A Lightweight CNN based Approach to Bangladeshi Paper Currency Recognition

Tazwar Mohammed Shoumik
tazwar.mohammed.shoumik@g.bracu.ac.bd

Shah Abul Hasnat Chowdhury
shah.abul.hasnat.chowdhury@g.bracu.ac.bd

Sifatul Amin
sifatul.amin@g.bracu.ac.bd

Oyshik Ahmed Aadi
oyshik.ahmed.aadi@g.bracu.ac.bd

Tanzim Mostafa
tanzim.mostafa@g.bracu.ac.bd

Sartaj Jamal Chowdhury
sartaj.jamal.chowdhury@g.bracu.ac.bd

Adib Muhammad Amit
adib.muhammad.amit@g.bracu.ac.bd

Md Humaion Kabir Mehedi
humaion.kabir.mehedi@g.bracu.ac.bd

Shadab Iqbal
shadab.iqbal@g.bracu.ac.bd

Annajiat Alim Rasel
annajiat@gmail.com

***Abstract*—Paper note currency is the most ubiquitous way of completing a transaction anywhere in the world and this applies to most third world countries such as Bangladesh. Hence, it is essential to recognize any of the currency within minutes for saving time as well as an uninterrupted transaction. The aim of this paper is to recognize the various bank notes that Bangladesh has to offer using CNN by utilizing the dataset at hand and update and augment the dataset if there it needs to be updated and augmented.**

***Index Terms*—Bengali Currency, CNN, Object Recognition, Dataset Enhancement**

I. Introduction

Paper currency is the monetary symbol for transaction in any country and Bangladesh is no exception. Throughout the decades, the paper currencies that they have has gone through many changes due to changes on many sides i.e. government, etc. It is essential that the banks and other establishments recognize all the paper notes that are in regulation in order to complete transactions. This technology has been tinkered with before but our aim is to see if we can push the boundaries in accuracy as well as update and augment the dataset that we have in our hands. The existing dataset is not updated (200 taka note) and does not contain all the paper currency data that are in regulation and there is not enough data for each bank note and we hope to mitigate that. Alongside the previously mentioned improvements, we are also looking to improve the accuracy even if it's a fraction worth.

In the following steps in this paper, we are going to be discussing the existing works on this specific topic, the methodology which includes the datasets and the methods and model we will be using to accomplish the task at hand. Then we discuss any future scopes in this particular field of interest and if any improvements can be implemented in this field of research in the future.

II. Existing Works

The classification of currency has been previously attempted before. Over time, a large number of researchers have played a role in currency recognition, incorporating various features such as color, texture, anti-spoofing features etc and with varying degrees of success.

In [1] and [5], a convolutional neural network was used and for [1] specifically, they tested their methods on every denomination of India with a 95% level of accuracy. That said, other forms of learning, such as Deep Learning or Transfer Learning, were also used such as that in [2] where the author also mentioned that most other methods often fail when it comes to damaged notes. In [3], K-NN based classification for the denomination was used for identifying banknotes after extracting color and texture features. Moreover, some papers were also concerned in the speed of recognition of these currencies such as that in [4], which had used YOLO V3 architecture for faster identification. As demonstrated by [6], currency notes can also be identified by using a unique identification mark. This mark can exist as certain shapes.

The paper [7] classified four currencies, namely, "pound", "yen", "euro" and "rupee" using Artificial Neural Network (ANN). For this they first preprocessed the images using a Median filter to denoise the images. In the feature extraction stage they used sobel masks and Canny edge detection to find out specific edges. Finally, they classified the currencies by training a feedforward backpropagation neural network with 100 samples of every currency type. Through this method, they achieved an average recognition rate of 93.84%. However, their system lacked high processing speed.

In the paper [8] the authors carried out experiments using three lightweight CNN models with Transfer Learning, the models being pre-trained on 'ImageNet' dataset'. They used ResNet152v2, MObileNet and NASNetMobile, to recognise

Bangladeshi notes from two different datasets. They achieved 98.88% accuracy on the 8000 image dataset using MobileNet and 100% on the 1970 image dataset using NASNetMobile. Finally, they combined the two datasets where they found that MobileNet performed the best, achieving 97.77% accuracy. According to their results, Transfer Learning helped greatly with the detection of notes. Nonetheless, their model struggled to recognise notes in case of complex background and arbitrary lighting.

The paper [9] used a pretrained Convolutional Neural Network (CNN), called AlexNet, to recognise Indian coins and classify them into 4 categories, namely, one, two, five and ten rupee coins. According to their results, only 1000 coin images of all instances are enough for training the model. Moreover, they proposed an architecture that involves high speed classification that is suitable for real-time requirements and they plan to deploy their algorithms in real time embedded systems.

The authors of the paper [10] performed analysis on the effect of orientation and region in the recognition of banknotes. They used various classification algorithms such as k-Nearest Neighbor (kNN), Decision Tree Classifier (DTC), Support Vector Machine (SVM), Bayesian Classifier (BC) as well as the CNN model AlexNet, with Malaysian Ringgit notes as their dataset. According to their results, DTC and kNN achieved 99.7% accuracy, while SVM and BC achieved 100% accuracy. They also concluded that while AlexNet performed well with 100% accuracy on a database of similar orientation, it performed below par in testing of notes with different orientation.

III. Methodology

This section demonstrates the procedures taken for acquiring Bangladeshi paper currency images and classifying them into their respective categories. We used 4 state-of-the-art CNN models to compare their validation accuracy and training time. These models are light enough to be implemented on IoT devices. Furthermore, the source, characteristics, and how the dataset was used for training, validation, and testing are provided in the subsections below.

A. Dataset Acquisition

The dataset we used is Kaggle's open-access dataset called "Bangla Money" [11] under Taka recognition datasets. It was created in 2018 for a project named "Bangla Taka Recognition" by Tahmida Akther, Tasnim Begum, and Mst. Fahmida Begum under supervision of Noushad Sojib for a final year project. It consists of nine target variables: Bangladeshi paper currency denominations of 1Tk, 2Tk, 5Tk, 10Tk, 20Tk, 50Tk, 100Tk, 500Tk, and 1000Tk.

Mobile devices of unknown models were used to capture a total of 1970 banknote images under different lighting conditions. Some of the banknotes in the dataset are worn out, old, and had images of both sides, which enabled the model to generalize the features of any class of banknotes, enabling it to correctly classify different paper currency denominations regardless of the note being worn out and old. Furthermore, some of the banknotes vary between the respective classes. For example, there are 2 different visual representations of 2Tk and 5Tk which further helps the model to learn the features of different banknotes within the same denomination.

Fig. 1. Nine different Bangladeshi Paper Currency Denomination

The dataset is divided into 80% training data(1637 images) and 20% for testing purposes(333 images). Each image was captured at various resolutions hence it was cropped and scaled down to a resolution of 120, 250, 3 which means the resulting image is an RGB image consisting of 3 color channels red, green, and blue. Each target class has around 100-250 images of training samples which the CNN models will be trained upon. Table 1 shows the number of training samples for each category of banknote. For example, the class of 1Tk has 102 training images and the class of 50Tk has 214 training images. For this inconsistency in training data of each target class, the overall training of feature extraction of images, in the successive convolution layers in the CNN model, will also be "learned" inconsistently across different classes which might result in wrong classification hence decreasing validation accuracy of the model. The primary reason for this is because the weights in any Deep Learning model, including CNN, are adjusted by back-propagation which depends on the value of the learning rate and the type of loss function used. The more the training data of a target class, the better the model can learn to classify that variable hence using this dataset, the model will learn to classify 50Tk with better overall accuracy compared to 1Tk.

B. Data Pre-Processing

Various methods were used on the Bangla Money dataset in order to account for the inconsistencies present which could result in overfitting of the models. Firstly we augment the dataset to increase the training samples of each currency denomination. Afterwards, we added a new target class which increased the number of total target variables from 9 to 10. Lastly, we split the augmented dataset into training, validating, and testing sets.

Target Variables	Training Samples
1Tk	101
2Tk	213
5Tk	213
10Tk	213
20Tk	173
50Tk	213
100Tk	208
500Tk	136
1000Tk	167

TABLE I
NUMBER OF TRAINING SAMPLES OF EACH CURRENCY DENOMINATION

1) Data Augmentation: Data augmentation is a set of techniques for producing additional data points from current data to artificially increase the amount of training data available. Making modest adjustments to data or utilizing deep learning models to produce additional data points are examples of this.

By generating new and varied instances to train datasets, data augmentation can help improve the performance and results of machine learning models. A machine learning model works better and more correctly when the dataset is rich and sufficient. Data collection and labeling can be time-consuming and costly for machine learning models. Companies can lower these operational costs by transforming datasets using data augmentation techniques.

As the Bangla Money dataset contains a significantly smaller number of training images compared to other popular datasets such as ImageNet(1,281,167 images), MNIST(60,000 images), and CIFAR-10(50,000 images), we perform data augmentation on the training images such that each class label has exactly 2000 images. Even though there are a lot of data augmentation techniques are out there, the techniques we used are as follows: rotation of 180 degrees and 90 degrees clockwise and anticlockwise, horizontal and vertical mirror flip. Fig. 2 illustrates the augmentation of a training sample of 20Tk banknotes.

Fig. 2. Data Augmentation of 20Tk Training Images

2) Addition of 200Tk: We added a new target variable to the existing dataset. This new target variable is the new paper currency of the 200Tk banknote of Bangladesh, released in early 2020. Fig. 3 shows a visual representation of both sides of a Bangladeshi 200Tk banknote. We manually take 100-150 photos of 200Tk notes with cameras from smartphones. Then we augment the images of this class up to 2000 images using the four data augmentation techniques as used earlier for other classes. This makes a total of 10 classes in our dataset, which equals the number of classes in the CIFAR-10 dataset. This addition brings novelty to the dataset and the research of classification of the paper currency of Bangladesh.

Fig. 3. Both sides of 200Tk Banknote of Bangladesh

3) Split Dataset into train, valid and test: Finally, after the addition of a new class, we split the resulting dataset into 80% training, 10% validation, and 10% testing samples. Previously the dataset was not divided into a validation set. The validation set is a different subset of the dataset used during training to see how well the model performs on data not used in training before deploying it.

C. Convolutional Neural Network

We used four lightweight CNN-based architectures for our research objective and compared their validation scores. The four architectures are as follows: EfficientNetV2B1 [12], MobileNetV3Large [13], DenseNet121 [14] and NASNetMobile [15]. All these models were trained on the ImageNet [16] dataset and achieved remarkable top-1 and top-5 accuracies, considering they have less than 10 million parameters. Table 2 shows the parameter count of each model. The weights of the pre-trained models are not suitable for classifying Bangladeshi banknotes. The models did not learn much about any paper currency, given that ImageNet does not have training samples of banknotes. As a result, we trained the models from scratch only on our augmented dataset, replacing any previous weights. Before training, we removed the output layer from all the models, which had 1000 nodes because ImageNet has 1000 target variables, and replaced them with layers of 10 nodes and softmax activation function since our dataset has ten target values.

1) EfficientNetV2B1: Introduced in 2021 at the 38th International Conference on Machine Learning, this model is the latest of all the other models used in this research. The

Model Name	Param(M)
EfficientNetV2B1	8.2
MobileNetV3Large	7.5
DenseNet121	8.1
NASNetMobile	5.3

TABLE II
PARAMETER COUNT OF EACH MODEL USED

EfficientNetV2B1 model is one of the members of the family of EfficientNetV2, which is an improved version over the EfficientNet [17] introduced in the 2019 ICML.

To develop the EfficientNetV2 model, a combination of training-aware neural architecture search and scaling was used in order to optimize training speed and parameter efficiency together. Unlike some previous works, the authors of this model propose an enhanced technique of progressive learning, where the network is trained with small-size images and weak regularization in the early epochs. With increase in epochs over time, the images are gradually increased in size, and regularization is increased. This dynamic approach can speed up the training without causing an accuracy drop. No layers were finetuned after training.

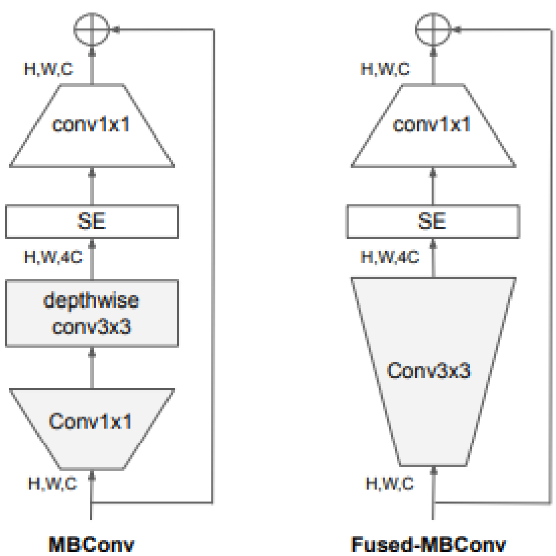

Fig. 4. Structure of MBConv and Fused-MBConv

The Fused-MBConv and MBConv block was proposed in [18] and [17] respectively. Fig. 4 shows the structure of both the blocks. During training in depthwise convolutions, EfficientNetV2 gradually replaces the MBConv block with Fused-MBConv. These two blocks need to be in correct proportions throughout the convolutions layers to attain maximum training efficiency. In order to do so, neural architecture search is used to automatically search for the best combination of both the blocks. Table 3 shows the architecture of EfficientNetV2.

2) MobileNetV3Large: The MobileNetV3 network was introduced at the ICCV 2019 as a successor to the MobileNetV2,

Stage	Operator	Stride	Channels	Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

TABLE III
PARAMETER COUNT OF EACH MODEL USED

presented at the CVPR 2018. The MobileNetV3 has two networks: MobileNetV3Large and MobileNetV3Small for high and low resource use cases, respectively. MobileNetV3 is engineered through novel architecture advances via a combination of hardware-aware network architecture search (NAS), which is complemented by the NetAdapt algorithm. MobileNetV3Large is 20% faster, in terms of training, and 3.2% more accurate on ImageNet classification compared to MobileNetV2. The architecture of MobileNetV3 is shown in the Fig. 5.

Input	Operator	exp size	$\#out$	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Fig. 5. Layered Architecture of MobileNetV3Large

The goal of this model is to improve validation accuracy while reducing training time and minimizing the number of parameters that can be implemented on mobile devices. To achieve this, it uses better complementary search techniques, efficient nonlinear practical versions for mobile setting, and a better-structured network design and segmentation decoder. MobileNetV3 uses a combination of blocks from MobileNetV2 [19] and MnasNet [20], which are the "Inverted Residual and Linear Bottleneck" and the "Squeeze-and-Excite" respectively. A structural difference between MobileNetV2 and MobileNetV3 is shown in Fig. 6.

3) DenseNet121: The DenseNet architecture was introduced at the Conference on Computer Vision and Pattern

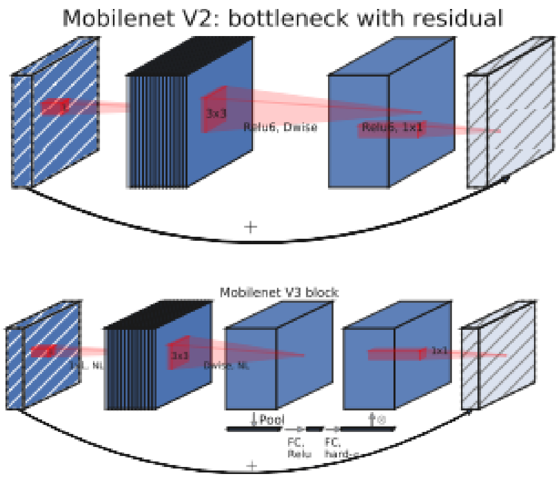

Fig. 6. Blocks of MobileNetV2 and MobileNetV3

2017. DenseNet consists of 4 individual networks of different number of layers and parameters: DenseNet-121, DenseNet-169, DenseNet-201 and DenseNet-264. The main motivation of DenseNet is to ensure maximum information flow between subsequent layers of the network by having direct connections from one layer to every other layer inside a dense block. A visual representation of a dense block is shown below in Fig. 7.

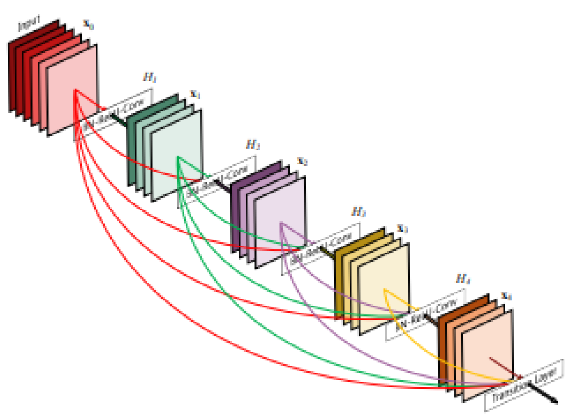

Fig. 7. A 5-layer dense block

Multiple dense blocks are infused into the network separated by transition layers via convolution and pooling. Fig. 8 shows a visual representation of a fully-fledged DenseNet architecture. In a dense block, each layer has access to all the preceding feature maps, and therefore, this concept extends to all the dense blocks hence, the entire network's "combined knowledge" enhances. The kernel convolution and downsampling of images happen inside the transition layers via a batch normalization layer, a 1×1 convolutional layer followed by a 2×2 average pooling layer.

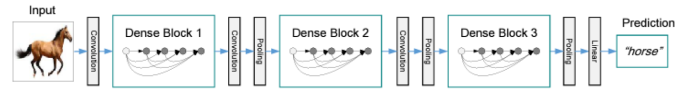

Fig. 8. A DenseNet Architecture with three 5-layer dense blocks

4) NASNetMobile: The paper for NASNet was published at the Conference on Computer Vision and Pattern Recognition in 2018. The key distinguishing property of NASNet is the introduction of "NASNet Search Space," which allows transpotability from dataset to dataset. With this model, a new regularization technique, ScheduledDropPath, is also introduced which greatly enhances the generalization when learning from a dataset. NASNet has two model: NASNet-Large and NASNetMobile for high and low resource use case respectively. This approach was inspired by LSTMs [21] and the Neural Architecture Search (NAS) framework [22], which uses a reinforcement Learning search technique for better optimization.

When taking in a feature map, two types of convolutional cells are employed to generate scalable architectures for input images of any size: Normal Cell and Reduction Cell. The Normal Cell is a convolutional cell that returns a feature map of the same dimension, while the Reduction Cell is a convolutional cell that returns a feature map by dividing the dimensions of the original feature map by two. Fig. 9 and 10 shows the NASNet architecture used for ImageNet and an individual block of a Normal Cell and Reduction Cell respectively.

IV. Conclusion

In this paper we aim to come up with an efficient method to classify the Bangladeshi paper currency. There is a tremendous scope for implementing this technology in ATMs, in the aid of visually challenged individuals, and help detect fake paper currencies. We experiment with an existing dataset, which we enhance by augmenting the data and adding extra class to make the dataset up to date. Furthermore, we use a CNN based model in order to recognise the bank notes. We use a specific method to evaluate the model. Finally, we suggest improvements.

In the long run, a prototype to identify the Bangladeshi currency notes could be developed further from this so that it can be useful to visually impaired persons in the future. Another extension of this work could be for the detection of Bangladeshi fake notes. Moreover, further expanding the dataset can help improve accuracy. Evaluation techniques involving testing on unique or other foreign currency to measure accuracy can also be utilized. Lastly, there is always some room to improve the processing time of the method used.

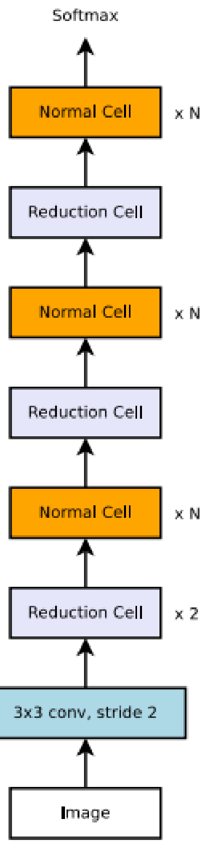

Fig. 9. A NASNet Architecture used for ImageNet Classification

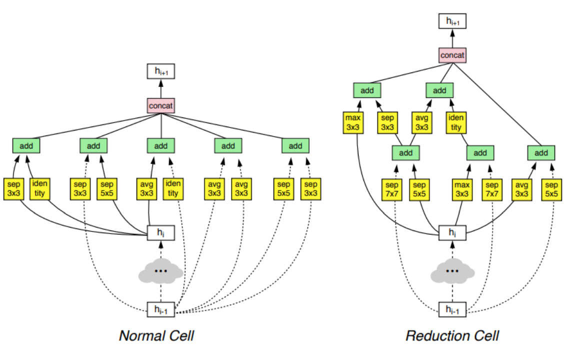

Fig. 10. An in-depth view of Normal Cell and Reduction Cell

References

[1] Rushikesh Jadhav, Swaraj Kalbande, Rohan Katkar, Rohan Katta, Prof. Rakhi Bharadwaj in ”Currency Recognition using Machine Learning”, Jan 1, 2022 vol 9, issue 1.

[2] ”Suyash Mahesh Bahrani” in ”Deep Learning approach for Indian currency classification”, Oct 2020, Vol. 5, Issue 6, ISSN No. 2455-2143, Pages 335-340

[3] ”Uttoran Roy Chowdhury, Susovan Jana, Ranjan Parekh” in ”Automated System for Indian Banknote Recognition using Image Processing and Deep Learning”

[4] ”Wafia Rarani, Vrushabh Rode, Chaitali Mahatme, Dhiraj Chavhan, Prof. Kalyani Gholap” in ”Indian Currency Note Recognition System using YOLO v3 Methodology”, Jan 2021, Volume 8, Issue 1.

[5] ”Tedele Elsai” in “Developing Ethiopian Paper Currency Recognition Model Using Convolutional Neural Network (CNN)”

[6] ”Syed Ejaz Ali, Mriganka Gogoi and Subra Mukherjee” in ”Artifical Neural Network To Recognize An Indian Currency Note Using Unique Identification Mark”

[7] Jyothi, C. R., SundaraKrishna, Y. K., SrinivasaRao, V. (2016, March). Paper currency recognition for color images based on Artificial Neural Network. In 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) (pp. 3402-3406). IEEE.

[8] Linkon, A. H. M., Labib, M. M., Bappy, F. H., Sarker, S., Jannat, M. E., Islam, M. S. (2020, December). Deep Learning Approach Combining Lightweight CNN Architecture with Transfer Learning: An Automatic Approach for the Detection and Recognition of Bangladeshi Banknotes. In 2020 11th International Conference on Electrical and Computer Engineering (ICECE) (pp. 214-217). IEEE.

[9] Tajane, A. U., Patil, J. M., Shahane, A. S., Dhulekar, P. A., Gandhe, S. T., Phade, G. M. (2018, February). Deep learning based indian currency coin recognition. In 2018 International Conference On Advances in Communication and Computing Technology (ICACCT) (pp. 130-134). IEEE.

[10] Sufri, N. A. J., Rahmad, N. A., Ghazali, N. F., Shahar, N., As’Ari, M. A. (2019, August). Vision based system for banknote recognition using different machine learning and deep learning approach. In 2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC) (pp. 5-8). IEEE.

[11] Bangla Money. https://www.kaggle.com/datasets/nsojib/bangla-money

[12] Tan, M., Le, Q. (2021, July). Efficientnetv2: Smaller models and faster training. In International Conference on Machine Learning (pp. 10096-10106). PMLR.

[13] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... Adam, H. (2019). Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 1314-1324).

[14] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[15] Zoph, B., Vasudevan, V., Shlens, J., Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8697-8710).

[16] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

[17] Tan, M., Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

[18] EfficientNet-EdgeTPU: Creating Accelerator-Optimized Neural Networks with AutoML. https://ai.googleblog.com/2019/08/efficientnet-edgetpu-creating.html

[19] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[20] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q. V. (2019). Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2820-2828).

[21] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[22] Zoph, B., Le, Q. V. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.