Quantitative Degradation Metrics through Visual Pattern Analysis for Currency Note Usability Assessment

Md Sultanul Islam Ovi
movi@gmu.edu

1 Introduction and Motivation

What Problem Are We Solving?

Automated systems like ATMs, vending machines, and bill validators often fail to process currency notes that are torn, wrinkled, or worn. Currently, identifying such unfit notes relies on manual inspection, which is time-consuming and inconsistent. This project aims to automate the assessment of a currency note's physical condition using computer vision, comparing it to a pristine reference to quantify its degradation.

Why Does It Matter?

Manual inspection leads to inconsistent decisions and operational delays. Damaged notes that go undetected can cause machine jams or transaction failures. Automating this process reduces human error and improves the reliability of cash-handling systems.

Impact and Applications

The system can help banks and businesses automatically detect and remove damaged notes, reducing machine malfunctions and improving the quality of currency in circulation. It enables faster, more consistent sorting of notes and supports efficient cash operations.

2 Literature Review

Recent studies underscore the growing need to automate banknote handling and reduce the risks posed by physically compromised currency. Oviedo *et al.* [1] introduced BankNote-Net, a large-scale repository designed primarily for multi-currency recognition in assistive contexts. While their dataset spans 17 currencies, it lacks labels pertaining to note condition or fitness. This gap reflects a common limitation among publicly released datasets, which frequently emphasize denomination identification over the detection of physical defects.

To address explicit damage scenarios, Meshram *et al.* [2] published a comprehensive set of Indian banknotes considered "unfit" due to tears, stains, or missing sections. Although the dataset omits fine-grained severity labels, it provides a unique window into real-world degradation modes. Subsequently, another work by the same group [3] examined how top-tier pretrained models perform on currency images, factoring in quality indicators. However, neither study supports a direct, quantitative scoring scheme for note usability; rather, they introduce data suited for classification or detection tasks.

In parallel, Jaman *et al.* [4] proposed a CNN-based pipeline for Bangladeshi currency detection, including a mobile application interface. Although their work enhances accessibility for visually impaired users, it predominantly focuses on denomination recognition rather than automated fitness assessment. A deeper exploration of banknote condition appears in Pham *et al.* [5], who leveraged one-dimensional reflection images and deep learning to classify note fitness. Their reflection-based approach illustrates how targeted sensing modalities can reveal subtle texture or surface changes indicative of wear.

From a broader perspective, Czimmermann *et al.* [6] surveyed visual defect detection methods in industrial applications, covering edge-based algorithms, learning-based classifiers, and convolutional networks. While not limited to currency, their examination of defect localization and severity estimation offers a conceptual framework for analyzing tears or stains on paper notes. Work by Farooq and Imran [7] similarly explored object detection

Table 1: Overview of Datasets Utilized in this Study

Dataset	Total Images	Classes	Image Size
Dataset 01	1,970	9	120×250
Dataset 02	10,000	10	224×224
Dataset 03	70,542	8	200×250
Combined Dataset	82,512	10	224×224

Figure 1: Sample Images from Each Class of the Dataset

in vending machines to improve currency handling, but again, the emphasis rested more on *recognition* than the intricacies of mechanical rejects caused by damaged bills.

In summary, the literature contains numerous strategies for currency *identification* and *recognition*, along with emerging datasets that highlight damaged notes. Yet there remains a notable shortfall in resources that systematically link a given note's physical defects to a standardized *usability score*. Most research either classifies bills as "fit" or "unfit" in a binary sense or focuses on generic recognition under varied lighting and angles. Consequently, our project aims to bridge this gap by developing *quantitative degradation metrics*—an approach that not only flags unfit notes but also ranks partial damage on a continuum, catering to real-world demands of automated systems in banking, vending, and cash validation.

3 Methodology

We divided the project into two parts: classification and damage detection. For classification, we created our dataset by combining three different sources of Bangladeshi currency note images. This helped us train a model to recognize the type of note from a given image. For damage detection, we could not find any suitable public dataset, so we collected sample images showing various types of damage. These images were used to design and test our methods for identifying and measuring torn or worn areas on the notes.

3.1 Setup Instructions and Dependencies

All experiments were conducted using Python 3.9.9. We used PyTorch 2.4.1 with CUDA 12.1 as the core deep learning framework, and the code was executed on an NVIDIA A100-SXM4-80GB GPU with 40GB of memory. Supporting libraries such as NumPy, OpenCV, and Pillow were used for numerical and image processing tasks. The user interface was developed using Streamlit to support real-time testing and demonstration.

The project is organized into modular components including data preprocessing, model training, damage detection, and a Streamlit-based web interface. This structure allows easy extension and maintenance.

A 'requirements.txt' file is provided with the code to help replicate the environment. The full implementation is not publicly released yet, as we plan to extend this work into a research paper. However, all necessary files have been shared privately for evaluation.

3.2 Dataset Description

We used three publicly available datasets of Bangladeshi banknotes to build a single, unified dataset for training and evaluation. Table 1 summarizes the details of each dataset, and Figure 1 shows example images.

Dataset 1 [8] contains 1,970 mobile-captured images under real-world conditions but does not include the 200 Taka note. Dataset 2 [9] provides 10,000 clean images with full denomination coverage. Dataset 3 [10] offers over 70,000 images, including old and new note designs, though it excludes 1 and 200 Taka. We combined all three and standardized the resolution to 224×224, resulting in a final dataset of 82,512 images across 10 classes.

Algorithm 1: Image Deduplication Process

Input : Source dataset D, similarity threshold t
Output: Deduplicated dataset D'

$D' \leftarrow \emptyset$;
foreach *class* c *in* D **do**
  $H \leftarrow \emptyset$;
  foreach *image* i *in class* c **do**
    $h \leftarrow \text{CalculatePerceptualHash}(i)$;
    if *no similar hash in* H *within* t **then**
      $H[h] \leftarrow i$;
      $D' \leftarrow D' \cup \{i\}$;
    end
  end
end
return D'

3.3 Dataset Analysis

To ensure the dataset was clean and reliable, we performed deduplication using perceptual hashing with a Hamming distance threshold of 5. Figure 2 outlines the deduplication steps. The process first removed duplicates within each source, followed by cross-dataset deduplication. This was essential to avoid overlap between training, validation, and test sets. The impact on class distribution is shown in Figure 3.

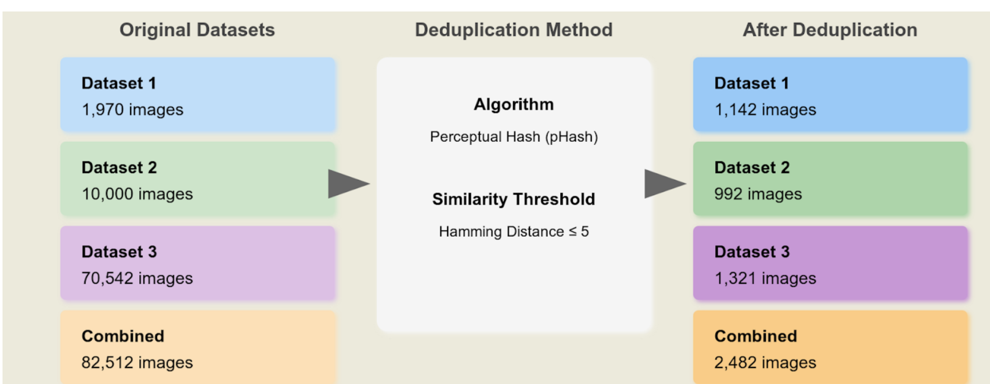

Figure 2: Overview of Dataset Deduplication Method

3.4 Data Preprocessing

Our preprocessing pipeline included three main steps: dataset splitting, image enhancement, and data augmentation.

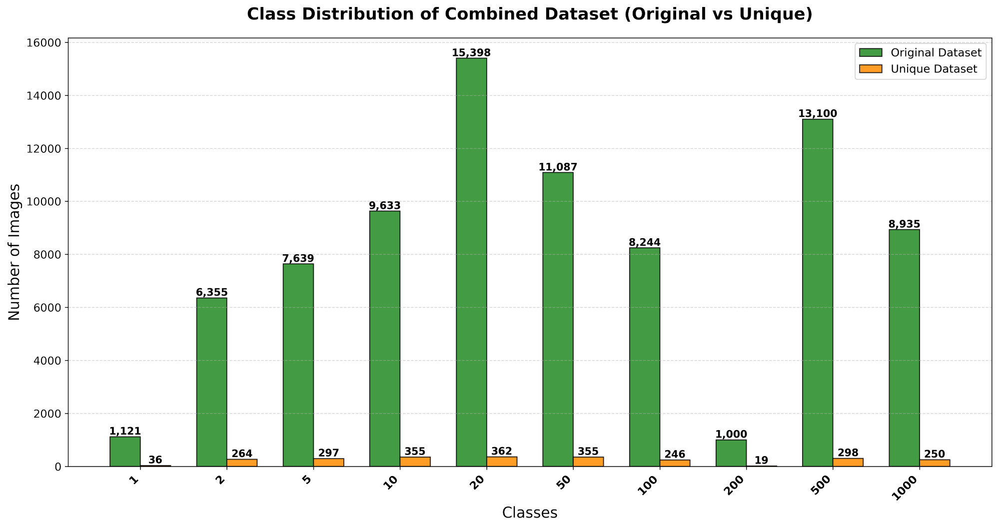

Figure 3: Class Distribution Analysis Before and After Deduplication

We first split the combined dataset into training (80%), validation (10%), and test (10%) sets, making sure the class distribution remained balanced.

To improve image quality, we applied several enhancement techniques:

1. **Median Blur:** Reduces noise while preserving edges. For a kernel size $k = 3$, each pixel value is replaced with the median of its $k \times k$ neighborhood:

$$I'(x, y) = \text{median}\{I(x + i, y + j)\}$$

2. **Basic Sharpening:** Enhances fine details using a convolution kernel:

$$K = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \quad I'(x, y) = \sum_{i,j} K(i, j) I(x - i, y - j)$$

3. **Contrast Stretching:** Adjusts brightness range based on percentiles:

$$I'_c(x, y) = 255 \times \frac{I_c(x, y) - p_2}{p_{98} - p_2}$$

where p_2 and p_{98} are the 2nd and 98th percentiles.

4. **CLAHE:** Improves local contrast while avoiding over-amplifying noise. Applied in LAB color space with a clip limit of 2.0:

$$I'(x, y) = \text{CDF}(I(x, y)) \cdot (I_{\max} - I_{\min}) + I_{\min}$$

We then applied data augmentation to create 10 variations per image to improve generalization. These included:

- Random resized cropping (scale: 0.8–1.0)
- Random rotation (±15°)
- Horizontal flipping
- Color jittering (brightness, contrast, saturation, hue)

- Affine transformations (translate, scale, shear)
- Random erasing (probability 0.5, scale 2–15%)

Finally, all images were resized to 224×224 pixels and normalized using ImageNet statistics to support transfer learning. This preprocessing pipeline improved both the visual consistency and robustness of the dataset, helping the model perform better across various input conditions.

3.5 Part 1: Currency Classification

The classification pipeline identifies Bangladeshi banknote denominations from input images. Figure 4 shows the full pipeline, from preprocessing to prediction.

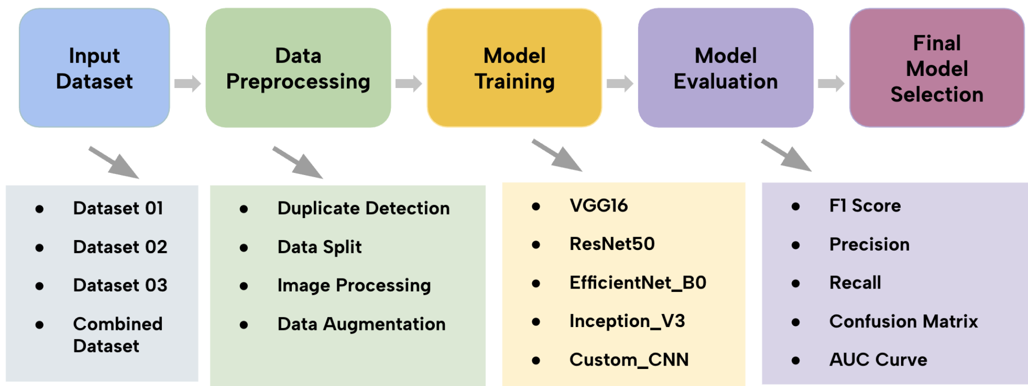

Figure 4: Classification Pipeline

Preprocessed images are fed into deep learning models trained under consistent settings. We used VGG16, ResNet50, EfficientNetB0, InceptionV3, and a custom lightweight CNN. Each model outputs a predicted class label representing the note's denomination. Performance was evaluated using the same dataset splits and standardized metrics.

3.5.1 Deep Learning Models

We evaluated four popular deep learning models for Bangladeshi banknote classification, each trained under the same conditions for fair comparison.

VGG16: VGG16 is a 16-layer CNN with stacked 3×3 convolutions. It has about 138 million parameters and is known for high accuracy, but it's heavy to run. We used ImageNet weights and fine-tuned the last layers for our task.

ResNet50: ResNet50 introduces skip connections that help train deeper networks without gradient issues. With 50 layers, it offers a good balance between depth and performance.

Inception V3: Inception V3 uses multiple filter sizes in each block to capture different levels of detail. Its structure is efficient and works well with images under different lighting and quality conditions.

EfficientNet B0: EfficientNet B0 uses a scaling method to balance depth, width, and resolution. It has only 5.3 million parameters but gives high accuracy, making it ideal for lightweight deployment.

3.5.2 Custom CNN Architecture

We built a custom lightweight CNN with three convolutional blocks. Each block includes a convolutional layer, batch normalization, ReLU activation, and max-pooling. The layers use 16, 32, and 64 filters, respectively. Two fully connected layers follow, and the final layer outputs class probabilities. The model uses around 287,000 parameters and processes 64×64 RGB images, providing fast and efficient inference.

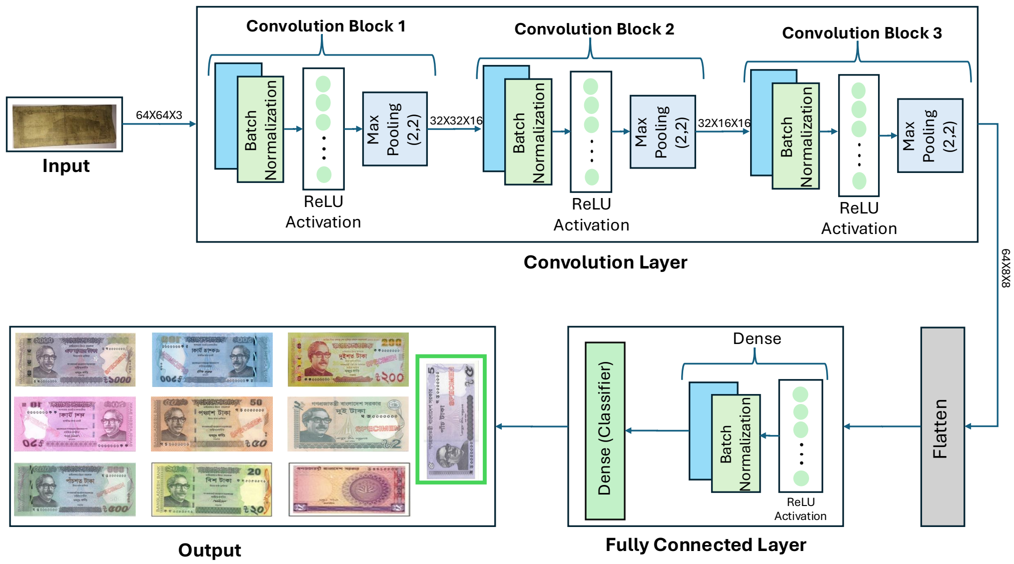

Figure 5: Custom CNN Architecture

3.5.3 Evaluation Metrics

We used several metrics to evaluate model performance:

- Accuracy, precision, recall, and F1-score for overall classification quality
- Confusion matrix to analyze misclassifications
- AUC-ROC curves to assess class separation
- Training and validation loss curves to monitor convergence

We also compared model sizes and number of parameters to assess how suitable each model is for real-time deployment.

3.6 Part 2: Damage-Based Usability Assessment

We developed a comprehensive damage detection framework that analyzes currency notes through multiple stages. We start by removing the background and extracting just the image using HSV color thresholding and morphological operations to clean up the mask. Next, a binary mask highlighting the non-white areas of the image is generated to isolate the note. We then align the 'damaged bank note' image with reference to a 'standard bank note' image using feature detection and matching. For this, we use the SIFT (Scale-Invariant Feature Transform) to detect key points and extract descriptors. We also use FLANN-based matches to match keypoints between images using the k-nearest neighbors approach. We use the Lowe's ratio test to retain only the good ones. We then use RANSAC for homography estimation to align the images based on the matched points. Finally, it warps the torn image to align with the standard image's perspective. This concludes the part of aligning the uploaded image with the reference image so that we can dive deeper into our analysis.

The image is then converted to grayscale, and high-value pixels (close to white) are thresholded to create a binary inverse mask of the note. Morphological operations are used to clean up noise and fill small gaps, refining the binary mask.

3.6.1 Dataset of Standard Notes and Damaged Notes

The dataset for all the front and back sides images of standard Bangladeshi notes has been collected from the official website of the Bangladesh Bank (National Bank of Bangladesh). These provide the most authentic source for these notes without any chance of error. The dataset for the damaged notes has been collected by us, by taking images of regularly used banknotes.

3.6.2 Torn note

A banknote can have several tears and damages at any location of the note. We use the binary mask of both notes to check for missing areas (e.g., torn parts). These areas are measured in terms of pixels by connecting regions that are close by. We then map them to general zones like top, middle, bottom and left, center, and right to describe the location of the damage.

3.6.3 Edge and corner damage

Most of the time, notes have damage along the edges and corners due to normal wear and tear. So, the image is divided into four edges and four corners, and the difference between the standard and torn binary mask is computed. This helps us detect if any edges or corners are damaged or not.

3.6.4 Checking significant features of a note

A banknote that has damage in significant areas are not acceptable in many cases. So, we look for them to make sure the note is still usable. We look for the integrity of different features present in a note, such as the image of a person, the note denomination, the name of the note, the name of the central bank, the signature, the serial number, and any other significant blobs that are present in any note. For this, the standard note is used as a reference. We detect the blobs in that note, and compare the detected blobs with the uploaded torn/damaged note. We use contour detection to isolate individual features. DBSCAN clustering is used to group nearby contours, as each individual letter on the note might be considered a separate blob otherwise. For each clustered feature, we extract a template and match it against the aligned torn image using normalized cross-correlation. It reports whether the feature is present or missing/partial based on the similarity score thresholding.

3.6.5 Color distortion

Next, we calculate the RGB differences and provide a normalized score to calculate the color distortion of the banknote. If the color of the bank note has been to washed up, it is highly likely that the note is severely damaged. So, we provide a score based on that.

3.6.6 Highlighting damaged regions

The areas identified as damaged are highlighted in red over the aligned torn image. A heatmap is generated showing the intensity of RGB differences, giving a visual representation of subtle color-based damages.

3.6.7 Final report

We provide a binary damage percentage that reports the proportion of the note's area that is physically damaged, RGB damage percentage that reports the proportion of color distortion across the note, useful for subtle damages not captured by binary masks, edges and corners damage report which shows if the note is fine along the edges, key feature report that provides a visual representation of the key features, detecting if any of these are missing or damaged, and finally, list out the locations of the damaged regions, the area of the damage, and the general location of that damage.

3.7 Application Development

We have also developed a web app that takes the front and back side image of any note, uses our custom CNN classifier to predict the denomination of the note, and then provides a usability analysis of both the front and back side of the note. This has been developed using 'streamlit', an open-source platform to develop web-based applications for Python.

4 Results and Analysis

4.1 Classification Results

Table 2: Model Performance on Combined Dataset

Model	Params	Batch	LR	Epochs	Classes	F1	Size (MB)
VGG16	138M	64	1e-05	25	10	0.9315	512
ResNet50	25.6M	128	5e-05	25	10	0.9176	90.1
EfficientNet_B0	5.3M	128	5e-05	25	10	0.9417	15.6
Inception_V3	23.8M	64	0.0001	25	10	0.9519	83.5
Custom_CNN	**287k**	128	0.0001	25	10	0.9342	**1.1**

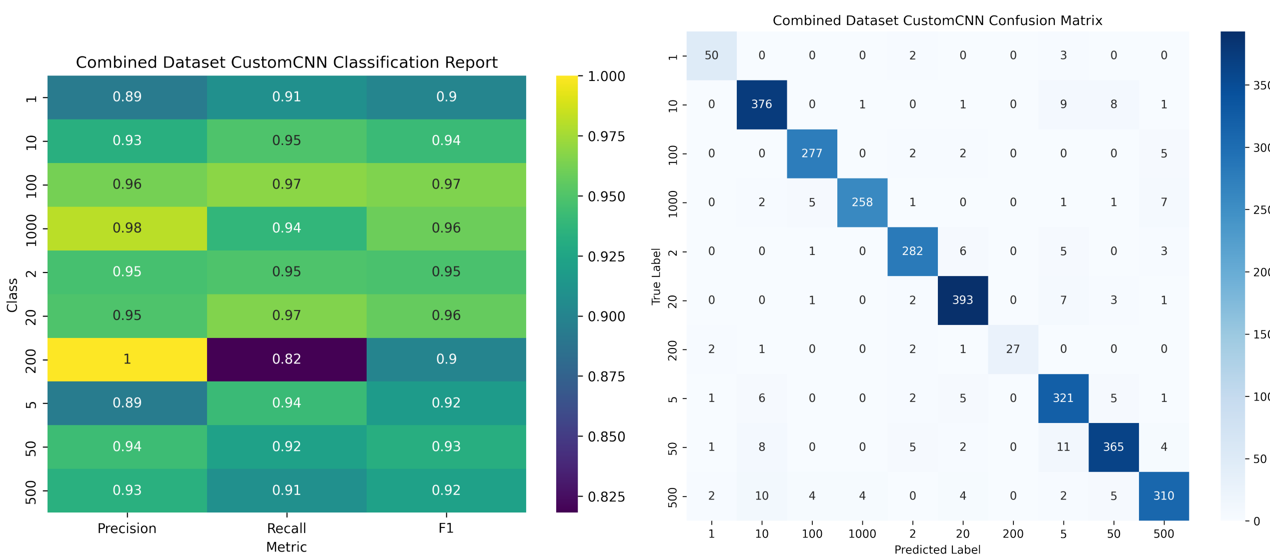

Figure 6: Classification report and confusion matrix of the Custom CNN model on the combined dataset.

We evaluated five convolutional neural network models on the combined dataset of Bangladeshi currency notes. All models were trained using the same training, validation, and test splits, with identical preprocessing and augmentation settings to ensure fair comparison. The combined dataset presents diverse image conditions and note designs, simulating real-world scenarios.

Table 2 summarizes the overall performance of each model. For brevity, only the classification report and confusion matrix of the Custom CNN model are included in the main text (Figure 6). Reports for VGG16, ResNet50, EfficientNet B0, and Inception V3 are provided in the Appendix.

All models achieved strong performance, with F1-scores ranging from 0.91 to 0.95. Inception V3 achieved the highest F1-score of 0.9519, while EfficientNet B0 and ResNet50 followed closely. The lightweight Custom CNN performed competitively, maintaining an F1-score above 0.91 with significantly fewer parameters.

Figure 7: Damaged bank notes used for usability analysis.

Precision and recall remained consistently high across most denominations, especially for frequently circulated notes like 100, 500, and 1000 Taka. Occasional misclassifications occurred between visually similar classes such as 5 and 50 Taka but were relatively rare.

Confusion matrix analysis confirms that all models effectively distinguish between note classes. The newly introduced 200 Taka note, despite being underrepresented, was correctly classified across all models, highlighting the robustness of the dataset and augmentation strategy.

While deeper models like Inception V3 performed slightly better, lightweight models such as EfficientNet B0 and Custom CNN provide strong accuracy with lower memory and computational demands, making them suitable for real-time applications.

4.2 Usability Detection Results

Figure 7 is used for the usability study. The note is extracted correctly, and the background is removed from the note.

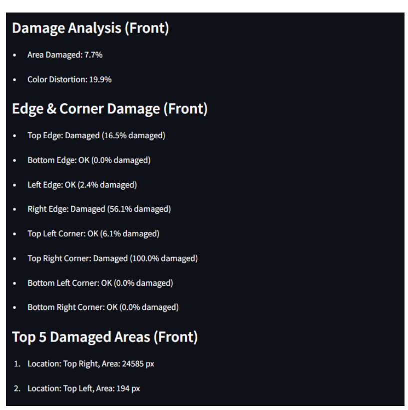

Figure 8: Report about the damage.

We then proceed to perform our analysis, which provides details about the damaged area, color distortion, percentage of damage along the edges and the corners, and the damaged area inside the note. This analysis is

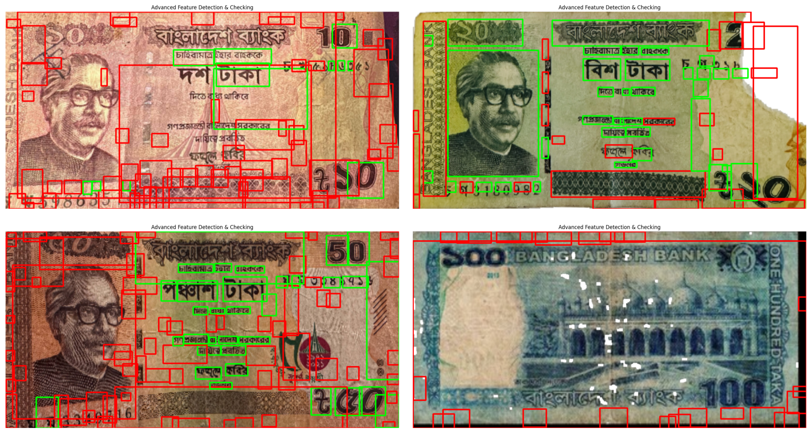

Figure 9: Feature detection for different banknotes.

provided for both the front and back sides of the note, as the damage can be on either side. Figure 8 illustrates the report that is generated.

Figure 9 then displays the features that are present in a standard bank note, and then overlaps those to the note that we uploaded of the damaged note. The green bounding boxes are used to represent the features that matched, whereas the red bounding boxes represent features which are not matched correctly, either because of color distortion or any sort of damage. It can be seen that since the top right corner is missing, the blob that detected the denomination in that corner is identified as missing. On the other hand, the image present has been highlighted as a match, as there is no distortion in that position.

Finally, the missing area has been marked in red to highlight the damage. It can be seen that even the top left corner, having a slight area missing, has been correctly identified and marked as missing in figure 10. This can also be noticed from the RGB difference that we calculated with the standard note.

4.3 Application Development

The application uses these functionalities to create a pipeline, where it asks the user to upload two images, the front and back sides of a note. It then uses the front side to identify the denomination of the note using the custom CNN classifier that we developed. After learning the denomination, it can now pull up the standard bank note of that particular denomination, and use that for comparison of the uploaded images of the note. It provides a detailed report of the note that contains all the analyses that have been mentioned earlier. The app can be used by visiting this link: Bank note usability.

5 Conclusion and Future Work

This project presented a complete pipeline for currency note classification and usability assessment using computer vision and deep learning. By combining multiple datasets and applying enhancement, alignment, and damage quantification techniques, we developed a system capable of identifying note denominations and evaluating their physical condition. The results show strong classification accuracy across all models and reliable detection of visible and subtle damage in real-world note images.

To further improve the system, we plan to create a dedicated dataset focused on usability assessment. This dataset will include various real-world damage scenarios such as handwritten marks, taped notes, and partial burns or stains. We also aim to handle challenging cases like extreme fading, ink spills, or overexposed images. A structured and labeled dataset targeting these conditions would enable more robust training and evaluation, helping to standardize damage classification across practical use cases.

Figure 10: Analysis of area torn and RGB difference with respect to standard note.

6 How to Run This Project

To reproduce the results or test the system locally:

1. Download the three public datasets from Kaggle (links provided in the codebase).
2. Install dependencies:

```
pip install -r requirements.txt
```

3. Follow the preprocessing steps to combine and deduplicate the datasets. Detailed Jupyter notebooks are provided for:
   - Dataset merging and cleaning
   - Classification model training and evaluation
   - Damage detection and usability analysis
4. To launch the web app:

```
streamlit run app.py
```

Upload front and back images of a note to receive denomination prediction and a full usability report.

All necessary code, notebooks, and expected outputs are included in the submission zip file.

References

[1] F. Oviedo, S. Vinnakota, E. Seleznev, H. Malhotra, S. Shaikh, and J. L. Ferres, "Banknote-net: Open dataset for assistive universal currency recognition," *arXiv preprint arXiv:2204.03738*, 2022.

[2] V. Meshram, V. Meshram, K. Patil, Y. Suryawanshi, and P. Chumchu, "A comprehensive dataset of damaged banknotes in indian currency (rupees) for analysis and classification," *Data in Brief*, vol. 51, p. 109699, 2023.

[3] V. Meshram, K. Patil, and V. Meshram, "Evaluation of top pretrained models using transfer learning on banknote dataset with quality parameter," *Ingénierie des Systèmes d'Information*, vol. 28, no. 3, p. 693, 2023.

[4] S. J. Jaman, M. Z. Haque, M. R. Islam, and U. A. Noor, "Bd currency detection: A cnn based approach with mobile app integration," *arXiv preprint arXiv:2502.17907*, 2025.

[5] T. D. Pham, D. T. Nguyen, W. Kim, S. H. Park, and K. R. Park, "Deep learning-based banknote fitness classification using the reflection images by a visible-light one-dimensional line image sensor," *Sensors*, vol. 18, no. 2, p. 472, 2018.

[6] T. Czimmermann, G. Ciuti, M. Milazzo, M. Chiurazzi, S. Roccella, C. M. Oddo, and P. Dario, "Visual-based defect detection and classification approaches for industrial applications—a survey," *Sensors*, vol. 20, no. 5, p. 1459, 2020.

[7] A. Farooq and M. Imran, "Innovating vending machine through object detection for currency recognition and enhanced cash handling," in *2023 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. IEEE, 2023, pp. 1–6.

[8] T. Akther, T. Begum, M. F. Begum, and N. Sojib, "Bangla money taka recognition dataset," https://www.kaggle.com/datasets/nsojib/bangla-money, 2018, kaggle.

[9] T. Mohammed, "Augmented bangla money dataset including 200 taka," https://www.kaggle.com/datasets/tazwarmohammed/augmented-bangla-money-dataset-including-200-taka/data, 2022, kaggle.

[10] R. Tasnim, S. T. Pritha, A. Das, and A. Dey, "Bangladeshi banknote dataset," https://www.kaggle.com/datasets/rahnumatasnim1604103/bangladeshi-banknote-dataset/data?select=dataset, 2021, kaggle.

Appendix

Figure 11: Classification reports of all models on the combined dataset.

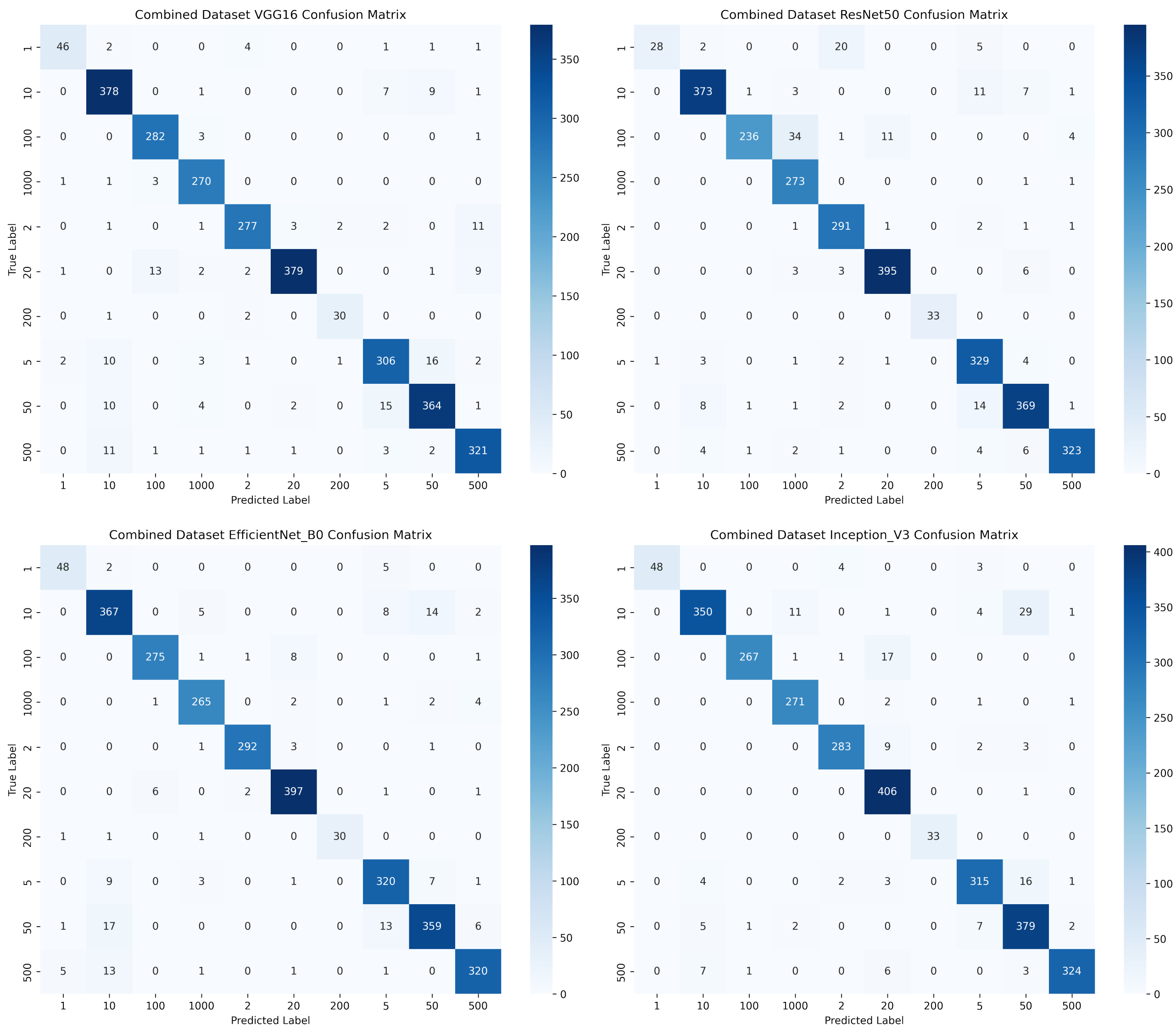

Figure 12: Confusion matrices of all models on the combined dataset.