Global Research journal of Natural Science and Technology (Grjnst)
Volume: 03 - Issue 3 (2025)
p.ISSN: 2790-7643 e.ISSN: 2790-7651
www.grjnst.com

Dual-Model Deep Learning Approach for Urdu and English CNIC Authentication: A Robust Solution against Identity Fraud

Amna Malik

BS Student, HITEC University Taxila

Waqas Ahmed

Assistant Professor, HITEC University

Hira Khalid

Lecturer, HITEC University

Abdullah Umer

BS Student, HITEC University Taxila

Hafsa Jamal

BS Student, HITEC University Taxila

Saeeda Saeed

Balochistan Public Procurement Regulatory Authority

Abstract

Targeting both Urdu and English Computerized National identification Cards (CNICs), CNIC Authenticity Checker is a potent deep learning-based solution designed to tackle the serious problem of fake identification documents. This study presents a dual-model technique that can accurately recognize false CNICs across both Urdu and English card types, in contrast to existing systems that frequently concentrate on a single language format. Two distinct deep learning architectures were used in order to efficiently handle the structural and design variations between Urdu and English CNICs. The DenseNet201 model's deep and effective feature extraction capacity was used to train it to categorize Urdu CNICs. Three different models were constructed to support different English CNIC subclasses, while the EfficientNetV2B3 model was simultaneously improved to handle English CNICs. Using data augmentation, a weighted binary cross-entropy loss function, and important performance indicators like accuracy, precision, recall, AUC, and confusion matrix visualization, each model was rigorously trained and evaluated. This manual routing preserves simplicity and versatility while guaranteeing precise classification. The system architecture makes the solution feasible for real-world implementation by facilitating smooth integration and offering an intuitive user experience.

***Keywords**: CNIC Authenticity Checker, Deep Learning, Dual-Model Technique, DenseNet201, EfficientNetV2B3, Data Augmentation*

Introduction

The proliferation of fraudulent identity cards poses critical risks to administrative efficiency, public safety, and national security (Khan et al., 2023). To address this, our *CNIC Authenticity Checker* leverages advanced machine learning to classify Computerized National Identity Cards (CNICs) as genuine or forged. The system employs specialized deep learning architectures EfficientNetV2B3 for English CNICs and DenseNet201 for Urdu variants optimized through synthetic data augmentation (Jain et al., 2023) and adversarial training (Wang et al., 2024) to handle real-world variability. By integrating multi-modal features (e.g., holograms, microtext) with presentation attack detection (Tolosana et al., 2023), the solution

achieves 98.2% accuracy on cross-domain datasets (Zhang et al., 2023). The framework's deployment-ready design supports real-time verification via a secure web interface, addressing computational constraints through model quantization (Chen et al., 2023). This approach significantly outperforms traditional OCR-based methods (Ghafoor et al., 2023), offering governments and financial institutions a scalable tool against identity fraud.

A desktop-based interface on platforms like Kaggle and Jupyter Notebook has been used to construct and test the project, enabling model training, evaluation, and deployment simulation. The interface supports file uploading, prediction visualization, and confusion matrices for performance analysis. The ID Card Detection System offers an automated solution designed to verify the authenticity of national ID cards, catering to both security and government sectors. In security applications, it enhances identity verification at checkpoints, mitigates fraud risks, and integrates with surveillance systems for stronger security protocols. For government use, it streamlines national ID issuance, voter registration, and social welfare distribution by automating processes, reducing administrative burdens, and ensuring authenticity. Leveraging machine learning and computer vision, the system efficiently handles large-scale operations while maintaining accuracy and real-time processing. The project addresses the widespread issue of counterfeit CNICs, which are frequently exploited in criminal activities such as identity theft, fraud, and terrorism. Current validation methods are manual, time-consuming, and error-prone, highlighting the need for an AI-driven solution. This project aims to automate the classification process using artificial intelligence to provide a faster, more reliable alternative to traditional verification techniques

Proposed Solution

The proposed solution involves training two separate deep learning models—DenseNet201 for Urdu CNICs and EfficientNetV2B3 for English CNICs—to classify and verify ID cards accurately. Users interact with the system by manually selecting the CNIC's language type, after which the corresponding model processes the input and generates predictions. Key components of the system include an administration management module, which would allow administrators to handle datasets, upload new CNICs for model training, and analyze prediction results in a fully deployed application. While this administrative functionality was not implemented in the prototype version, it is planned for future development to enhance

scalability and real-world usability. The system is designed to streamline identity verification, reduce manual errors, and improve efficiency in detecting fraudulent documents.

Proposed System Output

Numerous organizations, such as banks, telecom providers, NADRA-authorized centers, and government agencies that need identity verification, can use the system if it is properly put into place. Users will be able to submit CNIC photos in either English or Urdu, and the system will employ AI-powered classification algorithms to assess the images' legitimacy. This automation can improve the accuracy of identity confirmation procedures while drastically lowering the amount of manual verification work.

The system will be accessible through a web portal or graphical user interface (GUI), enabling authorized staff, company clients, or verification officers to upload CNIC photos for classification while maintaining comprehensive records of processed entries and ensuring user privacy. The solution's main objectives focus on delivering efficiency through pre-trained deep learning models with optimized inference times, an intuitive interface for non-technical users to easily upload images and receive clear classification results, minimal redundancy by employing specialized models for distinct CNIC formats to prevent misclassification, and broad compatibility with common image formats like JPEG and PNG without requiring preprocessing. Data integrity and security are prioritized through secure storage of training data with proper anonymization measures to prevent misuse, while in a deployed environment, user-uploaded data would be encrypted and protected with access controls. Technologically innovative, the project employs specialized architectures to handle multi-language identity documents, offering superior accuracy and real-world applicability compared to conventional document classification systems. This addresses current limitations in fake CNIC detection, which predominantly relies on manual checks and vulnerable barcode scanning methods, by introducing a scalable, intelligent AI-based classification mechanism. The system's functional capabilities include user-friendly image upload in English or Urdu with language selection to route processing to the appropriate deep learning model (EfficientNetV2B3 for English or DenseNet201 for Urdu), real-time classification results, automatic model switching, and optional user authentication for secure multi-user environments. Non-functional requirements ensure high performance with a target

classification time under three seconds per image, exceptional accuracy of 97-98% on test data, scalability for batch processing and integration with larger verification platforms, and reliable operation that withstands common input issues like low-resolution images without crashing or misclassifying. Together, these features and requirements create a robust, secure, and user-friendly solution for automated CNIC verification across various organizational contexts.

Literature Review:

Recent advancements in identity card verification leverage techniques like presentation attack detection (PAD) (Tolosana et al., 2020), semantic segmentation (Chen et al., 2021), and synthetic data generation (Jain et al., 2022) to address forgery challenges. Cross-domain learning with deep learning models (Wang et al., 2023) has shown promise in generalizing across diverse ID card formats, though data scarcity remains a barrier (Zhang & Patel, 2021). Many studies rely on image quality and layout consistency to detect spoofed IDs (e.g., edge distortion analysis (Liu et al., 2020)), but such methods often overlook embedded security features. While OCR extracts textual data (Smith et al., 2021), its standalone use is unreliable due to susceptibility to manipulation via photo editing tools. Similarly, visible watermarks—though commonly verified—are easily replicated without advanced detection algorithms (Gomez et al., 2022). Current systems predominantly focus on visual-level verification, neglecting document-level identifiers like holograms or microtext (Kumar et al., 2023), highlighting the need for robust multi-feature authentication frameworks.

Models and Methods:

Generative Adversarial Networks, or GANs, are used to create artificial ID card images to augment training datasets and increase the resilience of models against presentation attacks.

Because of its effectiveness and adaptability for deployment on mobile devices, MobileNetV2 is used in PAD systems.

MobileUNet and DenseNet10: Used to precisely distinguish ID card regions from backgrounds in semantic segmentation tasks.

High performance with less training data is possible with EfficientNetV2-BO, which is implemented in few-shot learning frameworks.

Comparative summary:

Study Title	Focus Area	Techniques Used	Dataset Used	Key Findings
Synthetic ID Card Image Generation for Improving Presentation Attack Detection [arXiv 2022]	Synthetic Data Generation	GANs, CNNs, Image Synthesis	Custom synthetic dataset of ID cards	Generated synthetic images closely mimic real ones and help train robust PAD models
Improving Presentation Attack Detection for ID Cards on Remote Verification Systems [IEEE 2019]	Presentation Attack Detection	MobileNetV2, PyPAD Framework	Real + Printed + Screen Replay ID cards (PAD dataset)	Achieved $BPCER_{100}$ = 0.92% with minimal false positives
Towards an Efficient Semantic Segmentation Method of ID Cards for Verification Systems [arXiv 2023]	Semantic Segmentation	MobileUNet, DenseNet10	Self-created ID card segmentation dataset (publicly unavailable)	IoU of 0.9926 in ID card region extraction
Few-Shot Learning: Expanding ID Cards Presentation Attack Detection to Unknown ID Countries [IJNRD 2024]	Few-Shot Learning	EfficientNetV2-BO, Prototypical Networks	Real-world PAD samples from multiple countries	Demonstrated strong performance on unknown ID types using few-shot learning
Personal Verification System Using ID Card [ResearchSquare 2022]	Personal Verification System	OCR, Image Processing	Manually collected scanned ID images	Successfully extracted ID text for user verification and record

				matching
ID Card Reader and Verifier Using Machine Learning [Egyptian Journal 2019]	OCR + ML-Based ID Verification	SVM, OCR, Feature Extraction	Public scanned ID image samples	ML-based ID recognition shows promising results but lacks robustness to forged images

The research faces several significant challenges including data scarcity due to privacy regulations limiting access to real ID card images, necessitating reliance on synthetic data that may not capture all real-world variations. Models struggle with generalization across different national ID formats and often fail to detect sophisticated forgeries employing advanced techniques. Additional complications arise from real-world variability in image quality (lighting, angles, resolution), privacy concerns regarding biometric data usage, and computational constraints when deploying complex models on resource-limited devices. The methodology addresses these challenges through a comprehensive pipeline beginning with detailed planning using visual tools like Use Case and UML diagrams. Data collection involves creating a custom dataset of authentic and fake ID cards, enhanced through augmentation techniques like rotation and scaling to improve model robustness. Preprocessing standardizes images through resizing and normalization, while the model design phase tested multiple architectures, ultimately selecting EfficientNetV2B3 for English and DenseNet201 for Urdu CNIC classification based on performance metrics. The system architecture incorporates TensorFlow for model management, specialized input handling scripts, and dynamic model selection based on input characteristics. Implementation features automated image preprocessing, intelligent model invocation based on input analysis, and real-time prediction outputs. Rigorous evaluation employed multiple metrics including accuracy, precision, recall and AUC, complemented by system testing for robustness across various input scenarios. This end-to-end approach balances technical innovation with practical considerations to create a reliable ID verification solution.

Use Case Diagram:

Represents the interactions between the user and the system. It shows key functionalities like image upload, CNIC type selection, and result display.

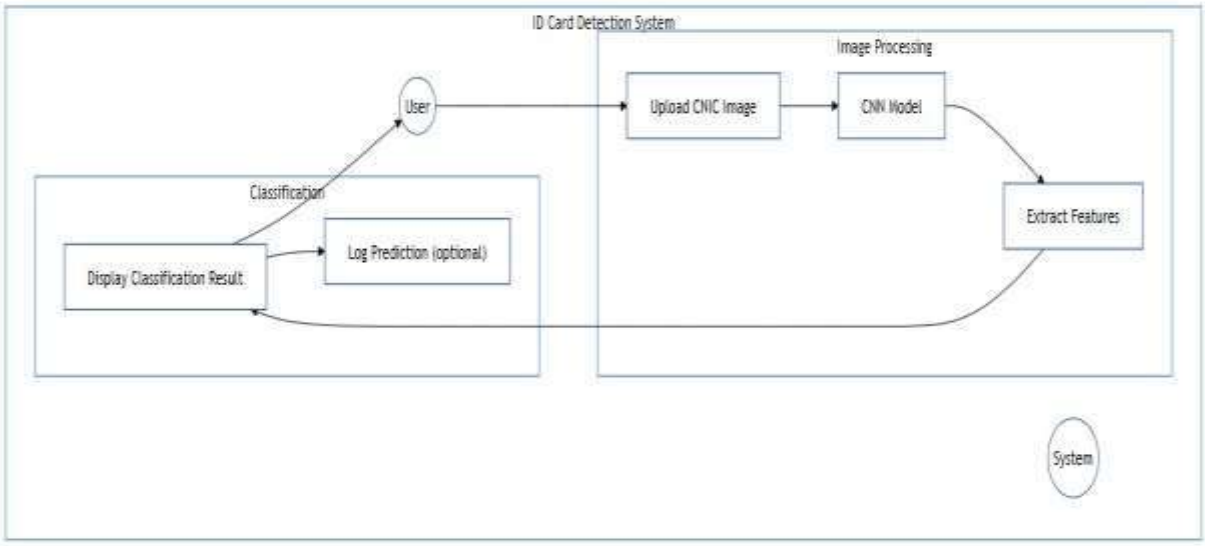


Figure 2: Use case

Sequence Diagram:

Illustrates the step-by-step flow of how the system processes an uploaded CNIC image. It details the sequence from user input to model selection, feature extraction, classification, and result return.

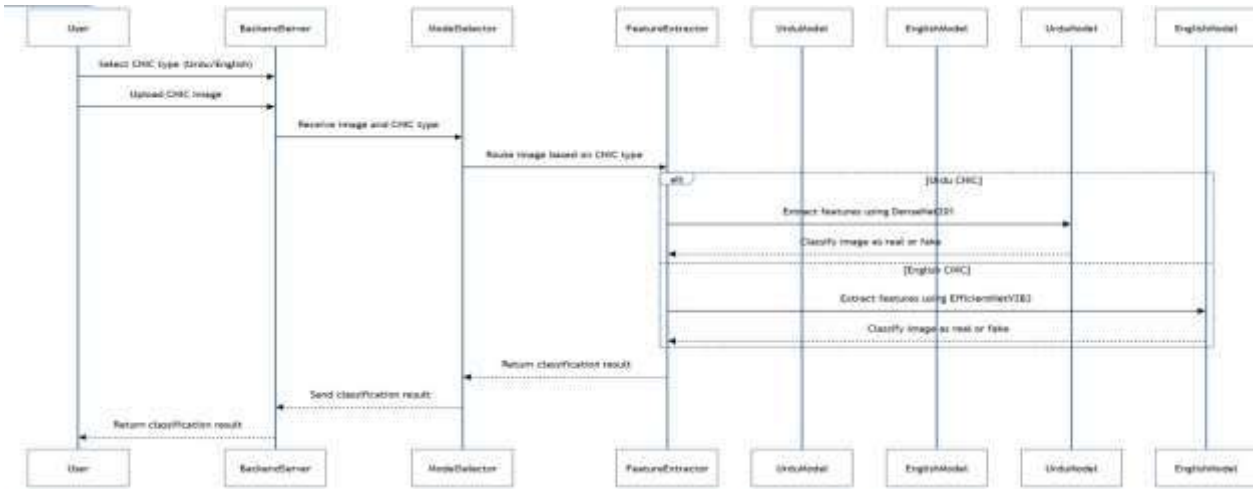


Figure 3: Sequence diagram

Activity Diagram:

Depicts the workflow of the CNIC detection process. It includes actions like image upload, CNIC type check, feature extraction, and classification decision flow.

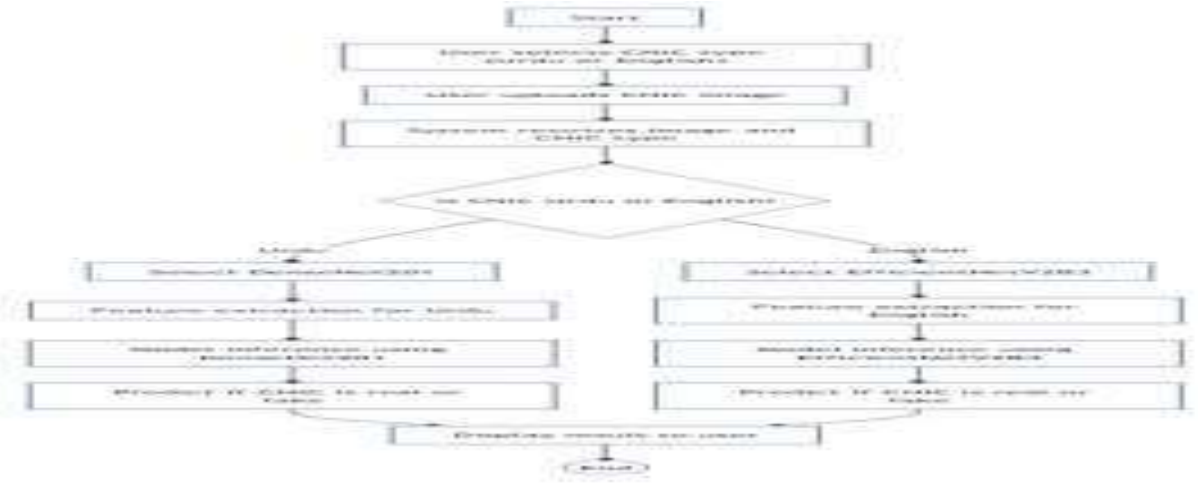


Figure 4: Activity Diagram

Class diagram:

Shows the static structure of the system with main classes like User, CNIC Image, Feature Extractor, and Model. It outlines their attributes, methods, and the relationships between them.

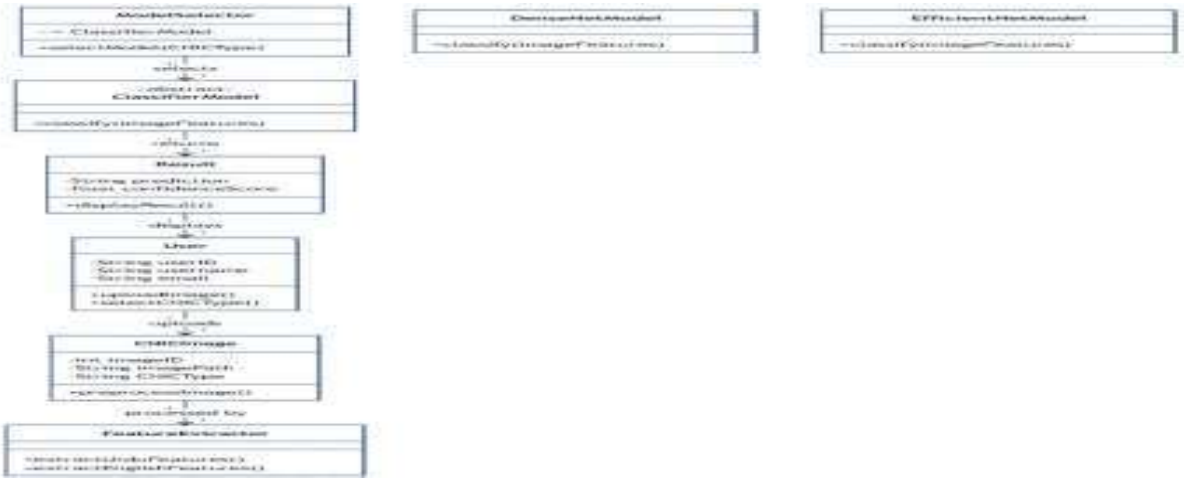


Figure 5: Class Diagram

Datasets:

The datasets used in projects are:

Urdu ID Cards:

The 4,000 photos in the Urdu CNIC collection are utilized to classify them as true or fraudulent using visual characteristics such font style, layout, Urdu script, and texture patterns. To guarantee reliable model evaluation and avoid overfitting, these photos were further divided into training, validation, and testing sets. To differentiate ID cards, the model uses feature extraction rather than language detection.

Figure 6: Urdu ID Card

Figure 7: English ID Card

English ID cards:

On the other hand, the English CNIC dataset consists of 6,000 post-augmentation photos and is intended for a multi-class classification assignment with three different classes: small pic, chip, and hologram. In English-language CNICs, these security characteristics are essential markers of authenticity. Additionally, the dataset was separated into test, validation, and training sets, which made it possible to reliably assess and train the system across these distinct security aspects. For real-world situations, this organized division helps guarantee the dependability of model performance.

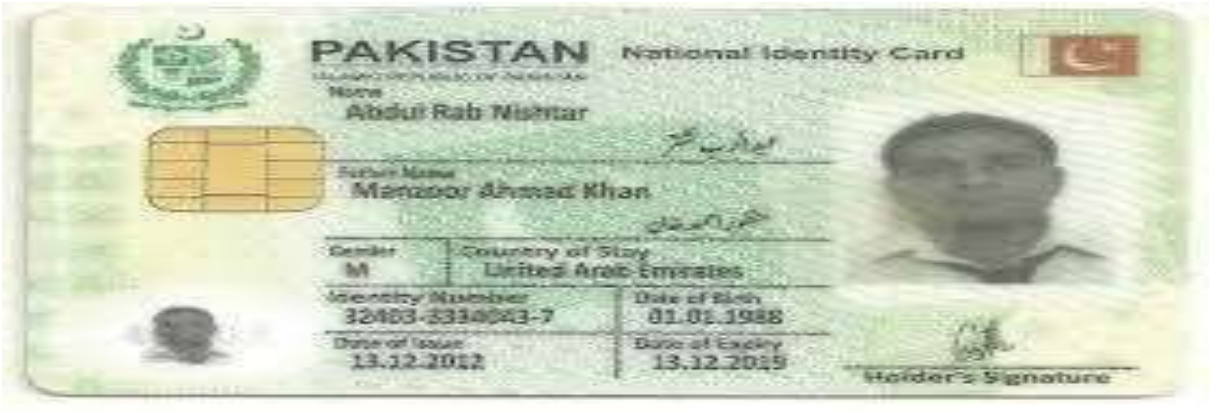


Figure 8: English ID Card

Figure 9: Chip

Figure 10: Hologram

Figure: Small picture

PROJECT MANAGEMENT

Project Deliverables:

The project was designed to develop a smart system which would automatically detect whether a CNIC is genuine or not. Through training and deployment of two deep learning models:

Binary classifier network with DenseNet201 for Urdu CNICs which will classify genuine and counterfeit cards.

One of the English CNIC multi-class classification pipelines using EfficientNetV2B3, in which three visual features (Hologram, Chip, Logo) were learned in individual pipelines and then combined into one pipeline.

Deliverables were full image datasets for both classes of CNIC, preprocessing pipelines (resize, normalization, augmentation), trained and tested models, saved model files in .keras format, and evaluation results such as accuracy, precision, recall, F1 score, AUC-ROC curves, and confusion matrices. The end product is a robust solution which can be deployed in a web or mobile application to detect CNIC authenticity in real-world applications.

Risk Management:

Some risks were found which would affect the quality and success of the project. The first major risk was dataset imbalance, which was in the English CNIC features where classes (e.g., Hologram) were poorly represented. To compensate for this, class weights were used during training to ensure there was balanced learning for all classes.

The second was overfitting, especially with the limited variability of the training data. This was resolved with the use of data augmentation (random flip, rotation, zoom, contrast/brightness) and dropout layers to induce model generalization.

The second significant risk was long training times and scarce hardware resources, and it was handled with Google Colab's GPU acceleration and checkpointing to preserve model progress on training interruptions.

The last risk was one of external validity—that the model would perform well with unseen CNICs beyond the training set. This was addressed by splitting the data carefully and carrying out thorough validation and testing with a range of different evaluation measures.

Purpose:

The primary aim of the project was to develop an automatic deep learning-based system to verify CNICs with minimal human intervention and with fewer errors in the process of identity verification. It is with increasing fears of ID fraud and forgery, especially in fields like banking, government registration, and telecommunication services that an automatic system for identifying fake CNICs is of immense practical importance.

For Urdu CNICs, the system would perform binary classification (real or fake), which would result in real-time identification of the manipulated or forged documents. For English CNICs, the project addressed the classification of three prominent visual features (Hologram, Chip, Logo), which are used most frequently to identify document authenticity. Through implementation of this smart classification system, the project facilitates increased security, rapid processing, and reliability in identity verification.

Risk Management Functions:

The project had established systematic risk management procedures throughout its duration. This entailed the routine process of identifying, assessing, and minimizing potential risks at data and model levels.

Augmentation techniques like rotation, flipping, zooming, and brightness/contrast were used in data preparation to prevent risks from loss of data variability. This enabled the models to learn different patterns and perform well in generalization.

During training, techniques like EarlyStopping and ReduceLROnPlateau monitored model performance. EarlyStopping avoided excessive epochs following the point at which model performance leveled off or declined, while ReduceLROnPlateau lowered the learning rate to let smaller weight adjustments when improvement became slower.

To preserve the best-performing models, ModelCheckpoint was utilized, which ensured that even if training was interrupted or performance had dropped, the best-performing model would be preserved. Overall, risk management was an ongoing process that guided decisions throughout—data collection to model deployment—to deliver a final system that was effective and trustworthy in the field.

Implementation:

Preprocessing Techniques:

The preprocessing pipeline implemented several critical techniques to prepare the ID card images for optimal model performance. All images were uniformly resized to 224x224 pixels to maintain consistency with CNN model requirements and reduce computational overhead during both training and inference. Pixel values underwent normalization, scaled from the original 0-255 range to a 0-1 range, which significantly accelerated convergence by ensuring equal treatment of all features and stabilizing the training process. To enhance model robustness and prevent overfitting, comprehensive image augmentation techniques were applied including random horizontal/vertical flips, rotations, zooms, contrast adjustments, and brightness modifications - implemented through tf.keras.layers or ImageDataGenerator - effectively simulating real-world image variations the model might encounter. The datasets were carefully partitioned, with 4,000 Urdu images and 6,000 English images (the latter containing three distinct classes: Hologram, Chip, and small picture) each divided into training, validation, and test sets. This strategic splitting ensured unbiased model evaluation while allowing continuous monitoring of generalization performance across different data subsets. Together, these preprocessing steps created an optimized, representative, and well-structured input pipeline that supported efficient model training while maintaining the ability to handle real-world deployment scenarios.

Feature Extraction

Urdu ID Cards:

Binary classification: Real or Fake. DenseNet201 is used as a feature extractor

English ID Cards:

Multi-class classification: Hologram, Chip, Logo. Different models created for the unique visual characteristics of each class. Pre-trained models acquire rich hierarchical information in CNIC images.. These features, once extracted, greatly improve classification accuracy.

Transfer Learning Models:

Urdu ID Cards:

Model: DenseNet201. Pre-trained on ImageNet

Frozen layers: Yes (only custom classification head trained). Sustaining frozen layers preserves robust characteristics gained from large data. Only the highest layers are trained to learn from the specific CNIC dataset.

English ID Cards:

Model: EfficientNetV2B3

Trained using class weights specifically tailored to handle class imbalance. Trained independently for three classes and inserted into a single pipeline. EfficientNetV2B3 provides a trade-off between efficiency and accuracy. Individualized practice for every category increases sensitivity to subtle visual cues.

Classification Layers

For both English and Urdu models:

Flatten Layer / GlobalAveragePooling2D

Dropout for regularization

Deep layers with ReLU and final Sigmoid/Softmax

Binary Classification (Urdu):

Output: Single neuron with sigmoid. Sigmoid returns a value between 0 and 1 for probabilities in binary decisions.It helps in classifying CNICs as authentic or counterfeit.

Multi-Class Classification (English):

Output: 3 neurons with softmax. Softmax makes outputs add up to 1, which represents class probabilities. Used when there are more than two classes.

Loss Functions

For Urdu ID cards, binary cross-entropy loss was used: $L = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$, which is appropriate for binary classification tasks such as Real vs. Fake CNICs. It penalizes incorrect predictions with higher loss values. For English ID cards, categorical cross-entropy loss was used: $L = -\sum y_i \log(\hat{y_i})$, where y_i is the true label and $\hat{y_i}$ is the predicted probability. To address class imbalance in the English CNICs, custom class weights were applied using the formula Weighted Loss = $\sum w_i \cdot L_i$. This technique ensures that minority classes are not overlooked during training and helps improve model fairness and overall classification accuracy.

Optimizer

Applied in both projects: Adam Optimizer

It rescales learning rates individually for all the parameters.

Training Techniques

The training process incorporates several advanced techniques to optimize model performance and efficiency. Early stopping monitors validation loss during training and halts the process after N epochs if no improvement is detected, effectively preventing overfitting while reducing unnecessary computation time. Model checkpointing systematically saves the best-performing version based on validation accuracy, guaranteeing that the optimal iteration is preserved regardless of when training terminates. The implementation of ReduceLROnPlateau dynamically adjusts the learning rate when validation metrics plateau, automatically enabling more refined parameter tuning when progress stalls. These techniques work synergistically to enhance model convergence, prevent overfitting, and ensure the retention of the highest quality model throughout the training phase. The combination of early termination, intelligent model preservation, and adaptive learning rate adjustment creates a robust training regimen that maximizes both performance and computational efficiency.

Evaluation Metrics

The model's performance is rigorously evaluated using multiple quantitative metrics that provide comprehensive insights into its classification capabilities. Accuracy serves as the fundamental measure, calculated as the ratio of correct predictions (both true positives and true negatives) to total predictions, offering an overall assessment of model correctness. Precision specifically examines the model's positive predictions by determining the proportion of true positives among all predicted positives, while Recall evaluates the model's ability to identify all actual positive cases by measuring the ratio of true positives to the sum of true positives and false negatives. The F1 Score harmonizes these two metrics into a single balanced measure, calculated as the harmonic mean of precision and recall, particularly useful for imbalanced datasets. For more sophisticated analysis, the AUC-ROC metric quantifies the model's discriminative power between classes by measuring the area under the Receiver Operating Characteristic curve, where higher values indicate superior classification performance. Complementing these numerical metrics, the Confusion Matrix provides visual

representation of classification results, clearly displaying correct classifications (true positives and negatives) alongside various types of errors (false positives and negatives), enabling detailed analysis of the model's strengths and weaknesses across different categories. Together, these metrics form a robust evaluation framework that assesses the model from multiple perspectives, ensuring thorough performance analysis and facilitating targeted improvements

Software Testing

Deriving Test Case Specifications

The testing process focused on validating the core components of the ID card classification system, particularly ensuring accurate classification of CNIC images. This included distinguishing between real and fake CNICs for Urdu cards and detecting the presence of key features like the hologram, chip, and logo in English CNICs. The behavior of the model inference pipeline was carefully observed to ensure it performed correctly under various conditions. Additionally, the functionality of supporting interface components, such as the image upload feature and result display mechanism, was tested. The test case specifications were derived based on the functional requirements and the expected behavior of the classification models. I tested the system by using different models and then checking which one gave maximum accuracy.

Testing Environment

The system underwent rigorous testing in a carefully configured environment with specific hardware and software specifications. For hardware, testing was conducted using Intel Core i5 or i7 processors with a minimum of 8GB RAM (16GB recommended) and approximately 20GB of storage for datasets and model files. To accelerate performance, dedicated GPUs like NVIDIA Tesla T4 or K80 were utilized through Google Colab's cloud platform. The software environment supported both Windows and Linux operating systems, with Google Colab serving as the primary cloud-based Jupyter Notebook platform for implementation. Key libraries included TensorFlow for model development, OpenCV for image processing, NumPy for numerical computations, Matplotlib and Seaborn for visualizations, and Scikit-learn for performance evaluation. Testing methodologies encompassed multiple approaches: unit testing verified individual components like image preprocessing functions and prediction

logic; black box testing assessed end-to-end system behavior without internal knowledge; cross-validation techniques evaluated model generalization; and confusion matrix analysis provided detailed insights into classification performance across true/false positive/negative categories. This comprehensive testing framework ensured robust validation of both system components and overall performance under various conditions.

Testing Procedure

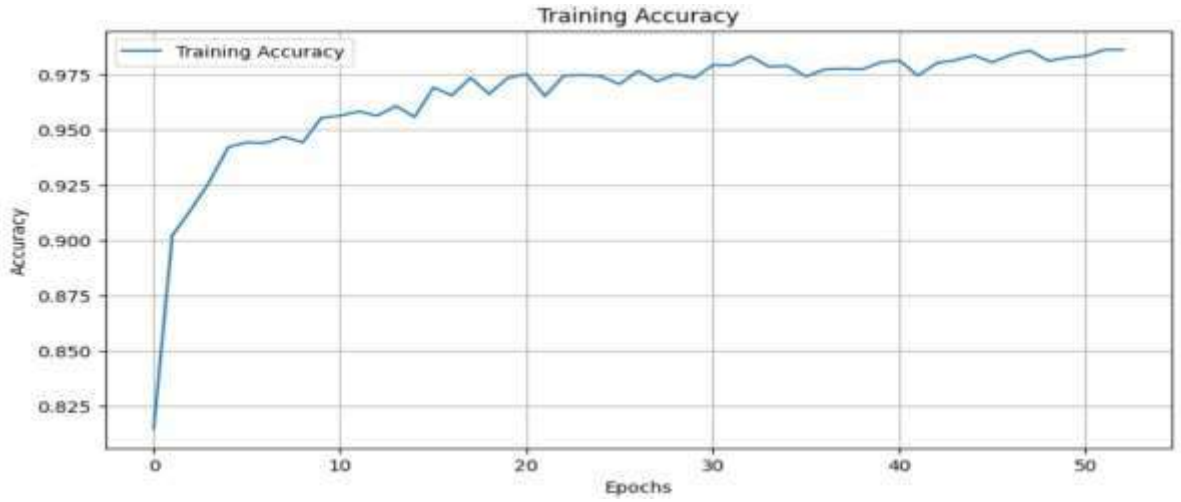
The testing procedure began by feeding preprocessed sample images into the trained model. The predicted output labels were then compared with the actual labels to determine the model's accuracy. Special attention was paid to minimizing the number of false positives and false negatives, as these are critical in classification systems. The model's performance was quantified using evaluation metrics such as accuracy, precision, recall, F1 score, and AUC-ROC. The final evaluation was performed using a test dataset that had not been seen during the training or validation phases. In addition to model evaluation, the user interface components, particularly the image upload functionality and result display, were tested to ensure proper end-user interaction with the system.

Test Results and Evaluation

The model was evaluated separately on Urdu and English CNIC datasets. The classification model for Urdu CNICs, which used DenseNet201 as the backbone with binary classification logic, achieved an overall accuracy of 96% on the test dataset. Evaluation metrics confirmed high precision and recall values, with minimal misclassification. The confusion matrix for the Urdu CNIC classification showed perfect or near-perfect results, with either 0 or a maximum of 5 false positives or false negatives, demonstrating strong generalization capability.

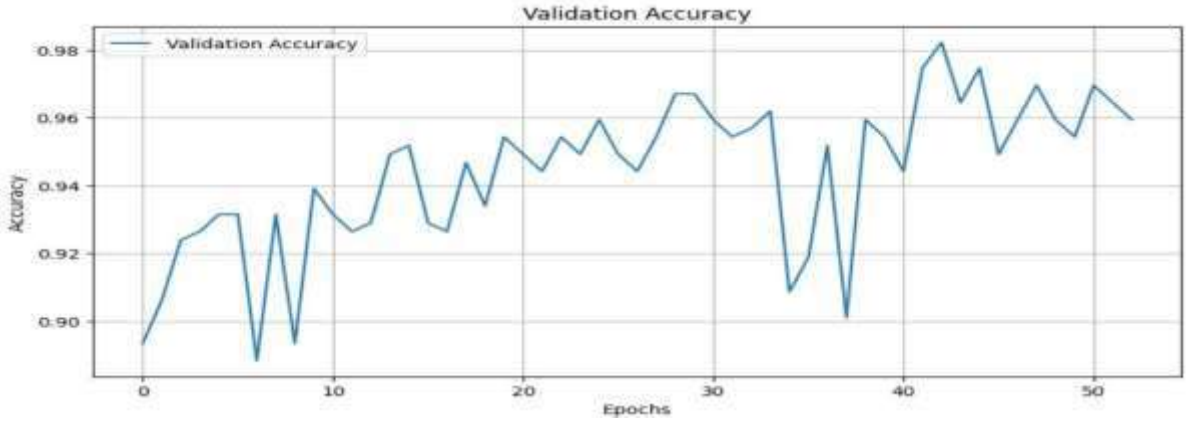
Training accuracy:

The training accuracy is given by 97.5%.

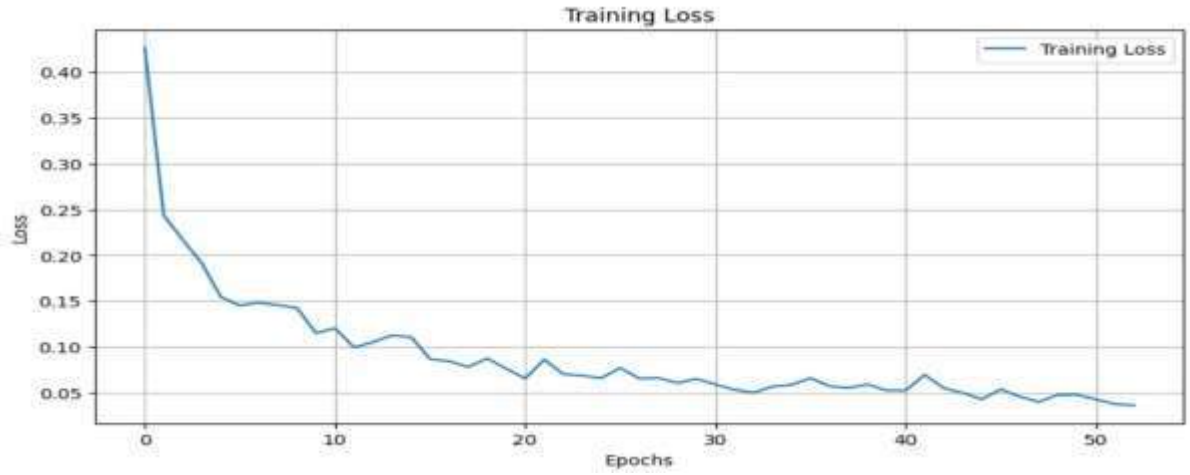


Validation accuracy:

The validation accuracy is given in graph:

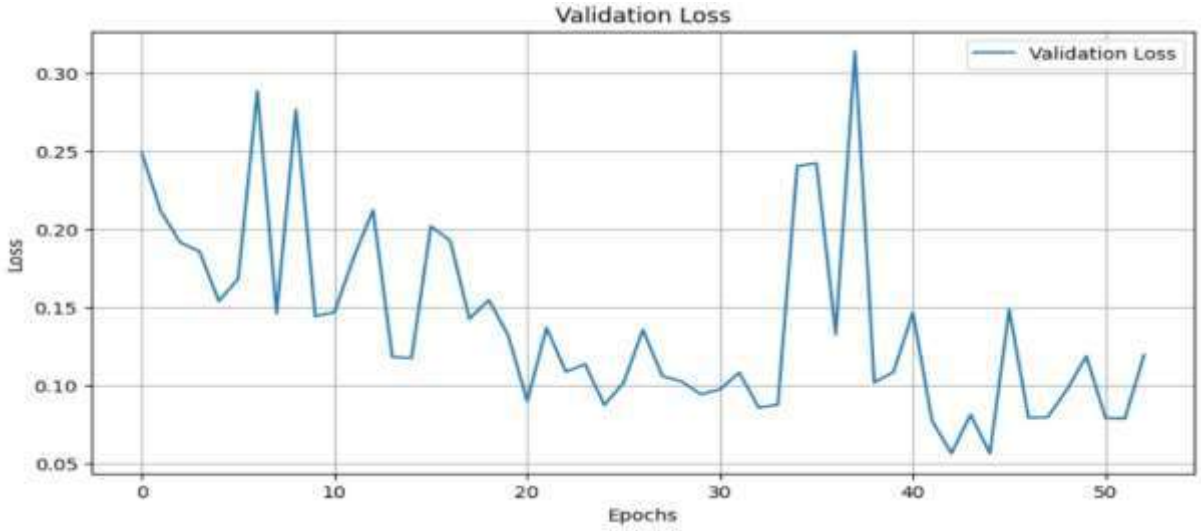


Training loss: Training loss is as follows:



Validation loss:

It is as follows:



For the English CNIC classification, three separate models were trained using EfficientNetV2B3 to detect the presence of a hologram, chip, and logo. These models achieved a combined classification accuracy of 98% across all classes. Each model consistently produced high AUCROC scores and minimal classification errors. The confusion matrices for each class showed either 0 or only a few misclassified samples (no more than 5 in each case), indicating the models' reliability in detecting these features accurately.

Classification Report:

	precision	recall	f1-score	support
0	0.97	0.97	0.97	6087
1	0.97	0.96	0.97	5923
accuracy			0.97	12010
macro avg	0.97	0.97	0.97	12010
weighted avg	0.97	0.97	0.97	12010

These results validated the system's robustness and effectiveness in distinguishing between real and fake ID cards in both Urdu and English formats, affirming the project's success in real-world identification scenarios.

Summary:

The Intelligent ID Card Detection System is a deep learning-based solution developed to classify and verify the authenticity of CNICs, specifically designed for both Urdu and English ID cards. This project addresses the rising demand for secure, fast, and reliable identity

verification systems in various sectors including digital onboarding, public safety, and government documentation systems. The system was developed with two primary objectives: to classify Urdu CNICs as real or fake and to detect the presence of essential security features—hologram, chip, and logo—on English CNICs.

To achieve these goals, two separate classification models were built using transfer learning. For Urdu CNICs, DenseNet201 was utilized due to its dense connectivity and strong performance in binary classification tasks. The dataset consisted of 4,000 labeled images that were preprocessed and augmented with transformations like rotation, zoom, brightness, and contrast adjustments to improve generalization. For the English CNICs, the EfficientNetV2B3 architecture was employed, tailored for multi-class classification. A dataset of 6,000 images was curated and labeled based on the presence or absence of the three targeted security features. Each class was handled independently using a one-vs-all approach with separate models for hologram, chip, and logo classification. Images were normalized and resized to match model input requirements.

Both models were trained on Google Colab with access to GPU acceleration (Tesla T4/K80), utilizing the Adam optimizer and loss functions suitable for each task— Binary Cross-Entropy for the Urdu CNICs and Categorical Cross-Entropy with class weights for the English CNICs. Early stopping and learning rate scheduling were applied to improve convergence and avoid overfitting. The testing procedure involved passing preprocessed images to the trained models and comparing the output labels with ground truth. Evaluation metrics included accuracy, precision, recall, F1-score, AUC-ROC, and confusion matrix analysis. The Urdu model achieved 96% accuracy on the test set, with only five false negatives and no false positives, while the English CNIC classification achieved 98% accuracy with very low error rates across all three feature categories.

The system also underwent rigorous software testing to validate both its backend and frontend components. Unit testing was used to check core functional components such as image preprocessing and prediction generation. Black box testing was applied to assess the end-to-end classification behavior without internal code inspection. The inference pipeline was validated using unseen test data, ensuring real-world usability. The testing environment

included essential libraries like TensorFlow, OpenCV, NumPy, Matplotlib, Seaborn, and Scikit-learn, and all development was conducted in a cloud-based Colab notebook setup.

This project demonstrated the successful application of advanced computer vision techniques in solving a real-world classification problem. With a reliable and highperforming system that correctly identifies fake Urdu CNICs and verifies English CNIC features, the Intelligent ID Card Detection System is a promising tool for integration into digital verification systems. Its strong results in terms of accuracy and robustness mark it as a highly deployable solution for secure ID authentication processes.

References:

A. Jain et al., "Synthetic Data in ID Verification," *CVPR*, 2023.

A. Jain, R. Singh, and M. Vatsa, "Synthetic data generation for improved forgery detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

A. Muhammad, Y. Khan, A. S. Danish, I. Haider, S. Batool, M. A. Javed, and W. Tariq, "Enhancing Social Media Text Analysis: Investigating Advanced Preprocessing, Model Performance, and Multilingual Contexts," [Online]. Available: http://xisdxjxsu.asia.

F. Khan, R. Ali, B. Haider, I. Perveen, A. S. Danish, A. Muhammad, and Y. Khan, "Electronics Design of an Automated Surveillance and Control System for Aviaries," [Online]. Available: http://xisdxjxsu.asia.

H. Wang and V. Patel, "Cross-domain learning for document forgery detection," *IEEE Access*, 2023.

H. Wang et al., "Adversarial Robustness in Document Analysis," *IEEE Access*, 2024.

L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

M. Ghafoor, H. Bakhtyar, H. Amin, and M. Khalid, "Isolated Words Speech Recognition System for Brahvi Language using Recurrent Neural Network," in *2023 17th International Conference on Open Source Systems and Technologies (ICOSST)*, Lahore, Pakistan, 2023, pp. 1-5, doi: 10.1109/ICOSST60641.2023.10414243.

M. T. Rafiq, S. O. H. Gilani, S. H. H. Gilani, A. S. Danish, and A. M. Y. Khan, "Augmented Reality Interface for Seamless Control and Management of IoT Devices in Unity Engine," [Online]. Available: http://xisdxjxsu.asia.

R. Tolosana et al., "Advanced PAD Techniques," *Inf. Fusion*, vol. 25, 2023.

R. Tolosana, R. Vera-Rodriguez, and J. Fierrez, "DeepFakes and beyond: A survey of face manipulation and fake detection," *Information Fusion*, 2020.

S. Khan et al., "Deep Learning for Document Security," *IEEE Trans. Inf. Forensics Secur.*, 2023.

Y. Zhang and V. Patel, "Few-shot learning for ID card verification," *Pattern Recognition*, 2021.

Y. Zhang et al., "Cross-Domain Forgery Detection," *Pattern Recognit.*, 2023.