tests, resulting in highly discriminative descriptors. ORB is significantly faster than SIFT and SURF—nearly two orders of magnitude faster than SIFT—while delivering comparable or superior performance in feature matching. It also exhibits higher robustness to noise, making it suitable for real-world applications such as object recognition and image stitching. Additionally, ORB's open-source availability under the BSD license eliminates the licensing restrictions associated with SIFT and SURF, making it a preferred choice for modern computer vision tasks.

An image processing-based project focused on scene text detection and recognition is presented, particularly targeting word detection and recognition in natural images. The methodology incorporates Tesseract V5, for text area detection and translation. The project is designed to extract ILU codes from images taken by a

camera mounted on a truck, achieving a character recognition accuracy exceeding 80%. The paper highlights the use of Long Short-Term Memory (LSTM) in Tesseract 5 for text recognition, which plays a crucial role in enhancing the pipeline's accuracy. Fine-tuning the page segmentation modes (PSM) in Tesseract 5 with LSTM led to substantial accuracy improvements. Different page segmentation strategies were applied at various recognition stages, such as treating the image as a single character or using automatic page segmentation. This comprehensive approach underscores the project's success in achieving reliable text recognition in challenging, natural image environments (Zacharias, et al., 2020).

OCRXNet, an OCR system designed for character recognition in identity documents, employs three different approaches: OCRXNetV1, OCRXNetV2, and OCRXNetV3. OCRXNetV1 incorporates image processing techniques such as Finding Contour and Cropping, Adaptive Thresholding, and Canny Edge Detection with Gaussian Blur, followed by text extraction using Tesseract. OCRXNetV2 structures the problem into four stages: dataset creation, identifying regions of interest, training the algorithm, and performing OCR on these regions, which demonstrated high accuracy for the specific use case. OCRXNetV3 employs the CRAFT text detector, an improvement over the previously used East text detector, for more efficient text detection. The paper highlights Tesseract's use within OCRXNetV1, leveraging image processing techniques to enhance text extraction. It also discusses Tesseract's integration of Long Short-Term Memory (LSTM) networks for improved OCR performance. The study underscores the critical role of accuracy in OCR applications such as Know Your Customer (KYC) processes and proposes future advancements involving expanded dataset collection and the development of deep networks tailored to various use cases (Arora, et al., 2020).

FaceNet is a groundbreaking system that directly learns a mapping from face images to a compact Euclidean space. Unlike traditional methods that rely on intermediate bottleneck layers, FaceNet optimizes the embedding. This space is designed such that distances directly correspond to a measure of face similarity. Once this embedding is produced, tasks like face recognition, verification, and clustering become straightforward using standard techniques. FaceNet employs a novel triplet loss function. Triplets consist of aligned matching/non-matching face patches, and the loss encourages separation between positive and negative pairs by a distance margin. This approach significantly improves representational efficiency, achieving state-of-the-art face recognition performance using only 128-byte embeddings per face. The paper introduces the concept of harmonic embeddings. These embeddings, produced by different networks, are compatible with each other, allowing direct comparison. This compatibility simplifies model upgrades and transitions. FaceNet achieves a new record accuracy of 99.63% on the widely used Labeled Faces in the Wild (LFW) dataset. It achieves an impressive classification accuracy of 95.12% on YouTube Faces DB, significantly outperforming previous methods. FaceNet's direct learning approach, triplet loss function, and harmonic embeddings represent significant advancements in face recognition technology. The system's scalability, robustness, and compatibility make it a powerful tool for real-world applications, from security to personal photo organization (Schroff, et al., 2015).

2. Methodology

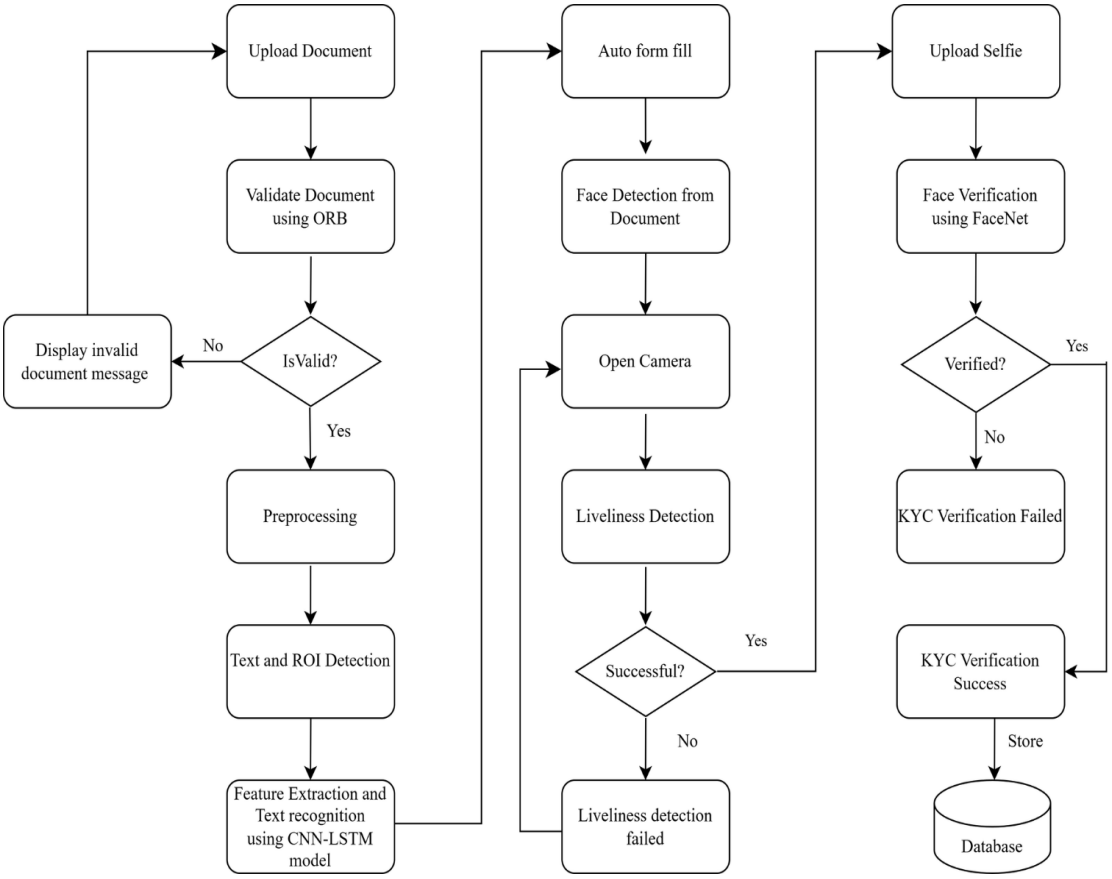

Figure 1. Working Mechanism of SmartKYC

2.1 Dataset Overview

3.1.1. Dataset Description

The MJSynth dataset is a comprehensive synthetic dataset designed to facilitate text recognition tasks, making it particularly useful for training and evaluating OCR (Optical Character Recognition) models. This dataset comprises approximately 9 million greyscale images of synthetically generated text, each typically depicting a single word. These words are rendered using various array of fonts, sizes, colours, and backgrounds to simulate real-world scenarios effectively. The dataset is meticulously organized, with a top-level directory containing around 4000 folders, each including seven subfolders. These subfolders house the images, each representing a different word. In addition to the photos and pictures, there is a text file (or multiple text files) containing the ground truth labels for each image. This file typically maps each image filename to the corresponding word it represents. This detailed structure and extensive variety make the MJSynth dataset an invaluable resource for advancing OCR technology.

Figure 2. The MJSynth Dataset

3.1.2. Data Pre-processing

For this project, we used a subset of 500,000 training and 100,000 test images, balancing computational efficiency and performance while ensuring sufficient data diversity. To enhance the dataset, we generated synthetic images containing special characters such as . - , + : and / using the TextRecognitionDataGenerator (trdg) library, adding 10,000 additional images to improve OCR robustness. Pre-processing steps included noise reduction (Gaussian blur/median filtering), resizing images to 64x128, pixel normalization (0-1), grayscale conversion, and encoding text labels using integer indices or one-hot encoding. The character list was updated to accommodate special characters, strengthening the OCR pipeline's accuracy and adaptability.

3.2. Template Matching

3.2.1. ORB Algorithm

The ORB algorithm is an efficient feature detection and description method used in computer vision. It is built upon two foundational techniques: the FAST (Features from Accelerated Segment Test) keypoint detector and the BRIEF (Binary Robust Independent Elementary Features) descriptor. ORB enhances these methods to address their limitations, particularly their lack of rotational invariance, making them robust and computationally efficient for various applications.

The FAST keypoint detector identifies corners in an image by comparing the intensity of pixels in a circular pattern around a candidate pixel. While FAST is known for its speed, it does not provide orientation information, which ensures consistent feature detection under image rotations. ORB overcomes this limitation by introducing an orientation operator. The orientation is computed using the intensity centroid method, which relies on the spatial distribution of pixel intensities in a keypoint's neighborhood. The moments of intensity distribution are calculated as follows:

$$m_{pq} = \sum_{x,y} x^p \, y^q I(x,y) \quad \text{(Equation 1)}$$

Where $I(x, y)$ is the intensity at pixel (x, y), and p, q determines the order of the moments. From these moments, the centroid coordinates (C_x, C_y) are computed as:

$$C_x = \frac{m_{10}}{m_{00}} \quad , \quad C_y = \frac{m_{01}}{m_{00}} \quad \text{(Equation 2)}$$

Where m_{00} is the total intensity of the patch. The vector from the keypoint O to the centroid C gives the orientation angle:

$$\theta = arctan2\left(C_y - O_y, C_x - O_x\right) \quad \text{(Equation 3)}$$

This angle θ represents the dominant direction of intensity variation around the keypoint, ensuring that the detected keypoints remain rotation-invariant.

The BRIEF descriptor encodes the local appearance of an image patch as a binary string by performing pairwise intensity comparisons. For a smoothed image patch p, each binary test is defined as:

$$\tau(p; x, y) = \begin{cases} 1, if\ I(x) < I(y) \\ 0, otherwise \end{cases} \quad \text{(Equation 4)}$$

Where $I(x)$ and $I(y)$ are the intensities at pixel locations x and y, respectively. BRIEF combines multiple such tests into a descriptor. However, BRIEF is not robust to rotations, as the tests are fixed relative to the image.

To address this, ORB introduces rBRIEF (rotated BRIEF), which aligns the binary tests with the keypoint's orientation θ. The locations of the pairs are rotated using a rotation matrix:

$$S_\theta = R_\theta S \quad \text{(Equation 5)}$$

$$R_\theta = \begin{bmatrix} cos\,\theta & -sin\,\theta \\ sin\,\theta & cos\,\theta \end{bmatrix} \quad \text{(Equation 6)}$$

Where S is the original set of test coordinates, and S_θ is the rotated set. This alignment ensures that the descriptor remains consistent, regardless of the image's rotation (Rublee, et al., 2011).

Combining oFAST (FAST with orientation) and rBRIEF enables ORB to achieve rotation invariance. By aligning the keypoints and descriptors to their dominant orientations, ORB ensures that features are detected and described consistently across rotated images. This robustness is essential for applications such as object recognition, image stitching, and visual tracking.

In addition to being rotation-invariant, ORB is computationally efficient, making it suitable for real-time applications. It delivers performance that is either comparable to or exceeds that of traditional methods, such as SIFT (Scale-Invariant Feature Transform) and SURF (Speeded-Up Robust Features), while also being significantly faster. Furthermore, ORB is free of licensing restrictions, facilitating its widespread use in open-source and commercial projects.

3.2.2. FLANN Algorithm

Fast Library for Approximate Nearest Neighbors (FLANN) is used to efficiently match features between the uploaded driving license image and the template image. FLANN is designed for fast and approximate nearest-neighbour searches in high-dimensional spaces, making it highly suitable for comparing feature descriptors generated by algorithms like ORB (Oriented FAST and Rotated BRIEF). This implementation enhances the accuracy and speed of template matching, ensuring that only valid driving licenses are processed for further verification.

FLANN organizes descriptors into hash tables using Locality Sensitive Hashing (LSH). Each descriptor is hashed into buckets based on specific hash functions, enabling similar descriptors to reside in the same or nearby buckets. In this project, a hash length of 12 was chosen to strike a balance between computational efficiency and matching accuracy. During the search process, a specific number of candidates are examined for each query descriptor to ensure reliable and fast matching.

FLANN uses the k-Nearest Neighbors (k-NN) algorithm to compare descriptors, retrieving the two closest matches for each descriptor in the template image from the descriptors of the uploaded image. The distances of these neighbours are evaluated using Lowe's ratio test.
Lowe's ratio test filters feature matches by comparing the distances of the two nearest neighbours for each descriptor. A match is accepted if the ratio of the closest to the second-closest distance is below a set threshold, ensuring distinct and reliable matches while reducing false positives.

3.3. Face Detection and Recognition

3.3.1. Multi-Task Cascaded Convolutional Networks (MTCNN)

MTCNN is a deep learning-based approach for face detection and alignment, ensuring input images are pre-processed correctly before feature extraction (Zhang, et al., 2016). It is used in this project to detect faces and align them for better consistency before passing them to the FaceNet model.

It is a three-stage deep learning model used for face detection and alignment. It consists of the following stages, where each stage progressively improves face localization accuracy while performing classification, bounding box regression, and facial landmark detection simultaneously.

1. **P-Net:** A fully convolutional network that generates candidate face regions using feature maps and bounding box regression.
2. **R-Net:** A CNN that refines the candidate regions by rejecting false positives and improving bounding box accuracy.
3. **O-Net:** A deeper CNN that further refines face bounding boxes and detects five facial landmarks (eyes, nose, mouth corners) for alignment.

3.3.2. FaceNet with Inception-Resnet-v1

FaceNet is a powerful face recognition system that directly maps face images into a compact Euclidean space, where the distances between points represent a measure of face similarity. This mapping allows for efficient face verification, recognition, and clustering using standard techniques (Schroff, et al., 2015).

It uses the Inception-ResNet-v1 architecture as its backbone model. This architecture combines Inception modules for multi-scale feature extraction and Residual blocks to enable deeper learning without performance degradation. The network outputs a 512-dimensional embedding vector that represents the identity of the face.

FaceNet is trained using Triplet Loss, which ensures that embeddings of the same person are closer together while pushing apart embeddings of different individuals. The Triplet Loss is defined as:

$$L = \sum max(|f(A) - f(P)|^2 - |f(A) - f(N)|^2 + \alpha, 0) \quad \text{(Equation 7)}$$

f(x): embedding function
A: anchor image
P: positive image (same identity),
N: negative image (different identity),
A: margin that ensures a sufficient gap between positive and negative pairs.

Once embeddings are obtained, the similarity between two faces is measured using Euclidean distance:

$$d(emb_1, emb_2) = \sum_{i=1}^{n} (emb_{1i} - emb_{2i})^2 \quad \text{(Equation 8)}$$

emb_1 and emb_2 are the embeddings of the two faces being compared.

n is the number of dimensions of the embeddings.

If the distance *d* is smaller than a threshold, the faces are considered the same person. If it is greater than or equal to the threshold, they are considered different individuals.

3.3.3. MTCNN and FaceNet Integration

The integration of MTCNN for face detection and FaceNet for face recognition enables a seamless pipeline for detecting and verifying faces. MTCNN first detects and aligns the faces by progressively refining the face regions through its three-stage architecture. Once a face is localized and aligned, it is passed to the Inception-ResNet-v1 backbone of FaceNet, which generates a 512-dimensional embedding representing the face's identity.

The embeddings produced by FaceNet are then compared using Euclidean distance to assess the similarity between faces. If the distance is below a set threshold, the faces are classified as belonging to the same person; otherwise, they are considered different individuals. This integration ensures accurate and efficient face verification by leveraging the strengths of both models.

3.4. Liveliness Detection

The liveliness detection module distinguishes between real and fake facial inputs by analyzing natural human behaviors using computer vision techniques. It detects facial features, assesses image clarity, and tracks eye movements for blinks and head motions using OpenCV. To determine liveliness, we use specific thresholds: an Eye Aspect Ratio (EAR) below 0.25 for at least two consecutive frames indicates a blink, while a total facial movement exceeding 50 pixels across 10 frames confirms natural motion. Additionally, eye movement variance greater than 2.0 ensures realistic gaze shifts. These combined criteria, evaluated over a five-second verification period, ensure that the detected face belongs to a live person, enhancing security and preventing spoofing attacks.

3.5. Algorithm Description

3.5.1. Convolutional Neural Network

A convolutional neural network is a unique deep neural network model. It is mainly reflected in two aspects. On the one hand, the connections between its neurons are not fully connected. On the other hand, in the same layer, some nerves share connection weights between the elements. It is precisely because of these two unique properties that the number of parameters is reduced, which in turn dramatically reduces the complexity of the network model (Wang & Hou, 2020).

Convolutional neural networks usually contain the following layers:

1. **Convolutional Layer**
   The primary function of the network's core component is to extract distinct characteristics from the input. The features extracted by shallow networks are often low-level, such as edges, lines, and corners. As the number of network layers increases, the network can extract increasingly complex features.

$$X^l_j = f\left(\sum_{i \in M_j} X_i^{l-1} * k_{ij}^l + b_j^l\right) \quad \text{(Equation 9)}$$

   The above equation is the convolution calculation formula. l represents the current layer in the neural network; X^l_j is output of the j[th] feature map in layer l, capturing specific patterns; i is the index iterating over convolution regions (receptive fields); X_i^{l-1} is the input data from the previous layer; k_{ij}^l is the convolutional kernel (filter) for the j[th] feature map in layer l; b_j^l is the bias term associated with the j[th] feature map; f is the activation function applied to the convolution result; M_j is the set of convolution regions for the j[th] feature map.
2. **Pooling Layer**
   After the convolutional layer, features with large dimensions were obtained, divided into several regions, and the maximum or average value was taken to obtain new features with smaller dimensions.
3. **Activation Function**
   Each layer of the neural network only applies linear transformations, resulting in a linear model that lacks sufficient expressive power. By adding an activation function and incorporating non-linear factors, the model achieved better performance in solving complex problems. In this regard, we incorporated the Rectified Linear Unit (ReLU) activation function to introduce non-linearity and enhance the network's ability to capture complex patterns.

3.5.2. Long Short-Term Memory (LSTM) Algorithm

LSTM network is comprised of different memory blocks called cells. These are used to solve the long-term storage problem. The repeating cell of LSTM has four Neural Network Layers. The Cell state of LSTM allows information to pass through without being changed. The neural network layer of LSTM consists of a sigmoid activation function and a point multiplication. The output of sigmoid activation function varies from 0 to 1, where a '0' state indicates that no information is passed and '1' state indicates that complete information is passed through. The figure below shows the block diagram of LSTM network. These networks do not allow any information to be manipulated in the cell state. LSTM cell has three gates: Input gate, Forget gate, and Output gate (Ujawla & Sumathi, 2019).

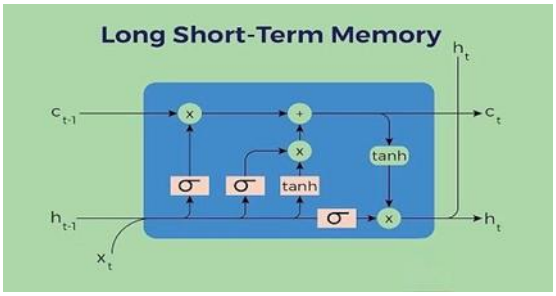

Figure 3. LSTM Architecture

1. **Forget Gate:** As the name indicates, the forget gate decides whether the information should be kept or thrown away. It also shows the proportion of data to be maintained. The logistic sigmoid function is the most widely used activation function in LSTM networks. The output of the previous hidden state and information from the current input is passed through the activation function. The output of the activation function lies between 0 and 1. The value closer to 1 indicates to keep and the value close to zero indicates to forget the information.
2. **Input Gate:** The input gate is responsible for adding information to the cell state. We pass the previous output state in the input gate and present state input into a sigmoid function. The sigmoid function squishes the values between 0 and 1. The output of the sigmoid function is passed through the tanh function. The tanh function outputs a vector containing all possible values that squinches outputs from -1 to +1. The output of the sigmoid gate is multiplied by the output of the tan h function, and the resulting value is added to the cell state. Once this three-step process is complete, it ensures that only essential information, not redundant data, is added to the cell state.
3. **Output Gate:** The output gate is responsible for selecting useful information from the current cell state and presenting it as the output. The functioning of an output gate can again be broken down into three steps. In the first step, the production of the previous hidden and current state input is passed into the sigmoid function. The new production is multiplied by the tanh function and sigmoid output. The output is a new cell state. The new cell state and the new hidden state output are passed to the next step.

Here are the key formulae used within an LSTM cell:

1. **Forget Gate:** The forget gate determines which information should be discarded from the cell state.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad \text{(Equation 10)}$$

f_t: Forget gate activation at time step *t*
σ: Sigmoid function
W_f: Weight matrix for the forget gate
h_{t-1}: Previous hidden state
x_t: Input at the current time step
b_f: Bias term for the forget gate

2. **Input Gate:** The input gate determines which new information should be added to the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{(Equation 11)}$$

i_t: Input gate activation at time step t
W_i: Weight matrix for the input gate
b_i: Bias term for the input gate

3. **Candidate Cell State:** The candidate cell state represents the new information that could be added to the cell state before modulation by the input gate.

$$\widetilde{C_t} = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad \text{(Equation 12)}$$

$\widetilde{C_t}$: Candidate cell state at time step t
$tanh$: Hyperbolic tangent function
W_C: Weight matrix for the candidate cell state
b_C: Bias term for the candidate cell state

4. **Updating the Cell State:** The cell state C_t is updated by combining the previous cell state C_{t-1}, modulated by the forget gate, and the candidate cell state, modulated by the input gate.

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \quad \text{(Equation 13)}$$

C_t: Cell state at time step t
*: Element-wise multiplication

5. **Output Gate:** The output gate determines the next hidden state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(Equation 14)}$$

o_t: Output gate activation at time step *t*
W_o: Weight matrix for the output gate
b_o: Bias term for the output gate

6. **Hidden State:** The hidden state h_t at the current time step is determined by the output gate and the updated cell state.

$$h_t = o_t * tanh(C_t) \quad \text{(Equation 15)}$$

h_t: Hidden state at time step *t*

3.6. Model Description

The model's architecture has been designed to integrate a CNN for feature extraction and an LSTM for sequence prediction, enabling robust text recognition for SmartKYC. The CNN model consisted of two convolutional layers with max-pooling layers, followed by a reshaping layer that prepared the spatial feature maps for sequential processing. This component used 74,496 trainable parameters to efficiently extract spatial features from the input data. The LSTM model employed two bidirectional LSTM layers with 512 units each to capture forward and backward dependencies in the sequential data, comprising 10,522,175 trainable parameters. The final output has been processed through a time-distributed dense layer to generate predictions. This hybrid system has effectively combined CNNs for spatial feature extraction and LSTMs for sequence modelling, demonstrating a robust capability for OCR tasks.

Table 1. Model Summary (CNN) Model: "sequential_1"

Layer (type)	Output Shape	Parameter Count
conv2d_2 (Conv2D)	(None, 64, 128, 64)	640
max_pooling2d_2 (MaxPooling 2D)	(None, 32, 64, 64)	0
conv2d_3 (Conv2D)	(None, 32, 64, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 16, 32, 128)	0
reshape_1(Reshape)	(None, 32, 2048)	0

Total Parameters: 74,496
Trainable params: 74,496
Non-trainable params: 0

Table 2. Model Summary (LSTM) Model: "model_2"

Layer (type)	Output Shape	Parameter Count
input_3 (InputLayer)	(None, 16, 4096)	0
bidirectional_2 (Bidirectional)	(None, 16, 512)	8914944
bidirectional_3 (Bidirectional)	(None, 16, 512)	1574912
time_distributed_1(TimeDistributed)	(None, 16, 68)	32319

Total Parameters: 10,522,175
Trainable params: 10,522,175

Non-trainable params: 0

Total Parameters in CNN Model: 74496
Total Parameters in LSTM Model: 10522175
Combined Model Parameters: 10596671

3.6.1. K-Fold Cross Validation

K-Fold Cross-Validation (KFCV) is a robust evaluation technique that splits your dataset into K equal parts (folds), trains the model K times (each time using K-1 folds for training and 1-fold for validation), and averages the results. This ensures:

i. Unbiased performance estimation (avoids luck-based splits).
ii. Better generalization (tests the model on all data subsets).

To evaluate the robustness and stability of our model, we performed 5-Fold Cross-Validation (KFCV) on a dataset comprising 7,224,612 samples. The dataset was divided into training (80%, ~5,779,690 samples) and validation (20%, ~1,444,922 samples) sets for each fold. The per-fold accuracies ranged from 0.859 to 0.864, with a mean accuracy of 0.8621, closely aligning with the original validation accuracy of 0.8625. The standard deviation of ±0.0018 indicated minimal variance, confirming the model's stability across folds. Furthermore, the F1 score from KFCV was 0.839 (±0.002), which was consistent with the original F1 score of 0.84, with only a negligible difference of 0.001 attributed to random sampling variations. These results demonstrate that our train/validation split was representative and validate the reliability of our model's performance metrics.

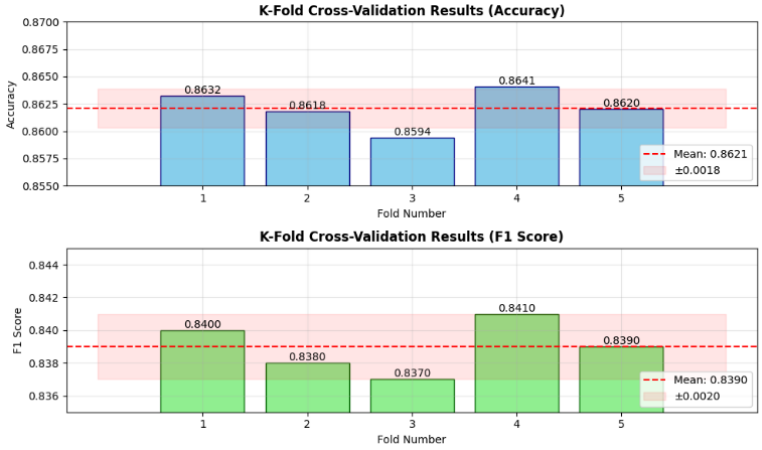

Figure 3: K-Fold Cross Validation Results Visualization

3.6.2. Accuracy

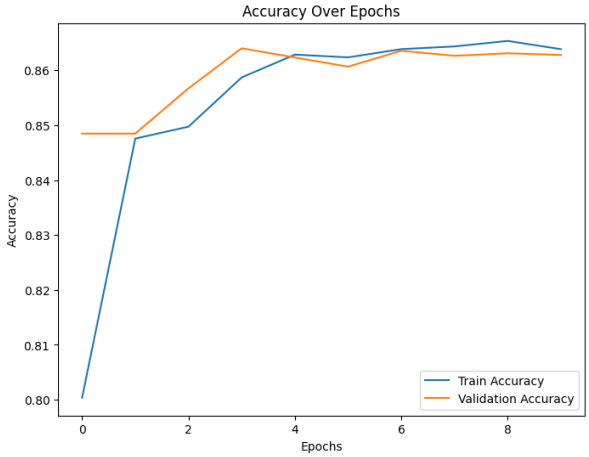

Figure 4. Accuracy Over Epochs

Training Accuracy: 0.8643
Validation Accuracy: 0.8625

The accuracy graph shows a steady improvement in both training and validation accuracy over epochs, indicating effective model learning. The training accuracy (0.8643) and validation accuracy (0.8625) are very close, suggesting that the model generalizes well without significant overfitting. The early convergence of validation accuracy also indicates a stable learning process.

3.6.3. Loss

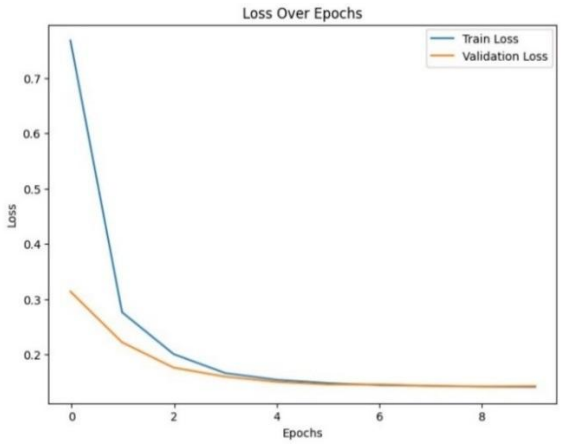

Figure 5. Loss Over Epochs

Training Loss: 0.146
Validation Loss: 0.142

The loss graph shows a steady decrease in both training and validation loss, indicating that the model is learning effectively over epochs. Initially, the train loss is higher but quickly converges with the validation loss, suggesting minimal overfitting. By the final epoch, the training loss (0.146) and validation loss (0.142) are nearly identical, demonstrating that the model generalizes well to unseen data.

3.6.4. F1-Score

Table 3. Performance Metrics of the OCR Model

Metric	**Pre-processed (average)**
F1 Score	0.84
Precision	0.83
Recall	0.86

The obtained performance metrics indicate the effectiveness of the proposed approach. The pre-processed average scores for different classes show that the F1 score is 0.84, highlighting a good balance between precision and recall. The recall value of 0.86 highlights the model's ability to correctly identify the most relevant instances, while the precision score of 0.83 indicates a high level of accuracy in its predictions.

Table 4. Evaluation Metrics of the OCR Model

Metric	Value
CER	0.0312
WER	0.102
LCSE	0.064

The low Character Error Rate (0.0312) and Word Error Rate (0.102) indicate that the OCR model accurately recognizes characters and words with minimal misclassification. The strong performance can be attributed to effective preprocessing techniques, such as noise reduction and normalization, which enhance text clarity. Additionally, the incorporation of synthetic data, including special characters, has likely improved the model's robustness. The Labeling Consistency Score Error (0.064) further supports the model's reliability, demonstrating that it maintains consistency across different samples. These results suggest that the OCR system is well-optimized for extracting text from Nepali driving licenses.

3.6.5. Comparison

To evaluate the effectiveness of our approach, we compare our results with those reported in "Improving the Accuracy of Tesseract 4.0 OCR Engine Using Convolution-Based Preprocessing" (Sporici, et al., 2020). In their study, the authors applied convolution-based preprocessing to enhance the accuracy of the Tesseract 4.0 OCR engine. Their pre-processed results are:

Table 3. Performance Metrics of (Sporici, et al., 2020)

Metric	Pre-processed (average)
F1 Score	0.729
Precision	0.725
Recall	0.734
CER	0.384
WER	0.593
LCSE	24.987

Our approach significantly outperforms (Sporici, et al., 2020) across all key performance metrics. The F1 score is much higher in our method (0.84 vs. 0.729), reflecting a better balance between precision and recall. In terms of precision, our approach achieves 0.83, compared to Tesseract's 0.725, meaning fewer false positives. Our recall score of 0.86 also surpasses Tesseract's 0.734, indicating better detection of true positives. Furthermore, our method shows a dramatic reduction in both Character Error Rate (CER) and Word Error Rate (WER), achieving values of 0.0312 and 0.102, respectively, compared to Tesseract's 0.384 and 0.593. This demonstrates that our approach generates far more accurate text. Additionally, the Levenshtein Character Substitution Error (LCSE) is drastically reduced in our method (0.064 vs. 24.987), indicating fewer character substitutions and closer alignment with the ground truth. Overall, these results highlight the superior performance of our approach in terms of both accuracy and efficiency.

4. Discussion

The SmartKYC system successfully automates the KYC process, enhancing accuracy, efficiency, and security through OCR, face recognition, and liveliness detection. By leveraging LSTM, CNN, and FaceNet, the system ensures reliable document validation, text extraction, and facial matching, thereby significantly reducing the need for manual effort. The system streamlines verification, offering a seamless user experience while ensuring compliance with regulatory requirements. Its robust performance highlights its potential as a scalable and effective solution for digital identity verification across various applications.

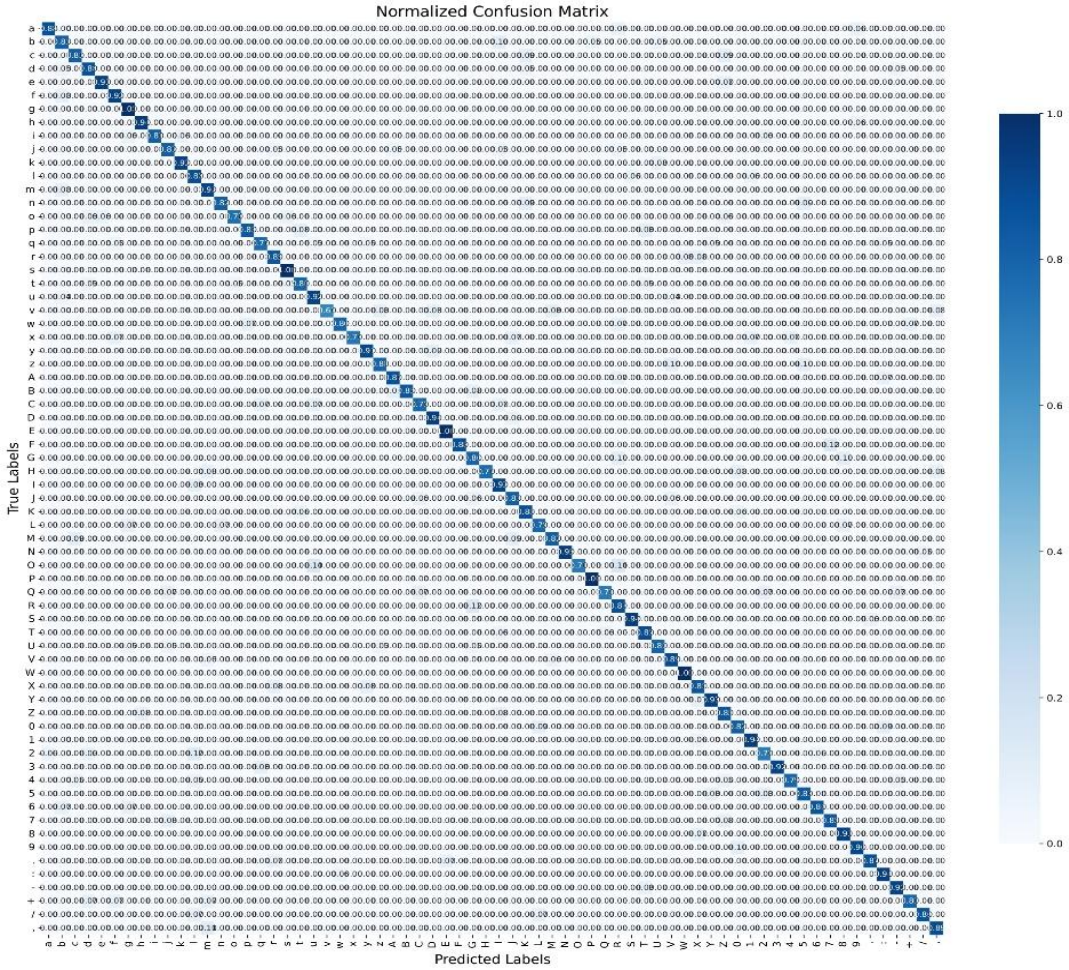

Figure 6. Confusion Matrix

The normalized confusion matrix shows strong model performance, with high values along the diagonal indicating correct predictions for most instances. A value of 1.0 along the diagonal signifies perfect classification. Off-diagonal values are small, indicating minimal misclassification, while the colour intensity, with darker shades representing higher values, reflects this accuracy. The x-axis represents the predicted labels, and the y-axis represents the true labels, with values representing A-Z, a-z, 0-9, and special characters such as . - , + : and /. Overall, the matrix reveals the model's effectiveness with minimal errors, and any misclassifications are represented as small off-diagonal values, offering insights into specific class-level performance.

To enhance model accuracy, several hyperparameters could be optimized, including learning rate (e.g., 0.001 to 0.0001) for better convergence, batch size (e.g., 32, 64, 128) for efficient gradient updates, dropout rate (e.g., 0.3 to 0.5) to mitigate overfitting, and LSTM units (e.g., 128 to 256) for improved context capture. However, due to computational constraints, we prioritized training stability and resource efficiency over exhaustive tuning. Our chosen configuration—Adam optimizer, batch size of 64, and dropout rate of 0.3—offered a balanced trade-off between performance and practicality, validated on our dataset. While further tuning could enhance results, it would demand significant computational overhead without assured benefits. Future work could automate hyperparameter optimization using tools like Optuna, focusing on critical parameters such as learning rate and LSTM depth.

5. Conclusion and Future Enhancements

The SmartKYC system automates the KYC process by integrating OCR, face recognition, and liveliness detection, utilizing LSTM, CNN, and FaceNet to enhance accuracy, efficiency, and security. With a validation accuracy of 86.25%, the system ensures reliable document validation, text extraction, and facial

matching, thereby reducing manual effort and minimizing errors. By streamlining the verification process, it provides a faster and more seamless user experience, improves operational efficiency, and enhances regulatory compliance, making it a scalable and effective solution for digital identity verification.

Future enhancements can improve SmartKYC, including multi-language OCR support for broader document compatibility, AI-powered anomaly detection to identify fraudulent modifications, and expansion to other government-issued IDs like passports and citizenship certificates. Additionally, integration with financial institutions and e-commerce platforms would enhance security in online transactions, at the same time advanced liveliness detection against deepfake attacks and better handling of overlapping text on licenses would further improve accuracy and fraud prevention.

Acknowledgements

We would like to extend our heartfelt appreciation to everyone who supported us throughout the completion of this project. Our deepest gratitude goes to the faculty members of the Department of Electronics and Computer Engineering for their continuous support, guidance, and insightful feedback. We are also thankful to our classmates for their constructive criticism and engaging discussions that helped refine our work. Our sincere thanks to all the lecturers in our department whose continuous guidance was invaluable. Special mention goes to our supervisor, Prof. Dr. Subarna Shakya, for his expert mentorship and direction. This project has been a tremendous learning experience, and we sincerely thank everyone who played a part in making it possible.

References

Arora, K., Bist, A., Prakash, R. & Chaurasia, S., 2020. Custom OCR for Identity Documents: OCRXNet. *Aptisi Transactions On Technopreneurship (ATT),* 06, Volume 2, pp. 112-119.

Bhushan, D. et al., 2024. OCR Based KYC Verification: A Machine Learning Approach. *International Journal of Creative Research Thoughts (IJCRT),* 01.Volume 12.

Kantipudi, M. P., Kumar, S. & Jha, A. K., 2021. Scene Text Recognition Based on Bidirectional LSTM and Deep Neural Network. *Computational Intelligence and Neuroscience,* 23 November, 2021(1), p. 11.

Patel, J. A., 2021. Handwritten And Printed Text Recognition Using Tesseract-OCR. *International Journal of Creative Research Thoughts (IJCRT),* 9(9), pp. 69-77.

Rublee, E., Rabaud, V., Konolige, K. & Bradski, G., 2011. *ORB: an efficient alternative to SIFT or SURF.* s.l., IEEE, p. 2564–2571.

Schroff, F., Kalenichenko, D. & Philbin, J., 2015. *FaceNet: A Unified Embedding for Face Recognition and Clustering.* s.l., IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Sporici, D., Cusnir, E. & Boiangiu, C.-A., 2020. Improving the Accuracy of Tesseract 4.0 OCR Engine Using Convolution-Based Preprocessing. *Symmetry,* Volume 12.

Ujawla, B. & Sumathi, K., 2019. A Novel Approach Towards Implementation of Optical Character Recognition Using LSTM and Adaptive Classifier. *JNNCE Journal of Engineering & Management (JJEM),* Volume 3.

Wang, H. & Hou, S., 2020. *Facial Expression Recognition Based on the Fusion of CNN and SIFT Features.* s.l., 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC), pp. 190-194.

Zacharias, E., Teuchler, M. & Bernier, B., 2020. Image Processing Based Scene-Text Detection and Recognition with Tesseract. 04.

Zhang, K., Zhang, Z., Li, Z. & Qiao, L., 2016. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *IEEE Signal Processing Letters,* Volume 23, pp. 1499-1503.