# B.N.M. Institute of Technology

## Department of Computer Science and Engineering


# CNN Based DeepFakes Detection Model For Forged Image Identification In Social Media

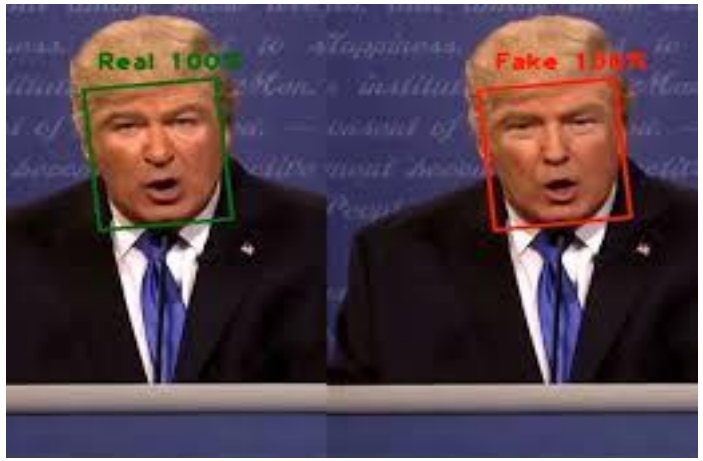
**GUIDED BY :**
Mrs. Priyanka Padki
Assistant Professor
Dept of CSE, BNMIT
Bangalore

**PRESENTED BY :**
Preethi V (1BG17CS072)
R Narendranath Reddy (1BG17CS075)
Suhas H (1BG17CS101)
Swarnamalya M (1BG17CS104)

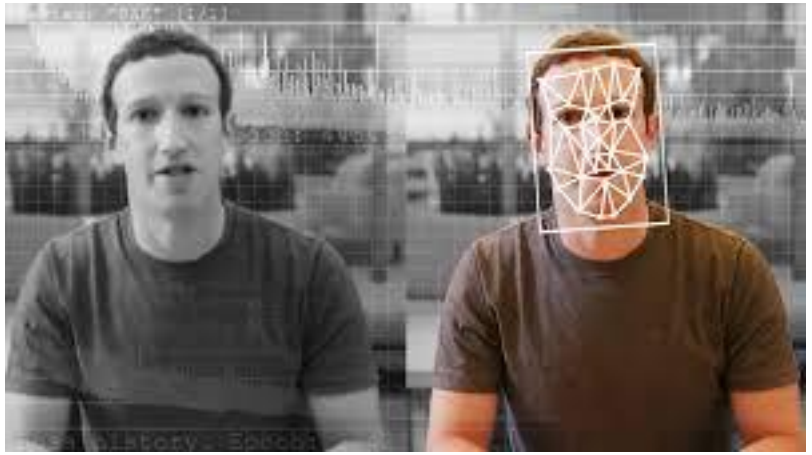
# GLIMPSE OF CONTENTS

- INTRODUCTION
- PROBLEM STATEMENT
- DETAILED LITERATURE SURVEY
- PROPOSED SYSTEM DESIGN
- DATASET DESCRIPTION
- DATA FLOW DIAGRAM
- RESULT ANALYSIS
- SUMMARY

# CNN BASED DEEPFAKES DETECTION MODEL FOR FORGED IMAGE IDENTIFICATON IN SOCIAL MEDIA







# DEEPFAKES : The Realistic Threat

- Photo manipulation was developed in the 19th century and soon applied to motion pictures. Technology steadily improved during the 20th century, and more quickly with digital video.

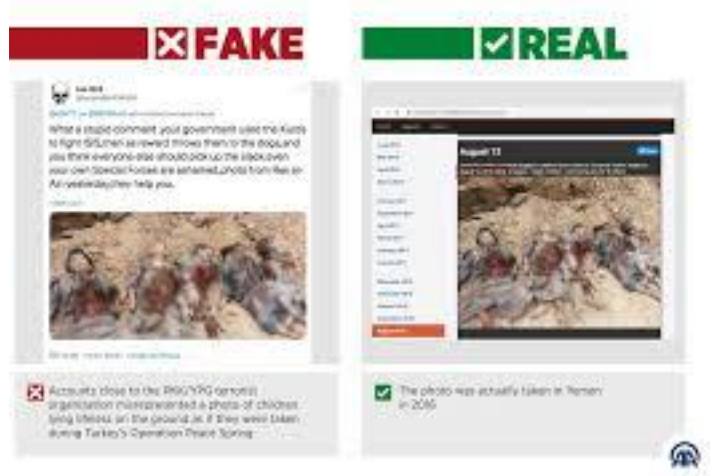
- Deepfakes refer to manipulated images, or other digital representations produced by sophisticated artificial intelligence, that yield fabricated images that appears to be real.

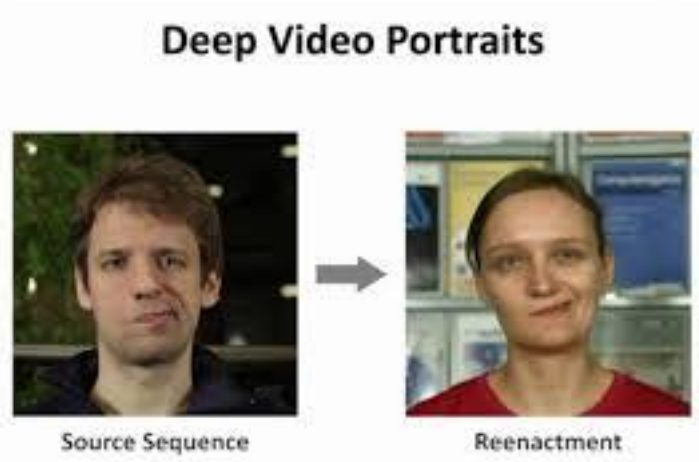

# DEEPFAKES : Potential Risks and Danger in Media

- Digital image acquisition is now a simple task and information in the form of digital images is drastically increasing on social media.

- Deepfakes that  leverage ML to manipulate images has garnered widespread attention for its fraudulent use in:

**Face swapping**
**Facial Reenactment**
**Pornographic images**
**Fake-news campaigns**
**Hoaxes and financial fraud.**



# PROBLEM STATEMENT

**WHAT**

- **Digital image manipulation** is the act of distorting the contents of an image
- Image forgeries has become a central problem in the last few years, especially after the advent of the so called **DeepFakes**
- These can be used for **malicious purposes in social media**
- In the long run , it may also reduce the **trust in journalism.**

## WHY

- This poses **enormous security threats** as they can be misused to abuse information and generate fake identification.
- Therefore, **detecting forged images in social media is critical** for protecting individuals from various misuses and the authenticity in media.
- The project **aims to implement an automated image forensic platform** that is capable of detecting forged and manipulated multimedia images.
- The tool is implemented by building a **CNN based automated model that utilizes Error Level Analysis (ELA).**

# DETAILED LITERATURE SURVEY

**[1] Chih-Chung Hsu , Chia-Yen Lee , Yi-Xiu Zhuang , " Learning to Detect Fake Face Images in the Wild" , IEEE 2018 , International Symposium on Computer, Consumer and Control (IS3C)**

- The main aim of this system is to **develop a deep forgery discriminator (DeepFD ).**
- It implements a **deep neural network based discriminator** that adopts the technique of **contrastive loss** .
- The proposed method has **two learning phases.**
- Experimental results demonstrate that the proposed DeepFD **successfully detected 94.7% fake images.**

**[2] Yuanfang Guo, Xiaochun Cao, Wei Zhang, Rui Wang, Member, " Fake Colorized Image Detection " , Submitted To IEEE Transactions On Information Forensics And Security, 2019**

- This system aims to **detect image colorization** in which grayscale images are colorized with realistic colors.
- Colorized images which are generated by state-of-the-art methods posses **statistical differences for the hue and saturation channels.**
- The two simple and effective detection methods proposed for fake colorized images are: **FCID-HIST and FCID-FE.**

**[3] Savita Walia & Krishan Kumar ,” Digital image forgery detection: a systematic scrutiny “, Australian Journal of Forensic Sciences, 2019**

- The aim of this system on **systematic survey** is to gain insights into the current research on the **detection of image forgeries** by comprehensively analysing the methods to implement the detection process.
- **Active methods** : **Digital watermarking and digital signatures**
- **Passive Methods** : The image processing operations such as **noise variation, lighting and shadows**
- **Copy-move forgery Detection Methods**: The major goal is to **match the regions in the image**.
- **Splicing Based Methods**: Bi-coherence features, camera response function, **DCT and DWT coefficients.**

**[4] Khurshid Asghar, Xianfang Sun4 , Paul L. Rosin4 , Mubbashar Saddique2 , Muhammad Hussain3, " Edge–texture feature-based image forgery detection with cross-dataset evaluation", Springer-Verlag GmbH Germany, part of Springer Nature 2019**

- The proposed system is composed of four major components, i.e., **(i) preprocessing, (ii) feature extraction, (iii) classifcation model building and (iv) testing, using the trained model with cross-dataset validation.**
- The model is **trained using an SVM classifer** on a set of images .
- A **cross-validation (CV) protocol** is used to divide each dataset or combination of datasets into k-fold (tenfold).
- **DRLBP** is a robust texture descriptor, which models the structural changes.

**[5] Khurshid Asghar, Zulfiqar Habib & Muhammad Hussain "Copy-move and splicing image forgery detection and localization techniques: a review ", Australian Journal of Forensic Sciences, 2019**

This paper aims to explore various forged image detection techniques employed for **copy-move and image splicing .**

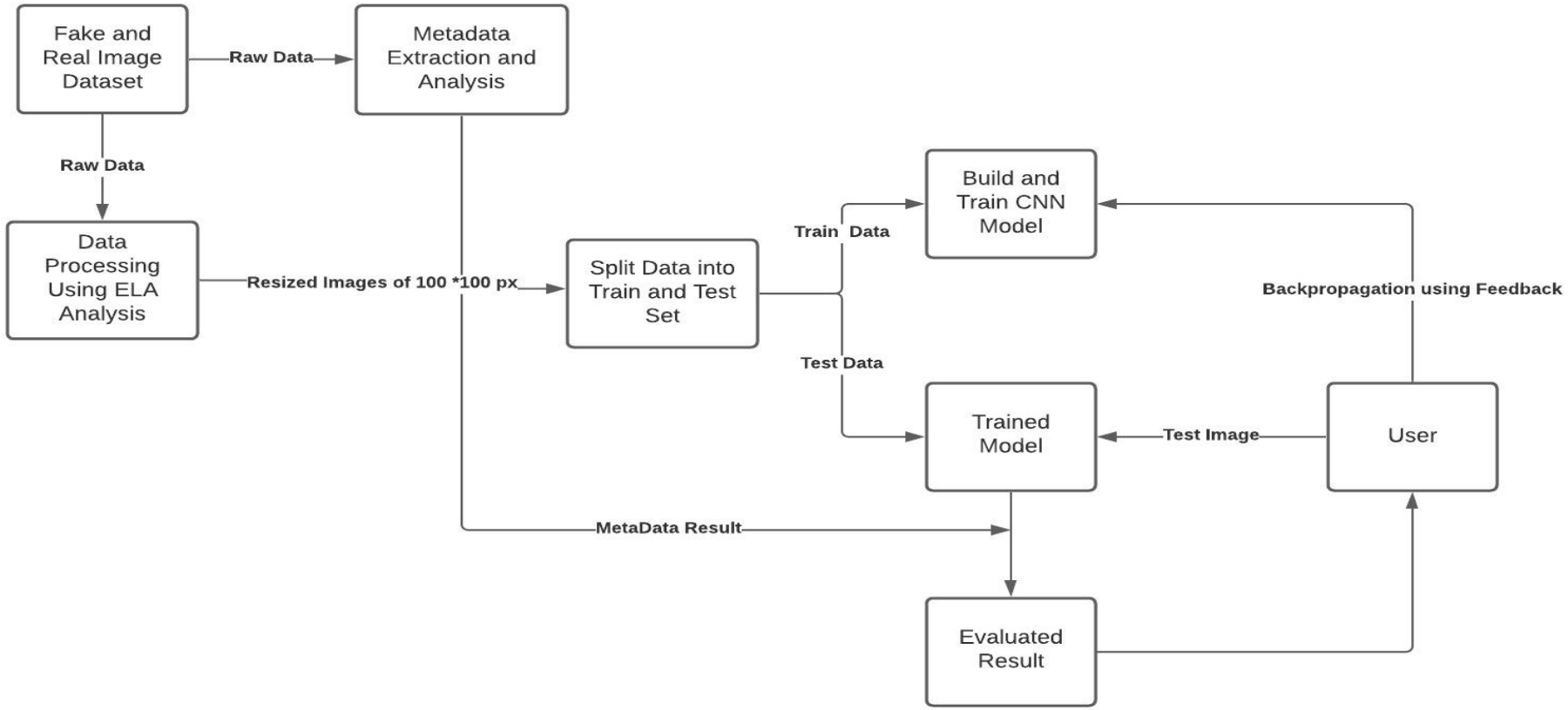
- **Texture and Intensity based Algorithms:** The **statistical analysis of pixels of small overlapped blocks** of an image.
- **SVD based Algorithms:** The proposed copy-move forgery detection algorithm is based on a **discrete wavelet transform (DWT) and SVD.**
- **PCA-based algorithms** : Principle Component Analysis (PCA) is to **extract image features** and has been used to detect copy-move forgery and spliced images.
- **DCT-based algorithms**: Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are used.

# MAJOR DRAWBACKS AND SHORTCOMINGS

The aforementioned traditional approaches deployed in detecting the forged and tampered digital images is certain to fall short of the following demands.

- **Benchmarking and standard datasets**
- **Performance evaluation**
- **Robustness**

# OVERVIEW OF PROPOSED SYSTEM



## *Stage 1* : METADATA ANALYSIS

- This system provides two level analysis for the image. At first level, it checks the image metadata.
- Metadata provides information related to how the file was generated and handled. Metadata provides information about a picture's pedigree, including the type of camera used, color space information, and application notes.


## METADATA EXTRACTION

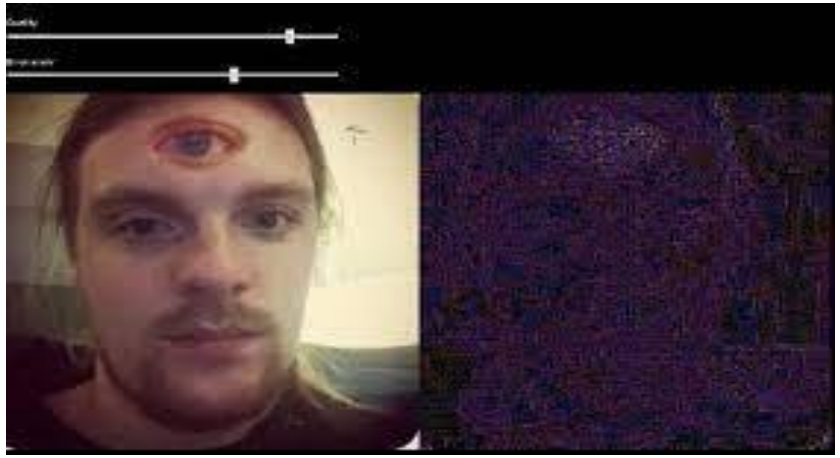
```
data = ImageMetadataReader.readMetadata(imageFile);
for every directories in data
        append the directory names to the extracted_data
        for every tag associated with the directory names
            append the tag to the extracted_data;
```

## METADATA ANALYSIS

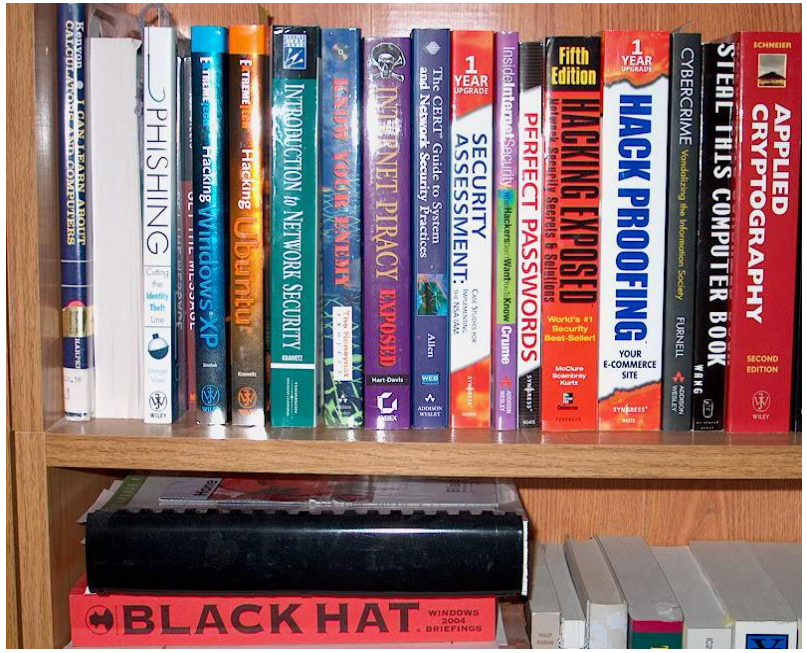
```
for  every string in extracted_data
        if string.toUpperCase() contains ("ADOBE") || (“PHOTOSHOP”) || (“GIMP”) ||
        (“COREL”) || (“PAINT”)
              fakeness += 6;

        if string contains ("Model") || (“Make”) || (“Exposure Time”) ||
        (“Exif IFD0”) || (“Focal Length”)
               real++;

      total = fakeness + real;
      fakenessPercentage = (float) fakeness / total;
      realPercentage = (float) real / total;
```

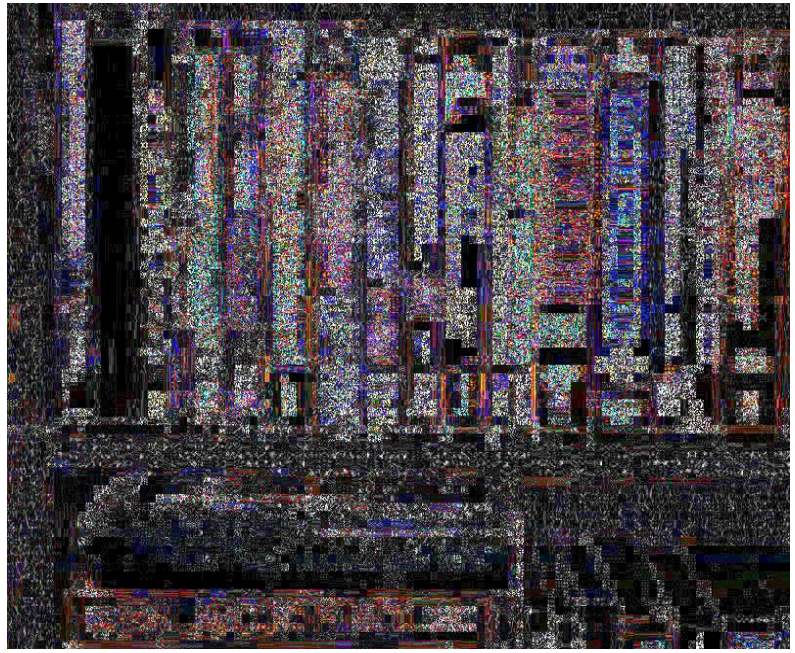
## *Stage 2* : DATA PREPROCESSING USING ERROR LEVEL ANALYSIS (ELA)

- **Error Level Analysis (ELA)** is a forensic technique on the image to analyze images through different **levels of compression**. This technique is used to find out digitally modified images.
- It works by **compressing the image to a 8X8 grid.**
- An **original digital photograph** has **high ELA values**
- A **digitally modified photograph** has **lower ELA values**

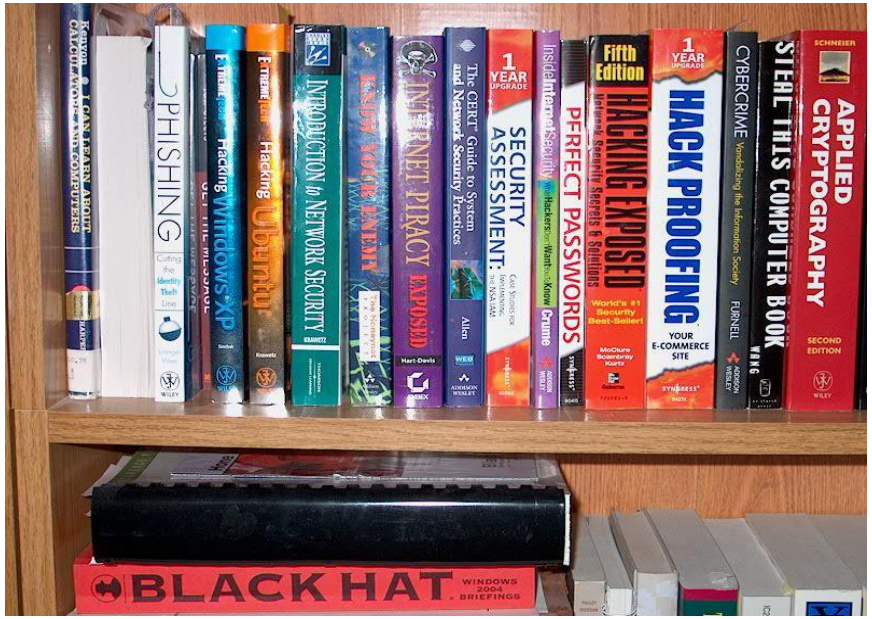



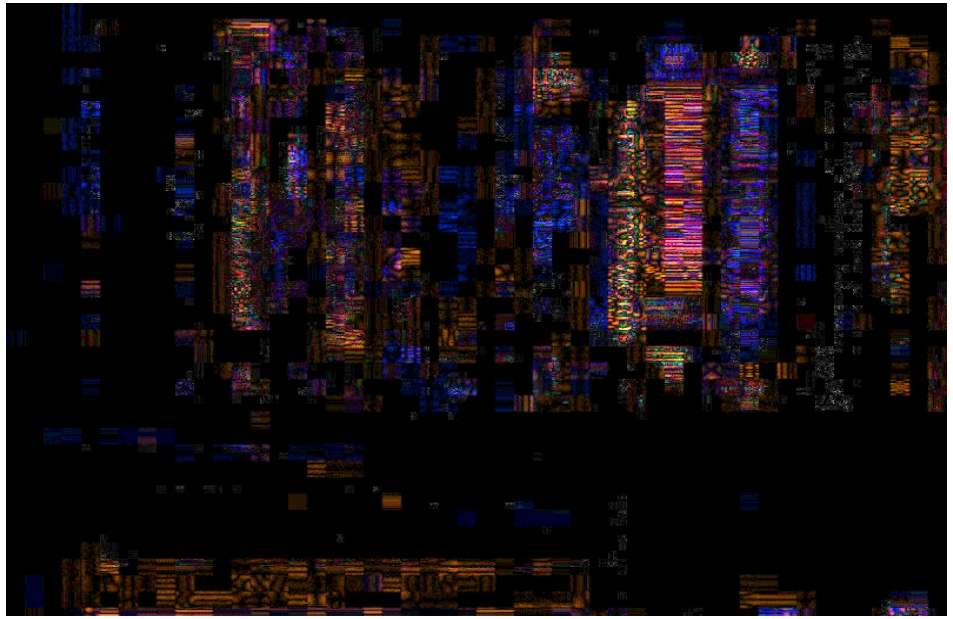
# ELA RESULT FOR ORIGINAL PHOTOGRAPH



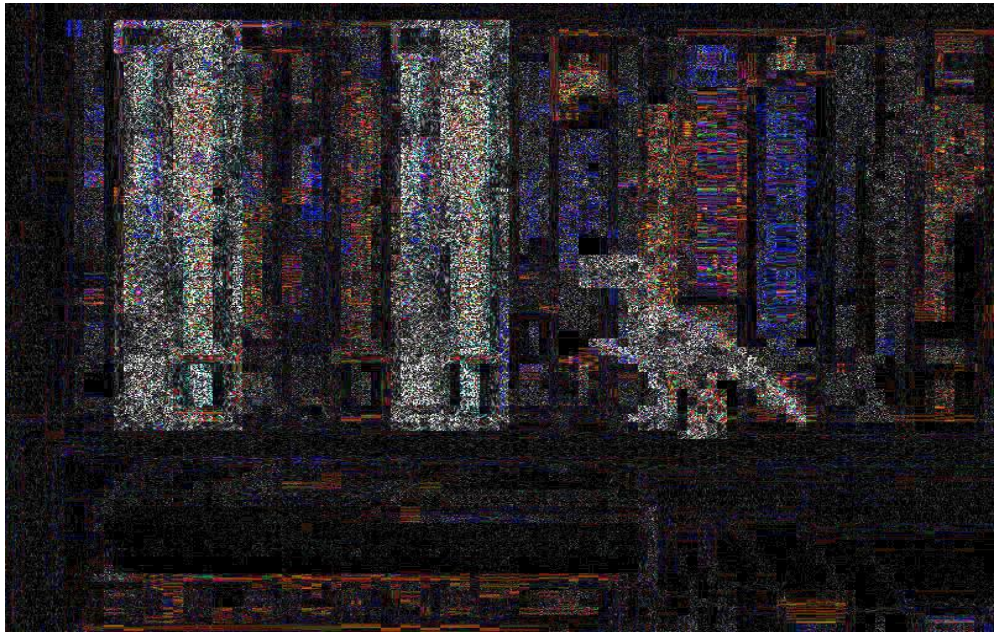



# ELA RESULT FOR RESAVED PHOTOGRAPH



# ELA RESULT FOR MODIFIED PHOTOGRAPH

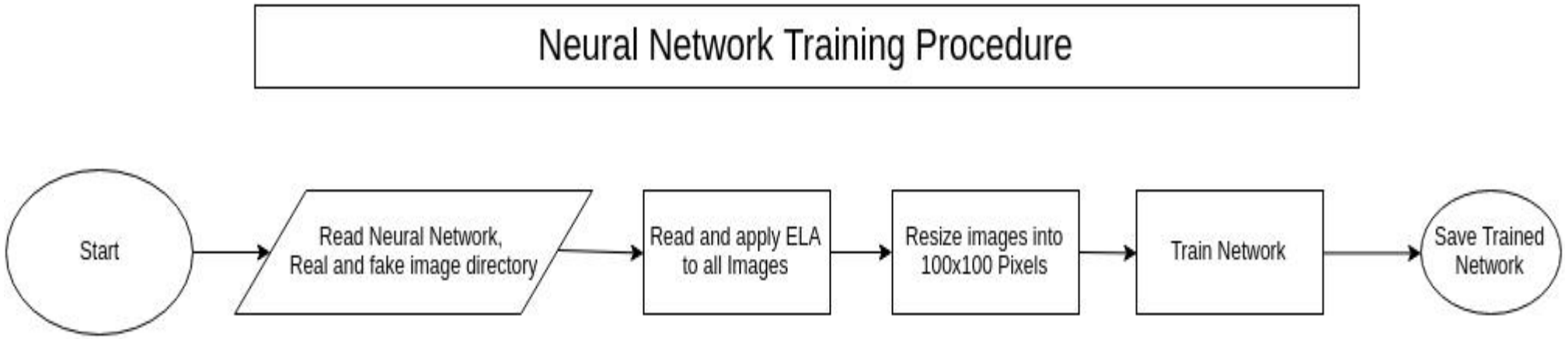
## ELA ANALYSIS

```
FileSaver fs = new FileSaver(orig);
setJpegQuality(100);
fs.saveAsJpeg(origPath);

setJpegQuality(95);
fs.saveAsJpeg(resavedPath);
ImagePlus resaved = new ImagePlus(resavedPath);

ImageCalculator calc = new ImageCalculator();
ImagePlus diff = calc.run("create difference", orig, resaved);
```

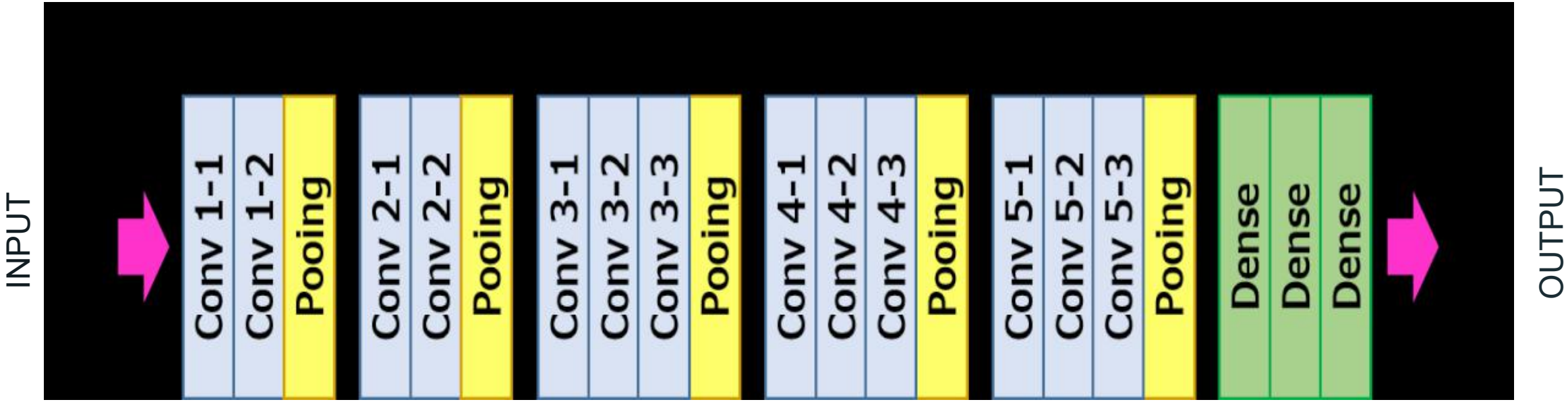
## *Stage 3* : BUILD AND TRAIN A CNN CLASSIFIER MODEL

- **Convolutional Neural Network** , CNN for short, is a specialized type of **deep learning algorithm** designed for working with **two-dimensional image data**.

- CNN image classification takes the **output image yielded after ELA** preprocessing, **as the input image** and performs **processing , training and classifies** it under certain categories.



# VGG16 CNN ARCHITECTURE



## MODEL CONFIGURATION

```
model.add(Conv2D(input_shape=(100,100,3),filters=64,kernel_size=(3,3),padding="same", activation="relu"))
model.add(Conv2D(filters=64,kernel_size=(3,3),padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
```

```
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Flatten())
model.add(Dense(units=4096,activation="relu"))
model.add(Dense(units=4096,activation="relu"))
model.add(Dense(units=2, activation="softmax"))

model.compile(optimizer=Adam, loss=LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
```

## *Stage 4* : FEEDBACK ANALYSIS USING MOMENTUM BACKPROPAGATION LEARNING RULE

- It is a supervised learning rule that tries to minimize the error function using the inputs received by the feedback.

- The algorithm is used to effectively train a neural network through a method called chain rule.

$$\Delta w_{ij} = (\eta * \frac{\partial E}{\partial w_{ij}})$$

weight increment — learning rate — weight gradient

$$\Delta w_{ij} = (\eta * \frac{\partial E}{\partial w_{ij}}) + (\gamma * \Delta w_{ij}^{t-1})$$

momentum factor — weight increment, previous iteration

Propagate the errors backward through the network

```
 for every node in the output layer
   calculate the error signal
end

for all hidden layers
   for every node in the layer
      1. Calculate the node's signal error
      2. Update each node's weight in the network
   end
end
```

Setting the Momentum and Learning Rate

```
MomentumBackpropagation mBackpropagation ;
        mBackpropagation.setLearningRate(learningRate);
        mBackpropagation.setMaxError(maxError);
        mBackpropagation.setMomentum(momentum);
```

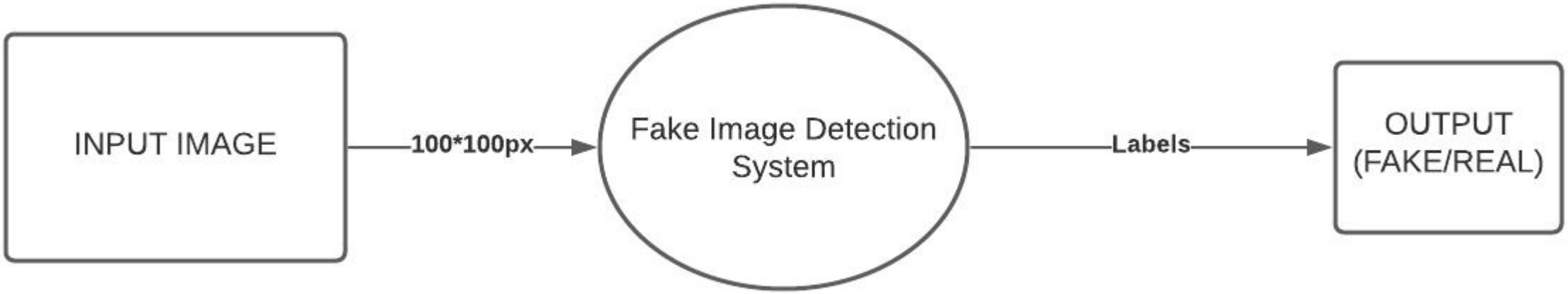
# DATASET DESCRIPTION

**REAL AND FAKE IMAGES TRAINING DATASET**

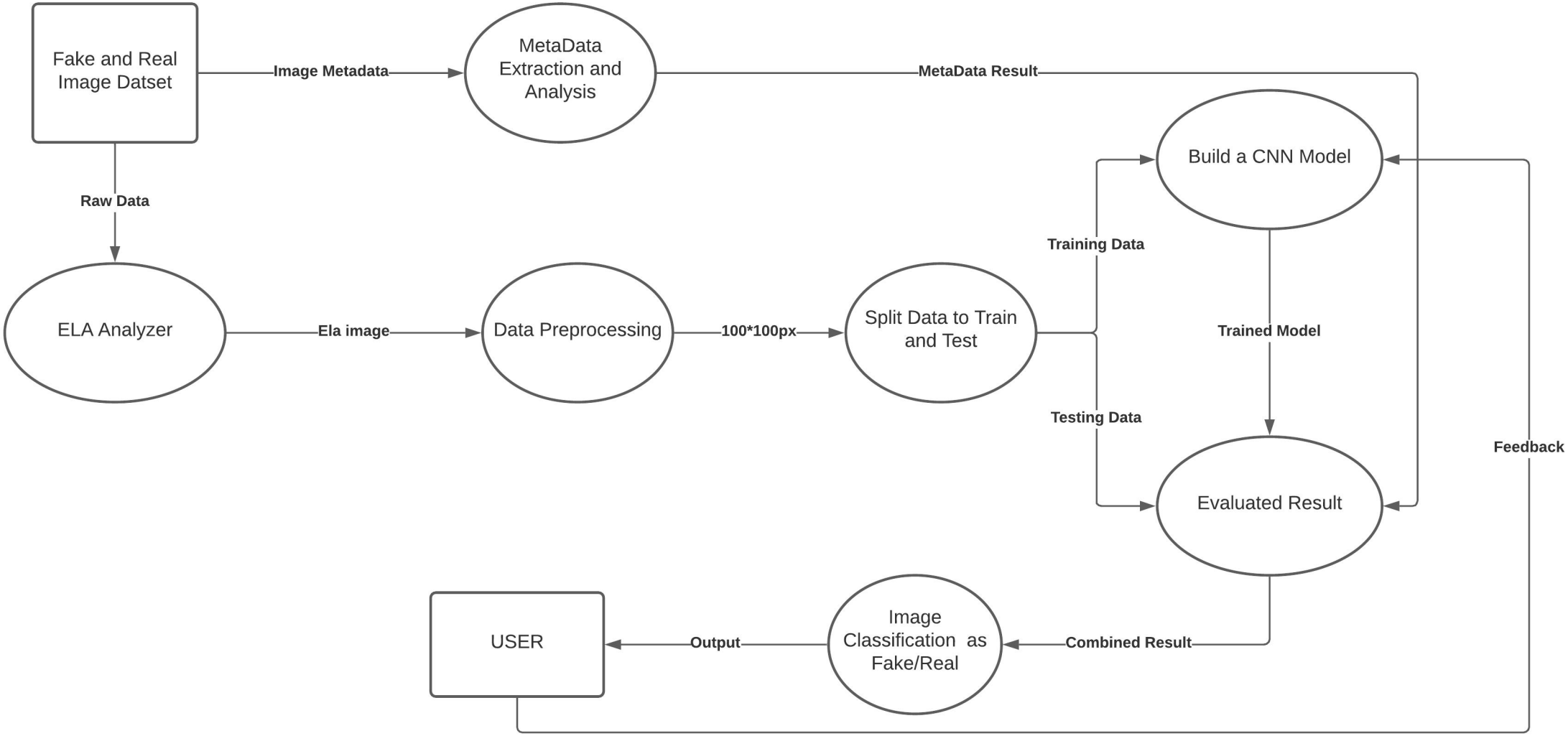
- Neural network is trained with CASIA dataset.
- The dataset contains 7491 real images and 5123 tampered images under varying sizes.
- From the dataset we have used 4000 real and fake images for training.

# DATA FLOW DIAGRAM

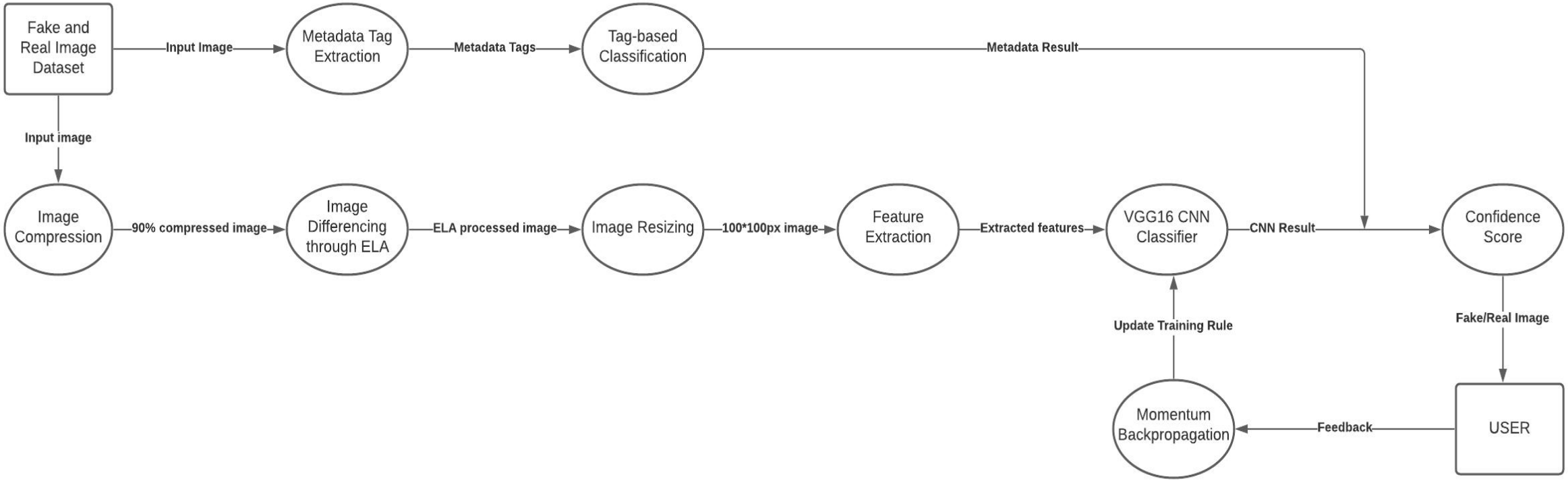
**LEVEL 0 DFD**



# Level 1 DFD


Fake and Real Image Datset
Image Metadata
MetaData Extraction and Analysis
MetaData Result
Raw Data
Build a CNN Model
Training Data
ELA Analyzer
Ela image
Data Preprocessing
100*100px
Split Data to Train and Test
Trained Model
Testing Data
Feedback
Evaluated Result
USER
Output
Image Classification as Fake/Real
Combined Result

# Level 2 DFD


Fake and Real Image Dataset
Input Image
Metadata Tag Extraction
Metadata Tags
Tag-based Classification
Metadata Result
Input image
Image Compression
90% compressed image
Image Differencing through ELA
ELA processed image
Image Resizing
100*100px image
Feature Extraction
Extracted features
VGG16 CNN Classifier
CNN Result
Confidence Score
Update Training Rule
Fake/Real Image
Momentum Backpropagation
Feedback
USER

# TEST SCENARIOS

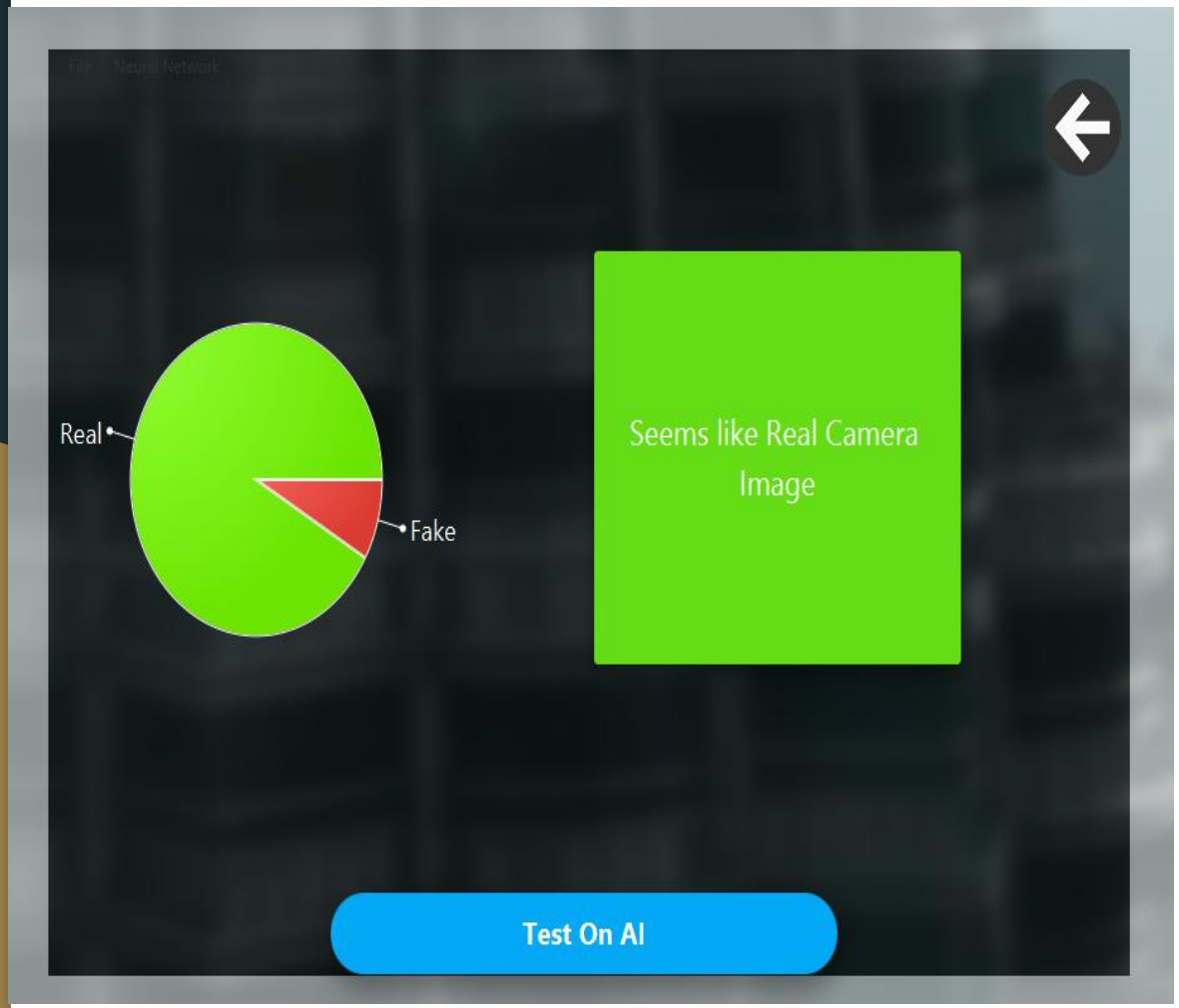
The fake Image Detector tool can be extensively used to detect the following test scenarios:

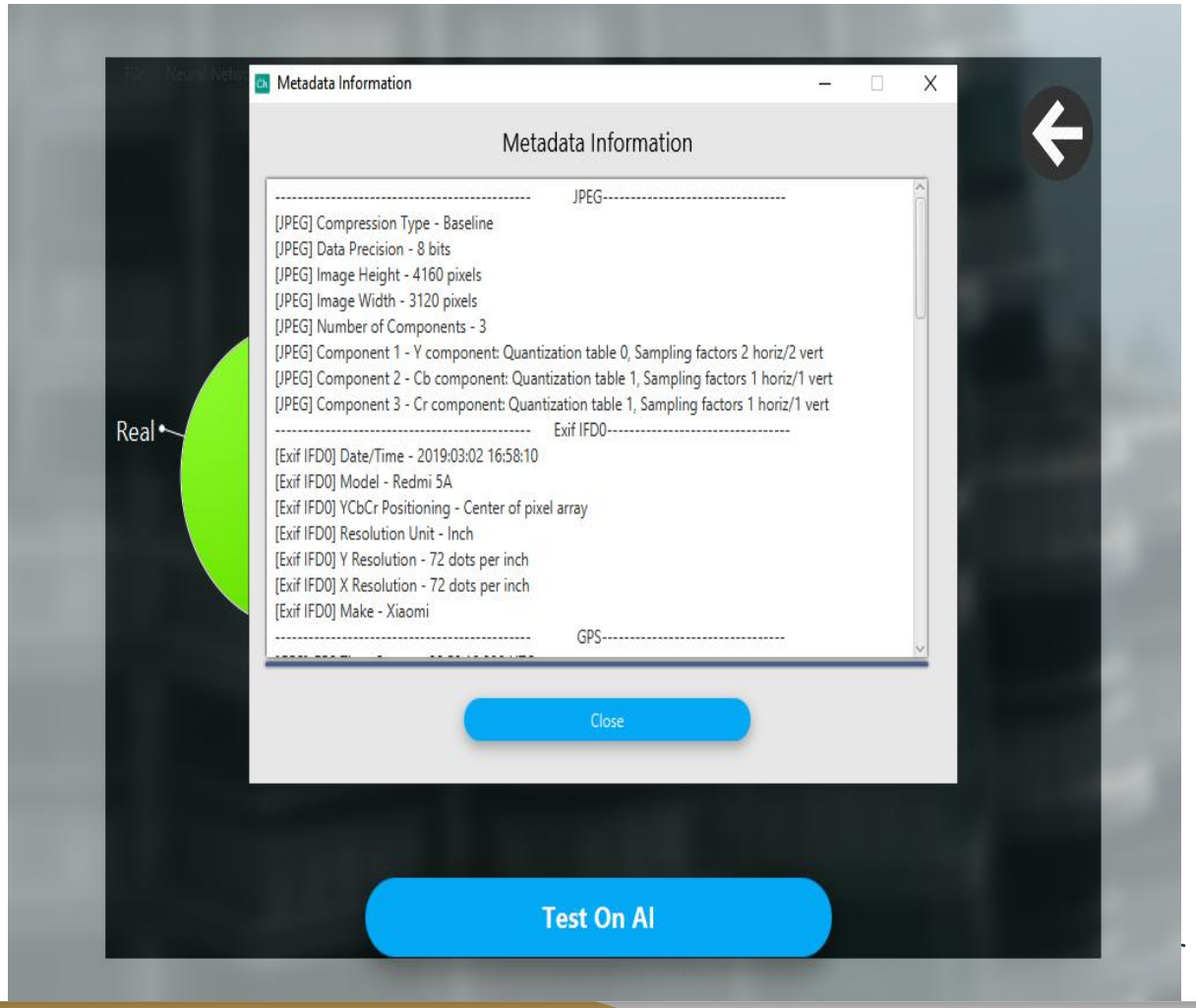
- Facial images
- News content
- Documents like Aadhar , Pan card and other proofs
- Scenic images , animal and plant images , textures , patterns

# RESULT ANALYSIS

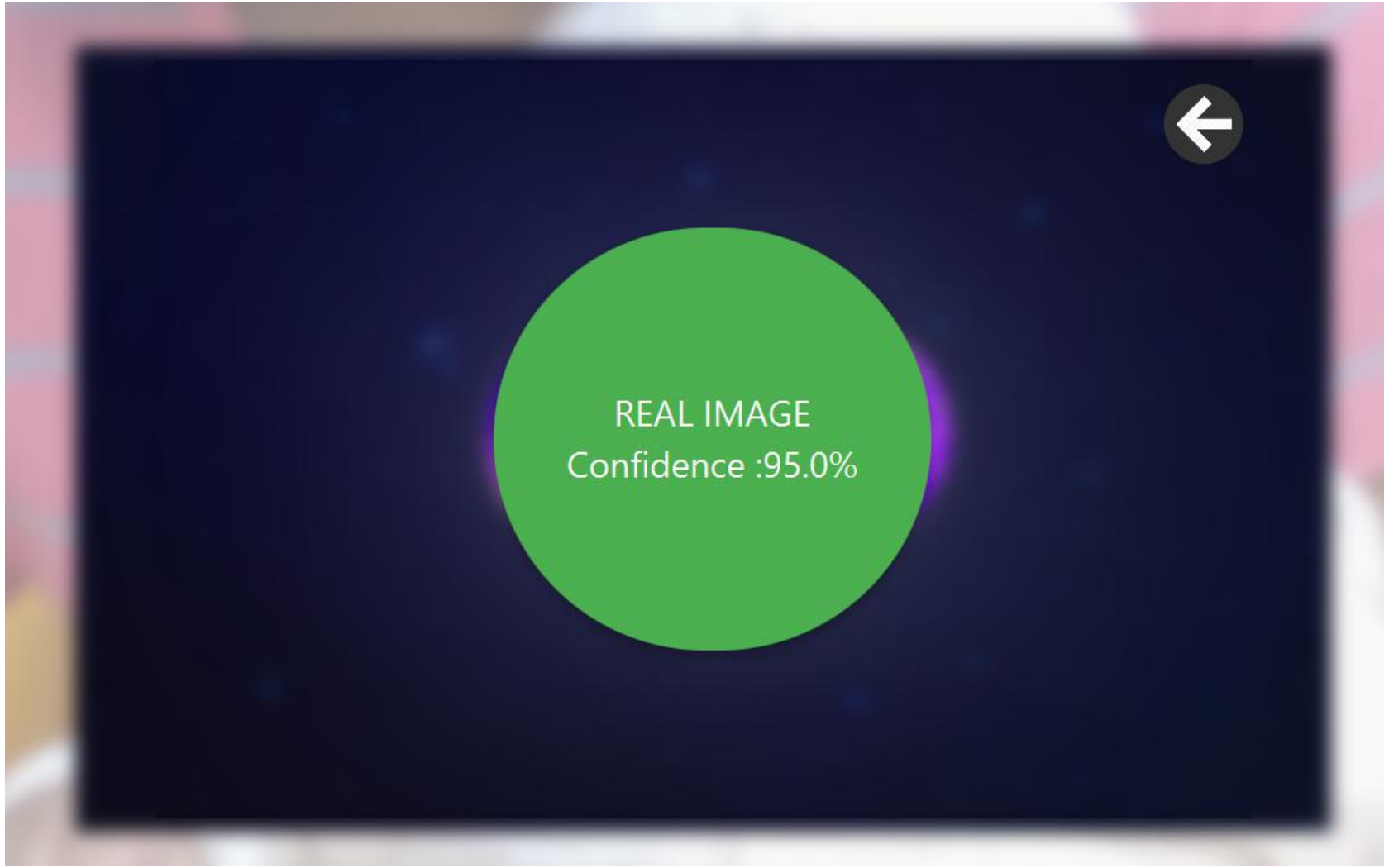
**CASE 1 : REAL IMAGE**

## METADATA ANALYSIS



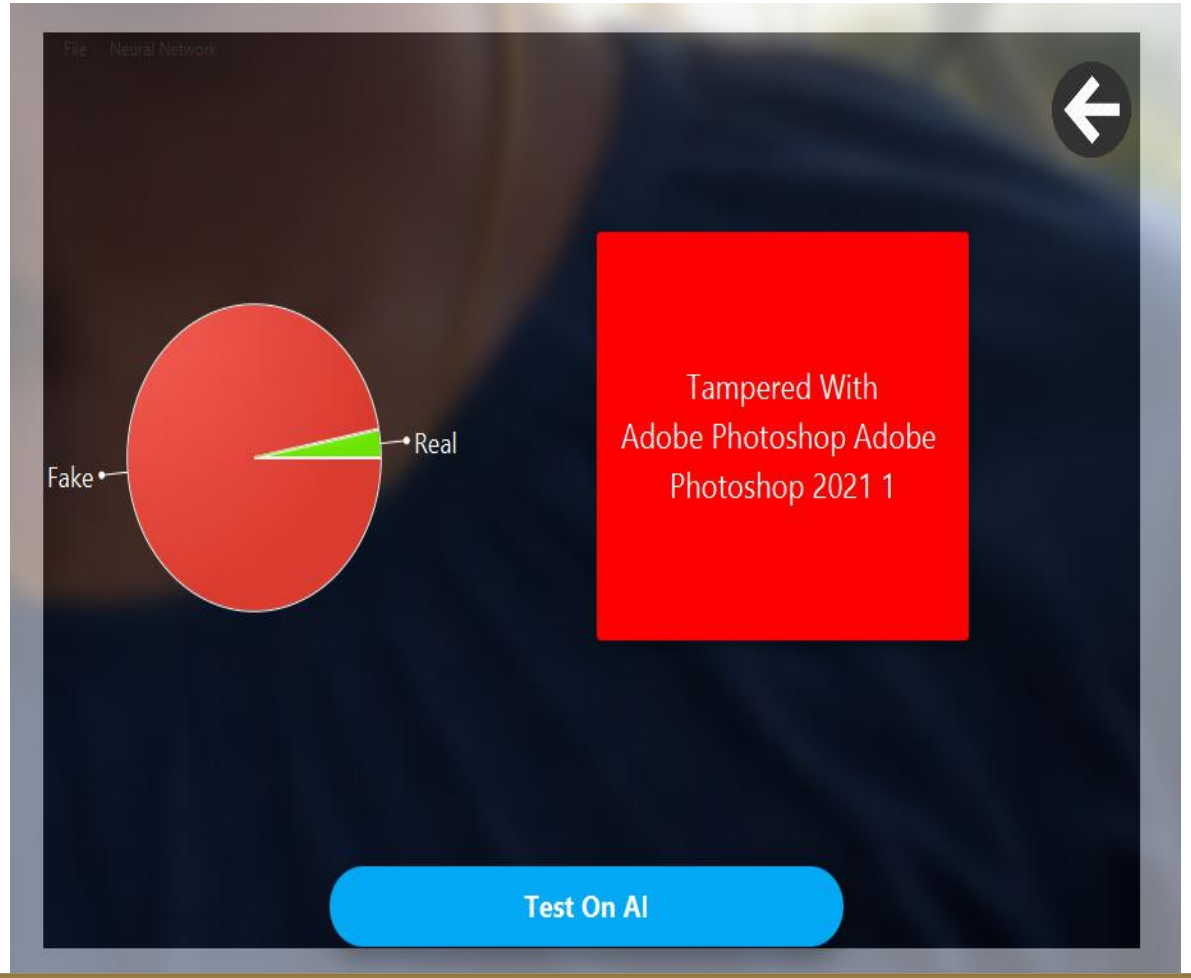



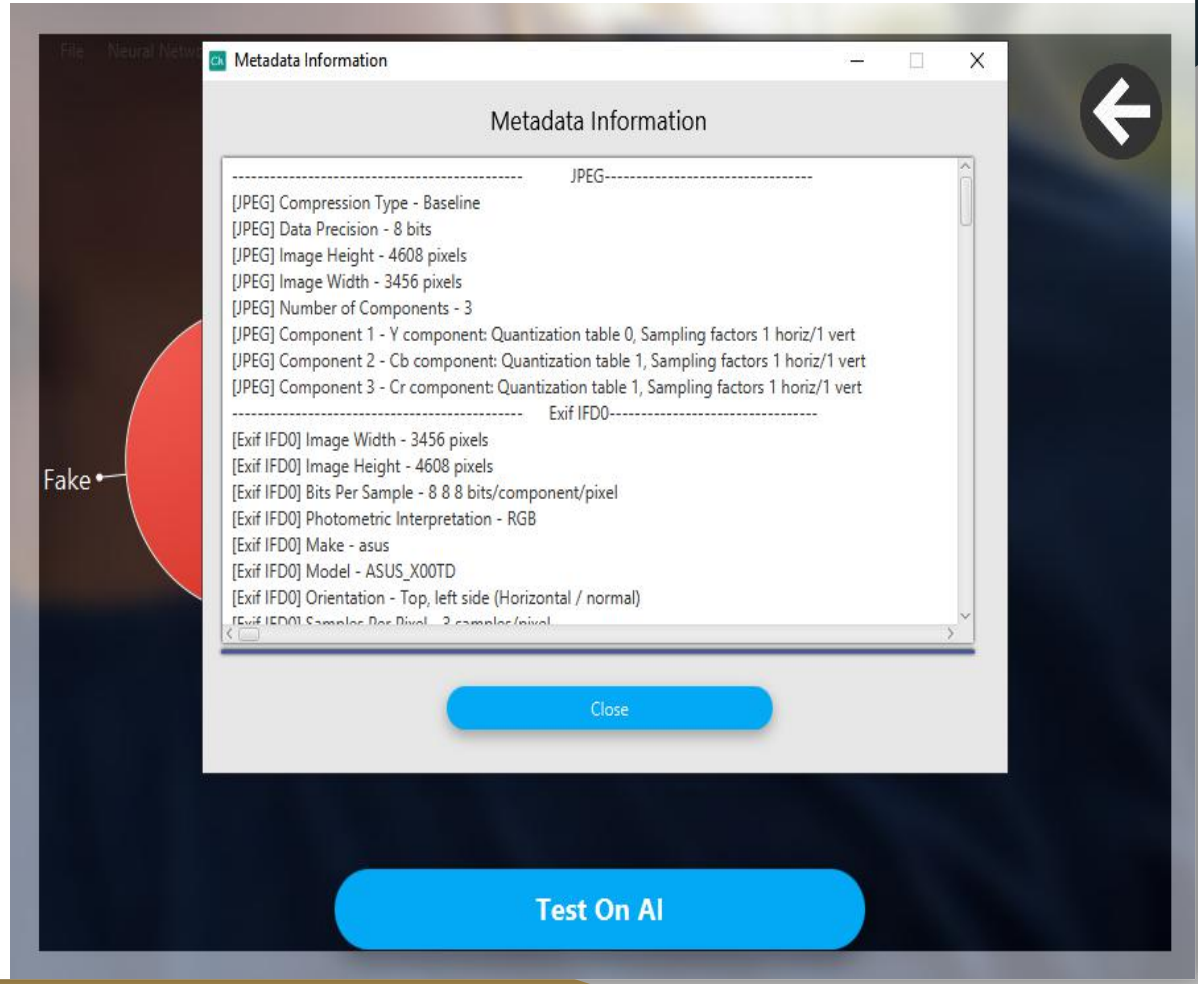
## NEURAL NETWORK ANALYSIS



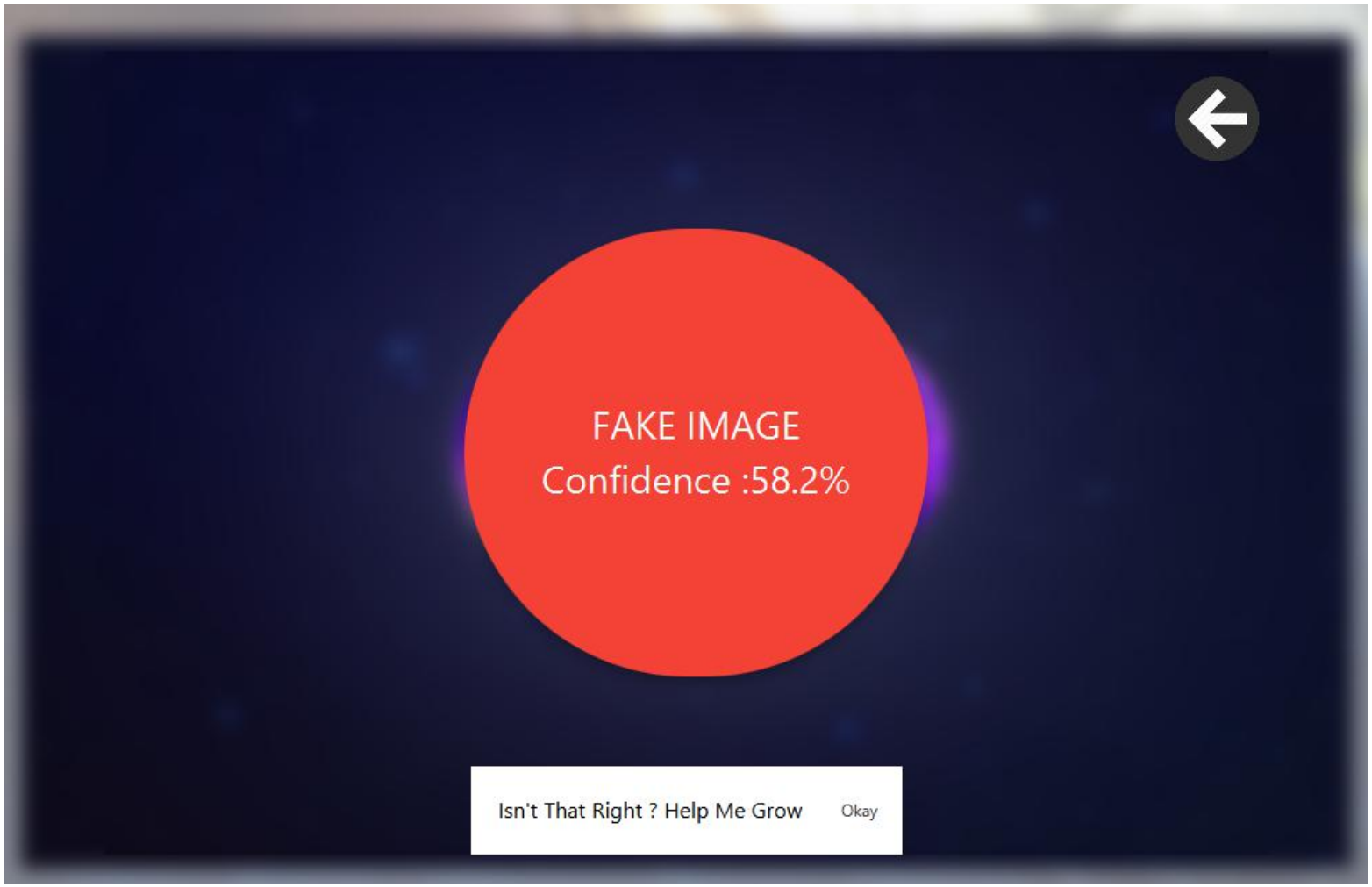
## CASE 2 : FAKE IMAGE

# METADATA ANALYSIS



## NEURAL NETWORK ANALYSIS



## RESULT COMPUTATION

```
NEURAL_NET_WEIGHT = 0.7f;
METADATA_NET_WEIGHT = 1-NEURAL_NET_WEIGHT;

real = (real*NEURAL_NET_WEIGHT) + (METADATA_NET_WEIGHT * ConstantObjects.realness*100);
fake = (fake*NEURAL_NET_WEIGHT) + (METADATA_NET_WEIGHT * ConstantObjects.fakeness*100);

if (fake > real)
navigation_button.setText("FAKE IMAGE" + "\nConfidence :" + df2.format(fake) + "%");

else if (fake < real)
navigation_button.setText("REAL IMAGE" + "\nConfidence :" + df2.format(real) + "%");

else
navigation_button.setText("It is equally likely to be real or fake");
```

# PERFORMANCE EVALUATION

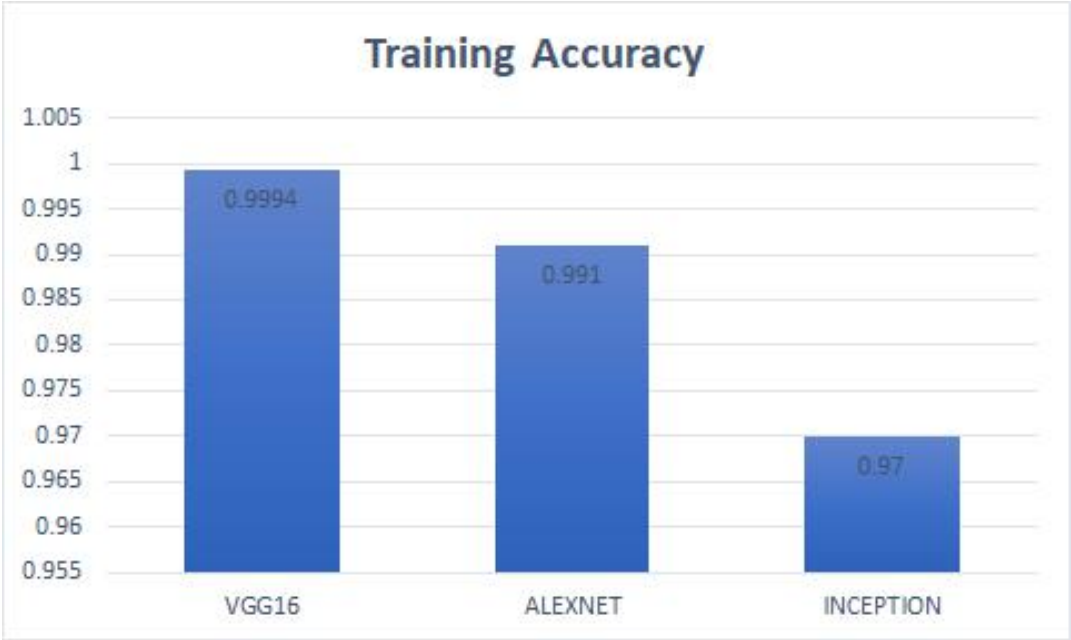
The models under comparison are:

- **VGG16**
- **AlexNet**
- **Inception**

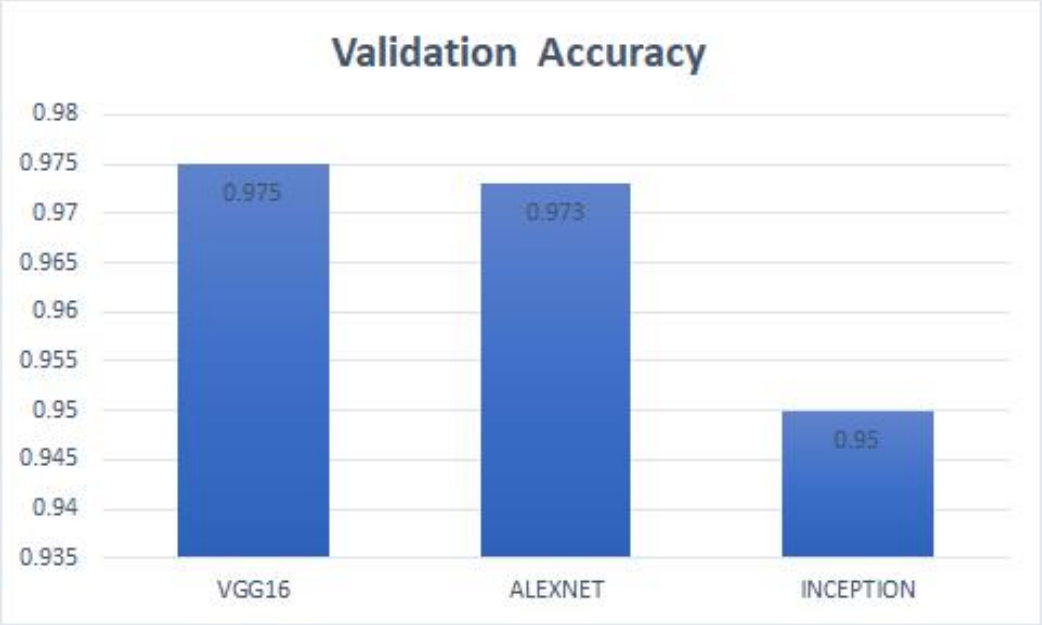
| TRAINING MODELS | TRAINING TIME (in seconds) | TRAINING ACCURACY | VALIDATION ACCURACY |
|---|---|---|---|
| **VGG16** | 900 | 0.9994 | 0.975 |
| **ALEXNET** | 1100 | 0.991 | 0.973 |
| **INCEPTION** | 3000 | 0.97 | 0.95 |

- **Model Training Time** : The time costing of 100 epochs of each model are compared and VGG16 has got the least training time.

- **Model Training Performance :** The training accuracy of various models are assessed and is highest for VGG 16 .



- **Model Validation Performance** : The validation accuracy of the various models are assessed.



# REAL-TIME APPLICATIONS

The fake Image Detector tool can be extensively used in the following scenarios:

- Monitor Images in Social Media that includes **TWITTER, FACEBOOK , INSTAGRAM.**
- In the field of **JOURNALISM** to ensure the credibility of the news.
- As a **FAKE DOCUMENT DETECTOR** to verify the authenticity of the documents proof like the aadhar card, pan card etc.
- In **MATRIMONIAL WEBSITES** to prevent being deceived by the fake images.

# FUTURE ENHANCEMENTS

- Increase in Confidence Score
- Model Upgradation
- Improve Processing Time with Resource Upgradation
- Improve the Look and Feel Of UI

# SUMMARY

- In this project survey, we **propose an automatic image forensic platform based on ELA and the deep learning techniques of CNN** to detect if the image is authentic or forged.
- First, the dataset comprising of tampered images and original images undergoes **processing** using the ELA method.
- The pre-processed images of the dataset is further **divided as training and test data**. The training data is utilized for training the model.
- We choose to use **VGG 16 architecture** of CNN because VGG 16 is perfect for training with minimal datasets. It results in a more precise classifier model.
- The test data is now given to the trained model to **experiment the accuracy** of the predicted results which yields as fake or original.

# THANK YOU