Jordanian Currency Recognition Using Deep Learning

Salah Alghyaline
Department of Computer Science, the World Islamic Sciences and Education University, Amman, Jordan

***Abstract*—Automatic Currency Recognition (ACR) has a significant role in various domains, such as assessment of visually impaired people, banking transactions, counterfeit detection, digital transformation, currency exchange, vendor machines, etc. Therefore, developing an accurate ACR system enhances efficiency across several domains. The contribution of this paper is three-fold; it proposed a large dataset of 2799 images and seven denominations for Jordanian currency recognition. The second contribution proposed an efficient multiscale VGG net to recognize Jordanian currency. Third, popular CNN architectures on the proposed dataset will be evaluated, and the result will be compared with the proposed architectures. Four metrics were used in the evaluation. The experimental result showed the accuracy of the proposed Multiscale VGG outperformed VGG16, DenseNet121, ResNet50, and ResNet101 and achieved 99.88%, 99.88%, 99.89%, and 99.98% accuracy, precision, sensitivity, and specificity.**

***Keywords*—Automatic currency recognition; deep learning; VGG**

I. Introduction

Currency recognition uses image processing techniques to identify currency. People use currency in their daily lives, and it is important to develop an automatic way to recognize it [1]. Currency recognition systems have many applications, such as ATM machines, vendor machines, money exchange shops, bank systems, blind people's assistants, and the detection of fake currency [2]. According to a study in 2020, 43.3 million people were blind, and 258 million had low vision ability, most of them from developing and poor countries [3].

Deep learning approaches showed super results in the image processing field [4]. Many approaches in the literature were proposed to develop currency recognition systems for different currencies worldwide. There are more than 180 currencies worldwide, and each country has its specifications for currency in terms of size, paper, pattern, and color [5]. Here is a shortage of publicly available datasets for Jordanian currency[1].

This study reviews the literature on Jordanian currency recognition and proposes a new dataset that includes 2799 images representing seven denominations of Jordanian currency. The proposed dataset includes paper banknotes and metals. Moreover, the proposed dataset is evaluated on different Deep learning architectures, and some of these architectures were modified to improve recognition accuracy.

The paper is organized as follows: Section II reviews the related work in automatic currency recognition. Section III presents the materials and methodology. This includes the proposed dataset, the CNN architectures used, and the proposed Multiscale VGG. Implementation Details are described in Section IV. It consists of an experimental environment, model parameter settings, performance metrics, and experimental results. Finally, Section V concludes the paper.

II. Related Works

A. Recognition of Currency

Due to rapid technological advancement, many applications require automatic currency recognition, such as detecting fake currency in vending machines and ATMs and assisting visually impaired persons [6].

Many approaches were proposed in the literature to address automatic currency recognition; some of them used traditional methods such as histogram analysis, edge detection, descriptors-based features like Histogram of Oriented Gradients (HOG) [7], Speeded-Up Robust Features (SURF) [8] and Scale-Invariant Feature Transform (SIFT) [9]. After extracting the image features, the simple way to predict the class label is to use Template matching between the extracted features and the predefined feature. Support Vector Machines (SVM), k-nearest Neighbors (k-NN), and Random Forests are also used by many approaches to classify the extracted features. These traditional approaches are usually computationally efficient but could struggle with challenged images with noise such as lighting, clutter, occlusion, orientation and background; moreover, they extracted a limited number of features [10] [11].

Deep learning has shown superior results in image recognition during the last few years. Many CNN architectures have been proposed and used broadly in image processing and computer vision applications, such as VGG [12], DenseNet121 [13], Reset [14], and Inception V3 [15]. It is reported that deep learning achieved high accuracy in automatic currency recognition [16] [17] [18][19].

B. Literature Review

Automatic currency recognition is significant in people's daily lives, helping blind people or those with vision problems, automatic selling machines, detecting fake bank notes, and banking applications. Therefore, many approaches were proposed to address this problem.

The study in [2] proposed a system to recognize Indian real currency from fake; the system starts with noise removal by converting the image into a grayscale image and resizing it into a fixed dimension. Then, the image histogram is extracted, and template matching is used to determine whether the currency is real or not.

[1] https://drive.google.com/drive/folders/1faAhWB7B8CohvTBTAxhNp5C2m4HuPa8A

The study in [20], developed a mobile application to recognize Yemeni paper currency. 1600 images and four currency denominations were used. The images trained on the MobileMe architecture, a group of images ranging from 18 to 29, were used to evaluate the model accuracy; the model achieved 100% accuracy.

The study in [21], proposed an approach to recognize three Nigerian paper currencies. The approach is based on a color histogram for feature extraction and a rule-based technique for image classification. The dataset includes 300 images, and the approach achieved 98.66% accuracy.

The study in [16] is employed to recognize Indian banknotes. Four CNN architectures were implemented using a basic sequential model, VGG 16, AlexNet, and MobileNet architectures were trained and tested on a dataset of 1270 images. The approach achieved 97.98%, 92.81%, 89%, and 71.66% on the four architectures, respectively.

The research [22] developed an approach to recognizing US banknotes. The approach is based on the Speeded-UP Robust Features (SUFR) descriptor and uses template matching to recognize currency.

The research [23] developed a banknote recognition system for three countries: the US, Egypt, and Saudi Arabia. SURF was used for key-point detection, a histogram of oriented gradients (HOG), and the scale-invariant feature transform (SIFT) were used to describe the features. A support Vector Machine (SVM) was used to classify the features into 12 classes of currency denominations. The system achieved a 99.2% accuracy rate.

The study in [24] developed a system using Python and Raspberry Pi to distinguish original and counterfeit 100 NTD; Mean Gray Values (MGVs) were analyzed in specific regions; these regions represent note security regions.

The research in [25] proposed an open dataset for the Indian currency. The dataset includes 5125 images resized into 1280 × 768 pixels, captured by a Galaxy A33 5G and Apple iPhone 6. The dataset includes four denominations (10, 20, 50 and 100 Rupees), including the new and old shapes of these denominations.

Faster R-CNN and YOLOv5 [17] were trained to recognize rupiah banknotes. A dataset of 1120 images that represent eight classes is used to make the comparison. R-CNN and YOLOv5 achieved accuracies of 98.65% and 82.1%, respectively.

The study in [18] proposed CNN architecture to predict four currencies: the US dollar, Euro, Jordan dinner, and Korean won. The architecture is an improved version of YOLO-v3 architecture. It consists of 69 conventional layers. The model was evaluated on a dataset of 21,020 images and achieved an accuracy of 83.96%. The Jordanian currency includes nine denominations, and the images are 1024×1024 pixels.

III. Materials and Methodology

A. Proposed Dataset

The dataset was collected from college students whose ages ranged from 18-22. The students used their phones, which have various characteristics. Each person was asked to capture the currency on both sides, and each side was captured from two angles.

Fig. 1 shows sample pictures from each denomination. The captured images were taken in different conditions of lighting, orientation, background, sizes, and quality. The images include many challenges, such as background objects, some parts of the currency not being captured, being taken from different distances, some parts of the currency being damaged, being captured in different lighting conditions, and being captured from different phones. The captured images were resized into 448×448 pixels. The dataset includes 2799 images with JPEG formats represent seven denominations of the Jordanian currency as follows: 50JD: 409 images, 20JD: 397 images, 10JD: 419 images, 5JD:419 images, 1JD: 425 images, 50 piasters: 363 images, 25 piasters: 367, as shown in Fig 2. The Jordanian banknotes and coins have old and new shapes, and both are used now. There are differences in the color, security features, and regions, as shown in Table I which makes it difficult to recognize them.

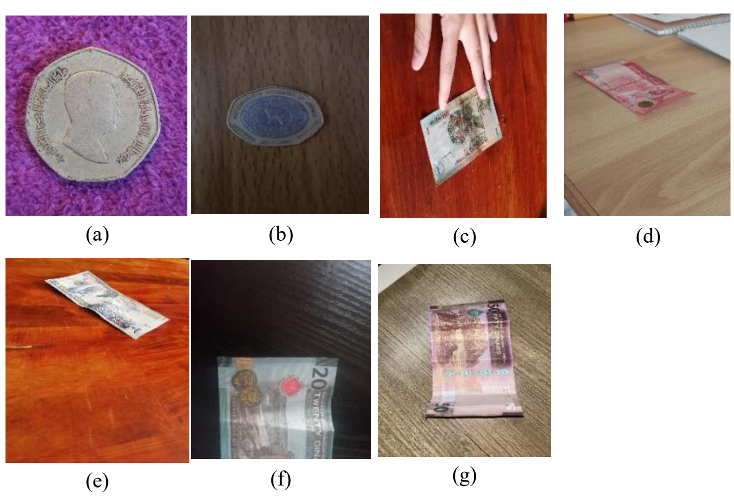


Fig. 1. Sample pictures from the proposed Jordanian currency dataset including images taken under different conditions: (a) 25 piasters (b) 50 piasters (c) 1JD (d) 5 JD (e) 10JD (f) 20 JD (g) 50 JD.

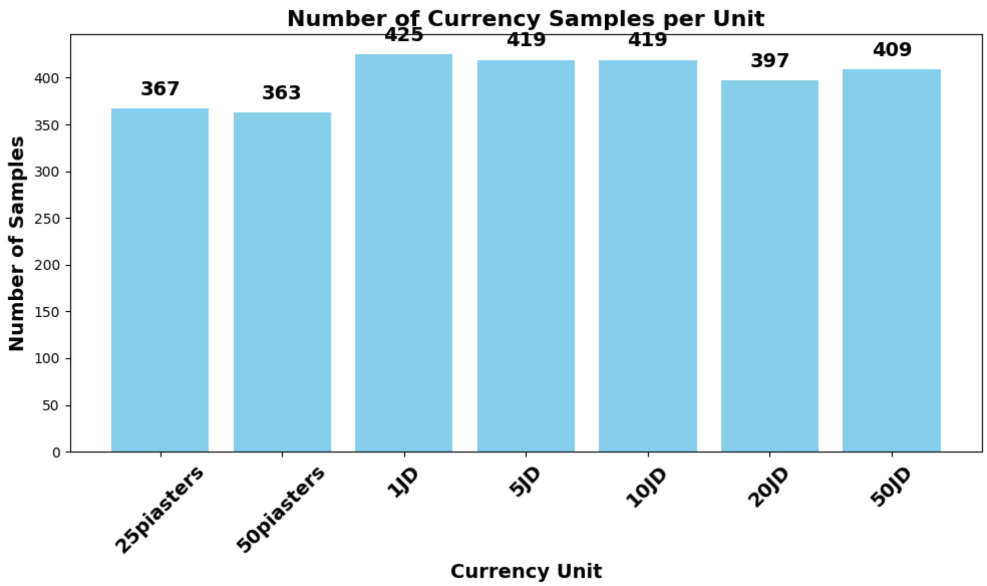


Fig. 2. Number of collected samples for each currency unit.

TABLE I. SAMPLE PICTURES FROM OLD AND NEW SHAPES FOR THE CURRENCIES IN JORDAN

Denomination	Old shape	New shape
25 piasters		
50 piasters		
1JD		
5JD		
10JD		
20JD		
50JD		

B. CNN Architectures

A set of popular CNN architectures was adopted in this paper. VGG has a simple structure and is considered a baseline for image classification tasks. ResNet shows a significant result when training deep networks with residual connections. DenseNet121 improved feature reuse by concatenating features from different layers.

The research in study [12] VGG developed at the University of Oxford by the Visual Geometry Group (VGG) in 2015. They studied the effect of increasing the number of convolutional layers from 16 to 19. The architecture achieved state-of-the-art on the ImageNet Challenge 2014. VGG16 has a simple and effective structure; it includes a sequence of five blocks. Each block consists of two to four convolutional layers followed by a max pooling. Finally, the features are flattened using four fully connected layers. The last layer represents the number of classes. The convolutional layers are 3×3 windows, and filter sizes range from 64 to 512. The max pooling is a window of 2×2 to reduce the feature size to half. Relu is used for activations, Table II explains the architecture of VGG16 model.

TABLE II. THE ARCHITECTURE OF VGG16

Layer Type	Filter Size	Number of Filters	Output Size	Details
Input	-	-	224×224×3	RGB image input
Convolution (x2)	3×3	64	224×224×64	Two 3×3 conv layers with ReLU
Max Pooling	2×2	-	112×112×64	Stride 2
Convolution (x2)	3×3	128	112×112×128	Two 3×3 conv layers with ReLU
Max Pooling	2×2	-	56×56×128	Stride 2
Convolution (x3)	3×3	256	56×56×256	Three 3×3 conv layers with ReLU
Max Pooling	2×2	-	28×28×256	Stride 2
Convolution (x3)	3×3	512	28×28×512	Three 3×3 conv layers with ReLU
Max Pooling	2×2	-	14×14×512	Stride 2
Convolution (x3)	3×3	512	14×14×512	Three 3×3 conv layers with ReLU
Max Pooling	2×2	-	7×7×512	Stride 2
Fully Connected (x2)	-	4096	1×1×4096	Two fully connected layers with ReLU
Output (SoftMax)	-	Number of classes	1×1×Number of classes	Classification layer

DenseNet introduced [13] the dense connection between CNN layers. Each CNN layer is connected with other layers in a forward manner. The architectures achieved state-of-the-art results on ImageNet, SVHN, CIFAR-10, and CIFAR-100 datasets. The architecture includes four Dense Blocks and three Transition Layers. The Dense Block consists of a group of convolutional layers that are densely connected. Each dense layer includes Batch Normalization, ReLU, 1x1 Convolution and 3x3 Convolution. At the same time, the Transition layers include 1x1 Convolution and 2x2 Average Pooling layers to reduce the size of the feature map, Table III presents the architecture of DenseNet121 model.

ResNet baseline architecture [14] was derived from the VGG architecture. It uses the same filler size of 3×3 and a pooling layer to down sample the feature map. However, it reduced the number of filters compared with VGG, which reduced the model size and introduced the residual networks instead of learning unreferenced data from the network layer. The network is 8x deeper than VGG; however, it is reported that the mAP of VGG16 and ResNet101 on PASCAL VOC 2007/2012, was 70.4% and 73.8%, with 3.2% improvement. In comparison, the VGG16 gained a 6% improvement (21.2%TO 27.2%) on the COCO dataset compared to VGG16. Stated the first place in the ILSVRC 2015 classification competition. The residual is calculated according to Eq. (1) if the input and output feature map have the same dimensions; if not, Eq. (2) is used.

$$\mathrm{y} = \mathcal{F}(\mathrm{x}, \mathrm{W_i}) + \mathrm{x} \tag{1}$$

TABLE III. ARCHITECTURE OF DENSENET121

Layer Type	Output Size	Filter Size / Details
Input	224x224x3	RGB image
Convolution	112x112x64	7x7 conv, stride 2
Max Pooling	56x56x64	3x3, stride 2
Dense Block 1	56x56x256	6 bottleneck layers (growth=32)
Transition Layer 1	28x28x128	1x1 conv, 2x2 avg pool
Dense Block 2	28x28x512	12 bottleneck layers
Transition Layer 2	14x14x256	1x1 conv, 2x2 avg pool
Dense Block 3	14x14x1024	24 bottleneck layers
Transition Layer 3	7x7x512	1x1 conv, 2x2 avg pool
Dense Block 4	7x7x1024	16 bottleneck layers
Global Avg Pooling	1x1x1024	
Fully Connected Layer	Number-of-classes	

where $\mathcal{F}$ is the residual mapping, x and y are the input and the output. W_i is the weights of the convolutional layers in the residual block.

$$\mathrm{y} = \mathcal{F}(x, W_i) + W_s\mathrm{x} \tag{2}$$

W_s is the weight of the 1×1 convolution used for projection.

InceptionV3 [26] reported that it is computationally more efficient compared with VGG architecture. Therefore, it can be trained and tested using bigger data. The architecture includes a sequence of inception modules. Each inception module represents several convolutional and pooling layers that operate in a parallel fashion and are then concatenated together. As shown in Fig. 3 [27], there are 1×1 convolutional layers used to reduce the feature size and feature extraction: 3×3 and 5×5 convolutional layers to capture spatial features. Max pooling and average pooling are used for feature-down sampling.

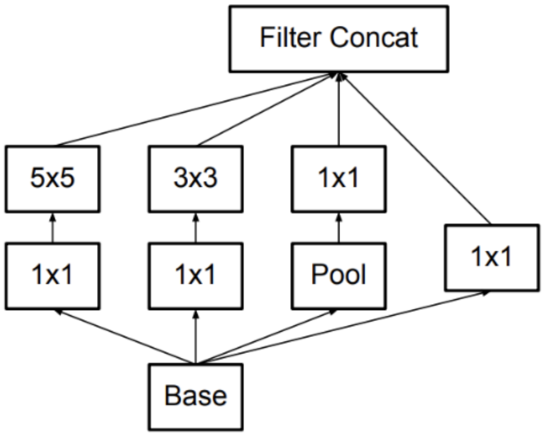


Fig. 3. The inception module.

C. Multiscale VGG

VGG net has a simple structure of a sequence of CNN blocks. Each block includes several convolution layers with different filters with a fixed size, followed by a max polling layer to reduce the feature size by 50% compared with the previous block. Fig. 4 and Fig. 5 show the proposed multiscale VGG architecture to recognize currency. The proposed model improved the VGG model architecture by implementing a multiscale VGG net to recognize Jordanian currency. The multiscale VGG net captures features with different respective fields compared with VGG architecture. Using multiple scales images, the last max-pooling layers of VGG extract four scales of feature map 7×7, 5×5, 3×3, and 1×1. The proposed multiscale VGG improves recognition rates by handling the variation of captured image sizes. The model can be applied to other currencies. The model learns text, color, security patterns and symbol features that exist in all other currencies. The trained model can also be fine-tuned and trained one other currencies, which can reduce the training time. The proposed dataset includes images with 448×448 size; the images are cropped from the center into 224×224. The input image with 224×224 size is resized into four scales: scale 1: 100%, scale 2: 75%, scale 3: 50%, and scale 4: 25% and passed into parallel four VGG nets. The outputs of the four scales are flattened and concatenated, then passed into a fully connected layer of 256 size, followed by a 50% dropout layer, and finally, a fully connected layer with size 7 (number of classes). The four scales capture more fine-grained local features (e.g., edges or textures) compared with one scale feature. The final extracted feature before the first flattened layer encodes the spatial and semantic and includes the most informative features for classification. The sizes of extracted features from the last block of convolutions are 7×7 for scale 1, 5×5 for scale 2, 3×3 for scale three and 1×1 for scale 4.

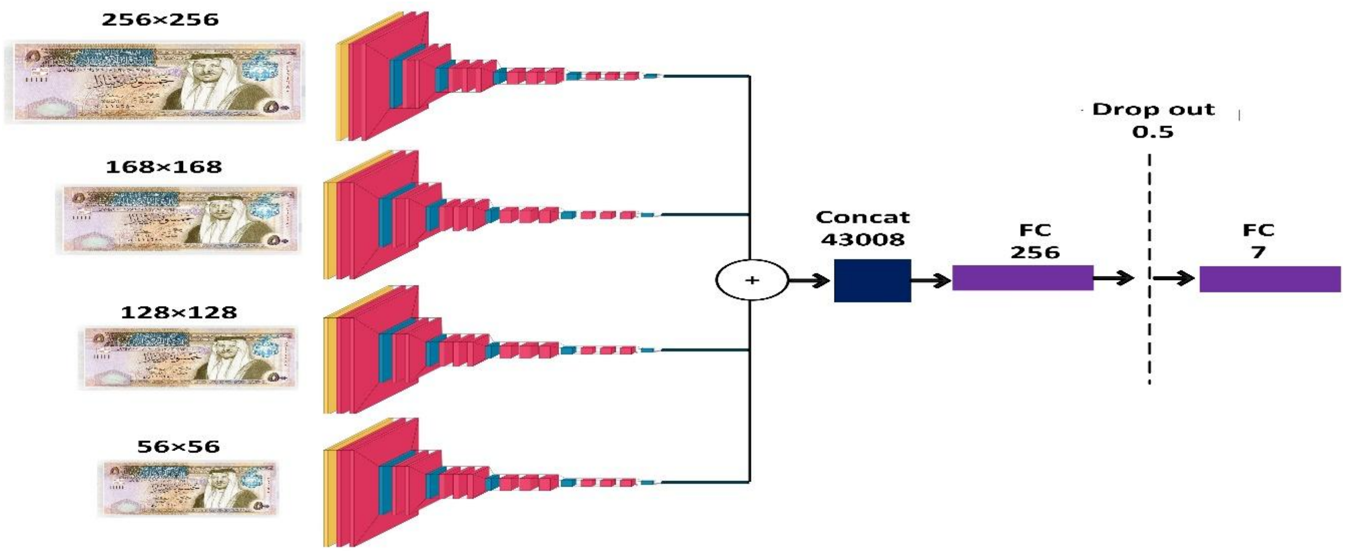
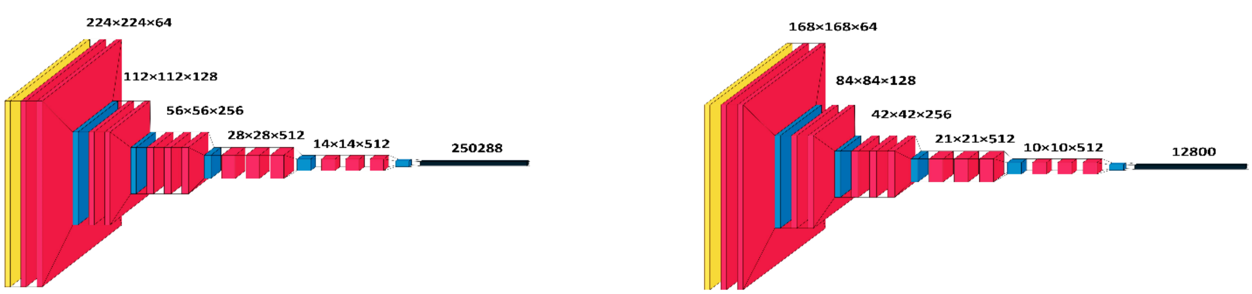


Fig. 4. The architecture of the proposed automatic multiscale currency recognition system.



(a) (b)

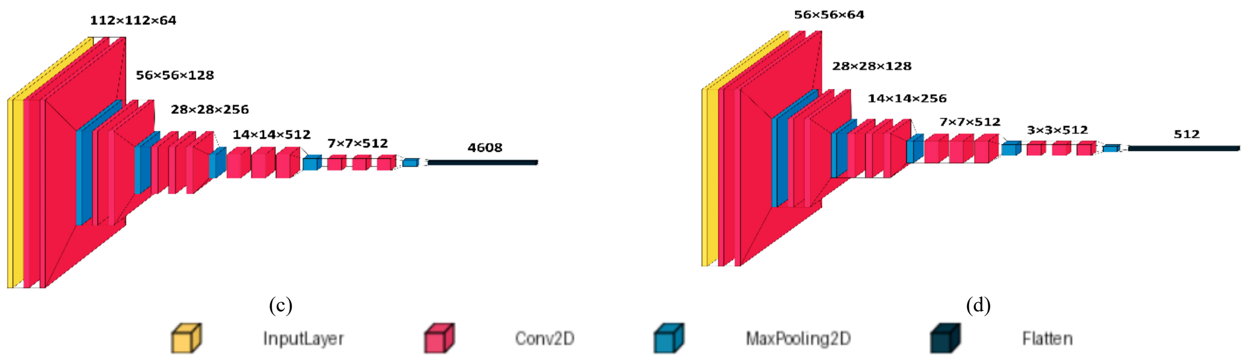


Fig. 5. Four scales of VGG16 architecture and the input images are (a) 224×224 (b) 168×168 (c) 112×112 (d) 56×56.

IV. Implementation Details

A. Experimental Environment

The experiments were done in a Windows 10 and Jupyter Notebook environment. The processor is an Intel(R) Core (TM) i5-8600K CPU @ 3.60GHz. The installed RAM is 40.0 GB, and the GPU is a GTX1080 with 8 GB memory.

B. Model Parameter Settings

Table IV shows the parameters that were used to perform data augmentation while training the model.

TABLE IV. Data Augmentation Used During Training

Augmentation Parameter	Value	Description
Rotation Range	20	up to 20 degrees
Width Shift Range	0.2	Horizontal shift by up to 20% of the image width
Height Shift Range	0.2	Vertical shift by up to 20% of the image height
Shear Range	0.2	up to 20%
Zoom Range	0.2	Zoom in or out by up to 20%
Horizontal Flip	TRUE	Randomly flips the image horizontally

C. Performance Metrics

Four metrics were used to evaluate the performance of the different CNN models in the dataset.

Accuracy: It measures the total of correctly predicted images compared to the total number of tested images.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \tag{3}$$

Precision: The percentage of true positive compared with all positive prediction; high precision indicates fewer false positive

$$\text{Precision} = \frac{TP}{TP+FP} \times 100 \tag{4}$$

Sensitivity (Recall): It measures the model's capacity to detect every real positive case. Therefore, the ratio of the true positives to the total of the true positives and false negatives is used to compute it.

$$\text{Sensitivity} = \frac{TP}{TP+FN} \times 100 \tag{5}$$

Specificity: It calculates the proportion of accurately categorized negative events to all negative instances that really occurred:

$$\text{Specificity} = \frac{TN}{TN+FP} \times 100 \tag{6}$$

where TP denotes True Positive, TN is True negative, FP is False Positive and FN represent False negative.

D. Experimental Results

This section shows the performance of the proposed Multiscale VGG net and compares it with the VGG16, DenseNet12, ResNet50, ResNet101, and InceptionV3. Four metrics are used in the evaluation: accuracy, precision, sensitivity, and specificity. 70% of the dataset samples were used in training the model, whereas 30% were used to test the model. As shown in Table V, using multiple scales of the VGG16 improved the recognition rates compared with the base model VGG16. VGG16 achieved 98.92% accuracy, whereas adding features from multiple scales improved accuracy in all combinations. The highest accuracy occurred when combining features from scale 1 and scale 0.75, where the accuracy reached 99.88%, with a 0.96% improvement compared with VGG16. The precision, sensitivity, and specificity reached 99.89%, 99.89%, and 99.98%, respectively.

TABLE V. Performance Evaluation of Proposed Multiscale VGG at Different Scale Values 100%, 75%, 50% and 25%

Models	Accuracy	Precision	Sensitivity	Specificity
VGG16	98.92%	98.96%	98.90%	99.82%
VGG16 SCALES 1,0.75,0.5,0.25	99.64%	99.66%	99.64%	99.94%
VGG16 SCALES 1,0.75	**99.88%**	**99.89%**	**99.88%**	**99.98%**
VGG16 SCALES 1,0.75,0.5	**99.88%**	**99.88%**	**99.89%**	**99.98%**
VGG16 SCALES 1,0.5,0.25	99.76%	99.78%	99.76%	99.96%
VGG16 SCALES 1,0.5	99.76%	99.78%	99.75%	99.96%

Table VI shows that the proposed method and InceptionV3 achieved 99.88% accuracy, after DenseNet121 with 99.52% accuracy, followed by ResNet101with 99.28% accuracy. The precision, sensitivity, and specificity metrics showed that the proposed multiscale and InceptionV3 achieved the best result, followed by DenseNet121 and ResNet101, respectively. Table VII shows the inference time and the model parameters. The VGG16 SCALES 1,0.75 runs in real-time and requires 200 ms to predict the class label for a patch of 32 images. At the same time, the model used 39,130,759 parameters.

TABLE VI. PERFORMANCE EVALUATION OF PROPOSED MULTISCALE VGG WITH OTHER CNN KNOWN CNN ARCHITECTURES

Models	Accuracy	Precision	Sensitivity	Specificity
VGG16	98.92%	98.96%	98.90%	99.82%
VGG16 SCALES 1,0.75	99.88%	99.88%	99.89%	99.98%
DenseNet121	99.52%	99.52%	99.52%	99.92%
ResNet50	99.16%	99.18%	99.17%	99.86%
ResNet101	99.28%	99.31%	99.30%	99.88%
InceptionV3	99.88%	99.89%	99.88%	99.98%

TABLE VII. INFERENCE TIME IN TERMS OF MS PER PATCH (PATCH=32 IMAGES) AND NUMBER OF PARAMETERS USED BY THE MODEL

Models	MS/Patch	#Parameters
VGG16	81	21,139,271
VGG16 SCALES 1,0.75	200	39,130,759
DenseNet121	78	19,884,615
ResNet101	146	68,350,343
InceptionV3	97	22,329,127

836 images were used to test the proposed architecture. The confusion matrix in Fig. 6 shows that the Multiscale VGG architecture accurately predicted 835 pictures, and only one image for 10 JD was predicted to be 50 JD.

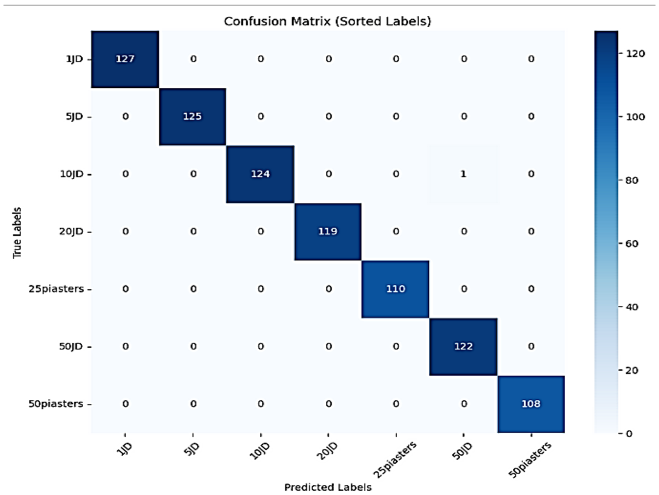


Fig. 6. Confusion matrix for the proposed multiscale VGG.

V. CONCLUSION

This paper proposed a dataset for Jordanian currencies and proposed Multiscale VGG architectures to recognize Jordanian currencies automatically. The dataset includes 2799 images for seven denominations (25piasters, 50piasters, 1JD, 5JD, 10JD, 20JD, 50JD). The images were captured in different conditions of lighting, backgrounds, clutter, occlusion, and orientation. Moreover, the experimental results showed that the proposed Multiscale VGG architectures achieved the best accuracy, precision, sensitivity, and specificity compared with VGG16, DenseNet121, ResNet50, and ResNet101.

REFERENCES

[1] D. S. Aljutaili, R. A. Almutlaq, S. A. Alharbi, and D. M. Ibrahim, "A Speeded up Robust Scale-Invariant Feature Transform Currency Recognition Algorithm," vol. 12, no. 6, pp. 346–351, 2018.

[2] P. Garkoti, P. Mishra, N. Rakesh, Payal, M. Kaur, and P. Nand, "Indian Currency Recognition System Using Image Processing Techniques," in 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), 2022, pp. 628–631.

[3] R. Bourne et al., "Trends in prevalence of blindness and distance and near vision impairment over 30 years: an analysis for the Global Burden of Disease Study," Lancet Glob. Heal., vol. 9, no. 2, pp. e130–e143, Feb. 2021.

[4] K. Sharifani and M. Amini, "Machine Learning: A Review of Methods and Applications," World Inf. Technol. Eng. J., vol. 10, no. 7, pp. 3897–3904, 2023.

[5] S. Salih and T. Nasih, "Image-Based Processing of Paper Currency Recognition and Fake Identification: A Review," Technium, vol. 3, no. 7, pp. 46–63, 2021.

[6] J. Lee, H. Hong, K. Kim, and K. Park, "A Survey on Banknote Recognition Methods by Various Sensors," Sensors, vol. 17, no. 2, p. 313, Feb. 2017.

[7] N. Dalal, B. Triggs, N. Dalal, and B. Triggs, "Histograms of Oriented Gradients for Human Detection To cite this version : Histograms of Oriented Gradients for Human Detection," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.

[8] A. Xu and G. Namit, "SURF: Speeded-Up Robust Features COMP 558-Project Report," pp. 2–29, 2008.

[9] D. G. Lowe, "Distinctive Image Features from Scale Invariant Keypoints," Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, 2004.

[10] D. Rika Widianita, "Combining Handcrafted and Deep Features For Scene Image Classification," J. Data Acquis. Process., vol. 38, no. 3, p. 2158, 2023.

[11] M. N. Abdi and M. Khemakhem, "Arabic writer identification and verification using template matching analysis of texture," Proc. - 2012 IEEE 12th Int. Conf. Comput. Inf. Technol. CIT 2012, pp. 592–597, 2012.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, vol. 39, no. 9, pp. 2261–2269.

[14] K. He, X. Zhang, S. Ren, and S. Jian, "Deep Residual Learning for Image Recognition," in In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[15] C. Szegedy et al., "Going deeper with convolutions," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.

[16] K. Reddy, G. Ramesh, C. Raghavendra, C. Sravani, M. Kaur, and R. Soujanya, "An Automated System for Indian Currency Classification and Detection using CNN," E3S Web Conf., vol. 430, p. 01077, Oct. 2023.

[17] M. Z. Hanif, W. A. Saputra, Y. H. Choo, and A. P. Yunus, "Rupiah Banknotes Detection : Comparison of The Faster R-CNN Algorithm and YOLOv5," J. INFOTEL, vol. 16, no. 3, pp. 502–517, 2024.

[18] C. Park and K. R. Park, "MBDM: Multinational Banknote Detecting Model for Assisting Visually Impaired People," Mathematics, vol. 11, no. 6, 2023.

[19] S. Alghyaline, "Optimised CNN Architectures for Handwritten Arabic Character Recognition," Comput. Mater. Contin., vol. 79, no. 3, pp. 4905–4924, 2024.

[20] E. AL-Edreesi and G. Al-Gaphari, "Real-time Yemeni Currency Detection," arXiv Prepr. arXiv2406.13034, 2024.

[21] I. O. A. Omeiza, O. Ogunbiyi, O. Y. Ogundepo, A. O. Otuoze, D. O. Egbune, and K. Osunsanya, “A Method of Colour-Histogram Matching for Nigerian Paper Currency Notes Classification,” Jordan J. Electr. Eng., vol. 9, no. 1, pp. 42–59, 2023.

[22] M. C. GENÇAL, “U.S. Banknotes Recognition By SURF Features,” 2nd Int. Congr. Innov. Technol. Eng., pp. 68–72, 2022.

[23] G. S. Hussein, S. Elseuofi, W. H. Dukhan, and A. H. Ali, “A Novel Method for Banknote Recognition Using a Combined Histogram of Oriented Gradients and Scale-Invariant Feature Transform,” Inf. Sci. Lett., vol. 12, no. 9, pp. 2121–2131, 2023.

[24] A. Mukundan, Y. M. Tsao, W. M. Cheng, F. C. Lin, and H. C. Wang, “Automatic Counterfeit Currency Detection Using a Novel Snapshot Hyperspectral Imaging Algorithm,” Sensors, vol. 23, no. 4, pp. 1–14, 2023.

[25] 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 Br., vol. 51, p. 109699, 2023.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” arXiv Prepr. arXiv1512.00567, 2015.

[27] C. Szegedy, Y. J. Wei Liu, P. Sermanet, V. Scott Reed, Dragomir Anguelov, Dumitru Erhan Vincent, and A. Rabinovich, “Going Deeper with Convolutions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.