Fake Image Detection Using Machine Learning

Muhammed Afsal Villan
Department of Computer Science and Engineering
Mar Athanasius College of Engineering, Kothamangalam
Kerala, India
Email:afsalashyana@gmail.com

Kuncheria Kuruvilla
Department of Computer Science and Engineering
Mar Athanasius College of Engineering, Kothamangalam
Kerala, India
Email:kuncheria94@gmail.com

Johns Paul
Department of Computer Science and Engineering
Mar Athanasius College of Engineering, Kothamangalam
Kerala, India
Email:johnspaul007@gmail.com

Prof. Eldo P Elias
Department of Computer Science and Engineering
Mar Athanasius College of Engineering, Kothamangalam
Kerala, India
Email:eldope@gmail.com

Abstract—Many fake images are spreading through digital media nowadays. Detection of such fake images is inevitable for the unveiling of the image based cybercrimes. Forging images and identifying such images are promising research areas in this digital era. The tampered images are a detected using neural network which also recognizes the regions of the image that have been manipulated and reveals the segments of the original image. It can be implemented on Android platform and hence made available to common users. The compression ratio of the foreign content in a fake image is different from that of the original image and is detected using Error Level Analysis. Another feature used along with compression ratio is image metadata. Although it is possible to alter metadata content making it unreliable on its own, here it is used as a supporting parameter for error level analysis decision

Keywords -- Image forensics, Metadata analysis, Error level analysis, Multilayer perception network, Deep neural networks

I. INTRODUCTION

In this technological era a huge number of people have become victims of image forgery. A lot of people use technology to manipulate images and use it as evidences to mislead the court. So to put an end to this, all the images that are shared through social media should be categorized as real or fake accurately. Social media is a great platform to socialize, share and spread knowledge but if caution is not exercised, it can mislead people and even cause havoc due to unintentional false propaganda. While manipulation of most of the photoshoped images is clearly evident due to pixelization & shoddy jobs by novices, some of them indeed appear genuine. Especially in the political arena, manipulated images can make or break a politician's credibility.

Current forensic techniques require an expert to analyze the credibility of an image. We implemented a system that can determine whether an image is fake or not with the help of machine learning and thereby making it available for the common public. This paper will unfold into three sections whereby first will focus on the second will focus on the Implementation details while the last part showcase the experimental result.

II. THEORY

A. Metadata Analysis

Most image files do not just contain a picture. They also contain information (metadata) about the picture. Metadata provides information about a picture's pedigree, including the type of camera used, color space information, and application notes. Different picture formats include different types of metadata. Some formats, like BMP, PPM, and PBM contain very little information beyond the image dimensions and color space. In contrast, a JPEG from a camera usually contains a wide variety of information, including the camera's make and model, focal and aperture information, and timestamps.

PNG files typically contain very little information, unless the image was converted from a JPEG or edited with Photoshop. Converted PNG files may include metadata from the source file format.

Metadata provides information related to how the file was generated and handled. This information can be used to identify if the metadata appears to be from a digital camera, processed by a graphical program, or altered to convey misleading information. Common things to look for include:

1) Make, Model, and Software

These identify the device or application that created the picture. Most digital cameras include a Make and Model in the EXIF metadata block. (However, the original iPhone does not!) The Software may describe the camera's firmware version or application information.

2) Image size

The metadata often records the picture's dimensions. Does the rendered image size (listed at the bottom of the metadata) match the other sizes in the metadata? Many applications resize or crop pictures without updating other metadata fields.

3) Timestamps

Look for fields that detail timestamps. These typically identify when a picture was taken or altered. Do the timestamps match the expected timeframe?

4) Types of metadata

There are many different metadata types. Some are only generated by cameras, while others are only generated by applications.

5) Descriptions

Many pictures include embedded annotations that describe the photo, identify the photographer, or itemize alteration steps.

6) Missing metadata

Are any metadata fields missing? If the picture came from a digital camera, then it should have camera-specific information. Some applications and online services strip out metadata. A lack of specific metadata usually indicates a resaved picture and not an original photo.

7) Altered Metadata

Metadata is analogous to the chain of custody for evidence handing. It can identify how a picture was generated, processed, and last saved. However, some people intentionally alter metadata. They may edit timestamps or photo information in an attempt to deceive

B. Error Level Analysis

JPEG is a lossy format, but the amount of error introduced by each resave is not line. Any modification to the picture will alter the image such that stable areas (no additional error) become unstable. Fig.1 shows a modified image using Photoshop. The modified picture was based on the first 75% resave. Books on the shelf were duplicated and a toy dinosaur was added to the shelf. The 95% ELA identifies the changes since they are areas that are no longer at their minimal error level. Additional areas of the picture show slightly more volatility because Photoshop merged information from multiple layers, effectively modifying many of the pixels.

A 90% image resaved at 90% is equivalent to a one-time save of 81%. Similarly, saving an image at 75% and then resaving it at 90% (75% 90%) will generate virtually the same image as 90% 75%, or saved once at 67.5%. 19 The amount of error is limited to the 8x8 cells used by the JPEG algorithm; after roughly 64 resaves, there is virtually no change. However, when an image is modified, the 8x8 cells containing the modifications are no longer at the same error level as the rest of the unmodified image. Error level analysis (ELA) works by intentionally resaving the image at a known error rate, such as 95%, and then computing the difference between the images. If there is virtually no change, then the cell has reached its local minima for error at that quality level. However, if there is a large amount of change, then the pixels are not at their local minima and are effectively “original”. JPEG is a lossy format, but the amount of error introduced by each resave is not linear modification to the picture will alter the image such that stable areas (no additional error) become unstable [1].

Books on the shelf were duplicated and a toy dinosaur was added to the shelf. The 95% ELA identifies the changes since they are areas that are no longer at their minimal error level. Additional areas of the picture show slightly more volatility because Photoshop merged information from multiple layers, effectively modifying many of the pixels. Nearly all pixels in the original image are not at their local minima. The first resave (75%) shows large areas where the pixels have reached their local minima. The second resave introduces more areas that have reached their local minima for error.

By analyzing the pattern in the ELA applied image (Fig 1 left part), we can determine which part of the image is possibly faked. It is hard for the human eye to detect small scale changes to image so that we have decided to use machine learning to detect the anomalies in the error level analyzed images.

Fig 1. Error level analyzed image on the left and fake image on the right

C. Machine Learning

The process of machine learning is similar to that of data mining. Both systems search through data to look for patterns. However, instead of extracting data for human comprehension as is the case in data mining applications machine learning uses that data to detect patterns in data and adjust program actions accordingly. Machine learning algorithms are often categorized as being supervised or unsupervised. Supervised algorithms can apply what has been learned in the past to new data. Unsupervised algorithms can draw inferences from datasets.

Facebook's News Feed uses machine learning to personalize each member's feed. If a member frequently stops scrolling in order to read or "like" a particular friend's posts, the News Feed will start to show more of that friend's activity earlier in the feed. Behind the scenes, the software is using statistical analysis and predictive analytics to identify patterns in the user's data and use to patterns to populate the News Feed. They observe the activities of the user like, comment, share etc on various posts and based on these activities the contents on the news feed will be adjusted continuously.

III. SYSTEM DESIGN

A. Metadata Analysis

The entire system is developed using java programming language. For extracting metadata of images, metadata - extractor library is used. Metadata-extractor [5] is able to

extract metadata information of large no of different image types. Once an image is selected for processing, it is tunneled into 2 separate stages. First stage is metadata analysis. After extracting metadata, the metadata text is fed into metadata analysis module.

Metadata analyzer is basically a tag searching algorithm. If keywords like Photoshop, Gimp, Adobe etc. is found in the text and then the possibility of being tampered is increased. Two separate variables are maintained which are called fakeness and realness. Each variable represents the weight of being real or fake image. Once a tag is taken, it is analyzed and corresponding variable is incremented by a certain predefined weight. The following table represents keywords and corresponding weight increments. After processing the entire tags, final values of fakeness and realness variable is fed into the output stage as shown in Fig 2.

Table 1. Keyword listing

Keyword	Realness / Fakeness	Inc. Value
Photoshop	Fakeness	5
Gimp	Fakeness	5
Corel	Fakeness	5
Adobe	Fakeness	3
Exif Info	Realness	2
Camera Tags	Realness	2

B. Error Level Analysis

Error level analysis is done with the help of ImageJ [2]

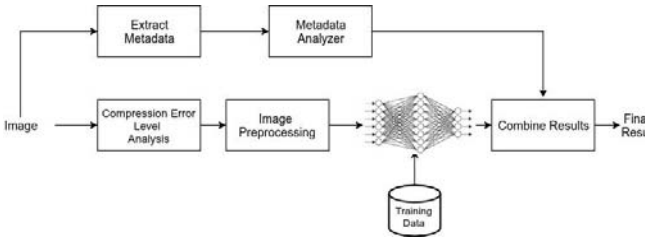

Fig 2. Flow chart of the system

library. ImageJ provides option to save image in JPEG format with certain percentage of compression. The system first saves an image at 100% quality. Then the same image is converted into 90% quality image using ImageJ. The difference between these two is found out though difference method. The resulting image is the required ELA image of the input image. This image is saved as a buffered image and sent to the neural network for further processing.

C. Machine Learning

Machine learning is implemented using Neuroph [4] library for java. Neuroph is selected because of the simplicity and easiness to implement neural networks. We have implemented a multilayer perceptron network with momentum back propagation learning rule. The structure of neural network is shown in Table 2.

A multilayer perceptron neural network is used having one input layer, 3 hidden layers and 1 output layer. Once the image is selected for evaluation, it is converted to ELA representation from Compression and Error Level Analysis stage. 100%, 90% images are used for the construction of ELA image

Once ELA is calculated, the image is preprocessed to convert into 100x100px width and height. After preprocessing, the image is serialized in to an array. The array contains 30,000 integer values representing 10,000 pixels. Since each pixel has red, green and blue components, 10,000 pixels will have 30,000 values.

During training, the array is given as input to the multilayer perceptron network and output neurons also set. The MLP is a fully connected neural network. There are 2 output neurons. First neuron is for representing fake and the second one for real image. If the given image is fake one, then the fake neuron is set to one and real is set to zero. Else fake is set to zero and real set to one.

We have used momentum backpropagation learning rule adjust the neuron connection weights. It is a supervised learning rule that tries to minimize the error function. The chosen learning rate and momentum along with achieved efficiency is given in Table 3.

During testing, the image array is fed into the input neurons and values of output neurons are taken. We have used sigmoid activation function.

Table 2. Structure of Neural Network

Layer	Remarks
Input Layer	30,000 neurons
Hidden Layer 1	5000 neurons, Sigmoid activation function
Hidden Layer 2	1000 neurons, Sigmoid activation function
Hidden Layer 3	100 neurons, Sigmoid activation function
Output Layer	2 neurons

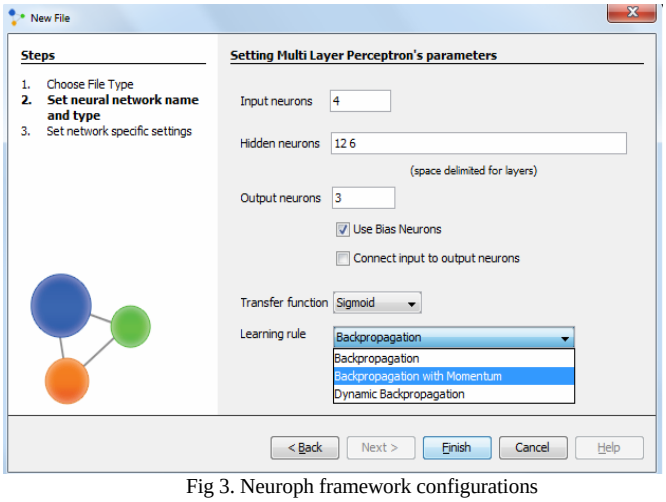

Fig 3. Neuroph framework configurations

IV. EXPERIMENTAL RESULT

Metadata analysis has shown promising result in non-shared images. It is able to detect anomaly in all 'photoshopped' or 'gimped' images under a very small processing. It failed on images shared through WhatsApp, Google+ etc. Moreover, it became completely erroneous when images with manipulated metadata given.

Neural network is trained with CASIA dataset [3]. The dataset contains 7491 real images and 5123 tampered images under varying sizes. All the images are preprocessed to 100x100 pixels so that total pixel values to be fed into the neural network will be 30,000. From the dataset we have used 4000 real and fake images for training. Remaining images were used for testing of the neural network. Table 3 shows various neural network configurations and corresponding neural network efficiency. Best is achieved when learning rate set to 0.2 and momentum to 0.7.

Table 3. Neural network training results

Learning Rate	Momentum	Epoch	Efficiency
0.01	0.5	500	60%
0.05	0.5	500	62%
0.1	0.5	500	68%
0.2	0.5	500	66%
0.1	0.4	500	69%
0.1	0.3	500	68%
0.1	0.6	500	75%
0.1	0.7	500	76%`
0.2	0.7	500	82%
0.2	0.7	1000	83%

V. CONCLUSION

Neural network has been successfully trained using the error level analysis with 4000 fake and 4000 real images. The trained neural network was able to recognize the image as fake or real at a maximum success rate of 83%. The use of this application in mobile platforms will greatly reduce the spreading of fake images through social media. This project can also be used as a false proof technique in digital authentication, court evidence evaluation etc. By combining the results of metadata analysis (40%) and neural network output (60%) a reliable fake image detection program is developed and tested. The complete project is available in GitHub [6].

REFERENCES

[1] A picture's worth, Digital Image Analysis and Forensics, N Krawetz - 2007 Ph D, Hacker Factor Solutions

[2] http://imagej.net/Welcome
ImageJ is an open source image processing program designed for scientific multidimensional images.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] http://forensics.idealtest.org/ CASIA v2.0
CASIA V2.0 is with larger size and with more realistic and challenged fake images by using post-processing of tampered regions. It contains 7491 authentic and 5123 tampered color images.

[4] http://neuroph.sourceforge.net/ Neuroph Framework
Neuroph is lightweight Java neural network framework to develop common neural network architectures. It contains well designed, open source Java library with small number of basic classes which correspond to basic NN concepts.

[5] https://github.com/drewnoakes/metadata-extractor
Metadata-extractor is a straightforward Java library for reading metadata from image files.

[6] https://www.github.com/afsalashyana/FakeImageDetection
GitHub repositor for fake image detector desktop application written in javafx.