Indian Currency Detection and Value Recognition Model

Saumya Gurnani
AI&DS Department
IGDTUW
New Delhi, India
saumyagurnani73@gmail.com

Shambhavi Tripathi
AI&DS Department
IGDTUW
New Delhi, India
shambhavi2177@gmail.com

Sejal Yadav
AI&DS Department
IGDTUW
New Delhi, India
Sejalyadavclg@gmail.com

***Abstract*— Despite the increase in cashless transactions recently, paper currency remains an important aspect of financial exchanges. Although digital payments are convenient, they are prone to errors from unsafe gateways, poor network connections or insufficient funds. On the contrary, cash payments are always successful with little to no risk of failure due to their physical nature. The paper presents a comparative study on two transfer learning models—VGG16 and ResNet-18, for recognizing and detecting values of Indian Currency Notes. The model is intended to successfully detect and classify the various denominations of Indian currency. The VGG16 model achieves 91.9% training accuracy and 87.71% testing accuracy, while the ResNet-18 model performs better with 92.66% training accuracy and 93.02% testing accuracy. For successful classification, the models run through several steps including: acquiring the data, pre-processing it, data augmentation, extracting the features and finally classification which ensures that input data is consistent and also increases the diverseness of the training data which impacts the model's generalization and robustness. The implementation of the proposed model can significantly improve the currency detection and verification process in the banking and financial sectors of the country. Furthermore, the proposed models have the potential to be expanded into a fake currency detection model. The development and deployment of such detection systems is important to reduce and prevent fraudulent activities and guarantee secure and safe transactions within the country.**

Keywords—Currency Recognition, CNN, Transferred Learning, VGG16, ResNet-18

I. Introduction

In recent years, there has been a global urge towards digitalization. With the ever-changing tech landscape, the financial sector too has been revolutionized with the shift from paper currency to paperless transactions through QR codes, net banking and credit and debit cards. Although this shift has made transactions easier and convenient, digital payments are not always reliable as they can fail due to network issues, insufficient funds or technical errors. Consequently, cash transactions remain relevant and a dependable option for financial exchanges especially in areas with limited or no digital infrastructure.

Financial fraud has always plagued the country, underscoring the need for robust and secure systems to prevent and reduce financial crimes throughout the country. The counterfeit currency pose a threat to the economic stability through inflation and reduced purchasing power. Moreover, there is a need for faster and more accurate currency processing in retail, banking and ATMs, which emphasize the importance of reliable, automated solutions. Automated cash handling will ease the process of paper transactions by reducing human error and speeding up processes. Implementing and installing a reliable currency detection system, will help prevent fraud and ensure secure financial operations. These challenges highlight the need for an efficient and automated solution for Indian Currency Recognition and value detection.

Although, there are automated cash detection systems in place, the diverse and intricate designs of Indian Currency, including variations in colour, size, texture and security features, present a significant challenge for automated detection systems. Due to this, the systems are considered are ineffective and unreliable to detect currency with maximum accuracy.

To counter this, robust and efficient Indian currency recognition models are required. The models which enable precise recognition by identifying unique patterns, textures and symbols, even in challenging conditions like poor lighting or damaged notes.

To address these issues, the research focusses on utilizing deep learning techniques, viz transfer learning by particularly employing and comparing VGG16 and ResNet-18 architecture. The study evaluates their performance in recognizing Indian currency denominations accurately. It also provides insights into the strengths and limitations of these models, particularly in handling Indian currency. These insights gained contribute to the development of a system capable of overcoming challenges associated with Indian currency's complex designs and counterfeit detection.

II. Literature Review

Dande et al. [1] presented the utilization of YOLOv5 model for Indian currency notes detection. YOLOv5 is a computer vision model that can be used for object detection, segmentation and classification. The images data comprised 200 images per class, which are captured on different backgrounds and different visual settings. By applying data augmentation 1400 images were transformed to 5600 images. As per usual, the division of the dataset takes place in the ratio 70:30 where the first seventy goes for training the model and the next thirty goes for testing the model. The evaluation metrics used were precision, recall and mAP. They obtained an accuracy of 85%. The future scope of this project lies in extending it to other currency notes and using Optical Character recognition to improve the model performance.

Rao et al. [2] performed transfer learning using ResNet50V2. It is a pre-trained deep learning model. The dataset included 50 photos of each currency which were downloaded using Bing Image Downloader API. It involved fine-tuning the model for performing the required task. The various layers in the model were GlobalAveragePooling 2D layer, Dense layer having 256 neurons and dropout layer. The final output layer was made by a dense layer which had 30 neurons because there were 30 classification classes. The optimizer used was Adam optimizer and loss function was Categorical Crossentropy. The model training consisted 100 epochs with early stopping. The resultant accuracy is 96.76%. They aim to improve the model in future by expanding the scope of its usage to damaged and counterfeit currency.

Pujar [3] proposed the employment of feature extraction using digital image processing and MATLAB for currency classification. The system involved two prominent steps mainly, currency recognition and currency verification. Initial stage was the preprocessing of images which included correction of distortion, degradation and noise. The recognition was done by image segmentation, edge detection, feature extraction and comparison.

Jamtsho et al. [4] investigated denomination recognition for Bhutanese currency using transfer learning. They performed a comparative analysis of the 3 models which were VGG16, ResNet-50 and InceptionV3. The dataset of 1000 images were acquired by crowdsourcing. They reached the conclusion that out of the 3 models, the model based on VGG16 has the maximum accuracy of 99.12%.

Nigam et al. [5] proposed the use of pre-trained models like faster RCNN and Mobile-Net SSDV-2 for developing a voice enabled application that could help the visually impaired in recognizing Indian Currency in English and Hindi. The dataset used consisted of 800 Indian currency notes. They pre-processed the data by marking images, resizing, adjusting brightness, reducing noise and data augmentation. It was concluded that faster RCNN model has a higher accuracy as compared to Mobile-Net SSDV-2 model.

Laavanya et al. [6] demonstrated the implementation of AlexNet for detecting fake currency notes. Transfer learning is used for training the model. The model is trained with a self-made dataset. The augmentation processes like rotating and resizing were performed on the database to prevent overfitting by improving the diversity of data. Transfer learning was preferred over building a deep neural network from scratch because of the limited availability of dataset. The accuracy of this model was 81.50%.

Vincent et al. [7] developed a method for recognizing and verifying Indian Currency using Transfer Learning . They implemented a system using transfer learning on AlexNet using MATLAB. The system scans a currency note using a mobile phone and returns its denomination. Further, it also detects if it is a fake note or not. In the system, a note was considered real if the security features like Gandhi watermark and thread were visible under light and rest all the images were considered fake. They used a very small dataset containing 80 images. However, through the use of AlexNet, they were able to get 93% accuracy.

Padmaja et al. [8] focussed on the development of a three-layer CNN model for predicting Indian currency denomination and identifying fake notes. The dataset consisted of 4002 images divided into 7 classes based on denominations. The preprocessing involved converting the images into arrays of size 100X200X3. First 2 dimensions are for height and width respectively and the 3rd dimension is for the RGB pixel values. They normalized the pixel values so that they lie in the 0-1 range. The CNN used Rectified Linear Unit (ReLu) activation function and 2 Max Pooling layers. SoftMax activation is used on the output layers to assign the denominations.

Kumar et al. [9] formulated an approach for detecting fake Indian currency notes using a 3 layer CNN. A self-curated dataset was used. The proposed algorithm used edge detection with image segmentation and filtering. Data was pre-processed using data augmentation, image smoothening and noise removal. The model is made more reliable by feeding currency characteristics. The accuracy achieved with this method is 96.6%

Patil et al. [10] developed a unique approach to fake currency detection. They developed an android application empowering the visually impaired with image processing and deep learning techniques. The image dataset was acquired through various sources. Pre-processing involves enhancing the quality of image, image resizing, contrast adjustment, image normalization and conversion to Gray scale for ease in feature extraction. The intensity of Region of Interest is calculated. If the calculated intensity lies in the range 75% to 100 % then the currency is real, else it is fake.

From the research papers analyzed, we conclude that different models like VOLO V5, InceptionV3, CNN and many others have been employed for currency note classification and fake note detection. A wide range of accuracies were achieved from 81.5% to 99.12%. This is a growing scope of study where different currency notes are being used as the data for classification

III. METHODOLOGY

We aim to classify the Indian currency notes of various denominations efficiently by developing a deep learning model based on the pretrained models VGG-16 and Resnet18. We further aim to compare the results obtained by using the 2 models and identify the better one. We have used transfer learning due to the availability of limited data for currency classification. Transfer Learning is a crucial technique used in machine learning. In this the knowledge which is gained using one dataset is used to improve the predictions made using some other dataset.

14 million high-resolution images from various classes make up the ImageNet dataset, which serves as the foundation for the VGG-16 model. There are sixteen layers in all in VGG16. These 16 layers consist of 3 fully connected layers and 13 convolutional layers [11]. Because it includes 16 learnable parameter layers, it is known as VGG16. These layers are separated into blocks. Every block has a pooling layer after a number of convolutional layers. In the VGG16 network, we see that the input layer is the initial layer. The input has the following dimensions: 224,224,3. A 3X3 filter with stride of 1 is used by all convolutional layers, while a 2X2 filters with stride of 2 is used by the max pooling layers.

Conv1 layer has 64 filters, conv2 has 128, conv3 has 256, conv 4 and conv 5 has 512 filters. After these convolutional layers, three fully connected layers are joined. As it performs a classification into thousand classes, the third one has 1000 channels, whereas the previous two have 4096 channels [12].
Figure 1 depicts the structure of the VGG 16 model.

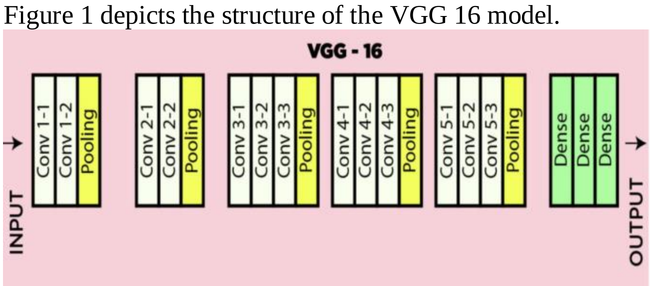

Figure 1: Architecture of VGG16 model
Source:[13]

Resnet 18 is another deep CNN trained on the imageNet dataset. It has 18 layers consisting of convolutional layers and residual blocks. It helps to prevent the problem of Vanishing Gradient. (224,224,3) is the size of the input. The first layer in ResNet-18 is a 7x7 convolutional layer which has 64 filters and stride of 2. This is followed by batch normalization and ReLu activation function. Next, a 3x3 max pooling layer with a stride of 2 is applied. The network then includes four groups of residual blocks, each with a unique number of filters. A global average pooling layer that creates a 1x1x512 feature map comes after the final residual block. Then at the end there is a fully connected layer which classifies the input images of ImageNet dataset into 1000 categories. Figure 2 shows the structure of Resnet18.

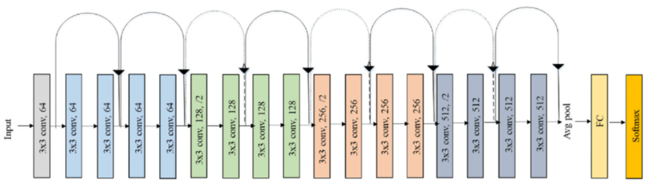

Figure 2: Architecture of RESNET18 model
Source:[14]

A. Data Collection

The dataset used for our investigation is the “IndianCurrency for classification” dataset available on Kaggle [15]. We researched for and tried different datasets but this dataset was the most extensive and robust among all. It consists of 2828 images of different denominations which include ten, twenty, fifty, hundred, two hundred, five hundred, two thousand. For reference: figure 2 depicts a sample of images from the datasets. It gives a clear idea of the type of data the model is trained on. 70% of the total data is used in training and the remaining 30% is utilized for testing the model.
However, the given dataset that we use for training the model has an unequal number of images for different denominations. This may lead to some minor bias while training the model.

Figure 3: 10 sample images from the dataset

B. Pre-Processing

The data collected from different resources are in raw form. Before the data is fed into the model, pre-processed.
Firstly, the images are labelled based on the names of the folders they belong to. All the images are resized to 224X224X3 by randomly cropping them. The goal is to convert the image size to the standard size of input for the VGG16 and Resnet 18 architecture.

Data augmentation is performed on the training data to improve the diversity of images and reduce overfitting. The images are flipped horizontally with 50% probability. Further, the input images are randomly rotated in the range of -90 degree to +90 degree.

The image is then converted into a PyTorch tensor. The dimensions of the image are rearranged to [C, H, W] format. Here, H stands for the height, while C represents channel and W means the width of the image.
The neural networks perform better when the pixel values are normalized. The range of pixel values is changed from [0,255] to [0,1] for improving the performance of the model. The normalization is performed using the mean and standard deviation of ImageNet dataset which is [0.485,0.456, 0.406] and [0.229, 0.224, 0.225] respectively. The three values represent to the 3 colour channels red, green and blue. The formula for normalization of each pixel is given in equation (1)

$$\text{Normalized } x = x\text{-mean/std} \qquad (1)$$

Here x is the original pixel value, mean is the mean of pixel values and std is the standard deviation of the pixel values. The data is also divided into batches with each batch containing 64 images. This makes the training process faster as images are processed in parallel. Finally, the data is shuffled which increases randomness and reduces overfitting.

Similarly, the images from the testing data are also resized to have a shorter side of 256 pixels and then centre cropped to obtain a final size of 224 X 224. They are converted to a PyTorch tensor and normalized to match the input distribution of the ImageNet dataset.

C. Model Development

The model uses pretrained models VGG 16 and Resnet18 from torchvision.models as the base models.
The process of Transfer Learning is used to develop the models for currency note classification. It uses a neural network which is trained on some other problem to solve the given problem. Transfer Learning helps to avoid building the model from scratch as the pre-trained models have already learnt how to detect the generic features.
In the proposed models, parameter fine tuning is utilized to increase the prediction accuracy. In this process, we do not train the already trained models, we rather perform the network surgery.

Training a pre-trained model from scratch will remove all the learned features and randomly distribute the weights.
Instead, the head of the network from the original model is removed and a new head which means new fully connected layers are added at the top of the architecture. This is called network surgery. In the pre-trained models, the convolutional layers are already well trained and have learnt the discriminative features while the fully connected layers are new and do not know the optimum weights. Therefore, to avoid destroying the already learnt features, the convolutional layers are frozen. The fully connected layers are trained for a few epochs.

The fully connected classifier is replaced with a custom sequential layer. The sequential layer is subdivided into 3 parts. The first part is a singular layer also known as the input layer, followed by hidden layers, total three in numbers and finally an output layer. ReLu activation function and a dropout layer is added after each hidden layer and the input layer. There is a gradual reduction of the number of neurons in the order 512, 256, 128, 64 and 7. The ReLu activation function introduces non-linearity for complex pattern learning. The dropout layer regularizes the model to prevent overfitting. The output layer has 7 neurons as there are 7 currency denominations for classification.

D. Compiling the Model

To compile the model, we utilize the Adam optimizer which has a learning rate set at 0.001 for adjusting the weights. We employ the Categorical Crossentropy loss function, specifically tailored for multi-class tasks. The aim is to reduce the inconsistency between the class probabilities that are predicted and the actual class labels. .

E. Justification of the methodology

The utilization of VGG16 and Resnet18 ensures that the model can benefit from the learned feature representations like edges, textures and shapes which are beneficial for computer vision tasks. The use of transfer learning techniques proved to be beneficial as in our case there was a small dataset and a model starting from scratch would have faced difficulty in learning.

We implemented the ReLu activation function in our model. The use of ReLu aids in adding non-linearity to the model [16]. Incorporating dropout layers contributes to avoiding overfitting.

The adam optimizer is used because it can calculate individual adaptive learning rate for each parameter based on historical magnitude of their gradients [17]. Therefore, it performs well even with sparse or noisy gradients. Moreover, adam is effective in models involving fine tuning because its adaptive learning rates reduces the risk of over adjusting the pre-trained weights and improves the stability during training.
For this model, the loss function chosen is the Categorical Cross entropy as it is suitable in classification problems that usually involve multiple classes.

F. Potential Limitations and Biases

One of the potential problems in using VGG16 and Resnet 18 could be the domain specificity of the architecture. Both of these models are pretrained on the ImageNet dataset. However, the dataset used in our model is the currency dataset which is significantly different from the ImageNet dataset. This could be the cause of decrease in the efficiency of the model.

Although a dropout layer is added, there is still a risk of overfitting due to the limited dataset. This could lead to poor generalization and the model may not be able to classify images correctly if they are significantly different from the training images. Moreover, the dataset used for training the model has unequal number of images for different classes. This can lead to introducing bias towards some particular classes during training.

IV. RESULTS

Taking VGG16 as the base model, the training accuracy is 91.90% and the testing accuracy is 87.71%. The final training loss is 0.25.

Further, prediction of the denomination was made for 6 random images from the test data. It correctly predicts all the images.
Figure 4 represents the 6 images from the test data and their predictions.

Figure 4: 6 sample images and their predictions using model based on VGG16

The confusion matrix was plotted for the model based on VGG16 and is shown in figure 5.

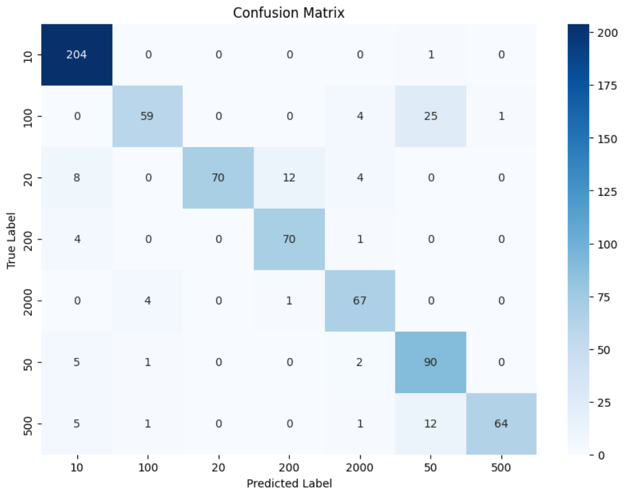

Figure 5: Confusion Matrix for VGG16 based model

Table 1 represents the classification report obtained using the VGG model.

Table 1. Classification Report for the VGG16 model

Label	precision	recall	f1-score	support
10	0.93	1.00	0.96	205
100	0.74	0.81	0.77	89
20	0.93	0.86	0.90	94
200	0.88	0.95	0.91	75
2000	0.80	0.90	0.85	72
50	0.83	0.80	0.81	98
500	1.00	0.67	0.81	83
accuracy			0.88	716
macro avg	0.87	0.86	0.86	716
weighted avg	0.88	0.88	0.88	716

Taking Resnet18 as the base model, the training accuracy is 92.66% and testing accuracy is 93.02%. The final training loss is 0.24.

Further, on making the prediction for 6 random images, the model correctly predicts the denominations of all the 6 images.
Figure 6 represents the 6 test images and their predictions.

Figure 6: 6 sample images and their predictions using model based on Resnet 18

The confusion matrix obtained using the Resnet 18 based model was plotted and is shown in Figure 7.

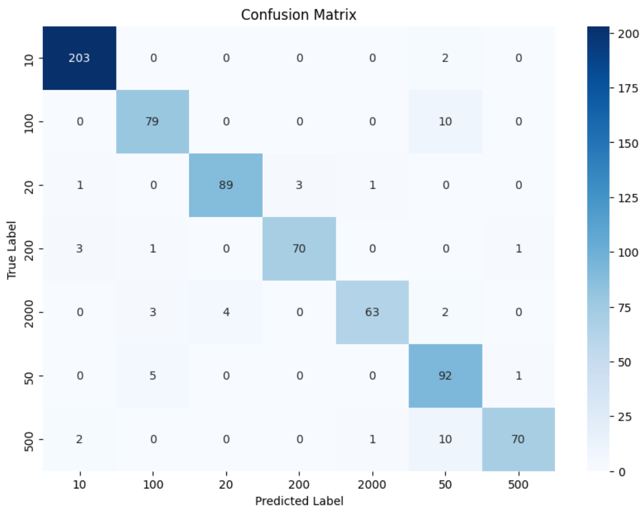

Figure 7: Confusion Matrix for Resnet18 based model

Table 2 represents the classification report obtained using the Resnet18 model.

Table 2. Classification Report for the Resnet 18 model

Label	precision	recall	f1-score	support

10	0.97	0.99	0.98	205
100	0.90	0.89	0.89	89
20	0.96	0.95	0.95	94
200	0.96	0.93	0.95	75
2000	0.97	0.88	0.92	72
50	0.79	0.94	0.86	98
500	0.97	0.84	0.90	83
accuracy			0.93	716
macro avg	0.93	0.92	0.92	716
weighted avg	0.93	0.93	0.93	716

V. Discussions

The confusion matrix for the VGG16 model in figure 5 represents that the model has accuracy less that the Resnet 18 model. However, the accuracy is fairly good, with the majority of predictions being correct, as shown by the diagonal elements.

204 instances of 10 rupees, 59 instances of 100 rupees, 70 instances of rupees 20, 70 instances of rupees 200, 67 instances of rupees 2000, 90 instances of rupees 50 and 64 instances of rupees 500 currency notes are predicted correctly. Rest of the images are predicted incorrectly. This misclassification among classes stemmed from shared resemblances in their features.

The classification report for VGG166 given in table 1 depicts that most classes display precision and recall values below 90%, suggesting that the model exhibited overfitting tendencies during testing.

The confusion matrix for the Resnet 18 model in Figure 7 represents that the model has high accuracy, with the majority of the predictions being correct, as shown by the diagonal elements. 203 instances of 10 rupees, 79 instances of 100 rupees, 89 instances of rupees 20, 70 instances of rupees 200, 63 instances of rupees 2000, 92 instances of rupees 50 and 70 instances of rupees 500 currency notes are predicted correctly. Rest of the images are predicted incorrectly. This misclassification among classes stemmed from shared resemblances in their features.

The classification report for Resnet18 given in table 2. depicts that a majority of the classes exhibit a recall exceeding 90%, indicating minimal occurrences of false negatives. Furthermore, the precision values for each class are notably high, corresponding to a reduced incidence of false positives.

Therefore, the model based on Resnet18 clearly outperforms the model based on VGG16 as it achieves a testing accuracy of 93.02 % as compared to a testing accuracy of 87.71% for VGG16.

In the future, we aim to increase the accuracy of the models and expand the scope of this model to include the detection of counterfeit currency as well. There is a scope to integrate the model with a voice enabled web application which can be used by the visually impaired for making their currency related transactions easier.

VI. Conclusion

The research compared the performance of VGG16 and ResNet-18 models for Indian currency recognition and value detection. The Res-Net-18 model achieved superior results with a testing accuracy of 93.02%, outperforming VGG16, which attained a testing accuracy of 87.71%. Both the models successfully recognize and detect the 7 denominations of Indian Currency. The comparative analysis underline ResNet-18’s capability to handle the complex patterns and intricate design of the Indian currency effectively due to advanced learning framework. The insights emphasize on the importance of model architecture and the preprocessing techniques for improving recognition accuracy. The study also underscores the potential of deep learning-based systems to increase financial independence and secure financial transactions. The currency recognition models can further be developed into a counterfeit detection models, or can be integrated with a text to speech model to help the visually impaired in recognizing the currency, which will promote financial independence.

Acknowledgment

We would like to thank the Centre of Excellence-Artificial Intelligence and the Indira Gandhi Delhi Technical University for Women(IGDTUW) for providing us with this incredible opportunity to research on the Indian Currency Detection and Value Recognition Model. We would also like to express our kind gratitude to Mr. Govind Gupta, our instructor during the internship, for his support and guidance during the research. Also, extending our heartfelt gratitude to Dr. Ritu Rani for her constant encouragement and mentorship during the internship.

References

[1] Dande, S. A., Uppunuri, G. R., & Raghuvanshi, A. S. (2022). YOLOv5 BASED WEB APPLICATION FOR INDIAN CURRENCY NOTE DETECTION. *Int. Res. J. Eng. Technol*, 2801-2807.

[2] Rao, M. S. N. V. B., & Gowtham, N. S. (2024). Efficient currency recognition and value detection system using image processing. *World Journal of Advanced Research and Reviews*, *22*(1), 241-255.

[3] Pujar, M. R. (2018). Indian currency recognition and verification using image processing. *International Journal for Advance Research and Development*, *3*(2), 175-180.

[4] Pujar, M. R. (2018). Indian currency recognition and verification using image processing. *International Journal for Advance Research and Development*, *3*(2), 175-180.

[5] Nigam, B., & Saggar, A. Voice Enabled Application for the Visually-Impaired to Detect & Recognize Indian Currency in Hindi and English Using Deep Learning. *Available at SSRN 4191843*.

[6] Laavanya, M., & Vijayaraghavan, V. (2019). Real time fake currency note detection using deep learning. *Int. J. Eng. Adv. Technol.(IJEAT)*, *9*.

[7] Vincent, R., Bhatia, P., Rajesh, M., Sivaraman, A. K., & Al Bahri, M. S. S. (2020). Indian currency recognition and verification using transfer learning. *International Journal of Mathematics and Computer Science*, *15*(4), 1279-1284.

[8] Padmaja, B., Bhargav, P. N. S., Sagar, H. G., Nayak, B. D., & Rao, M. B. (2021, November). Indian Currency Denomination Recognition and Fake Currency Identification. In *Journal of Physics: Conference Series* (Vol. 2089, No. 1, p. 012008). IOP Publishing.

[9] Kumar, S. N., Singal, G., Sirikonda, S., & Nethravathi, R. (2020, December). A novel approach for detection of counterfeit Indiancurrency notes using deep convolutional neural network. In *IOP conference series: materials science and engineering* (Vol. 981, No. 2, p. 022018). IOP Publishing.

[10] Patil, N., Ingale, M. M., Shirsath, A. V., Bagga, K. R., & Patil, D. S. (2024). VISUAL CURRENCY VALIDATOR: AN ANDROID APP EMPOWERING THE VISUALLY IMPAIRED WITH IMAGE PROCESSING AND DEEP LEARNING FOR FAKE CURRENCY DETECTION.

[11] Mogan, J. N., Lee, C. P., Lim, K. M., & Muthu, K. S. (2022). VGG16-MLP: gait recognition with fine-tuned VGG-16 and multilayer perceptron. Applied Sciences, 12(15), 7639.

[12] Sudhakar, J., Iyer, V. V., & Sharmila, S. T. (2022, January). Image caption generation using deep neural networks. In 2022 International Conference for Advancement in Technology (ICONAT) (pp. 1-3). IEEE.

[13] pawan. (2024, March 21). *VGG-16: CNN model*. GeeksforGeeks. https://www.geeksforgeeks.org/vgg-16-cnn-model/

[14] Ramzan, F., Khan, M. U. G., Rehmat, A., Iqbal, S., Saba, T., Rehman, A., & Mehmood, Z. (2020). A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks. *Journal of medical systems*, *44*, 1-16.

[15] Abu, N. (2021). *Indian Currency for Classification* [Data set]. Kaggle. https://www.kaggle.com/datasets/najiaboo/indiancurrency-for-classification

[16] Mishra, D. K., Dey, N., Deora, B. S., & Joshi, A. (Eds.). (2020). ICT for competitive strategies: Proceedings of 4th International Conference on Information and Communication Technology for Competitive Strategies (ICTCS 2019), December 13th-14th, 2019, Udaipur, India (1st ed.). CRC Press.

[17] Afrin, N., Ahamed, F., & Rahman, A. (2024). Development of a convolutional neural network based regional flood frequency analysis model for South-east Australia. Natural Hazards, 1-28.