International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 04, April 2026)

Deep Neural Network Based Fake Currency Detection Using FPGA for Real Time Verification

K. Usha[1], M.Tulasi Venkat[2], K.Shiva Karthik[3], M. Praveen Kumar[4]
[1]Prof, Dept. of ECE, MVSR Engineering College, Hyderabad, India
[2,3,4]Student, Dept. of ECE, MVSR Engineering College, Hyderabad, India

***Abstract*—Fraudulent notes detection is essential for safeguarding the counties financial system, and will help in preventing the fake currency circulation by upholding the reliable error-free transaction process. This paper presents a stacking ensemble method by which we can have multiple models and train each of them with one particular feature that we want to extract from the currency and by employing a meta-model to learn the best way to combine will be a better approach to solve this issue. The recommended strategy adopts a squeezenet lite model and a simplified 6-layer CNN structure for lightweight structure and compatibility for hardware oriented implementation. Squeezenet-lite model consist of a initial input convolution layer followed by the 9 fire module each consists of the squeeze and expand layers which has 1×1 convolutional filters, enabling efficient feature extraction. The features considered for the training are RGB images & Image Quality Features(IQF) for acquiring both color-based and quality-related characteristics of currency notes. The dataset contains 200 and 500 denominations used for training and evaluation. Squeeznet model will analyze the RGB images and the IQF features are processed by 6-layer CNN. Based on these two model outputs the Support Vector Machine learns the thresholds of the classification of the currency will provide better performance.The models are trained in the MATLAB and targeted for FPGA implementation to enable faster inference and real-time counterfeit detection.**

***Keywords* — squeezenet lite model, fpga, image quality features, support vector machine, matlab.**

I. INTRODUCTION

The risk of counterfeit currency can have serious implications for economic security. The counterfeiters take advantage of the availability of high-resolution scanners & advanced technology in printing the forgeries, even if the bank notes have advanced security features such as security threads, watermarks, microprinting, and color-shifting inks.. The Reserve Bank of India(RBI) reports that even after the commencement of new Indian currency notes with enhanced security features, counterfeit versions began showing up within a short period.

Traditional identification approaches have manual inspection, UV lamp, magnetic sensors and infrared scanners.And these are bound to human error, processing delays & lack of adaptability to evolving forgery methods.

Recent studies have incorporated advanced Deep Learning Architectures such as CNN, ResNet, VGG and MobileNet for counterfeit currency detection to yield better predictive capabilities. This method evaluates features such as color, texture, edges & security patterns inscribed on the bank notes. We use Image pre-processing and data augmentation to further enhance the adaptability of these models. These systems facilitate dependable real-time currency authentication by leveraging camera based image acquisition.

Deep neural network based fake currency detection using FPGA will manifest a hardware-software co-design that integrates Edge AI and FPGA to provide fast & accurate currency authentication. However, the accuracy of these deep learning techniques is high due to the extraction of features from the images. But the application of these techniques is limited due to high computational complexities and latency involved in the process. The problem is overcome by implementing a lightweight CNN architecture on FPGA for efficient detection of counterfeit currencies.

The major highlights of this research are as follow:

1. A dual-modality input strategy using RGB images and Image Quality Features(IQF) for ₹200 and ₹500 Indian Rupee denominations.
2. A lightweight SqueezeNet Lite architecture using only 1×1 convolution Fire modules for efficient RGB feature extraction suitable for FPGA deployment.
3. A simplified 6-layer CNN for IQF-based quality feature extraction alongside RGB features are jointly supplied to a Support Vector Machine(SVM), which learns the detection boundaries for improved classification accuracy.
4. MATLAB based model training and FPGA targeted implementation for real-time, low-latency counterfeit currency detection.

II. LITERATURE REVIEW

The theory and method on which the proposed system is based are derived from three major studies on currency authentication with deep learning techniques.

In 2024, the study **“Develop a Robust System for Detecting Counterfeit Iraq Currencies Based on Deep Learning Techniques”** tested the feasibility of five machine learning techniques and implemented three different CNN architectures, namely, VGG16, InceptionV3, and MobileNetV2. The researchers used the techniques with the help of the transfer learning approach on a database that increased from 1,359 to over 8,000. The final system was implemented on a Raspberry Pi 5 with the help of a camera and a UV light module [1].

The **"Deep Learning-Based Fake Banknote Detection System"** paper, published in 2025, has contributed to this field by proposing a hybrid CNN model that integrated ResNet-50, Sobel, and Canny edge detection to detect fine details such as microprinting and watermarking, along with using GAN to synthesize additional training data and achieving 98.3% accuracy, 97.5% precision, and 98.1% recall on 4,000 images [6].

Aseffa et al. performed an exhaustive comparative assessment of InceptionV3, MobileNetV2, XceptionNet, and ResNet50 with six optimizers: Adam, SGD, RMSProp, Nadam, Adadelta, and Adagrad on 30,580 augmented images of banknotes, where MobileNetV2 with RMSProp yielded 96.98% training accuracy [3].

These models prove the efficiency of deep learning in the detection of currencies. However, these models also have some common drawbacks. The models used in these examples, such as VGG16 with 138 million parameters, ResNet50, and InceptionV3, require a lot of computations which make them unsuitable for real-time embedded systems. Even the model with the lowest computational requirements, namely MobileNetV2, took 87 seconds to classify the image on the Raspberry Pi 3 B+, making it unsuitable for point-of-transaction authentication. The proposed model eliminates these drawbacks with the use of the lightweight model SqueezeNet, which achieves similar accuracy with 50 times fewer parameters than the VGG16 model, and the parallel dual-stream architecture, which processes both RGB color features and IQF-based structural quality maps in parallel, running on the Xilinx Vivado FPGA.

III. DATASET AND PREPROCESSING

A. Dataset and Preprocessing

We need a dataset of Indian currency with both the fake and real note images which has captured the both visual richness and the structural quality characteristics. This dataset consists of the ₹200 and ₹500 Indian Rupee denominations. Because these Indian Rupee denominations are the most frequent forged notes in circulation.

The RGB dataset has a total of 2,736 images, in which 1,368 are genuine and other 1,368 are fake images. Here, we necessarily decided to divide the dataset equally across the both categories which will help us to get better training accuracy without overlifting of the model. The RGB method includes all the colors of the visible spectrum of each note. This makes it particularly effective at detecting alterations to the color of the security ink, color gradient, and color watermark. These factors are difficult to replicate for a counterfeiter.

The dataset for the IQF (Image Quality Feature) is created independently by the IQF v3 process and includes 2,620 images in total—1,310 genuine and 1,310 counterfeit images, maintaining perfect equilibrium. Instead of collecting color data, the focus of the IQF maps is on the quality of the printed surface, which includes the sharpness of edges and micro-texture regularity. The quality descriptors highlight the tiny differences between authentic and fake currencies that are not immediately apparent from regular color images.

Both data sets have been split independently by means of a stratified split of 80:20. The RGB data set has 2,188 training images and 548 validation images, while the IQF data set has 2,096 training images and 524 validation images. Here, during the inference phase, the banknote passes through two branches of Convolutional Neural Network (CNN) models. One branch works with the input of the RGB image, while the other branch works with the input of the IQF map.Both of these CNN models have learned feature weights, which are in turn used for training a Support Vector Machine (SVM), which will classify the banknote as genuine or fake. The complete dataset structure is presented in Table I.

Model	Genuine	Counterfeit	Training(80%)	Validation (20%)	Total
RGB	1368	1368	2188	548	2736
IQF	1310	1310	2096	524	2620

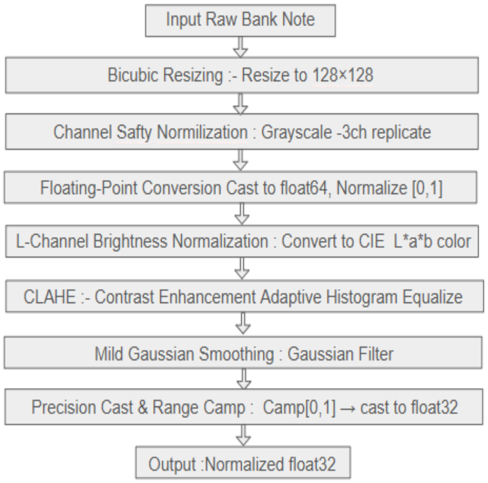
B. Training of RGB images with squeezenet

Extracted features via RGB Branch :`

- *Colour Palette Fidelity:* The proprietary ink ratios used in the real currency are not replicable with commercial CMYK printers. The RGB approach is able to learn the exact limits of the real colour palette, which results in the “Fake” classification when the mixture of the colour palette crosses the standard limits of the CMYK printing domain.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 04, April 2026)

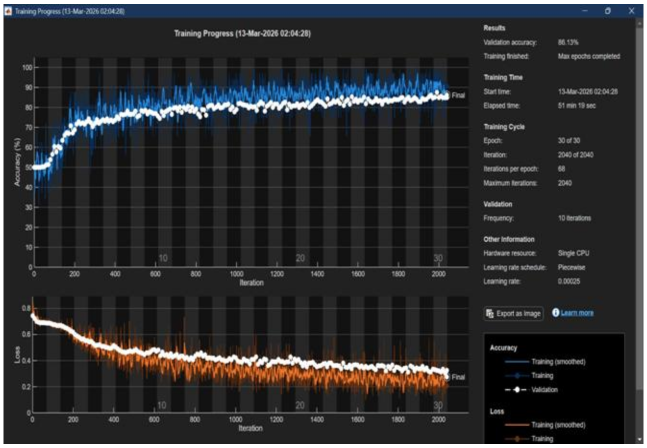
- *Spatial Geometry:* The deeper layers learn the semantic geometry, i.e., the location of the Gandhi portrait with respect to the denomination text. The real currency must always show the coloured blocks at the exact pixel locations.
- *Optically Variable Ink Patterns:* The real currency shows security threads with the characteristic colour change when the currency is tilted. The flat photocopy shows a static monochromatic line. The RGB approach is able to learn the gradients associated with the real optically variable ink.
- *Image Preprocessing Pipeline:* The unified preprocessing function is applied in an identical fashion during training, validation, and real-time FPGA inference to maintain distributional consistency. The data processing pipeline is divided into seven stages:



Squeezenet lite Overview: The key innovation in the overall SqueezeNet architecture is the Fire Module, a two-stage structure that replaces large-kernel convolutional layers with a much smaller parameter count and a much lower multiply-accumulate count, two of the most critical bottlenecks in FPGA inference.

The Fire Module consists of a squeeze layer, containing 1x1 convolutional filters, and an expand layer, containing a combination of 1x1 and 3x3 convolutional filters, to reduce and then restore representational ability, while keeping overall parameter count minimal. The Fire Module has a dramatic impact by restricting the number of channels presented to the costly 3x3 filters by the 1x1 squeeze filters, leading to a large reduction in multiply-accumulate and memory bandwidth, two of the most critical bottlenecks in FPGA inference. 1x1 convolution is a powerful operation, serving a variety of different functions in the overall SqueezeNet architecture.

The RGB-CNN was trained on a balanced dataset of 2,736 RGB images, using a stratified 80-20 split for training and validation sets, yielding 2,188 and 548 samples, respectively. We have performed training over 30 epochs with a piecewise learning rate policy starting at 0.00025. This equates to a total of 2,040 iterations, or 68 iterations per epoch. The accuracy level after the training process was found to be at 86.13% and the loss curve had an almost constant decreasing pattern, starting at 0.8 and decreasing to about 0.4 which indicates a stable training process with convergence. The training process was performed on a CPU with one core, taking 51 minutes and 19 seconds. Every 10 iterations, the validation process was performed.



Results
Validation accuracy: 86.13%
Training finished: Max epochs completed

Training Time
Start time: 13-Mar-2026 02:04:28
Elapsed time: 51 min 19 sec

Training Cycle
Epoch: 30 of 30
Iteration: 2040 of 2040
Iterations per epoch: 68
Maximum iterations: 2040

Validation
Frequency: 10 iterations

Other Information
Hardware resource: Single CPU
Learning rate schedule: Piecewise
Learning rate: 0.00025

Fig I Training graph of RGB Branch

Table II
Layers Present in RGB Branch

Layer	**Type**	**Output Size**	**Parameters**	**Notes**
Conv1(x2 streams)	Conv(3x 3)	112x56x 64	3520	ReLu+Bn,R GB, IQF
Maxpool1 (x2)	Maxpool 3x3,s2	55x27x 64	—--	Both Streams
Fire2(x2)	sq:16,ex: 128	55x27x 128	11,408	Fire module
Fire3(x2)	sq:16,ex: 128	55x27x 128	15,376	Fire module
Fire4(x2)	sq:32,ex: 256	55x27x 256	33,024	Fire module
Maxpool2 (x2)	Maxpool 3x3,s2	27x13x 256	—---	Both streams
Fire5(x2)	sq:32,ex: 256	27x13x 256	68,096	Fire module
Fire6	Sq:48, ex:192	13x13x 192	15,600	Fire module
Fire7	sq:48,ex: 192	13x13x 192	18,672	Fire module
Fire8	sq:64,ex: 256	13x13x 256	28,992	Fire module
Fire9	sq:64,ex: 256	13x13x 256	33,088	Fire module
Fusion	Concatenation	27x13x5 12	—---	RGB+LBPmerge
Dropout	p=0.5	27x13x 384	—-	Regularization
Conv-Cls	Conv 1x1	27x13x2	770	Classifier head

C. Training of IQF images with squeezenet lite

Extracted features via IQF Branch : The IQF CNN model is used for counterfeit currency detention by focusing on physical and structural properties of banknotes rather than focusing on the visual appearance of currency notes. The purpose of using IQF is to transform input image into feature rich representation highlights with some features such as ink behaviour, edge sharpness and paper texture.

- *Ink Behaviour:* The counterfeit currency is created using an inkjet printer or laser printer, and hence the ink appears flat and lacks even saturation, along with blending. The counterfeit currency is detected by distinguishing the saturation in the real currency and the counterfeit currency using the IQF algorithm.

- *Sharpness of Details / sobel edge magnitude:* The counterfeit currency is identified as:

(1) Blurred edges while scanned.
(2) Presence of dot patterns, usually found in printers.
(3) Loss of details.

The problem is solved by using the CLAHE (Contrast Limited Adaptive Histogram Equation) method, and then the sobel function is applied to find the edges and the magnitude of the edges.

$$M(x,y) = \sqrt{Gx(x,y)^2 + Gy(x,y)^2}$$

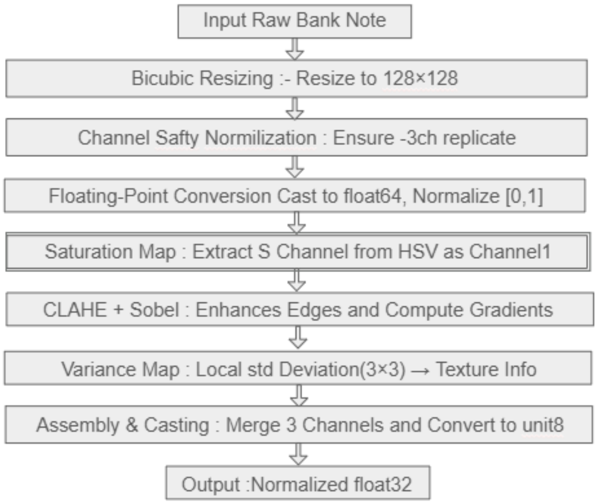
If the value is high, then the edges are sharp, and hence the currency is real. If the value is low, then the edges are blurred, and hence the currency is fake.

- *Local variance / Micro-Roughness:* The standard deviation filter is applied to capture the texture and detect the paper as well as the real and fake currency.

$$\sigma(x,y) = \mathrm{std}\{\, I(i,j)=(i,j) \ni N3(x,y) \,\}$$

If the variance is high, then the texture is well defined, and hence the currency is real.

Image Pre-processing pipeline: The unified preprocessing function is applied in an identical fashion during training, validation, and real-time FPGA inference to maintain distributional consistency. The data processing pipeline is divided into seven stages:



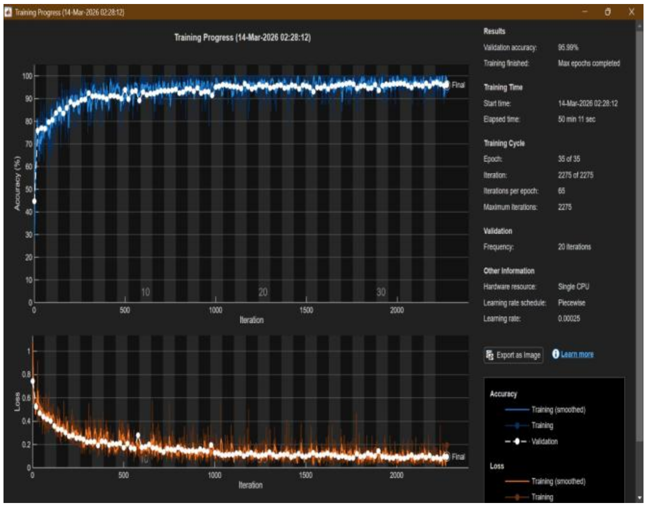
Deep Structural 6-Layer CNN Overview:

Architectural Distinctions: Unlike the lightweight SqueezeNet structure, optimized for spatial RGB classification, the modality uses a 6-Block Deep Structural Convolutional Neural Network structure optimized for its use case.

Deeper Linear Feature Extraction: It replaces SqueezeNet's "bottleneck compression" approach in Fire Modules with a deeper linear hierarchy. This is critical in distinguishing subtle numerical map differences, such as distinguishing variation in texture from ink saturation.

Progressive Context Building: The structure gradually constructs complex textual meanings by extracting core map gradient features in Block 1 and develops deeper structural context features as inputs progress through Block 6.

Aggressive Regularization: To prevent overfitting and prevent the CNN from memorizing features such as the exact location of text features such as "Children Bank," a dense 60% Dropout regularization structure and Global Average Pooling are incorporated throughout the CNN structure



Results
Validation accuracy: 95.99%
Training finished: Max epochs completed

Training Time
Start time: 14-Mar-2026 02:28:12
Elapsed time: 50 min 11 sec

Training Cycle
Epoch: 35 of 35
Iteration: 2275 of 2275
Iterations per epoch: 65
Maximum iterations: 2275

Validation
Frequency: 20 iterations

Other Information
Hardware resource: Single CPU
Learning rate schedule: Piecewise
Learning rate: 0.00025

Fig II Training graph of IQF Branch

Table III
Layers Present in IQF Branch

Layer	**Type**	**Output Size**	**Parameters**	**Notes**
Conv1	Conv(3x3)	112x56x64	3520	ReLu+Bn, RGB, IQF
Maxpool1	Maxpool 3x3,s2	55x27x64	——-	Both Streams
Fire2	sq:16,ex:128	55x27x128	11,408	Fire module

Fire3	sq:16,ex:128	55x27x 128	15,376	Fire module
Fire4	sq:32,ex:256	55x27x 256	33,024	Fire module
Maxpool2	Maxpool 3x3,s2	27x13x 256	—---	Both streams
Fire5	sq:32,ex:256	27x13x 256	68,096	Fire module
Fusion	Concatenation	27x13x512	—---	RGB+LBPmerge
Fire6	Sq:48, ex:384	27x13x 384	182,976	Post-Fusion
Dropout	p=0.5	27x13x 384	—-	Regularization
Conv-Cls	Conv 1x1	27x13x2	770	Classifier head
Global AvgPool	GAP	1x1x2	—--	Spatial Collapse
Output	Softmax	2	—-	Genuine// Fake

IV. PREDICTION METHODS

A. *Prediction by SVM*

The system takes the features of both RGB, IQF models which are actually different. The RGB model focuses on evaluating surface level ink colors, whereas the IQF model focuses on examining structural characteristics such as paper micro-roughness & edge sharpness. It will merge this final probability and help each model to work independently and make a final decision for more reliable classification.

The softmax output values represent its confidence when classifying currency either real or fake. Both RGB, IQF model probabilities are combined into single 4 dimensional continuous feature vector which includes

- ➔RGB model probability [RGB_real_Probability, RGB_Fake_probability].
- ➔IQF model probability [IQF_Real_Probability, IQF_Fake_probability]

B. Radial Basis Function(RBF)

Architectural Distinctions: The system uses a support vector machine (SVM) to take that decision better. SVM uses RBF allowing flexible decision boundaries which further can handle complex situations. Ex: If the RGB model confidently ensures the note is real due to good ink color, but IQF suspects it as fake due to improper paper texture, to overcome this type of situation SVM balances them, makes accurate decisions.

Training Configuration : SVM is trained only on 4 number probability outputs from RGB, IQF modules which improves decision making avoiding feature extraction.

Data Standardization : SVM analyzes real / fake based on 4 dimensional space, reduces complexity by avoiding tuning of parameters there by making simple boundaries.

Strict Error Penalty : SVM avoids mistakes by avoiding misclassification during training, making precise boundaries which exactly distinguish real, fake notes instead of allowing overlaps.

5-Fold cross-validation: Data set is classified into 5 parts. SVM is trained on 4 parts and tested on other parts; this process repeats 5 times. Ensuring the model is learning patterns instead of memorizing data helps to work well on unknown notes.

Table IV :
SVM Fusion Configuration

Parameter	**Configuration**	**Purpose/Deep explanation**
Vector input size	1x4 double array	Analyzes opinions: RGB_Real, RGB_Fake, IQF_Real,

		IQF_Fake
Algorithm Class	Super Vector Machine	Advanced margin classifier optimization
Kernel Function	rbf(radial basis)	Allows the boundary to curve around contradictory logic
Kernel Class	Heuristic(auto)	Curvature dynamically adjusts to the data's spread
Standardization	True	Equalizes the mathematical weighting of RGB and IQF branches
Box Constraint (C)	10	Enforces a strict penalty mathematically by outlawing overlapping scores
Validation Stratagem	5-fold cross validation	KFold integrity test; prevents memorization by hiding 20% of data during 5 separate training runs

C. Prediction By Score-level Fusion

This method is used to check whether the currency note is real or fake by combining outputs of 2 independently trained Neural networks. The 2 CNN networks are RGB and IQF.

The fusion is done such a way that the system combines both networks by using a weighted average like RGB model weight :0.60 , IQF model weight:0.40.

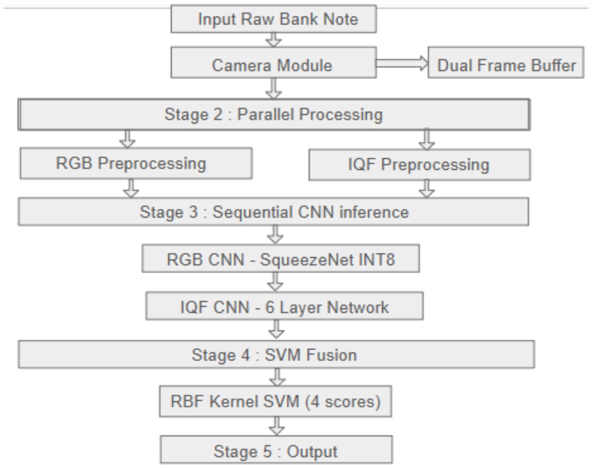
Fused fake score=0.60*RGB_Fake+0.40*IQF_Fake

IV. FPGA IMPLEMENTATION

A. Hardware Platform and Deployment

The dual-stream currency authentication system, pre-trained for real-time embedded inference, was implemented on the Xilinx Nexys 7 Artix-7 100T FPGA (XC7A100TCSG324-1) with no host processor required. The selection was based on the provision of a guaranteed latency, the flexible parallel data path that is well-suited for pixel-level feature extraction, and the logic capacity that is sufficient to allow for a faithful reproduction of the software inference pipeline. The Nexys 7 FPGA has 101,440 LUTs, 240 DSP48E1 slices, 135 BRAM tiles, and 128 MB of external DDR2 SDRAM. The use of DDR2 memory was critical because the two quantized SqueezeNet architectures would collectively consume 24 MB of weight memory, which exceeds the capacity of BRAM.

The complete hardware system follows a five-stage pipeline:



B. Model Quantization

The MATLAB Deep learning toolbox is used for quantizing both Squeezenet lite models from FP32 to INT8.There are 3 steps involved they are

*Calibration Pass:*_ Some test images are used to understand data range which are passed through the model, observed activation values for each layer. The use of doing this step is to identify minimum and maximum values that help to convert into INT8.

Scale Factor Computation:_ Range of INT8 value is -128 to +127 . The use of calculating Scale Factor is to convert floating point values to integers without loss of information.

Validation Pass:_ After converting model into INT8 it is tested again and checked accuracy to ensure model works even after conversion.

C. Hardware Description language generation and deployment

The quantized model is given to MATLAB Deep Learning Toolbox; this tool converts matlab code to Verilog Code . The tool generates two outputs they are:

Computation engine:_ Convolution, ReLu activation,Global average pooling are some of the verilog files which are used to perform all cnn operations.

Weight Memory files: These files contain trained weights which are stored inside FPGA memory during Programming.

These trained weights are insufficient for inference, without corresponding computation engine no mac operations, activation or pooling is performed.

D. Parallel Processing Architecture & Sequential CNN inference

Dual-Port Frame Buffer: System captures image (128x128) and stores into frame buffer. Double buffering technique is used where 2 memory slots are maintained in which one is processed and other is filled with the next frame.

Two-Processing modules work in Parallel: Dual port B-RAM allows two parallel reads . If one Port is working on RGB Preprocessing , Another is working on IQF Feature Generation both modules can read simultaneously ,and works independently.

CNN Inference: CNN Inference happens sequentially, since after preprocessing RGB data goes into RGB Squeezenet ie., [RGB_real,RGB_fake], IQF data goes into IQF Squeezenet ie., [IQF_real,IQF_fake] which runs one after other but not parallel.

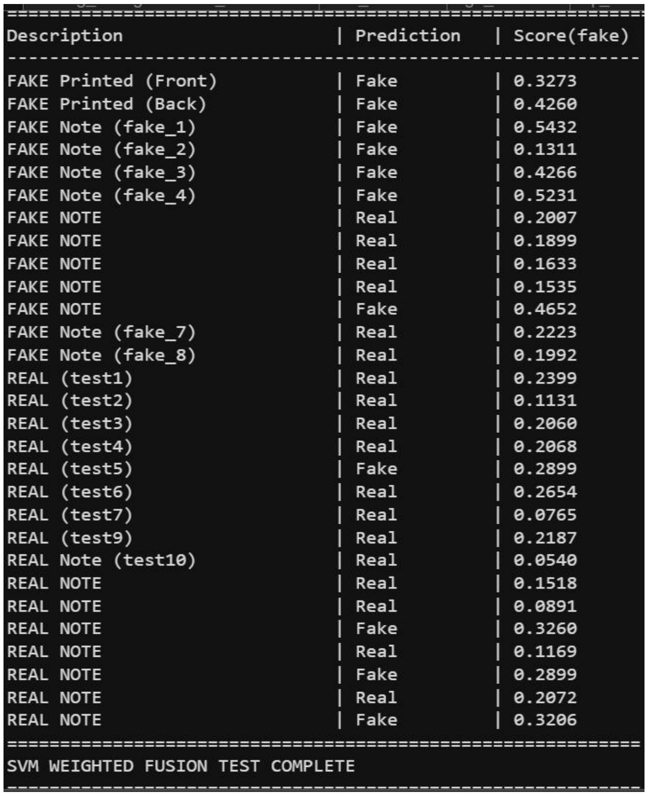
V. RESULTS AND DISCUSSION

The proposed SNN-based air We have done using both methods ie.,Prediction by SVM and Prediction using score level Fusion

A. Result of Prediction by SVM

The weights used here are RGB model :0.55 and IQF model:0.45.

The SVM-based fusion results demonstrate that the system is able to make more reliable and balanced decisions by combining the confidence scores from both the RGB and IQF models into a 4-dimensional feature space. From the output, most fake samples (e.g., fake_1, fake_3, fake_4) are correctly classified as *Fake* with relatively higher fake scores (around 0.42–0.54), indicating strong detection capability. At the same time, a majority of real samples (test1–test10) are correctly classified as *Real* with lower fake scores (generally below 0.25), showing that the model maintains good discrimination between classes. The results indicate that SVM fusion provides a more adaptive and accurate classification mechanism, especially in complex scenarios where individual models alone may not be sufficient.

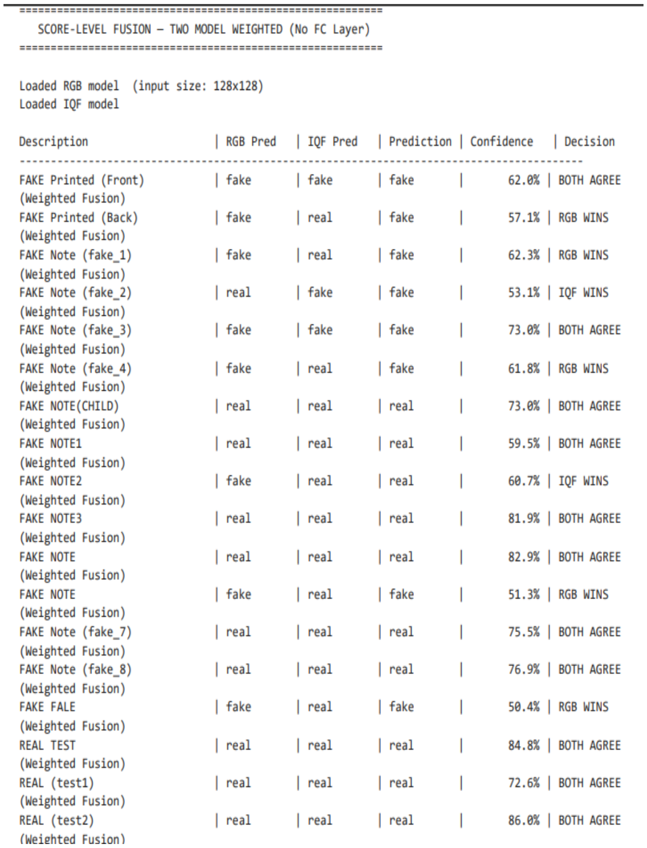


```
============================================================
Description                     | Prediction    | Score(fake)
------------------------------------------------------------
FAKE Printed (Front)            | Fake          | 0.3273
FAKE Printed (Back)             | Fake          | 0.4260
FAKE Note (fake_1)              | Fake          | 0.5432
FAKE Note (fake_2)              | Fake          | 0.1311
FAKE Note (fake_3)              | Fake          | 0.4266
FAKE Note (fake_4)              | Fake          | 0.5231
FAKE NOTE                       | Real          | 0.2007
FAKE NOTE                       | Real          | 0.1899
FAKE NOTE                       | Real          | 0.1633
FAKE NOTE                       | Real          | 0.1535
FAKE NOTE                       | Fake          | 0.4652
FAKE Note (fake_7)              | Real          | 0.2223
FAKE Note (fake_8)              | Real          | 0.1992
REAL (test1)                    | Real          | 0.2399
REAL (test2)                    | Real          | 0.1131
REAL (test3)                    | Real          | 0.2060
REAL (test4)                    | Real          | 0.2068
REAL (test5)                    | Fake          | 0.2899
REAL (test6)                    | Real          | 0.2654
REAL (test7)                    | Real          | 0.0765
REAL (test9)                    | Real          | 0.2187
REAL Note (test10)              | Real          | 0.0540
REAL NOTE                       | Real          | 0.1518
REAL NOTE                       | Real          | 0.0891
REAL NOTE                       | Fake          | 0.3260
REAL NOTE                       | Real          | 0.1169
REAL NOTE                       | Fake          | 0.2899
REAL NOTE                       | Real          | 0.2072
REAL NOTE                       | Fake          | 0.3206
============================================================
SVM WEIGHTED FUSION TEST COMPLETE
------------------------------------------------------------
```

B. Result of Prediction by Score level fusion

Also we have done using another method ie., Prediction using Score level Fusion. This approach gives higher importance to the RGB model because color features like ink distribution and gradients are strong indicators of authenticity, while the IQF model contributes structural details such as texture and edge sharpness.

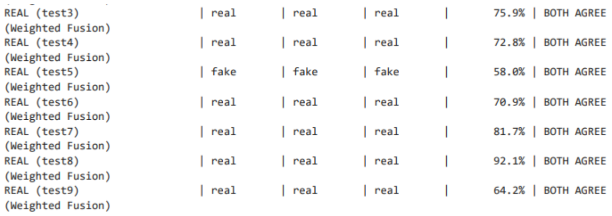
The main advantage of this method is its low computational complexity and fast execution, making it highly suitable for FPGA-based real-time systems. However, compared to SVM-based fusion, it lacks adaptive decision boundaries since the weights are fixed and cannot handle conflicting scenarios as intelligently. Overall, score-level fusion provides a good trade-off between speed and performance, especially for hardware-constrained environments. In this method we have used weights RGB model:0.60,IQF model:0.40.



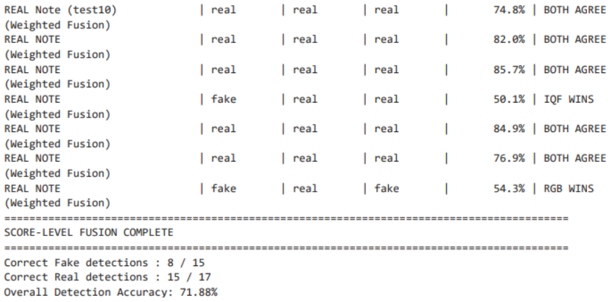
```
============================================================
   SCORE-LEVEL FUSION – TWO MODEL WEIGHTED (No FC Layer)
============================================================

Loaded RGB model  (input size: 128x128)
Loaded IQF model

Description                | RGB Pred   | IQF Pred   | Prediction | Confidence   | Decision
------------------------------------------------------------------------------------
FAKE Printed (Front)       | fake       | fake       | fake       |        62.0% | BOTH AGREE
(Weighted Fusion)
FAKE Printed (Back)        | fake       | real       | fake       |        57.1% | RGB WINS
(Weighted Fusion)
FAKE Note (fake_1)         | fake       | real       | fake       |        62.3% | RGB WINS
(Weighted Fusion)
FAKE Note (fake_2)         | real       | fake       | fake       |        53.1% | IQF WINS
(Weighted Fusion)
FAKE Note (fake_3)         | fake       | fake       | fake       |        73.0% | BOTH AGREE
(Weighted Fusion)
FAKE Note (fake_4)         | fake       | real       | fake       |        61.8% | RGB WINS
(Weighted Fusion)
FAKE NOTE(CHILD)           | real       | real       | real       |        73.0% | BOTH AGREE
(Weighted Fusion)
FAKE NOTE1                 | real       | real       | real       |        59.5% | BOTH AGREE
(Weighted Fusion)
FAKE NOTE2                 | fake       | real       | real       |        60.7% | IQF WINS
(Weighted Fusion)
FAKE NOTE3                 | real       | real       | real       |        81.9% | BOTH AGREE
(Weighted Fusion)
FAKE NOTE                  | real       | real       | real       |        82.9% | BOTH AGREE
(Weighted Fusion)
FAKE NOTE                  | fake       | real       | fake       |        51.3% | RGB WINS
(Weighted Fusion)
FAKE Note (fake_7)         | real       | real       | real       |        75.5% | BOTH AGREE
(Weighted Fusion)
FAKE Note (fake_8)         | real       | real       | real       |        76.9% | BOTH AGREE
(Weighted Fusion)
FAKE FALE                  | fake       | real       | fake       |        50.4% | RGB WINS
(Weighted Fusion)
REAL TEST                  | real       | real       | real       |        84.8% | BOTH AGREE
(Weighted Fusion)
REAL (test1)               | real       | real       | real       |        72.6% | BOTH AGREE
(Weighted Fusion)
REAL (test2)               | real       | real       | real       |        86.0% | BOTH AGREE
(Weighted Fusion)
```



```
REAL (test3)               | real       | real       | real       |        75.9% | BOTH AGREE
(Weighted Fusion)
REAL (test4)               | real       | real       | real       |        72.8% | BOTH AGREE
(Weighted Fusion)
REAL (test5)               | fake       | fake       | fake       |        58.0% | BOTH AGREE
(Weighted Fusion)
REAL (test6)               | real       | real       | real       |        70.9% | BOTH AGREE
(Weighted Fusion)
REAL (test7)               | real       | real       | real       |        81.7% | BOTH AGREE
(Weighted Fusion)
REAL (test8)               | real       | real       | real       |        92.1% | BOTH AGREE
(Weighted Fusion)
REAL (test9)               | real       | real       | real       |        64.2% | BOTH AGREE
(Weighted Fusion)
```



```
REAL Note (test10)         | real       | real       | real       |        74.8% | BOTH AGREE
(Weighted Fusion)
REAL NOTE                  | real       | real       | real       |        82.0% | BOTH AGREE
(Weighted Fusion)
REAL NOTE                  | real       | real       | real       |        85.7% | BOTH AGREE
(Weighted Fusion)
REAL NOTE                  | fake       | real       | real       |        50.1% | IQF WINS
(Weighted Fusion)
REAL NOTE                  | real       | real       | real       |        84.9% | BOTH AGREE
(Weighted Fusion)
REAL NOTE                  | real       | real       | real       |        76.9% | BOTH AGREE
(Weighted Fusion)
REAL NOTE                  | fake       | real       | fake       |        54.3% | RGB WINS
(Weighted Fusion)
==========================================================================================
SCORE-LEVEL FUSION COMPLETE
==========================================================================================
Correct Fake detections : 8 / 15
Correct Real detections : 15 / 17
Overall Detection Accuracy: 71.88%
```

C. Discussion

The proposed system demonstrates an effective hardware–software co-design for real-time counterfeit currency detection by combining lightweight deep learning models with FPGA implementation. The dual-modality approach, using SqueezeNet Lite for RGB features and a 6-layer CNN for IQF-based structural features, significantly enhances detection reliability by capturing both visual and texture-based characteristics of currency notes. The integration of an SVM with an RBF kernel for score-level fusion further strengthens decision-making by resolving conflicts between the two models and creating robust classification boundaries. Additionally, the use of INT8 quantization, parallel preprocessing, and optimized memory utilization makes the system suitable for FPGA deployment with reduced latency and resource consumption. However, the system is limited by moderate accuracy (~86%), a restricted dataset focused only on ₹200 and ₹500 notes, and potential sensitivity to real-world variations such as lighting and note condition, indicating scope for further improvement.

VI. CONCLUSION AND FUTURE WORK

This work presents an efficient FPGA based currency detection system combining deep learning,hardware acceleration for real time detection. Both models RGB,IQF analyze structural,physical characteristics of currency notes by preserving their strengths individually which helps the system to achieve more reliability and accuracy. For better decision making RGB,SVM are used to resolve conflicts between both models by interpreting probability scores of both models. The system is optimized for hardware deployment through 16-bit Quantization with high processing speed, Parallel execution.

Future enhancements can focus on improving both accuracy and real-world applicability of the system. Expanding the dataset to include multiple denominations, varying lighting conditions, and damaged or worn notes will improve generalization. More advanced lightweight architectures such as MobileNetV3 or EfficientNet-Lite can be explored to boost performance without compromising hardware efficiency. The fusion mechanism can be upgraded using attention-based or neural fusion techniques instead of SVM for better adaptability. On the hardware side, implementing fully parallel CNN execution or transitioning to Zynq-based systems can reduce latency and improve throughput. Additionally, integrating multi-modal sensing techniques such as UV or infrared analysis, along with mobile or IoT-based image acquisition, can make the system more robust and practical for real-world deployment in ATMs and currency verification systems.

Acknowledgment

The authors would like to express their sincere gratitude to Prof. K. Usha, Department of Electronics and Communication Engineering, MVSR Engineering College, Hyderabad, for her invaluable guidance, continuous encouragement, and technical support throughout the course of this research work.

REFERENCES

[1] F. Al Azzo, "Develop a Robust System for Detecting Counterfeit Iraq Currencies Based on Deep Learning Techniques," Journal of Computational Analysis and Applications, vol. 33, no. 2, pp. 754-767, Sep. 2024.

[2] S. Veling, "Fake Indian Currency Recognition System by using MATLAB," International Journal for Research in Applied Science and Engineering Technology, vol. 7, no. 4, Apr. 2019.

[3] G. Gebremeskel, "Developing a Model for Detection of Ethiopian Fake Banknote Using Deep Learning," Journal of Sensors, Dec. 2022.

[4] S. T. Kara, S. S. Raju, B. Rajulapati, S. Loya, and N. Vanteru, "Detection of Fake Indian Currency Using Deep Convolutional Neural Network," in Proc. 2023 IEEE 3rd Mysore Sub Section International Conference (MysuruCon), 2023, doi: 10.1109/MysuruCon59703.2023.10396993.

[5] M. Laavanya and V. Vijayaraghavan, "Real Time Fake Currency Note Detection using Deep Learning," International Journal of Engineering and Advanced Technology, vol. 9, no. 1S5, pp. 95-98, Dec. 2019.

[6] P. Saraswathi, R. Swapna, N. Priyanka, K. Rajya Lakshmi, and P. Bhavana, "Artificial Intelligence Based Fake Bank Notes Classification and Detection," International Advanced Research Journal in Science, Engineering and Technology, vol. 12, no. 4, Apr. 2025.

[7] S. Patil and R. K. Jain, "Fake Currency Detection Using Image Processing," International Journal of Computer Applications, vol. 153, no. 6, pp. 45-49, Nov. 2016.

[8] A. Roy, D. Ghosh, and S. Banerjee, "Indian Currency Recognition and Verification Using Machine Learning," International Journal of Engineering Research and Technology, vol. 9, no. 5, pp. 1120-1124, May 2020.

[9] M. S. Ali and A. Kumar, "Detection of Counterfeit Currency Using Convolutional Neural Networks," International Journal of Scientific and Technology Research, vol. 9, no. 3, pp. 6231-6235, Mar. 2020.

[10] K. N. Reddy, P. S. Rao, and M. Harsha, "Deep Learning Based Currency Note Verification System," in Proc. IEEE International Conference on Intelligent Systems and Signal Processing, pp. 214-219, 2021.