Journal of Computing, Science &Technology
https://focjournal.unidel.edu.ng/
editorjcst@unidel.edu.ng
ISSN:......Print ISSN: .... Online

Development of Convolutional Neural Network-Based Banknote Recognition System

D. Olusoga Akinmosin[1] & O. S. Adewale[2]
[1]Department of Mathematics & Computer Science, Elizade University, Ilara-Mokin, Nigeria
[2]Department of Computer Science, School of Computing, FUTA, Akure, Nigeria
[1]dolusoga@gmail.com; [2]adewale@futa.edu.ng

Corresponding Author's Email: *dolusoga@gmail.com*

Article Info
Date Received: 17-01-2024
Date Accepted: 13-02-2024

ABSTRACT

Banknotes play a crucial role in various financial transactions, particularly in daily electronic currency exchanges. The ability to distinguish between authentic and counterfeit currency is a major challenge in the realm of computer vision. Despite the availability of different digital currency transaction platforms, physical cash continues to be the favoured choice for everyday transactions at vending machines, banks, shopping centres, railway station counters, ATMs, and foreign exchange bureaus. Moreover, there are cases where individuals unknowingly come into possession of fake money. Regrettably, this problem persists, and financial institutions are unable to outfit all their branches with the essential counterfeit currency detection mechanisms. Hence, this study introduces a banknote recognition system based on convolutional neural networks, which utilises a range of security features found on Nigerian banknotes. The experimental findings demonstrate the potential of the suggested methodology, yielding highly notable outcomes. It effectively carries out a proficient and resilient classification by utilising actual scene images obtained from the Internet and those images captured under both natural and artificial lighting conditions. Furthermore, it remains unaffected by the rotation and translation of banknotes. Notably, a commendable recognition rate was attained for Nigerian banknotes. Currently, the outcomes presented in this study represent a significant advancement compared to the existing state of the art.

1.0 INTRODUCTION

A banknote recognition system is designed to identify and authenticate paper currency. Such systems are used in a variety of applications, including ATMs, vending machines, cash counters, and point-of-sale terminals. Banknote recognition systems play a crucial role in ensuring the integrity of cash transactions and preventing the circulation of counterfeit currency. With ongoing advancements in technology, these systems continue to improve in accuracy, speed, and reliability.

In the era of significant technological advancements, the Internet has become an integral part of our daily lives. This includes financial transactions, where digital methods are increasingly common. However, physical banknotes still play a crucial role, despite challenges in automated handling. The rise in counterfeit banknotes has led to the necessity of developing detection systems. While money-counting machines equipped with sensors like magnetic, infrared, or ultraviolet are utilized, their effectiveness is limited by noise interference and high costs. As a result, these systems are not widely accessible to the public. To address these issues, computer vision and traditional Machine Learning techniques such as k-nearest neighbour, genetic algorithms, and fuzzy systems have been employed. Nevertheless, these methods may struggle with accuracy when dealing with worn or dirty banknotes, and they may lack generalization capabilities.

Modern computer vision methodologies, notably convolutional neural networks and deep learning, are extensively employed to leverage their effective pattern extraction capabilities. Convolutional neural network has demonstrated its exceptional performance in comparison to conventional machine learning approaches when it comes to image classification tasks. Its application has been successfully demonstrated in the identification of counterfeit currencies from diverse nations, including the Pakistani Rupee (Ali *et al*., 2019), Indian Rupee (Pham *et al*., 2019), Euro, and United States Dollar (Han and Kim, 2019). Despite achieving high levels of accuracy, these methods fail to provide clear guidance on which approach is most effective in identifying counterfeit banknotes. Based on the existing literature, there is scanty research on the Nigeria currency across all its various denominations using the various banknotes security features. The prevention of counterfeit banknotes plays a crucial role in maintaining the security and trustworthiness of financial systems while combating fraudulent activities. The primary objective of this study is to develop a dependable system that can effectively identify counterfeit Nigerian banknotes, addressing the limitations observed in previous research that failed to concentrate on this specific currency. The main challenge lies in accurately distinguishing between authentic and counterfeit Nigerian banknotes. Conventional detection methods have their own deficiencies, thus necessitating the adoption of enhanced methodologies that make use of contemporary techniques (Khan *et al*., 2020). This article introduces an innovative approach that utilises convolutional neural networks to tackle the issue of counterfeit Nigerian banknote detection. The main contribution lies in customising this advanced deep learning technique for this specific purpose. By doing so, this study not only fills a gap in previous research but also offers a practical solution for identifying fraudulent banknotes in Nigerian financial transactions. The proposed approach shows potential for enhancing efficiency and accuracy compared to conventional methods.

2.0 RELATED WORKS

Ballado *et al*. (2015) utilised a distinctive optically variable device patch to detect counterfeit Philippine notes. The canny edge algorithm was employed to identify fake currency by leveraging the security features of the optically variable device. The results demonstrated statistically significant detection rates across all four tests, with a threshold of significance set at 5 percent. A method for detecting counterfeit currency based on AlexNet architecture was introduced by Laavanya & Vijayaraghavan (2019). The convolutional neural network, which was pre-trained on a large dataset, was fine-tuned using 50, 200, 500, 2000 Indian rupee notes to extract relevant feature vectors. The model achieved an average accuracy of 81.5% in distinguishing genuine currency from fake currency, with a slightly lower accuracy of 75% specifically for identifying counterfeit notes. Kamble *et al*. (2019) introduced a sophisticated convolutional neural network (CNN) model for the identification of counterfeit currency. The model was designed specifically to identify fake notes on handheld devices such as smartphones and tablets. To train and evaluate the CNN model, a dataset of 10,000 images was created, consisting of 500 genuine notes, 500 counterfeit notes, 2000 real notes, and 2000 fake notes. Through this approach, a testing accuracy of 85.6% was achieved.

Narra & Kirar (2021) utilised a combination of classifiers to tackle the task of fake currency classification. This system incorporates multiple security features to enhance its effectiveness. The classifiers employed in this system include DT, SVM, LDA, and KNN. Among these classifiers, the SVM classifier demonstrates superior performance, surpassing all others, with an impressive accuracy rate of 82.7 percent across all features. Bhatia *et al*. (2021) employed KNN, SVC, and GBC to detect counterfeit currency. KNN stands out as a viable option for computer vision tasks due to its remarkable accuracy when dealing with smaller datasets. The integration of machine learning algorithms and image processing techniques enables the attainment of the desired outcome and accuracy. Notably, both KNN and GBC exhibit superior accuracy in the recognition task. In the study conducted by Latha *et al*. (2021), a novel approach for identifying counterfeit currency utilizing edge detection has been introduced. The detection system employs a training dataset that mirrors the dataset to be evaluated in subsequent tests. Remarkably, this system, which relies on an edge detector, achieves an impressive accuracy rate of 90.45% in detecting counterfeit currency. Sumalatha et al. (2022) employed four distinct convolutional neural networks namely Alexnet, Resnet50, Darknet53, and Google net to conduct Indian currency recognition. The study revealed that each of these preconfigured networks exhibited exceptional performance in a specific parameter, albeit at the expense of other parameters. While certain research has shown promising results in terms of accuracy, it is important to note that these findings are not generalisable real-world situations that involve other nation's banknote scenarios. In addition, certain research endeavours may necessitate extended processing durations as a result of employing intricate computational algorithms and internal architectural layers. Hence, this study.

3.0 METHODOLOGY

3,1 The proposed system

The Naira banknotes are designed with several security elements that serve the purpose of facilitating the easy identification of genuine notes. These distinctive features can be detected through both touch and sight. They include the raised print, security thread, and watermark. Moreover, the notes also possess embossed features such as the portrait, lettering, and denominational numerals on both sides. The raised print provides a tactile aspect, while the security thread, which may seem broken under normal light, reveals the letters "CBN" when held up to light. This enhances the security measures. Additionally, the Naira notes are equipped with additional security measures that are visible under ultraviolet light, making them highly resistant to counterfeiting. For example, the serial number on each banknote appears black but changes to green when exposed to ultraviolet light. According to the guidelines set by the CBN, there are several security features that should be taken into consideration. These features include intaglio, portrait watermark, optically viable ink (OVI), kinegram, iridescent band, and engraved portrait. Intaglio involves carving the image on the naira note's surface, creating a raised design that holds ink. The new naira notes also feature a portrait watermark, which is a transparent logo, text, or signature superimposed on a photograph. The OVI security ink reflects light at an angle, causing the colour to change when viewed from different perspectives. The Kinegram, found on the ₦1000 note, displays the Nigerian Coat of Arms and serves as an effective way to detect counterfeit notes. Additionally, the new ₦1000 note has an iridescent band on the top that changes appearance when viewed from various angles. Lastly, the image of Nnamdi Azikiwe on the new ₦500 note is engraved into the surface, with additional details on the right side for easy identification of the original note.

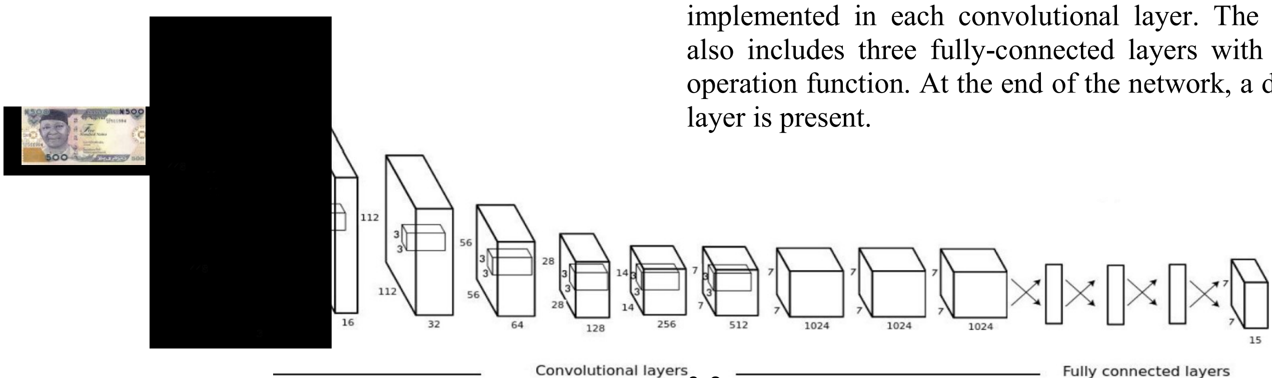
The proposed system relies on convolutional neural networks for the recognition of banknotes. The approach enhances the capability to differentiate banknotes to a level equal to or greater than the results previously presented in the literature. The design of the object detection network utilised in this article is influenced by Redmon *et al.* (2016). This network takes an image as its input, processes it through a Convolutional Neural Network (CNN), and generates a vector containing bounding boxes and class predictions as the output. The input image is segmented into an $S \times S$ grid of cells, with each grid cell responsible for predicting the presence of an object within it based on the centre of the object. Each grid cell provides predictions for B bounding boxes and C class probabilities. The bounding box prediction comprises five components: (x, y, w, h, confidence). The coordinates (x, y) denote the centre of the box in relation to the grid cell position, and these values are normalised to range between 0 and 1. Similarly, the dimensions (w, h) of the box are also normalised to [0, 1] with respect to the image size. In addition to this, it is essential to estimate the class probabilities, Pr(Class(i)|Object), which are dependent on the presence of an object within the grid cell. If no object is detected in the grid cell, the loss function will not penalise incorrect class predictions. The model predicts a single set of class probabilities per cell, irrespective of the number of boxes B, resulting in $S \times S \times C$ class probabilities in total. The architecture comprises 9 convolutional layers, 3 fully-connected layers and detection layer. While the final layer employs a linear activation function, the preceding layers utilise a leaky RELU activation function (equation 1) with a parameter α set to 0.1.

$$f(x) = \begin{cases} x \text{ if } x > 0 \\ \alpha x \text{ otherwise} \end{cases} \qquad (1)$$

The loss function $l_f$is composed of several parts:

$$\begin{aligned} l_f &= \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i\text{-}\hat{x}_i)^2\text{-}(y_i\text{-}\hat{y}_i)^2] \\ &+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i}\text{-}\sqrt{\hat{w}_i}\right)^2 \text{-} \left(\sqrt{h_i}\text{-}\sqrt{\hat{h}_i}\right)^2 \right] \\ &+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(C_i\text{-}\hat{C}_i\right)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(C_i\text{-}\hat{C}_i\right)^2 \\ &+ \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \varepsilon classes} (p_i(c)\text{-}\hat{p}_i(c))^2 \qquad (2) \end{aligned}$$

The initial segment of equation 2 calculates the loss associated with the anticipated positions of the bounding boxes (x, y), which involves a summation across each bounding box predictor ($j = 0 \ldots B$) within each grid cell ($i = 0 \ldots S^2$). The value of $1_{ij}^{obj}$is set to 1 if an object is detected in grid cell i and the j-th bounding box predictor is accountable for that particular prediction. The variables (x, y) represent the predicted bounding box positions, while ($\hat{x}$, $\hat{y}$) denote the current positions derived from the training data. The subsequent segment of the equation pertains to the error metric, which aims to emphasise that minor deviations in larger boxes hold less significance compared to those in smaller boxes. To address this, the square root of the bounding box width and height is predicted instead of directly predicting the width and height. The third component of the equation deals with the loss linked to the confidence score for each bounding box predictor. Here, C signifies the confidence score, and $\hat{C}$ represents the Intersection over Union of the predicted bounding box with the ground truth. The value of 1^{obj} is set to 1 when an object is detected in the cell (thus increasing the confidence to the Intersection over Union), and 0 otherwise (leading to a decrease in confidence to 0). Conversely, 1^{noobj} represents the opposite scenario. The final segment of the loss function pertains to the object classification loss. The term 1^{noobj} is utilised to penalise a classification error in instances where no object is present in the cell. The cost function as outlined in Redmon *et al.* (2016) is imperative for conducting the training process. The network architecture employed for training the proposed system is depicted in Figure 1. The CNN architecture described here comprises 9 convolutional layers with 3 × 3 kernels. In each convolutional layer, the number of kernels is doubled. The initial layer consists of 16 kernels, while the final three layers each have 1,024 kernels. Following each convolutional layer, a maxpool operation of 2 × 2 is applied to the remaining image size. Additionally, a leaky ReLU operation function is implemented in each convolutional layer. The network also includes three fully-connected layers with a linear operation function. At the end of the network, a detection layer is present.



3.2

Figure 1: The CNN architecture for banknote recognition system

3.2 Evaluation Metrics

The assessment of the object detection and localisation algorithm's performance is conducted using a measurement known as Average Precision (AP) (and mean average precision). mAP is determined by utilising various metrics including IoU, confusion matrix (TP, FP, FN), precision, recall, and other relevant factors as depicted in the Figure 2.

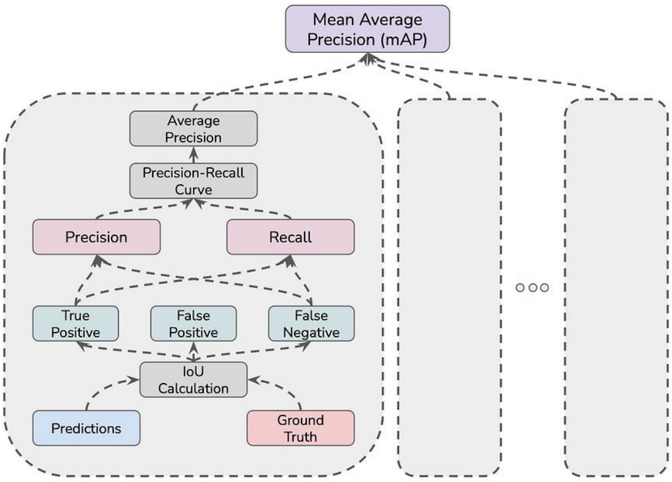


Figure 2: Computation of Mean Average Precision

In order to comprehend AP, it is imperative to gain a comprehensive understanding of these metrics. First, Intersection over Union (IoU). Intersection over Union (IoU) measures the proximity between two bounding boxes, namely the ground truth and the prediction. It is a numerical value ranging from 0 to 1. A perfect prediction occurs when the two bounding boxes completely overlap, resulting in an IoU of 1. Conversely, if there is no overlap between the two bounding boxes, the IoU is 0. The IoU is computed by dividing the area of intersection by the area of the union of the two bounding boxes. Second, True Positive, False Positive, False Negative. To determine the accuracy of a prediction, it is necessary to compare the class label of the predicted bounding box with the ground truth bounding box. Additionally, the Intersection over Union (IoU) between these two bounding boxes must exceed a specified threshold value. Based on the IoU, threshold, and the class labels assigned to the ground truth and predicted bounding boxes, three metrics are calculated: (i) True Positive: This metric indicates that the model correctly predicted the existence of a bounding box at a specific position. In other words, the model identified a positive instance and its prediction was accurate. (ii) False Positive: This metric represents cases where the model incorrectly predicted the presence of a bounding box at a particular position. It indicates that the model identified a positive instance, but its prediction was incorrect. (iii) False Negative: This metric occurs when the model fails to predict the existence of a bounding box at a certain position. In other words, the model identified a negative instance, but its prediction was incorrect because a ground truth bounding box actually existed at that position. It is important to note that the True Negative metric is not used to calculate the final metrics. This metric corresponds to the background area without any bounding boxes and indicates that the model correctly did not predict the presence of a bounding box.

Third, Precision & recall. Based on the values of true positives (TP), false positives (FP), and false negatives (FN), precision and recall are calculated for each labelled class. Precision indicates the accuracy of the model by showing the proportion of correctly identified instances in relation to the total number of predicted instances. It is calculated as the ratio of true positives to the sum of true positives and false positives. Recall, on the other hand, measures the model's ability to correctly identify instances from the total number of actual instances. It is calculated as the ratio of true positives to the sum of true positives and false negatives. Fourth, Precision-Recall curve. Ideally, it is desirable for both precision and recall to be high, meaning that the model can accurately detect all occurrences of a class while also ensuring that what is detected is correct. The values of precision and recall are influenced by the number of true positives detected by the model, which in turn determines the assignment of bounding boxes as true positives (TP), false positives (FP), and false negatives (FN). The determination of TP, FP, and FN is based on two main factors: the comparison between the predicted label and the ground truth label, as well as the Intersection over Union (IoU) between the two bounding boxes. In the context of a multiclass classification problem, the model provides a conditional probability indicating the likelihood that a bounding box belongs to a specific class. A higher probability for a certain class suggests that the bounding box is more likely to contain that class. By utilising a probability distribution in conjunction with a user-defined threshold value (ranging from 0 to 1), bounding boxes can be classified accordingly. The choice of the probability confidence threshold plays a crucial role in the model's performance. A lower threshold leads to a greater number of detections by the model, potentially increasing recall while reducing the likelihood of missing ground-truth labels. Conversely, a higher confidence threshold results in a more confident prediction by the model, potentially enhancing precision. However, there exists a trade-off between precision and recall based on the selected confidence threshold, as we aim to maximize both metrics. To assess the performance of the model across different confidence threshold values, a precision-recall curve is constructed. This curve illustrates the relationship between precision and recall for varying threshold values, providing a visual representation of the optimal threshold for a specific application. Fifth, Average Precision. Choosing an appropriate confidence level for your application may pose a challenge as it involves subjective judgment. Average precision serves as a crucial metric that aims to mitigate the reliance on a single confidence threshold value. It is calculated as the area under the precision-recall curve. Average precision condenses the information from the precision-recall curve into a single numerical value. A high average precision indicates high levels of both precision and recall, while a low average precision suggests that either precision or recall (or both) are low across various confidence threshold values. The average precision score typically ranges between 0 and 1. Finally, Mean Average Precision. The AP value can be computed for every individual class. The mean average precision is determined by averaging the AP values across all the classes that are being considered. In other words, the mean average precision provides an overall measure of the precision across multiple classes and it is obtained as shown in equation 3.

$$\mathrm{mAP} = \frac{1}{\mathrm{k}} \sum_{\mathrm{i}=1}^{\mathrm{k}=\mathrm{n}} \mathrm{AP_i} \tag{3}$$

4.0 IMPLEMENTATION & EXPERIMENTAL SETUP

The implementation process involves the following:

a. Data collection

Nigerian Naira currency images totalling 140, including One Hundred Naira (₦100) (30 in number), Two Hundred Naira (₦200) (25 in number), Five Hundred Naira (₦500) (30 in number), and One Thousand Naira (₦1000) (19 in number) notes, were sourced from online platforms and captured using a camera. To preserve image quality and prevent loss of information during resizing, all images were saved in .png format.

b. Data preprocessing

Data preprocessing involves two stages. First, image resizing. Resizing images is crucial to ensure uniform dimensions across the dataset, facilitating mini-batch learning, enhancing computational efficiency, and effectively managing memory usage. In this article, images were resized to dimensions 600 by 1024 pixels. Second, image normalisation. Normalisation was employed to standardise pixel values, ensuring a mean of 0 and a standard deviation of 1. This process fosters model stability during training by maintaining values within a standardized range, thus aiding in convergence.

c. Data annotation

In the current era of machine learning and computer vision, the significance of annotated data cannot be emphasised enough. Annotating images is a vital stage in preparing models for activities such as object detection, instance segmentation, and semantic segmentation. Nevertheless, manually labelling images can be a laborious and time-intensive task. This is where platforms like LabelMe become valuable, providing a straightforward interface to simplify the annotation procedure. LabelMe is a graphical image annotation tool that is open-source and created to simplify the image annotation process. Its main objective is to offer users an easy-to-use tool for annotating images using different primitives like polygons, rectangles, circles, lines, and points. LabelMe is a valuable tool for image annotation, which is a crucial step in the development of machine learning models for tasks like object detection, instance segmentation, and semantic segmentation. Moreover, LabelMe stands out from traditional annotation tools by incorporating machine learning capabilities directly into its system. This incorporation allows users to take advantage of pre-trained models for automated segmentation tasks, making the annotation process more efficient and effective. LabelMe's architecture revolves around two fundamental elements: the graphical user interface (GUI) and the Python backend that supports it. This architectural framework has been meticulously crafted to facilitate the seamless annotation of images by users, while simultaneously offering a resilient infrastructure for data processing and management.

Data labelling was performed manually using the LabelMe tool, which entailed outlining polygons around each object of interest. During this procedure, LabelMe automatically generated metadata files in JSON (JavaScript Object Notation) format for each annotated image. JSON (JavaScript Object Notation) serves as a lightweight format for exchanging data. It is designed to be easily readable and writable by humans, while also being simple for machines to parse and create. JSON is rooted in a subset of the JavaScript Programming Language Standard ECMA-262. JSON is structured around two main components: (i) a set of name/value pairs, which can be represented as an object, record, struct, dictionary, hash table, keyed list, or associative array in different programming languages; and (ii) an ordered list of values, which can be depicted as an array, vector, list, or sequence in most programming languages. These JSON files encapsulate vital information such as (i) Label Names: Identifiers for annotated objects; (ii) ImageData: Base64-encoded image data facilitating visualization within the JSON file; (iii) ImagePath: File path leading to the annotated image; (iv) Dimensions: Width and height specifications of the annotated image; and (v) Coordinates: Vertex coordinates or bounding box details of annotated objects.

Table 1 illustrates each currency item with bounding boxes delineating the objects of interest, denoting the authenticity of the currency. The colour of the bounding box lines corresponds to the respective tag names in the label name column.

Table 1: Samples of currencies with bounding boxes

	1000TriangleSign 1000GoldPatch 1000Digit
	500Crest 500Digit 500MettalicSecurity 500SilverPatch
	200GoldPatch 200MettalicSecurity 200Crest 200Digit
	100MettalicSecurity 100Crest 100Digit 100Sign

d. Data conversion
In order to ensure compatibility with the developed model. The data in LabelMe format was convert into COCO (Common Objects in Context) data format. COCO represents a comprehensive dataset for object detection, segmentation, and captioning on a large scale. More so, it serves as benchmark for evaluating object detection models.

e. Data splitting
The process data split involves randomly partitioning the dataset into two subsets: one for training the model and another for evaluating its performance. The ratio between the training and testing sets was 80:20 ratio. The larger portion of the data was used to train the model while the remaining number was used to validate the model's performance.

Experimental setup
The model was implemented using python programming language with the PyTorch deep learning framework on the Google Colab platform with access to GPU acceleration using T4 GPU. PyTorch, a comprehensive framework for constructing deep learning models, is widely employed in various applications such as image recognition and language processing. Developed using Python, it offers a user-friendly interface for machine learning developers. Notably, PyTorch stands out due to its exceptional GPU support and utilisation of reverse-mode auto-differentiation. This unique feature allows for dynamic modification of computation graphs, making PyTorch a preferred option for rapid experimentation and prototyping. Google Colab, also known as Google Colaboratory, is a complimentary platform provided by Google which enables users to create and execute Python code directly within their web browser. This platform eliminates the need for users to be concerned about the specifications of their hardware or the software configurations on their personal computers. Additionally, Google Colab simplifies access to computational resources and popular machine learning libraries.

The trained model is applied to the testing dataset to generate predictions (bounding boxes and class labels) for each input image. The following parameters were involved

```
[05/06 09:12:31] d2.engine.train_loop INFO: Starting training from iteration 0
[05/06 09:12:56] d2.utils.events INFO:  eta: 0:52:26  iter: 19  total_loss: 4.628  loss_cls: 2.922  loss_box_reg: 0.1097  loss_mask: 0.6935
loss_rpn_cls: 0.8923  loss_rpn_loc: 0.2284   time: 0.8741  last_time: 1.1415  data_time: 0.1193  last_data_time: 0.0085  lr: 3.9962e-05  max_mem:
5741M
[05/06 09:13:18] d2.utils.events INFO:  eta: 0:53:28  iter: 39  total_loss: 2.854  loss_cls: 1.476  loss_box_reg: 0.1293  loss_mask: 0.6827
loss_rpn_cls: 0.2487  loss_rpn_loc: 0.2104   time: 0.9115  last_time: 1.2642  data_time: 0.0101  last_data_time: 0.0164  lr: 7.9922e-05  max_mem:
5741M
[05/06 09:13:37] d2.utils.events INFO:  eta: 0:53:13  iter: 59  total_loss: 1.45  loss_cls: 0.3469  loss_box_reg: 0.1485  loss_mask: 0.6533
loss_rpn_cls: 0.07551  loss_rpn_loc: 0.1825   time: 0.9184  last_time: 1.2820  data_time: 0.0124  last_data_time: 0.0116  lr: 0.00011988  max_mem:
5741M
[05/06 09:13:54] d2.utils.events INFO:  eta: 0:53:08  iter: 79  total_loss: 1.379  loss_cls: 0.3473  loss_box_reg: 0.1844  loss_mask: 0.617
loss_rpn_cls: 0.04681  loss_rpn_loc: 0.1718   time: 0.9112  last_time: 0.7561  data_time: 0.0109  last_data_time: 0.0099  lr: 0.00015984  max_mem:
5741M
[05/06 09:14:11] d2.utils.events INFO:  eta: 0:52:52  iter: 99  total_loss: 1.342  loss_cls: 0.3605  loss_box_reg: 0.224  loss_mask: 0.5817
loss_rpn_cls: 0.06433  loss_rpn_loc: 0.1149   time: 0.8993  last_time: 0.7883  data_time: 0.0105  last_data_time: 0.0082  lr: 0.0001998  max_mem:
5741M
[05/06 09:14:30] d2.utils.events INFO:  eta: 0:53:28  iter: 119  total_loss: 1.312  loss_cls: 0.3574  loss_box_reg: 0.2377  loss_mask: 0.5298
loss_rpn_cls: 0.04476  loss_rpn_loc: 0.1301   time: 0.9012  last_time: 1.1177  data_time: 0.0130  last_data_time: 0.0064  lr: 0.00023976  max_mem:
5741M
```

training the model. These include (i) learning rate: this determines the step size used for updating the model parameters during optimization. This is set at 0.002; (ii) batch size: This specifies the number of samples processed in each training iteration. Due to the few samples used in training batch size is set 5; (iii) number of iterations: Specifies the total number of iterations or epochs for training the model. This is set at 4200 iterations. The sample iteration is shown below as the various loss functions such as the classification loss, bounding box loss regression loss, and prediction loss.

5.0 RESULTS & DISCUSSIONS

Through a series of trials, various experiments were conducted in order to identify the specific areas on the banknotes that are most pertinent, thus enabling a more effective training process with enhanced generalisation capabilities. In total, 140 of banknote images were utilised for the experiments. The outcomes of the complete banknote are presented in Table 2.

Category	Recognition (Originality)
1000Digits	99.73%
1000GoldPatch	99.56%
1000TriangleSign	99.67%
500Digit	99.73%
500Crest	99.56%
500SilverPatch	99.67%
200Digit	99.73%
200GoldPatch	99.56%
200Crest	99.67%
100Digit	99.73%
100Sign	99.67%
100Crest	99.36%

The proposed system achieved an efficiency rate of 99.65% in recognising the selected Nigerian banknotes. The results of the study indicate that the selected key characteristics are conducive to achieving a high recognition rate. Additionally, the system exhibits enhanced generalisation capabilities when presented with new samples.

5.1 Performance Evaluation

Average Precision (AP) and its variants, such as AP50, AP75, APs, APm, and APl were used to evaluate the performance of the developed model in the context of instance segmentation and object localisation. These metrics provide insights into how well a model detects objects across different thresholds of confidence and object sizes. Mean Average Precision (mAP) summarises the precision-recall curve generated by varying the confidence threshold for detection. It computes the area under the precision-recall curve, indicating the average precision across all recall levels. AP provides an overall measure of detection performance, considering all object sizes and confidence thresholds. Table 3 & 4 show the evaluation result for bounding boxes on training set and the evaluation result for bounding boxes per category on training set respectively. Table 5 & 6 show the evaluation result for bounding boxes on test set and the evaluation result for bounding boxes per category on test set respectively.

Table 3: evaluation result for bounding boxes on Training set

AP	AP50	AP75	Aps	APm	APL
88.247	98.602	94.961	92.500	89.513	88.454

Table 4: Evaluation result for bounding box per category on Training set

Category	AP	Category	AP	Category	AP
1000Digits	89.158	1000GoldPatch	97.772	1000TriangleSign	91.188
100Barcode	100.000	100CoatofArm	100.00	100Crest	83.267
100Digit	91.122	100MettalicSecurity	85.050	100Sign	70.198
200CoatofArm	90.000	200Crest	90.363	200Digit	90.644
200GoldPatch	96.985	200MettalicSecurity	93.985	500Crest	82.463
500Digit	62.475	500MetallicSecurity	84.901	500SilverPatch	88.879

Table 5: Evaluation result for bounding box on Test set

AP	AP50	AP75	Aps	APm	APl
91.233	98.602	97.310	87.500	91.910	92.160

Table 6: Evaluation result for bounding box per category on Test set

Category	AP	Category	AP	Category	AP
1000Digits	94.455	1000GoldPatch	100.000	1000TriangleSign	95.092
100Barcode	100.000	100CoatofArm	100.000	100Crest	89.901
100Digit	88.845	100MettalicSecurity	85.145	100Sign	80.198
200CoatofArm	100.000	200Crest	97.456	200Digit	94.594
200GoldPatch	100.00	200MettalicSecurity	95.644	500Crest	77.942
500Digit	64.590	500MetallicSecurity	86.592	500SilverPatch	91.743

An AP of 91.233% indicates a high level of accuracy in identifying objects across various classes. An AP50 of 98.602% indicates a very strong performance. The value of 97.310% at AP75 indicates a very high accuracy even under stringent conditions. The values of 87.500%, 91.910%, and 92.160% at APs, APm, and APl respectively indicates that the developed model performs slightly better on medium and large objects compared to small ones.

6.0 CONCLUSION

The study introduced a method for identifying selected security features and values of banknotes with a remarkable level of accuracy for Nigerian currencies. The approach involved utilising a convolutional neural network model integrated with deep learning techniques specifically designed for Nigerian banknotes. The model underwent training using a supervised algorithm, which necessitated a substantial amount of annotated data for effective learning. A total of 140 images were used for the banknote denomination during the training phase. The outcomes of this study outperformed existing methods in terms of efficiency. The proposed system exhibited superior results when processing banknotes with values and the selected security features, as well as when analysing the complete banknote. In future research endeavours, the proposed system will be expanded to include all the entire security features in the Nigerian banknotes, with a focus on implementing deep learning neural networks to facilitate the classification.

REFERENCES

Ali, T., Jan, S., Alkhodre, A., Nauman, M., Amin, M. & Siddiqui, M.S. (2019). DeepMoney: counterfeit money detection using generative adversarial networks, J Comput. Sci., 5, p. e216

Ballado A. H., Dela Cruz J. C. D.; Avendaño G. O.; Echano N. M.; Ella J. E.; Medina M.E.M.; & Paquiz B.K.C. (2015). "Philippine currency paper bill counterfeit detection through image processing using Canny Edge Technology," International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), pp. 1-4, doi: 10.1109/HNICEM.2015.7393184.

Bhatia, A., Kedia, V., Shroff, A., Kumar, M., & Shah, B. K. (2021, May). Fake currency detection with machine learning algorithm and image processing. In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 755-760). IEEE.

Han, M. & Kim, J. (2019). Joint banknote recognition and counterfeit detection using explainable artificial intelligence sensors, 19(16):3607

Narra, P., & Kirar, J. S. (2021). Indian currency classification and fake note identification using feature ensemble approach. In 2021 IEEE International Conference on Computational Performance Evaluation, pp. 22-29.

Kamble, K., Bhansali, A., Satalgaonkar, P. & Alagundgi, S. (2019). Counterfeit currency detection using deep convolutional neural network, IEEE Pune Section International Conference (PuneCon), pp. 1-4, doi: 10.1109/PuneCon46936.2019.9105683.

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A.S. (2020). A survey of the recent architectures of deep convolutional neural networks, Artif. Intell. Rev., 53(8):5455-5516

Laavanya, M., & Vijayaraghavan, V. (2019). Real time fake currency note detection using deep learning. Int. J. Eng. Adv. Technol., 9.

Latha, L., Raajshree, B. and Nivetha, D. (2021). "Fake currency detection using Image processing," International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation, pp. 1-5, doi: 10.1109/ICAECA52838.2021.9675592.

Pham, T.D., Park, C., Nguyen, D.T., Batchuluun, G. & Park K.R. (2020). Deep Learning-Based Fake-Banknote Detection for the Visually Impaired People Using Visible-Light Images Captured by Smartphone Cameras, IEEE Access, 8 (2020):63144-63161

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. arXiv: 1506.02640v5

Sumalatha, R., Reddy, B. J., & Reddy, T. V. R. (2022, March). Identification of Fake Indian Currency using Convolutional Neural Network. In 2022 6th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 1619-1623). IEEE.