UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Development of an AI-based Application for Counterfeit Medicine Detection in the Nigerian Drug Market

Glory Nosawaru Edegbe[1*] & Muobonuvie Christabel Tone[2]
Department of Computer Science
Edo State University Uzairue, Nigeria
Email: edegbe.glory@edouniversity.edu.ng[1]; tone20.muobonuvie@edouniversity.edu.ng[2]

Submitted: 6/9/2024. Revised edition: 18/2/2025. Accepted: 19/2/2025. Published online: 27/5/2025
DOI: https://doi.org/10.11113/ijic.v15n1.486

***Abstract*—Counterfeit medicines pose a significant threat to public health worldwide, creating a necessity for detection systems to ensure consumer safety. This research focuses on developing a web-based application using computer vision and Natural Language Processing (NLP) techniques for counterfeit medicine detection. The application integrates logo detection, Optical Character Recognition (OCR), and spell-checking functionalities to validate the authenticity of pharmaceutical products and packaging. By utilizing transfer learning on the YOLO-NAS model and leveraging the Microsoft Common Objects in Context (COCO) dataset, a custom logo detection model was trained to identify approved brands. The OCR functionality utilizes the Google Vision API for accurate text extraction, followed by a Named Entity Recognition model to filter out non-English words and names of places before spell-checking. The custom logo detection model was trained for 200 epochs, achieving an overall mAP of 79.3%, Precision of 85.3%, and Recall of 75.3%. This indicates that the model when integrated into the application is optimized for detecting counterfeit medicine efficiently. The application features a user-friendly interface with three main pages: the home page, the authenticated successfully page, and the failed to authenticate page, providing intuitive navigation and feedback based on authentication results. Equally, the Series of evaluations by the 16 undergraduate students suggests that the application can be reliable and effective in real-world scenarios.**

***Keywords*—Counterfeit Medicine, Logo, Artificial Intelligence (AI), Convolutional Neural Networks (CNNs)**

I. INTRODUCTION

Counterfeit medicines in developed and developing countries, particularly Nigeria, pose a significant public health threat [18]. According to the World Health Organization (WHO), over 280,000 children die annually due to consuming substandard medicines [2]. WHO further stated that nearly 10.5% of the medicines globally are either below par or fake [3]. Counterfeit drugs are defined as products deliberately and fraudulently produced or mislabeled concerning their identity and source, including those with incorrect ingredients, incorrect quantities of active ingredients, or manufactured under poor quality control conditions [4]. The prevalence of counterfeit medicines in Nigeria has reached a concerning level. Current observations show that the circulation of counterfeit drugs may surpass that of authentic drugs. This trend underscores the urgent need for comprehensive strategies to address the issue and protect public health [4].

The consequences of consuming counterfeit medicines can be devastating for individuals. Patients taking these fake drugs may experience treatment failure which is a result of the lack of essential ingredients or incorrect dosages in these counterfeits. This can render the medication ineffective in treating the intended illness. It may worsen health outcomes because they contain contaminants or toxic substances, leading to additional health complications. Counterfeit medicines also lead to antibiotic resistance due to the presence of random antibiotics in these fake drugs. This can contribute to the rise of antibiotic-resistant bacteria and pose a broader public health threat. In severe cases, counterfeit medicines can be fatal [4].

The circulation of counterfeit medicines in Nigeria is amplified by factors such as unregulated open drug markets, lack of counterfeit detection technology, poor local pharmaceutical manufacturing capacity, and porous cross-border monitoring and surveillance systems [5]. Notably, open drug markets in Nigeria, like the Idumota market in Lagos, serve as outlets for substandard and counterfeit medicines, contributing to the challenge of detecting and preventing these harmful products from entering the legitimate supply chain [5].

Despite efforts to combat the issue, the prevalence of counterfeit medicines remains a pressing concern. In 2011, it was reported that 64% of antimalarial circulating in Nigeria was substandard, highlighting the persistent challenge in ensuring the quality and safety of pharmaceutical products in the country [5]. Moreover, the difficulty in detecting counterfeit medicines is increased by the fact that over 70% of drugs in Nigeria are imported, mainly from countries like India and China, which are known sources of counterfeit medicines [5].

The impact of counterfeit medicines on public health is profound, with estimates suggesting that fake medicines account for more than 10% of the global drug market and pose a significant challenge to eradication efforts [5]. The consequences of consuming counterfeit drugs are often hidden in public health statistics, making it challenging to quantify the mortality and morbidity resulting from their consumption [5]. The lack of reliable data on the extent of harm caused by counterfeit medicines underscores the urgent need for robust regulatory measures and enforcement to safeguard public health in Nigeria and other developing countries.

The prevalence of counterfeit medicines is a critical challenge. Addressing this challenge requires a multi-faceted approach involving strong regulatory measures, enhanced surveillance systems, and collaborative efforts between government agencies, pharmaceutical companies, and international organizations to combat the production, distribution, and consumption of counterfeit drugs effectively ([2], [5]).

Statement of the Problem

According to the Food and Drug Administration (FDA), counterfeit medicines account for more than 10% of the world drug market and are difficult to eradicate in both developed and developing nations ([2], [17]). Despite global efforts at eliminating drug counterfeiting, fake medicines still contribute to an increasing percentage of the global drug market ([2], [16]). Between 2001 and 2005, drug regulatory agencies played active roles in decreasing the circulation of counterfeit medicines from 40% to 17%. Yet, the issue remains a major public health and socio-development problem, particularly with essential medicines [15].

Several studies have been undertaken to explore strategies for reducing the prevalence of counterfeit drugs; however, each study has presented its distinct limitations ([6]-[14]).

Research Questions

The following research questions will guide this study:

i. What is the most effective machine-learning algorithm for detecting counterfeit medicines?
ii. How can image processing techniques be used to improve the detection of counterfeit medicines?
iii. What are the important features that distinguish original from counterfeit medicine packaging?

Aim and Objectives

The aim of this study is to develop an AI-based application for counterfeit medicine detection in the Nigerian drug market. While the objectives are to:

i. Design a system to predict the authenticity of a drug using images uploaded by the consumer, achieving at least 85% accuracy in distinguishing genuine from counterfeit medicines.
ii. Train an AI model using Roboflow to recognize the logo of the National Agency for Food and Drug Administration and Control (NAFDAC) approved pharmaceutical product manufacturing companies, ensuring at least 90% recognition accuracy across a dataset of approved logos.
iii. Implement the designed system to process images as input and return a prediction on drug authenticity, with less than 3 seconds and overall system precision of at least 80% in real-world testing.

II. REVIEW OF RELATED WORKS

In recent years, several studies have concentrated on the development and application of computer vision and machine learning techniques for image classification and counterfeit detection. [1] and [6] provide an extensive review of image classification algorithms based on Convolutional Neural Networks (CNNs). Their work highlights the effectiveness of CNNs in image classification. But it also records the lack of discussion on the challenges and real-world deployment issues faced during the implementation of these systems. [7] proposed and developed a deep learning-based solution for logo detection using pre-trained CNN models. Their study demonstrates the potential of pre-trained models for logo detection. It also reveals the limitations, including complex image preprocessing pipelines and poor robustness when handling a large number of logos.

[8] developed an application that the end-users can use to detect fake products, utilizing image and text recognition and classification through machine learning. Their study however acknowledges a major limitation: the vast amount of data required to effectively combat counterfeiting, contrasted with the limited availability of existing data sources. While [9] focused on detecting counterfeit electronics using image processing and machine learning techniques. It involves image segmentation, feature extraction, and classification. Their work emphasizes the need for high-quality images and the potential for misclassification due to variations in lighting conditions and image quality. [10] evaluated several analytical tools to authenticate the primary and secondary packaging of medicinal products. They employed visual comparison, spectrometry, and ink analysis. However, their research indicates that the increasing quality of counterfeits complicates the authentication process of each packaging component.

The work of [11] focused on the challenge of distinguishing between fake and genuine goods through a combination of microscopy and machine learning. They highlight the complexity and time-consuming nature of microscopic analysis, which could become a problem in large-scale detection efforts. While [12] developed a mobile application to identify and classify counterfeit pharmaceutical drugs using Natural Language Processing (NLP) and Optical Character Recognition (OCR). The study indicates that the system's accuracy is heavily dependent on the performance of the

Tesseract OCR engine, which can be influenced by image quality and variations in font styles. Whereas [13] proposed an image-based approach to detect counterfeit medicines and identify key regions of the tablet for classification using a Support Vector Machine (SVM) for heat map generation. They highlight the importance of considering specific tablet characteristics and the potential variations in counterfeit tablet appearances. This could impact the accuracy of their method. [14] tackled the cognitive aspects of human identification confusion with look-alike images, using transfer learning on the YOLOv9 CNN. Their study proposes that while the model offers insights into cognitive counterparts of deep learning solutions, it may not be suitable for contexts with limited resources. They noted that the limited data used in their work might affect the general applicability of their model to all medications. In addition, the model's complexity requires significant computational power.

III. METHODOLOGY

The Rapid application development approach was adopted in the design and implementation. It provides a flexible and adaptable process for how software is designed and built. It equally prioritizes rapid prototyping and feedback and also emphasizes the speed and efficiency of the development process by focusing on delivering software quickly and iteratively. This is done by breaking down the development process into smaller, more manageable parts.

Analysis of the Existing Systems

Several laboratory test-based techniques have been certified by WHO for use in the detection of sophisticated counterfeit medications. They include Melting Point Determination, Test-tube Color Reaction, Thin Layer Chromatography (TLC), Rapid Counterfeit Medicines Detection Method, and Analytical Techniques. These techniques are lab-based and can only be used by experts because they need a deep understanding of various chemical substances. There are also high-tech tools available for spotting fake medications, such as Mobile Authentication Service (MAS), TruScan, and The Black-eye Device. These portable gadgets can quickly ascertain whether a medication is authentic, fake, or of inferior quality. However, it is not accessible to the end customer and is quite expensive.

Other approaches, including the Test Card Method from Veripad and the QR Code-based approach from NeuroTag, integrate machine learning algorithms with hardware devices to detect counterfeit medications. The detection of counterfeit medication may take weeks or months using conventional methods. It is worth noting that only government agencies are equipped to perform the lab-based tests. Other state-of-the-art methods rely on hardware that is not always accessible to the users. Therefore, a system that can detect fake medications without the need for a specialist tool would be useful to the average consumer.

Advantages of Existing Systems

The existing systems for counterfeit medicine detection have several advantages:

i. Increased Speed: Portable gadgets like the Mobile Authentication Service (MAS), TruScan, and The Black-eye Device are fast in determining whether a medication is authentic, fake, or of inferior quality.
ii. Integration with Machine Learning: The integration of machine learning algorithms into the Test Card Method from Veripad and the QR Code-based approach from NeuroTag greatly increases the capability of these systems, making them more effective in identifying counterfeit medications.
iii. Improved Accuracy: Lab-based methods such as Thin Layer Chromatography (TLC) can provide more accurate results compared to manual methods, which are prone to human error.

Disadvantages of the Existing Systems

These existing methods have the following disadvantages:

i. Lab-based techniques like Melting Point Determination, Test-tube Color Reaction, Thin Layer Chromatography (TLC), etc. require specialized equipment and expertise, making them inaccessible to the average consumer.
ii. High-tech portable devices like Mobile Authentication Service (MAS), TruScan, and The Black-eye Device, while faster and more convenient than lab tests, are still quite costly and not readily available to end-users.
iii. The existing methods are time-consuming and can take weeks or months to detect counterfeits.

Analysis of the Proposed System

The proposed system is designed to be a web-based application accessible from any device with an internet connection. The application comprises a deep-learning logo detection model that can be used to detect medicine from approved companies. This is done by analyzing the logo image taken with a mobile phone and performing spell-checks on text extracted from the image of the medicine. The main objective of the proposed System is to alert the general consumer that the medication they are purchasing is not from an approved company. Thus, has a very high probability of being fake.

Architecture of the Proposed System

An application's diagrammatic structure, or application architecture, provides developers with a visual representation of how the program will appear. The suggested system's application architecture is shown in Fig. 3.1.

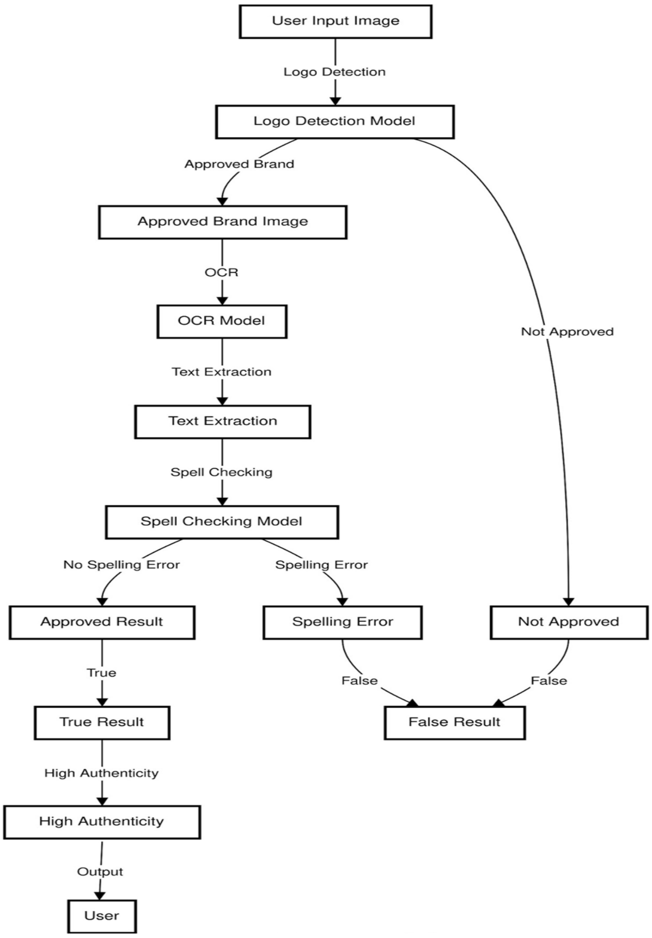


Fig. 3.1. Architecture of proposed system on counterfeit medicine detection

Data Collection Method

For this study, primary data were collected through a web search for pharmaceutical companies approved by Nigeria's National Food and Drug Commission (NAFDAC). As the drugs approved by NAFDAC are more likely to be genuine. Images of brand logos were then collected from their official websites and social media posts.

Data Preparation

Data preparation is a critical step in training object detection models. It involves collecting, annotating, and organizing your data in a way that the model can understand and learn from effectively.

For deep learning models to acquire strong features and perform well in terms of generalizing to new, unseen cases, they need a lot of training data. The quality of the input images directly impacts the model's ability to learn meaningful representations. Blurry, low-resolution or noisy images can hinder the model's learning process. Ensuring consistent image quality, resolution, and format is important. The breakdown of the key processes involved are: Data Collection, Data Annotation, Data Splitting and Data Augmentation.

Model Training

The custom Logo detection model was trained using supervised learning. It was trained with the Microsoft Common Objects in Context (COCO) dataset providing a benchmark for evaluating the periodic improvement of the model. The COCO dataset is the gold standard benchmark for evaluating the performance of state-of-the-art computer vision models. COCO contains over 330,000 images, of which more than 200,000 are labelled, across dozens of categories of objects. COCO is a collaborative project maintained by computer vision professionals from numerous prestigious institutions, including Google, Caltech, and Georgia Tech. The sequence diagram is indicated in Fig. 3.2.

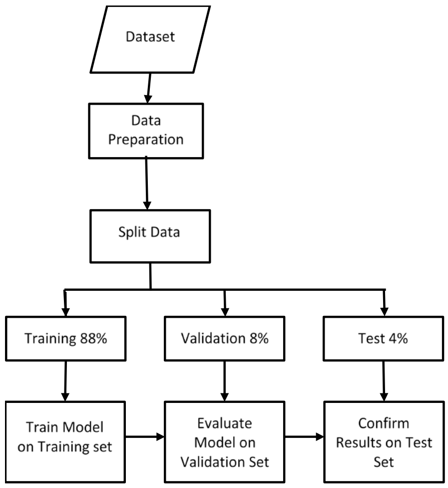


Fig. 3.2. Sequence diagram

System Design

Optical Character Recognition (OCR)

For the Optical Character Recognition (OCR) component of the proposed system, selecting an effective and robust model is crucial for accurately extracting text from images of drug packaging. Once a model is selected, it is crucial to evaluate its performance in real-world scenarios to ensure it meets the application's requirements.

Accuracy Assessment: Measure the OCR model's accuracy in extracting text from the images. This involves comparing the extracted text with the ground truth text and calculating metrics such as character error rate (CER) and word error rate (WER).

Performance Metrics:

i. Character Error Rate (CER): This metric calculates the percentage of characters that are incorrectly recognized. A lower CER indicates better performance.
ii. Word Error Rate (WER): This metric evaluates the percentage of words that are incorrectly recognized. It is particularly useful for assessing the model's ability

to understand context and handle different languages and scripts.

Speed and Efficiency: Evaluate the model's processing speed and computational requirements. For real-time applications, the OCR model needs to be both fast and efficient to ensure seamless user experience.

Error Analysis: Conduct a thorough analysis of the errors made by the OCR model to identify common issues such as the misrecognition of certain characters, confusion between similar-looking characters, and errors in challenging conditions. This analysis helps in fine-tuning the model and improving its accuracy.

Spell-Checking

For the spell-checking module, selecting an appropriate algorithm is essential to handle both English and non-English words effectively. Once the algorithm is selected, the next step is to integrate it with the OCR output. It involves passing the text extracted by the OCR model to the spell-check algorithm for validation. The algorithm checks the spelling of each word and identifies any potential errors. For any words that are flagged as not correct, it implies that the medicine did not pass the validation test.

Application Development

The application is developed with the following components to ensure seamless functionality and user experience:

User Interface (UI)

The User Interface (UI) is designed to be simple and intuitive, allowing users to easily upload images of medicine packaging. Key features of the UI include:

i. Image Upload Mechanism: Users can either drag and drop images or use a file upload button to select images from their devices. The interface supports various image formats to accommodate different types of packaging photos.
ii. Real-Time Feedback: The UI provides immediate feedback to the users, showing the progress of the image upload and processing. This includes status indicators for uploading, processing, and validation stages.
iii. Clear Instructions and Notifications: To guide users through the process, the UI includes clear instructions on how to take and upload pictures. Additionally, notifications inform users about the results of the validation process, such as whether the medicine is likely authentic or counterfeit.
iv. Responsive Design: The UI is designed to be responsive. Ensuring a consistent and user-friendly experience across various devices, including desktops, tablets, and smartphones.

Backend Processing

The backend processing component is a server-side application. It is responsible for handling the core functionalities of image processing, model inference, and validation. Key aspects of the backend processing include:

i. Image Preprocessing: Upon receiving the uploaded image, the backend performs necessary preprocessing steps such as resizing, normalization, and noise reduction to prepare the image for model inference.
ii. Model Inference: The preprocessed image is then passed through the trained machine-learning model. The logo detection model first identifies and verifies the presence of an approved brand logo. If the logo is approved, the Optical Character Recognition (OCR) model extracts text from the back strip of the drug packaging.
iii. Validation: The extracted text is then subjected to a spell-checking algorithm to detect any spelling errors. This step is crucial in identifying potential counterfeit medicines, as genuine packaging typically has accurately spelt text.
iv. Result Compilation and Delivery: Based on the outcomes of the logo detection and text validation processes, the backend compiles the results and determines the authenticity of the medicine. These results are then sent back to the user interface, where users can view them.

Requirements Specification

Functional Requirements Specification

i. Image Upload: Users should be able to upload images of medicine packaging.
ii. Display Results: The application should display the results of the validation checks (authenticity status).
iii. Logo Detection: The system must detect and identify logos on the medicine packaging.
iv. Optical Character Recognition (OCR): The system must extract text from the medicine packaging images.
v. Logo Validation: The application must check if the detected logo is part of an approved list of brands.
vi. Spell-Checking: The system must check the extracted text for spelling errors against a predefined dictionary.
vii. Result Notification: Users should be notified if the medicine is likely authentic or counterfeit based on the validations.

Non-Functional Requirements

i. Response Time: The system should provide validation results within a few seconds of image upload.
ii. Scalability: The system should be able to handle multiple concurrent users without significant performance degradation.
iii. User-Friendly Interface: The UI should be intuitive and easy to navigate.
iv. Accessibility: The application should be accessible to users with disabilities.
v. Accuracy: The system should have a high accuracy rate in logo detection and text recognition.

Technical Requirements

i. Software: The system will work on any computing device with access to the internet.
ii. Internet Connectivity: Reliable internet connection for users to upload images and receive results.
iii. APIs: Secure APIs for communication between the frontend, backend, and any external services (e.g., cloud-based ML models).
iv. Training Data: A comprehensive dataset for training in the logo detection and OCR models, such as the Microsoft COCO dataset.
v. Validation Data: Datasets used to validate the performance of the model.

IV. IMPLEMENTATION

Input and Output Requirements

Input Requirements:

i. User Input Image: An image of the medicine packaging uploaded by the user.
Format: JPEG, PNG, or other common image formats.
Resolution: The application should handle varying resolutions, but a minimum resolution of 300x300 pixels is recommended for accurate detection.
ii. Approved Brand Logos: A dataset of logos from approved pharmaceutical brands used for training the logo detection model.
Format: A collection of labelled images in JPEG, PNG, or other common formats.
iii. Text Data for OCR: Text data extracted from the images using OCR for further processing and validation.
Format: Plain text.

Output Requirements

i. Logo Detection Result: The result of the logo detection process indicates whether the uploaded image contains an approved brand logo.
Format: JSON response with fields indicating the presence and type of logo detected.
ii. Extracted Text: Text extracted from the medicine packaging image.
Format: Plain text.
iii. Spell-Check Result: The result of the spell-check process.
Format: JSON response with fields indicating the correctness of the extracted text.
iv. Authentication Result: The final result of the counterfeit medicine detection process indicates whether the medicine is likely authentic or counterfeit.
Format: JSON response or HTML page indicating "Success" or "Failure".
v. User Feedback: Feedback is provided to the user based on the authentication result.
Format: HTML pages (success or failure) displayed on the web interface, indicating the probability of authenticity and any identified issues.

Hardware and Software Requirements

The following are the hardware requirements of the system:

i. Memory (RAM): At least 8 GB of RAM for handling multiple simultaneous requests and processing large datasets.
ii. Storage: SSD storage with a minimum of 100 GB to store the application, models, and datasets.
iii. Graphics Processing Unit (GPU): Optional but recommended for faster model inference, especially for deep learning models.
iv. Network: High-speed internet connection for handling user requests and accessing external APIs (e.g., Google Vision OCR, Roboflow inference API).
v. Processor: Modern multi-core CPU.

The following are the software requirements of the system:

i. Development OS: Windows, macOS, or Linux.
ii. Python: Version 3.8 or later for developing the application and implementing machine learning models.
iii. Flask: Lightweight web framework for building the application.
iv. OpenCV: For image processing tasks.
v. SpaCy: For Named Entity Recognition (NER) and text processing.
vi. Hunspell: For spell-checking.
vii. Roboflow Inference API: For model inference and processing images.
viii. Google Vision API: For OCR and text extraction.
ix. Integrated Development Environment (IDE): Visual Studio Code, PyCharm, or any preferred IDE for Python development.
x. Version Control: Git and GitHub or GitLab for version control and collaboration.

Overview of the Design

The proposed system is a web application with the following functionalities:

i. Logo detection and identification
ii. Optical Character Recognition and text extraction
iii. Text evaluation and spell-checking

Choice of Programming Language

The programming language adopted to implement this research is Python. Python is a well-suited language for the development of the application due to its versatility in handling various tasks, such as image processing for logo detection using libraries like OpenCV and text extraction with Tesseract. Python's readability and maintainability make it easier to understand and modify code, crucial for long-term maintenance and updates. The extensive Python community provides a wealth of libraries, enabling the use of existing solutions for functionalities like data analysis and statistical modelling. Python's simplicity and ease of use also make it ideal for rapid prototyping, allowing quick validation of the application's core functionalities. Complementing Python, Flask is a lightweight and flexible web application framework that offers core functionalities without overwhelming complexity, making it easy to tailor to our specific needs. Flask's simple learning curve is beneficial for quick development, and its API-friendliness makes it an excellent choice for exposing the application's functionalities to other systems. Additionally, Flask can scale well as the application grows, allowing for the integration of additional libraries and functionalities to handle increased complexity.

Programming Environment

The programming environment adopted for this work is the Visual Studio Code Integrated Development Environment. Visual Studio Code (VS Code) is an open-source code editor developed by Microsoft. It is renowned for its lightweight design, robust performance, extensive features and support for a wide range of programming languages and frameworks. We also utilized Roboflow for building the computer vision model. Roboflow is a comprehensive platform designed to streamline the process of building, deploying, and managing computer vision models. It offers a suite of tools for data collection, annotation, preprocessing, and augmentation, making it easier to create high-quality datasets. Roboflow supports various machine learning frameworks and facilitates model training and deployment with minimal friction. Additionally, Roboflow integration capabilities and collaborative features enhance productivity and accelerate the development of sophisticated visual recognition systems.

Dataset

In this work, we used logos of 14 pharmaceutical companies available in Nigeria: Acino, Benadryl, Benylin, Bliss GVS, Doppel, Drugfield, Evans Baroque, Emzor, Espen, Fidson, GlaxoSmithKline plc (GSK), Novartis, Sanofi, and Swipha. The selection of the companies was based on the following criteria: high public acceptance, mass production of pharmaceutical products, a large variety of pharmaceutical products produced by the company, and availability of high-definition images containing the company logo. In this data set, the number of logos collected for each company ranges from 20 to 40 logos from different sources leading to a total of 322 images. Then, we adopted data augmentation to enlarge the dataset. For data augmentation, we used 7 parameters in the Image Data Generator function including rotation, shear, re-scale, zoom, width shift, height shift, and horizontal flip. After augmentation, the number of total logos increased to 2960. See Figs. 4.1, 4.2 and 4.3.

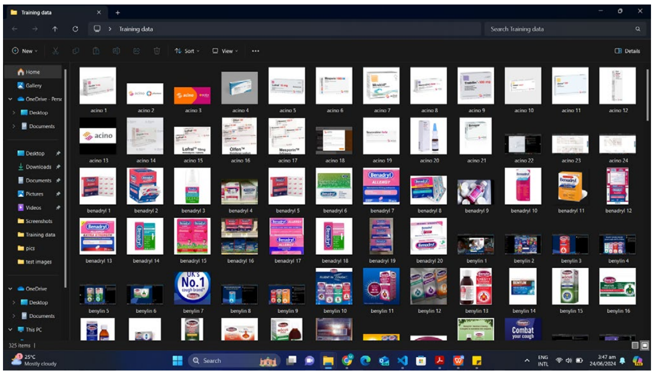


Fig. 4.1. Collected image data

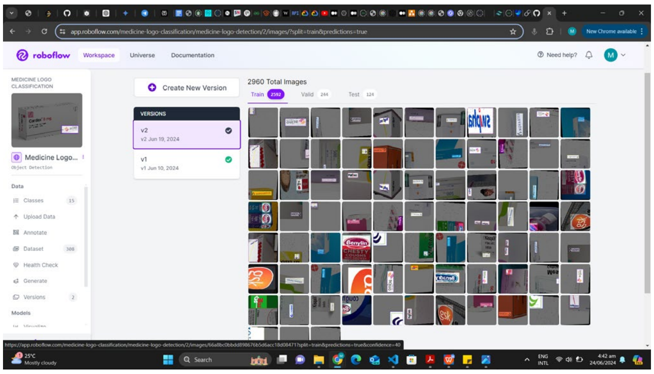


Fig. 4.2. Annotated and split image dataset

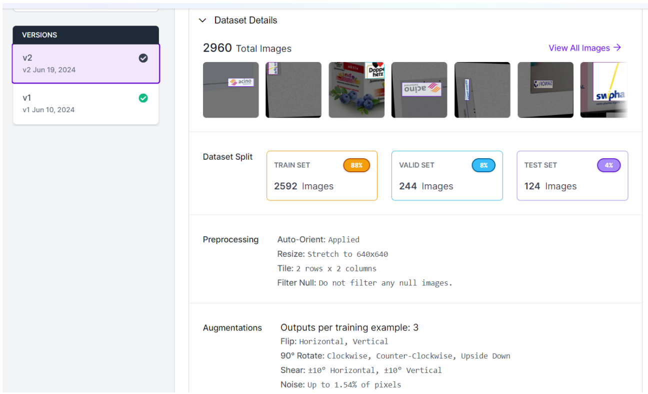


Fig. 4.3, Dataset details

Implementation with Roboflow

Roboflow is a web platform designed to streamline the development process for computer vision applications. It offers a suite of tools to help developers in various stages: data management (collecting, annotating, and organizing image datasets), model training (integration with popular frameworks like TensorFlow and PyTorch), and deployment (making trained models accessible through web APIs or mobile applications). We utilized Roboflow for the preprocessing of our image dataset. Subsequently, the model was trained on Roboflow using transfer learning with the YOLO-NAS S pre-trained model. YOLO-NAS is a state-of-the-art object detection model developed by Deci. It is trained on the "COCO dataset", "object365 Dataset" and "roboflow100 Dataset". It achieves a mean average precision (mAP) of 50.1% at 220 ms latency on an NVIDIA V100 GPU. This is significantly better than the previous state-of-the-art model, YOLOv8, which achieves a mAP of 46.9% at 260 ms latency. Mean Average Precision (mAP) is a key metric used to evaluate the performance of object detection models. It summarizes the precision-recall curve into a single value, providing a clear indication of how well a model is performing. On the COCO dataset, state-of-the-art models typically achieve mAPs in the range of 40% to 60% at 0.5 Intersection over Union (IoU). Higher values, especially above 50%, are considered excellent. The custom logo detection model was trained for 200 epochs, achieving an overall mAP of 79.3%, Precision of 85.3%, and Recall of 75.3%. The training graphs are shown in Figs. 4.4 and 4.5.

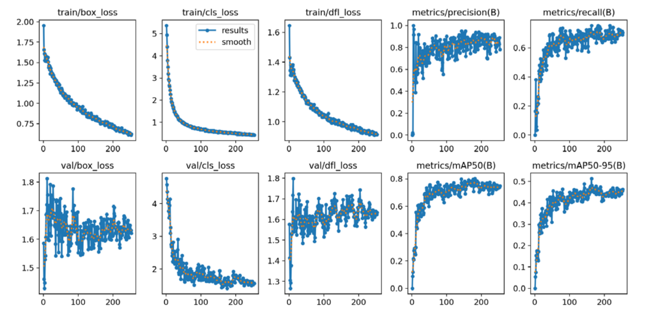


Fig. 4.4. Training graphs

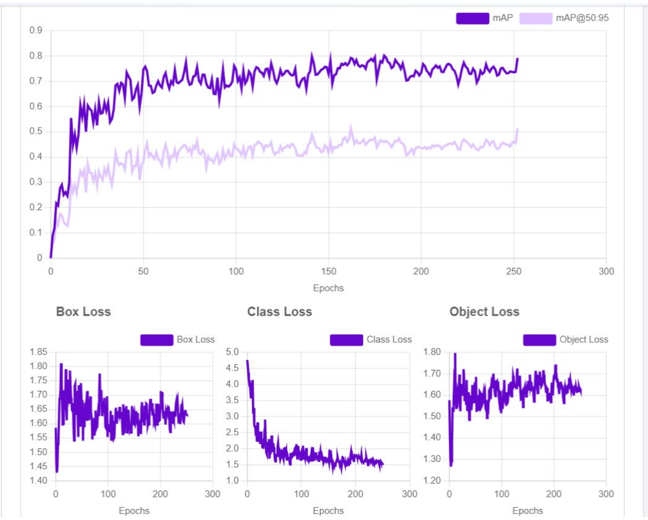


Fig. 4.5. Training Graphs

Application Design

A modular design approach using Flask was adopted for the development of the application to enhance the maintainability, reusability, and testability of the code. The procedure involves breaking down the main functionalities of the application-logo detection, OCR text extraction, and spell-checking into separate, self-contained units known as modules. The architecture of the modular approach is in Fig. 4.6.

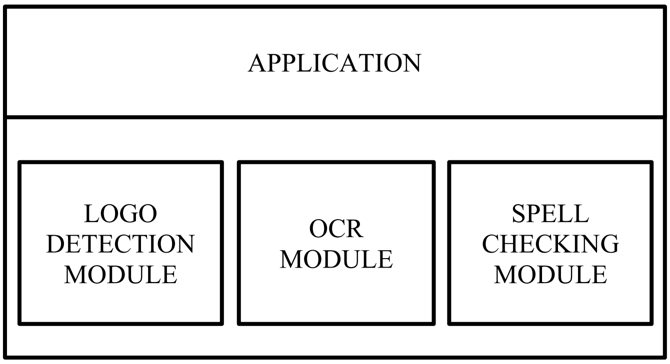


Fig. 4.6. Architecture of modular approach

System Implementation

Here, three implementation tasks were identified. They are:

i. Integration of the custom logo detection into the web application
ii. Integration of the Optical Character Recognition and text extraction functionality
iii. Implementation of the spell-checking on extracted text
iv. Development of the web application Interface

Integration of Custom Logo Detection into the Web Application

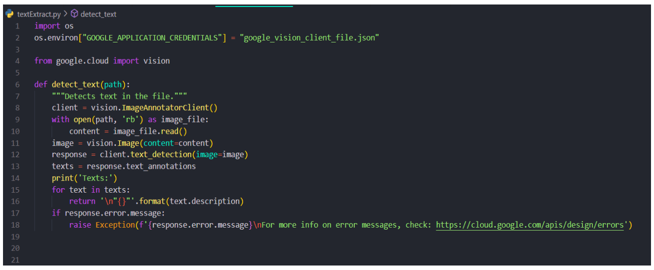
In this study, we integrated the custom logo detection model into the web application which involves the use of the Roboflow Inference API. The process was streamlined by creating a dedicated function within a file named logodetect.py. This function, suitably named logodetection, was designed to interact with the Roboflow Inference API, allowing for seamless model inference. Once the function receives an image input, it sends a request to the API endpoint, which processes the image using the pre-trained custom logo detection model hosted on Roboflow. The API responds with detailed information about the detected logos, identifying the object class detected and class ID (Fig. 4.7).

```python
# logoDetect.py > ...
# import the inference-sdk
from inference_sdk import InferenceHTTPClient

# initialize the client
CLIENT = InferenceHTTPClient(
    api_url="https://detect.roboflow.com",
    api_key="5pXtpPywOyOhT2JhMnnF"
)

# infer on a local image
def logodetection(path):
    result = CLIENT.infer(path, model_id="medicine-logo-detection/2")

    return result['predictions']
```

Fig. 4.7. Model inference module

Integration of the Optical Character Recognition and Text Extraction Feature

The OCR and text extraction functionality in the web application were accomplished using the Google Vision OCR API. Google Cloud Vision offers an OCR feature through its Vision API. This API allows developers to extract text from images. The API can recognize text in various formats, including scanned documents, photos of signs, and handwritten notes. It supports over 60 languages and provides features like text detection (bounding boxes around text areas) and document text detection (optimized for dense text layouts). Similar to the logo detection integration, this functionality is encapsulated within a dedicated module for clarity and modularity. The file, named textExtract.py, contained a function called detect_text, which is responsible for interfacing with the Google Vision OCR API. When an image of the medicine's backstrap is uploaded, the extract_text function sends this image to the Google Vision OCR API endpoint. The API processes the image, utilizing its advanced text recognition algorithms to extract and return the textual content present in the image (Fig. 4.8).



```python
textExtract.py > detect_text
import os
os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = "google_vision_client_file.json"

from google.cloud import vision

def detect_text(path):
    """Detects text in the file."""
    client = vision.ImageAnnotatorClient()
    with open(path, 'rb') as image_file:
        content = image_file.read()
    image = vision.Image(content=content)
    response = client.text_detection(image=image)
    texts = response.text_annotations
    print('Texts:')
    for text in texts:
        return '\n"{}"'.format(text.description)
    if response.error.message:
        raise Exception(f'{response.error.message}\nFor more info on error messages, check: https://cloud.google.com/apis/design/errors')
```

Fig.4.8. OCR and text extraction module

Implementation of Spell-Checking on Extracted Text

To implement spell-checking for the extracted text from the OCR process, it is crucial to first identify and handle named entities. Such as place names, brand names, and specific drug names, which should not be subjected to spell-checking. This can be effectively achieved by passing the extracted text through a Named Entity Recognition (NER) model. The extracted text from the textExtract.py module might contain a mix of common words, chemical names, brand names, place names, and numerical values. We applied a NER model to parse through the extracted text and identify named entities. The NER model will organize words into categories such as PERSON, ORGANIZATION, LOCATION, and others, effectively identifying names of places, brands, and other specific entities. Once named entities are identified, they are filtered out from the text. This ensures that the spell-checking process only includes common English words, chemicals, numbers, and values. With the filtered text containing only common words, chemicals, numbers, and values, we employed a spell-checking algorithm using libraries Hunspell. This will identify and correct any spelling errors in the remaining text.

Development of the Web Application Interface

The development of the web application interface using Flask involves the use of HTML, tailwind CSS and JavaScript to create a user-friendly and interactive experience. The web app consists of three main pages. The home page serves as the initial interface where users can input their data, featuring a form and an image upload function to submit the medicine packaging for authentication. Upon submission, the backend system processes the image through the integrated logo detection, OCR, and spell-checking modules. Based on the results, the application redirects the user to either a success or failure page. The success page is displayed when the medicine passes all the tests, indicating a high probability of authenticity. Conversely, the failure page is shown if the medicine does not pass the tests, suggesting a low probability of authenticity. Flask's templating system, allows for the dynamic generation of these pages, ensuring a seamless flow of information and user interaction. Each page is designed to provide clear feedback and next steps for the user, enhancing the overall usability and effectiveness of the application. See Figs. 4.9, 4.10, 4.11, 4.12 and 4.13

Implemented Modules

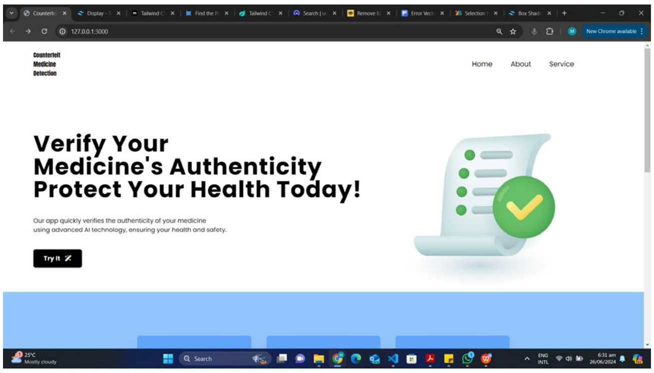


Fig.4.9. User interface home page

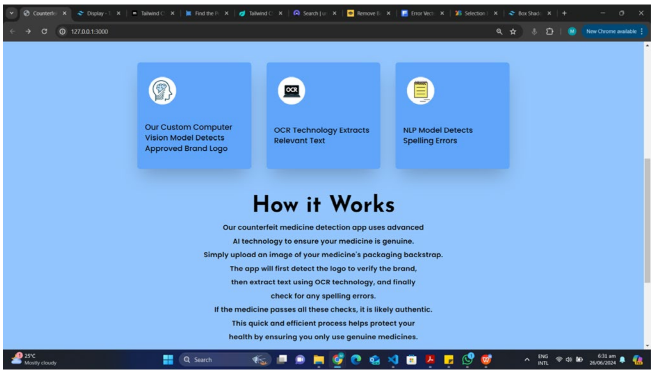


Fig. 4.10. User interface home page cont. showing upload options

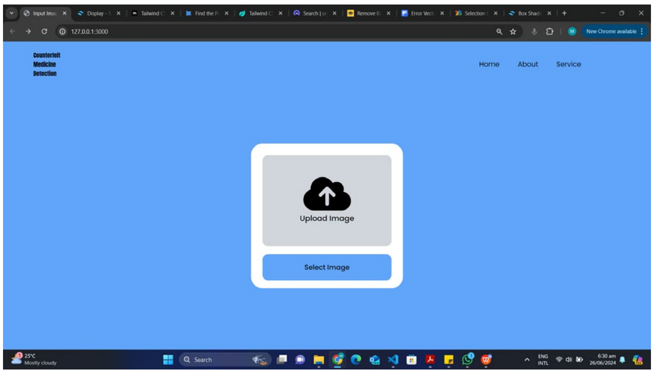


Fig. 4.11. Upload image page

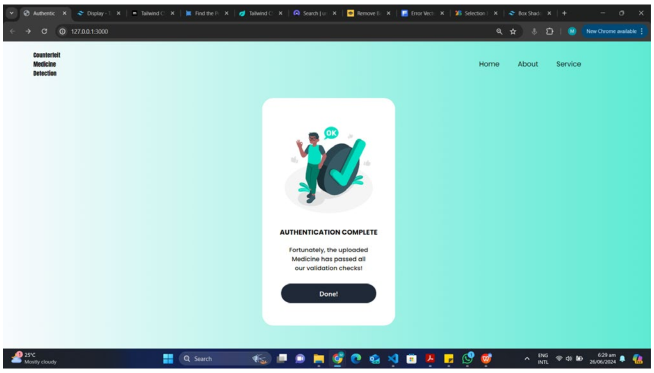


Fig. 4.12. Success page

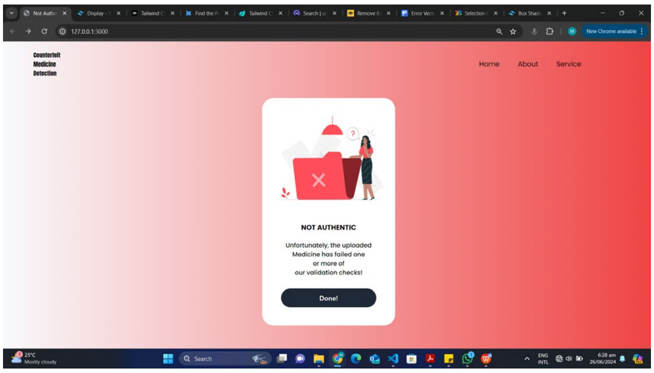


Fig. 4.13. Failure Page

Evaluation of the System

This phase of testing involves various aspects to validate the entire system's behaviour and performance under different scenarios. The application undergoes rigorous testing to verify its ability to accurately detect logos, extract text using OCR, and perform spell-checking on the extracted text. The report of the evaluation carried out by 16 undergraduate students randomly selected indicates that the application is reliable and effective in real-world scenarios as shown in Table 1. The test was based on these Metrics and Benchmarks**:**

1. Response Time (Seconds):
   ≤ 2.5s (Optimal) – Based on Google's Web Performance Standards.
2. Throughput (Requests per Second):
   ≥ 20 requests/sec – Industry standard for web-based AI applications.
3. OCR Accuracy (Word Error Rate - WER%):
   ≤ 10% WER (Good Performance) – Industry Standard for OCR models.
4. Logo Detection Accuracy (Mean Average Precision - mAP%):
   ≥ 85% mAP (Ideal) – Standard for trained object detection models.

TABLE 1. SYSTEM PERFORMANCE EVALUATION

Evaluation #	**Response Time (s)**	**Throughput (requests/sec)**	**OCR Error Rate (WER %)**	**Logo Detection mAP %**
1	2.7	16	12.0	85.2
2	3.0	14	13.1	82.8
3	2.5	17	11.8	76.5
4	2.9	15	12.4	74.9
5	3.2	13	14.0	80.3
6	2.6	18	11.5	83.1
7	3.1	12	14.5	79.7
8	2.8	15	12.9	83.4
9	2.7	16	11.7	76.8
10	3.0	14	13.6	81.9
11	2.9	15	12.2	81.5
12	3.3	11	14.8	80.9
13	2.5	17	11.4	78.2
14	2.8	16	12.7	85.8
15	3.1	13	13.9	82.8
16	2.6	18	11.3	79.0

V. CONCLUSION AND RECOMMENDATIONS

Conclusion

We developed an AI-based application for counterfeit medicine detection in the Nigerian drug market. The application was built with Python and Flask that leverage a modular design to enhance maintainability and scalability. The core functionalities of the application include logo detection, optical character recognition (OCR), and spell-checking to validate the authenticity of pharmaceutical products.

The model adopted was trained on Roboflow using transfer learning with the YOLO-NAS S pre-trained model. It was trained on the "COCO dataset", "object365 Dataset" and "roboflow100 Dataset". It achieves a mean average precision (mAP) of 50.1% at 220 ms latency on an NVIDIA V100 GPU. Similarly, based on system performance evaluation report (Table 1) undertaken by the 16 undergraduate students, suggests that the application can be reliable and effective in real-world scenarios.

Recommendations

We recommend partnerships between researchers, pharmaceutical companies, and regulatory authorities to collect and incorporate more comprehensive datasets, including various brands, packaging variations, and regional differences in medicine labelling into the AI models. Such collaboration would increase the robustness and accuracy of the counterfeit medicine detection system to identify a wide range of counterfeit products and improve public health safety.

Furthermore, due to the application's scalability, we recommend integrating it with the relevant regulatory authorities to enhance its real-world impact.

ACKNOWLEDGMENT

We acknowledge the reviewers for their beneficial comments, which help improve the quality of this work. We also acknowledge the pharmaceutical companies and NAFDAC for making the approved Logos available.

CONFLICTS OF INTEREST

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] Hou, S., Li, J., Min, W., Hou, Q., Zhao, Y., Zheng, Y., & Jiang, S. (2023). Deep Learning for Logo Detection: A Survey. *ACM Transactions on Multimedia Computing, Communications and Applications. 20*(3), 1–23. https://doi.org/10.1145/3611309.

[2] Adigwe, O. P., Onavbavba, G., & Wilson, D. O. M. (2022). Challenges Associated with Addressing Counterfeit Medicinesin Nigeria: An Exploration of Pharmacists' Knowledge, Practices, and Perceptions. *Integrated Pharmacy Research and Practice.* https://doi.org/10.2147/iprp.s387354.

[3] Pathak, R., Gaur, V., Sankrityayan, H. & Gogtay, J. (2023). Tackling Counterfeit Drugs: The Challenges and Possibilities. *Pharmaceutical Medicine, 37,* 281–290. https://doi.org/10.1007/s40290-023-00468-w.

[4] Akinyandenu, O. (2013). Counterfeit drugs in Nigeria: A Threat to Public Health. *African Journal of Pharmacy and Pharmacology,7*(36).

[5] Okereke, M., Anukwu, I., Solarin, S., & Ohuabunwa, M. Z. (2021). Combating Substandard and Counterfeit Medicines in the Nigerian Drug Market: How Industrial Pharmacists Can Rise Up to the Challenge. *Innovations in Pharmacy, 12*(3).

[6] Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S., & Miao, Y. (2021). Review of Image Classification Algorithms based on Convolutional Neural Networks. *Remote Sensing*, *13*(22). https://doi.org/10.3390/rs13224712.

[7] Sahel, S., Alsahafi, M., Alghamdi, M., & Alsubait, T. (2021). Logo Detection using Deep Learning with Pretrained CNN Models. *Engineering, Technology and Applied Science Research*, *11*(1), 6724–6729. https://doi.org/10.48084/etasr.3919.

[8] Daoud, E., Vu, D., Nguyen, H., & Gaedke, M. (2020). Improving Fake Product Detection using AI-based Technology. *18th International Conference e-Society 2020, Sofia, Bulgaria.* Doi: 10.33965/es2020_202005L015.

[9] Asadizanjani, N., Tehranipoor, M., & Forte, D. (2017). Counterfeit Electronics Detection using image processing and machine learning. *Journal of Physics. Conference Series,* 787. DOI 10.1088/1742-6596/787/1/012023.

[10] Dégardin, K., Guillemain, A., Klespe, P., Hindelang, F., Zurbach, R., & Roggo, Y. (2018). Packaging Analysis of Counterfeit Medicines. *Forensic Science International, 291,* 144–157. https://doi.org/10.1016/j.forsciint.2018.08.023.

[11] Sharma, A., Srinivasan, V., Kanchan, V., & Subramanian, L. (2017). The Fake vs Real Goods Problem: Microscopy And Machine Learning to the Rescue. *Applied Data Science Paper. August 13–17, 2011–2019, Halifax, NS, Canada.* https://doi.org/10.1145/3097983.3098186.

[12] Sajidha, S. A., Mairaj, A., Tyagi, A. K., Vijayalakshmi, A., Nisha, V. M., Nair, S., Ganesan, C., Gunasekaran, R., & Menon, H. (2024). Counterfeit Pharmaceutical Drug Identification. *Conversational Artificial Intelligence*, 269–285. https://doi.org/10.1002/9781394200801.ch17

[13] Puglia, F. D. P., Anzanello, M. J., Scharcanski, J., De Abreu Fontes, J., De Brito, J. B. G., Ortiz, R. S., & Mariotti, K. (2021). Identifying the Most Relevant Tablet Regions in the Image Detection of Counterfeit Medicines. *Journal of Pharmaceutical and Biomedical Analysis, 205*, 114336. https://doi.org/10.1016/j.jpba.2021.114336.

[14] Ting, H. W., Chung, S. L., Chen, C. F., Chiu, H. Y., & Hsieh, Y. W. (2020). A Drug Identification Model Developed using Deep Learning Technologies: Experience of a Medical Center in Taiwan. *BMC Health Services Research, 20*(1). https://doi.org/10.1186/s12913-020-05166-w

[15] Ogbonna, B. O., Ilika, A. L. & Nwabueze, S. A. (2015). National Drug Policy in Nigeria (1985-2015). *Word Pharm Res 4*(6), 246–264.

[16] Blackstone, E.A., Fuhr, J.J.P. & Pociask, S. (2014). The Health and Economic Effects of Counterfeit Drugs. *Am Health Drug Benefits*. *7*(4), 216–224.

[17] Zagare, S., Khodaskar, M., Sonawane, Y., & Verma, H. (2024). Counterfeit Medicine Detection using Blockchain. *International Journal of Soft Computing and Engineering (IJSCE), 14*(2).

[18] Islam, I. & Islam, M. N. (2022). Digital Intervention to Reduce Counterfeit and Falsified Medicines: A Systematic Review and Future Research Agenda. *Journal of King Saud University-Computer and Information Science. 34*, 6699–6718.