IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

Veriforge:An Image Forgery Detection Model

RISHIKESH RAVI[1], HAMZA RANGWALA[2], AYAZ AFZAL[3] and TRUSHIT PATEL [4],
[1]Wilfrid Laurier University, Waterloo ON Canada (email:rish8180@mylaurier.ca)
[2]Wilfrid Laurier University, Waterloo ON Canada (email:rang8720@mylaurier.ca)
[3]Wilfrid Laurier University, Waterloo ON Canada (email:afza9880@mylaurier.ca)
[4]Wilfrid Laurier University, Waterloo ON Canada (email:pate9410@mylaurier.ca)

ABSTRACT This paper proposes a new method of tampered image detection by combining the CNN with a backbone of ResNet50 with ELA. Our method is designed for digital forensics, which is a very important domain for the detection and assessment of image modifications in many other domains, such as cybersecurity, journalism, and criminal investigations This procedure comprises image pre-processing to ascertain altered areas by comparing variations in compression levels to find compression artifacts using ELA. It serves as the foundation for reliable feature extraction: the ResNet50 model, pre-trained on ImageNet, has its basic layers frozen to preserve learned weights. For customization with regard to binary classification, some custom layers are appended on top: global average pooling, batch normalization, dense layers with ReLU, and dropout regularization. The architecture here comes up with a generator-based approach so that large datasets can be handled with better memory consumption, enabling real-time data preprocessing during training of models.Our methodology is assured to optimize resource utilization and enhance generalization, which has been evidenced with extensive experiments. The obtained model offers high accuracy and robustness in detecting tampered images; thus, it presents a scalable solution for real-world forensic applications. This integration of ELA and CNN provides the backbone of fast and accurate analysis of digital evidence to help in better decision-making in sensitive situations.

INDEX TERMS Tampered Image Detection, Error Level Analysis (ELA), Convolutional Neural Network (CNN), ResNet50, Digital Forensics, Image Manipulation Detection, Cybersecurity, Journalism, Criminal Investigations, Feature Extraction, Binary Classification, Data Preprocessing, Dropout Regularization, Compression Artifacts, Real-time Analysis, Scalable Forensic Applications.

I. INTRODUCTION

IN this digital era, where information is usually imparted through the means of digitization, a tampered image might start a wildfire of misinformation, controversies, and disasters in a lot of sectors.Over the last few decades, deep learning has grown to be one of the strongest and leading methods for the solution of challenging problems arising in numerous fields. [1]The more use of social media escalates the scale of it, which calls for this challenge to be taken seriously.The instances of tampered images being used for spreading fake news, malicing individuals or organizations, and manipulating public opinion have started becoming alarmingly common. Images manipulated through image processing tools or even completely fabricated with the help of AI, in order to distort the truth and mislead people or clinch wrongful convictions in a court of law. [2].A serious issue of this aspect, our research focuses on VeriForge, which is a model designed for the detection and exposure of image tampering using the powerful deep learning and Error Level Analysis [3]. This innovative approach is targeted at offering an effective solution in finding discrepancies in digital images and building trust in visual content. But accessibility to sophisticated image editing software has put the art of doctoring images within everybody's reach, irrespective of any kind of technical expertise. For that reason, it has become an increasingly demanding task to differentiate between original and tampered images by commoners, but also by professionals in vital fields like law and order or journalism. But available techniques of image tampering detection, though at one's fingertips, are not short-circuited by numerous limitations. Most of the known methods are outdated, slowing down, and cannot find a complex type of manipulation. Nowadays, there is a serious demand to create modern and effective methods which could cope with complications arising in image forgery techniques [4].

An automatic computer-based system is required to be developed which can identify the originality of input image [5],VeriForge fills this gap by using the synergy of traditional forensic methods and deep learning. ELA, for exam-

ple, is a well-established method in forensics: it underlines inconsistencies between image compression levels, hence highlighting the image anomalies caused by editing [6]. VeriForge couples ELA with deep learning to extract meaningful features from these discrepancies in a way that has allowed the attainment of high accuracy enabled by advanced neural network architecture. It allows VeriForge to find the tampered images with quite an intelligent attack.

For the training and verification of VeriForge, we utilize CASIA V2, a database of well-labeled images in their original and forged aspects. In this database, tampering is performed manually with several image editing tools; thus, it is one of the good sources for training a model to detect different types of tampering [7]. This is also ensured by the great variety of images in the dataset, thus the model generalizes well to unseen data, avoiding overfitting and improving its application in real-world scenarios.

The VGG-16 architecture was adopted for the first version of VeriForge. Although a promising result could be achieved in a preliminary way, we obtained only mediocre performance, given that after some point the model converged, resulting in an inability to increase its accuracy. Then, in order to capture the intricate image features, residual learning with the much deeper convolutional neural network architecture, ResNet50 [8], was chosen. This switch to ResNet50 allowed VeriForge to capture more complicated patterns of tampered images, thereby increasing its detection capability by a big margin.

The general objective of the work is to protect the integrity of digital content by offering a reliable, scalable, and efficient tool for image forgery detection. In this regard, VeriForge uses advanced techniques such as convolutional neural networks (CNNs) [9], transfer learning [10] , and state-of-the-art architectures like ResNet50 [11] . Put together, these technologies ensure that VeriForge will find tampered images and continually improve by keeping pace with the changing world of image manipulation.

In a nutshell, VeriForge is a leap into the future of image tampering detection: it merges classical forensic techniques-such as ELA-with the latest powers of deep learning for an emerging need to have reliable tools safeguarding digital content in this world where visual communication is well on its way to dominating other forms.

II. LITERATURE REVIEW

Image forgery detection has become a very important area of research because of the increasing prevalence of digital image manipulation and its possible misuse in spreading misinformation, undermining trust in media, and enabling cybercrime. Social media platforms have emerged as one of the most used mediums for sharing content. Inappropriate images shared on these platforms can lead to legal consequences. Image manipulation techniques such as copy-move or splicing are standard practices. [12] So, there is a need to develop techniques to validate integrity and authenticity of the images, as this image is considered as the evidence in various fields like in investigation or in medical field as medical record or as financial documents, etc. [13]

A. TRADITIONAL APPROACHES TO IMAGE FORGERY DETECTION

Image manipulation detection is different from traditional semantic object detection because it pays more attention to tampering artifacts than to image content. [14] Traditional approaches rely on the inherent changes introduced by manipulation tools.To enhance performance of ELA, data augmentation techniques and meticulous hyperparameter tuning are employed, along with the exploration of alternative model architectures. [15] Traditional approached involved resampling features are used to capture artifacts, such as JPEG quality loss, upsampling, downsampling, rotation, and shearing. [16]

1) Copy-move Forgery

The most frequent form of image fraud is called a copy-move forgery, where a portion of the original image is copied and pasted in a different spot within the same image [17]. It simply entails copying picture blocks into the same image and concealing vital information or objects from view. [18] a copy move forgery detection method is proposed in this paper using regional Gestalt wavelet analysis structures, which draw upon the rotation invariant ability of uniform local binary patterns and a high-performance texture analysis ability of Gaussian filters.

2) Image splicing Forgery

Image splicing is a picture compositing technique that involves joining image fragments from the same or separate images without further post-processing, such as smoothing the borders between the pieces.Image splicing forgery, where parts of one image are pasted and then copied into another image to merge a new image. [19] Image splicing is fundamentally different from copy-move in the sense that the pasted region cannot be found elsewhere within the same image. [20]

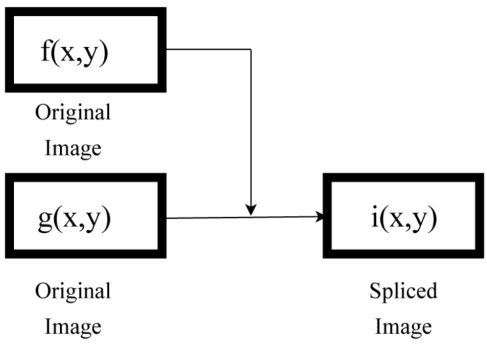


FIGURE 1. Image Slicing Mechanism, adapted from [20]

B. MACHINE LEARNING TECHNIQUES

Early machine learning models, such as those using Gaussian Discriminant Analysis (GDA) or Support Vector Machines (SVMs), have been leveraged to classify images based on handcrafted features like Local Binary Patterns (LBP) or texture characteristics. Although promising, these models are often constrained by the limited generalizability of their features to diverse tampering types.

C. CNN ARCHITECTURES

Recent works involving VGG-16, ResNet, and Xception demonstrate that deeper architectures perform better in capturing complex patterns; this is evident from the very best performance of models such as ResNet50. A CNN model trained on ELA images outperforms other pre-trained models. [21]. CNN uses a pooling layer which reduces the dimension of the data and serves the dual purpose of reducing the enormous amount of output to be fed into the next input layer and by losing some information, the problem of overfitting is solved. [22]

1) Convolution Layer

The major function of the convolutional layer is to extract features and to initialize fixed-size kernels to scan the input matrix. The size of the kernels and the stride of sliding are set when a network is to be constructed. Let a_{ij} denote the pixel at the i-th line and the j-th column of a feature map, the computation of which follows:

$$a_{i,j} = f\Big(\sum_{m=0}^{\mathrm{M}}\sum_{n=0}^{\mathrm{N}} W_{m,n}X_{i+m,j+n} + W_b\Big) \tag{1}$$

where x_{ij} denotes the pixel at (i,j) of the input image, $W_{m,n}$ denotes the shared weight at the corresponding position of filters, W_b is the bias of the filters, and f denotes the activation function. The size of the kernel is $M \times N$, meaning that m ranges from 0 to M and n ranges from 0 to N. [23]

2) ReLU Function

The method proposed by [24] which uses gradient information flowed from the final convolutional layer in our ResNet-50 model to provide, for each pixel in the image, the amount of contribution in final class decision, resulting in an RGB gradient map in which dark red represents areas with high contribution to final decision and dark blue denotes otherwise.

The gradient map G belonging to Ru×v, of width u and height v, given a target class t is

$$\mathbf{G}^t = ReLU\overbrace{\left(\sum_{\kappa}\alpha_{\kappa}^{t}A^{\kappa}\right)}^{\text{linear combination}}, \tag{2}$$

$$\alpha_{\kappa}^{t} = \overbrace{\frac{1}{Z}\hat{\sum_i}\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^t}{\partial A_{ij}^{\kappa}}}_{\text{gradients via backprop}} \tag{3}$$

where A^{κ} represents the activation map of the κ-th feature channel of the last convolutional layer. The coefficients α_{κ}^{t} determine the importance of each feature channel for the target class t and are computed using:

- Z is the normalization factor, equal to the total number of elements in the activation map ($Z = u \times v$), - $\frac{\partial y^t}{\partial A_{ij}^{\kappa}}$ represents the gradients obtained via backpropagation, showing how the change in the activation A_{ij}^{κ} affects the class score y^t, - The global average pooling ($\frac{1}{Z}\sum_i\sum_j$) aggregates these gradients spatially over the activation map, assigning a single importance weight to each feature channel κ.

Only positive contributions of feature maps within ReLU highlight regions supporting the model's prediction for target class t since ReLU masks negative responses. It combines the information effectively, both in a spatial and feature level, from the activation maps so that the class-relevant region of interest could be well-localized within the input image.

D. INTEGRATING ELA WITH DEEP LEARNING

The ELA of Krawtez [25] helps to determine the areas in an image at different levels of compression. The method focuses on the lossy compression of the manipulated images to identify them. Original quality itself is a unique feature of the image.The error levels calculated are related to the compression loss. A small change in these error levels implies that the pixel is at its local minima for that particular error rate [26]. ELA operates as a forensic tool that identifies variations in error levels within compressed images, facilitating the discernment of manipulated regions. [27] In case of alteration, traces are left behind in one way or another. In the process of ELA, we compress the image through a lossy function. In our implementation we use the JPEG format for such a purpose.

$$ELA(n_1,n_2) = |X(n_1,n_2) - X_{rc}(n_1,n_2)| \tag{4}$$

For each colour channel in the above equation, n_1 and n_2 represent the row and the column indices, respectively, X represents the image that is required to be checked for manipulation, and X_{rc} represents the recompressed image.

The overall error levels are computed by averaging across all colour channels as given below:

$$ELA(n_1,n_2) = \frac{1}{3}\sum_{i=1}^{3}|X(n_1,n_2,i) - X_{rc}(n_1,n_2,i)| \tag{5}$$

where $i = 1,2,3$ for Red, Green, and Blue (RGB) image.

Several distinct artifacts arise from the diverse phases of the picture formation process. [28]. The detection of photographic splicing by bringing together the high representation power of Illuminant Maps and Convolutional Neural Networks is a way of learning directly from available training data the most important hints of a forgery. [29]

Our proposed system is based on these insights; it leverages ELA for feature enhancement and ResNet50 for binary classification, Since the RGB model is one of the most popular color models used in image presentation. [30] we use it parallel with ELA thus overcoming the limitations of earlier models.While the traditional methods, such as ELA, give valuable insights into the problem, integrating it with deep architectures such as ResNet50 can therefore provide a strong solution against modern challenges. Our contribution, therefore, to this effort continues with Veriforge: a method that joins strengths from both ELA and deep learning, providing great accuracy and robustness on CASIA V2.

III. PROPOSED METHOD

In this paper, we demonstrate that combining ELA with CNN model while freezing the base layers and adding custom layers to the base model which will provide a significant increase in the robustness of the final model and significantly improve performance of the classification task to detect images, which are tampered. We explore this idea in the context of digital forensics, where the goals are to analyze the images and detect suspicious activity in the field of crime, journalism etc. Since digital forensics helps uncover activities and patterns, determine the root causes of incidents, and establish a chain of evidence admissible in court, it becomes essential to identify, recover, analyze and present digital evidence from electronic devices and digital storage. Additionally, an ELA based model helps in identifying the difference in various layers of the images to help us uncover the underlying changes made to the image that is suspected to be tampered with. The model allows us to help various fields where criminal investigations, legal disputes and cybersecurity incidents occur and images need to be analyzed quicky to help user make quick informed decisions. The following sections outline the details of each step in the process.

A. INITIALIZATION

The initialization of the framework is preprocessing the images based on the number of images available in the dataset, it corresponds to the total number of images which are categorized in two classes specifically: 1) Authentic and 2) Tampered. The classes with their labels are split initially to ensure we have a sizeable amount of image to work with, we ensure that the classes are not biased and are equally balanced to provide a robust model and reduce overfitting. The classes are initialized with ones and zeros, zero if the image is authentic and one if the image has been tampered. For images to get preprocessed, we need to handle a chuck of images at a time, which will require us to take images in batches to preprocess and analyze further.

After reshaping the images according to our input shape which will reduce the size of the image so that our next function would work effectively. Creating batches to handle memory overflow and preprocessing image before feeding them to the ELA function ensures that the system doesn't require a lot of resources and is more optimized to create a more robust model.

Further, we take each resized image and input it to our ELA function, this step is iterated over multiple images to get robust images with reduced quality to take forward for our ELA function.

B. ELA PROCESS WITH GENERATORS

For each image in our ELA function, we generate a buffer to save the original image to a temporary in-memory file with reduced quality, which helps us in reducing the overall memory utilization of the system. We load the compressed image from the buffer and calculate the difference between the original image and the one which was compressed to get the Pillow image object, this is a technique used in image forensics to detect tampering or inconsistencies in an image. We enhance the brightness of the image according to the extrema we get in the pillow image object.

This returns us the ELA image which is a new image that emphasizes regions with differing compression levels, useful for detecting tampering or manipulation. This image will highlight differences due to compression artifacts. To feed the image forward to the model for training, it requires the image to be converted to an array for numerical processing and is normalized making the data compatible with many machine learning models. We close the image further to free up resources and it is a good practice to avoid file locks or memory leaks, especially when processing multiple images in a loop. For the model to handle such large number of images at once would require a lot of resources, to tackle this issue we provide generators to the model.

These generators help in handle image preprocessing and feeding batches of data to the model during training. They are often used in deep learning workflows when the model requires to load images in batches and apply transformations or augmentation. Instead of loading all images into memory at once, generators load and process batches on-the-fly, reducing memory consumption. Generators often include on-the-fly preprocessing, which enhances the dataset and helps the model generalize better, increasing its robustness and generalizability for better performance. We also try to visualize the batch of images after the ELA function to ensure that the shape of the generator and preprocessed images are according to the requirements.

C. RESNET50 MODEL ARCHITECTURE WITH TRAINING EVOLUTION

In this study, we employ a Resnet50 model to classify the images into two categories (Authentic and Tampered), we use a transfer learning approach based on the ResNet50 architecture, pre-trained on ImageNet to solve our classification

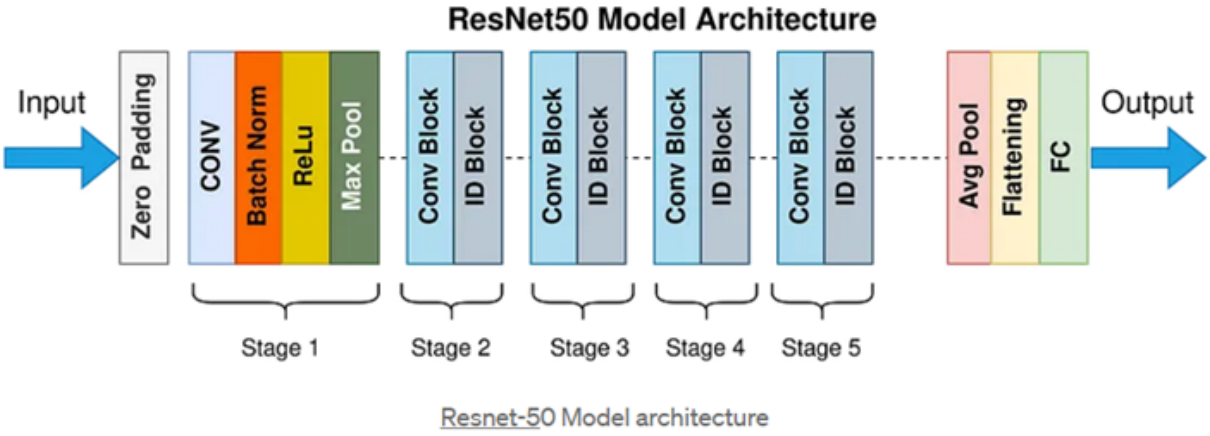


FIGURE 2. **Architecture of the base model**

problem. This method capitalizes on the robust feature extraction capabilities of ResNet50, significantly reducing training time and computational resources while improving model accuracy. The model was customized to adapt the pre-trained base for the task by introducing additional trainable (Custom) layers specifically designed for binary classification.Figure 2: shows the architecture of the base model we have used featuring Resnet50 in a five-stage design to represent the overall function of the layers

The base of our architecture is the ResNet50 model, known for its depth and efficient handling of the vanishing gradient problem through residual connections. The model was loaded without the top layers, removing the full-connected top layers helped us retained only the convolutional layers which are responsible for extracting hierarchical features from the input images. The input shape was fixed to match the requirement of the model while ensuring compatibility with pre-trained weights. The freezing of all layers in the base model during training aimed at preserving the learned weights and keeping the feature extraction capability unaltered by the new task-specific data. To this binary classification problem, we attached five different types of custom layers to the base pre-trained model in an attempt to make it adapt. First, there is a global average pooling layer that reduces the high-dimensional feature maps output by ResNet50 into a single vector for each feature map, reducing the number of trainable parameters and hence improving generalization. Unlike the traditional Flatten layer, this preserves spatial information and improves model performance when using ResNet-based architectures. Further, we added batch normalization to normalize the activations from the pooling layer, reducing internal covariate shift and accelerating convergence.

Once we are done with that, we added a series of two dense layers. The first layer consists of 512 neurons providing substantial capacity to capture complex patterns. The second contains 256 neurons for further refinement of feature extraction. Both are activated by the ReLU activation function, which introduces non-linearity in deep networks and provides output zero for negative inputs, creating sparsity in the activations that could further be helpful in efficient computations with a reduction in overfitting. While being computationally less expensive than sigmoid or tanh function. The custom layers then involved a dropout regularization technique which was employed after each dense layer to with a rate of 0.75 after the first dense layer to impose strong regularization and rate of 0.5 was applied after the second dense layer for moderate regularization. This helped in preventing overfitting of the model during training. After dropout regularization, there was a single neuron in the output layer with sigmoid activation to predict the probability of the positive class, which is suitable for binary classification tasks.

It achieves a very good trade-off among accuracy, training efficiency, and robustness, making it pretty effective in real-world applications. This model was trained for more than 100 epochs with a step size of about 145 while capturing essential evaluation metrics such as accuracy, the f1 score, and validation accuracy. This will make sure we minimize the gap between the training and validation accuracy of the model, which will be robust and very effective in real-world applications.

IV. RESULTS AND ANALYSIS

A. DATASETS

The CASIA V2 dataset, created by the Institute of Automation at the Chinese Academy of Sciences, CASIA, is one of the more general resources for research in the area of image forgery detection. Designed to support studies on image tampering, it includes manipulative techniques such as

splicing, copy-move forgery, and other methods.The Table 1 outlines the details of the dataset and gives us an overview of what the dataset contains. The dataset is a very diverse collection of 12,614 images of roughly equal amounts of authentic and tampered examples from various categories such as nature, animals, objects, and urban settings. Such diversity will enable models trained with CASIA V2 to generalize rather well across different scenarios. There are also ground truth masks provided for every tampered image in the dataset, highlighting manipulated regions and thus making the dataset indispensable for supervised learning tasks and performance evaluation.

Feature	Description
Dataset Name	CASIA TIDE v2.0
Purpose	Designed for research and development in image forgery detection
Released By	Chinese Academy of Sciences Institute of Automation (CASIA)
Image Type	Digital images, both authentic and tampered
Number of Images	12,614 images (7,491 authentic and 5,123 tampered)
Forgery Techniques	Splicing, copy-move, and other common image tampering techniques
Resolution	Varies (typically medium resolution)
File Format	JPEG (for compressed images)
Dataset Structure	Contains two folders: one for authentic images and one for tampered images
Applications	Used for image forgery detection, digital forensics, and tampered image localization research
Public Availability	Available for academic and non-commercial use upon request from CASIA

TABLE 1. Overview of the CASIA TIDE v2.0 Dataset

The resolution of the images also varies to reflect real-world conditions, as images naturally come from different devices and sources.CASIA V2 has a variety of applications, including training and validation of image forgery detection models, testing of pre-processing techniques such as Error Level Analysis, and benchmarking deep learning architectures like CNNs for forgery classification. Features such as these make it invaluable in the field of digital forensics and research on image authenticity.

B. RESULTS EVALUATION

This section compiles the quantitative results and presents an analysis of the project outcomes. The performance of ResNet50 and VGG16 CNN models trained on CASIA V2 dataset by using traditional methods like ELA will be judged on Accuracy and F1 metrics.Accuracy measures the overall correctness of the model's predictions. At the same time, the F1 score represents the balance between precision and recall, which is suited for imbalanced datasets. Or in other words, it shows the likelihood that the model correctly classifies an image with its true label.

According to the qualitative results, RestNet50, combined with improved ELA, performed better than VGG16, which used traditional ELA.

Reference	Model	Accuracy
Jabaar et al. (2021)	ELA, CNN	93.8%
Mashaan & Ahmed (2023)	Local Binary Patterns, Tamura	96%
Pandiyan (2023)	Xception	95%
Singh et al. (2023)	ELA, CNN	98%
Geethanjali et al. (2024)	ELA, CNN	95%
Sarkar et al. (2024)	SegNet, MobileNet	89%
Anvekar et al. (2024)	CNN	96%
Roy et al. (2024)	Modified MultiResUnet Architecture	98%
Proposed Method	VGG16 + ELA	83%
Proposed Method	ResNet50 + Improved ELA	89%

TABLE 2. Summary of Different Studies and Quantitative Results

C. ANALYSIS OF THE INITIAL VGG16 MODEL

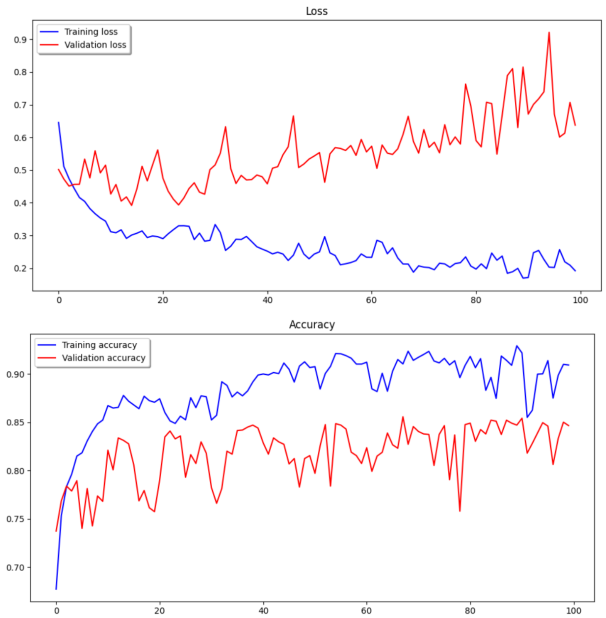


FIGURE 3. VGG16 Model: Loss and Accuracy Graphs

The first prototyping at the earlier stages of the project used a VGG16 CNN model combined with traditional ELA preprocessing. This approach focused on finding edges to detect forged regions on a grey-scale-converted image. Although this yielded a promising 83% in validation accuracy, the training and validation graphs were too inconsistent and abrupt, making the F1 score stand at a low of 0.86. That meant the balance between precision and recall had to be promoted, reducing the higher number of false positives and false negatives, which could be achieved by using more sophisticated preprocessing techniques. The model VGG16 has anomalies shown in Figure due to sudden changes and also huge differences between Training and Validation accuracy graphs. Because the model failed to generalise into a valid test set, it could be defined that the model went to overfit while in training, also justified with the low F1 value of the VGG16 model

Figure 3 reveals inconsistencies, with abrupt variations and a wide gap between the Training and Validation accuracy graphs for the VGG16 model. As the model failed to generalise on the validation test set, it can be termed that the model went overfitting during the training, which is evident by the low F1 score of the VGG16 model.

D. RESNET50 MODEL AND IMPROVED ELA WITH EVALUATION METRICS

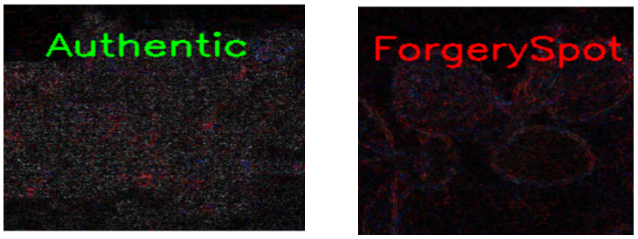


FIGURE 4. **Authentic and Forgery Predicted Label images**

Resnet50 with an enhanced ELA preprocessing was implemented to address these limitations. The Improved preprocessing technique modified the ELA to analyse the RGB channels instead of grayscale. This allowed the model to detect colour discrepancies and variations in JPEG compression artefacts which are more effective in detecting forgery compared to finding edges. The improved ELA led to the clear identification of tampered regions, as shown in Fig 4.

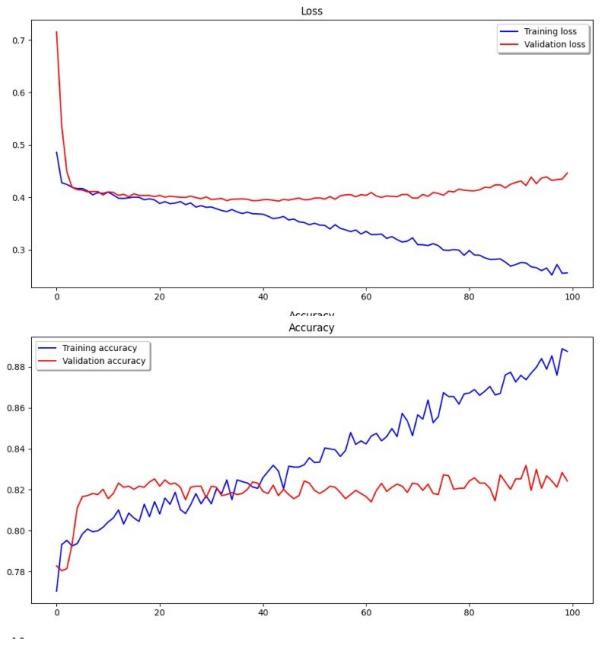
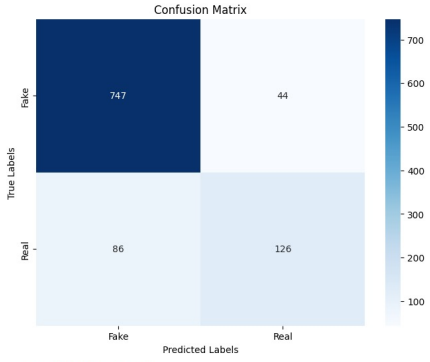


FIGURE 5. **Resnet50 Model: Loss and Accuracy Graphs**

Besides that, compared to VGG16, several advantages have been gained with the ResNet50 architecture. Instead of flattening the feature maps, global average pooling was used that reduces overfitting because of minimized trainable parameters. More than that, batch normalisation layers stabilize training by normalizing the inputs, and dropout with various intensities added to the regularization helped reduce overfitting. With these changes, training-validation accuracy curves were smoother and more convergent, as depicted in Fig 5.



Classification Report:

	precision	recall	f1-score	support
Fake	0.90	0.94	0.92	791
Real	0.74	0.59	0.66	212
accuracy			0.87	1003
macro avg	0.82	0.77	0.79	1003
weighted avg	0.86	0.87	0.86	1003

FIGURE 6. **Confusion Matrix and Evaluation Chart**

The improved ELA coupled with transitioning from VGG16 to Resnet50 significantly reduced the number of false positives. Smoothening of both the validation and training accuracy curves showed reduced overfitting compared to the curves of the VGG16 model. The resultant 89% accuracy and 0.92 weighted F1 score reflect the better generalization capability of the model compared to VGG16. A higher F1 score indicated a balanced recall and precision where false positives and false negatives are minimised as evident in Fig 6.

It is noteworthy to mention that data augmentation techniques – rotation, flip, zoom, width shift height shift etc – when applied did not have any effect on forged images. The best possible explanation for this phenomenon is that data augmentation does not present any variation in the forged region.After finalising the model configuration, training, and validation, it was programmatically uploaded to Hugging Face, from where it would be remotely instantiated in the backend service layer using the API provided by Hugging Face. Python Django was used to create a backend application to serve user requests, process the images present in the user requests, predict the labels, and respond with the predicted results.

V. CONCLUSION

This study is essential in maintaining integrity is crucial in fields like Law enforcement, Cyber security, and Legal investigations. In this project, we aimed to design a robust system capable of determining whether an image is tampered with. The research compares the performance of two widely known CNN models, VGG16 and ResNet50, with traditional

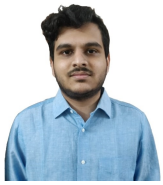
preprocessing methods like ELA and data augmentation. The project concludes with viable results that taking colour inconsistencies and compression differences significantly improves model capability compared to conventional grayscale-based ELA methods.The system can be integrated into social media to moderate content in real-time, flagging misleading images and advancing platform integrity.This needs to be handled in future work by enrichment of the dataset with diversity and updates for further strengthening the resistance and longevity of the model. Moreover, interpreting it with Explainable AI by using Grad-CAM will illustrate the regions of interest influential in the model's choices for transparent decisions and gaining the trust of the users. The dataset and code for further research are available [here] and [repository link].

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Prof. ANK Zaman for his invaluable guidance and support throughout the course of this research. His insightful suggestions, expert advice, and continuous encouragement significantly contributed to the successful completion of this project. We also appreciate his assistance in refining our methodology and providing constructive feedback, which greatly enhanced the quality of our work. This research would not have been possible without his mentorship.

REFERENCES

[1] N. T. Pham and C.-S. Park, "Toward deep-learning-based methods in image forgery detection: A survey," *IEEE Access*, vol. 11, pp. 11 224–11 237, 2023.
[2] Y. Zheng, Y. Cao, and C.-H. Chang, "A puf-based data-device hash for tampered image detection and source camera identification," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 620–634, 2020.
[3] A. Gupta, R. Joshi, and R. Laban, "Detection of tool based edited images from error level analysis and convolutional neural network," *arXiv preprint arXiv:2204.09075*, 2022.
[4] A. Kaur, D. Chahal, and L. Kharb, "Weak form efficiency of currency futures: Evidence from india," in *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE, 2019, pp. 30–35.
[5] K. H. Hingrajiya and R. K. Sheth, "Comparative study of digital image forgery detection techniques," in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2021, pp. 83–86.
[6] A. Singh and J. Singh, "Image forgery detection using deep neural network," in *2021 8th International conference on signal processing and integrated networks (SPIN)*. IEEE, 2021, pp. 504–509.
[7] N. B. A. Warif, M. Y. I. Idris, A. W. A. Wahab, and R. Salleh, "An evaluation of error level analysis in image forensics," in *2015 5th IEEE International Conference on System Engineering and Technology (ICSET)*, 2015, pp. 23–28.
[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1409.1556
[9] S. D. Thepade, S. Bhandari, C. Bagde, R. Chaware, and K. Lodha, "Image forgery detection using machine learning with fusion of global and local thepade's sbtc features," in *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, vol. 1. IEEE, 2021, pp. 234–238.
[10] S. Han, W. Gao, Y. Wan, and Y. Wu, "Scene-unified image translation for visual localization," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 2266–2270.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[12] N. I. R. Prasetya and Irmawati, "Detection of image splicing and copy-move forgery using the prewitt operator and cnn approach," in *2024 4th International Conference of Science and Information Technology in Smart Administration (ICSINTESA)*, 2024, pp. 475–480.
[13] S. Jadhav and N. Ramlal Shelot, "Analysis of image forgery detection using canny edge detector," in *2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, 2021, pp. 1–6.
[14] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1053–1061.
[15] C. Boustany and A. Wehbe, "Image fraud detection application using convolutional neural networks (cnns)-"imageguard"," in *2024 22nd International Conference on Research and Education in Mechatronics (REM)*. IEEE, 2024, pp. 23–28.
[16] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, "Hybrid lstm and encoder–decoder architecture for detection of image forgeries," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3286–3300, 2019.
[17] D. Narayan, Himanshu, and R. Kamal, "Image forgery detection," in *2023 International Conference on Disruptive Technologies (ICDT)*, 2023, pp. 549–552.
[18] C.-L. Chou and J.-C. Lee, "Copy-move forgery detection based on local gabor wavelets patterns," in *Security with Intelligent Computing and Big-data Services*. Springer, 2018, pp. 47–56.
[19] Y. Wei, X. Bi, and B. Xiao, "C2r net: The coarse to refined network for image forgery detection," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 1656–1659.
[20] M. T. H. Majumder and A. B. M. Alim Al Islam, "A tale of a deep learning approach to image forgery detection," in *2018 5th International Conference on Networking, Systems and Security (NSysS)*, 2018, pp. 1–9.
[21] M. Baviskar, S. Rathod, and J. Lohokare, "A comparative analysis of image forgery detection techniques," in *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*, 2022, pp. 1–6.
[22] R. Agarwal, D. Khudaniya, A. Gupta, and K. Grover, "Image forgery detection and deep learning techniques: A review," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2020, pp. 1096–1100.
[23] N. Huang, J. He, and N. Zhu, "A novel method for detecting image forgery based on convolutional neural network," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 1702–1705.
[24] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that?" *arXiv preprint arXiv:1611.07450*, 2016.
[25] N. Krawtez, "A pictures worth digital image analysis and forensics," *Black Hat Briefings*, vol. 131, no. 1, p. 31, 2007.
[26] A. Ghai, P. Kumar, and S. Gupta, "A deep-learning-based image forgery detection framework for controlling the spread of misinformation," *Information Technology & People*, vol. 37, no. 2, pp. 966–997, 2024.
[27] T. M. Geethanjali, T. S. Darshan, K. Surya, H. U. Rahul, and I. N. Sheety, "Detectify : Image tampering detection using error level analysis (ela) and convolutional neural network (cnn)," in *2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)*, 2024, pp. 1–6.
[28] D. P. Singh, V. K. Nair, R. M. Singh, P. Bafna, M. Vedaraj, and V. V. Priya, "Image forgery detection and classification using deep learning," in *2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0*, 2024, pp. 1–5.
[29] T. Pomari, G. Ruppert, E. Rezende, A. Rocha, and T. Carvalho, "Image splicing detection through illumination inconsistencies and deep learning," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 3788–3792.
[30] E. I. Abd El-Latif, A. Taha, and H. H. Zayed, "Digital images authentication technique based on multi-blocks dct and singular value decomposition," in *2023 5th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, 2023, pp. 200–204.



RISHIKESH RAVI is pursuing a Master of Applied Computing at Wilfrid Laurier University, Waterloo, ON, expected to graduate in December 2024. He earned his Master of Computer Applications from Veermata Jijabai Technological Institute, Mumbai, India in 2021.

From July 2021 to June 2023, he served as a Jr. Software Developer at Locobuzz Solutions in Mumbai, where he enhanced mobile applications for Android and cross-platform environments using Xamarin Forms. His contributions included optimizing performance and revamping user interfaces, which significantly improved user engagement. He also interned at Thomas Cook India Ltd., developing web applications and REST APIs.

With expertise in languages such as Java, C#, and Python, along with tools like Android Studio and AWS, Mr. Rishikesh Ravi is focused on software development and web application security. His recent projects include developing applications to mitigate XSS attacks and streamline job postings. He is dedicated to creating innovative software solutions that enhance user experiences.

HAMZA RANGWALA is pursuing a Master of Applied Computing at Wilfrid Laurier University, Waterloo, ON, expected to graduate in December 2024. He has completed significant coursework in artificial intelligence, data analytics, and machine learning.

From September 2023 to January 2024, he worked as a Research Assistant at Wilfrid Laurier University, contributing to projects focused on natural language processing and predictive analytics. Previously, as a Software Engineer at Solution-Wise, he developed scalable software solutions and implemented machine learning models for user discovery and geolocation tagging.

Proficient in Java, Python, JavaScript, and Dart, Mr. Hamza A. Rangwala has developed expertise in using frameworks and tools such as TensorFlow, Flutter, and AWS. His recent projects include Web application utilizing OpenAI's API for text analysis. He is passionate about leveraging AI and ML technologies to solve complex real-world problems.

TRUSHIT PATEL received the Bachelors of Technology degree in Computer Science and Engineering from the India University, Ahmedabad, in 2023 and is currently pursuing a Master's degree in Applied Computing at Wilfrid Laurier University, which he is expected to complete in December, 2024.

His professional experience includes roles as an AI Trainer and Software Development Engineer, focusing on AI data annotation, real-time applications, and scalable software solutions. Trushit's research and professional interests lie in artificial intelligence, software development, and microservices architecture.