Enhancing Digital Image Forgery Detection Using Transfer Learning

Dr. N Pradeep Kumar, Yarakala Sreeja, Boya Srija, Gona Teja Sree

[1]Assistant Professor, Department Of Ece, Bhoj Reddy Engineering College For Women, India.

[2,3,4]B. Tech Students, Department Of Ece, Bhoj Reddy Engineering College For Women, India.

ABSTRACT

Nowadays, digital images are a main source of shared information in social media. Meanwhile, malicious software can forge such images for fake information. So, it's crucial to identify these forgeries. This problem was tackled in the literature by various digital image forgery detection techniques. But most of these techniques are tied to detecting only one type of forgery, such as image splicing or copy-move that is not applied in real life. This paper proposes an approach, to enhance digital image forgery detection using deep learning techniques via CNN to uncover two types of image forgery at the same time, The proposed technique relies on discovering the compressed quality of the forged area, which normally differs from the compressed quality of the rest of the image. A deep learning-based model is proposed to detect forgery in digital images, by calculating the difference between the original image and its compressed version, to produce a featured image as an input to the pre-trained model to train the model after removing its classifier and adding a new fine-tuned classifier. A comparison between eight different pre-trained models adapted for binary classification is done. The experimental results show that applying the technique using the adapted eight different pre-trained models outperforms the state-of-the-art methods after comparing it with the resulting evaluation metrics, charts, and graphs. Moreover, the results show that using the technique with the pre-trained model MobileNetV2 has the highest detection accuracy rate (around 95%) with fewer training parameters, leading to faster training time.

1-INTRODUCTION

The tampering of a digital image is called digital image forgery; these forged images cannot be detected by the naked eye. Such images are the primary sources of spreading fake news and misleading information in the context of society with the aid of diverse social media platforms like Facebook, Twitter, etc. The editing software tools that can make these forgeries are available for free with some advanced features that are used for image tampering such as GNU, GIMP, and Adobe Photoshop. Such forgeries can be detected using digital image forgery algorithms and techniques, these algorithms are used in image security especially when the original content is not available. Digital image forgery means adding unusual patterns to the original images that create a heterogeneous variation in image properties and an unusual distribution of image features. Figure 1 shows the classification of digital image forgery. Active approaches require essential information about the image for the verification process. The inserted information within the picture is employed to observe the modification in that picture. The active approach consists of two types: digital signatures which insert some additional data obtained from an image by the end of the acquisition process, and digital watermarking which is inserted into images either during the acquisition phase or during the processing phase.

methodologies do not require past information about the image. These approaches exploit that the tampering actions modify the contents of information of the image that can facilitate tampering detection

2-DEEP LEARNING AND CNNS IN IMAGE FORGERY DETECTION

The rapid advancement of digital image editing tools has made it easier than ever to manipulate images, leading to concerns about authenticity and integrity. Image forgery detection has become a critical field in digital forensics, addressing the challenges posed by such manipulations. Forgeries can take various forms, including splicing, copy-move, and retouching, all of which can be used for malicious purposes such as spreading misinformation, defaming individuals, or tampering with evidence.

Traditional image forgery detection methods relied on handcrafted features and statistical analysis to identify inconsistencies in image properties. These approaches often used classifiers like Support Vector Machines (SVMs) and Naïve Bayes to distinguish between authentic and tampered regions. However, the effectiveness of these methods was limited by their dependency on feature engineering and their inability to adapt to complex forgeries.

Recent advancements in machine learning and deep learning have revolutionized the field. Techniques such as Convolutional Neural Networks (CNNs) and transfer learning have significantly improved the accuracy and scalability of forgery detection systems. These models can automatically extract meaningful features from images, enabling them to detect intricate manipulations with higher precision. Additionally, the integration of pre-trained networks and hybrid approaches has further enhanced the capability to detect multiple types of forgeries, making modern systems more robust and reliable.

This chapter provides a comprehensive overview of the current state-of-the-art techniques in image forgery detection, focusing on deep neural network-based methods, pretrained networkbased approaches, and the challenges and future directions in the field.

In the digital age, images serve as a powerful medium of communication, widely used across social media, journalism, legal evidence, and more. However, the increasing accessibility of sophisticated image editing tools has raised serious concerns about the authenticity and integrity of digital images. Image forgery, the act of altering an image to mislead or manipulate its interpretation, has become a prevalent issue with implications in security, legal proceedings, journalism, and public trust.

Image forgery encompasses a range of manipulative techniques, including splicing, where segments from multiple images are combined; copy-move, which involves duplicating and repositioning regions within an image; and retouching, which subtly alters an image's content for enhancement or deception. These manipulations are often employed maliciously for creating fake news, defaming individuals, tampering with legal evidence, and promoting propaganda, making their detection a critical need in various domains.

The implications of image forgery extend across multiple fields. For instance, in journalism, manipulated images can mislead public opinion or distort facts. In legal systems, tampered evidence can compromise justice, while on social media, fake images can rapidly spread misinformation, inciting public unrest. Thus, image forgery detection is pivotal in ensuring the reliability and integrity of visual content.

Early approaches to image forgery detection relied heavily on handcrafted features and statistical methods. These techniques aimed to identify inconsistencies in image properties, such as pixellevel anomalies, noise patterns, and compression artifacts. Classifiers like Support Vector Machines (SVMs) and Naïve Bayes were commonly employed to distinguish between genuine and manipulated regions. While effective for simple manipulations, these methods struggled with complex and high-quality forgeries due to their reliance on manual feature engineering.

Deep Neural Network-Based Techniques

Deep Neural Networks (DNNs) have become a cornerstone in the development of image forgery detection techniques due to their remarkable ability to autonomously extract hierarchical features from image data. These techniques bypass the need for handcrafted features, allowing for the detection of subtle, high-dimensional patterns that are often indicative of forgeries. This section delves into the various applications of DNNs in detecting splicing, copy-move, and combined manipulations, emphasizing the advancements and strategies adopted for robust and accurate detection. For splicing and copy-move together, a multimodal system was proposed in, which covers classification and localization, forgery detection through a deep neural network followed by part-based image retrieval classification. The localization of manipulated regions was accomplished using a deep neural network. InceptionV3 was employed for feature extraction. The Nearest Neighbor Algorithm was used to retrieve Potential donors and nearly duplicates. In a novel approach to detect copy move and splicing image forgery using a Convolutional Neural Network (CNN), with three different models was presented, namely, ELA (Error Level Analysis), VGG16, and VGG19. The proposed method applied the pre- processing technique to obtain the images at a particular compression rate. These images were then utilized to train the model, where the images were classified as authentic or forged.

3-PROPOSED METHOD

The proposed approach considers the fact shown in, that copying a part of an image from one to another may impose some changes in the image properties due to the different sources of the images. Although these changes may not be detectable to the human eye, they can be detected by CNNs in manipulated images. The proposed model aims to avoid all of the forementioned drawbacks, by adapting the idea of calculating the difference in compression qualities to produce the featured image as an input to a deep neural network with the assistance of a pretrained model to benefit from the power of transfer learning. As a result, the evaluation matrix will be improved including the accuracy rate that will get better than that which was recorded when using CNN.

This will be elaborated and discussed in the following section. In a forged image, if the image is compressed, the forged section of the image will be compressed differently than the rest of the image. This is because the source of the original image differs from the source of the forged section. When analyzing the difference between the original image and its compressed version, the forgery component becomes more distinguished. Therefore, this aspect can be utilized

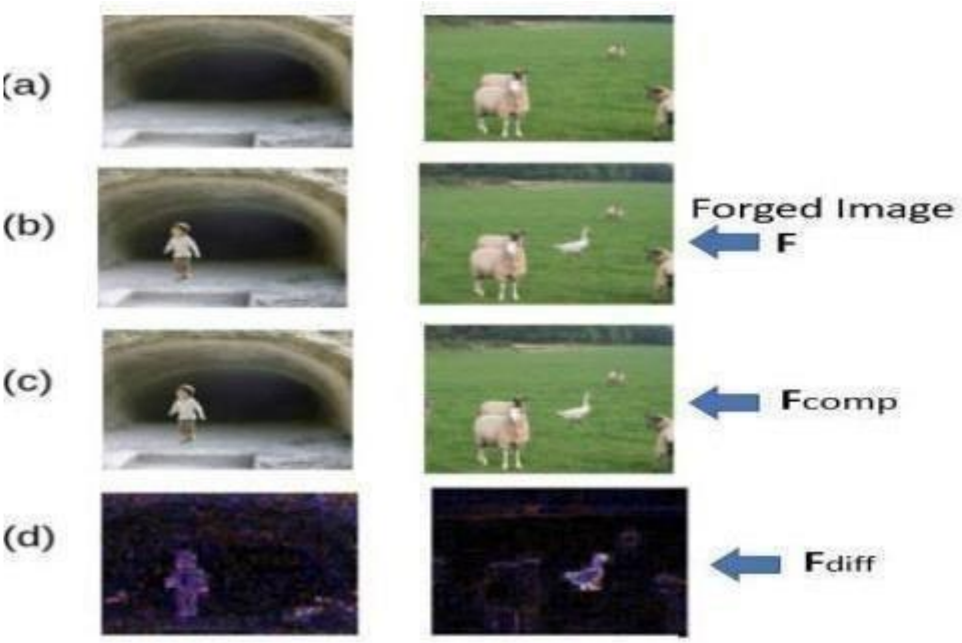


Fig 1 Set of images created in the proposed work.

As a result, the forged part of the image appears in (Fdiff) due to the difference between the source of the forged and original parts. Fdiff is then reshaped to 160×160 pixels to fit as an input feature image for training a pre-trained model (M), which is then used to classify images as forged or authentic. Shows the overall architecture of the proposed system.

The pre-trained model, shown as block (M), is used to extract features from input images (Fdiff) and classify them as authentic images or forged images. In this block (pre-trained model), eight different pre-trained models are considered (one at a time) namely, VGG16, VGG19, Reset, DenseNet, Xception, and MobileNet for fine training with input images (Fdiff), to nominate the model with the best performance among them.

Each model of the forementioned eight pre-trained models has its own architecture which consists of a set of convolutional layers with activation functions and ends with a set of fully connected layers that can classify up to 1000 classes of images. So, each model architecture has to be modified to fit the binary classification problem with only two classes (authentic or forged images) as in the case of image forgery detection problems. Therefore, the native fully connected layers in each model are replaced with a new set of fully connected classification layers able to handle the binary classification problem at hand. The convolutional layers in every model should remain untouched since they contain all the trainable parameters used in transfer learning shows the detailed architecture of the proposed model classifier with the newly added layers.

After removing the fully connected layers of the pre-trained model, a flatten layer is added to convert the input data, which is typically a multi-dimensional array, into a one-dimensional vector that can be fed to the next layers. The next two (new) layers are fully connected layers added with the ReLU activation function. The two layers have 1024 and 256 neurons, respectively. After each layer, a dropout

0.5 was added to prevent overfitting by randomly dropping out (setting to zero) about 50% of the output values of the previous layer will be randomly set to zero during the training phase. The last fully connected layer with a sigmoid activation function is added, which is the common activation function used in binary classification problems.

1.1 System Architecture

Figure 3.3 shows the overall architecture of the proposed system. In Figure 3.2, the pretrained model, shown as block (M), is used to extract features from input images (Fdiff) and classify them as authentic images or forged images. In this block (pre-trained model), eight different pre-trained models are considered (one at a time) namely, VGG16, VGG19, ResNet, DenseNet, Xception, and MobileNet for fine training with input images (Fdiff), to nominate the model with the best performance among them. Each model of the forementioned eight pretrained models has its own architecture which consists of a set of convolutional layers with activation function and ends with a set of fully connected layers that can classify up to 1000 classes of images.

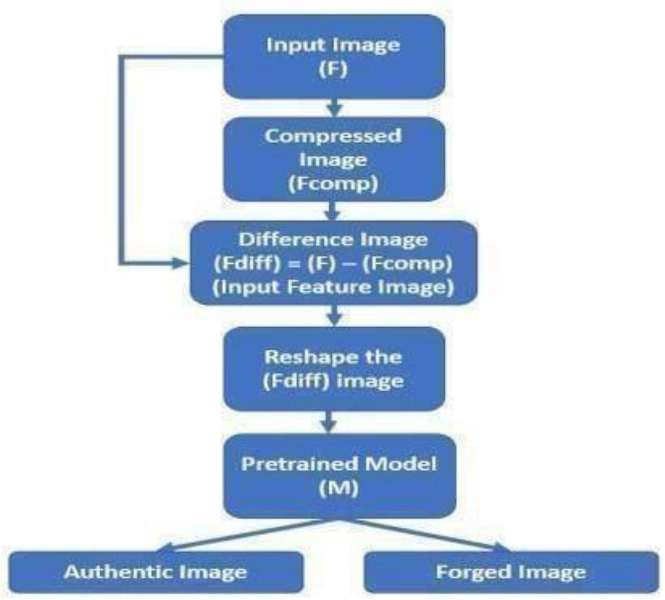


Fig 3.3 Flowchart of the proposed system (System Architecture).

4-MODEL TRAINING AND TESTING EVALUATION

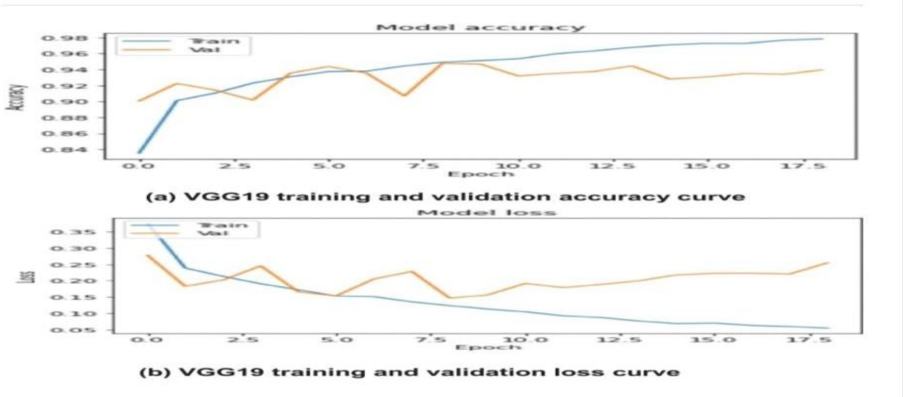
This section presents the training and evaluation process of the proposed image forgery detection model. The model is trained using a pre-trained Convolutional Neural Network (CNN) architecture through transfer learning, allowing it to leverage learned features from large datasets while minimizing the need for extensive data collection and computational resources. The training phase involves fine-tuning the model on a targeted forgery dataset, ensuring it can accurately distinguish between authentic and manipulated regions.

To assess the model's performance, various evaluation metrics such as accuracy, precision, recall, F1- score, True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), and False Negative Rate (FNR) are computed. These metrics provide a comprehensive understanding of the model’s strengths and areas that require improvement. Additionally, the impact of different preprocessing techniques and parameter settings is analyzed to optimize the model's performance. This evaluation aims to demonstrate the model’s reliability and effectiveness in real-world forgery detection scenarios.

Model Training

In order to fairly evaluate the training and testing phase for the eight different pre-trained models, a set of initial value parameters should be fixed all over the eight experiments. These parameters are as follows: The size of the input images is 160×160, with initial weight ‘ImageNet’, the number of epochs =100 with early stopping condition monitoring the minimum validation loss with patience =

10. The optimizer used is the “Adam” optimizer with learning rate = 1e-5 and loss ‘binary cross entropy’. In the experiments, the relation between the training and validation curve for the accuracy and the loss for each pre-trained model experiment is drawn, and three samples from them are displayed in Figures 4.1, 4.2 and 4.3. In each figure, (a) displays the relationship between the training and validation accuracy, and (b) displays the relationship between the training and validation loss for each model.

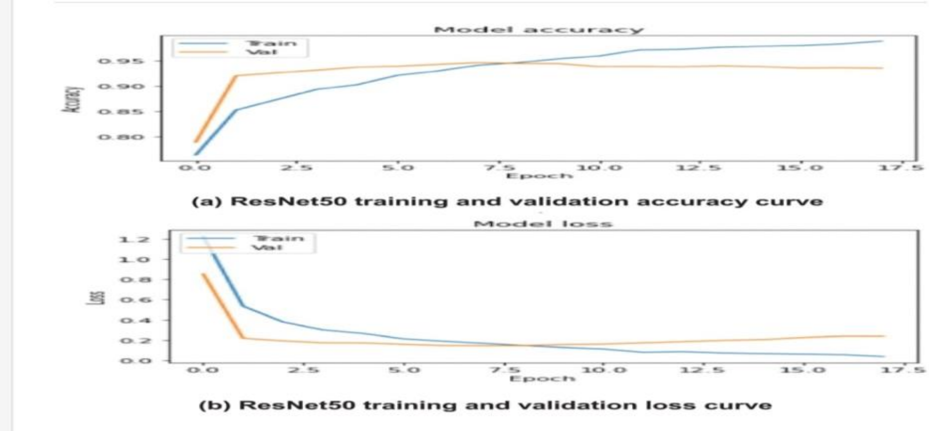


(a) VGG19 training and validation accuracy curve

(b) VGG19 training and validation loss curve

Fig 4.1 VGG 19 training and validation curve.

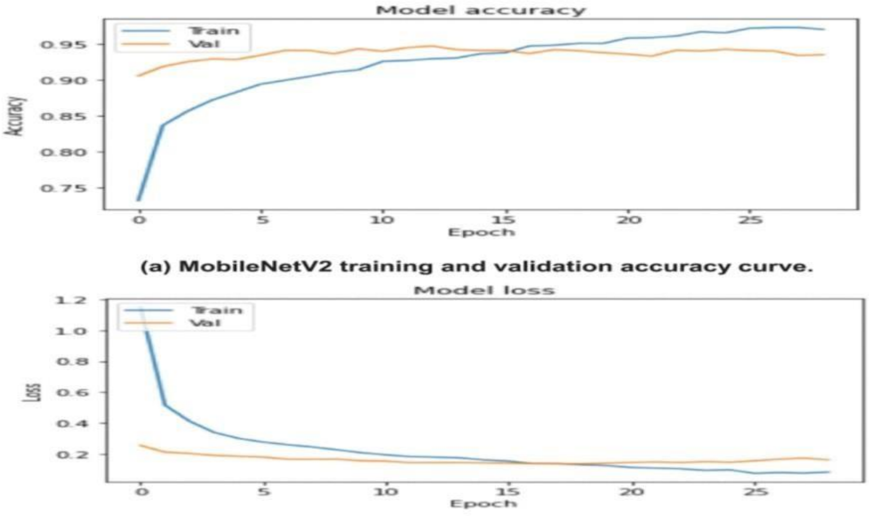
In the above we can see the graphs of both VGG19 training and validation curve of model accuracy and model loss, here bule line shows the training and orange line shows validation.



(a) ResNet50 training and validation accuracy curve

(b) ResNet50 training and validation loss curve

Fig 4.2 Resnet50 training and validation curves.



(a) MobileNetV2 training and validation accuracy curve.

(b) MobileNetV2 training and validation loss curve

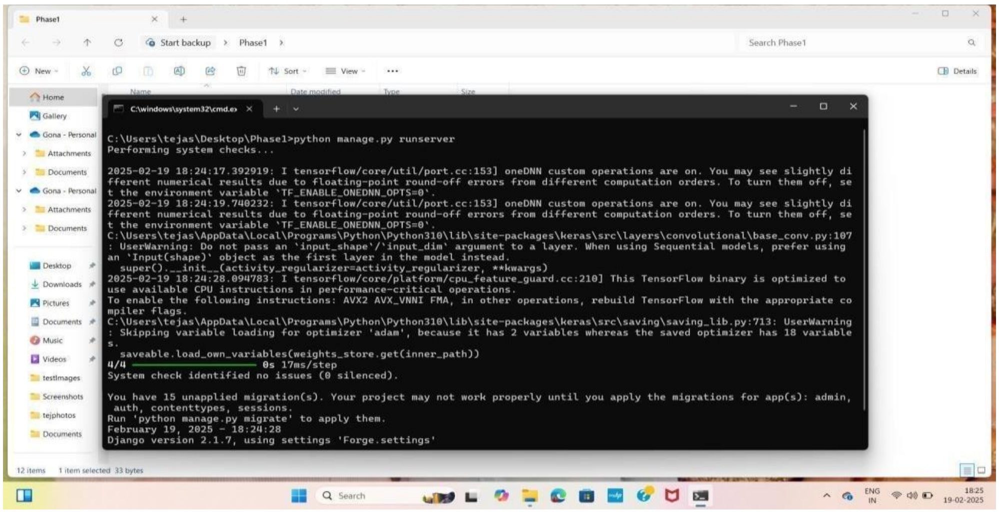
Fig 4.3 MobileNetV2 training and validation curves.

These graphs are useful in many directions; the training accuracy curve shows how well the model is learning from the training data over time. As shown the curve generally increases as the model gets better at fitting the training data.

On the other side, the validation accuracy curve shows how well the model is performing on a separate set of testing data that has not been seen during the training. The curve generally follows the training accuracy curve, but it may not increase as quickly or may plateau earlier. When the validation accuracy curve starts to decrease or diverge from the training accuracy curve, it indicates that the model is overfitting the training data, and is not generalizing well to new data.

5-RESULT AND WORKING

Running Django Server with TensorFlow



```
C:\Users\tejas\Desktop\Phase1>python manage.py runserver
Performing system checks...

2025-02-19 18:24:17.392919: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly di
fferent numerical results due to floating-point round-off errors from different computation orders. To turn them off, se
t the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-02-19 18:24:19.740232: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly di
fferent numerical results due to floating-point round-off errors from different computation orders. To turn them off, se
t the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
C:\Users\tejas\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\layers\convolutional\base_conv.py:107
: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using
an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
2025-02-19 18:24:28.094783: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to
use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate co
mpiler flags.
C:\Users\tejas\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\saving\saving_lib.py:713: UserWarning
: Skipping variable loading for optimizer 'adam', because it has 2 variables whereas the saved optimizer has 18 variable
s.
  saveable.load_own_variables(weights_store.get(inner_path))
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
 auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 19, 2025 - 18:24:28
Django version 2.1.7, using settings 'Forge.settings'
```

Fig 1 Server window

In above screen python server started and now open browser and then enter URL as http://127.0.0.1:8000/index.html and then press enter key to get below page

The output you shared appears to be from running a Django server (python manage.pyrunserver) while also using TensorFlow in the project. Here's a breakdown of what's happening:

Django System Check & Server Start:

- The command python manage.py runserver is used to start the Django development server.
- The system check was performed, and no critical issues were found (System check identified no issues (0 silenced).

1.2 Enhancing Digital Image Forgery Detection Using CNN

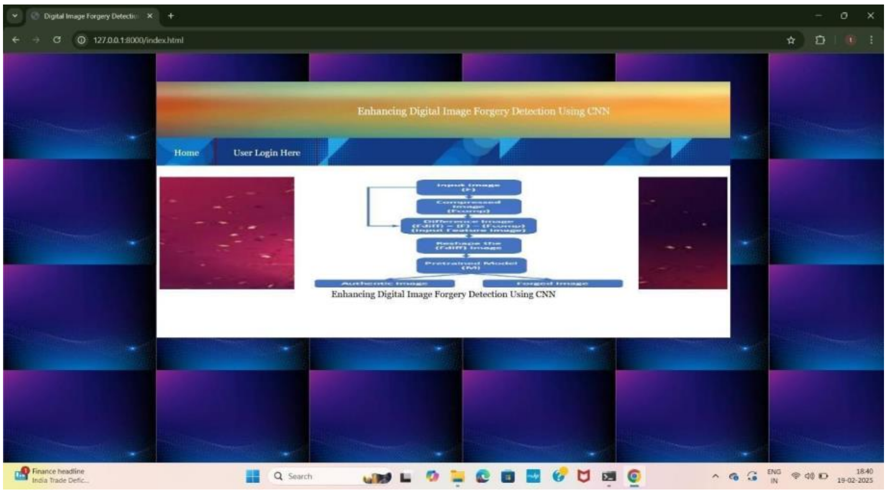
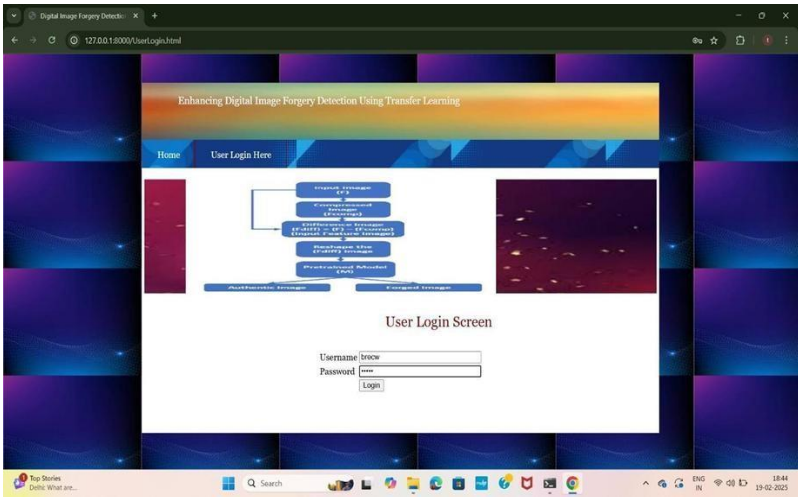

Digital Image Forgery Detectio
127.0.0.1:8000/index.html
Enhancing Digital Image Forgery Detection Using CNN
Home
User Login Here
Input Image (F)
Compressed Image (Fcomp)
Difference Image (Fdiff) = (F) − (Fcomp) (Input Feature Image)
Reshape the (Fdiff) Image
Pretrained Model (M)
Authentic Image
Forged Image
Enhancing Digital Image Forgery Detection Using CNN
Finance headline
India Trade Defic...
Search
ENG
IN
18:40
19-02-2025

Fig 2 User Login Page

In above screen click on 'User Login Here' link to get below page

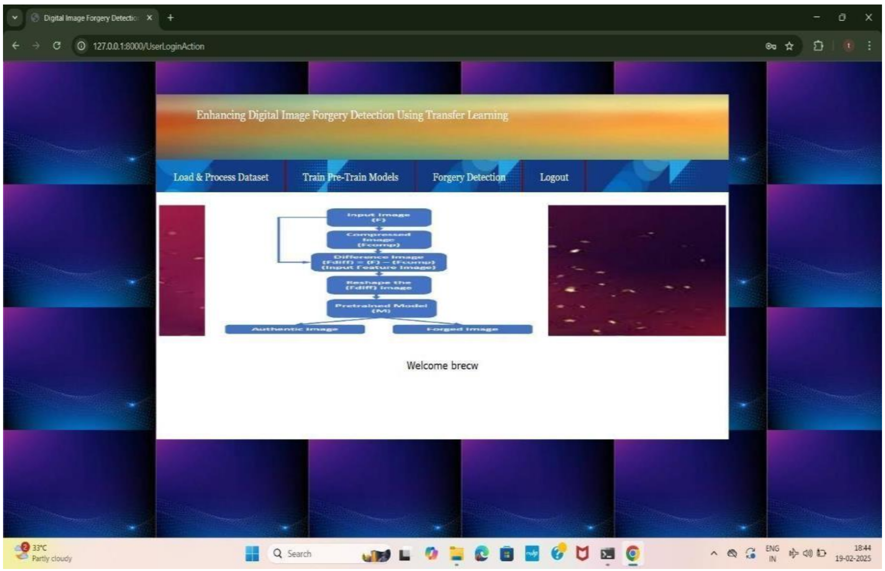
Screen shows a user login Interface


Digital Image Forgery Detectio
127.0.0.1:8000/UserLogin.html
Enhancing Digital Image Forgery Detection Using Transfer Learning
Home
User Login Here
Input Image (F)
Compressed Image (Fcomp)
Difference Image (Fdiff) = (F) − (Fcomp) (Input Feature Image)
Reshape the (Fdiff) Image
Pretrained Model (M)
Authentic Image
Forged Image
User Login Screen
Login
Top Stories
Delhi: What are...
Search
ENG
IN
18:44
19-02-2025

Username brecw
Password •••••

Fig 3 Login Details

In above screen user is login by entering username and password as 'admin and admin' and then press button to get below page

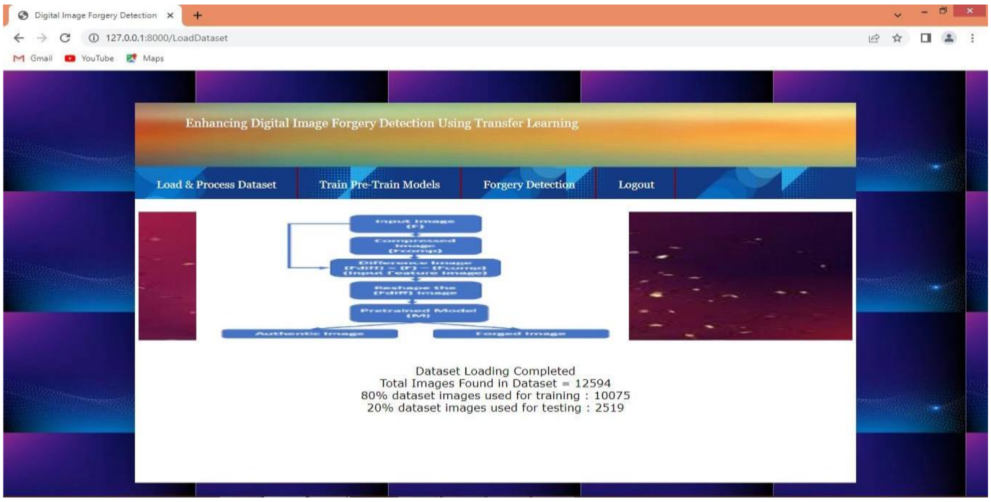

Digital Image Forgery Detectio
127.0.0.1:8000/UserLoginAction
Enhancing Digital Image Forgery Detection Using Transfer Learning
Load & Process Dataset
Train Pre-Train Models
Forgery Detection
Logout
Input image (F)
Compressed image (Fcomp)
Difference image (Fdiff) = (F) − (Fcomp) (Input feature image)
Reshape the (Fdiff) image
Pretrained Model (M)
Authentic image
Forged image

Welcome brecw

Fig 4 After login page

In above screen click on ‘Load & Process Dataset’ link to get below page

Dataset


Digital Image Forgery Detection
127.0.0.1:8000/LoadDataset
Gmail
YouTube
Maps
Enhancing Digital Image Forgery Detection Using Transfer Learning
Load & Process Dataset
Train Pre-Train Models
Forgery Detection
Logout
Input image (F)
Compressed image (Fcomp)
Difference image (Fdiff) = (F) − (Fcomp) (Input feature image)
Reshape the (Fdiff) image
Pretrained Model (M)
Authentic image
Forged image

```
Dataset Loading Completed
Total Images Found in Dataset = 12594
80% dataset images used for training : 10075
20% dataset images used for testing : 2519
```

Fig 5 Dataset

In above screen dataset loaded and can see total images loaded from dataset along with training and testing size and now click on ‘Train Pre-Train Models’ link to train models and get below page

6-CONCLUSION

In the digital age, images are widely used for communication, journalism, legal evidence, and social media. However, the ease with which digital images can be manipulated has raised serious concerns regarding their authenticity. Image forgery—where an image is altered to mislead viewers—has become increasingly common and sophisticated, posing challenges in various sectors such as media, law enforcement, and cybersecurity.

Traditional image forgery detection techniques, including statistical analysis and handcrafted feature extraction, often fall short when dealing with highresolution forgeries or advanced manipulation techniques like copy-move, splicing, and deepfake generation. These methods are typically sensitive to noise and fail to generalize across diverse forgery types.

To address these limitations, deep learning models, particularly Convolutional Neural Networks (CNNs), have been explored for their ability to learn hierarchical representations directly from data. However, training deep networks from scratch requires vast labeled datasets and extensive computational resources, which are not always feasible.

This project focuses on enhancing digital image forgery detection using transfer learning techniques. By integrating state-of-the-art pre-trained CNN architectures, the objective is to build a robust and efficient detection system that can effectively identify various types of forgeries with high accuracy and low computational overhead.

REFERENCE

[1] K.D. Kadam, S. Ahirrao, andK.Kotecha, ‘‘Multipleimagesplicingdataset (MISD): A dataset for multiple splicing,’’ Data, vol. 6, no. 10, p. 102, Sep. 2021.

[2] R. Agarwal, O. P. Verma, A. Saini, A. Shaw, and A. R. Patel, ‘‘The advent of deep learning-based,’’ in Innovative Data Communication Technologies and Application. Singapore: Springer, 2021.

[3] M. A. Elaskily, M. H. Alkinani, A. Sedik, and M. M. Dessouky, ‘‘Deep learning-based algorithm (ConvLSTM) for copy moves forgery detection,’’ J. Intell. Fuzzy Syst., vol. 40, no. 3, pp. 4385– 4405, Mar. 2021.

[4] A. Mohassin and K. Farida, ‘‘Digital image forgery detection approaches: A review,’’ in Applications of Artificial Intelligence in Engineering. Singapore: Springer, 2021.

[5] K. B. Meena and V. Tyagi, Image Splicing Forgery Detection Techniques: A Review. Cham, Switzerland: Springer, 2021.

[6] S. Gupta, N. Mohan, and P. Kaushal, ‘‘Passive image forensics using universal techniques: A review,’’ Artif. Intell. Rev., vol. 55, no. 3, pp. 1629–1679, Jul. 2021.

[7] W. H. Khoh, Y. H. Pang, A. B. J. Teoh, and S. Y. Ooi, ‘‘In-air hand gesture signature using transfer learning and its forgery attack,’’ Appl. Soft Comput., vol. 113, Dec. 2021, Art. no. 108033.

[8] Abhishek and N. Jindal, ‘‘Copy moves and splicing forgery detection using deepconvolutionneuralnetwork, andsemanticsegmentation,’’Multimedia Tools Appl., vol. 80, no. 3,
pp. 3571–3599, Jan. 2021.

[9] M. M. Qureshi and M. G. Qureshi, Image Forgery Detection & Localization Using Regularized U-Net. Singapore: Springer, 2021.

[10] Y. Rao, J. Ni, andH.Zhao, ‘‘Deeplearninglocaldescriptorforimagesplic ing detection and localization,’’ IEEE Access, vol. 8, pp. 25611–25625, 2020.

[11] K. M. Hosny, A. M. Mortda, N. A. Lashin, and M. M. Fouda, ‘‘A new method to detect splicing image forgery using convolutional neural network,’’ Appl. Sci., vol. 13, no. 3, p. 1272, Jan. 2023.

[12] F. Li, Z. Pei, W. Wei, J. Li, and C. Qin, ‘‘Image forgery detec tion using tamper-guided dual self-attention network with multireso lution hybrid feature,’’ Secur. Commun. Netw., vol. 2022, pp. 1–13, Oct. 2022.

[13] C. Haipeng, C. Chang, S. Zenan, and L. Yingda, ‘‘Hybrid features and semantic

reinforcement network for image,'' Multimedia Syst., vol. 28, no. 2, pp. 363–374, 2021.

[14] Q.Li,C.Wang,X.Zhou,andZ.Qin,''Imagecopy-moveforgerydetection and localization based on super-BPD segmentation and DCNN,'' Sci. Rep., vol. 12, no. 1, Sep. 2022, Art. no. 14987.

[15] A. K. Jaiswal and R. Srivastava, ''Detection of copy-move forgery in digital image using multi-scale, multi-stage deep learning model,'' Neural Process. Lett., vol. 54, no. 1, pp. 75–100, Aug. 2021.

[16] S. Koul, M. Kumar, S. S. Khurana, F. Mushtaq, and K. Kumar, ''An efficient approach for copy-move image forgery detection using convolution neural network,'' Multimedia Tools Appl., vol. 81, no. 8, pp. 11259–11277, Mar. 2022.

[17] S. S. Ali, I. I. Ganapathi, N.-S. Vu, S. D. Ali, N. Saxena, and N. Werghi, ''Image forgery detection using deep learning by recompressing images,'' Electronics, vol. 11, no. 3, p. 403, Jan. 2022.

[18] K. Kadam, S. Ahirrao, K. Kotecha, and S. Sahu, ''Detection and localization of multiple image splicing using MobileNet v1,'' IEEE Access, vol. 9, pp. 162499–162519, 2021.

[19] E.U.H. Qazi, T. Zia, andA.Almorjan, ''Deeplearning-baseddigitalimage forgery detection

system,'' Appl. Sci., vol. 12, no. 6, p. 2851, Mar. 2022.

[20] A.-R. Gu, J.-H. Nam, and S.-C. Lee, ''FBI-Net: Frequency-based image forgery localization via multitasks learning with self-attention,'' IEEE Access, vol. 10, pp. 62751–62762, 2022