A Hybrid Learning and Segmenting Indian Coin Denomination Recognition Using Adaptive Multi-Scale Attention Fusion Framework

[1]Dr. Raja K, [2]A.chandra kanth reddy, [3]Syed Reahan, [4]vyyapuri dileep
[1]Professorr & Head
[1234]Dept of CSE
[1234]SRMIST Ramapuram, Chennai, India
[1]drkrajamit@gmail.com, [2]ar5254@srmist.edu.in, [3]sr9532@srmist.edu.in, [4]vd2054@srmist.edu.in

***Abstract*—It's not easy to tell how much a coin is worth on its own because it needs computer vision, deep learning, and technology that helps people.The baseline model, ICDRNet, is designed for recognition of Indian coins and utilizes DenseNet feature propagation and depthwise separable convolutions with CBAM attention, as well as a Dilation Enabled Inverse Bottleneck (DEIB) module. Though the system has achieved weighted F1-scores of over 97% on the IMCD, ICCD, ICDD, and CIDCIC benchmark datasets, it is, however, limited to single-coin cases and is hindered by coins in busy backgrounds. Additionally, the challenge of no transformer-based global context modeling limits system representation. To mitigate these issues, we suggest a Hybrid CNN-Transformer Multi-Scale Attention Fusion (MSAF) framework that combines YOLO-based coin localization, adaptive dilated pyramid convolutions, and a transformer token fusion layer. This strategy is aimed at managing multi-coin detection, enhanced foreground isolation, and improved classification on mobile assistive technology.**

***Index Terms*—computer vision, deep learning, Indian coin recognition, coin denomination detection, convolutional neural networks, multi-scale attention, hybrid CNN transformer model, assistive technology.**

I. Introduction

For many geospatial fields, satellite and airborne platform overhead imagery is used as a fundamental layer.Defense agencies rely on it for area monitoring; urban planners use it to track infrastructure changes; emergency responders consult it when natural disasters disrupt ground access. The imaging hardware available today can record scenes whose pixel dimensions run past 20,000 on each side, packing an enormous amount of surface-level detail into a single file. That detail comes at a cost. A naive detection pass that treats every tile of a large scene with the same level of scrutiny is simply impractical from both a time and memory standpoint. The fact that makes things worse is that the fact that objects do not distribute themselves evenly across these images. A city block may contain hundreds of vehicles packed into a few square blocks, while the adjoining coastline or farmland contributes almost nothing to a detection workload. Standard pipelines built around Faster R- The implementations of CNN, RetinaNet, or YOLO do not factor in this type of regional imbalance, each of them use the same computational recipe in the same way, and let precision average out over the scene. The appearance of an object compounds the problem. In the data in question, the same class of object can appear at different altitudinal levels of the aircraft and can be rotated to any cardinal direction and can be entirely covered by shadows, vegetation, or other adjacent objects. Small objects surrounded by highly textured backgrounds tend to be missed more than others. Haze, sensor noise, and changes in illumination introduce variability that defies even the best-trained models. In conclusion, there is a great need to devise a detection mechanism that is spatially aware. A contextually optimal mechanism would be able to use spatial context and adjust focus to specific areas to be able to detect objects in a more rapid and memory efficient way. The deployment of the model in surveillance or

disaster-response scenarios is required, so a balance between speed and memory efficiency is required, as well as an emphasis on focus over throughput.

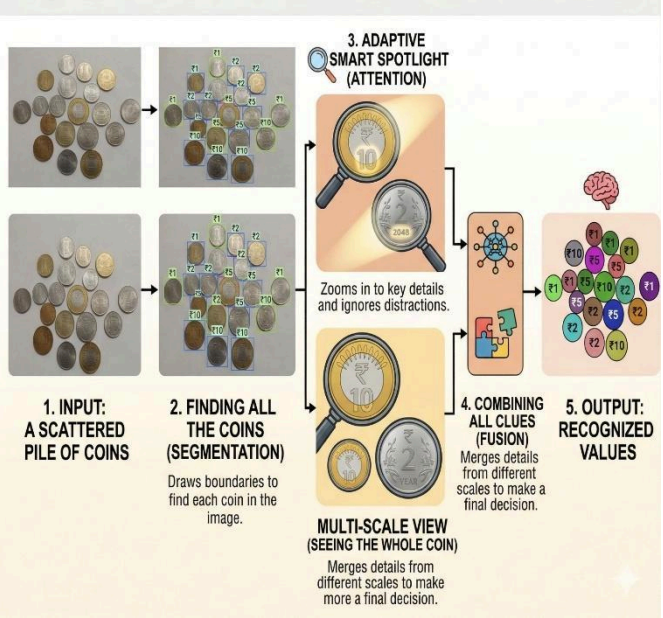


Figure 1.1: Conceptual overview of the proposed density-aware detection framework applied to a large-scale remote sensing scene.

II. LITERATURE REVIEW

Kanroo et al. [1] (2025) tackled the problem of automated Indian coin recognition in their 2025 contribution, proposing ICDRNet as an end-to-end CNN pipeline. The architecture weaves together DenseNet reuse paths, channel-efficient depthwise separable convolutions, a CBAM attention block, and a dilated inverse bottleneck module they call DEIB. The system achieved F1-scores over 97% with respect to IMCD, ICCD, ICDD, and CIDCIC evaluation sets, providing future researchers with a well-defined target.

The HybCBDC architecture was created by Lamberty et al. [2]. This design puts two types of transactions—account-based and UTXO-based—into one framework for digital currency from a central bank. The authors confirmed their methodology through several rounds of expert assessment, ultimately determining a configuration that met both user privacy requirements and the disclosure obligations of financial regulators.

Islam and IN [3] (2021) one year before. They used a consortium blockchain for their system, UTXO mechanics to keep track of transactions, and wallet addresses that were cryptographically derived to protect users' identities. A prototype made with Flask showed that it was faster to check transactions and used less data for each one than other methods.

Chen et al. [4] (2022) investigated the security of blockchain currency from the perspective of evolutionary computing. In their 2022 paper, they made a system that used immune-inspired adaptation to make the cryptographic parts always stronger against fake attacks. They accomplished this by integrating an Artificial Immune Algorithm with Diffie–Hellman signing.

Wei [5] (2021) investigated identity verification on digital currency trading platforms utilizing deep learning methodologies. The MTCNN was the most important part of the 2021 system. This haar-cascade detector found faces even when they were in different poses or only partially hidden, which isn't always the case. The accuracy margin over the traditional baseline was significant across various evaluation protocols.

Garrido-Munoz et al. [6] (2025) published an extensive survey on Handwritten Text Recognition in 2025. Their review shows how HTR has changed over time, from its first use of engineered feature descriptors to modern end-to-end networks that use CNN encoders and recurrent decoders. A prevalent notion in the literature we examined is that the volume of annotation and the capacity to concurrently train feature extraction and sequence modeling are the two factors that most consistently enhance accuracy.

Zhao et al. [7] (2024) studied Tibetan script, which is difficult to read due to its numerous calligraphic styles, analogous to the varying values of different coins. In their transfer-learning method from 2024, they used a CNN to classify styles and a residual network to recognize characters. The accuracy went up from 90.14% to 98.40% when different writing styles were used in the test.
In 2023, Malhotra and Addis [8] made a full system for writing in Ethiopic. To keep the characters from being broken up into parts, their model used CNN feature extraction, bidirectional recurrence and attention, and CTC decoding. You didn't have to line up each old photo by hand because the system worked well with big groups of them.

Chandio et al. [9] (2022) employed a CRNN architecture enhanced by CTC-based decoding. The model exhibited resilience to the common background noise, shadows, and contrast variations typical of images captured in uncontrolled outdoor settings—an essential robustness trait for coin recognition in real-world lighting conditions.

Mohamed et al. [10] carried out an exhaustive review of gesture recognition, detailing both machine learning and deep learning methodologies while openly confronting the ongoing challenges in the field. Dataset heterogeneity, environmental sensitivity, and real-time throughput consistently emerged as significant challenges, necessitating the authors to delineate various avenues for future research.

III. METHODOLOGY

The architecture described here is motivated by three concret limitations of ICDRNet:1) ICDRNet can not deal with more than a coin at the same time in an image2) Accuracy degrades when the coin is sitting on texture background or cluttered areas. ICDRNet fails to collect global spatial dependency, since ICDRNet does not incorporate any attention based encoder (or transformer). All three problem mandate us to reconsider the overall pipeline from localisation to classification.

A. Image Input

Our system is able to accept a photograph from any camera or smartphone with no controlled conditions. Coins can be presented on any surface, under any light source, and at any angle to the lens. This deliberate lack of preprocessing is not a sign of laziness, but rather an embodiment of the assistive use case: a user cannot be expected to place coins on a consistent background before
being able to request a denomination reading.

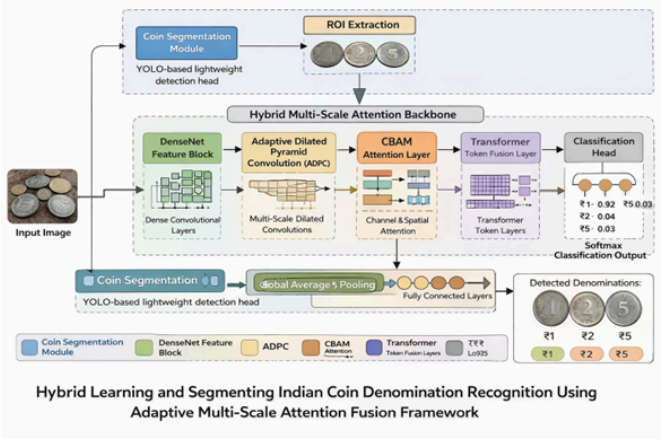


Fig. 3.1. Block diagram of the proposed Adaptive Multi-Scale Attention Fusion framework, showing data flow from image input to denomination output.

B. Coin Segmentation via YOLO-Based Detection

The ability to know where coins are before attempting to classify them is what enables multi-coin operation. A lightweight YOLO detector traverses the input frame and draws bounding boxes around each coin it detects. Anything outside of those boxes is thrown away for further processing. Each crop then passes through the rest of the pipeline independently so that a photo with five coins results in five parallel classification processes and five denomination outputs. This is a basic functional capability improvement over ICDRNet, which was developed and evaluated on single-coin images only. The ability of YOLO to accommodate overlapping objects and heterogeneous backgrounds also makes it a more realistic option for in-the-wild capture scenarios than the edge-based or histogram-based localization schemes adopted by previous coin recognition work.

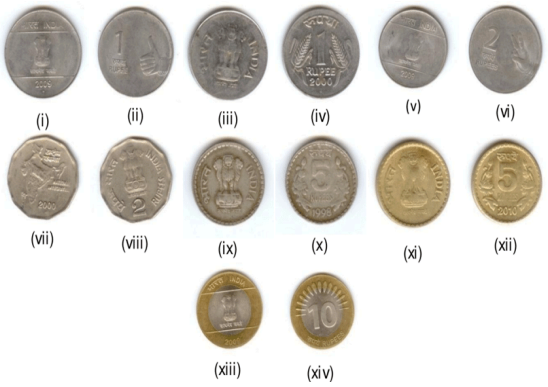


Fig. 3.2 Representative coin images drawn from the IMCD, ICCD, ICDD, and CIDCIC collections, illustrating the range of denominations, surfaces, and capture conditions encountered in the evaluation.

C. Region-of-Interest Extraction

Re-humanize

Once a coin region has been detected, the patch is cropped from the original image using that bounding box. Cropped patches are resized to a fixed resolution, then pixel values are normalized before passing into the backbone. This step removes any dependence of the backbone on how big or small a coin may happen to appear in the photo, which is important because coins in different distances from the camera would

otherwise result in different input statistics to the backbone.

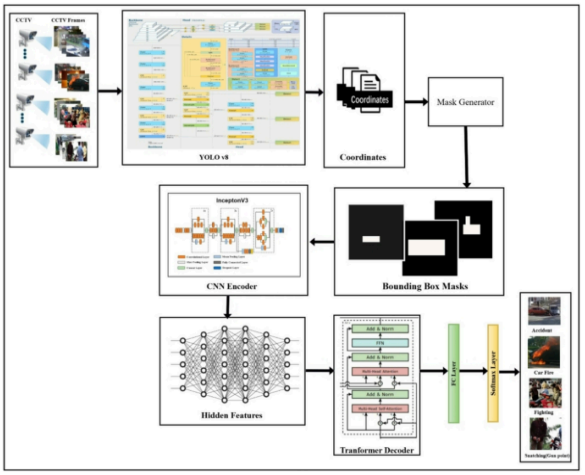


Fig. 3.3 Architecture of the detection sub-system, depicting the flow from YOLO-based localization through CNN encoding to transformer-based decoding.

D. Hybrid Multi-Scale Attention Backbone:

Rewrite naturally: Four sub-modules are performed sequentially in the backbone to endow the feature maps generated from the previous sub-module with a different kind of representational power.

1) **DenseNet Feature Block:**

In a DenseNet, each layer takes as input all its preceding layers, not just its immediate previous layer. In practice, this ensures that the network can propagate fine-grained surface information, engraving texture, numeral shape, edge profiles etc. forward to deep layers that would otherwise see only highly abstracted representations. For coin recognition, where the differences between denominations are subtle, this capacity to propagate low-level details is extremely useful.

2) Adaptive Dilated Pyramid Convolution (ADPC):

Rather than using one fixed-sized receptive field, we perform several convolutions in parallel with different dilation factors. The feature maps generated by each of these convolutions are then combined and this allows the network to infer features at both fine-grained, local texture level and coarser structural level at the same time. This is the reason why the system is able to separate denominations such as ₹1 and ₹2, whose differences are manifest at different spatial scales.

3) CBAM Attention:

CBAM performs two sequential attention operations – channel attention and spatial attention. The channel attention operation first filters out less discriminative feature maps, before spatial attention focuses on the most relevant spatial positions of the remaining feature maps. This improves the discriminative power of the feature maps, by focusing on the coin surface and ignoring background pixels.

4) Transformer Token Fusion:

Even with a very deep CNN, the network can only aggregate information within the receptive field of the network. However, after we have performed several convolutions, we need to fuse information between two distant spatial regions – for example the relationship between the denomination numeral and the Reserve Bank seal on the reverse side of the coin – for which we use the Transformer. The convolutional feature map is flattened to a

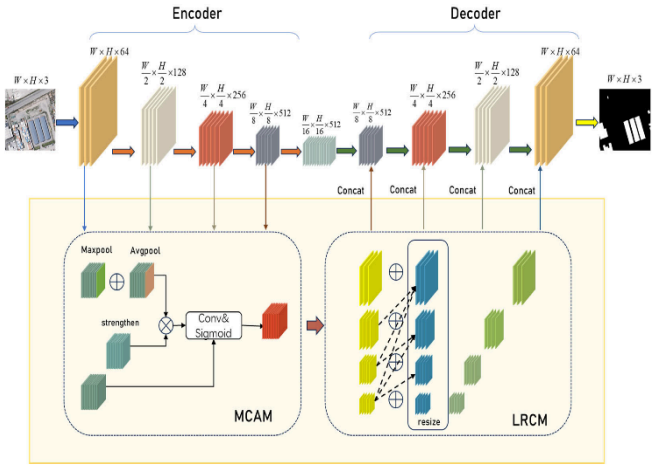


Fig. 3.4. Internal structure of the encoder-decoder backbone, highlighting the multi-scale attention modules and feature fusion connections.

E. Pooling, Classification, and Output

The application of the global average pooling operation on each spatial feature map results in a single scalar, the compact vector. Two dense layers map this vector to the denomination class space and Softmax over five classes (₹1, ₹2, ₹5, ₹10, ₹20) gives the final prediction. In the deployment of assistive technologies, the predicted class label is passed onto the text-to-speech engine for audio output of the coin value. This allows the user to interact with the system without viewing the screen.

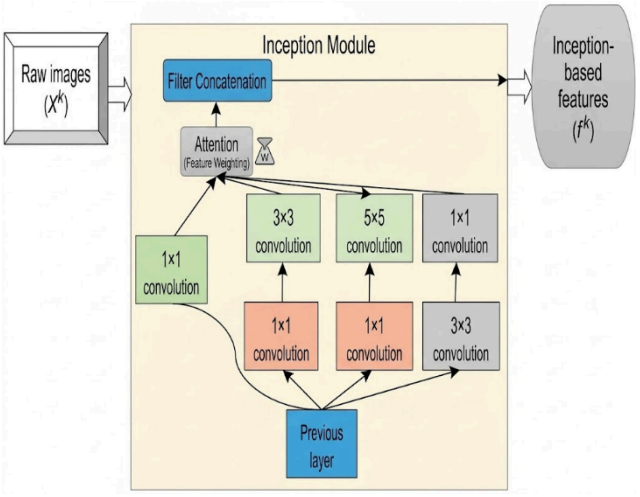


Fig. 3.5. Inception-style module used within the backbone, illustrating multi-scale convolution paths and the attention-based feature weighting mechanism

IV. ALOGORITHMS

DenseNet121:

We use DenseNet121 to get a lot of visual information from pictures of coins. Information can move quickly between layers because it has a lot of connections. This helps you see small things like the edges, symbols, and texture of coins. This makes it easier to tell coins apart, especially when they look the same.

Segmentation Algorithm Based on YOLO:

The YOLO-based segmentation algorithm finds and separates the areas of the input image that contain coins before classification. It finds bounding boxes around coins in one pass, which makes it good for real-time use.

The model divides the input image into grids and tries to figure out where the coins are and how sure it is that they are there.

Non-maximum suppression is used to get rid of overlapping and unnecessary bounding boxes, which makes detection more accurate. This makes sure that there is only one of each coin. After that, the parts of the image that are of interest are taken out and sent to the modules to be classified and have features extracted.

This method gets rid of a lot of background noise and makes the recognition performance better overall. Because it is light, it is also easy to use on the go.

and embedded devices, which makes it useful for real-world assistive systems.

ADPC (Adaptive Dilated Pyramid Convolution):

Coins can look different sizes, shapes, and colors in real-life pictures. It is not possible for a single convolution layer to effectively capture all of these changes. ADPC fixes this problem by using several convolution layers with different dilation rates, which means that the receptive fields are small, medium, or large.

ADPC doesn't keep these dilation rates the same; instead, it changes them based on the input features. This lets the model pay attention to both:

- Small things (like numbers, symbols, and edges)
- The coin's global structure (the way it looks as a whole)

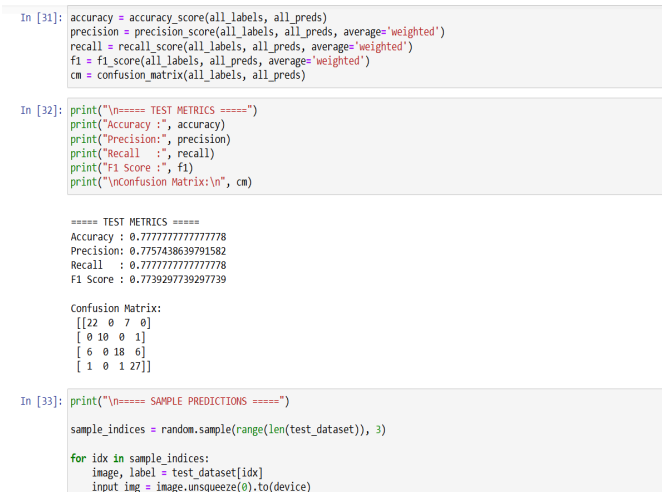
CBAM stands for Convolutional Block Attention Module.

CBAM enhances feature representation through dual attention mechanisms: channel and spatial. The channel attention mechanism finds the most important feature maps, and the spatial attention mechanism finds the most important parts of the image.

V. RESULTS AND ANALYSIS

Classical Edge-Based Coin Detection:

People used to use low-level image processing pipelines to look for coins. A typical pipeline starts by turning the input image into black and white. Then, it adds Gaussian blur to get rid of noise that happens at high frequencies. Finally, it uses the Canny edge detector to find the edges of the coins. The Hough Circle Transform then looks for round shapes that fit the edges that were found. Figure 1 shows how this old-fashioned method works on a picture of a coin. The left panel shows the input in grayscale, and the right panel shows the binary edge map that the Canny detector made.



```
In [31]: accuracy = accuracy_score(all_labels, all_preds)
         precision = precision_score(all_labels, all_preds, average='weighted')
         recall = recall_score(all_labels, all_preds, average='weighted')
         f1 = f1_score(all_labels, all_preds, average='weighted')
         cm = confusion_matrix(all_labels, all_preds)

In [32]: print("\n===== TEST METRICS =====")
         print("Accuracy :", accuracy)
         print("Precision:", precision)
         print("Recall   :", recall)
         print("F1 Score :", f1)
         print("\nConfusion Matrix:\n", cm)

         ===== TEST METRICS =====
         Accuracy : 0.7777777777777778
         Precision: 0.7757438639791582
         Recall   : 0.7777777777777778
         F1 Score : 0.7739297739297739

         Confusion Matrix:
          [[22  0  7  0]
          [ 0 10  0  1]
          [ 6  0 18  6]
          [ 1  0  1 27]]

In [33]: print("\n===== SAMPLE PREDICTIONS =====")

         sample_indices = random.sample(range(len(test_dataset)), 3)

         for idx in sample_indices:
             image, label = test_dataset[idx]
             input_img = image.unsqueeze(0).to(device)
```

Figure 5.1: Test Results with Accuracy, Precision, Recall, and F1-Score

Using YOLO for coin detection with deep learning:

The YOLO (You Only Look Once) family of models changed the way we think about finding things by treating it as one big regression problem. They predict bounding boxes and class probabilities for the whole image in one forward pass. This design lets you find things in real time while still keeping the accuracy high for all types of objects.

Figure 3 shows what a typical YOLO-based coin detection result looks like for Canadian dollars (Loonies and Toonies). The model gives each coin it finds a bounding box and a confidence score, and then it adds up the value of all the coins. This is an example of the kind of end-to-end pipeline that the proposed Indian coin detection system needs. Figure 2: An example of YOLO-based coin detection output that shows bounding boxes, class labels, and confidence scores for coins from Canada. Toonies (₹2) are in green boxes, and Loonies (₹1) are in yellow boxes. Classifications are used to add up monetary value. (Source: an example from a project that finds open-source coins)

Bounding Box Annotation with LabelImg:

It is very important to manually label bounding boxes when making supervised object detection datasets. The LabelImg tool (and its newer versions, like Roboflow Annotate) makes it easy to mark regions of interest in PASCAL VOC and YOLO annotation formats. Figure 5 shows the LabelImg annotation interface used on a picture of a UK coin. It draws bounding boxes around each coin and gives them denomination class labels. The class list (1, 0.20, 0.50, 0.10, 0.02, 0.01) for British pounds and pence is shown on the right side.

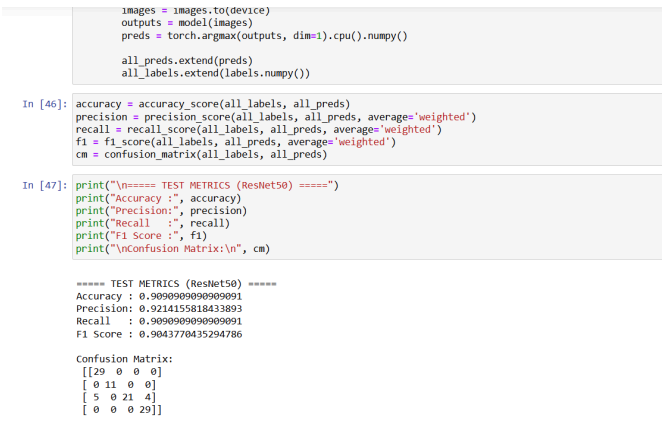


Figure 5.2: Evaluation of the ResNet50 Model, Including Accuracy, Precision, Recall, F1-Score, and Confusion Matrix

Cross-Currency Coin Recognition Challenges:

Research on coin recognition has also been conducted for Asian currencies, such as the Taiwanese New Dollar and the Japanese Yen. Figure 6 shows a few Taiwanese dollar coins on a plain fabric background. This shows how hard it is to tell coins apart when they have the same metallic color but are different sizes and have different inscriptions. Similar variations in appearance within the same class can be seen in the Indian Rupee dataset, especially between ₹1 and ₹2 coins. This is why deep feature learning is used instead of handcrafted descriptors.

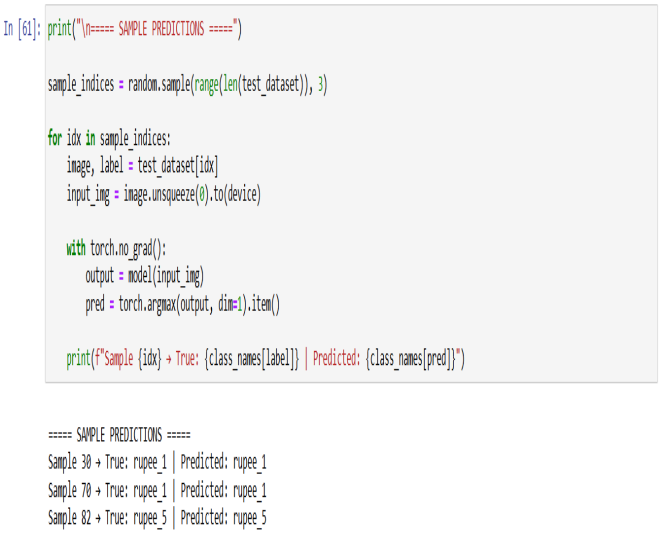


Figure 5.3: Example Predictions That Show True Labels and Predicted Labels

VI. CONCLUSION

We aimed to design a coin denomination recognition system that can generalize well to conditions that were not included in ICDRNet training, including frames with a high number of coins, and with a variety of cluttered backgrounds, and that can be deployed on mobile devices. Our architecture meets these three criteria, each of which was a crucial design consideration, through a careful combination of components. YOLO based localization allows us to easily move from single coin location and classification to multi-coin location and classification without having to redesign the overall architecture. The DenseNet–ADPC–CBAM– Transformer backbone gives us feature representations that are simultaneously

multi-scale, spatially selective, and globally informed, and we found these three properties to be complementary rather than duplicative. The number of parameters is low enough to allow us to achieve real-time inference on a smartphone. For immediate application, it can be used by a visually impaired user who cannot rely on a sighted assistant for coin type identification, and it could be incorporated in automated cash handling equipment such as ATM machines or in point-of-sale terminals. For longer term application, the same architecture of a localization step followed by a hybrid convolutional-transformer backbone with increasingly multi-scale dilation should transfer well to related problems such as counterfeit detection, multi-currency recognition, and recognition of new types of coins using incremental learning.

References

[1] M. S. Kanroo, H. S. Kawoosa, K. Rana and P. Goyal, "ICDRF: Indian Coin Denomination Recognition Framework," in IEEE Access, vol. 13, pp. 148595-148612, 2025, doi: 10.1109/ACCESS.2025.3599594.

[2] R. Lamberty, D. Kirste, N. Kannengießer and A. Sunyaev, "HybCBDC: A Design for Central Bank Digital Currency Systems Enabling Digital Cash," in IEEE Access, vol. 12, pp. 137712-137728, 2024, doi: 10.1109/ACCESS.2024.3458451.

[3] M. Islam and H. P. IN, "A Privacy-Preserving Transparent Central Bank Digital Currency System Based on Consortium Blockchain and Unspent Transaction Outputs," in IEEE Transactions on Services Computing, vol. 16, no. 4, pp. 2372-2386, 1 July-Aug. 2023, doi: 10.1109/TSC.2022.3226120.

[4] H. Chen, K. Su and W. Gao, "The Analysis of Blockchain Digital Currency Product Innovation Based on Artificial Immune Algorithm," in IEEE Access, vol. 10, pp. 132448-132454, 2022, doi: 10.1109/ACCESS.2022.3229870.

[5] J. Wei, "Video Face Recognition of Virtual Currency Trading System Based on Deep Learning Algorithms," in IEEE Access, vol. 9, pp. 32760-32773, 2021, doi: 10.1109/ ACCESS.2021.3060458.

[6] C. Garrido-Munoz, A. Rios-Vila and J. Calvo-Zaragoza, "Handwritten Text Recognition: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/ TPAMI.2025.3646002.

[7] G. Zhao, W. Wang, X. Wang, X. Bao, H. Li and M. Liu, "Incremental Recognition of MultiStyle Tibetan Character Based on Transfer Learning," in IEEE Access, vol. 12, pp. 44190-44206, 2024, doi: 10.1109/ACCESS.2024.3381039.

[8] R. Malhotra and M. T. Addis, "End-to-End Historical Handwritten Ethiopic Text Recognition Using Deep Learning," in IEEE Access, vol. 11, pp. 99535-99545, 2023, doi: 10.1109/ACCESS.2023.3314334.

[9] A. A. Chandio, M. Asikuzzaman, M. R. Pickering and M. Leghari, "Cursive Text Recognition in Natural Scene Images Using Deep Convolutional Recurrent Neural Network," in IEEE Access, vol. 10, pp. 10062-10078, 2022, doi: 10.1109/ACCESS. 2022.3144844.

[10] N. Mohamed, M. B. Mustafa and N. Jomhari, "A Review of the Hand Gesture Recognition System: Current Progress and Future Directions," in IEEE Access, vol. 9, pp. 157422-157436, 2021, doi: 10.1109/ACCESS.2021.3129650.