DeepMoney: Counterfeit money detection using generative adversarial networks

Toqeer Ali, Salman Jan, Ahmad Alkhodre, Mohammad Nauman, Muhammad Amin, Muhammad Shoaib Siddiqui

1 Faculty of computer and information systems, Islamic University of Madinah, Madinah, Saudi Arabia, Saudi Arabia
2 Malaysian Institute of Information Technology, University Kuala Lumpur, Kuala Lumpur, Kuala Lumpur, Malaysia
3 Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, Madinah, Saudi Arabia
4 Computer Science, FAST-NUCES, Peshawar, Pakistan

Corresponding Author: Toqeer Ali
Email address: toqeer@iu.edu.sa

Conventional paper currency and modern electronic currency are two important modes of transactions. In several parts of the world, conventional methodology has clear precedence over its electronic counterpart. However, the identification of forged currency paper notes is now becoming an increasingly crucial problem because of the new and improved tactics employed by counterfeiters. In this paper, a machine assisted system - dubbed DeepMoney- is proposed which has been developed to discriminate fake notes from genuine ones. For this purpose, state-of-the-art models of machine learning called Generative Adversarial Networks (GANs) are employed. GANs use an unsupervised learning to train a model that can then be used to perform supervised predictions. This flexibility provides the best of both worlds by allowing unlabelled data to be trained on whilst still making concrete predictions. This technique was applied to Pakistani banknotes. State-of-the-art image processing and feature recognition techniques were used to design the overall approach of a valid input. Augmented samples of images were used in the experiments which show that a high-precision machine can be developed to recognize genuine paper money. An accuracy of 80% has been achieved. The code is available as an open source to allow others to reproduce and build upon the efforts already made.
DeepMoney: Counterfeit Money Detection Using Generative Adversarial Networks

Toqeer Ali\(^1\), Salman Jan\(^2\), Ahmad Alkhodre\(^3\), Mohammad Nauman\(^4\), Muhammad Amin\(^5\), and Muhammad Shoaib Siddiqui\(^6\)

\(^1\)Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, Saudi Arabia
\(^2\)Malaysian Institute of Information Technology, University Kuala Lumpur, Malaysia
\(^3\)Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, Saudi Arabia
\(^4\)Computer Science, FAST-NUCES, Peshawar, Pakistan
\(^5\)Computer Science, FAST-NUCES, Peshawar, Pakistan
\(^6\)Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, Saudi Arabia

Corresponding author:
Toqeer Ali\(^1\)
Email address: toqeer@iu.edu.sa

ABSTRACT

Conventional paper currency and modern electronic currency are two important modes of transactions. In several parts of the world, conventional methodology has clear precedence over its electronic counterpart. However, the identification of forged currency paper notes is now becoming an increasingly crucial problem because of the new and improved tactics employed by counterfeiters. In this paper, a machine assisted system – dubbed DeepMoney– is proposed which has been developed to discriminate fake notes from genuine ones. For this purpose, state-of-the-art models of machine learning called Generative Adversarial Networks (GANs) are employed. GANs use an unsupervised learning to train a model that can then be used to perform supervised predictions. This flexibility provides the best of both worlds by allowing unlabelled data to be trained on whilst still making concrete predictions. This technique was applied to Pakistani banknotes. State-of-the-art image processing and feature recognition techniques were used to design the overall approach of a valid input. Augmented samples of images were used in the experiments which show that a high-precision machine can be developed to recognize genuine paper money. An accuracy of 80% has been achieved. The code is available as an open source to allow others to reproduce and build upon the efforts already made.

INTRODUCTION

Currency is a common system to perform transactions for trading or exchange of goods among people. Various currencies are recognized for trading between nations in foreign exchange markets. The problem with paper currency is that it may be counterfeit. Counterfeit is imitation money which is produced without the legal sanction of the state or government, considered as fraud (7). Anti-counterfeiting measures can be adopted which involve the fine details of the raised intaglio printing on notes which allows non-experts to easily spot forgeries (33).

With advancement in technology, fakes and forgery rates are increasing. In 2015, almost 70% of the $78 million in counterfeit currency circulating in the U.S was made using digital printing technologies (22; 3). As reported by Bloomberg news, a 34-year-old hairstylist forged up to $20,000 in counterfeit notes (17). Another research carried out by Gallup in Pakistan found out that more than a quarter of the population (26%) have received counterfeit money while buying items from the market. Elsewhere, a raid by Peshawar police yielded fake documents and 0.7 million in counterfeit currency notes. The chairman and managing director of a security printing corporation in Pakistan said that enemy countries were producing counterfeit notes of Pakistani rupee which was being used exclusively for terrorist activities.
in the country (29; 32). The State Bank of Pakistan has carried out various types of campaigns to raise
awareness of the security features of bank notes, both directly to its customers and in collaboration with
other banks. Bank’s efforts to raise awareness amongst the general public involve a media campaign and
a mobile application that detects counterfeit notes. The awareness of the security features of bank notes
in public may help some people identify counterfeit notes; however, most of the public, especially the
illiterate people are not able to differentiate between a forged currency note and a genuine one. Also,
these features are hard to recognized by human eye or touch when the currency notes are old, dirty, and
damaged.

To solve the issues in classifying a currency note as a fake or a genuine note, we have proposed a
machine assisted system named DeepMoney. For discriminating fake notes from genuine ones, state-of-
the-art models of machine learning called Generative Adversarial Networks (GANs) are employed. GANs
use an unsupervised learning to train a model that can then be used to perform supervised predictions.
This flexibility provides the best of both worlds by allowing unlabelled data to be trained on whilst still
making concrete predictions. This technique is applied to Pakistani banknotes. State-of-the-art image
processing and feature recognition techniques were used to design the overall approach of a valid input.
The rest of paper is organized as follows. Section 2 discusses the related work and provides details of
various deep learning models in the subject domain. Section 0.2 provides details of the proposed solution
and how the dataset was developed. Section 0.5 presents the results and evaluation details of employing
models on the dataset. The paper is concluded in Section 0.11.

RELATED WORK

Bearing the aforementioned issues in mind, a number of research solutions have been provided in the past
to check the validity of the banknotes(34; 19; 5; 24; 13). Mirza et al.(20) offered a solution for a currency
verification system using image processing based on the extraction of characteristics. The solution was
applied to Indian banknotes. The edge detection and image segmentation were used to make a comparison
between the original and the counterfeit notes.

Singh et al. have presented an image processing-based approach for detecting forged Indian currency.
The authors have utilized security thread and latent image, embedded on the note to identify forgery.
First the security features are extracted and encoded, then a clustering algorithm, k-means is applied for
classification. Then the latent image, segmented via template matching is encoded using HOG descriptor
and classified with an SVM model to predict if the note is fake or real (31).

Abburu et al. proposed a system for automated currency recognition using image processing techniques
for accurately identifying both the country of origin and the denomination of a given banknote (1). However, they do not discriminate between a fake and a real currency note. However, the DeepMoney
solution proposed here does not use image processing and differs in many ways.

Ross et al. have proposed a database for detecting counterfeit items using digital fingerprint records
which can be used for detecting counterfeit currency note (26). It takes an image of the authentication
region and creates a digital fingerprint of the object. It uses signal processing techniques, such as, FFT of
the image to create the digital fingerprint to extract features which are used to compare the fake and real
objects.

Kayani presents a bank note processing system based on florescence and phosphorescence detection
(14). The illumination source is used to direct light on a note and the sensors measures the florescence
and phosphorescence that are used to identify if the note is fake or real. Micali et al. proposed a solution
for counterfeit prevention using physically unclonable value for unique identification for each currency
note (18).

Phillips has presented a miniature counterfeit detector in his patent(23). It applies multiple tests to
assure the authenticity of a currency note. Back light illuminators are used for visual inspection of the
watermarks, florescent and anti-counterfeiting features. Sensors, such as, magnetic ink sensor, are used to
detect the magnetic ink based security features on the note. However, some of the features are detected
but with old notes the rate of false negative would be high.
Alicherry has given a digital signature based solution for verifying the authenticity of a currency note and tracking duplicate notes. A digital signature based on the serial number of the currency note is attached to the currency note and people can query the authenticity of the note by sending a photograph of the note to a centralized server for verification and tracking (Alicherry).

Before GANs (10), a neural network approach was used to authenticate banknotes (21). In this research, Generative Models to train a neural network are preferred. Python has also been used to develop and implement a framework for the identification of Pakistani currency.

Albert et al. (4) also worked on methods by which to identify genuine and photocopied bank notes. The technique applied was to differentiate the texture between the original and photocopied notes. Other studies (6) also worked on the detection of counterfeit banknotes by using optical coherence tomography (OCT). To differentiate between genuine and counterfeit notes, the researchers used a three-dimensional imaging security feature according to the FF-OCT system. Their results show that it is possible to recognize original notes with the DeepMoney technique. However, their technique was based on a particular feature of specific banknotes which may prove to be less effective on other notes. This also differs from the DeepMoney perspective of recognizing banknotes. Table 1 elaborates some recent research work regarding counterfeit money.

There have been a few solutions provided which use machine learning techniques. For example, Hassanpour et al. (11) used Hidden Markov Models for the recognition of banknotes. In the following subsection, we discuss some of the deep learning model that could be utilized in the field of identifying currency forgery and counterfeit currency.

Deep Learning Models
Some of the best known deep learning models in comparison to our proposed GANs model on the grounds for selecting the one which will offer the most robust results, are discussed in the following subsections.

0.1 Recurrent Neural Network
A number of techniques exist in traditional machine learning and deep learning which allow patterns in sequences and contents to be learned. A recurrent neural network is one such technique. Context sensitive and inherently ordered data is used in this network. The network can operate with both audio and text input. Recurrent neural networks are adept at handling arbitrary length sequence data. This network is a powerful tool which requires the sequence to be contextual. Recurrent neural networks are still very dominant although, in the past, they were very hard to train. However, there is now a solution, Hessian-free optimization, which offers the ability to train recurrent neural networks. The model is depicted in fig 1.

![Figure 1. A recurrent neural network](image-url)
| Authors                      | Objective                                                                 | Method                                      | Limitations                                                                 | Year |
|-----------------------------|---------------------------------------------------------------------------|---------------------------------------------|------------------------------------------------------------------------------|------|
| Thakur et al. (34)          | Review of fake currency detection techniques                              | Survey paper                                | Not applicable                                                               | 2014 |
| Chakraborty et al. (5)      | Recent developments in paper currency recognition system                 | Survey paper                                | Not applicable, performance is less than machine learning based systems     | 2013 |
| Prasanthi et al. (24)       | Indian paper currency authentication system                               | Image processing                            | Feature extraction and classification require high computation                | 2015 |
| Kang et al. (13)            | Fake banknote detection                                                  | multispectral imaging sensors               |                                                                              | 2016 |
| Mirza et al. (19)           | Currency verification                                                     | Image Processing: edge detection and image segmentation | Only for Indian notes                                                       | 2012 |
| Snehlata et al. (31)        | Fake currency identification                                             | UML activity model using class descriptors  | Only for Rs 2000 note of Indian currency                                     | 2017 |
| Singh et al. (30)           | detecting forged Indian currency                                         | UML activity model using class descriptors  | Limited to Rs 500 note of Indian currency                                    | 2018 |
| Abburu et al. (1)           | automated currency recognition for identifying country of origin and denomination | Image processing                            | Cannot detect counterfeit or forgery                                         | 2017 |
| Ross et al. (26)            | database for detecting counterfeit items                                  | digital fingerprint records using images of security features | performance is less than machine learning based systems, Many security features are not detectable using florescence and phosphorescence detection | 2016 |
| S. Kayani (14)              | bank note processing system                                              | florescence and phosphorescence detection   | Needs Internet connection for sending images to centralized server, Many security features are not detectable using florescence and phosphorescence detection | 2017 |
| Micali et al. (18)          | counterfeit prevention                                                   | physically unclonable value for unique identification for each currency note | Back light illuminators are used for visual inspection of the 93 watermarks, florescent and anti-counterfeiting features | 2017 |
| R. Phillips (23)            | miniature counterfeit detector                                           | digital signature based on the serial number of the currency note             | Needs Internet connection for sending images to centralized server           | 2018 |
| Alicherry (Alicherry)       | verifying the authenticity of a currency note and tracking duplicate notes| digital signature based on the serial number of the currency note             |                                                                              | 2017 |
| Albert et al. (4)           | identify genuine bank notes                                              | differentiate the texture between the original and photocopied notes using OFT characterization of safety feature on banknote with full-field optical coherence tomography | Accuracy is less than machine learning based systems                        | 2016 |
| Choi et al. (6)             | Counterfeit detection                                                    | Machine Learning: Hidden Markov Models      | Accuracy is less than the proposed system                                   | 2010 |
| Hassanpour et al. (11)      | paper currency recognition                                               | Image Processing: edge detection and image segmentation |                                                                        | 2009 |
| Mohamad et al. (21)         | Banknote authentication                                                  | artificial neural network                   | Accuracy is less than the proposed system                                   | 2014 |

**Table 1.** Related work done in the recent years in the field of counterfeit currency detection

---

### 0.2 Fully Connected Neural Networks

As its name indicates, a fully connected neural network is one in which all the neurons are connected to the next layer of neurons. There are many layers such as the max pooling and the convolutional layers. The high level perception in these networks uses only one type of layer and these layers are fully connected. These connected layers in the neurons link to all the activations in the previous layers. The activation function may also be calculated by the multiplication of a matrix trailed by a network’s 0 set, also known as bias. A fully connected neural network has an input layer, a hidden layer and an output...
Convolutional Neural Network

This type of neural network processes the dimensioned and order data in different ways. Convolutional neural networks (15) are learned through the same method as the stochastic gradient descent method traditionally learns. The following are the convolutional neural network layers:

Convolutional

In this layer there is a grid of neurons. Commonly, this grid is rectangular which requires that the previous layer should also take the form of the same rectangular shaped grid. In the convolutional layer, the neurons have the same weight. Each neuron takes the input from the rectangular section with the input coming from the previous layer.

Max-Pooling

Pooling layers are present after each convolutional layer. The layer usually grabs tiny rectangular blocks from the convolutional layer and then samples them so that only one output is created. Pooling can be achieved in many ways such as by taking an average or by learning patterns and combinations, for example learning linear associations or the combination of neurons in that small block.

Autoencoder

There are many machine learning models but auto-encoder are fairly basic models. They come from a family of neural networks for which the input is same as the output. Auto-encoders compress the latent space representation and then reconstruct the output from the representation. Auto-encoders are used for unsupervised learnings and so are an artificial network. The learning of encoders are efficient enough. Today, auto-encoders are an emerging field of research in numerous areas, such as, anomaly detection.

Deep Belief Networks

Deep belief networks (16) consist of layers of variables, both latent as well as stochastic. Latent variables are usually composed of values that are binary and known as feature detectors. There are many layers in this network. The top layers consist of connections that are undirected which allows them to form a kind of associative memory. On the other hand, the lower layers are directed links from any previous or top layer. A lower layer represents a data vector. An example of Deep Belief Networks is depicted in Figure 3:
The connected neural network, also called the recurrent neural network, works but, when applied on different models, it suffers from two problems. Firstly, it produces a vanishing gradient and, secondly, it is an exploding gradient. LSTM (long short term memory) was invented to solve these issues by introducing a memory unit, which is called a cell, into the network. The cell is used as storage or memory that basically remembers a value for long or possibly short time periods (12; 27; 35). LSTM are represented in Figure 4.

Problem Statement
The existing solutions only work to detect the security features of the notes and at present this is the only viable solution used to tackle this issue. Forged notes are not detectable by the naked eye and this results in financial loss to the general public. Degraded bank notes are also in circulation which may result in usage problems. The solution given in this paper is based on Generative Adversarial Networks (GANs) (10) that use generative and discriminative models for the recognition of real and counterfeit currency notes. To the best of our knowledge, this is first research being done to detect forged money using GANs. We trained the discriminative model with Pakistani rupee notes while the generative model G was used to input the counterfeit notes to classify with discriminative model D. The objectives of the studies is two
fold:

1. To build a dataset of real and counterfeit Pakistani currency which is not available at present in the public domain.

2. To present, design, and implement DeepMoney, a method to differentiate counterfeit notes from genuine banknotes.

PROPOSED SOLUTION

In this section, the proposed solution is elaborated and details are provided as to how Generative Adversarial Networks cope with real and fake data.

Generative Adversarial Networks

A very new and effective Generative Model is utilized for generative counterfeit samples and for recognizing original data items from the generative ones, known as Generative Adversarial Networks (GANs) (9, 8, 25, 28). GANs can differentiate with maximum accuracy between the fake and real banknotes. GANs are quite interesting phenomena of neural networks that work on two modules. One is called the generative network and the second is known as the discriminator network. Quite promising results have been achieved after employing GANs on the dataset and then subsequently used to classify the real and fake notes. The proposed model is depicted in Fig 5 wherein the Discriminator Neural Network (D) is trained by providing data from training set (original distribution) and generated data (after perturbation i.e. adding noise from the latent space). The loss functions are updated during training process of each Discriminator’s network and the Generator Network (G). After training is completed, the model is able to classify real and fake currency notes with accuracy of 80 percent.

In GANs, the discriminator part works as a classifier for recognizing the images. However, in the learning process, both the generator and discriminator co-ordinate with each other. A generated image is sent to the discriminator module to classify whether it is a fake or a real image. If it is recognizable, the discriminator will produce its output, if it is not, the module will send it back to the generator to regenerate the image. Based on the feedback received, the generator improves its cast and creates the image. The process continues until both the models are optimal in correctly generating and classifying the same image.

In the case of currency note identification, the basic idea is that there are two models as defined by GANs. The generative model generates fake banknotes and the discriminative model D estimates the probability that whatever data is received from G, it is from training data or it is from generator G itself. To understand the generator’s distribution $P_g$ over data ($a$), a priori is defined on input noise variables $P_b(b)$,
which then represents a mapping to the banknotes data space as \( G(b; \theta_g) \), where \( G \) is a differentiable function represented by a multilayer perceptron with parameters \( \theta_g \).

According to GANs networks, a second multilayer perceptron was also designed \( D(y; \theta_d) \) that outputs a single scalar. \( D(a) \) represents the banknotes that came from the data, not from the generator. The discriminator \( D \) was trained with real banknotes to increase the probability of correctly recognizing the data \( a \) and the sample \( b \) was produced from the generative model \( G \). The loss or energy function of the generative model \( G \) can be represented mathematically as:

\[
\frac{1}{m} \sum_{i=1}^{m} \left[ \log(1 - (D(G(b)))) \right]
\]

and the loss function of the Discriminator can be represented as:

\[
\frac{1}{m} \sum_{i=1}^{m} \left[ \log D(a) + \log(1 - (D(G(b)))) \right]
\]

Figure 5 shows a complete flow of how our DeepMoney process works. Real images are given as input to the discriminator, while for training, communication is performed between the discriminator and the generator.

Different components have been built to perform the verification of the currency note. Figure 6 shows the basic structure of the DeepMoney architecture, its actors, and the way they interact with each other. There is a single actor here, which is shown as the user. The user can input an image and request the authentication of the currency note. The system will respond through various other functions, as shown in the Figure. External systems may be used to receive assistance from the sensors if necessary. As DeepMoney progresses, other more appropriate features may be found and the use of these initial features may be minimized and other more versatile features may evolve. In this case, the ‘input image’ will remain the same but the dependent functionalities may change.

**Figure 6.** DeepMoney Currency Bill Verification components

### 0.3 Data preparation and augmentation

Data collection and its preprocessing is carried out before training neural networks or deep learning models for subsequent recognition tasks. The following subsections provide the necessary details of how
data is prepared and augmented with image datasets through the following function:

```python
datagen = ImageDataGenerator()
```

Taking pictures and building a dataset is a laborious and time-consuming task. An API specially designed for creating augmented image data is used, which takes less time to augment the data and reduces memory overhead. ImageDataGenerator augments the data and when data has been created and configured, then it is time to fit the data by calling the fit() function. Once the function is called on the data generator, it is then passed in the training dataset.

```python
datagen:fit(train)
```

Data generator is an iterator and it returns the batches of image back when they have been requested.

```python
X_batch, Y_batch = datagen:flow(train; train; batchsize = 32)
```

Finally, fitgenerator() is called with arguments of the desired length of epochs on which to train the data. The other one is the data generator.

```python
fitgenerator (datagen; samplesPerEpoch = len (train) ; epochs = 100)
```

There are many ways to configure the data preparation and its augmentation. These include, feature-wise standardization, ZCA whitening, random rotation, shifts, shear and ips, dimension reordering, saving augmented images to disk. All the functions/augmentation are performed by calling the ImageDataGenerator function. As a result of providing the input image as depicted in Figure 7, the function ImageDataGenerator produces augmented data through incorporating the aforementioned features as shown in Figure 8.

![Figure 7. Input Image](image1)

![Figure 8. Output Images](image2)

### 0.4 Experimental Setup

The experiments are carried out on a GPU machine in FAST University with the following configuration. Keras is configured with Theano as backend. From the Keras library, the packages, Dense, Dropout, Activation, and Flatten layers are imported. The Dense layer is used to define filters with model parameters to identify various features of the currency notes. The Dropout layer is used to address overfitting and in order to ignore some of the features which are not contributing in identifying the actual features of the
notes. The Activation function, “Relu” is used in order to represent the learned information in ranges of 0 and 1. The number of epochs and batch size is set to 100 and 32, respectively.

To use the ImageDataGenerator function, Keras was utilized beside configuring scipy, numpy, h5py and pyyaml. Furthermore, a number of parameters were set for achieving augmentation which included: samplewise center, featurewise std normalization, ZCA Pesilon, ZCA whitening, rotation range, width shift range, height shift range, shear range, zoom range, channel shift, horizontal, vertical, and rescale. These are set to either, 0, 1 or in float values for the required changes. After specifying the parameters and storing the same in a datagen variable, the images were imported.

As specified by the 36 iterations, 36 images of the Bank Note with changes that were specified in the datagen will be produced and will be stored in the folder called preview.

0.5 Implementation

Comprehensive details are provided as to how the environment was set up for executing this project of GANs on the dataset. The first step involved the setup and configuration of the requirements that would help to execute the code. After running certain required script, GAN was trained on the DeepMoney dataset. Both the discriminator and the generator used a learning rate of 0.0002. The generator updated this twice for each update of the discriminator. This actually avoids the small discriminator loss. In the next step, results are produced and saved in PNG format.

The class diagram in Figure 9 represents the major classes of DeepMoney. As can be seen, the main controller class is InputImage and every other class is developed from that class. InputImage reads an image and passes that image to one of those classes, or to multiple classes, according to the requirement. The respective classes will then call their functions that may call another function of different classes.

![Class Diagram for DeepMoney Architecture](image)

**Figure 9.** Class Diagram for DeepMoney Architecture

The sequence diagram in Figure 10 shows the functions and their activities after they have been executed. As discussed earlier, functions may or may not call other functions of different classes if the task required is beyond their scope. The functions of other classes are called through instances of their respective class. The called function then returns the result to the parent function which, in turn, processes the user request.

The algorithm for the GANs is provided in Figure 11, while algorithm for generating images is provide in Figure 12 and the algorithm for augmentation is provided in Figure 13.
Algorithm for Generative Adversarial Network

1 for <number of training iterations> do
2 for <k steps> do
3 Sample minibatch of m noise samples b1, ..., bm from noise prior Pg(b)
4 Sample minibatch of m examples a1, ..., am from data
5 generating distribution pdata(a)
6 Update the discriminator by ascending it's stochastic gradient
7 end for
8 Sample minibatch of m noise samples b1, ..., bm from noise prior Pg(b)
9 update the generator by descending it's stochastic gradient
10 end for

Figure 11. Algorithm for Generating Currency Notes

Algorithm for Generating Images

1 x = img to array (img)
2 creating a Numpy array with shape (3, 150, 150)
3 x = x.reshape ((1) + x.shape)
4 converting to a Numpy array with shape (1, 3, 150, 150)
5 Loopl runs for 36 times
6 for each iteration
7 datagen:flow ()
8 #function was used with x being given the numpy array for the input image
9 save to dir, the directory to save output
10 save pre x, the pre x for the names of the images and
11 save format, the image format as input

Figure 12. Algorithm for Generating Currency Notes
Augmentation Algorithm

Procedure from keras.preprocessing.image(import ImageDataGenerator)

datagen ImageDataGenerator rotRng, wtSft, hsSft, zoomRng, hFlp,

Mde x = x:reshape ((1) + x:shape)

while r6= 0 do
    img loadImage
    x imgToArr
    x x:reshape
    i 0
    end while
    for <batch in datagen.flow> do
        <i+=1>
        end for
        if i > 36
        break
    end procedure

Figure 13. Algorithm for Generating Currency Notes
RESULTS

This section elaborates the results obtained through the experiments conducted on the DeepMoney dataset as depicted in Table 2. Moreover, a model evaluation is presented with the confusion matrix, supervised loss function, the generator and discriminator respective losses and finally the classification accuracy.

0.6 Area Under the Curve (AUC)

The trained model was able to achieve 80% accuracy in correctly classifying data. The Area Under the Curve (AUC)/Receiver Operating Characteristic (ROC) is presented in Figure 14.

![Figure 14. DeepMoney Area Under Curve](image)

0.7 Confusion Matrix

The confusion matrix consists of three classes along the x and y axis. Each row of the matrix represents each class along with the y-axis which shows its comparison with all the remaining classes. The classes belong to the DeepMoney dataset having images of Rs. 50, Rs. 500, and Rs. 1000. Each class is assigned a unique density color. High intensity shades represent no confusion between classes while the lower intensity band represents confusion rates between the classes. In Figure 15 the diagonal showing 1’s indicate that the confusion matrix calculation is near perfection.

0.8 Supervised Loss

Cross-entropy was minimized for the multi-class softmax and for that purpose the labels were adjusted. A label mask in code was used to ignore the images that are unlabeled for the SS-learning problem. The TensorFlow graph visualization for the (percentage vs No. Epochs) are shown in Figure 16, 17, and 18.

0.9 Total Discriminator Loss (TD Loss)

Using the objective function of GANs, the Total Discriminator loss (TD-Loss) is brought down to 0.5. This helped to make the Discriminator more efficient in classification. The Tensorflow Graph Visualization below is pitted against (Percentage-Loss No. Epochs) in figure 16.

0.10 Generator Loss (G Loss)

Using the Objective function of GANs, the Generator’s loss (G loss) was brought up to (2.65), which can be seen in Figure 17. This helped to make the Generator more robust in training the Discriminator. The Tensorflow Graph Visualization below is visualised against (Percentage-Loss vs No. Epochs).

Table 2. Experimental Results

| No. of Folds | Accuracy  | Sensitivity | Specificity | Precision | Kappa   | F1 Score | ROC/AUC |
|--------------|-----------|-------------|-------------|-----------|---------|----------|---------|
| 5            | 76.8761%  | 0.7301      | 0.9771      | 0.7518    | 0.1283  | 0.7567   | 0.7598  |
| 10           | 79.5420%  | 0.7356      | 0.9783      | 0.7540    | 0.1854  | 0.7632   | 0.7745  |
| 15           | 73.9300%  | 0.7393      | 0.9798      | 0.7387    | 0.1767  | 0.7689   | 0.7965  |
The Discriminator is trained to classify the unseen images and the efficiency of the same is measured in terms of correct classification. As can be seen in Figure 18, a remarkable classification accuracy (80%) was achieved, which was made possible by training a Discriminator through DeepMoney dataset and Generator images. The Tensorflow Graph Visualization below is depicted in Figure 18 (PercentageAccuracy vs No. Epochs).
Figure 17. Generator Loss Graph Visualization

Figure 18. Classification Accuracy Graph Visualization
CONCLUSION

Implementation of GANs models in the domain of computer vision has been proven to be effective. Upon testing in the currency originality, the discriminator model is seen as a viable contender in the classifier area. Additionally, classifier should be sufficiently trained with enough images by means of the real dataset and the generator module, especially in the Semi Supervised Generative Adversarial Networks (SSGANs). The dataset alteration can be performed in various other parametric tweaking with the Keras framework. 80% accuracy is achieved the proposed GANs framework for counterfeit money detection. However, there is still room for improvement. As new generative models are created in the machine learning domain, DeepMoney can be tested upon these to achieve better and more effective results. Multi-class classifiers can be made good enough to result in improved accuracy. Classification is regarded to be at the forefront of machine learning, therefore, a better multi-class classifier would yield an even better retrospective score for the model.

REFERENCES

[1] Abburu, Vedasamhitha, S. G.-S. R. R. M. M. and Koolagudi, S. G. (2017). Currency recognition system using image processing. In Conference Computing (IC3), 2017 Tenth International Conference on, pages 1–6. IEEE.

[2] Alicherry, M. (2013). Verifying authenticity of currency and tracking duplicates. In conference on Image Processing, Analysis & Recognition, 2013 International Conference on, pages 11–16.

[3] Bartkiewicz, K., Černoch, A., Chimczak, G., Lemr, K., Miranowicz, A., and Nori, F. (2017). Experimental quantum forgery of quantum optical money. npj Quantum Information, 3(1):7.

[4] Berenguel, A., Terrades, O. R., Lladós, J., and Cañero, C. (2016). Banknote counterfeit detection through background texture printing analysis. In Document Analysis Systems (DAS), 2016 12th IAPR Workshop on, pages 66–71. IEEE.

[5] Chakraborty, K., Basumatary, J., Dasgupta, D., Kalita, J. C., and Mukherjee, S. (2013). Recent developments in paper currency recognition system. Int. J. Res. Eng. Technol, 2:222–226.

[6] Choi, W.-J., Min, G.-H., Lee, B.-H., Eom, J.-H., and Kim, J.-W. (2010). Counterfeit detection using characterization of safety feature on banknote with full-field optical coherence tomography. Journal of the Optical Society of Korea, 14(4):316–320.

[7] Derrida, J. (1994). Given time: I. Counterfeit money, volume I. University of Chicago Press.

[8] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). Deep learning, volume I. MIT press Cambridge.

[9] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680.

[10] Goodfellow, Ian, J. P.-A. M. M. B. X. D. W.-F. S. O. A. C. and Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems, International Conference on, pages 2672–2680. NIPS.

[11] Hassanpour, H. and Farahabadi, P. M. (2009). Using hidden markov models for paper currency recognition. Expert Systems with Applications, 36(6):10105–10111.

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

[13] Kang, K. and Lee, C. (2016). Fake banknote detection using multispectral images. In Information, Intelligence, Systems & Applications (IISA), 2016 7th International Conference on, pages 1–3. IEEE.

[14] Kayani, S. (2017). Bank note processing system having a combined florescence and phosphorescence detection system. U.S. Patent 9,761,077, issued September 12, 2017. [Online; accessed 01-Feb-2018].

[15] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105.

[16] Lee, H., Pham, P., Largman, Y., and Ng, A. Y. (2009). Unsupervised feature learning for audio classification using convolutional deep belief networks. In Advances in neural information processing systems, pages 1096–1104.

[17] Magdaleno, A. (2016). Woman Pleads Guilty for Printing Thousands of Money as Fake. http://on.mash.to/21U2vZV. [Online; accessed 01-Jan-2018].

[18] Micali, S. and Devadas, S. (2017). Counterfeit prevention. U.S. Patent Application 15/522,348, filed November 16, 2017. [Online; accessed 01-Feb-2018].
[19] Mirza, R. and Nanda, V. (2012a). Design and implementation of indian paper currency authentication system based on feature extraction by edge based segmentation using sobel operator. International Journal of Engineering Research and Development, 3(2):41–46.

[20] Mirza, R. and Nanda, V. (2012b). Paper currency verification system based on characteristic extraction using image processing. International Journal of Engineering and Advanced Technology (IJEAT) ISSN, 2249:8958.

[21] Mohammad, N. S., Hussin, B., Shibghatullah, A. S., and Basari, A. (2014). Banknote authentication using artificial neural network. Science International, pages 1865–1868.

[22] Murakami-Fester, A. (2016). Counterfeit Cases in US. https://www.usatoday.com/story/money/personalfinance/2016/10/23/counterfeit-money-spot-fake/92080738/. [Online; accessed 01-Jan-2018].

[23] Phillips, R. (2018). Miniaturized counterfeit detector. U.S. Patent Application 15/817,848, filed June 14, 2018. [Online; accessed 01-Feb-2018].

[24] Prasanthi, B. S. and Setty, D. R. (2015). Indian paper currency authentication system using image processing. Int. J. Sci. Res. Eng. Technol, 4:973–981.

[25] Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

[26] Ross, David Justin, B. J. E. M. T. J. F. and Ross, H. W. (2016). Database for detecting counterfeit items using digital fingerprint records. U.S. Patent Application 15/208,339, filed November 17, 2016. [Online; accessed 01-Feb-2018].

[27] Sak, H., Senior, A., and Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth annual conference of the international speech communication association.

[28] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. In Advances in Neural Information Processing Systems, pages 2234–2242.

[29] Shoaib, M., Ilyas, M., and Khiyal, M. S. H. (2013). Official digital currency. In Digital Information Management (ICDIM), 2013 Eighth International Conference on, pages 346–352. IEEE.

[30] Singh, Mrutunjay, P. O. and Abhiram, K. (2018). Image processing based detection of counterfeit indian bank notes. In 9th International Conference on Computing, Communication and Networking Technologies, pages 1–5. ICCCNT.

[31] Snehlata and Saxena, V. (2017). Identification of fake currency: A case study of indian scenario. International Journal of Advanced Research in Computer Science, 8(3):213–218.

[32] Taillard, M. (2018). Counterfeiting. In Economics and Modern Warfare, pages 153–157. Springer.

[33] Tanaka, T., Nishiyama, S., and Koyama, M. (1996). Method for making an anti-counterfeit latent image formation object for bills, credit cards, etc.

[34] Thakur, M. and Kaur, A. (2014). Various fake currency detection techniques. International Journal for Technological Research in Engineering, 1(11):1309–1313.

[35] Warrington, E. and Baddeley, A. (2017). Amnesia and the distinction between long-and short-term memory 1. In Exploring Working Memory, pages 18–38. Routledge.
Figure 1

A recurrent neural network
Figure 2

A Fully Connected Neural Network
Figure 3

Deep Belief Networks

\[
P(V, h^1, h^2, \ldots, h^l) = P(V|h^1) P(h^1|h^2) \ldots P(h^{l-2}|h^{l-1}) P(h^{l-1}|h^l)
\]
Figure 4

Long Short-Term Memory
Figure 5

DeepMoney: Proposed Counterfeit Model
Figure 6

DeepMoney Currency Bill Verification components
Figure 7

Input Image
Figure 8

Output Images
Figure 9

Class Diagram for DeepMoney Architecture
Figure 10

DeepMoney Sequence Diagram
Figure 11

Algorithm for Generating Currency Notes

---

for <number of training iterations> do
  for <k steps> do
    Sample minibatch of m noise samples f1, ..., bm from noise prior Pg(b)
    Sample minibatch of m examples f1, ..., amg from data generating distribution pdata(a)
    Update the discriminator by ascending it’s stochastic gradient
  end for
  Sample minibatch of m noise samples f1, ..., bm from noise prior
  Pg(b) update the generator by descending it’s stochastic gradient
end for

**Figure 11.** Algorithm for Generating Currency Notes
Figure 12
Algorithm for Generating Currency Notes

```
x = img to array (img)
creating a Numpy array with shape (3, 150, 150)
x = x:reshape ((1) + x:shape)
converting to a Numpy array with shape (1, 3, 150, 150)
for each iteration
  datagen:flow ()
  #function was used with x being given the numpy array for the input image
  save to dir, the directory to save output
  save pre x, the pre x for the names of the images and
  save format, the image format as input
```

Figure 12. Algorithm for Generating Currency Notes
Figure 13

Algorithm for Generating Currency Notes

```
Augmentation Algorithm

Procedure from keras.preprocessing.image(import ImageDataGenerator)
datagen ImageDataGenerator rotRng, wtSft, htSft, zoomRng, hFlp,
Mdx x = x:reshape ((1) + x:shape)
while r6= 0 do
  img loadImage
  x imgT oArr
  x x:reshape
  i 0
end while
for <batch in datagen.flow> do
  <i+=1>
end for
if i > 36
  break
end procedure
```

**Figure 13.** Algorithm for Generating Currency Notes
Figure 14

DeepMoney Area Under Curve
Figure 15

Confusion Matrix Calculation
Figure 16

Discriminator Loss Graph Visualization
Figure 17

Generator Loss Graph Visualization
Figure 18

Classification Accuracy Graph Visualization