Addis Ababa University

School of Graduate Studies

College of Natural Science

Department of Computer Science

A Model for Recognition and Detection of the Counterfeit of Ethiopian Banknotes using Transfer Learning

Hailemikael Tesfaw Haile

A Thesis Submitted to Addis Ababa University, Department of Computer Science in Partial fulfillment of the Requirements for the Degree of Master of Science in Software Engineering

Addis Ababa, Ethiopia

June 2024

Addis Ababa University

School of Graduate Studies

College of Natural Sciences

Department of Computer Science

Hailemikael Tesfaw Haile

Advisor: Ayalew Belay (PhD)

This is to certify that the thesis prepared by Hailemikael Tesfaw Haile, titled: ***A Model for Recognition and Detection of the Counterfeit of Ethiopian Banknotes using Transfer Learning*** and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Software Engineering complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:	Ayalew Belay (PhD)	______________	_________________
Examiner:	Yaregal Assabie (PhD)	______________	_________________
Examiner:	Mesfin Kifle (PhD)	_______________	_________________

Abstract

Paper currency recognition systems play a pivotal role in various sectors, including banking, retail, and automated teller machines (ATMs). This paper presents a novel approach to the design and development of a paper currency recognition system using customized deep learning techniques. The proposed system utilizes image-processing algorithms to extract features from currency images, followed by customized convolutional neural network models for classification and detection of the counterfeit. The system is trained on a diverse dataset of currency images to ensure robustness and accuracy in recognizing various denominations and currencies. We implemented feature learning techniques architectures. To obtain the best accuracy and efficiency we used RLUs and Softmax as an activation, Adam optimizer, and sparse categorical cross-entropy as a loss function for both as a training strategy. The data was collected from the National Bank of Ethiopia, Commercial Bank of Ethiopia, NIB International Bank, and Bank of Abyssinia. From the experimental results of the alex_customed-design network, **99.82%** accuracy is recorded.

Keywords: Paper currency, Deep learning, Machine learning, Artificial intelligence, AlexNet

Acknowledgment

First and foremost, I express my deepest gratitude to God, whose grace, guidance, and blessings have been the cornerstone of my journey throughout this thesis. I am profoundly thankful for the strength, wisdom, and inspiration entrusted to me, enabling me to undertake and complete this effort.

I extend my heartfelt gratitude to my advisor, Ayalew Belay (PhD), for his unwavering support, invaluable guidance, and profound insights throughout this research. His advice has been instrumental in shaping this thesis and my academic journey. I would like to thank Yaregal Assabie (PhD), whose insightful comments and valuable input during the proposal stage influenced the direction of this thesis.

I am deeply thankful to the staff of the National Bank of Ethiopia, Commercial Bank of Ethiopia, NIB Bank, and Abyssinia Bank for their support. I would also like to thank my family for their unconditional love, encouragement, and understanding during the highs and lows of this effort.

I am grateful to my friends and colleagues for their encouragement, insightful discussions, and moral support throughout this journey. Their Friendship made the challenges more manageable and the successes more joyful.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acronyms and Abbreviations

ANN	Artificial Neural Networks
CDP	Counterfeit Detection Pen
CPOCNN	Class Probability Output Convolutional Neural Network**.**
CNN	Convolutional Neural Network
Co-ML	Co-Training Multi-Label Learning
DL	Deep learning
FCLF-CNN	Fully Connected Layer First CNN
GLCM	Gray Level Co-occurrence Matrix
HOG	Histogram of Oriented Gradients
KNN	K-Nearest Neighbors
LVQ	Linear Vector Quantization
LBP	Local Binary Patterns
LBP	Local Binary Pattern
ML	Machine Learning
PCR	Paper Currency Recognition
RF	Random Forests
R-CNN	Region-based Convolutional Neural Network**.**
RGB	Red Green Blue
RL	Reinforcement Learning
SIFT	Scale Invariant Feature Transform
SSL	Semi-supervised Learning
SURF	Speed-Up Robust Features
SVM	Support Vector Machine
TCSPC	Time-correlated Single Photon Counting
UV	Ultraviolet

Chapter 1. Introduction

1.1 Background

Within our interconnected world, countless products are manufactured across various countries, each contributing its unique goods. These products traverse borders, reaching both urban and rural areas globally. To facilitate this exchange, a diverse array of tools is essential. Among these tools are digital and cash-based transaction methods. In the realm of cash transactions, paper currency and coins serve as primary instruments.

There are around 180+ currencies presently circulating in the world. Each of these currencies differs greatly in features such as size, color, shape, density, and arrangement and also differs in their security features. Unlike the olden times, trade and commerce between countries has increased at all sorts of levels. The need for acquiring knowledge about all the currencies by many sector offices such as transaction machines, ticket counters, shopping in malls, and banknote exchange services has been extremely important [1]. In addition to this, the World Health Organization (WHO) in 2002 estimated blind or visually impaired people to be around 161 million with a concentration in developing countries [2]. Therefore, acquiring knowledge for this population is needed.

However, counterfeit notes are one of the biggest problems occurring in cash transactions. For countries like Ethiopia, it is becoming a big problem. Because of the advances in printing and scanning technologies, it is easily possible for a person to print counterfeit notes with the use of the latest hardware tools. Detecting counterfeit notes manually becomes a time-consuming and untidy process [3], and also blind or visually impaired people can't identify whether a banknote is genuine or not. As a result, there is a need for automation techniques that can efficiently perform currency recognition.

With the development of modern banking services, automatic methods for paper currency recognition have become important in many applications, such as automated teller machines and automatic goods seller machines [4].

Paper currency recognition (PCR) is an important area of pattern recognition. A system for the recognition of paper currency is one kind of intelligent system that is a very important need of the current automation systems in the modern world of today. It has various potential applications, including electronic banking, currency monitoring systems, money exchange machines, etc.[5]. Therefore, individuals, groups, and governmental and non-governmental organizations involved could enjoy the benefits of this research.

1.2 Motivation

Every country worldwide issues its own paper and coin currencies, essential for daily economic transactions. Ethiopia currently utilizes both forms of currency. However, the proliferation of counterfeit currency is on the rise, leading to a surge in its circulation within the economy. Consequently, this undermines the economic stability of nations.

The primary drive is to design a model and develop a model aimed at recognizing genuine currency and identifying counterfeit notes. This endeavor encompasses various sector offices, including transaction machines, ticket counters, malls for shopping, banknote exchange services, and similar entities worldwide.

1.3 Statement of the Problem

The way of extracting the most qualified monetary characteristics features from paper currency images is an important problem area that needs a solution today[6]. Various studies have been conducted by various scholars in various parts of the world to propose a solution for detecting counterfeit and recognizing the denomination value of various countries' banknotes.

Hasanuzzaman et al.[7] proposed a component-based framework to automatically recognize banknotes of USA dollars (USD) by using Speeded Up Robust Features (SURF). Lamont et al. [8]proposed a recognition method for Mexican banknotes using artificial vision. Omatu et al.[9] proposed a method based on principal component analysis, a linear vector quantization (LVQ), and network to classify banknotes of USA dollars (USD). Dunai et al. [10] Proposed the detection and recognition of Euro banknotes using the haar technique and SURF. Rashid et al. [11]proposed a support vector machine (SVM), artificial neural network, and hidden Markov model (HMM) to recognize and classify USA dollars (USD) and Euro banknotes. Pham et al. [12]proposed a method to classify banknotes of USA dollars (USD), South African rand (ZR), Angola kwanza (AOA), and Malawian kwacha (MWK) using a K-base mean classifier. Doush et al. [13] proposed a banknote recognition system based on the scale-invariant feature transform (SIFT) algorithm for the Jordanian currency. Kogilawanee et al.[14] proposed recognizing and classifying Sir Lankan currency notes for visually impaired people based on an augmented approach. Alnowaini et al.[15] proposed detecting and recognizing the currency system for Yemeni paper currency using preprocessing techniques. Mittal et al.[16] proposed a banknote recognition system for Indian banknotes using a convolutional neural network. Jegnaw Fentahun and Yaregal Assabie [17] proposed a solution for the recognition and detection of the previous Ethiopian paper currency based on the four characteristic features of the banknotes such as the dominant color, the distribution of the dominant color, the hue value, and speeded up robust features.

According to the above studies, the scholars presented different methods to classify the denomination value of banknotes and identify whether the currency is genuine or not. The banknotes of various countries have different characteristic features. Consequently, the classification methods of different scholars in their proposed solution were based on the types of characteristics features, and the complexity of the problem. Based on such variations in characteristic features, and the automatic paper currency recognition system invariably depends on the characteristic feature that comprises the banknotes and feature extraction [14], none of the methods guarantees that it will work effectively across all currencies.

As a result, the goal of this thesis work is to propose deep learning approaches for the recognition and verification of the new Ethiopian banknotes due to its ability to execute feature extracting on its own and having an architecture that can be adapted to the new problems relatively easily.

1.4 Objectives

General Objectives

The main objective of this study is to design and develop a model that recognizes and detects counterfeit Ethiopian Banknotes using a deep learning approach.

Specific Objectives

To achieve the general objective, identify the following specific objectives

- Conduct a literature review on paper currency recognition and counterfeit detection systems.
- Collect genuine and counterfeit images of Ethiopian Banknotes.
- Design and develop a model that recognizes and detects counterfeit Ethiopian banknotes.
- Develop the prototype for the model.
- Evaluate the developed model through the prototype and test data.

1.5 Methodology

The design science research methodology has found very good ground as a method in information science and computer science because it is a method that works with human, organizational, and social types of problem-solving through artifact development.

In this thesis, the researchers chose a design science research approach that this thesis intends to propose a solution and repeatedly improve the solution through an iterative process.

The research will pass through multiple iterations of problem identification, analysis, component/model identification, component specification, prototype development, and evaluation. Each iteration will raise a new question and uncover new problems that need to be solved in the following iterations.

After multiple iterations of the design model, a working design model for the recognition and detection of counterfeit notes will be developed as an artifact.

To achieve the specified objectives, the following list of methodologies will be employed.

Literature Review: A literature review on books, journals, and other publications will be used throughout the research process to better understand the problem the research is trying to solve and to explore better methods, techniques, and tools that can help achieve the research goals.

Data collection: To achieve the main goal of this research, we collected 5274 genuine banknotes and fake notes from the National Bank of Ethiopia, Commercial Bank of Ethiopia, Bank of Abyssinia, NIB International Bank, Zemen Bank, and other relevant source.

Tools and Development Environments: The prototype will be developed using Python programming language and other appropriate tools.

Testing and Evaluation: Implementing an experiment by using genuine, counterfeit, and previous Ethiopian paper currencies, and then the accuracy will be calculated by testing the model with these sample paper currencies.

1.6 Scope of the Study

The scope of this study is to recognize the new Ethiopian Paper Currency and detect Counterfeit.

1.7 Significance of the Study

The result of this study will be useful for many sector offices or organizations. These are:-

- Transaction machines, ticket counters, shopping in malls, and Banknote exchange services used the result of this research for counting, arranging, and detecting counterfeit notes easily and efficiently.
- It will also be used by regulatory bodies to control counterfeits and to prepare the rules and regulations that help to minimize the distribution of counterfeits.
- The researchers used this study as a reference to improve the system or minimize the problems that have not been covered yet.

1.8 Organization of the Rest of the Thesis

The remaining chapters of this study will be organized into five chapters. The literature review will be covered in Chapter Two. Chapter Three will present the related work. Chapter Four will introduce the proposed model and its components. Chapter Five will discuss the experimental evaluation, the performance of the proposed system, and the overall discussion. Finally, Chapter Six will conclude the study and outline future work.

Chapter 2. Literature Review

2.1 Introduction

This chapter effort to clarify the foundational basis of the present research endeavor. It will commence with a comprehensive examination of paper currency, encompassing an exploration of counterfeit currency, the currency recognition system, and the traditional method of Ethiopian currency recognition. Subsequently, the chapter will provide a detailed overview of machine learning and deep learning approaches, including an analysis of various types of deep learning architectures and feature extraction techniques. Finally, a summary will encapsulate the key insights extracted from these discussions.

2.2 Currency Recognition System

Currency is a medium of exchange that is widely accepted in transactions for goods and services. It typically takes the form of coins and paper banknotes issued by a government or central authority. The characteristics of currency include physical attributes such as size, color, and pattern, which vary significantly across different countries and denominations [18]. Currency recognition systems are technological solutions designed to identify and classify different types of currency, often leveraging image processing and machine learning techniques to analyze these physical attributes [18].

Researchers develop various currency recognition systems to address the diverse challenges presented by the wide array of global currencies and their conditions in circulation. These systems must be capable of handling banknotes that are new, worn, torn, or defaced, and must accurately recognize currencies from different countries, which may be intermixed in financial institutions [18, 19].

Additionally, the development of such systems is driven by the need to assist specific populations, such as visually impaired individuals, by providing them with tools to independently identify banknotes [20, 21].

The systems must also be robust against various environmental factors such as occlusion, rotation, scaling, and changes in illumination [20].

Currency recognition systems are developed to facilitate the accurate and efficient processing of banknotes in various applications, from aiding the visually impaired to enhancing the operations of financial institutions. The development of these systems is necessitated by the complexity of handling multiple currencies that may be in varying states of wear and by the need to operate reliably under diverse conditions. Different researchers focus on creating systems tailored to specific currencies, conditions, and user requirements, employing advanced techniques in image processing and machine learning to achieve high accuracy and robustness [19, 21].

2.3 Banknote Recognition and Classification Practice in Ethiopia

The propagation of counterfeit notes has surged alongside genuine currency circulation. Consequently, to fight counterfeiting or mitigate its risks, an abundance of techniques, such as first-line and second-line methods, have been adopted globally. In Ethiopia, banking experts, retailers, and vendors predominantly employ the first-line inspection manual method, leveraging their extensive experience [22]. This method entails examining common security features of banknotes, including the security thread, golden strips (predominantly found in 10, 50, 100, and 200 denominations), and numeric denomination inscriptions in Amharic and Geez as shown below in Figure 2.1, Figure 2.2, Figure 2.3, and Figure 2.4. Despite its widespread use, the first-line inspection method is susceptible to human errors [23].

In contrast, the second-line inspection method offers enhanced security but necessitates additional equipment, thereby escalating manufacturing and verification costs. This method encompasses automatic and semi-automatic inspection techniques commonly utilized worldwide for banknote authentication. Semi-automatic techniques typically involve manual feeding or supervision by an operator. Counterfeit detection pens and fluorescence detection are examples of semi-automatic techniques. In contrast, ultraviolet (UV) detection exemplifies automatic techniques, as it can independently count and dispense cash without needing manual intervention for denomination recognition [22].

Presently, Ethiopian financial institutions such as the Commercial Bank of Ethiopia, as well as non-governmental banks like Nib Bank and Abyssinia Bank, routinely employ counterfeit detection pens, ultraviolet light detectors, and fluorescence for authenticating Ethiopian banknotes [23].

Figure 2. 1 Image of the front side of two hundred birr

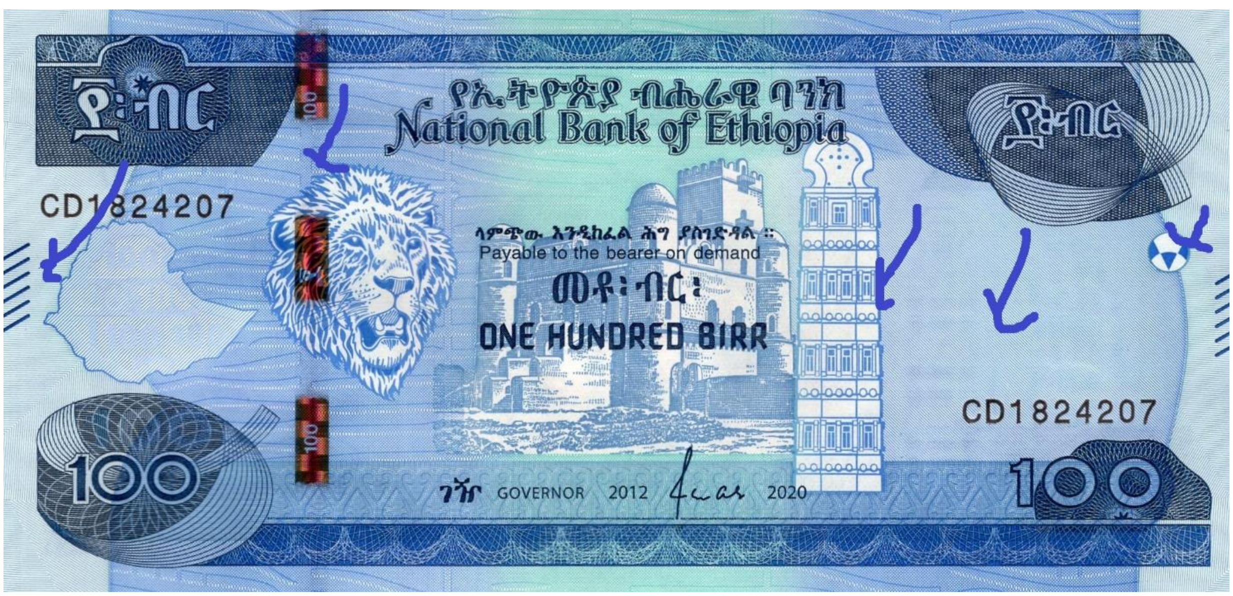

Figure 2. 2 Images of the front side of one hundred birr

Figure 2. 3 Images of the front side of fifty birr

Figure 2. 4 Images of the front side of ten birr

2.3.1 Counterfeit Detection Pen

A counterfeit detection pen (CDP) is a basic tool designed to identify counterfeit banknotes by exploiting the presence of starch in standard printing paper, a characteristic absent in genuine currency composed of cotton or linen fibers. The pen contains an iodine solution that reacts visibly when applied to suspected banknotes: if the note is counterfeit and contains starch, the iodine produces a dark mark. In contrast, genuine banknotes typically exhibit no reaction or a faint, fleeting mark due to the absence of starch. While the CDP provides a quick and straightforward means to flag potential forgeries, it has notable limitations. It fails to detect high-quality counterfeits crafted from starch-free paper, thereby rendering it ineffective against sophisticated counterfeit operations. Furthermore, the pen may yield false positives when authentic banknotes inadvertently acquire starch contamination, leading to erroneous suspicions of forgery. As such, while useful for initial screening, the reliance on starch detection necessitates caution and supplementary verification methods to enhance the reliability of counterfeit detection efforts [24].

2.3.2 Ultraviolet Counterfeit Detection Scanner (UV)

Ultraviolet (UV) counterfeit detection scanners operate by exposing banknotes to UV light, which causes certain security features on genuine currency to fluoresce. These features, such as security threads or intricate patterns, emit distinct UV responses that are challenging counterfeiters to replicate accurately. The scanners are designed to detect these UV-responsive elements, providing a reliable method to distinguish genuine banknotes from counterfeits. Despite their effectiveness, UV scanners have limitations. They may fail to identify counterfeits that incorporate UV-reactive printing to mimic genuine features. Additionally, operators must own knowledge of the specific UV characteristics inherent to each currency they examine, as variations exist among different denominations. Therefore, while UV scanners offer robust detection capabilities based on UV fluorescence, their effectiveness depends on user expertise and the ability to perceive refined UV signatures across various banknotes [25, 26].

In Ethiopia banknotes, there are security features, which are made by using UV-printed images, and they can be visualized by being subjected to UV light only. However, it is also affected by noise is not fully automated, and requires human intervention to recognize the banknote [27].

2.3.3 Fluorescence Detector

Fluorescence detectors, particularly those employing time-correlated single photon counting (TCSPC), are sophisticated tools used to analyze the fluorescence properties of inks used in authentic banknotes. For instance, U.S. Federal Reserve Notes typically display a distinct two-component intrinsic fluorescence lifetime pattern that differs from counterfeit counterparts [28]. These detectors are exceptionally sensitive and capable of distinguishing genuine currency from counterfeits by examining the precise decay profiles of fluorescence emitted by the ink.

Despite their effectiveness, fluorescence detectors are characterized by their complexity and higher cost compared to alternative counterfeit detection methods. Additionally, their operation may necessitate skilled personnel trained to accurately interpret the complex results provided. This requirement underscores their reliance on expert analysis to ensure reliable counterfeit detection and to mitigate potential misinterpretations. Thus, while fluorescence detectors offer a powerful tool for verifying the authenticity of banknotes based on fluorescence characteristics, their deployment typically involves investment in specialized equipment and knowledgeable personnel to achieve optimal results [28].

2.4 Machine learning algorithms

Machine learning (ML) is a subset of artificial intelligence that employs algorithms and statistical models to help computers enhance their performance on tasks through experience. ML works by identifying patterns in data and making decisions or predictions based on those patterns without being explicitly programmed to perform the task [21].

The process of creating the machine-learning model involves selecting an appropriate algorithm, providing it with training data, and allowing the model to learn from that data. The training data must be representative of the real-world phenomena the model is intended to capture and it is used to adjust the model's parameters. Once trained, the model can then be tested using a separate dataset to ensure its accuracy and generalizability[29]. There are contradictions and interesting facts to consider. For instance, while ML models are powerful, they can also be vulnerable to attacks such as membership inference attacks, which can extract information about the training data, posing security and privacy risks.

Additionally, the quality and diversity of the training data are crucial for building inclusive ML systems, and tools like Co-ML have been designed to promote learning of dataset design practices [30]. Moreover, ML models can be affected by noisy data, which can degrade performance, but novel training approaches like co-teaching can mitigate this by using both noisy and noise-free data.

Machine Learning is a dynamic field with a broad range of applications across industries. For example, in the legal sector, ML can automatically identify industry sector coverage in articles, aiding both readers and content creators. In the context of Industry 4.0, ML algorithms are used for predictive analysis in systems to maintain optimal production environments [31]. Furthermore, ML techniques are employed for predictive maintenance in manufacturing, enhancing efficiency and safety[32].

In the building sector, ML models are integrated into model predictive controllers for energy and cost savings, with different model types and workflows being explored for practical applicability [33]. These applications demonstrate the versatility and impact of ML in real-world industrial settings.

Conventional machine learning algorithms encompass a variety of data-driven techniques used for tasks such as classification, prediction, and pattern recognition. These traditional algorithms include, but are not limited to, K-Means, Support Vector Machines (SVM), Naïve Bayes Classifier Algorithm**,** k-nearest Neighbors (KNN), and Random Forests (RF) [34] and are widely applied across different domains, from facial recognition to healthcare and renewable energy forecasting.

K-Means

The K-Means algorithm represents a fundamental method in unsupervised learning, specifically designed for clustering datasets into K distinct and non-overlapping groups. Unlike supervised learning, it operates without prior knowledge of the labels assigned to individual data points. The algorithm functions through an iterative process where it begins by randomly assigning each data point to one of the K clusters. It then calculates the centroid, or mean value, of the points within each cluster and reassigns each data point to the cluster whose centroid is closest in terms of distance metrics, commonly Euclidean distance. K-Means is effective for clustering applications where the number of clusters K is known or can be reasonably estimated. It efficiently organizes data points into clusters based on their similarity in feature space, providing insights into inherent patterns or structures within the dataset. However, its performance can be sensitive to the initial selection of cluster centroids and is influenced by the distribution and scaling of the data. Moreover, the algorithm assumes clusters that are spherical and of roughly equal size, which may not always align with the actual distribution of data in more complex datasets. Therefore, while K-Means is a powerful tool for exploratory data analysis and segmentation tasks, careful consideration and preprocessing of data are essential to maximize its effectiveness in practical applications [35].

The Support Vector Machine (SVM)

The Support Vector Machine (SVM) stands as a versatile supervised learning model employed extensively for both classification and regression tasks. Its core principle revolves around identifying the optimal hyperplane that effectively divides data points belonging to different classes within the feature space. The SVM achieves this by aiming to maximize the margin, which refers to the distance between the hyperplane and the nearest data points from each class. By maximizing this margin, the SVM not only enhances robustness against overfitting but also promotes better generalization to unseen data.

In scenarios where classes are not linearly separable in the original feature space, SVMs can employ kernel functions to transform the data into higher-dimensional spaces where linear separation becomes feasible. These kernel functions enable SVMs to handle complex, non-linear decision boundaries effectively. Popular kernel functions include the polynomial kernel, radial basis function (RBF) kernel, and sigmoid kernel, each tailored to different types of data distributions and separation requirements.

Despite its strengths, SVMs may encounter challenges with large datasets due to their computational complexity, particularly in scenarios with numerous features or data points. Moreover, the performance of SVMs can be sensitive to the choice of kernel and regularization parameters, necessitating careful tuning to optimize model performance. Nonetheless, SVMs remain a robust choice for classification tasks across various domains, offering a balance between effective separation of classes and flexibility in handling diverse data distributions through kernel methods [35].

Naïve Bayes Classifier Algorithm

The Naive Bayes Classifier Algorithm applies Bayes' theorem assuming conditional independence between all features given the class variable, making it efficient for classification tasks, even if this assumption is not strictly accurate. It computes class probabilities based on training data and assigns the class with the highest probability to new instances. Despite its reliance on simplification, Naive Bayes achieves high accuracy in many practical cases where features are mostly independent given the class. However, it may struggle with highly correlated features or complex interactions, where more sophisticated models could perform better. Nonetheless, its computational efficiency and effectiveness with large datasets make Naive Bayes valuable, especially when paired with robust feature selection and preprocessing methods to mitigate its assumptions' impact [36].

The K-Nearest Neighbors (K-NN) algorithm

The K-Nearest Neighbors (K-NN) algorithm is a non-parametric method used for both classification and regression. It operates by predicting labels or values based on the similarity of data points in the feature space, assuming that similar data points share common characteristics. In classification, K-NN assigns a class label to a data point based on the majority class of its k nearest neighbors, determined using distance metrics like Euclidean distance. For regression tasks, it estimates a target variable by averaging values from its k-nearest neighbors. K-NN is straightforward to implement but sensitive to parameters such as k and distance metrics, impacting decision boundaries and prediction accuracy. As a lazy learning algorithm, K-NN stores training data and performs predictions only when new data points require classification or regression, making it suitable for datasets with unknown distributions or non-linear relationships [36].

Random Forest

Random Forest is a versatile ensemble learning method used in machine learning for classification, regression, and diverse tasks. It constructs multiple decision trees during training and combines their predictions to determine the final output. In classification, the most frequent prediction among the trees is chosen, while in regression, predictions are averaged. This approach mitigates overfitting inherent in individual decision trees by training each tree on different data subsets and using random feature sets. This randomness enhances model generalization and robustness to new data. Random Forests accommodate complex data relationships, including non-linearities and feature interactions, and are resilient to outliers and noise. They also provide insights into feature importance, aiding in model interpretation. However, their computational demands and sensitivity to hyperparameters necessitate careful tuning for optimal performance, underscoring their effectiveness in delivering accurate results across various machine-learning applications [36].

2.5. Deep Learning Algorithms

Deep learning (DL) is a section of Artificial intelligence (AI). DL employs artificial neural networks with multiple layers to model complex patterns in data. While ML can utilize a variety of algorithms for learning from data, DL specifically relies on deep neural networks, which are capable of learning unsupervised from unstructured or unlabeled data [37].

Deep learning differs from traditional ML in its ability to automatically learn and improve from experience without being explicitly programmed. It excels at identifying patterns in data with minimal human intervention, thanks to its deep architecture that can learn hierarchical feature representations. In contrast, ML is often manual feature extraction and is typically not as effective at handling high-dimensional data or complex tasks like image recognition [19]. Research areas of Deep learning include a wide range of applications such as computer vision, natural language processing, and speech recognition. In computer vision, DL has been instrumental in advancing image segmentation, object detection, and image classification [38, 39].

Image segmentation includes dividing an image into segments to simplify or alternate the illustration of an image into something extra significant and simpler to analyze. DL models, particularly convolutional neural networks (CNNs), have been widely used for efficient and automatic image segmentation [19].
Object detection is another area where DL has made significant strides, with architectures like R-CNN bringing DL into the forefront of object detection research.

Image classification, which involves assigning a label to an entire image, has also benefited from DL, especially in tasks like hyperspectral image classification.

Deep learning is a specialized form of machine learning that uses deep neural networks to learn from data. It differs from traditional ML in its ability to learn feature representations automatically and handle complex tasks with high-dimensional data. DL has made significant contributions to various research areas, particularly in computer vision for tasks such as image segmentation, object detection, and image classification [19, 40].

2.5.1 Types of Deep Learning Approach

Deep learning (DL) approaches can be broadly categorized into supervised, semi-supervised, and unsupervised learning, each with distinct methodologies for training models.

Supervised Learning Algorithm

Supervised learning algorithms form the backbone of machine learning, aiming to learn mappings from input data to output labels using labeled datasets. These algorithms rely on comprehensive datasets where each input is paired with its correct output, enabling them to discern underlying patterns and relationships during training. By analyzing these pairs, supervised learning algorithms generalize their understanding to accurately predict outputs for new, unseen data. This approach contrasts with unsupervised learning, which operates without predefined output labels to uncover inherent structures. Techniques in supervised learning include regression for continuous predictions and classification for discrete labels, employing models like linear regression, decision trees, support vector machines, and neural networks. The success of supervised learning hinges on dataset quality, representation, and the algorithm's ability to generalize to new examples. It finds practical use in diverse domains such as image recognition, speech processing, medical diagnostics, and recommendation systems, crucial for making informed decisions and automating tasks effectively [41].

Unsupervised Learning Algorithm

Unsupervised learning differs from supervised learning by focusing on datasets lacking predefined labels. Instead of aiming to predict specific outputs, unsupervised learning seeks to uncover inherent structures or patterns within the data itself. This approach is pivotal for tasks where understanding the natural organization or distribution of data is paramount, achieved through techniques such as clustering or density estimation.

Clustering involves grouping similar data points into clusters based on their proximity in the feature space, revealing natural groupings or segments within the dataset. Density estimation, on the other hand, involves estimating the probability density function of the data to understand its distribution across the feature space.

Unlike supervised learning, which relies on labeled examples for training, unsupervised learning operates independently of explicit guidance, making it versatile for exploring and discovering insights in diverse datasets. By uncovering hidden relationships or structures, unsupervised learning facilitates tasks such as anomaly detection, pattern recognition, and dimensionality reduction, contributing crucially to fields like data exploration, market segmentation, and image processing where unlabeled data prevails. However, the evaluation of unsupervised learning models can be challenging due to the absence of ground truth labels, necessitating robust methodologies to assess model performance and interpret results effectively [41].

Semi-supervised Learning (SSL) Algorithm

Semi-supervised learning (SSL) combines labeled and unlabeled data to enhance learning accuracy, particularly valuable when fully labeled datasets are costly or impractical to obtain. SSL algorithms utilize the limited labeled data to guide learning and extrapolate patterns across the unlabeled data, improving overall model performance. Methods like graph-based techniques and support vector machines are common in SSL, leveraging relationships and iterative boundary improvements to effectively classify unlabeled instances. This approach addresses the challenges of traditional supervised learning by optimizing labeling resources and leveraging unlabeled data for robust, generalized models [41, 42]. SSL finds application in critical domains such as image recognition, natural language processing, and medical diagnostics, where accurate predictions are crucial despite incomplete data annotation.

Reinforcement Learning (RL) Algorithm

Reinforcement learning (RL) is distinct from the aforementioned learning paradigms. RL is concerned with how agents ought to take actions in an environment to maximize some notion of cumulative reward. The agent learns to achieve a goal in an uncertain, potentially complex environment. In RL, an agent makes observations and takes actions within an environment, and in return, it receives rewards. Its objective is to learn a policy that maximizes the expected cumulative reward over time [41].

2.5.2 Deep Learning Architecture

Deep learning models include artificial neural networks (ANN), autoencoders, RNNs, and RLs. The most diffused DL architectures are Convolutional Neural Networks (CNN), which can classify images into several categories, automatically learning features through convolutional layers that combine multiple non-linear processes. It can learn effective hierarchical feature representations that characterize the typical variations observed in visual data, including images and video, which makes them very well suited for most of the visual classification tasks.

Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are a class of deep learning algorithms predominantly used in processing data that has a grid-like topology, such as images. CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input data, which is essential in tasks like image and video recognition, image classification, and natural language processing [43]. CNNs consist of multiple layers that each perform different transformations on their inputs.

Convolutional layer

The convolutional layer is a fundamental element in Convolutional Neural Networks (CNNs), essential for extracting intricate features from input data, especially in image analysis. It operates by applying kernel filters that convolve across the input, detecting spatial patterns and relationships crucial for tasks like classification and regression. Each filter computes dot products with input regions, preserving spatial dependencies to capture edges, textures, and higher-level features throughout the input. Stacking multiple convolutional layers allows CNNs to learn increasingly abstract features, improving their ability to generalize and predict accurately on new data. Key parameters like filter size, stride, and padding adjust the output's spatial dimensions, influencing network complexity and performance. This layer's capability to extract hierarchical representations underscores its vital role in deep learning, powering applications across computer vision, speech recognition, and natural language processing [43].

Pooling layer

The pooling layer in Convolutional Neural Networks (CNNs) is strategically placed after convolutional layers to compress the spatial dimensions of feature maps. This process, achieved through operations like max pooling or average pooling with small window sizes such as 2x2 or 3x3, reduces the number of parameters and computational workload in the network. By aggregating information from local regions, pooling preserves essential features while enhancing the network's ability to generalize and handle variations in input data. Additionally, pooling introduces spatial invariance, enabling the CNN to recognize patterns irrespective of their precise location within the input. This critical component not only aids in preventing overfitting but also improves the efficiency and robustness of CNNs in tasks such as image recognition, object detection, and semantic segmentation [44].

Fully connected layer

The fully connected layer in Convolutional Neural Networks (CNNs) is crucially positioned at the network's end to synthesize and utilize features extracted by preceding convolutional and pooling layers for final classification or regression tasks. Beginning with the convolutional layer, multiple filters apply to input data, generating linear activations and capturing spatial patterns.

In the subsequent pooling layer, down sampling like max or average pooling reduces feature map dimensions, preserving essential details while optimizing computational efficiency [44]. The fully connected layer then integrates these processed features, connecting every neuron densely to each in the previous layer. This allows the network to learn intricate feature relationships and make comprehensive decisions. For instance, in classification tasks, it computes class probabilities from combined features, enabling accurate outputs like class labels [43]. Overall, the fully connected layer represents the last stage where hierarchical input features are synthesized for precise CNN predictions, pivotal in applications like image recognition requiring robust classification capabilities.

In practice, the architecture and performance of CNNs can be significantly enhanced by optimizing the convolution parameters, as demonstrated by the Convolution parameters optimization for CNNs (CPOCNN) model, which adapts the upper bounds of convolution parameters to improve classification performance. Additionally, innovations like fully connected layer first CNN (FCLF-CNN) have shown that reordering the layers can lead to improved performance on structured data [19]. Figure 2.5 shows the three types of CNN layers: convolutional layers, pooling layers, and fully connected layers as follows

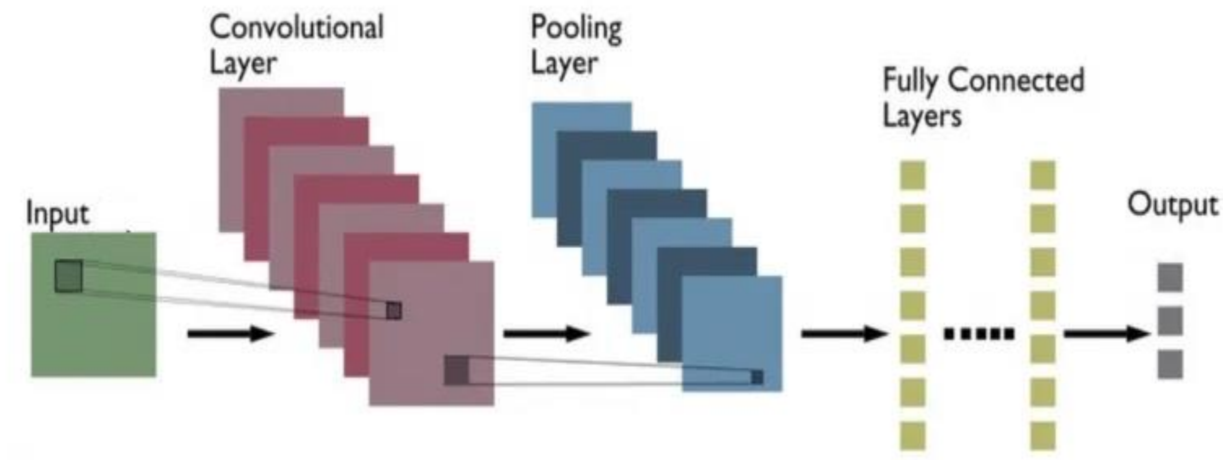

Figure 2. 5 Architecture of CNN

2.6 Feature Extraction

Feature extraction plays a vital role in the paper currency recognition system, involving the identification and extraction of specific characteristics or attributes from currency images to aid in their classification [45]. These features can include textural patterns, color, size, and other unique identifiers that are inherent to each currency note. The extracted features are then used to train classifiers, such as k-nearest Neighbor (k-NN), neural networks, or Support Vector Machines (SVMs), which can distinguish between different denominations and authenticate banknotes [45, 46].

Interestingly, while the general principle of feature extraction remains consistent, the specific features and methods used can vary significantly. For instance, the Gray Level Co-occurrence Matrix (GLCM) is employed for textural analysis in one study while another uses the Histogram of Oriented Gradients (HOG) for feature extraction [46].

Additionally, deep learning approaches have been applied, utilizing pre-trained models like AlexNet and ResNet-50, which can automatically learn to identify relevant features from data. Feature extraction is a fundamental component of paper currency recognition systems, enabling the conversion of raw currency images into a set of representative features. These features are then used by various classifiers to accurately identify and authenticate currency notes. The effectiveness of these systems is evidenced by high recognition accuracies reported in the literature, demonstrating the practical applicability of these methods in real-world scenarios [45].

2.6.1 AlexNet

AlexNet is a deep convolutional neural network (CNN) architecture that has been influential in the field of computer vision, particularly in image recognition tasks. It operates by automatically extracting features through convolutional layers and classifying images with fully connected layers, a process that differs from traditional machine learning methods, which often require manual feature extraction [47].

AlexNet's structure includes multiple layers that apply filters to input images, pooling layers that reduce dimensionality, and fully connected layers that interpret the features to make predictions [48]. AlexNet's performance is generally characterized by high accuracy in image recognition tasks. For instance, in fruit recognition, AlexNet demonstrated excellent accuracy, although it was slower compared to a conventional CNN [49].

AlexNet, an innovative CNN architecture, has transformed image recognition by showcasing remarkable accuracy across diverse applications [50]. Its layered approach to feature extraction and classification marks a departure from conventional methods that depend on manual feature extraction and selection. Furthermore, its adaptability and capacity for specific tasks underscore its versatility and effectiveness in achieving high performance in image recognition [51, 52].

2.6.2 ResNet-50

ResNet-50 is a deep convolutional neural network architecture that is widely used for image classification and feature extraction tasks. It is part of the ResNet (Residual Network) family, which was introduced to address the problem of vanishing gradients in deep neural networks. The key innovation of ResNet-50 is its use of residual connections, also known as skip connections or shortcut connections. These connections allow the network to learn residual functions with reference to the layer inputs, rather than learning unreferenced functions [20].

ResNet-50 Used a stack of residual blocks, each containing convolutional layers, batch normalization, and ReLU activation functions, employing shortcut connections that skip one or more layers, allowing the network to learn both simple and complex features and utilizing a global average pooling layer followed by a fully connected layer for final classification [19]. For feature extraction, ResNet-50: Processes input images through its convolutional layers, extracting hierarchical features from low-level to high-level representations, automatically learns and extracts relevant features without human supervision, making it highly effective for various image analysis tasks and can be used as a pre-trained model, allowing transfer learning for specific tasks by fine-tuning the network on domain-specific datasets.

2.7 Summary

In this chapter, a comprehensive review of the literature is presented, covering various aspects related to paper currency, counterfeit detection, traditional and proposed methods for Ethiopian currency recognition systems, as well as an exploration of machine learning and deep learning algorithms.

Firstly, an overview of paper currency and the challenges associated with counterfeit detection is provided. This includes discussions on the importance of currency security measures and the economic implications of counterfeit currency circulation.

Next, both traditional and proposed methods for Ethiopian currency recognition systems are examined. This involves an analysis of existing approaches, their strengths, limitations, and areas for improvement. Additionally, the chapter provides a detailed explanation of various machine learning algorithms such as K-Means, Support Vector Machine (SVM), Naïve Bayes Classifier Algorithm, K-NN (K-Nearest Neighbors), and Random Forest. Each algorithm's underlying principles, applications, and suitability for currency recognition tasks are explained, providing readers with a comprehensive understanding of their capabilities
Furthermore, the concept of deep learning is defined and its relationship with machine learning and artificial intelligence is explored. The chapter explains the distinctions between these domains and highlights how deep learning techniques have revolutionized pattern recognition tasks, including currency recognition.

The discussion extends to the three primary approaches within deep learning: supervised learning, semi-supervised learning, and unsupervised learning. Additionally, Deep Reinforcement Learning (DRL), a less familiar category of learning approach, is explored in detail, shedding light on its potential applications in currency recognition systems.
In addition, the chapter provides a comprehensive overview of the layers that constitute Convolutional Neural Networks (CNNs), explaining their functions and contributions to the network's overall architecture.

Overall, this chapter serves as a foundational resource for understanding the landscape of currency recognition systems, machine learning, and deep learning algorithms, and the methodologies employed in contemporary research efforts within this domain.

Chapter 3. Related Works

3.1 Related Works on Paper Currency Recognition

In this Chapter, various research works done in the area of paper currency denomination classification and verification the validity are presented. There are several researches done on paper currency denomination and recognition of various countries' currencies. However, to the best knowledge of the researchers, no research has been done till now for automatic recognition of the new Ethiopian paper currency. Because of this reason, the review is focused on research done on other countries and the previous Ethiopian currency recognitions.

3.1.1 Ethiopian Paper Currency Recognition System

Jegnaw Fentahun and Yaregal Assabie [17] proposed an automatic recognition and counterfeit detection system for the previous Ethiopian paper currency. This research work considered the four characteristic features of the banknotes such as the dominant color, the distribution of the dominant color, the hue value, and speeded-up robust features were extracted as the discriminative features of banknotes. Based on this research result, the denomination accuracy for genuine previous Ethiopian paper currency, and counterfeit currencies was found to be 90.42% and 83.3% respectively, and the verification accuracy of the system is 96.13%.

Dinku et al. [22] presented a case study on the development of a Counterfeit Currency Identification System (CCIS) specifically focusing on Ethiopian Birr notes. The system utilizes the Cauchy-Schwarz inequality algorithm to distinguish between genuine and fake currency notes. The process involves collecting genuine and fake currency samples, processing their images using digital scanners and cameras, and comparing them using the algorithm. The system sets a threshold value of 0.80 to determine the authenticity of the notes. The study concludes that the proposed methodology shows promise in identifying counterfeit Ethiopian 100 Birr notes and highlights the importance of factors like paper quality and note condition in the identification process.

Asfaw Alene and Million Meshesha [23] proposed the Ethiopian Paper Currency Recognition System aims to address the issue of forged banknotes in Ethiopia by developing an optimal feature extraction system. The study explores techniques such as color momentum, SIFT, GLCM, and CNN for feature extraction, with FFANN as a classifier. The CNN feature extraction technique stands out with a high recognition accuracy of 99.4% in classifying Ethiopian banknote denominations. Additionally, for fake currency recognition, the CNN feature outperformed other techniques with an accuracy level of 96.46%. The study recommends further research on enhancing the CNN model with advanced architectures like GoogLeNet and ResNet using larger datasets to improve banknote recognition systems.

Tesfaw et al. [27] proposed "Ethiopian Banknote Recognition and Fake Detection Using Support Vector Machine" which focuses on developing a system that can accurately recognize Ethiopian paper currency and detect counterfeit notes. The system utilizes image-processing techniques such as image acquisition, preprocessing, feature extraction, and classification using a support vector machine. By capturing images of Ethiopian banknotes through cameras or scanners, enhancing image quality through preprocessing, and extracting relevant features using the local binary pattern technique, the system achieves a high accuracy rate of 98% in classifying Ethiopian paper currency denominations. Additionally, the system can verify the validity of given banknotes with an average accuracy rate of 93%.

Assefa et al. [53] focused on developing a system for Ethiopian banknote recognition using Convolutional Neural Network (CNN) technology and implementing a prototype on an embedded platform. The system aims to classify Ethiopian banknotes into denominations for use in automated machines like ATMs and vending machines. The study collected a dataset of Ethiopian currency images, tested various CNN architectures, and optimized the models for accurate recognition. The MobileNetV2 model, optimized using RMSProp, achieved 96.80% accuracy. The prototype system was implemented on a Raspberry Pi computer with a camera and a Web-based interface for real-time identification of banknotes. The research does not consider the newly introduced two hundred banknotes.

Gebremeskel et al. [54] focused on developing a model for detecting fake Ethiopian banknotes using deep learning, specifically a deep convolutional neural network (CNN) technique. The proposed model aims to differentiate between real and fake banknotes by analyzing computer vision features of digital content captured using smartphone cameras. By leveraging the capabilities of deep learning, the model demonstrates high accuracy in detecting fake banknotes. The study outlines the model architecture, including image acquisition, normalization, grayscale conversion, and histogram equalization, to optimize performance and reduce computational complexity during deployment and training. Through experimental iterations, the model achieved 97.6% training, validation, and testing accuracies, displaying its effectiveness in detecting counterfeit banknotes.

Woldehana et al. [55] focused on developing a novel approach using similarity learning for classifying counterfeit and genuine Ethiopian banknotes. The study addresses the persistent issue of counterfeit currency in Ethiopia by proposing a deep learning model that combines transfer and similarity learning techniques. The experimental results demonstrate high classification accuracy scores for the Inception and Vgg16 models, with the Inception model showing better performance in terms of distance score. The proposed model aims to enhance the detection of counterfeit banknotes, offering a more efficient and accurate solution compared to traditional methods. However, the focus appears to be on the front sides of the banknotes for the detection and classification of counterfeit and genuine Ethiopian currency. A small dataset was collected for 100 and 200 banknotes.

3.1.2 USA Dollars Currency Recognition System

Hasanuzzaman et al. [7] proposed a component-based framework to automatically recognize banknotes of USA dollars (USD) for visually impaired people. In this study, the speeded-up robust features (SURF) were applied to generate scale-invariant and rotation-invariant interest points with descriptors. To validate the robustness and generalizability of this proposed approach, the researchers collected a large dataset from a variety of environments, including bills taken with occlusion, cluttered backgrounds, rotation, and changes in illumination, scaling, and viewpoints, and achieved 100% accuracy on this dataset.

Omatu et al. [9]proposed a method based on principal component analysis (PCA) for increasing the reliability of the banknote recognition system. The researchers examined six different bill types of USA dollars (USD). The data is acquired through a line sensor, the PCA algorithm is used both to extract the main features and to reduce the size of the data. A linear vector quantization (LVQ) and network are applied as the main classifier of the classification of banknotes. In this study, hidden Markov models (HMMs) are also applied as an alternative classifier. The experimental results show that the reliability of the recognition system is increased up to 95% when the number of PCA components as well as the number of LVQ codebook vectors, are taken properly.

3.1.3 Mexican Banknotes Recognition System

Lamont et al. [8] proposed a recognition method for Mexican banknotes by using artificial vision. In this study, the scholars examined the five different types of Mexican banknotes and they performed nine different characterizations of the banknotes by combining color and texture features. RGB space model and the local binary patterns (LBP) are applied to examine the color and texture features of the banknotes, respectively. The average recognition rate of the proposed system is 97.50%.

3.1.4 European Banknotes Recognition System

Dunai et al. [10]proposed the detection and recognition of Euro banknotes for blind people using the haar technique and speed-up robust features (SURF). Detection of the banknotes is based on the haar features while identifying the banknote value relies on the speed-up robust features technique. Instead of analyzing each pixel, the haar features are used to identify the zone of interest in the image. The accuracies of banknote detection and banknote value recognition are 84% and 97.5%, respectively.

3.1.5 Other's Countries Banknotes Recognition System

Doush et al. [13] proposed a banknote recognition system based on the scale-invariant feature transform (SIFT) algorithm. In this paper, the scholars developed a dataset for Jordanian currency and applied an automatic mobile recognition system using a smartphone on the dataset using the scale-invariant feature transform (SIFT) algorithm. The experimental results show that the color SIFT approach outperforms the gray SIFT approach in terms of processing time and accuracy. Kogilawanee et al. [14] proposed recognizing Sri Lankan currency for visually impaired subjects using an Augmented Approach. The recognition process of this proposed system includes skew correction, feature extraction of color and texture, and counting the number of heavily printed dots on the currency notes. The wiener filter and canny edge detection techniques are used to remove noises and detect the edges in the captured images, respectively. In this proposed system, the test results show that the overall recognition rate of the system is 96%.

Alnowaini et al. [15] have also proposed a Yemeni paper currency detection system. In this paper, the authors built a dataset for each feature of the image, and the dataset was trained by a support vector machine. Gray-Level Co-Occurrence matrix (GLCM), speeded-up robust features (SURF) discrete wavelet transform (DWT), and principal component analysis (PCA) techniques have been used for extracting texture features from the image.

Mittal et al. [16] have developed Indian banknote recognition using a convolutional neural network. In this study, the scholars propose a deep learning-based technique for the identification of denominations of Indian currency rupee notes from their color images. A classification framework has been implemented using the concept of transfer learning where a large convolutional neural network pre-trained on millions of natural images is employed for the classification of images from new classes. According to the experimental results, the retrained lightweight model achieves an accuracy of 96.6 % on a held-out testing subset.

Sarfraz [5] proposed an intelligent paper currency recognition system, which is based on interesting features and correlation between images. In this paper, the method uses the case of Saudi Arabian paper currency as a model. In this study, first, the banknotes are collected, then the collected banknotes are scanned, and finally, image processing such as noise removal, resizing the image, changing the image to grayscale, and so on is performed. Then the feature extraction phase continues and finally, weighted Euclidian distance and radial basis neural network is applied for the classification of the image and, based on the experimental result, the average recognition rate is 91.51%.

3.2 Summary

Several studies have explored the realm of Ethiopian banknote recognition and counterfeit detection using various methodologies and datasets. Woldehana et al.[55]'s research, focusing on classifying counterfeit and genuine Ethiopian banknotes, utilized a dataset augmented to 2000 images of 200 ETB and 100 ETB denominations. Employing transfer learning with Inception and Vgg16 models, the study achieved high accuracies of 99.25%. However, it was limited by its use of only the front side of banknotes, 10 and 50 notes not addressed, and a relatively small dataset captured through imaging rather than scanning.

Gebremeskel et al.[54], in a study, concentrated solely on detecting fake Ethiopian banknotes using a deep CNN model. Their dataset comprised both fake and real images of 50, 100, and 200 ETB notes, achieving an accuracy of 97.6%. Nevertheless, the study did not address the detection of the back side of banknotes, 10 ETB notes, and relied on images captured by cameras rather than scanned images.

Assefa et al.[53], developed a recognition model for Ethiopian banknotes using MobileNetV2 with RMSProp optimization, attaining an accuracy of 96.80%. Their study focused on recognizing and classifying the front side of old Ethiopian banknotes, excluding new 200 ETB notes and neglecting analysis of the back side of the banknotes.

Tesfaw et al.[27] focused on "Ethiopian Banknote Recognition and Fake Detection using SVM," utilizing a small dataset and employing Support Vector Machines (SVM) alongside image processing techniques. Their approach achieved an accuracy of 93%, although it required human intervention for feature extraction and was limited to old Ethiopian banknotes. Asfaw Alene and Million Meshesha [23] developed the "Ethiopian Banknote Denomination Classification and Fake Detection System," achieving an accuracy of 96% with techniques like Color Momentum, SIFT, GLCM, CNN, and FFANN. However, their study's computational intensity and exclusive focus on the front side of old banknotes. Similarly, Jegnaw Fentahun and Yaregal Assabie [17] explored "Automatic Recognition and Counterfeit Detection of Ethiopian Paper Currency," attaining an accuracy of 96% using correlation coefficient and template matching techniques on a small dataset of front-side images. Their study focuses on old Ethiopian banknotes. These studies collectively contribute to the evolving landscape of Ethiopian banknote security, each offering insights into the challenges and advancements in the field of banknote recognition and counterfeit detection.

Our proposed solution integrates deep learning approaches to design a model for recognize and detect counterfeit Ethiopian banknotes. Utilizing a dataset of over 4764 scanned images and more than 510 captured images that include both the front and back sides of new Ethiopian banknotes(10,50,100,200), our customized CNN network achieved a notable accuracy of 99.82%, leveraging less computational resources compared to previous approaches. This underscores a notable advancement in the field, addressing previous limitations by encompassing comprehensive banknote analysis and detection capabilities

Chapter 4. Proposed System

4.1 Introduction

This chapter provides an in-depth exploration of the proposed solution aimed at designing a model for Ethiopian banknote recognition and counterfeit detection. Each element of the proposed solution will be examined, covering aspects such as the system, image preprocessing, data splitting, feature extraction, classification, identification, and concluding with a summary.

4.2 System Architecture

The proposed solution's architecture encapsulates the design choices about the overall structure and behavior. It constitutes the foundational arrangement of a system, encompassing its components, their interrelationships, and the guiding principles of its design. The objective of the proposed architecture is to tackle the identified problem and fulfill the functional and quality criteria outlined in the design.

The primary criterion considered in formulating the proposed architecture is its ability to produce the desired outcomes that align with the problem statements, constituting the overarching goal. The system architecture, as proposed, incorporates Image preprocessing using the techniques of image resizing and normalization. Data splitting for training, validation, and test data is also incorporated into our architecture after image preprocessing. Specifically, the training data is employed to train the model, while the validation data aids in parameter selection. On the other hand, the testing data serves the purpose of evaluating the model's performance. Feature extraction techniques are applied to all data types. Annex A provides a detailed implementation of the proposed system architecture. Figure 4.1 shows the overall structure of the proposed system architectures as follows.

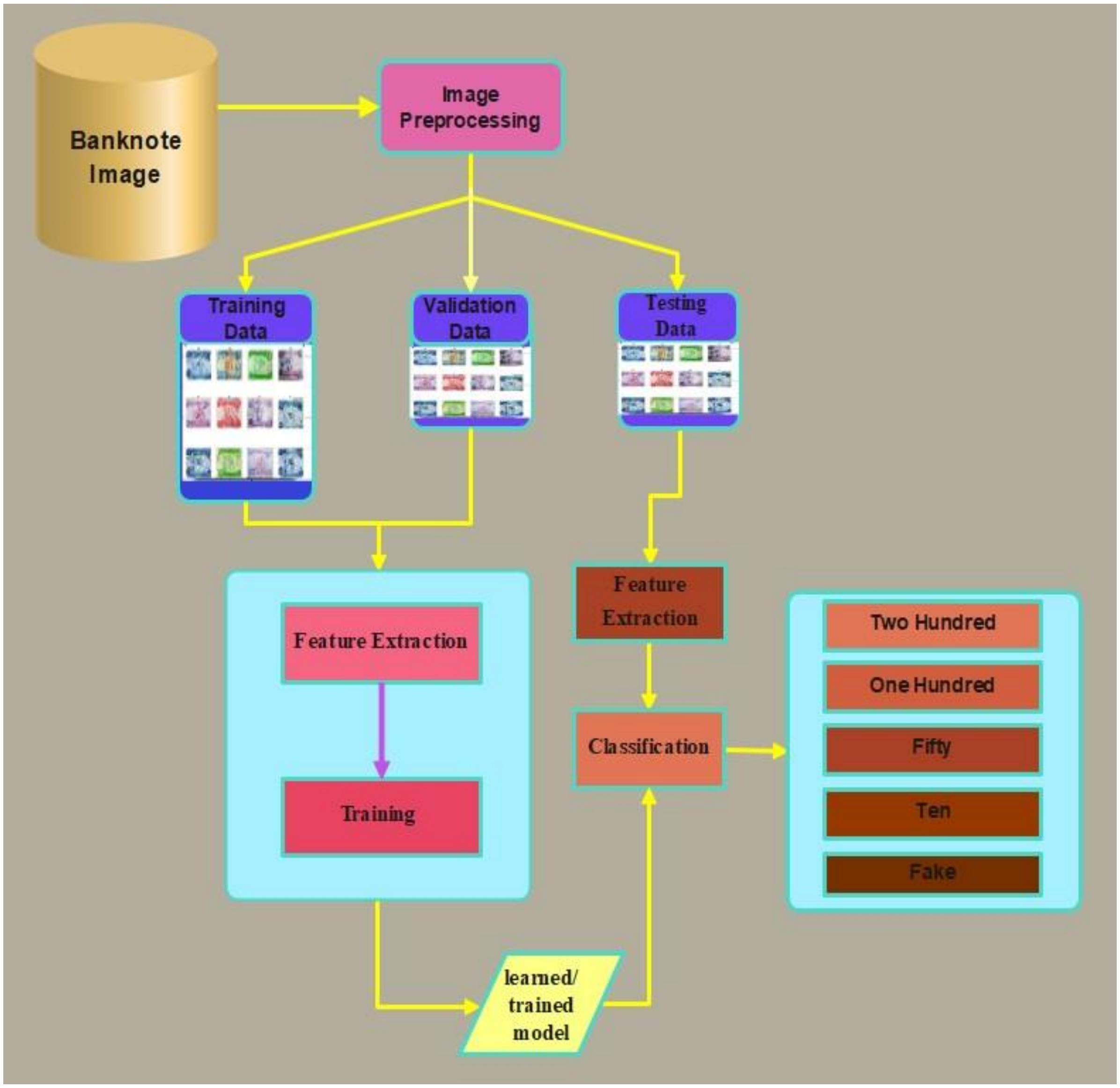

Figure 4. 1 System Architecture

Generally, to split data to train_validate_test data, extract feature of the image and create a model that is used to identify two hundred, one hundred, fifty, and ten notes image from the fake notes image, the following procedures are used:

i. Load the dataset.
ii. Determine the batch size and image input size.
iii. Determine the appropriate number of epochs.
iv. Split the data into train_validate_test data
v. Select the activation function.
vi. Extract the feature of images

```
   vii.     Train the model.
  viii.     Validate the accuracy.
    ix.     Save the optimal model.
```

Algorithm 4. 1 System Architecture

4.3 Data Preprocessing

Data preprocessing is an essential step in the field of image processing and convolutional neural networks (CNNs). In the context of CNNs, data preprocessing refers to the methods and techniques applied to raw data to prepare it for the network to effectively learn and make predictions. This often includes preprocessing steps such as normalization and resizing to improve the quality and consistency of the input data, which is crucial for the performance of CNNs [56].

In image processing, data preprocessing is crucial for improving the quality of images before they are fed into a CNN. This can involve tasks such as denoising, image enhancement, or feature extraction. These preprocessing steps help to reduce noise, enhance relevant image details, and extract important features that are essential for the CNN to make accurate predictions. By preprocessing the data in this way, the network can focus on learning the underlying patterns and relationships within the images, leading to improved performance and generalization [54].

Overall, data preprocessing plays a critical role in optimizing the performance of CNNs in image processing tasks. By carefully preparing and enhancing the input data, we can ensure that the network can learn effectively and make accurate predictions. As we continue to advance in the field of artificial intelligence, the importance of data preprocessing will only continue to grow, as it lays the foundation for successful machine learning models.

4.3.1 Image Resizing

Various researchers considered different sizes when developing the banknote recognition system concerning the dimensions of the banknote. Resizing images is a crucial step in deep-learning image processing for a variety of reasons. One important reason is to standardize the dimensions of images within a dataset, which is essential for feeding them into neural networks that require a fixed input size for training. Resizing images ensures that they all have the same width and height, which simplifies the computational operations, carried out by the neural network and improves efficiency. Additionally, resizing images reduces the complexity of the data, making it easier for the neural network to learn patterns and features present in the images. By resizing images, unnecessary information is discarded and only the most relevant details are retained, leading to better model performance and faster training times [56].

Moreover, resizing images can also help to prevent overfitting in deep learning models. When images are resized to a smaller size, the model is forced to focus on capturing the most important features and patterns present in the images, rather than memorizing the training data. This regularization technique can help improve the generalization ability of the model and make it more robust to variations in the input data. In addition, resizing images can also save computational resources by reducing the amount of memory and processing power required to train the neural network. Overall, resizing images is a critical preprocessing step in deep learning image processing that enhances model performance, efficiency, and generalization capabilities.

In their study on Ethiopia's banknote recognition system, [17] emphasized the significance of considering a banknote image size of 1122×570 for the design of the recognition system. It is noteworthy that the choice of size is contingent upon the specific dimensions of the banknote, given the variability in sizes among banknotes from different countries [17]. In this study, all the input images are resized to 256x256 for the alex_customed-design network. Figure 4.2 and Figure 4.3, show the original and 256x256 resized images respectively.

Figure 4. 2 Sample original two hundred birr image

Figure 4. 3 Sample resized two hundred birr image

4.3.2 Normalization

Normalization is a crucial technique in deep learning that involves adjusting the range of values of input data to improve the performance of the model during training. The main goal of normalization is to ensure that all input features are on a similar scale, which helps prevent certain features from dominating the learning process and allows the model to converge faster. There are several techniques used for normalization in deep learning, such as Min-Max scaling, Z-score normalization, and Batch normalization. Min-Max scaling involves scaling the data to a specific range (typically [0, 1] or [-1, 1] [57].

Additionally, normalization can help increase the model's generalization ability by reducing overfitting and ensuring that the model is robust to variations in the input data. Overall, normalization plays a critical role in deep learning by improving the efficiency of the learning process and enhancing the performance of the model [58].

4.4 Feature Learning

The primary goal of feature extraction is to carefully identify and extract the most significant features from extensive datasets. This process involves selecting features that effectively capture the essence of the entire dataset. The chosen features are deliberately designed to be straightforward to process and provide a concise representation of the original data. In the realm of image classification and detection, numerous researchers have introduced various deep convolutional neural network (CNN) architectures. Notable among these architectures are LeNet-5, AlexNet, VGGNet (VGG16, VGG19), DenseNet and MobileNet.

In our research, we customized AlexNet CNN architectures and employed Alex_customed Design Network for feature extraction and learning. This decision was informed by achieving the effectiveness of deep learning, making them well suited for our research objectives.

Alex_customized Designed Network configuration consists of a total of six convolutional layers, each followed by Rectified Linear Unit (ReLU) activation functions. These convolutional layers are crucial as they serve to extract complex hierarchical features from the input data, facilitating the network's ability to perceive meaningful patterns and structures within the images under analysis.

Furthermore, six max-pooling layers are distributed among these convolutional layers. These pooling layers play a crucial role in reducing the spatial dimensions of the feature maps generated by the preceding convolutional layers. By downsampling the feature maps by selecting the maximum value within each pooling region, the network effectively diminishes computational load while retaining significant spatial information essential for subsequent processing stages.

At the core of our customized AlexNet architecture lie two fully connected layers. These layers are designed to integrate the high-level features extracted by the preceding convolutional and pooling layers. The first and final fully connected layers were enhanced with ReLUs and SoftMax activation functions respectively. These layers were crucial in the subsequent stages of our research, where they facilitated feature classification, selection, and learning based on our predefined five classes.

By employing this customized CNN architecture, our research aimed to leverage the power of deep learning methodologies to analyze and classify complex image data. By optimizing the architecture to include specific layers and activation functions, we aimed to enhance the network's capacity to identify complex patterns within our dataset. Figure 4.4 shows the customized designed architecture for feature extraction and feature learning as follows.

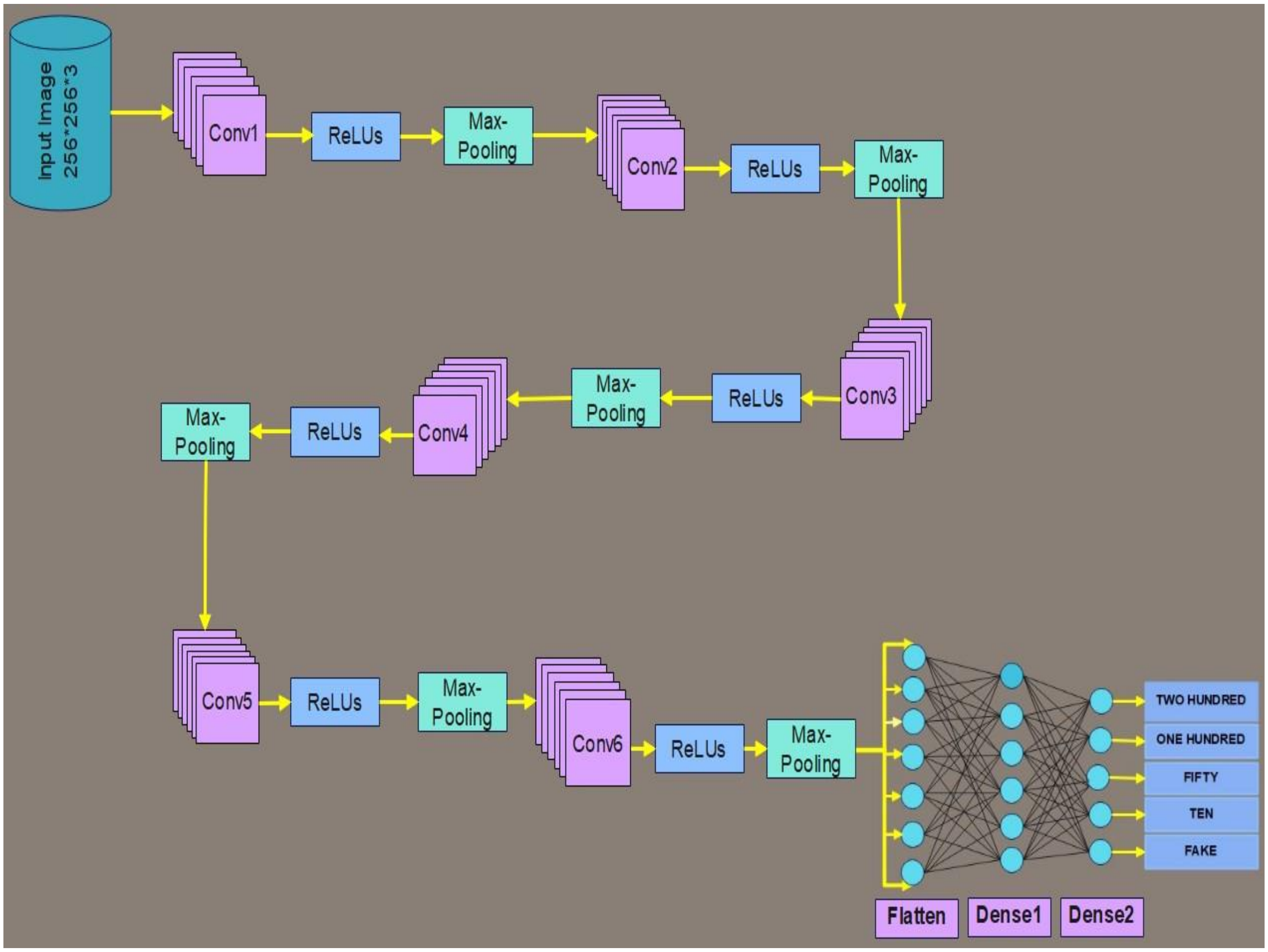

Figure 4. 4 Alex_customed designed network

Alex_customed designed a network algorithm

// Define input parameters

```
Set BATCH_SIZE
Set IMAGE_SIZE
Set CHANNELS
// Create a variable input_shape.
Assign the value (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS to input_shape.
//Set output classes
SET n_classes = 5
// Initialize the model
Initialize model as Sequential
```

```
        // Add layers to the model
      // Call the function to add layers to the model with
     the following Corresponding specifications:
   Add Convolution layer with 32 filters, kernel size (3,3),
activation 'relu', input shape input_shape to model
    Add MaxPooling layer with pool size (2,2) to model
    Add Convolution layer with 64 filters, kernel size (3,3),
activation 'relu' to model
    Add MaxPooling layer with pool size (2,2) to model
    Add Convolution layer with 64 filters, kernel size (3,3),
activation 'relu' to model
    Add MaxPooling layer with pool size (2,2) to model
    Add Convolution layer with 64 filters, kernel size (3,3),
activation 'relu' to model
    Add MaxPooling layer with pool size (2,2) to model
    Add Convolution layer with 64 filters, kernel size (3,3),
activation 'relu'  to model
    Add MaxPooling layer with pool size (2,2) to model
    Add Convolution layer with 64 filters, kernel size (3,3),
activation 'relu' to model
    Add MaxPooling layer with pool size (2,2) to model
    Add Flatten layer to model
    Add Dense layer with 64 units, activation 'relu' to model
    Add Dense layer with n_classes units, activation 'softmax'
to model
        // Build the model
    CALL model. Build(input_shape)
```

Algorithm 4. 2 Alex_customed designed network

4.5 Classification

Upon completing the data processing and feature extraction phases for both training and validation datasets, these datasets serve as input for the subsequent classification process. The models we designed were trained using the alex_customed-designed network. The model design process involves crucial considerations such as determining the batch size and image input size, as elaborated in Chapter Five, specifically in Section 5.4. The training phase necessitates thoughtful choices in training strategies, encompassing decisions on the loss function, activation functions, and optimization techniques. Chapter five, Section 5.4 provides a brief overview of the selected training strategy for our model. The validation dataset plays a pivotal role in optimizing parameters for achieving the highest accuracy. Following these steps, the constructed Convolutional Neural Network (CNN) model is saved for future use.

The alex_customed-designed network model has been specifically crafted to identify Ethiopian banknotes. Once the identification model is successfully generated, it becomes the cornerstone for all subsequent identification processes. Consequently, this carefully designed model plays a crucial role in the deployment of the prototype developed in this study. Furthermore, its versatility extends to potential applications in the future, including deployment in mobile applications, embedded systems, and other platforms.

4.6 Identification

Within the system architecture, identification constitutes a crucial component that assesses the provided image data, categorizing it into one of five classes: two hundred, one hundred, fifty, ten, or fake notes. This identification phase incorporates the preprocessed testing data, from which features have been extracted, and utilizes carefully designed models to predict the accurate classes. The anticipated classes encompass a two hundred-note class, one hundred class, fifty notes class, ten notes class, and fake notes class. This component signifies the concluding stage within the system architectures of the proposed study.

4.7 Summary

In this chapter, the system architecture and its components are described. The two data preprocessing techniques, image resizing and normalization, are explained. Our alex_customized_designed network and its implementation process are discussed in this chapter. The steps used to train the designed models are also briefly covered. Finally, the component for recognition or detection, which integrates the test data with the designed model to identify counterfeit and genuine Ethiopian banknote classes, is also discussed.

Chapter 5. Experiment and Evaluation

5.1 Introduction

In this chapter, we elaborate on the experimental procedures employed in the recognition and counterfeit detection of Ethiopian Banknotes. Thorough details regarding the dataset, including sources and volume, as well as the tools, programming languages, and the experimental setup utilized for designing and developing the prototype to implement the model, are presented. Furthermore, we delve into the model-building and compilation process for the alex_customed-designed network, explaining the training methodology and its components. The chapter also covers the development of the prototype; the evaluation method employed, and the results of the experiment, and concludes with a detailed discussion and summary of the findings.

5.2 Dataset Preparation

In the realm of deep learning, supervised algorithms demand sample data for effective training, making meticulous dataset preparation indispensable for optimal learning quality and prediction accuracy. Specifically, in the domain of image processing for deep learning, the process of dataset preparation assumes a pivotal role in shaping the success and precision of the model.

For this particular project, the dataset was carefully collected from various financial institutions, including the National Bank of Ethiopia, Commercial Bank of Ethiopia, Abyssinia Bank, and NIB International Bank. Additionally, a diverse collection of Ethiopian banknotes, ranging from two hundred to ten denominations, was gathered from these sources. Notably, the counterfeit notes are sourced exclusively from the National Bank of Ethiopia, introducing a crucial dimension for the model to discern genuine and counterfeit currency. The collected banknotes exhibited a wide spectrum of characteristics, including color variations, and image quality (spanning new, old, and very old banknotes), contributing to a rich and diverse dataset.

Both scanner and camera images of these currencies are acquired and stored for evaluating and testing the system. The scanner used for capturing images is the Canon CanoScan LiDE 300 Slim Color Image Scanner. Each genuine Ethiopian currency is scanned with a resolution from 300-350 dpi and saved in a JPEG image. However, the fake notes were captured using a camera with a Samsung Galaxy S20. Figure 5.1: shows samples of the dataset used for the denomination type of each banknote and fake notes for the front side and backside of Ethiopian Banknotes.

Figure 5. 1 Sample data Ethiopian banknotes and fake notes

To enhance the dataset's variability and ensure the model's robustness. Consequently, the model becomes more resilient and adaptable to different real-world scenarios, ultimately bolstering its ability to accurately process and classify images across diverse settings.

The final dataset comprised 5274 original images and an augmented set of 5274 * 3 images. In order to increases the size and diversity of the dataset by applying data augmentation.
Data augmentation is a crucial technique in the preparation of datasets for Convolutional Neural Networks (CNNs), designed to artificially increase the size and variability of training data by applying various transformations to images. These transformations may include rotations, translations, scaling, cropping, flipping, and adjustments to brightness or contrast. The primary goal of data augmentation is to expose the CNN to a broader range of image variations, enhancing its ability to generalize to new, unseen data and reducing the risk of overfitting. In this study, a sequential model is established for data augmentation, starting with a Random Flip layer that performs random horizontal and vertical flips on the input images, followed by a Random Rotation layer that allows for rotations of up to 0.2 radians. This augmentation pipeline dynamically modifies batches of images during training, thereby enriching the dataset and improving the model's robustness [57].

Subsequently, following dataset preparation and augmentation, a standard practice involved dividing the dataset into three subsets: training, validation, and testing. In this case, 80% of the dataset was allocated for training purposes, while 10% each was reserved for validation and testing, ensuring a comprehensive evaluation of the model's performance across distinct datasets. This strategic division facilitates effective training and evaluation, contributing to the overall success of the deep learning model. Table 5.1 shows the details of the data used for the denomination type of each banknote and fake note as follows.

Table 5. 1 Tabular summary of the dataset

Banknotes Type	Quantity
Two hundred	1595
One hundred	1642
Fifty	910
Ten	617
Fake	510
Total	5274

5.3 Developmental Tools and Experimental Environment

5.3.1 Tools and Programming Languages

Development tools encompass various types of hardware, software, and programming tools employed to implement and design the proposed model. Within this study, the following tools have been utilized.

Hardware Tools

Lenovo laptop computer that is Intel(R) Core (TM) i5-3320M having CPU @ 2.60GHz and 16.00 GB RAM is used for processing overall experimental tasks and writing the documentation. In addition to this T4 GPU is used to train the designed model, Canon CanoScan LiDE 300 Slim Color Image Scanner, and Samsung Galaxy S20 for scanning and capturing Ethiopian genuine and fake notes.

Software Tools

In this research, we utilized Microsoft Word 2016 for document editing and Wondershare Edrawmax 2024 for creating system architectures.

Programming languages

This research employs a variety of programming tools for code development and visualization of the ultimate results. The specific tools used in this study are outlined as follows.

Jupyter Notebook

In this study, we used Jupiter Notebook, which is an open-source web application, used for creating and sharing documents with live code, equations, visualizations, and narrative text.

Keras

Keras is a high-level neural networks API written in Python, specifically designed for deep learning applications. Its user-friendly interface allows developers to easily build and train neural networks without having to worry about the low-level implementation details. In deep learning image processing, Keras is commonly used for tasks such as image classification, object detection, and image segmentation. Keras also supports data augmentation techniques, which can help improve the generalization and robustness of the deep learning models by artificially increasing the size of the training dataset. Overall, Keras simplifies the process of developing deep learning models for image processing tasks, making it a popular choice among researchers and developers in the field[31].

TensorFlow

TensorFlow is a powerful open-source machine-learning library developed by Google that is widely used in deep-learning image processing. It works by creating computational graphs that represent the flow of data through a network of nodes, each of which performs mathematical operations on the data. These graphs are then executed using a backend engine that can run on CPUs, GPUs, or even specialized hardware like TPUs. In deep learning image processing, Tensorflow is used to build and train convolutional neural networks, which are a type of deep learning model specifically designed to process visual data. By feeding images into these networks and adjusting the weights of the nodes through a process called backpropagation, Tensorflow can learn to extract meaningful features from the images and make predictions based on them. This allows for tasks like image classification, object detection, and image generation to be performed with high accuracy and efficiency. It is simple for beginners and even experts. It is also used to build multi-layer, large-scale neural networks [31].

5.3.2 Environmental setup

5.3.2.1 Collaboratory

Collaboratory, also known as Colab, is a web-based platform developed by Google that allows users to write and execute Python code in a Jupyter Notebook environment. Jupyter notebooks have become increasingly popular among data scientists and researchers due to their interactive and user-friendly nature. Hosting Jupyter in Colab provides several advantages, such as access to GPU resources for running machine-learning models at a faster speed, seamless integration with Google Drive for data storage and sharing, and the ability to collaborate with others in real time. Additionally, Colab offers a cloud-based infrastructure that eliminates the need for local installation of software packages, making it a convenient and efficient tool for conducting data analysis and research projects.

5.3.2.2 Google *Drive*

Training the dataset on Colab began by storing it in Google Drive and subsequently integrating it into Colab through the mounting method. As a result, for this study, the dataset remains hosted on Google Drive, ensuring accessibility and seamless integration with Colab's computational resources. This setup facilitates efficient data handling and model training directly within the Colab environment, leveraging its convenience and capability for machine learning tasks.

5.4 Alex_customed-designed network model Training

Building and compiling an alex_customed-designed network model involves a series of steps that require careful consideration and expertise. The proposed model or architecture of the alex_customed-designed network consists of six convolutional layers, six max pooling layers followed by two fully connected layers, with ReLU activation functions and Softmax activation function to extract features from the input images.

In this study, Training alex_customed-designed network involves optimizing the model's weights and biases to minimize the classification error using stochastic gradient descent or other optimization algorithms, Adam as an optimizer, and sparse categorical cross-entropy as a loss function.

During training the dataset using the alex_customed-designed network for designing a model for Ethiopian banknote recognition and counterfeit detection, the batch size was set to 32, the image size set to 256x256, the number of epochs set to 30, and T4 GPU was used.

The alex_customed-designed network was built from 896 input layers and 183877 total parameters. From this 183877 are trainable parameters and zero parameters for non-trainable flatten parameters. 0 parameters for flattened and 325 for dense. Figure 5.2 shows the screenshots of the proposed models-built details of the alex_customed-designed network.

```
model.summary()
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
sequential_1 (Sequential)	(32, 256, 256, 3)	0
conv2d (Conv2D)	(32, 254, 254, 32)	896
max_pooling2d (MaxPooling2D)	(32, 127, 127, 32)	0
conv2d_1 (Conv2D)	(32, 125, 125, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(32, 62, 62, 64)	0
conv2d_2 (Conv2D)	(32, 60, 60, 64)	36,928
max_pooling2d_2 (MaxPooling2D)	(32, 30, 30, 64)	0
conv2d_3 (Conv2D)	(32, 28, 28, 64)	36,928
max_pooling2d_3 (MaxPooling2D)	(32, 14, 14, 64)	0
conv2d_4 (Conv2D)	(32, 12, 12, 64)	36,928
max_pooling2d_4 (MaxPooling2D)	(32, 6, 6, 64)	0
conv2d_5 (Conv2D)	(32, 4, 4, 64)	36,928
max_pooling2d_5 (MaxPooling2D)	(32, 2, 2, 64)	0
flatten (Flatten)	(32, 256)	0
dense (Dense)	(32, 64)	16,448
dense_1 (Dense)	(32, 5)	325

Total params: 183,877 (718.27 KB)
Trainable params: 183,877 (718.27 KB)
Non-trainable params: 0 (0.00 B)

Figure 5. 2 Alex_customed-designed network model building

Loss Function

A loss function is used to measure the error between the predicted output and the actual output. It essentially quantifies how well the model is performing and provides feedback for the model to adjust its parameters during the training process. The loss function is a mathematical function that takes in the predicted output and the ground truth label as input and outputs a single scalar value that represents the error or loss.

From different types of loss functions, sparse categorical cross-entropy is recommended for the tasks of multi-class classification. Where the input can only belong to one out of many possible classes. Sparse Categorical Cross entropy is a loss function commonly used in categorical classification tasks where the target variable is integer-encoded. It is particularly useful when dealing with problems involving multiple classes[31].

The following equation describes how the sparse categorical cross-entropy loss function can be calculated:

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N} \log(\hat{y}_{yi}) \qquad (1)$$

- **N** is the total number of samples in the dataset.
- y_i is the integer class label for the i-th sample, representing the correct class.
- $\hat{y}_{yi}$ is the predicted probability of the true class for the i-th sample.

We take the negative logarithm of the predicted probability of the true class for each sample.

Finally, we take the average over all samples to get the overall loss.

Activation Function

Activation function is essential in neural networks, particularly in deep learning that determines the output of a node, or neuron, given the input it receives. It plays a key role in introducing non-linearity into the network, allowing it to learn complex patterns and relationships in the data. There are several types of activation functions commonly used in deep learning, including sigmoid, tanh, ReLU, Leaky ReLU, and softmax[19].

For this study, the ReLU (Rectified Linear Unit) and Softmax activation functions were employed:

ReLU (Rectified Linear Unit):

The Rectified Linear Unit (ReLU) is a popular activation function used in deep learning. It works by applying a simple threshold function to the output of each neuron. Specifically, ReLU sets any negative values to zero and leaves positive values unchanged. This activation function helps address the vanishing gradient problem by introducing non-linearity in the neural network, which allows more complex relationships to be learned during the training process. Additionally, ReLU is computationally efficient and easy to implement, making it a preferred choice for many deep-learning practitioners[19].

In CNNs, ReLU is typically used after the convolutional and pooling layers to introduce non-linearity and allow the network to learn complex patterns in image data. By applying ReLU after each convolutional layer, CNNs can capture more complex features in the input data, ultimately improving the network's ability to classify objects accurately [19].
The Rectified Linear Unit (ReLU) activation function was defined mathematically as follows:

$$\text{ReLU}(x) = \text{ReLU}(x) = \max(0, x) \tag{2}$$

In this formula:

- x represents the input to the ReLU function.
- The function outputs 0 if the input x is less than or equal to 0.
- If the input x is greater than 0, the function outputs the input value x itself.

Softmax Activation Function

It is an activation function used in neural computing. It is used to compute probability distribution from a vector of real numbers. It produces an output having a range between [0, 1], and the sum of its probabilities has been equal to 1.

Given a vector of raw output scores x=[x1, x2,…,xC] from a model, where C is the number of classes, the softmax function is defined as follows:

$$\text{Softmax}(xi) = \frac{e^{xi}}{\sum_{j=1}^{C} e^{xj}} \quad \text{for i=1, 2,…,C} \tag{3}$$

It is used in multi-class models where it returns the probabilities of each class, with the target class, which has the highest probability. The Softmax activation function mostly appears in almost all the output layers of the deep learning architectures[19].

Adam Optimization Algorithm

It is a stochastic function that optimizes using first-order gradients. It is the best method to implement for any model in terms of large datasets and with several parameters. In terms of hardware resource usage, it requires less memory space and it is computationally very efficient. It is also well suited for non-stationary targets with noisy and sparse gradients. Adam is a strategy that separates adaptive learning rates from each parameter. It also uses a single learning rate for every weight update and does not change during the training time. Adam is a hybrid of RMSprop and Stochastic Gradient Descent since it estimates the first and second moments of the gradient to balance the learning rate for each model network weight. Therefore, there are many benefits to using the Adam optimizer in terms of speed of training and other listed benefits.

5.5 Discussion and System Evaluation Result

This section presents the experimental findings of the study, focusing on the performance evaluation of the proposed models. The model has repeatedly experimented with the prepared dataset to confirm the correct result. The experiment process has evaluated the required designed model. Comprehensive graphs depicting training accuracy, validation accuracy, training loss, and validation loss for each proposed model were discussed herein. The study utilizes alex_customed-design network architectures to discern Ethiopian banknote images, categorizing them into denominations of two hundred, one hundred, fifty, and ten, as well as distinguishing fake notes. Each proposed model's outcomes for identifying Ethiopian banknotes were elaborated sequentially. Additionally, Figure 5.3 and Figure 5.4 illustrate the predicted sample image results for reference.

Figure 5. 3 Sample predicted image result

Figure 5. 4 Sample predicted image results with confidence in percentage

Predict result algorithm

```
//Set Figure Size:
Create a new figure with a size of 10 by 10.
For each batch of images and labels in
test_ds, (take only the first batch):
//Loop through Images:
  For i from 0 to 14 (total of 15 iterations):
Create a subplot at position i + 1 in a 3x3
grid.
Display the image at index i as an unsigned
integer (uint8).
//  Make Prediction:
Call the predict function with the model and
the image at index i to get:
     predicted_class
     confidence
//Get Actual Class:
Set actual_class to the class name
corresponding to labels[i].
//Set the title of the subplot to:
     "Actual: [actual_class], \n predicted:
     [predicted_class], \n confidence:
     [confidence]%"
```

Algorithm 4. 3 Result prediction

5.5.1 System Evaluation Result for alex_customed-design network

After training the alex_customed-design network model with the provided dataset, the performance metrics were thoroughly evaluated. The model achieved an impressive 99.92% training accuracy and equally noteworthy is the model's validation accuracy of 99.87%. This slight reduction from the training accuracy reflects an almost negligible difference, suggesting that the model generalizes well to new, unseen data. Such a minimal gap between training and validation accuracy implies that the model is robust and not overfitting to the training set.

The training loss is notably low at 0.078%, signifying that the model's predictions are very close to the actual labels in the training set. The validation loss, though slightly higher at 0.127%, remains quite low as well. The small increase in validation loss compared to the training loss is expected and typically reflects the natural variance between training and validation data.

When tested on a separate test set, the model achieved an accuracy of 99.82% and a loss of 0.173%. This result confirms that the model's performance remains robust and consistent even on entirely new data not seen during the training or validation phases. The slight decrease in accuracy and increase in loss compared to the training and validation metrics are expected and highlight the model's effective generalization to real-world scenarios.

Overall, these results reflect a model with outstanding accuracy and minimal loss across all evaluated datasets, demonstrating strong generalization and reliability in diverse conditions. The comprehensive analysis of the training and validation phases, including accuracy and loss metrics, is visually detailed in Figure 5.5. This figure illustrates the overall progression and performance of the alex_customed-design network model, highlighting its performance and stability throughout the training process.

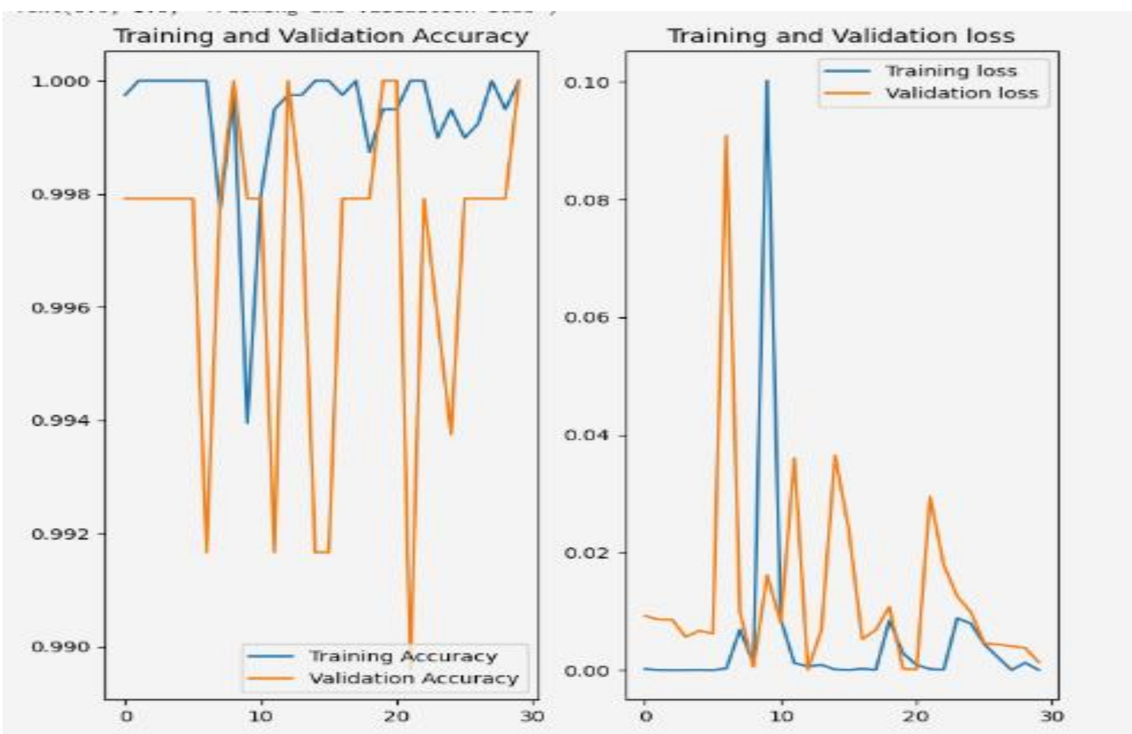

Figure 5. 5 Training and Validation accuracy

Train and validation plotting accuracy algorithm

```
//Create Figure for Accuracy Plot
       //Plot Training and Validation Accuracy:**
   Create a subplot in a 1x2 grid at position 1.
    Plot the training accuracy against the range of epochs:
     Use `acc` for training accuracy.
     Label the plot as "Training Accuracy".
   Plot the validation accuracy against the range of epochs:
     Use `val_acc` for validation accuracy.
     Label the plot as "Validation Accuracy".
   Add a legend in the lower right corner.
   Set the title of the subplot to "Training and Validation
Accuracy".
//Plot Training and Validation Loss:
   Create a subplot in a 1x2 grid at position 2.
   Plot the training loss against the range of epochs:
     Use `acc` for training loss.
     Label the plot as "Training Loss".
   Plot the validation loss against the range of epochs:
     Use `val_loss` for validation loss.
      Label the plot as "Validation Loss".
    Add a legend in the upper right corner.
   Set the title of the subplot to "Training and Validation
Loss".
```

Algorithm 4. 4 Train and validation plotting accuracy loss

5.5.2 Comparative analysis of our proposed solution with existing related works

As discussed in Section 3.1.1 different researchers proposed different solution for solving recognition and detection of the counterfeit of Ethiopian banknotes. Each of the studies has its final results and our study has its results. Table 5.2 shows the comparison of our proposed solution with others as follows:

Table 5. 2 Comparison of our proposed solution with others

Authors	Models/Methods	Accuracy
Jegnaw & Yaregal[17]	Dominant & distribution of dominant color, hue value, SURF	96.13%.
Tesfaw et al.[27]	SVM	93.00%.
Assfaw & Million[23]	Color momentum, SIFT, GLCM, CNN, FFANN	96.46%
Gebremeskel et al.[54]	Deep learning approach	97.60%
Aseffa et al.[53]	MobileNetV2 model using RMSProp	96.80%
WoldeHana et al. [55]	Inception and Vgg16 models.	99.25%
Our model	Alex_customed design network	99.82%

From Table 5.2 we can understand that the proposed system records the highest accuracy than others. Therefore, by using the customized deep learning approach we can achieve a good result for the recognition and detection of the counterfeit of Ethiopian Banknotes.

5.6 Prototype Development

In this study, we have developed a prototype to achieve our defined objectives. This prototype demonstrates our research concept and methodology, displaying the practical implementation and feasibility of our proposed solution. The following GUI is displayed using collab and ngrok account. Figure 5.6 displays screenshots of the prototype's graphical user interface (GUI) to upload images.

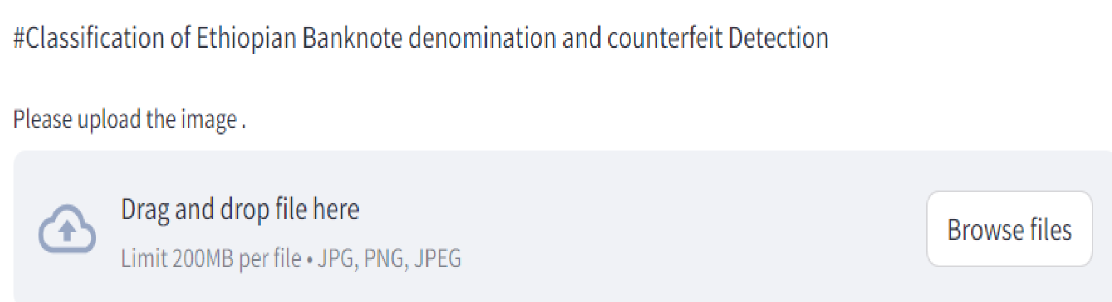

Figure 5. 6 Data importing interface

Upon importing the image, the prototype simultaneously displays the image along with either its class name, denomination value, or counterfeit detection result. Figure 5.7 shows the banknote image and the predicted class names.

Figure 5. 7 The prototype of the proposed model prediction result interface for genuine

Figure 5. 8 The prototype of the proposed model prediction result interface for fake

5.7 Summary

This chapter provides a comprehensive overview of the experimental work conducted, detailing various aspects of the research process. Firstly, the chapter explores the specifics of the dataset used, including its size and sources. Additionally, it discusses the distribution of the dataset into three splits: training, validation, and testing, highlighting the importance of each subset in model development and evaluation.

Furthermore, the chapter briefly outlines the tools utilized in designing the proposed models, offering insights into the technological infrastructure employed in the research endeavor. A detailed explanation of the model building and compilation process for the Alex_customed-design network is provided, illuminating on the details of model architecture and configuration.

Moreover, the chapter elaborates on the training strategy employed to train the designed model, exploring various methodologies and optimization techniques utilized to enhance model performance. Additionally, an overview of the prototype developed for experimental purposes is presented, presenting the practical implementation of the proposed model.

Finally, the chapter explores the evaluation results of the system, providing a comprehensive analysis of key performance metrics such as training accuracy, training loss, validation accuracy, and validation loss. Graphical representations of these metrics offer visual insights into the model's training and validation processes, facilitating a deeper understanding of its performance dynamics.

Chapter 6. Conclusion and Recommendation

6.1 Conclusion

In our study, we aimed to develop a customized deep-learning model named as alex_customized_design network, capable of accurately recognizing and detecting counterfeit Ethiopian banknotes. To achieve this, we gathered a large dataset consisting of images sourced from prominent financial institutions such as the National Bank of Ethiopia, Commercial Bank of Ethiopia, NIB International Bank, and Bank of Abyssinia.

To prepare the collected data for model training, we implemented several preprocessing techniques including normalization and resizing. These steps were essential for ensuring uniformity and enhancing the diversity of our dataset, thereby improving the model's ability to generalize.

For feature extraction and representation learning, we employed Convolutional Neural Network (CNN) architectures, specifically utilizing the Alex_customed-design network. This choice was informed by the alex_customed-design network's proven efficacy in image classification tasks, making it a suitable candidate for our objectives.

The training process was conducted on a T4 GPU within the Google Colab virtual environment, leveraging its computational power to expedite model training.

To optimize the performance of our model, we adopted sparse categorical cross-entropy as the loss function, Adam optimizer for gradient descent optimization, and Rectified Linear Unit (ReLU) activation function alongside softmax activation. These choices were made based on their demonstrated effectiveness in similar image recognition tasks.

Upon evaluating the trained model, we achieved an acceptable accuracy of **99.82%** in recognizing and detecting counterfeit Ethiopian banknotes. To further validate the model's performance, we developed a prototype for real-world testing using separate test data. The prototype successfully identified banknote denominations including Two Hundred, One Hundred, Fifty, and Ten, while also detecting instances of counterfeit banknotes.

In summary, our proposed system demonstrates high accuracy and efficiency in identifying Ethiopian banknote denominations and detecting counterfeit instances. This underscores its potential utility in safeguarding financial transactions and maintaining the integrity of currency circulation.

6.2 Contribution

In this research, we have contributed to the domain of deep learning, particularly in the realm of recognizing and detecting counterfeit Ethiopian banknotes. Our contributions can be summarized as follows:

- We carefully collected a dataset comprising 5,274 genuine Ethiopian banknotes across four denominations and fake notes, serving as a valuable reference for researchers in our domain.
- We designed and developed an Alex-customized_design network model for the recognition and detection of counterfeit Ethiopian banknotes with acceptable, best accuracy and efficiency.
- We have developed a prototype to evaluate our developed model with real-world data.

6.3 Future Work

The designed model can recognize and detect counterfeit Ethiopian banknotes called Two Hundred, One Hundred, Fifty, and Ten. To implement the model in real application areas the following things are extended as future work.

- Developing mobile applications and integrating the designed models.
- Implementing the designed models on different currency counting machines" or a "money counter to recognize and detect counterfeits.

References

[1] V. Abburu, S. Gupta, S. Rimitha, M. Mulimani, and S. Koolagudi, “Currency recognition system using image processing,” Aug. 2017, pp. 1–6.

[2] R. F. Olanrewaju, F. F. Eniola, and S. M. Mahmoud M., “Automated Bank Note Identification System for Visually Impaired Subjects in Malaysia,” *2016 Int. Conf. Comput. Commun. Eng. ICCCE*, pp. 115–120, Jul. 2016.

[3] S. Veling, “Fake Indian Currency Recognition System by using MATLAB,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 7, pp. 3605–3612, Apr. 2019.

[4] “(99+) An Automated Recognition of Fake or Destroyed Indian Currency Notes Using Image Processing | Ijesrt Journal - Academia.edu.” Accessed: Jul. 01, 2024. [Online]. Available: https://www.academia.edu/9171283/An_Automated_Recognition_of_Fake_or_Destroyed_Indian_Currency_Notes_Using_Image_Processing

[5] M. Sarfraz, “An Intelligent Paper Currency Recognition System,” *Procedia Comput. Sci.*, vol. 65, pp. 538–545, Jan. 2015.

[6] E.-H. Zhang, B. Jiang, J.-H. Duan, and Z.-Z. Bian, “Research on paper currency recognition by neural networks,” in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*, Nov. 2003, pp. 2193-2197 Vol.4.

[7] F. M. Hasanuzzaman, X. Yang, and Y. Tian, “Robust and effective component-based banknote recognition by SURF features,” in *2011 20th Annual Wireless and Optical Communications Conference (WOCC)*, Apr. 2011, pp. 1–6.

[8] F. García-Lamont, J. Cervantes, and A. Lopez-Chau, “Recognition of Mexican banknotes via their color and texture features,” *Expert Syst. Appl.*, vol. 39, pp. 9651–9660, Aug. 2012.

[9] S. Omatu, M. Yoshioka, and Y. Kosaka, “Reliable Banknote Classification Using Neural Networks,” *Adv. Eng. Comput. Appl. Sci. Int. Conf. On*, vol. 0, pp. 35–40, Oct. 2009.

[10] L. Dunai, M. Chillarón Pérez, G. Peris-Fajarnés, and I. Lengua, “Euro Banknote Recognition System for Blind People,” *Sensors*, vol. 17, p. 184, Jan. 2017.

[11] A. Rashid, A. Prati, and R. Cucchiara, “On the design of embedded solutions to banknote recognition,” *Opt. Eng.*, vol. 52, p. 093106, Aug. 2013.

[12] “Sensors | Free Full-Text | Efficient Banknote Recognition Based on Selection of Discriminative Regions with One-Dimensional Visible-Light Line Sensor.” Accessed: Jul. 01, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/16/3/328

[13] I. Abu Doush and S. AL-Btoush, “Currency recognition using a smartphone: Comparison between color SIFT and grayscale SIFT algorithms,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 29, no. 4, pp. 484–492, Oct. 2017.

[14] B. Kogilawanee and A. Ramanan, “An Augmented Approach for Recognising Sri Lankan Currency Notes,” in *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, Dec. 2018, pp. 1–5.

[15] G. Alnowaini, A. Alabsi, and H. Ali, “Yemeni Paper Currency Detection System,” in *2019 First International Conference of Intelligent Computing and Engineering (ICOICE)*, Dec. 2019, pp. 1–7.

[16] S. Mittal and S. Mittal, “Indian Banknote Recognition using Convolutional Neural Network,” in *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, Feb. 2018, pp. 1–6.

[17] Information Network Security Agency, Addis Ababa, Ethiopia, J. Fentahun Zeggeye, and Y. Assabie, “Automatic Recognition and Counterfeit Detection of Ethiopian Paper Currency,” *Int. J. Image Graph. Signal Process.*, vol. 8, no. 2, pp. 28–36, Feb. 2016.

[18] A. A. S. Amol A. Shirsath, “A Review of Paper Currency Recognition System,” *IOSR J. Comput. Eng.*, vol. 10, no. 1, pp. 71–76, 2013.

[19] Y. Liu, J. He, and M. Li, “New and Old Banknote Recognition Based on Convolutional Neural Network,” in *Proceedings of the 2018 1st International Conference on Mathematics and Statistics*, in ICoMS ’18. New York, NY, USA: Association for Computing Machinery, Jul. 2018, pp. 92–97.

[20] F. M. Hasanuzzaman, X. Yang, and Y. Tian, “Robust and Effective Component-Based Banknote Recognition for the Blind,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, no. 6, pp. 1021–1030, Nov. 2012.

[21] S.-C. Ng, C.-P. Kwok, S.-H. Chung, Y.-Y. Leung, and H.-S. Pang, “An Intelligent Banknote Recognition System by using Machine Learning with Assistive Technology for Visually Impaired People,” in *2020 10th International Conference on Information Science and Technology (ICIST)*, Sep. 2020, pp. 185–193.

[22] Z. Dinku and K. Raimond, “COUNTERFEIT CURRENCY IDENTIFICATION SYSTEM - A CASE STUDY ON ETHIOPIAN BIRR NOTE,” vol. 26, 2009.

[23] A. S. Alene, “Ethiopian Paper Currency Recognition System: An Optimal Feature Extraction,” vol. 7, no. 8, 2019.

[24] L. M. Andrade *et al.*, “Pocket test for instantaneous quantification of starch adulterant in milk using a counterfeit banknote detection pen,” *Food Chem.*, vol. 405, p. 134844, Mar. 2023,.

[25] A. A. Chowdhury, A. Das, and D. Karmaker, “Evaluation of Deep Learning models on UV ink : a Fake Money detection scheme with RPN.” Aug. 17, 2021.

[26] D. Harjunowibowo, S. Hartati, and A. Budianto, “A Counterfeit Paper Currency Recognition System Using LVQ based on UV Light,” *IJID Int. J. Inform. Dev.*, vol. 1, no. 2, p. 9, Dec. 2012.

[27] E. Ayalew Tessfaw, B. Ramani, and T. Kebede Bahiru, “Ethiopian Banknote Recognition and Fake Detection Using Support Vector Machine,” in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore: IEEE, Apr. 2018, pp. 1354–1359.

[28] T. H. Chia and M. J. Levene, “Detection of counterfeit U.S. paper money using intrinsic fluorescence lifetime,” *Opt. Express*, vol. 17, no. 24, pp. 22054–22061, Nov. 2009.

[29] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, “ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models.” arXiv, Dec. 14, 2018..

[30] T. Tseng *et al.*, “Co-ML: Collaborative Machine Learning Model Building for Developing Dataset Design Practices,” *ACM Trans Comput Educ*, vol. 24, no. 2, p. 25:1-25:37, Apr. 2024.

[31] I. S. Candanedo, E. H. Nieves, S. R. González, M. T. S. Martín, and A. G. Briones, “Machine Learning Predictive Model for Industry 4.0,” in *Knowledge Management in Organizations*, L. Uden, B. Hadzima, and I.-H. Ting, Eds., Cham: Springer International Publishing, 2018, pp. 501–510.

[32] S. Mutsuddi, “Machine Learning for Predictive Maintenance in Manufacturing Industries,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 4, pp. 1175–1181, Apr. 2023.

[33] P. Stoffel, M. Berktold, and D. Müller, “Real-life data-driven model predictive control for building energy systems comparing different machine learning models,” *Energy Build.*, vol. 305, p. 113895, Feb. 2024.

[34] P. Tan, “Ensemble-based Hybrid Optimization of Bayesian Neural Networks and Traditional Machine Learning Algorithms.” arXiv, Oct. 09, 2023.

[35] B. Akbugday, “Classification of Breast Cancer Data Using Machine Learning Algorithms,” in *2019 Medical Technologies Congress (TIPTEKNO)*, Oct. 2019, pp. 1–4. d.

[36] Y. Oktafriani, G. Firmansyah, B. Tjahjono, and A. M. Widodo, “Analysis of Data Mining Applications for Determining Credit Eligibility Using Classification Algorithms C4.5, Naïve Bayes, K-NN, and Random Forest,” *Asian J. Soc. Humanit.*, vol. 1, no. 12, Art. no. 12, Sep. 2023.

[37] Adikavi Nannaya University, N. Venkateswara Rao, P. Venkateswara Rao, and Adikavi Nannaya University, “A REVIEW ON OBJECT DETECTION IN THE HEALTHCARE USING DEEP LEARNING TECHNIQUES,” *Int. J. Eng. Appl. Sci. Technol.*, vol. 6, no. 7, pp. 159–166, Nov. 2021.

[38] “A Study on Machine Learning And Deep Learning In Medical Imaging Emphasizes MRI: A Systematic Literature Review,” JMIR Preprints. Accessed: Jul. 01, 2024. [Online]. Available: https://preprints.jmir.org/preprint/47192

[39] P. Bhanarkar and D. S. Y. Amdani, “A Survey on Hyperspectral Image Classification and Object Detection Techniques,” *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 8, no. 1, pp. 236–249, Jan. 2022.

[40] X. Zhao and Z. Su, “Using deep learning in image hyper spectral segmentation, classification, and detection,” vol. 10697, p. 106972U, Feb. 2018.

[41] Jayadeva, R. Khemchandani, and S. Chandra, “TWSVM for Unsupervised and Semi-supervised Learning,” in *Twin Support Vector Machines: Models, Extensions and Applications*, Jayadeva, R. Khemchandani, and S. Chandra, Eds., Cham: Springer International Publishing, 2017, pp. 125–152.

[42] Y. Dorogyy and V. Kolisnichenko, “Unsupervised Pre-Training with Spiking Neural Networks in Semi-Supervised Learning,” in *2018 IEEE First International Conference on System Analysis & Intelligent Computing (SAIC)*, Oct. 2018, pp. 1–4.

[43] P. Purwono, A. Ma’arif, W. Rahmaniar, H. I. K. Fathurrahman, A. Z. K. Frisky, and Q. M. ul Haq, “Understanding of Convolutional Neural Network (CNN): A Review,” *Int. J. Robot. Control Syst.*, vol. 2, no. 4, Art. no. 4, 2022.

[44] N.-I. Galanis, P. Vafiadis, K.-G. Mirzaev, and G. A. Papakostas, “Convolutional Neural Networks: A Roundup and Benchmark of Their Pooling Layer Variants,” *Algorithms*, vol. 15, no. 11, Art. no. 11, Nov. 2022.

[45] K. N. N. Hlaing and A. K. Gopalakrishnan, “Myanmar paper currency recognition using GLCM and k-NN,” in *2016 Second Asian Conference on Defence Technology (ACDT)*, Jan. 2016, pp. 67–72.

[46] T. V. Dittimi, A. K. Hmood, and C. Y. Suen, “Multi-class SVM based gradient feature for banknote recognition,” in *2017 IEEE International Conference on Industrial Technology (ICIT)*, Mar. 2017, pp. 1030–1035.

[47] A. Tuama, F. Comby, and M. Chaumont, “Camera model identification with the use of deep convolutional neural networks,” in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec. 2016, pp. 1–6.

[48] "Some Insights Into Convolutional Neural Networks | IEEE Conference Publication | IEEE Xplore." Accessed: Jul. 01, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/8107753

[49] N. N. A. A. Hamid, R. A. Razali, and Z. Ibrahim, "Comparing bags of features, conventional convolutional neural network and AlexNet for fruit recognition," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 1, Art. no. 1, Apr. 2019.

[50] M. A. R. Khan, R. Talukder, M. A. Hossen, and N. Jahan, "Car make and model recognition using convolutional neural network: fine-tune AlexNet architecture," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 33, no. 1, Art. no. 1, Jan. 2024.

[51] "Development of Image-Based Emotion Recognition using Convolutional Neural Networks | IEEE Conference Publication | IEEE Xplore." Accessed: Jul. 01, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9526329

[52] "Predicting Retinal Diseases using Efficient Image Processing and Convolutional Neural Network (CNN)." Accessed: Jul. 01, 2024. [Online]. Available: https://www.researchgate.net/publication/357353993_Predicting_Retinal_Diseases_using_Efficient_Image_Processing_and_Convolutional_Neural_Network_CNN

[53] D. Aseffa, H. Kalla, and S. Mishra, "Ethiopian Banknote Recognition Using Convolutional Neural Network and Its Prototype Development Using Embedded Platform," *J. Sens.*, vol. 2022, pp. 1–18, Jan. 2022.

[54] G. Gebremeskel, T. A. Tadele, D. W. Girmaw, and A. O. Salau, "Developing a Model for Detection of Ethiopian Fake Banknote Using Deep Learning." Dec. 05, 2022.

[55] Y. D. WoldeHana and M. A. Yimer, "Counterfeit and Genuine Ethiopian Banknote Classification Model Using Similarity Learning." 2023.

[56] J. Yang, H. Xu, Y. Zhao, H. Liu, and W. Lu, "Stereoscopic image quality assessment combining statistical features and binocular theory," *Pattern Recognit. Lett.*, vol. 127, pp. 48–55, Nov. 2019,.

[57] "Text normalization with convolutional neural networks | International Journal of Speech Technology." Accessed: Jul. 01, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s10772-018-9521-x

[58] L. Nanni, G. Maguolo, S. Brahnam, and M. Paci, "Comparison of different convolutional neural network activation functions and methods for building ensembles." arXiv, Apr. 01, 2021.

Annexes

Annex A: The main components of Implementation

I. Data Preprocessing function

```
images_dataset=tf.keras.preprocessing.image_dataset_from
_directory(
 '/content/drive/MyDrive/final_data/last_Ethio_banknote'
,
 shuffle=True,
 image_size=(256, 256)

resize_and_rescale=tf.keras.Sequential([

    layers.experimental(IMAGE_SIZE, IMAGE_SIZE),
    layers.experimental.preprocessing.Rescaling(1.0/255)
])
```

II. Train, Validation and Testing building function

```
def get_dataset_partitions_tf(ds,train_split=0.8, val_split=0.1,
test_split=0.1,shuffle=True, shuffle_size=5000):
    ds_size=len(ds)
    if shuffle:
        ds=ds.shuffle(shuffle_size,seed=12)


    return train_ds, val_ds, test_ds
```

III. Model building function

```
input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
n_classes = 5
model = models.Sequential([
                                    layers.Conv2D(32,activation='relu',
input_shape=input_shape),
     layers.MaxPooling2D((2, 2)),
     layers.Conv2D(64,  kernel_size = (3,3),
activation='relu'),
     layers.MaxPooling2D((2, 2)),
     layers.Conv2D(64), activation='relu'),
     layers.MaxPooling2D((2, 2)),
     layers.Conv2D(64, activation='relu'),
     layers.MaxPooling2D((2, 2)),
```

```
        layers.Conv2D(64, (3, 3)),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, activation='relu'),
        layers.MaxPooling2D((2, 2)),
        layers.Flatten(),
        layers.Dense(64, activation='relu'),
        layers.Dense(n_classes, activation='softmax'),
    ])
    model.build(input_shape=input_shape)
```

IV. Model Compilation function

```
model. compile(
    optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_lo
gits=False),
    metrics=['accuracy']
)
```

V. Model Training function

```
history = model.fit(
    train_ds,
    batch_size=BATCH_SIZE,
    validation_data=val_ds,
    verbose=1,
    epochs=30,
)
```

VI. Model saving function

Model = "/content/drive/My Drive/Alex.h5"

VII. Making prediction function

```
making predictions
for images_batch, label_batch in test_ds.take(1):
  image1=images_batch[0].numpy().astype('uint8')
  label1=label_batch[0].numpy()
  print("predicting the first image")
  plt.imshow(image1)
  print('image1 True Label', class_names[label1])
  batch_prediction=model.predict(images_batch)
  print("Image1's predicted label:",
class_names[np.argmax(batch_prediction[0])])
```

VIII. Making prediction function with confidence in percentage

```
function to predict with confidence
def predict(model, img):
  img_array=tf.keras.preprocessing.image.img_to_array(images[i
].numpy())
  img_array=tf.expand_dims(img_array,0)

  predictions=model.predict(img_array)
  predicted_class=class_names[np.argmax(predictions[0])]
  confidence=round(100*(np.max(predictions[0])),2)
  return predicted_class, confidence
```

IX. Deep Learning wepApp with Streamlit

```
%%writefile app.py
import streamlit as st
import tensorflow as tf
st.set_option('deprecation.showfileUploaderEncoding', False)
@st.cache(allow_output_mutation=True)


def load_model():
     model=tf.keras.models.load_model('/content/Alex_model.hdf5')
  return model

   model = load_model()
file = st.file_uploader("Please upload the image .", type = ["jpg",
"png"])
import cv2
from PIL import Image, ImageOps
```

Signed Declaration Sheet

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university and that all sources of materials used for the thesis have been duly acknowledged.

Declared by:

Name: Hailemikael Tesfaw Haile

Signature: ____________________

Date: ______________________

Confirmed by advisor:

Name: Ayalew Belay (PhD)_____

Signature: ____________________

Date: ______________________