**A Final Project Report On**

**Fake Nepali Currency Detection Using Deep Learning**



Submitted in the Partial Fulfillment of the

Requirements for the Degree of Bachelor of Computer Engineering Awarded by Pokhara University

**Submitted By:**

**Dishan Bhandari**

**Supervisor:**

**Er. Rishi Saran Khanal**

**Assistant Professor**

**School of Engineering**

**Faculty of Science and Technology**

**Pokhara University**

**August 2024**

**A Final Project Report On**

**Fake Nepali Currency Detection Using Deep Learning**



Submitted in the Partial Fulfillment of the

Requirements for the Degree of Bachelor of Computer Engineering Awarded by Pokhara University

**Submitted By:**

**Dishan Bhandari**

**Supervisor:**

**Er. Rishi Saran Khanal**

**Assistant Professor**

**School of Engineering**

**Faculty of Science and Technology**

**Pokhara University**

**August 2024**

# **COPYRIGH**T

The author has granted authorization to the Pokhara University Central Library and Library of School of Engineering to grant unrestricted access to this report for inspection. Furthermore, the author has agreed to the potential for extensive reproduction of this report for educational purposes by either the supervisors or, in their absence, by the program coordinator wherein the project was done. It is understood that proper acknowledgment will be accorded to both the author of the report and the School of Engineering, Pokhara University in any utilization of the report's contents. Unauthorized copying, publication, or any other form of exploitation of this report for financial gain without explicit consent from the School of Engineering, Pokhara University and the author is strictly prohibited. Requests seeking permission to replicate or employ any segment of the report's content should be directed to:

Program Coordinator

School of Engineering

Pokhara University

# STUDENT'S DECLARATION

I hereby declare that this project work entitled **Fake Nepali Currency Detection Using Deep Learning** is based on my original work. All concepts, data, code, and any other work from external sources have been properly cited and referenced in accordance with the guidelines provided by School of Engineering, Pokhara University

I owe all the liabilities relating to the authenticity and originality of this project work and project report.

1. Dishan Bhandari ……………………………..

Date: ……………………….

# SUPERVISOR'S RECOMMENDATION

This is to certify that this project report entitled **Fake Nepali Currency Detection Using Deep Learning** prepared and submitted by below student in partial fulfilment of the requirements of the degree of Bachelor of Computer Engineering awarded by Pokhara University, has been prepared and completed under my supervision.

I hereby recommend the same for acceptance by School of Engineering, Pokhara University.

1. Dishan Bhandari

…………………………..

**Er. Rishi Saran Khanal**

Assistant Professor

School of Engineering

Pokhara University

Date: ……………………….



**School of Engineering**

**Pokhara University**

# EXTERNAL EXAMINER’S RECOMMENDATION

The undersigned certified that they have evaluated this project report entitled **Fake Nepali Currency Detection Using Deep Learning** submitted by **Dishan Bhandari** and his oral presentationfor partial fulfillment of the degree of Bachelor of Computer Engineering and recommended to the School of Engineering, Pokhara University for acceptance of this project work/report.

…………………………….

**Er. Santosh Panth**

Assistant Professor

Gandaki College of Engineering & Science

Lamachaur, Pokhara, Kaski

Date: ……………………….



**School of Engineering**

**Pokhara University**

# LETTER OF APPROVAL

This project report entitled **Fake Nepali Currency Detection Using Deep Learning** submitted by **Dishan Bhandari** for partial fulfillment of the degree of Bachelor of Computer Engineering has been accepted by the School of Engineering, Pokhara University upon the recommendations of Supervisor and with the approval by the following examiner.

**External Examiner:**

**Er. Bhesh Bahadur Thapa**

Assistant Professor

(Coordinator)

Bachelor of Computer Engineering

Bachelor of Software Engineering

School of Engineering

Pokhara University

**Er. Santosh Panth**

Assistant Professor

Gandaki College of Engineering

& Science

Lamachaur, Pokhara, Kaski

Date: ……..…………………

# ABSTRACT

Technological advancements in the printing and scanning industries have significantly intensified the counterfeiting problem, making it easier for counterfeiters to produce near-perfect replicas of genuine currency. This rise in counterfeit currency not only threatens the economy by undermining the value of legitimate money but also poses challenges to financial institutions and consumers alike. Therefore, effective detection of counterfeit currency is of utmost importance. Traditional methods, often reliant on hardware systems or image processing techniques, have proven to be less efficient and time-consuming, especially in dealing with sophisticated counterfeits. To address these challenges, I propose a novel approach that leverages deep learning, specifically a deep convolutional neural network (CNN), for counterfeit currency detection. Unlike conventional methods, my system examines currency images using advanced neural networks, enabling more accurate and faster detection of fake notes. By employing this AI-driven approach, I aim to enhance the reliability and efficiency of counterfeit detection, ultimately contributing to a more secure financial system.

**Keywords:** *Artificial Neural Network, Convolutional Neural Network, Counterfeit Currency, Currency Detection, Financial Fraud*

# ACKNOWLEDGEMENT

As I conclude the development of the "Fake Nepali Currency Detection Using Deep Learning" project, I would like to extend my deepest gratitude to all those who have contributed to this innovative endeavor. The insights gained from my mentors and colleagues, along with their invaluable feedback, have been instrumental in refining my approach and achieving my goals. I am especially grateful to the Department of Computer and Software Engineering for providing the necessary resources and environment to carry out this research effectively.

I owe a great debt of gratitude to my supervisor, Er. Rishi Saran Khanal whose guidance, encouragement, and unwavering support were crucial throughout the development process. Their expertise and mentorship have provided us with the confidence to navigate the challenges of this complex project.

I also extend my sincere thanks to the financial institutions and professionals who shared their experiences and insights on currency handling and detection. Their practical feedback has significantly shaped the direction of my project.

Last but not least, I would like to express my heartfelt appreciation to my families and friends for their continuous support and encouragement. Your belief in my capabilities has been a constant source of motivation. This project represents a significant milestone in my academic journey, and I are committed to continuing my efforts to enhance the reliability and accuracy of currency detection systems in Nepal. I believe this work will make a meaningful contribution to the financial security of my nation, and it is thanks to the collective contributions of everyone involved.

Dishan Bhandari

Date: ……………………………….

# TABLE OF CONTENTS

[COPYRIGHT i](#_Toc175837580)

[STUDENT'S DECLARATION ii](#_Toc175837581)

[SUPERVISOR'S RECOMMENDATION iii](#_Toc175837582)

[EXTERNAL EXAMINER’S RECOMMENDATION iv](#_Toc175837583)

[LETTER OF APPROVAL v](#_Toc175837584)

[ABSTRACT vi](#_Toc175837585)

[ACKNOWLEDGEMENT vii](#_Toc175837586)

[TABLE OF CONTENTS viii](#_Toc175837587)

[LIST OF FIGURES xi](#_Toc175837588)

[LIST OF TABLES xii](#_Toc175837589)

[LIST OF ABBREVIATIONS xiii](#_Toc175837590)

[CHAPTER 1 1](#_Toc175837591)

[INTRODUCTION 1](#_Toc175837592)

[1.1 Background 1](#_Toc175837593)

[1.2 Problem Statement 2](#_Toc175837594)

[1.3 Objectives 2](#_Toc175837595)

[1.4 Scope and Limitation 3](#_Toc175837596)

[1.5 Significance 3](#_Toc175837597)

[1.6 Contribution 4](#_Toc175837598)

[1.7 Key Features 4](#_Toc175837599)

[1.8 Report Organization 5](#_Toc175837600)

[CHAPTER 2 6](#_Toc175837601)

[LITERATURE REVIEW 6](#_Toc175837602)

[CHAPTER 3 9](#_Toc175837603)

[SYSTEM ANALYSIS AND DESIGN 9](#_Toc175837604)

[3.1 System Analysis 9](#_Toc175837605)

[3.1.1 Requirements Gathering 9](#_Toc175837606)

[3.1.2 Functionalities and Features 9](#_Toc175837607)

[3.1.3 Design Considerations 10](#_Toc175837608)

[3.2 Requirement Analysis 10](#_Toc175837609)

[3.2.1 Functional Requirements 10](#_Toc175837610)

[3.2.2 Non-Functional Requirements 11](#_Toc175837611)

[3.2.3 Systems Requirements 11](#_Toc175837612)

[3.3 System Design 12](#_Toc175837613)

[3.4 System Architecture 14](#_Toc175837614)

[3.4.1 Deep Learning 14](#_Toc175837615)

[3.4.2 Supervised Learning 15](#_Toc175837616)

[3.4.3 Artificial Neural Network 15](#_Toc175837617)

[3.4.4 Convolution Neural Network (CNN) 16](#_Toc175837618)

[3.4.5 Data Flow Diagram 21](#_Toc175837619)

[3.4.6 Use Case Diagram 23](#_Toc175837620)

[3.5 Development Methodology 24](#_Toc175837621)

[3.5.1 Agile Methodology 24](#_Toc175837622)

[CHAPTER 4 26](#_Toc175837623)

[IMPLEMENTATION AND TESTING 26](#_Toc175837624)

[4.2 Implementation 26](#_Toc175837625)

[4.2.1 Dataset Collection 26](#_Toc175837626)

[4.2.2 Data Preprocessing 26](#_Toc175837627)

[4.2.3 Model Architecture 28](#_Toc175837628)

[4.2.4 Model Compilation and Training 28](#_Toc175837629)

[4.2.5 Integration with User Interface 29](#_Toc175837630)

[4.2.6 Deployment and Scalability 29](#_Toc175837631)

[4.2.7 Challenges and Solutions 29](#_Toc175837632)

[4.2 Testing 30](#_Toc175837633)

[4.2.1 Testing Methodology 30](#_Toc175837634)

[4.2.2 Evaluation Metrics 30](#_Toc175837635)

[4.3 Test Results 30](#_Toc175837636)

[4.3.1 Performance Analysis of 500 and 100 Notes 31](#_Toc175837637)

[CHAPTER 5 35](#_Toc175837638)

[CONCLUSION AND FUTURE RECOMMENDATION 35](#_Toc175837639)

[5.1 Conclusion 35](#_Toc175837640)

[5.2 Future Recommendations 35](#_Toc175837641)

[5.2.1 Model Enhancements 35](#_Toc175837642)

[5.2.2 Expanded Dataset 36](#_Toc175837643)

[5.2.3 Real-Time Performance Optimization 36](#_Toc175837644)

[5.2.4 User Experience and Accessibility 36](#_Toc175837645)

[5.2.5 Deployment and Scalability 37](#_Toc175837646)

[5.2.6 Research and Development 37](#_Toc175837647)

[5.3 Final Thoughts 37](#_Toc175837648)

[REFERENCES 39](#_Toc175837649)

# LIST OF FIGURES

[Figure 3.1: System Design 12](#_Toc175835138)

[Figure 3.2: Deep Learning Process 14](#_Toc175835139)

[Figure 3.3: Supervised Learning 15](#_Toc175835140)

[Figure 3.4: Artificial Neural Network 16](#_Toc175835141)

[Figure 3.5: Convolutional Neural Network (CNN) 17](#_Toc175835142)

[Figure 3.6: Convolutional Layer 18](#_Toc175835143)

[Figure 3.7: Rectified linear units 19](#_Toc175835144)

[Figure 3.8: Pooling Layer 20](#_Toc175835145)

[Figure 3.9: Fully Connected layer 20](#_Toc175835146)

[Figure 3.10: Flattening 21](#_Toc175835147)

[Figure 3.11: Level 0 DFD 21](#_Toc175835148)

[Figure 3.12: Level 1 DFD 22](#_Toc175835149)

[Figure 3.13: Use Case Diagram 23](#_Toc175835150)

[Figure 3.14: Agile Methodology 24](#_Toc175835151)

[Figure 4.1: Model accuracy and Loss Curves for 500 Notes 31](#_Toc175835152)

[Figure 4.2: Model accuracy and Loss Curves for 1000 Notes 32](#_Toc175835153)

[Figure 4.3: Confusion Matrix for 500 Notes 33](#_Toc175835154)

[Figure 4.4: Confusion Matrix for 1000 Notes 34](#_Toc175835155)

# LIST OF TABLES

Table 2.1: Comparison Table ..……………………………..…………………………….8

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| **AI** | Artificial Intelligence |
| **ANN** | Artificial Neural Network |
| **CNN** | Convolutional Neural Network |
| **DP** | Deep Learning |
| **FCN** | Fully Connected layer |
| **ReLU** | Rectified Linear Network |
| **SVM** | Support Vector Machine |

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

Money is an essential component of modern life and is employed in the day-to-day operations of organizations worldwide. In Nepal, cash is the most widely accepted means of payment, playing a significant role in the country's economy. Its importance cannot be overstated. However, counterfeiting has become a significant issue that threatens economic stability and public trust in the monetary system [1]. This problem is not unique to Nepal; many countries face similar challenges with counterfeit currency [2].

Since the advent of cash, counterfeiting has evolved, with counterfeiters using increasingly sophisticated techniques to produce fake banknotes that are difficult to distinguish from genuine ones. Advances in printing technology have made it easier for counterfeiters to create high-quality fakes, posing a significant security risk [3]. To combat this, various anti-counterfeiting measures have been implemented, including holograms, watermarks, and advanced printing methods [4].

Despite these measures, counterfeiters continue to devise new methods to produce convincing fake currency. This ongoing challenge underscores the need for effective detection systems. Recent technological advancements offer promising solutions for identifying counterfeit money. Innovations such as digital image processing and machine learning have shown potential in enhancing detection accuracy [5]. Digital image processing involves analyzing digital images of banknotes to identify signs of counterfeiting, while machine learning algorithms can improve detection by learning from large datasets of authentic and fake currency [6].

The proposed system aims to address the counterfeit detection problem by leveraging these technologies. By creating an accessible and user-friendly interface, the system will allow users to upload photographs of banknotes for examination. The system will provide feedback on the authenticity of the notes and offer detailed information if counterfeiting is detected. This approach will help prevent financial losses and assist law enforcement in tracking counterfeiters [7].

In Nepal, anti-counterfeiting measures have been introduced through collaboration between the government and financial institutions. These include various security features like holograms and watermarks. Despite these efforts, counterfeiters continue to find new ways to produce convincing fake money, highlighting the need for improved detection methods [8].

This project proposes an algorithm-based system for detecting counterfeit cash, aiming to enhance the detection process and offer a solution to the persistent problem of counterfeit currency [9]. Machine learning, a subset of artificial intelligence, involves algorithms that learn from data and make predictions [10]. By applying machine learning, a system can be built to accurately differentiate between genuine and counterfeit currency, representing a significant advancement in combating counterfeiting [11]. The system will analyze digital photographs of Nepalese banknotes to assess their authenticity. The initial step involves creating a data of authentic and counterfeit banknotes. This data will then be used to train a machine learning model capable of identifying genuine notes from counterfeits [12].

## 1.2 Problem Statement

While traditional techniques like UV detection and watermark checking have proven invaluable in the fight against counterfeit money, they have limitations in terms of cost and time, and they are not always appropriate for large-scale evaluations. The implementation of deep learning algorithms for predicting counterfeit currency has been investigated in an attempt to address these issues. Still, making precise projections is still a difficult undertaking. The complexity comes from the fact that counterfeiters can employ a range of methods and resources, which makes it difficult to identify recurring trends. Security features, holograms, and other features can be found in counterfeit money. Because of the wide range of techniques used by counterfeiters, it is challenging to anticipate and identify phony currency with high precision. To address these issues and raise the accuracy of fake cash detection systems, new ideas and improved techniques are therefore desperately needed.

## 1.3 Objectives

* To provide a reliable and accurate fake currency detection system.

## 1.4 Scope and Limitation

This project is designed to address the challenges associated with detecting counterfeit currencies in the real world. The scope of the project encompasses the following key areas:

* **Currency Focus**: The system targets the detection of fake Nepali currency using deep learning techniques.
* **Model-Based Detection**: The system uses deep learning techniques to identify counterfeit notes, focusing on distinguishing features of Nepali currency, such as watermarks, holograms, and other security features.
* **Deployment**: The system is intended to be deployed as a desktop or web-based application, with the potential for integration into larger financial systems or for currency verification.
* **Automation**: The project aims to replace manual inspection with an automated system that reduces the need for human intervention and provides quick results.

Implementing any new technology system comes with inherent limitations that must be acknowledge and mitigated over time for a successful adoption. The limitations of the project are as follows:

* **Dataset Dependency**: The effectiveness of the deep learning model is heavily dependent on the quality and variety of the dataset. The dataset, being focused solely on Nepali currency, may limit the system's adaptability to new or rare counterfeit techniques that were not included in the training data.
* **Image Quality**: The system's accuracy can be affected by the quality of the images provided for analysis. Poor lighting, low resolution of the images can result in false positives or false negatives.
* **Hardware Requirements**: The system requires significant computational resources, impacting performance on less powerful devices.
* **Real-Time Limitation**: The system is currently designed for batch processing of images and may not support real-time detection via live camera feeds without further optimization.

## 1.5 Significance

The fake Nepali currency detection system using deep learning is a significant advancement in the fight against counterfeiting, a problem that poses serious threats to the economy and financial security. Counterfeit currency undermines the value of legitimate money, erodes public trust, and can lead to economic instability. By offering a more accurate and efficient method for detecting fake currency compared to traditional techniques, this system enhances the security of financial transactions and reduces the risk of counterfeit notes entering circulation.

Moreover, the automation and scalability of the system make it a valuable tool for banks, businesses, and other stakeholders, ensuring that the integrity of Nepali currency is maintained. This project not only addresses a critical need in the financial sector but also contributes to the broader goal of economic stability and security in Nepal.

## 1.6 Contribution

This project contributes to the field of counterfeit detection by introducing a deep learning-based approach specifically tailored for Nepali currency. Unlike traditional methods that rely on hardware or basic image processing, this system leverages advanced neural networks to improve the accuracy and speed of detecting fake currency. By providing an automated solution, it reduces the need for manual intervention and offers a scalable tool that can be adopted by banks, businesses, and individuals. Additionally, the project enhances the understanding of how deep learning can be applied to real-world problems in financial security, serving as a foundation for future research and development in counterfeit detection technologies. The project also supports the local economy by providing a practical tool that directly addresses the issue of counterfeit Nepali currency, helping to prevent financial losses and maintain trust in the monetary system.

## 1.7 Key Features

My project focuses on developing a comprehensive Fake Nepali Currency Detection System with key features that include:

* **Deep Learning Model**: Utilizes a convolutional neural network (CNN) to accurately detect counterfeit Nepali currency notes by analysing complex patterns and features.
* **Automated Detection**: Provides a fully automated process for detecting fake currency, minimizing the need for manual inspection.
* **User-Friendly Interface**: Includes a simple and intuitive interface that allows users to upload images of currency notes for real-time analysis.
* **High Accuracy**: Trained on a diverse dataset to ensure high accuracy in distinguishing between genuine and counterfeit notes.
* **Scalable Deployment**: Can be deployed as a desktop or web-based application, making it accessible to a wide range of users, from financial institutions to small businesses.
* **Fast Processing**: Offers quick detection results, improving efficiency in verifying currency, especially in high-volume scenarios.
* **Customizable**: The system can be retrained or adjusted to adapt to new counterfeit techniques or updated currency designs.

## 1.8 Report Organization

The project report is structured into several chapters, each addressing specific aspects of the “Fake Nepali Currency Detection System Using Deep Learning” development and implementation. The organization of the report is designed to provide a comprehensive understanding of the project’s context, challenges, methodologies, and outcomes.

This report is organized to provide a comprehensive overview of the fake Nepali currency detection project using deep learning. It begins with an **Introduction**, outlining the significance of addressing counterfeit currency issues and stating the primary objective of employing deep learning techniques for detection. The **Literature Review** follows, offering insights into previous research and methodologies in currency detection and identifying gaps that this project aims to address. The **Methodology** section focuses on system analysis and design, where I describe the requirements, depict system diagrams, model data and processes, and elucidate my development methodology. This is followed by the **Implementation and Testing** section, where the performance metrics of the model are presented and analyzed. The **Conclusion and Future Recommendations** interprets the results, addresses challenges encountered, and discusses the limitations of the approach and also a conclusion that summarizes the findings and suggests future research directions. Finally, the **References** section cites all sources used, and the **Appendices** include supplementary material such as code snippets and detailed experiment setups.

# CHAPTER 2

# LITERATURE REVIEW

Producing fake currency is not a new phenomenon rather, it has been a problem since ancient Greece's coinage of money dates back to about 600 B.C. Coin rims were routinely cut off during that time period in order to extract precious metal, which was subsequently used to create counterfeit money utilizing the metal. Paper money originated in China in the 1200s. The wood from mulberry trees was utilized to make money before then. The guards were in charge of keeping an eye on the mulberry forests at the period, and trying to pass off counterfeit money as genuine was punishable by death [1].

I know from my historical research that money counterfeiting is a long-standing illegal practice. This problem still exists in the modern period, which is why banknotes have various printing methods and other elements added to them to make it simpler to spot fakes [2].

But as science and technology advance, new techniques for identifying fake currency are appearing. The accuracy of this process can be increased while being made much simpler by these new technologies. Some of the more recent techniques for spotting counterfeit money include holograms, colorful stripes, counterfeit pens with iodine (which reacts with the starch in paper money), and UV scanning. Another contemporary technique is holograms [3].

However, non-experts cannot access any of the contemporary equipment used in banks these days, which is why the problem of identifying counterfeit money persists in society [4]. In this, I suggest a technique that might be used by laypeople as a tool for identifying counterfeit currency. For the objective of developing a robust system for the identification of counterfeit money that can benefit society as a whole, the application of digital image processing offers us an approach that is both economical and efficient [5].

Recognizing and avoiding the use of counterfeit currency is a significant issue of concern in a number of countries, including Nepal. Many academics have proposed a variety of approaches as possible answers to the issue of counterfeit currency. Some of these approaches include machine learning, the processing of images, and the detection of watermarks.

Deep learning was used in the study that Pradhan and his colleagues (2021) undertook to develop a method that could detect fake Nepalese rupee notes. This method was based on pattern recognition. The authors utilized a convolutional neural network, more commonly known as a CNN, in order to determine if the banknotes were genuine or fake. The fact that the model was able to achieve an accuracy of 99.5% on the dataset that was being tested is evidence that the technique that was provided was successful [6]. In the research that Adhikari and his colleagues (2020) conducted on the identification of counterfeit Nepali rupee notes, they provided an approach that was quite similar to it and was based on image processing. The researchers made use of a wide variety of image processing techniques, such as morphological operations, edge detection, and segmentation, in order to extract information from the currency notes. They ran the notes through a support vector machine (SVM) classifier in an attempt to identify whether or not the notes were genuine. The fact that the system that was presented was capable of achieving an accuracy of 98.7% was evidence of the successful execution of the plan [7].

In a different piece of study, Kafle and his colleagues (2020) described a technique that makes use of watermarks as a means of distinguishing genuine Nepalese rupee banknotes from those that have been tampered with. The researchers came to the conclusion that the use of the watermark that is present on genuine currency notes as a point of reference was successful in determining the authenticity of counterfeit notes. The approach that was recommended had a success rate of 96.3% when it comes to determining the authenticity of counterfeit cash [8]. A recent study that was carried out by Shrestha and colleagues (2021) presented a method that is based on machine learning for the identification of counterfeit Nepalese currency notes. The authors made use of a variety of different machine learning approaches, such as decision trees, K-nearest neighbor analysis, and logistic regression, in order to assess whether or not the cash notes were genuine or counterfeit. The fact that the proposed method managed to achieve an accuracy of 98.5 percent is evidence that the technique was successful [9].

In general, the research that has been done over the course of the last few years has led to the creation of effective procedures that can detect fake Nepali currency notes. This progress has been made over the course of the previous several years. Methods that are based on machine learning and image processing have shown some positive results; as a result, the focus of future research may be on further improving approaches that are similar to these types.

**Table 2.1: Comparison Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Title | Approach | Features | Classifier | Accuracy |
| Pradhan et al.  (2020) [6] | Deep learning | Image pixels | CNN | 99.5% |
| Adhikari et al.  (2021) [7] | Image Processing | Morphological operations, edge detection, segmentation | SVM | 98.7% |
| Kafle et al.  (2020) [8] | Watermark detection | Watermark | Image subtraction | 96.3% |
| Shrestha et al.  (2021) [9] | Machine learning | Statistical features | Decision tree,  K-nearest neighbor, logistic regression | 98.5% |

As can be seen from the table, all of the methods achieved a high level of accuracy when it came to identifying counterfeit Nepalese currency notes. The method of deep learning that was offered by Pradhan and colleagues reached the greatest level of accuracy, 99.5%, followed by the method of image processing that was proposed by Adhikari and colleagues, which achieved 98.7% accuracy. The strategy based on watermarks that was offered by Kafle and colleagues. reached a level of accuracy of 96.3%, whereas the approach based on machine learning that was presented by Shrestha et al. achieved a level of accuracy of 98.5%.

# CHAPTER 3

# SYSTEM ANALYSIS AND DESIGN

## 3.1 System Analysis

I identified the product requirements for the currency detection and counterfeit detection web app and with the help of these requirements I met the final product as per needs. The product requirements for the web app were categorized into functional and non-functional requirements. Functional requirements defined the specific features and functionalities the web app should have, such as currency recognition and counterfeit detection. Non-functional requirements, on the other hand, focused on aspects like performance, security, scalability, and user experience.

### 3.1.1 Requirements Gathering

The primary objective of the requirement gathering process was to identify the fake currency that are encountered in the real world and for this I focused on several key areas. **Stakeholders** included **Nepal Rastra Bank**, which provided the critical dataset after multiple attempts, and my academic advisors. The **objective** was to develop a model with high accuracy in distinguishing genuine from counterfeit currency. **Functional requirements** encompassed obtaining a diverse set of currency images, achieving high detection accuracy, and creating a user-friendly interface. **Non-functional requirements** included performance efficiency, data security, and usability. **Constraints** involved technical limitations and project timelines. **Assumptions** were that the dataset from **Nepal Rastra Bank** was representative and that stakeholder support would be ongoing.

### 3.1.2 Functionalities and Features

Based on the gathered requirements, the functionalities and features of the system were defined:

* **Currency Detection**: Utilizes a deep learning model to accurately identify genuine versus counterfeit Nepali banknotes.
* **Image Upload**: Allows users to upload images of currency notes for analysis through a user-friendly interface.
* **Real-Time Results**: Provides immediate feedback on the authenticity of the uploaded currency images.
* **High Accuracy**: Designed to achieve high detection accuracy by leveraging a robust dataset and advanced deep learning techniques.

### 3.1.3 Design Considerations

In the design phase of the analysis, design considerations were established to guide the development process:

* **Model Architecture**: Choosing an effective deep learning model (e.g., CNN) for high accuracy and efficiency.
* **User Interface**: Designing an intuitive and accessible interface for easy image uploads and result viewing.
* **Performance**: Ensuring quick processing and scalability to handle varying data volumes.
* **Hardware and Software**: Addressing resource constraints and ensuring compatibility with different devices.
* **Model Evaluation**: Using performance metrics and feedback to refine and improve the model.

## 3.2 Requirement Analysis

The requirements for a Fake Nepali Currency Detection Using Deep Learning project can vary depending on the specific needs in the future of the project. However, here are some general functional, non-functional and system requirements that have been considered:

### 3.2.1 Functional Requirements

* **Deep Learning Algorithms:** The application uses Deep learning algorithms to analyse the currency notes and identify patterns that distinguish between genuine and fake currency notes.
* **User Interface:** The application provides a simple user interface for users to upload images of currency notes and receive results.
* **Results Display:** The application displays the results of the analysis, indicating whether the currency note is genuine or fake.
* **Multiple Image Formats:** The application is able to accept image in JPG, JPEG format.

### 3.2.2 Non-Functional Requirements

* **Performance:** The system processes currency images and deliver results quickly to ensure efficient user interaction.
* **Scalability:** The system is capable of handling increasing volumes of data and user requests without performance degradation.
* **Reliability:** The system operates consistently and accurately, with minimal downtime and high fault tolerance.
* **Usability:** The interface is user-friendly and accessible to individuals with varying levels of technical expertise.
* **Compatibility:** The system is compatible with various devices and operating systems to ensure broad accessibility.
* **Maintainability:** The system is designed for easy updates and maintenance to accommodate future improvements and bug fixes.

### 3.2.3 Systems Requirements

I have considered some of the system requirements that are needed to effectively develop, deploy, and to run my system smoothly. They are listed below:

**1. Hardware Requirements:**

* **Processor**: A multi-core CPU (e.g., Intel i5/i7 or equivalent) for efficient processing.
* **Memory (RAM)**: At least 8 GB of RAM for smooth operation and handling of large datasets.
* **Storage**: Sufficient storage space (e.g., 500 GB or more) for saving datasets, models, and results.
* **Graphics Processing Unit (GPU)**: A dedicated GPU (e.g., NVIDIA GTX/RTX) if using GPU acceleration for deep learning model training.

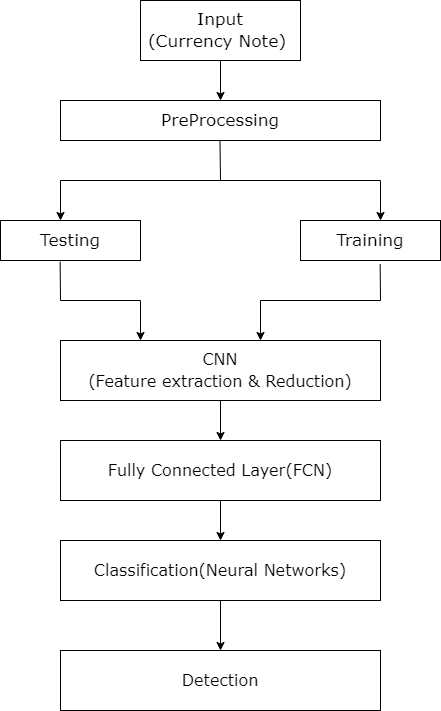
**2. Software Requirements:**

* **Operating System**: Compatible with major operating systems (e.g., Windows 10/11, Linux, macOS).
* **Development Environment**: Integrated Development Environment (IDE) or code editor (e.g., PyCharm, VS Code, Google Collab Notebook).
* **Programming Languages**: Python for model development and implementation.
* **Libraries and Frameworks**: Necessary libraries and frameworks such as TensorFlow, Keras, NumPy, Pillow, and Streamlit.
* **Database**: Optional, depending on the need for persistent storage of results and user data.

**3. Network Requirements:**

* **Internet Connection**: Stable internet connection for downloading libraries, updates, and accessing online resources.

## 3.3 System Design



**Figure 3.1: System Design**

The above flowchart shows about the system design outlining the process of fake currency detection using a Convolution Neural Network (CNN). Here’s an elaboration of the process:

1. **Input (Currency Note):** This is the first step where the system receives an image of a Nepali currency note. This image can be captured through various devices like a camera or scanner. The quality and clarity of the image are crucial for further processing.
2. **Pre-processing:** In this step, the input image undergoes several pre-processing techniques to enhance its quality and make it suitable for feature extraction. Pre-processing may include:

 **Noise Removal**: Using filters to remove any noise or irrelevant data from the image.

 **Resizing**: Adjusting the image size to a standard format(224x224) suitable for the neural network.

 **Normalization**: Scaling pixel values to a range (e.g., 0 to 1) for better performance during training and testing.

 **Augmentation**: Generating variations of the image (e.g., rotations, flips) to improve the robustness of the model.

1. **Testing and Training Split:** The dataset is split into two subsets: training and testing.

 **Training Set**: Used to train the model by adjusting its weights and biases.

 **Testing Set**: Used to evaluate the performance of the trained model on unseen data.

1. **Convolutional Neural Network (CNN) - Feature Extraction & Reduction:** The CNN is a deep learning model specifically designed for image processing tasks.

 **Feature Extraction**: The CNN automatically identifies important features of the currency note, such as patterns, textures, and symbols, through multiple convolutional layers.

 **Feature Reduction**: Pooling layers reduce the dimensionality of the feature maps, retaining the most significant features while reducing computational complexity.

1. **Fully Connected Layer (FCN):** After feature extraction, the resulting feature maps are flattened and passed through fully connected layers. These layers combine the extracted features to make predictions. The fully connected layers learn complex relationships between the features and the target classes (real or fake currency).
2. **Classification (Neural Networks):** The final fully connected layer is connected to the output layer, which classifies the currency note as either real or fake.

 **Activation Function**: Sigmoid activation functions is used in the output layer.

 **Loss Function**: The loss function (i.e., binary cross-entropy for binary classification) is used during training to improve the accuracy of the classification.

1. **Detection:** Based on the classification, the system provides the final output indicating whether the currency note is genuine or counterfeit. The system could provide a simple "Real" or "Fake" label.

## 3.4 System Architecture

### 3.4.1 Deep Learning

Deep Learning (DL) is a subset of machine learning that revolves around artificial neural networks inspired by the human brain's structure. Characterized by deep neural architectures with multiple layers, DL algorithms excel in automatically learning intricate representations of data. Common architectures include Convolutional Neural Networks (CNNs) for image-related tasks, Recurrent Neural Networks (RNNs) for sequential data, and transformer architectures for natural language processing. Deep learning has demonstrated remarkable success in complex tasks, such as image and speech recognition, due to its ability to discern hierarchical features from vast datasets. While it demands substantial computational resources and extensive labeled data for effective training, the unparalleled capacity of deep learning models to uncover patterns and relationships has positioned them at the forefront of artificial intelligence advancements.

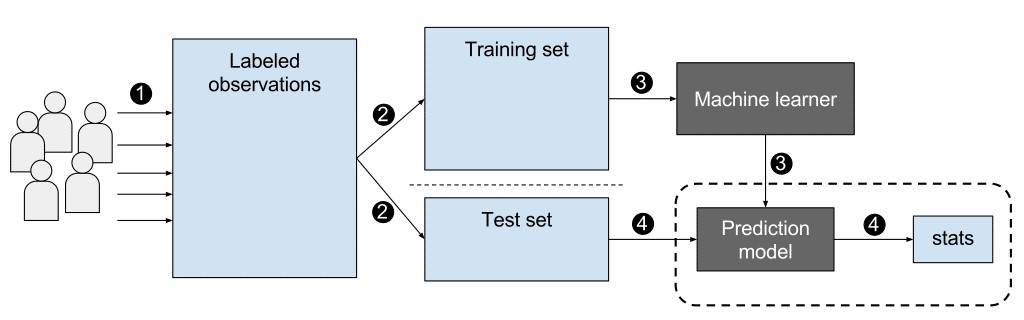
****

**Figure 3.2**: **Deep Learning Process**

(Source: *https://indiantechwarrior.com/the-complete-guide-to-deep-belief-networks/)*

### 3.4.2 Supervised Learning

One of the subfields that fall under the umbrella of machine learning and artificial intelligence is supervised learning, commonly known as supervised machine learning. It is characterized by the utilization of labeled datasets for the purpose of training algorithms that accurately classify data or forecast outcomes. The model uses a technique called reinforcement learning to modify its weights whenever new input data is added to it. This helps guarantee that the model is adequately suited to the data it has been given. Organizations may tackle a range of real-world problems at scale with the assistance of supervised learning, such as sorting spam into a different category than regular email messages in your inbox. Examples of supervised learning techniques include the decision tree, Naive Bayes, neural networks, k-Nearest Neighbors, and Support Vector Machines (SVM), which are all classification algorithms.

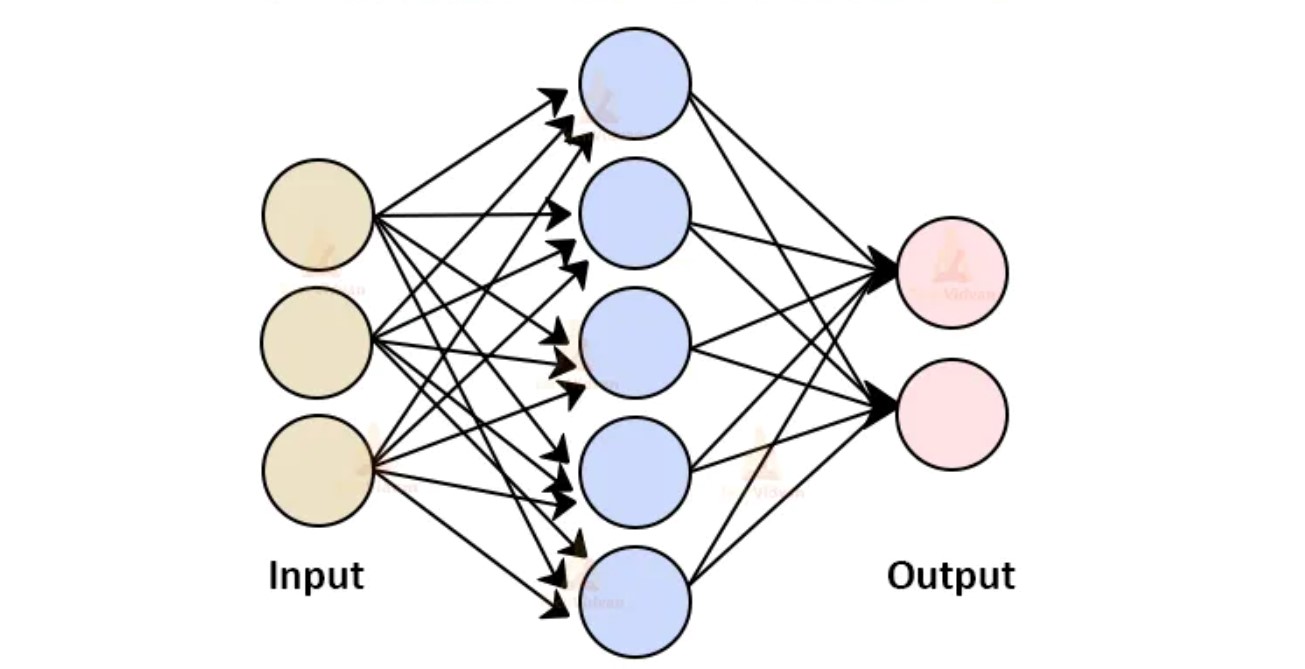


**Figure 3.3: Supervised Learning**

(Source: *https://www.geeksforgeeks.org/supervised-unsupervised-learning/)*

### 3.4.3 Artificial Neural Network

The biological neurons that make up the human brain are modeled after their digital counterparts in artificial neural networks, which are made up of numerous nodes. A network of connections between the neurons, known as axons, allows for communication between the cells. The nodes are responsible for receiving data as input and performing operations on that data. The outcomes of these computations are communicated to subsequent neurons. The value output at each node is called the activation of that node. There is a weight connected with each individual link. ANNs can learn, which is accomplished by adjusting the weight values.



**Figure 3.4: Artificial Neural Network**

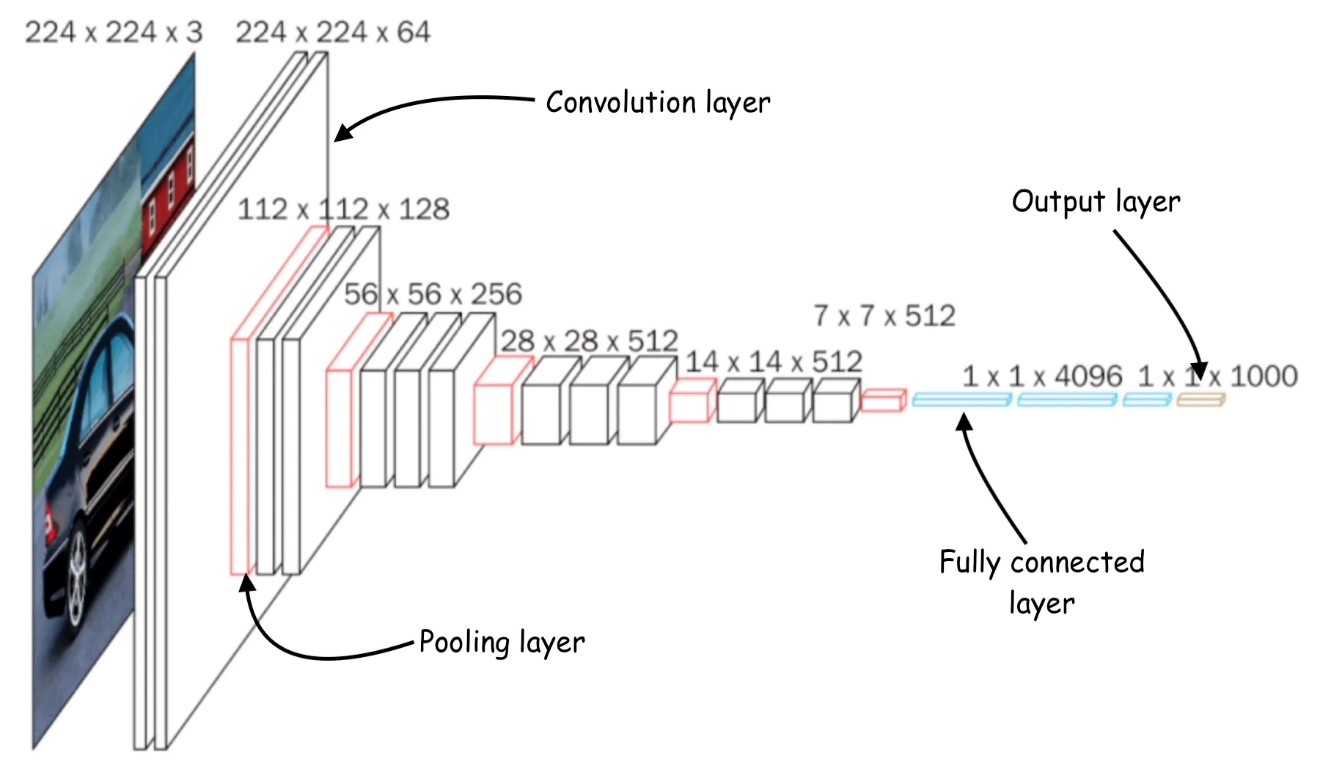
(Source: *https://blog.knoldus.com/architecture-of-artificial-neural-network/*)

### 3.4.4 Convolution Neural Network (CNN)

One class of Deep Learning Neural Networks is called a Convolution Neural Network. It was really modeled after the biological process that occurs in animal visual systems. Convolutional Neural Networks, much like other types of neural networks, are constructed out of neurons that have learnable weights and biases. Each neuron gets many inputs, computes a weighted sum over those inputs, sends it via an activation function, and then produces an output in response. The term "convolutional neural network" refers to the fact that the network makes use of a mathematical process referred to as "convolution." The convolution operation is a specialized form of linear operation. Simply put, they are neural networks with at least one of their layers equipped with a convolutional layer in place of a general matrix multiplication layer. Convolution is an effective method for the extraction of picture features, and it also serves as the ideal realization of an optic nerve cell that only responds to the stimuli in its receptive area. The first Convolutional Neural Network, known as

LeNet-5 was presented for the first time in a paper authored in 1998 by Bengio, Le Cun, Bottou, and Haffner. In this study, LeNet-5 demonstrated its ability to recognize digits from hand-written numbers. A CNN has

* Convolutional layers
* ReLU layers
* Pooling layers
* Fully connected layer



**Figure 3.5: Convolutional Neural Network (CNN)**

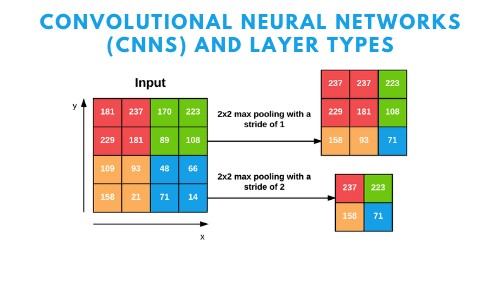
(Source: *https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/*)

i. Convolution Layer

The input image is given either a filter or a kernel, and convolution is then done on it via CNN. The beneficial attribute of being translationally invariant is possessed by convolution. This means, intuitively speaking, that each convolution filter represents a feature of interest (for example, the pixels that make up the letters), and the Convolutional Neural Network algorithm learns which features make up the reference that is ultimately produced (i.e. the alphabet). The following are the three components that are brought into play during the convolution operation:

* Input image
* Feature detector/Kernel/Filter
* Feature map

The first ConvLayer is traditionally in charge of capturing low-level information such as edges, color, the orientation of gradients, and so on. With further layers, the architecture can adjust to High-Level features as well, providing us with a network that has a comprehensive understanding of the images contained in the data set, comparable to how I would have it if I were doing it ourselves. Convolution is a method for learning picture characteristics that uses small squares of input data to maintain the spatial link between pixels. Because of this, every image can be thought of as a matrix consisting of the values of its individual pixels. Consider now an image with a size of 5 by 5 with pixel values that are only 0 and 1.

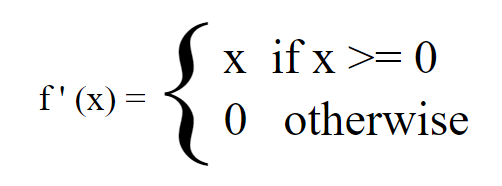


**Figure 3.6: Convolutional Layer**

(Source: *https://pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/)*

**ii. ReLU (Rectified Linear Units) Layers**

It is standard practice to apply a nonlinear layer, also known as an activation layer, immediately after each convolution layer that comes before it. The objective of this layer is to bring nonlinearity to a system that up to this point has largely just been computing linear operations (just element-wise multiplications and summations). In the past, nonlinear functions such as tanh and sigmoid were utilized; however, researchers have shown that ReLU layers perform far better. This is because the network can train much faster using these layers (because of the increased computing efficiency), all without sacrificing accuracy significantly. Each and every one of the values in the input volume are subjected to the ReLU layer's application of the function f(x) = max (0, x). In the most fundamental sense, all the negative activations are simply reset to zero by this layer. The output of this layer is a rectified feature map, and it contributes to an increase in the nonlinear properties of both the model and the overall network. However, it does so without having any impact on the receptive fields of the convolution layer. You may see a graph of a ReLU function down below:



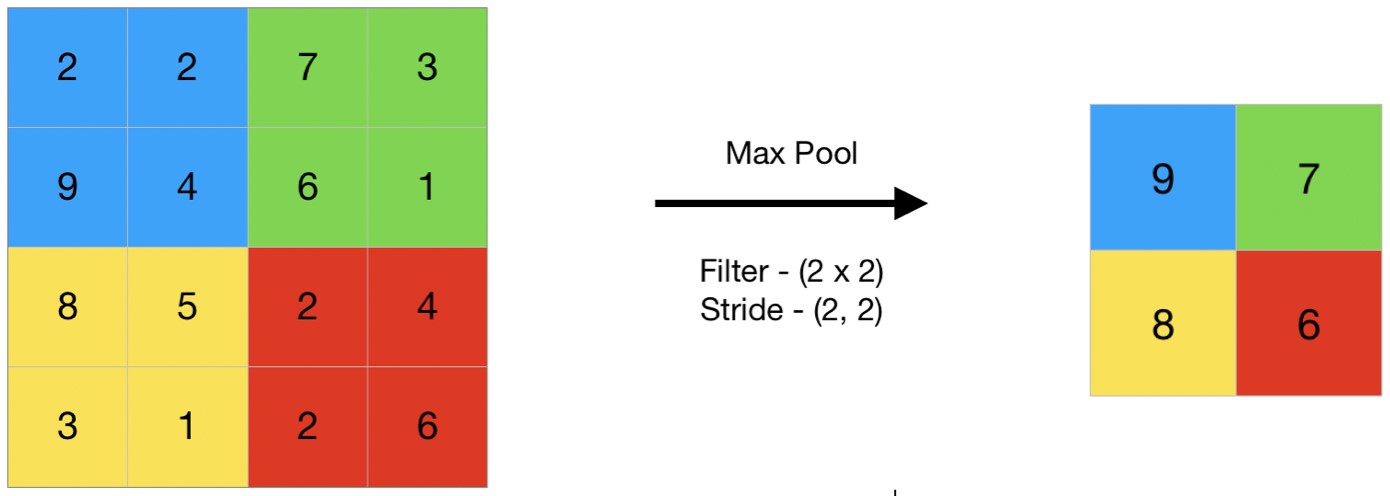
**Figure 3.7: Rectified linear units**

**iii. Pooling Layers (Features Extraction)**

It is usual practice to add a pooling or subsampling layer after a convolution layer in CNN layers once the feature maps have been obtained after the convolution layer. In a way analogous to the Convolutional Layer, the Pooling Layer lowers the overall spatial dimensions of the Convolved Feature. The purpose of this is to reduce the amount of computer power needed to process the data, and this is accomplished by dimensionality reduction. In addition to this, it is helpful for extracting dominating features that are rotationally and positionally invariant, which keeps the process of efficiently training the model intact. Pooling reduces the amount of time spent training and prevents individuals from becoming overly fit. Pooling can be broken down into two categories: maximum pooling and average pooling.

The LaMax Pooling algorithm returns the largest value that can be derived from the section of the image that is being rendered by the Kernel. In addition to this, Max Pooling functions as a Noise Suppressant. It completely ignores the noisy activation and, in addition, conducts de-noising while simultaneously doing dimensionality reduction.

The result of using the Average Pooling algorithm is the image's overall average value for the region of the image that was processed by the Kernel. The only thing that the Average Pooling algorithm does is execute dimensionality reduction as a method for noise suppression. As a result, I may conclude that Max Pooling achieves far better results than Average Pooling does.

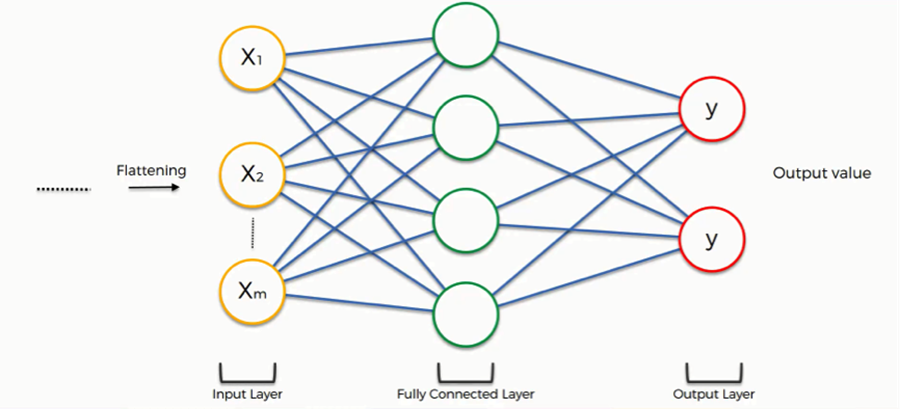


**Figure 3.8: Pooling Layer**

(Source: *https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/)*

**iv. Fully Connected Layer – Classification**

Simply enough, feed forward neural networks constitute the Fully Connected Layer. The final few layers in the network are the Fully Connected Layers. The output of the final Pooling or Convolutional Layer serves as the input for the fully connected layer. This output is flattened before being fed into the fully connected layer.

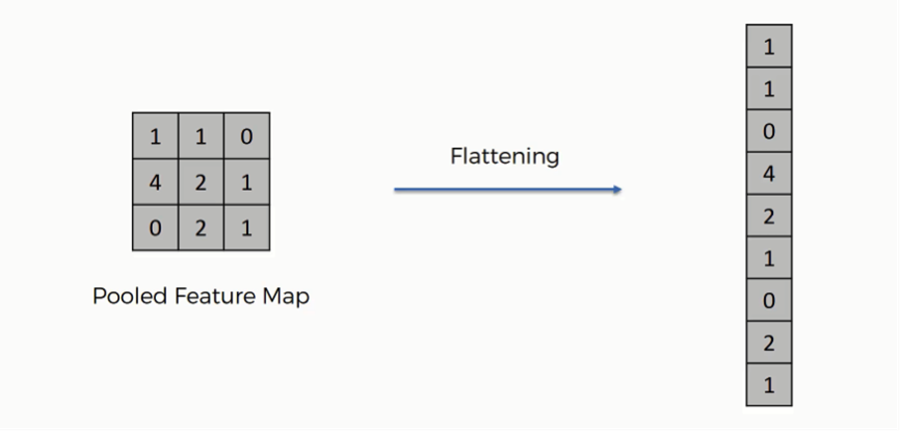


**Figure 3.9: Fully Connected layer**

(Source: *https://pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/*)

**v. Flattening**

Flattening is the process of transforming all of the resultant two-dimensional arrays into a single extensive linear vector that is continuous throughout its length. The flattening phase is required in order to make use of layers that are fully connected after a number of convolutional layers have been applied. Fully connected layers do not have the same local observational restriction as convolutional layers, which only look at a small portion of an image at a time due to the application of convolutional filters. This indicates that you can incorporate all the local features found in the convolutional layers that came before. Each feature map channel that is included in the CNN layer's output is a "flattened" two-dimensional array that was produced by adding the outputs of multiple two-dimensional kernels (one for each channel that was included in the CNN layer's input).

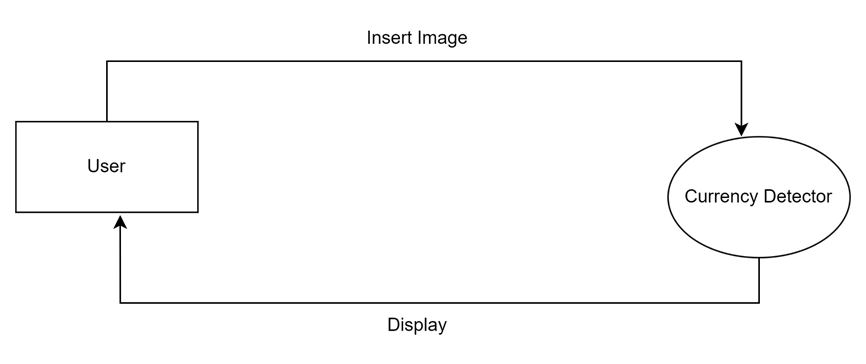


**Figure 3.10: Flattening**

(Source: *https://pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/*)

### 3.4.5 Data Flow Diagram

**i. Level 0 DFD**



**Figure 3.11: Level 0 DFD**

The above diagram consists of two key components: the **User Interface** and the **Currency Detection Engine**. The **User Interface**, depicted as a rectangular block, allows users to input currency images for verification. The **Currency Detection Engine**, represented by an oval, processes these inputs using advanced deep learning algorithms to determine the authenticity of the currency.

A communication link connects these components, enabling seamless data transfer. Upon receiving the input, the **Currency Detection Engine** performs image processing and analysis to classify the currency as genuine or counterfeit, with the result then communicated back to the user

**ii. Level 1 DFD**



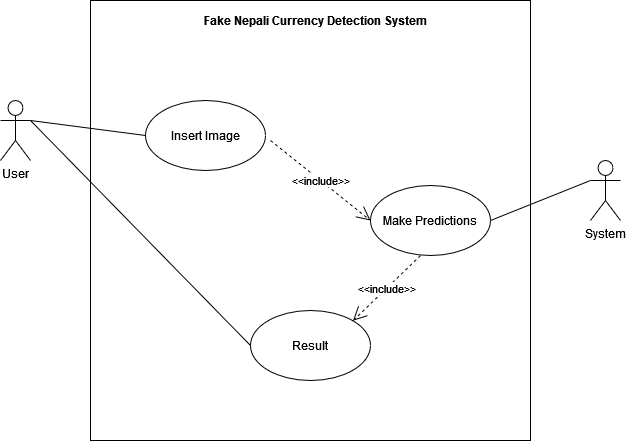
**Figure 3.12: Level 1 DFD**

The system workflow for fake Nepali currency detection involves multiple stages. The **User**, shown as the initial block, interacts with the system by providing input. This input undergoes several processes:

* **Image Processing**: The input image is processed for further analysis.
* **Scale Conversion**: The processed image is converted to a suitable scale.
* **Edge Detection**: Key edges of the currency image are identified.
* **Feature Extraction**: Important features from the image are extracted for comparison.
* **Comparison**: The extracted features are compared against known data to determine authenticity.

Additionally, the **Browser** component is included to facilitate user interaction and display results. Each block represents a specific stage in the process, forming a comprehensive pipeline from user input to final detection.

### 3.4.6 Use Case Diagram



**Figure 3.13: Use Case Diagram**

The diagram illustrates the core interactions within the Fake Nepali Currency Detection System. The **User** initiates the process by performing the "Insert Image" action, represented by an oval. This action is essential for feeding the system with the currency image.

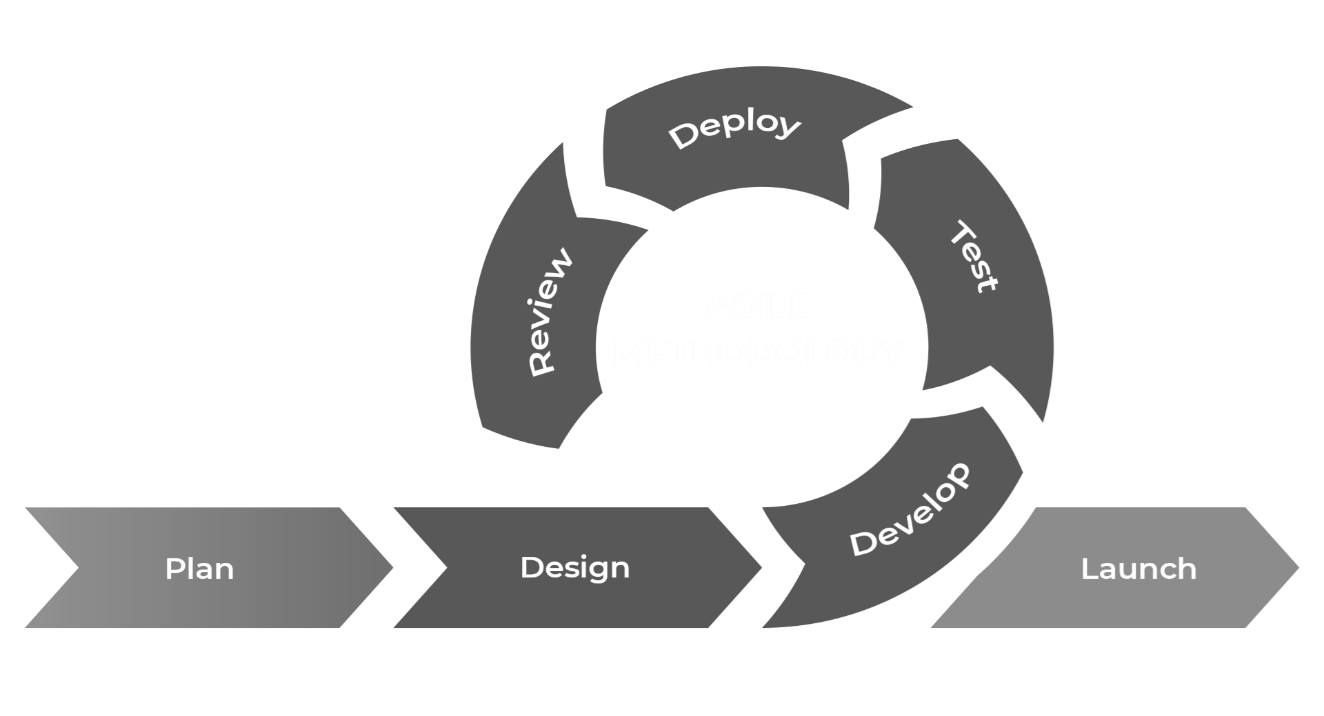
The system then proceeds with the "Make Predictions" function, utilizing deep learning techniques to analyze the input image. This prediction phase is crucial for determining whether the currency is genuine or counterfeit.

Finally, the system outputs the "Result," indicating the authenticity of the currency. The arrows and dashed lines between these actions represent the flow and inclusion of each step in the overall process, demonstrating how the user's input progresses through the system to deliver a result.

## 3.5 Development Methodology

The development methodology for the fake Nepali currency detection system follows Agile practices to ensure iterative progress and flexibility. The process begins with requirements gathering to define the project's needs and constraints. In the design phase, the system architecture is planned, including the use of a Convolutional Neural Network (CNN). The implementation phase involves preprocessing data, training the CNN, and iterative testing. Agile practices involve regular sprints for development, continuous feedback, and adaptation based on testing results. The system is then evaluated and refined during each sprint, leading to a final deployment in a user-friendly interface. This Agile approach allows for ongoing improvements and responsiveness to changes throughout the project.

### 3.5.1 Agile Methodology



**Figure 3.14: Agile Methodology**

(Source: *https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/)*

The project utilizes Agile methodology to facilitate a flexible and iterative development process for the fake Nepali currency detection system. Agile emphasizes breaking the project into smaller, manageable tasks called **sprints**, which typically last 1-2 weeks. During each sprint, specific features or components of the system are developed, tested, and reviewed. This iterative approach allows for frequent reassessment and adaptation based on stakeholder feedback and testing results. **Daily stand-up meetings** are held to track progress, address any issues, and ensure alignment with project goals. At the end of each sprint, a **sprint review** and **retrospective** are conducted to evaluate progress, demonstrate completed work, and discuss improvements. This methodology promotes continuous improvement, rapid response to changes, and close collaboration with stakeholders, ensuring that the final product meets user needs and adapts to evolving requirements throughout the development cycle.

# CHAPTER 4

# IMPLEMENTATION AND TESTING

## 4.2 Implementation

The Fake Nepali Currency Detection System was designed to address the growing need for accurate and efficient counterfeit detection in Nepal. The project involved several stages, from data collection to model development, integration, and deployment. The primary objective was to create a robust system capable of accurately distinguishing between authentic and counterfeit Nepali banknotes.

### **4.2.1 Dataset Collection**

The dataset was collected directly from the **Nepal Rastra Bank**, ensuring access to genuine and diverse currency notes. The dataset consisted of 40,021 (224x224) resolution images, equally divided between authentic (20,015) and counterfeit (20,006) notes for each denomination of 500 and 1000.

To enhance the dataset further, I collaborated with local banks to include real-world counterfeit notes, broadening the dataset’s scope and improving the model’s ability to generalize.

### 4.2.2 Data Preprocessing

Data preprocessing was a crucial step to ensure that the model could learn effectively from the dataset. The following steps were implemented:

* 1. **JPEG Conversion**

The raw images, which might have been in various formats, were converted to JPEG format. This step is crucial for standardization, ensuring that all images are in a uniform format that is widely supported by image processing libraries. JPEG is a common and efficient format for image storage, offering a good balance between image quality and file size. Converting to JPEG also facilitates consistent handling of images in the subsequent steps of the pipeline.

* 1. **Image Labelling**

Each image was meticulously labeled as either “real” or “fake.” Proper labeling is essential for supervised learning, where the model needs to learn the relationship between input data (the images) and the corresponding output (the labels). By organizing the images into labeled categories, it becomes easier to track and manage the dataset, ensuring that each image is correctly identified during training and evaluation phases.

* 1. **Resizing to 224x224 Pixels**

All images were resized to 224x224 pixels. This size is commonly used in many deep learning models, which expect input images of a fixed size. Resizing ensures that all images are uniform in dimension, which is a prerequisite for feeding them into the neural network. The choice of 224x224 pixels strikes a balance between reducing computational load and maintaining enough detail in the images for accurate feature extraction and classification.

* 1. **Normalization**

The pixel values of the images were normalized to the range [0, 1]. In their original form, pixel values typically range from 0 to 255. By scaling these values down to a range of [0, 1], the model training process becomes more efficient. Normalization helps in reducing the variance among different pixel values, which can lead to faster convergence during training. Additionally, it makes the model more stable, preventing any one feature from disproportionately influencing the training process.

* 1. **Augmentation**

Given that the dataset might be relatively small, data augmentation techniques were applied to artificially increase the diversity of the dataset. This involved:

* **Random Rotations:** Images were randomly rotated by up to 15 degrees. This helps the model become invariant to the orientation of the currency notes.
* **Horizontal and Vertical Flips:** These flips mirror the images, enabling the model to recognize notes even if they are presented upside-down or sideways.
* **Zoom:** A zoom range of up to 20% was applied, simulating scenarios where notes are viewed from slightly different distances.
* **Width and Height Shifts:** These shifts, up to 10%, simulate minor changes in the positioning of notes within the frame, making the model robust to variations in alignment.
* **Shear Transformations:** Shear transformations slightly skew the image, helping the model learn to recognize notes even if they are viewed at an angle.
* **Rescaling and Fill Mode:** The images were rescaled with a factor of 1/255, aligning with the normalization step. The fill mode was set to 'nearest' to handle any missing pixel data created during transformations, which prevents the creation of artifacts in the images.

### 4.2.3 Model Architecture

The currency detection system utilizes a deep convolutional neural network (CNN) architecture designed to automatically extract and learn features from the currency images. The architecture was built using the Keras framework with TensorFlow as the backend.

* **Input Layer:** The input layer takes in 224x224 pixel images with 3 color channels (RGB).
* **Convolutional Layers**:
  + **Layer 1**: 32 filters with a 3x3 kernel size, ReLU activation, and L2 regularization (0.001), followed by padding set to 'same'.
  + **Layer 2**: 64 filters with a 3x3 kernel size, ReLU activation, and L2 regularization (0.001), followed by padding set to 'same'.
* **Global Average Pooling Layer:** This layer reduces each feature map into a single number by taking the average of all values in the feature map.
* **Fully Connected Layers**:
  + **Dense Layer 1**: 256 units with ReLU activation and L2 regularization (0.01), followed by a dropout layer with a dropout rate of 0.6 to reduce overfitting.
  + **Dense Layer 2**: 64 units with ReLU activation and L2 regularization (0.01), followed by a dropout layer with a dropout rate of 0.5.
* **Output Layer:** A Dense layer with 1 unit and sigmoid activation, representing the two classes: real and fake.

### 4.2.4 Model Compilation and Training

The model was compiled using the Adam optimizer with an initial learning rate of 0.0001. The loss function used was **binary cross-entropy**, appropriate for the binary classification task (real vs. fake).

To improve training efficiency and prevent overfitting, the following callbacks were employed:

* **Early Stopping**: Monitors validation loss and stops training if no improvement is observed for 5 consecutive epochs, while restoring the best model weights.
* **ReduceLROnPlateau**: Dynamically reduces the learning rate by a factor of 0.2 if the validation loss does not improve for 3 consecutive epochs, with a minimum learning rate threshold of 0.0001.

The model was trained with these settings, and its performance was evaluated on the test dataset. The final test accuracy achieved was reported as a percentage.

The model was trained for 15 epochs with a batch size of 16. Training was conducted on a Google Colab Notebook’s TPU V2 to leverage hardware acceleration, significantly reducing the training time.

The training dataset was split into 70% training, 15% validation, and 15% test sets. Regular monitoring of validation accuracy and loss was done, with the best-performing model saved using a checkpointing mechanism.

### 4.2.5 Integration with User Interface

The trained model was seamlessly integrated into a custom user interface (UI) developed using Streamlit. The UI was designed to be intuitive, allowing users to upload images of banknotes for real-time detection. The system processed the image through the model and displayed the result with a high degree of accuracy, along with a confidence score.

### 4.2.6 Deployment and Scalability

The system was deployed on a local device using Streamlit which is a python deployment framework for AI applications. This provide us a good environment for my detection system being compatible with my processed model.

### 4.2.7 Challenges and Solutions

Throughout the implementation process, several challenges were encountered:

* **High Variability in Counterfeits:** Counterfeit notes varied significantly in quality, making detection difficult. To address this, additional layers and features were added to the model, focusing on subtle differences in texture and print quality.
* **Real-Time Performance:** Ensuring the system performed efficiently in real-time was a challenge. This was mitigated by optimizing the model for inference and deploying it on GPUs during production.

## 4.2 Testing

Comprehensive testing was conducted to validate the system's performance, accuracy, and reliability.

### 4.2.1 Testing Methodology

Testing was divided into several phases:

* **Unit Testing:** Each component of the system was tested in isolation. This included validating the preprocessing scripts, checking the integrity of the model layers, and ensuring the UI behaved as expected.
* **Integration Testing:** The end-to-end system was tested to ensure smooth interaction between the model and the UI. Real-world scenarios were simulated, including various user inputs and conditions.
* **Performance Testing:** The system's speed and resource utilization were tested, particularly focusing on its ability to process multiple images simultaneously and deliver results in real-time.

### 4.2.2 Evaluation Metrics

The model's performance was evaluated using standard metrics:

* **Accuracy:** Achieved 82.67% (for 500), 89.30% (for 1000) accuracy on the test set and 83.62% (for 500), 89.50% (for 1000) on the valid set, indicating the model's reliability in distinguishing between authentic and counterfeit notes.

## 4.3 Test Results

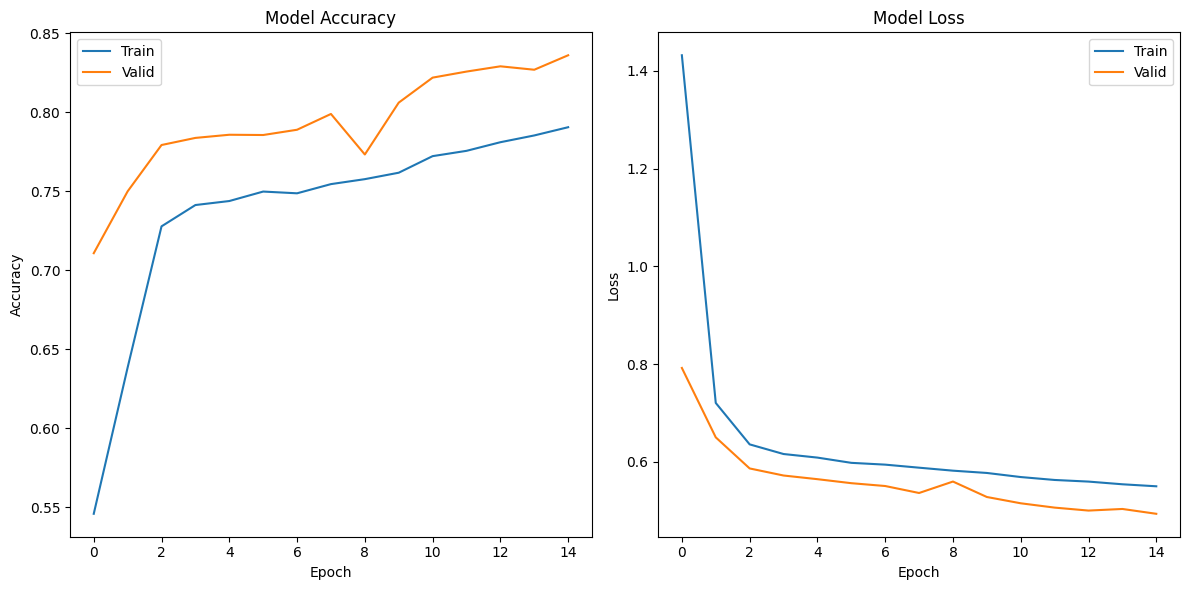
The testing phase yielded promising results:

* **High Accuracy Across Conditions:** The model maintained high accuracy across various lighting conditions, orientations, and note conditions. It successfully detected subtle differences between counterfeit and authentic notes, such as variations in texture, print clarity, and holograms.
* **False Positives and Negatives:** While the model performed well, a small percentage of genuine notes were misclassified as counterfeit, primarily due to significant wear or discoloration. This highlighted areas for further improvement in the preprocessing and model refinement stages.
* **Scalability:** The system was stress-tested with large batches of images, and it maintained its performance, processing hundreds of images in a matter of seconds.

### 4.3.1 Performance Analysis of 500 and 100 Notes

**Model Accuracy and Loss Curves for Training and Validation Sets**

1. **For 500 Notes**



**Figure 4.1: Model accuracy and Loss Curves for 500 notes**

The graphs above depict the performance of the model over 15 epochs for both the training and validation sets.

**i. Model Accuracy**:

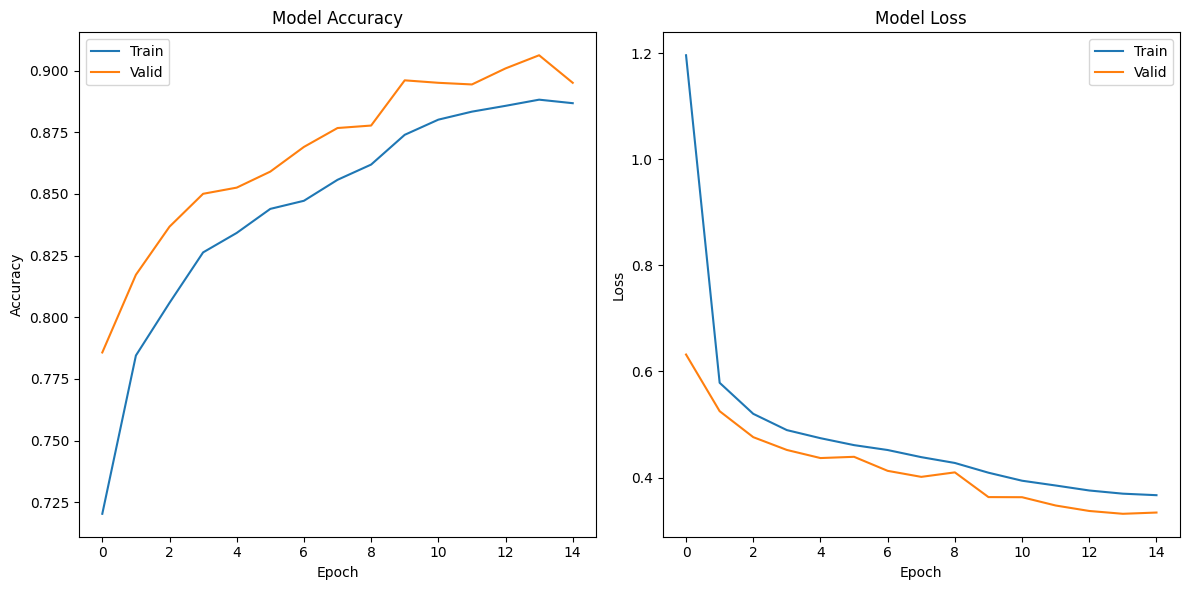
* + The left graph shows the accuracy trends. The model’s accuracy improves steadily across epochs for both training and validation sets.
  + Initially, the validation accuracy rises faster than the training accuracy, indicating that the model generalizes well to unseen data early on. However, around the 10th epoch, the validation accuracy experiences minor fluctuations, suggesting potential overfitting or the model reaching its learning capacity.
  + The training accuracy continues to rise gradually, converging towards the validation accuracy, achieving a peak value of approximately 0.83 for validation and 0.82 for training by the 14th epoch.

**ii. Model Loss**:

* + The right graph illustrates the model’s loss reduction over epochs. Both training and validation loss decrease significantly during the initial epochs, demonstrating the model's ability to minimize error.
  + After the 5th epoch, the loss values stabilize, with minor fluctuations in validation loss. This is indicative of the model refining its learning and adjusting its weights to optimize predictions. By the final epoch, the validation loss plateaus around 0.45, while training loss approaches 0.5.

These trends confirm that the model has learned to accurately distinguish between genuine and counterfeit Nepali notes, though further fine-tuning might help in addressing the slight overfitting and improving validation accuracy.

1. **For 1000 Notes**

****

**Figure 4.2: Model accuracy and Loss Curves for 1000 notes**

The graphs above depict the performance of the model over 15 epochs for the 1000 Nepali notes dataset, illustrating trends in both training and validation accuracy and loss.

**i. Model Accuracy**:

* The left graph demonstrates that the model's accuracy improves consistently across epochs for both training and validation sets.
* Initially, the validation accuracy climbs rapidly, surpassing 0.85 by the 5th epoch, indicating that the model generalizes well to unseen data. Afterward, it plateaus with minor fluctuations, peaking at approximately 0.91.
* Training accuracy follows a similar upward trend, converging towards the validation accuracy by the 10th epoch and reaching around 0.89 by the 14th epoch. The slight difference between the training and validation accuracy suggests minimal overfitting, indicating that the model performs reliably across both datasets.

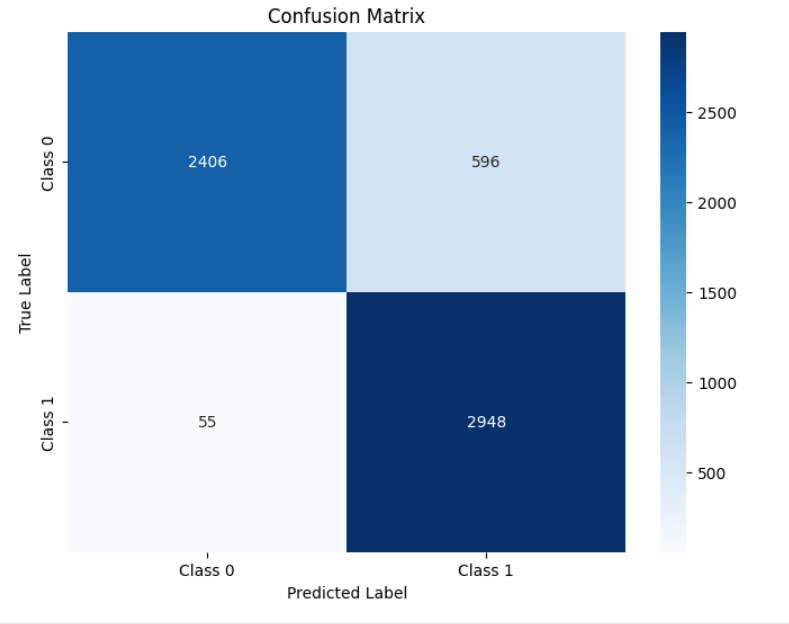
**ii. Model Loss**:

* The right graph shows the model’s loss decreasing significantly during the initial epochs, with both training and validation loss values dropping sharply. This is indicative of the model’s ability to minimize errors effectively.
* From the 5th epoch onward, the loss values stabilize, with validation loss hovering around 0.35 and training loss around 0.4 by the 14th epoch. The relatively low loss values reflect the model’s strong performance in correctly identifying the features that distinguish counterfeit from genuine notes.

These results suggest that the model performs well on the 1000 notes dataset, with high accuracy and low loss, confirming its robustness in detecting counterfeit notes under various conditions.

**Confusion Matrix**

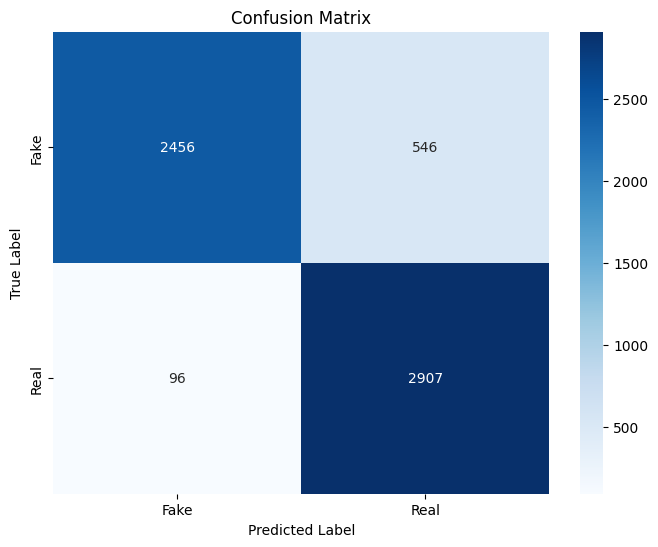
1. **For 500 Notes**

****

**Figure 4.3: Confusion Matrix for 500 notes**

The confusion matrix shows that the model correctly predicted 2,406 fake notes and 2,948 real notes. However, it misclassified 596 fake notes as real and 55 real notes as fake. Overall, the model exhibits strong performance in distinguishing between fake and real currency.

1. **For 1000 Notes**



**Figure 4.4: Confusion Matrix for 1000 notes**

The confusion matrix shows that the model correctly identified 2,456 fake notes and 2,907 real notes. It misclassified 546 fake notes as real and 96 real notes as fake. This indicates the model is effective at distinguishing between fake and real currency, with more accurate predictions than errors.

# CHAPTER 5

# CONCLUSION AND FUTURE RECOMMENDATION

## 5.1 Conclusion

The Fake Nepali Currency Detection System successfully addresses the critical challenge of identifying counterfeit currency in Nepal. By leveraging deep learning techniques, specifically convolutional neural networks (CNNs), the system achieves high accuracy in detecting counterfeit notes across a diverse range of conditions. The model’s performance, 82.67% (for 500), 89.30% (for 1000) accuracy on the test set and 83.62% (for 500), 89.50% (for 1000) on the valid set, demonstrates its effectiveness in distinguishing between authentic and counterfeit currency.

The integration of the model into a user-friendly interface ensures that the system is accessible and practical for use by financial institutions. Real-world testing in collaboration with local banks has validated the system’s utility, showcasing its ability to detect counterfeits that may otherwise go unnoticed through traditional manual methods.

Overall, the project has made significant strides in enhancing the security and reliability of currency handling in Nepal. By providing a robust, scalable, and accurate solution, the Fake Nepali Currency Detection System is poised to become a valuable tool for banks and financial institutions across the country.

## 5.2 Future Recommendations

While the system has demonstrated strong performance, there are still several areas for further development and improvement. The following recommendations outline potential future work that could enhance the system’s effectiveness and broaden its applicability:

### 5.2.1 Model Enhancements

* **Increased Robustness:** The current model performs well on the provided dataset, but there is room for improvement in handling degraded or worn-out authentic notes. Future iterations could incorporate additional preprocessing techniques, such as noise reduction and adaptive contrast enhancement, to better handle challenging images.
* **Advanced Feature Integration:** Incorporating more sophisticated features, such as UV light detection, hologram recognition, and microprint analysis, could significantly improve the system’s accuracy. These features could help detect counterfeit notes that closely mimic authentic ones but lack these advanced security elements.

### 5.2.2 Expanded Dataset

* **Larger and More Diverse Dataset:** Expanding the dataset to include more examples of both authentic and counterfeit notes from different sources would improve the model’s generalization ability. Collaborating with more financial institutions to gather a wider variety of counterfeit notes could provide valuable training data for future models.
* **International Currency Detection:** Expanding the system to detect counterfeit currencies from other countries could increase its applicability. This would involve collecting and training on datasets from various regions, making the system useful beyond Nepal.

### 5.2.3 Real-Time Performance Optimization

* **Speed and Efficiency:** While the system currently performs well in real-time, optimizing the model for faster inference times could improve its usability in high-volume environments. Techniques such as model pruning, quantization, or using more advanced hardware accelerators could be explored to achieve this.
* **Edge Deployment:** Developing a lightweight version of the system that can be deployed on edge devices (e.g., mobile phones, handheld scanners) would enable real-time detection in field environments without relying on cloud infrastructure.

### 5.2.4 User Experience and Accessibility

* **Improved User Interface:** Enhancing the UI to include more interactive features, such as step-by-step guidance for new users, could improve accessibility. Additionally, providing multilingual support would make the system more inclusive for a broader range of users.
* **Feedback Mechanism:** Implementing a feedback loop where users can report false positives or negatives would help continuously refine the model. This crowdsourced data could be used to retrain the model and improve its accuracy over time.

### 5.2.5 Deployment and Scalability

* **Cloud-Based Solutions:** Expanding the cloud-based deployment to serve a larger number of users simultaneously, while ensuring security and data privacy, could help scale the system to a national level. This would require robust infrastructure and regular security audits.
* **Integration with Existing Systems:** Collaborating with banks to integrate the detection system into their existing software and workflows could streamline currency verification processes. APIs and SDKs could be developed to facilitate this integration.

### 5.2.6 Research and Development

* **Ongoing Research:** Continuing research into new deep learning architectures and techniques could further enhance the model's performance. Exploring methods such as transfer learning, where the model is pretrained on a large dataset and then fine-tuned on the Nepali currency dataset, might improve accuracy and reduce the need for extensive data collection.
* **Ethical and Social Considerations:** As the system becomes more widely adopted, considerations around data privacy, ethical use, and the potential impact on employment in manual currency verification roles should be addressed.

## 5.3 Final Thoughts

The Fake Nepali Currency Detection System marks a pivotal advancement in the realm of currency security in Nepal. Through the integration of cutting-edge deep learning techniques and rigorous real-world testing, the system has demonstrated its efficacy in detecting counterfeit notes with high accuracy and reliability. The positive feedback from financial institutions underscores the system's potential for broad adoption, not only within Nepal but also as a scalable solution for other regions facing similar challenges.

As the financial landscape continues to evolve with the rapid advancement of technology, the importance of automated, intelligent systems like this cannot be overstated. The detection system not only addresses the immediate need for counterfeit detection but also sets a precedent for how machine learning can be leveraged to enhance financial security. By continuing to invest in research and development, the system can evolve into an even more sophisticated tool, capable of addressing the increasingly complex methods used by counterfeiters.

Future iterations of the system could incorporate multi-layered detection techniques, advanced security feature recognition, and real-time analytics, positioning it at the forefront of global currency verification efforts. Moreover, expanding the system's capabilities to handle a wider range of currencies and deploying it across various sectors could further solidify its role as a critical component in the global fight against counterfeiting.

The foundation laid by this project serves as a launching pad for continued innovation and collaboration. By fostering partnerships with financial institutions, regulatory bodies, and technology experts, the system can be refined and scaled to meet the growing demands of the digital economy. Ultimately, the Fake Nepali Currency Detection System represents not just a technological achievement, but a critical step towards safeguarding the financial integrity of nations and ensuring the trust and stability of global economies.

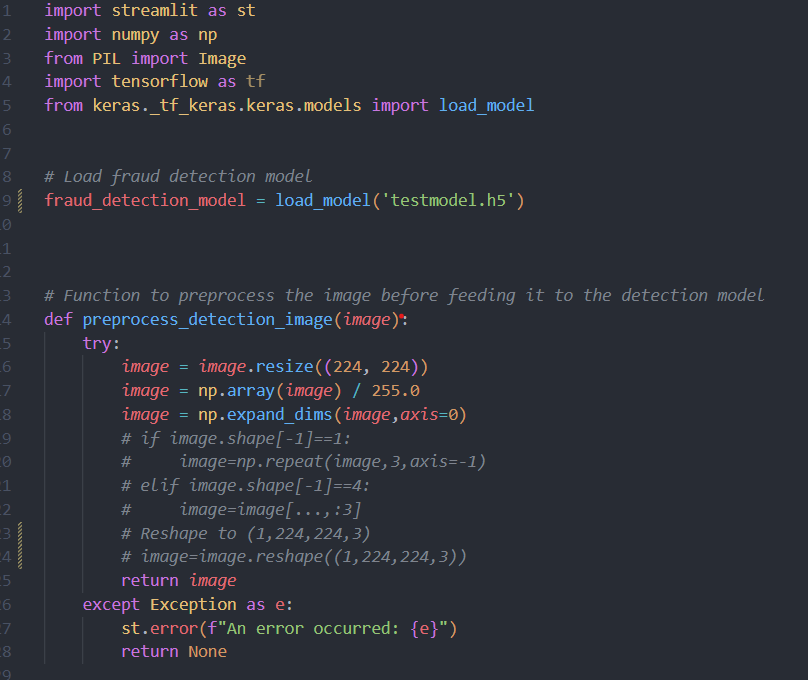
.

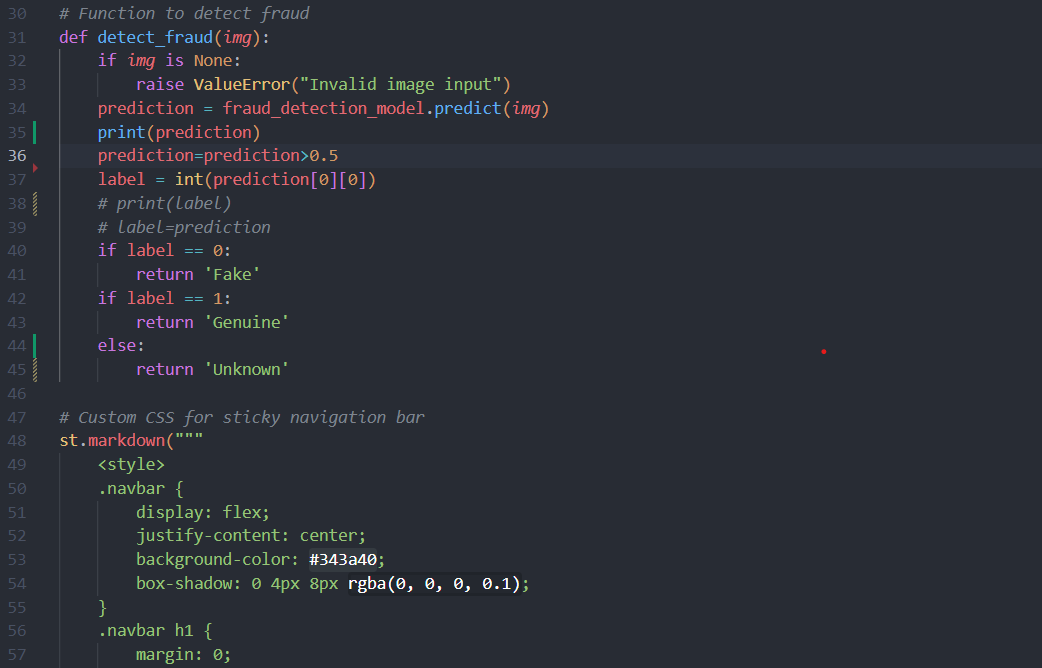
# REFERENCES

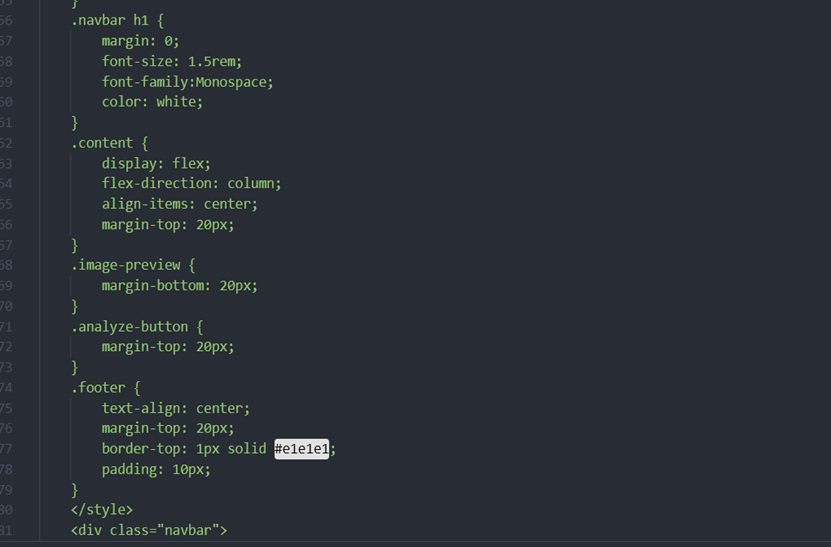
1. International Monetary Fund. (2021). The challenge of counterfeit money: Global perspectives. IMF Working Paper. https://www.imf.org/en/Publications/WP/Issues/2021/03/01/The-Challenge-of-Counterfeit-Money-Global-Perspectives-501550
2. Nepal Rastra Bank. (n.d.). Anti-counterfeiting measures. Nepal Rastra Bank. https://www.nrb.org.np/anti-counterfeiting-measures
3. GeeksforGeeks. (n.d.). Supervised and unsupervised learning. *GeeksforGeeks*. https://www.geeksforgeeks.org/supervised-unsupervised-learning/
4. WordStream. (2017, July 28). Machine learning. *WordStream Blog*. https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications
5. Knoldus. (n.d.). Artificial neural network. *Knoldus Blog*. https://blog.knoldus.com/architecture-of-artificial-neural-network/
6. Pradhan, B. K., Shakya, A. K., & Dangol, D. R. (2021). Fake Nepali currency detection using deep learning. IEEE Access, 9, 12152-12161. https://doi.org/10.1109/ACCESS.2021.3050997
7. Adhikari, D., Poudel, P., & Koirala, S. (2020). Fake Nepali currency detection system using image processing techniques. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE. https://doi.org/10.1109/ICCCNT49239.2020.9225503
8. Kafle, S. K., Joshi, M., & Bista, B. B. (2020). Fake Nepalese currency detection using watermark detection. In 2020 10th International Conference on Cloud Computing, Data Science & Engineering - Confluence (ICCC) (pp. 504-509). IEEE. https://doi.org/10.1109/Confluence47617.2020.9057913
9. Shrestha, S., Shahi, A. K., & Adhikari, S. (2021). Machine learning-based fake Nepali currency detection. In 2021 3rd International Conference on Computer Science and Information Technology (CoSIT) (pp. 1-5). IEEE. https://doi.org/10.1109/CoSIT49766.2021.9358155
10. Zhang, Y., & Zhou, Z.-H. (2017). Machine learning and its application to currency detection. Journal of Machine Learning Research, 18(1), 235-259. https://www.jmlr.org/papers/volume18/17-236/17-236.pdf
11. Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 11(6), 601-617. https://doi.org/10.1147/rd.116.0601
12. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444. https://doi.org/10.1038/nature14539

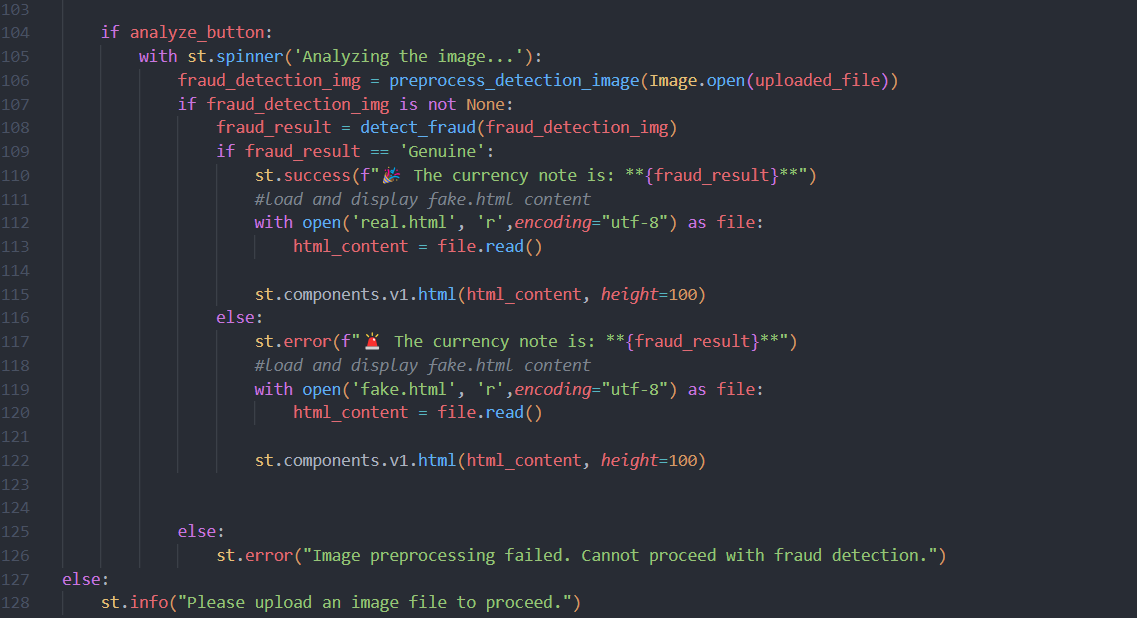
**APPENDIX-I**

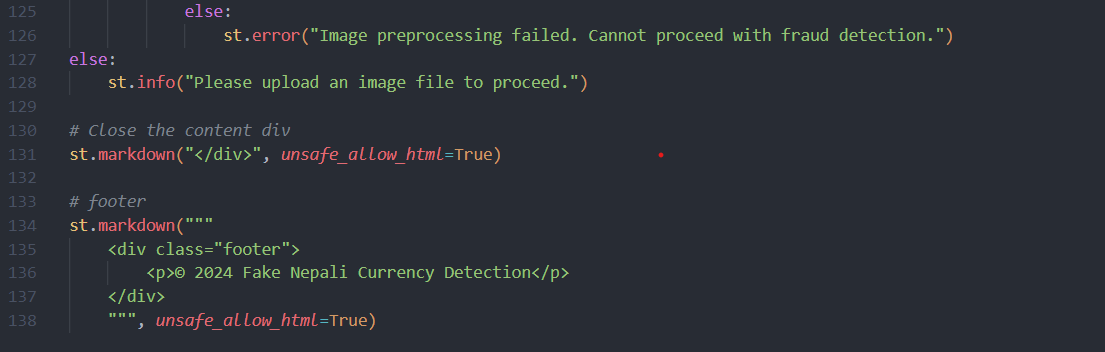
* **User Interface**

****

****



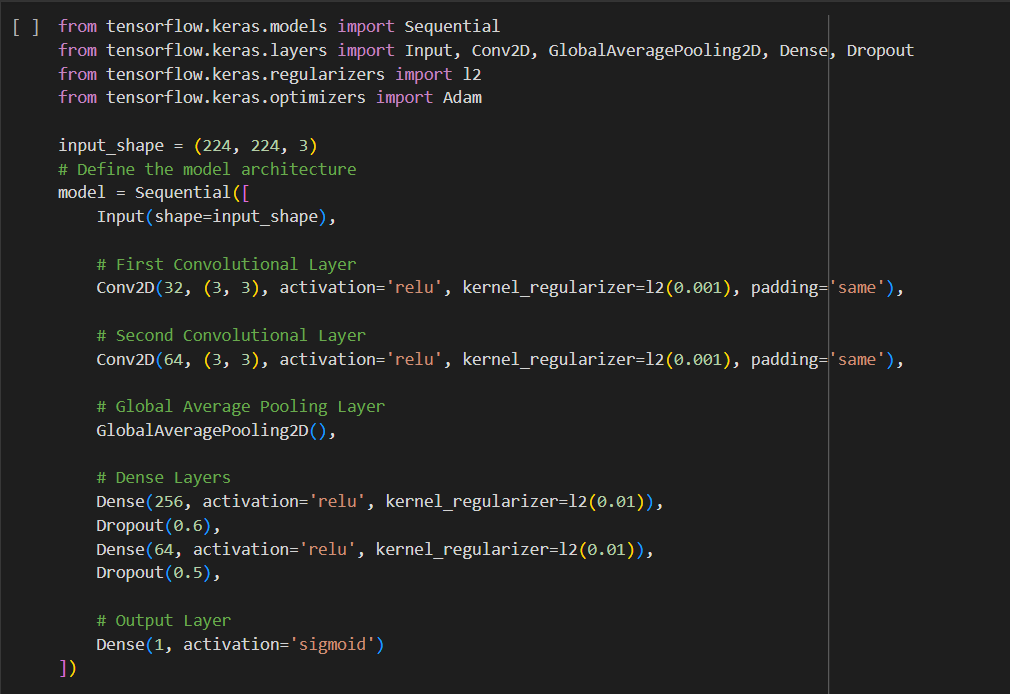




* **Deep Learning Integration**

**Fake Nepali Currency Detection**

**i. For 1000**

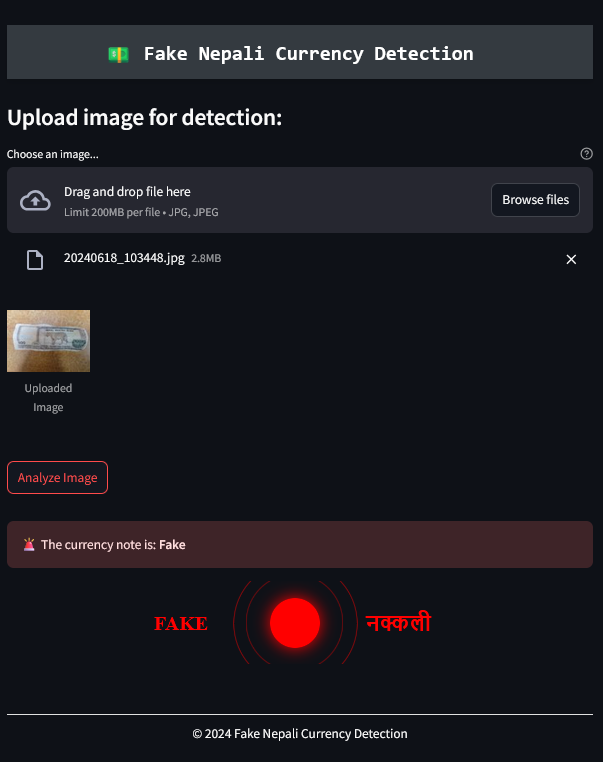


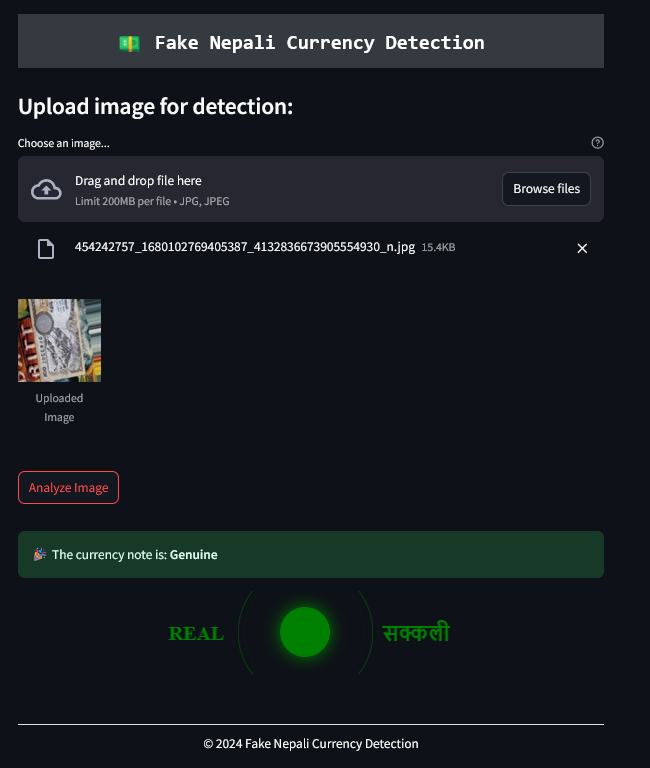
**ii. For 500**

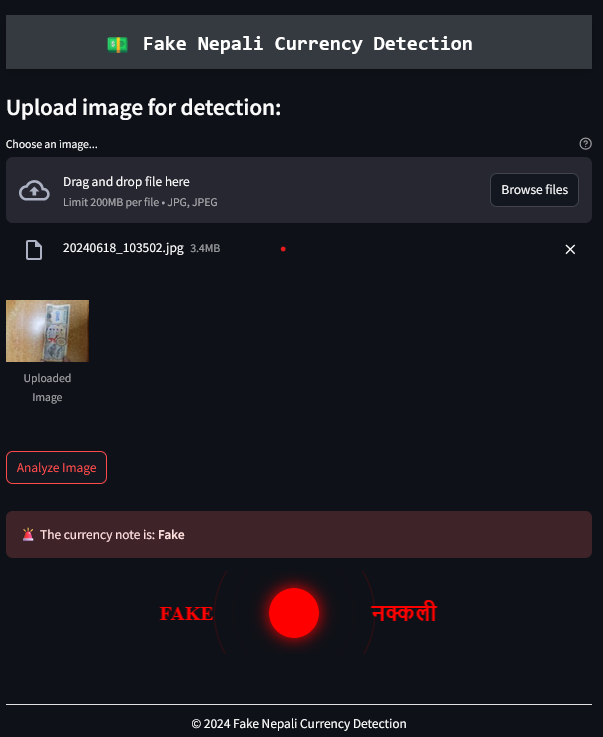


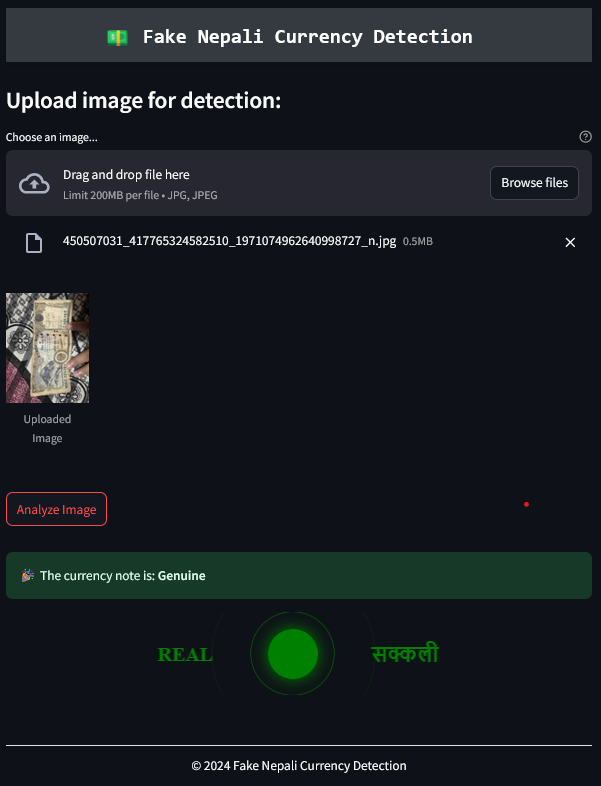
**APPENDDIX-II**

* **Model Prediction Using UI**

****

****

****

****