Received Date:22nd December, 2024
Revision Date: 6th January, 2025
Accepted Date:15th January, 2025

Image Manipulation Detection and Localization

Nhujaw Tuladhar[1*], Pralisha Kansakar[2], Krishna B.K[3], Prayas Silwal[4], Sudeep Shakya[5]

[1]Dept. of Computer Engineering, Kathmandu Engineering College, E-mail: tuladhar135@gmail.com
[2]Dept. of Computer Engineering, Kathmandu Engineering College, E-mail: pralishakansakar@gmail.com
[3]Dept. of Computer Engineering, Kathmandu Engineering College, E-mail: kb3936884@gmail.com
[4]Dept. of Computer Engineering, Kathmandu Engineering College, E-mail: silwal.prayas@gmail.com
[5]Assoc. Professor, Dept. of Computer Engineering, Kathmandu Engineering College, E-mail:sudeep.shakya@kecktm.edu.np

Abstract —Image manipulation along with image splicing imposes a significant problem in the real world as it involves the act of maliciously combining parts from different images to create a forged image. This can lead to the spread of misinformation, deceptive visual narratives, and the decline of trust in authentic visual media. To address this challenge, a solution has been implemented by us which explores and evaluates various pre-processing methods such as ELA, JPEG compression, for the manipulation detection part and use of RGB, DCT channels along with CNN for the manipulation localization with the help of CASIA dataset. Our emphasis lies in the innovative application and integration of these methodologies to achieve a seamless and effective solution. The system takes in data in the form of an image file (JPG, JPEG, PNG), resizes it and feeds it to the machine learning algorithm. Experimental results are demonstrated that include training accuracy, validation (test) accuracy and loss.

Keywords — image splicing, copy-move, detection and localization, CNN, ELA, DCT, RGB, CASIA

Introduction

Image is a spatial representation of a scenery or objects. Image is a continuous media. Digital representation of image involves converting the spatial domain into frequency domain. Images can be two-dimensional or three-dimensional, static or dynamic, and can convey a wide range of emotions, ideas, and information. In the context of digital technology, an image often refers to a file containing visual data that can be displayed on a screen or printed. Images are used extensively in art, communication, science, entertainment, advertising, and many other fields.

Adobe published a research study on disinformation in 2020. In a total of 1,200 people (800 consumers and 400 creative professionals), consumers reported encountering false photos frequently at a rate of 63%, while creative professionals reported this rate at 72%. 69% of consumers and 76% of creative professionals worried that altered images might drive distrust in the news. 86% of creative professionals worried about having their work stolen or plagiarized in the future [1]. For instance, the photograph of a snow leopard captured in Nepal by American photographer Kitia Pawlowski, which gained significant attention on social media, was determined to be manipulated by experts [2]. Similarly, in 2016, Dinesh and Tarakeshwari Rathod, who claimed to be the first Indian couple to summit Mount Everest, were exposed for faking their achievement as an investigation by the Nepalese government revealed that the photos had been doctored [3].

Image manipulation is the practice of enhancing or modifying digital photographs using a variety of tools and techniques. In order to obtain the intended visual effects, it may be necessary to alter an image's appearance by adding or removing features, adjusting colors and tones, and using artistic effects. Images can be altered in a variety of ways, from straightforward alterations to more complex ones including cropping, resizing, color correction, eliminating objects or defects, adding texts and shapes, etc.

Through experimentation with various machine learning models, our research endeavors culminated in the development of a robust image manipulation detection system. After comprehensive evaluation, the Convolutional Neural Network (CNN) algorithm, coupled with Error Level Analysis (ELA) pre-processing, emerged as the most effective solution. As such, we adopted this algorithm for the core manipulation detection component of our project.

Similarly, in the pursuit of precise manipulation localization, our methodology involved integrating RGB and Discrete Cosine Transform (DCT) streams with the utilization of a custom CNN model inspired by the HRNet architecture. While our localization architectural implementation drew

* *Corresponding Author*

heavily from prior research, specifically referenced in the corresponding section, our emphasis lies in the synergy achieved through the amalgamation of the manipulation detection and localization models, showcasing the practical utility of our combined approach.

Architecture

CNN with ELA for image manipulation detection

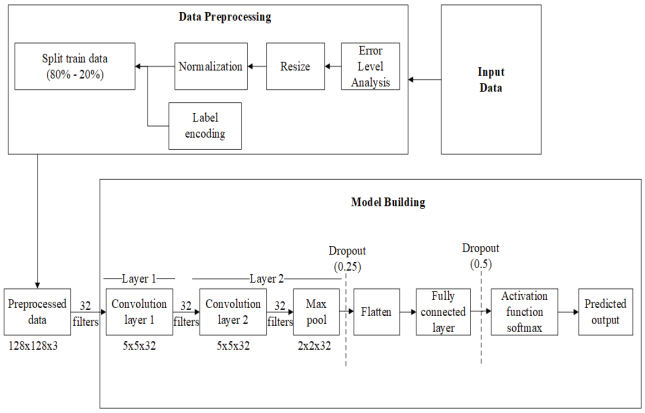


Figure 1: CNN model architecture for manipulation detection

A CNN works as a feedforward network, processing information is unidirectional way from input to output. Common CNN architectures include multiple convolutional and pooling layers, followed by one or more fully connected layers. In image classification, the CNN processes image inputs pixel by pixel. The model's first CNN layer comprises a 5x5 kernel and 32 filters, followed by a second layer with similar specifications, including a Max Pooling layer with a size of 2x2.

We evaluated the results of using only CNN, CNN with JPEG compression technique and finally, used ELA conversion followed by CNN for the manipulation detection. Our results showed the combination of ELA and CNN to have a higher accuracy.

Architectural design involves two main parts: data preparation and model building. This model was run for 30 epochs, with a learning rate of 0.0001 and a batch size of 32. Early stopping conditions were applied in order to prevent overfitting. In the initial stage, dataset of images with tampered and real labels are taken. Data preparation involves the following steps:

ELA conversion:

Converting raw data to ELA images increases CNN training efficiency by focusing on relevant information. The primary formula for calculating the ELA image is given by:

ELA (x, y) = | Reference (x, y) – Compressed (x, y) |

Resizing:

We convert each image using Error Level Analysis, followed by resizing to 128 x 128 pixels in RGB channel. Normalization: After resizing, the next step involves normalizing each RGB value by dividing it by 255 to ensure convergence speed during training, where RGB values range between 0 and 1.

Label encoding:

It assigns 1 to tampered and 0 to real, and the dataset is divided into 80% training data and 20% validation data. The pre-processed data is then passed into the model:

Convolution Layer:

Convolutional layers serve as feature extractors, learning representations from input images. There are limitations in forged images detected through ELA. This is typically applied to JPEG or PNG images. There are two convolution layers used with each layer having 32 number of filters and filter size of (5,5). Both convolutional layers use a uniform glorot initializer kernel and a ReLU activation function. The convolution operation works with the following formula:

(n,n)*(f,f)=(n-f+1,n-f+1)

Where, (n,n) is the initial size of image,

(f,f) is the filter size

and (n-f+1, n-f+1) is the matrix size after convolution.

Pooling Layer:

The pooling layer is responsible for reducing spatial resolution in feature maps. Before the fully connected layer, a stack of convolutional and pooling layers extracts more abstract feature representations. A (2,2) maxpool is used with a stride of 1.

Fully Connected Layer:

Initially, the fully connected layer has 256 neurons with a ReLU activation function. The last layer has 2 neurons with a SoftMax activation function for binary classification.

Activation Function:

The fully connected layer interprets these features, performing high-level reasoning, and the CNN classifies using an activation function. The output layer uses a SoftMax activation function whereas convolution layers use ReLU activation function.

Dropout:

Dropout is a regularization technique used to prevent overfitting by randomly deactivating neurons during training and effectively creating diverse sub-networks for improved generalization performance. To prevent overfitting, a dropout of 0.25 is applied after the MaxPooling layer. A dropout of 0.5 is added before the final dense layer so that the output doesn't depend heavily on a certain set of inputs.

Optimizer:

The Adam optimizer, an adaptive learning rate method, is applied during training for back propagation. It stands for adaptive moment estimation. This optimization algorithm is a stochastic gradient descent extension that updates network weights during training. It is a hybrid of the "gradient descent with momentum" and the "RMSProp" algorithms. It is used by us due to its fast convergence and stability.

Mathematically:

$$w_{t+1} = w_t - \alpha m_t$$

Where,

$$m_t = \beta m_{t-1} + (1-\beta)\left[\frac{\delta L}{\delta w_t}\right]|$$

Where, mt = Aggregate of gradients at time t [Current] (Initially, mt = 0)

m_{t-1} = Aggregate of gradients at time t-1 [Previous]

W_t = Weights at time t

W_{t+1} = Weights at time t+1

αt = Learning rate at time t

∂L = Derivative of Loss Function 29

∂W_t = Derivative of weights at time t

β = Moving average parameter (Constant, 0.9)

CNN with DCT and RGB analysis for image manipulation localization

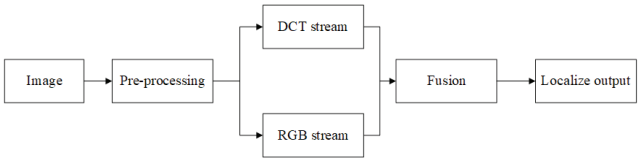


Figure 2: CNN with DCT and RGB analysis for image manipulation localization

In our localization methodology, we employed the HRNet architecture. This architecture includes an RGB stream, a DCT stream, and a final fusion stage. The RGB stream takes as input RGB pixels of the image and DCT stream takes input as Y-channel DCT coefficients and a Y-channel quantization table.

Image Preprocessing:

We preprocess the captured image if necessary. This involves tasks such as grid align cropping and component separation.

RGB Stream:

Most digital images are represented in RGB format, where each pixel's color is represented by three color channels: Red, Green, and Blue. Each channel typically has an 8-bit value ranging from 0 to 255, indicating the intensity of that color component.

DCT Stream:

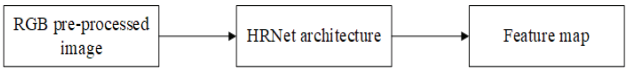
The DCT is a widely used technique in image and video compression. It converts spatial information (pixels) into frequency information (coefficients) using a mathematical transform. The DCT helps in capturing the image's frequency content, which is crucial for compression and efficient storage.

DCT Coefficients:

DCT coefficients represent the frequency components of an image. They indicate the strength of different frequency components in small blocks of the image. A JPEG quantization table is used which is typically an 8X8 matrix table of values that determine the level of compression applied to each DCT coefficient. 30 CNNs cannot automatically learn the compression artifacts from raw DCT coefficients because the convolution assumes a translation invariant property and handles every coefficient the same. However, the spatial coordinates are critical for DCT coefficients. Thus, we convert the input array of DCT coefficients to a binary volume using some transformation.

Fusion:

After separately processing the RGB images and DCT coefficients, the extracted features from both streams are combined or fused. The fused features are then used for further processing and learning within the CNN through bilinear interpolation (up sampling) or strided convolution (down sampling) operations. This joint learning process allows the network to integrate the complementary information from RGB and DCT domains, enhancing the network's ability to capture both spatial and frequency related aspects of the data.

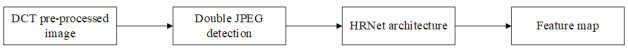


RGB Stream

Figure 3: Block diagram for RGB stream

We initialized the weights of the network for RGB stream by pretraining on ImageNet dataset [4] in the HRNet CNN architecture. It includes learning of visual traces which consist of detecting edges, textures, shapes, and other visual features. The weights of the CNN layers, are initialized with the pre-trained weights obtained from ImageNet to recognize and classify objects in the ImageNet dataset. The RGB stream focuses on visual clues to extract features.

HRNet is used to extract visual features from the image. It utilizes multiple paths that progressively process the image at different resolutions while still maintaining the original high resolution. Each resolution path goes through repetitive processing units called blocks that perform filtering and combining operations to extract features. Lower resolution paths help capture the overall structure of the image. Higher resolution paths maintain fine details and subtle changes that might indicate tampering. HRNet combines information from all these resolutions to generate the final output of the RGB stream.



DCT Stream

Figure 4: Block diagram for DCT stream

DCT stream weights were initialized by pretraining on double JPEG detection method [5]. This algorithm provides output on whether an image has been compressed once or twice. Pretraining the DCT stream helps in capturing compression artifacts.

HRNet is used to extract features from compression artifacts, which are traces left behind by the way the image was compressed.

Pretraining DCT artifact using double jpeg detection involves training a model to recognize and interpret compression artifacts by analyzing the DCT coefficients and their characteristics in compressed images to extract relevant features and patterns that can distinguish between double compressed and single compressed/authentic images.

The basic block structure is similar to the RGB stream, but the first block is replaced with a JPEG artifact learning module. This module transforms the DCT coefficients into a special format that highlights patterns related to compression. HRNet then analyzes these patterns at different resolutions using multiple paths, looking for inconsistencies that might suggest tampering.

Similar to the RGB stream, HRNet combines information from all resolutions to generate the final output of the DCT stream. Output feature maps have resolutions (1/4, 1/8, 1/16, 1/32) and (1/8, 1/16, 1/32) for the RGB and DCT streams, respectively. The outputs from the two-stream feature maps are concatenated resolution-wise in the channel dimension and passed to the final fusion stage. This stage also uses HRNet architecture. The final output of the model is a 2-dimensional array (logits) for each class (authentic and tampered), where each element corresponds to a certain class prediction at a particular spatial location in the image. The array has dimensions of 2 × (H/4) × (W/4), where H and W are the height and width of the original image, respectively, divided by 4 to account for down sampling during feature extraction.

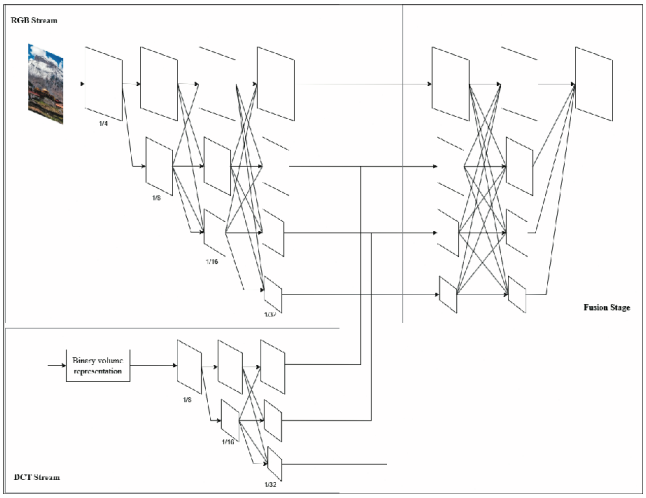


Figure 5: Implementation model architecture

The implemented architecture includes an RGB stream, a DCT stream, and a final fusion stage. The RGB stream takes RGB pixels and the DCT stream takes Y-channel DCT coefficients and a Y-channel quantization table as input. The RGB stream learns visual artifacts from a picture on a pixel level, the DCT stream learns DCT coefficients distributions (compression artifacts) from the frequency domain of JPEG images. The fusion stage then fuses the output feature maps from these two streams and generates the final mask.

The RGB stream is used so that the model can handle non-JPEG images. Since non-JPEG images do not contain DCT coefficients, the RGB pixel values are used for such images. Further, the quantization table for such images are considered

as JPEG image with quality 100. The input consists of RGB pixel values, and the initial convolutional unit reduces the resolution by a factor of 4. Progressing from the high-resolution path, the stream progressively incorporates high-to-low resolution paths, adding them one by one and establishing connections between multi-resolution paths in parallel. Each resolution level is retained throughout the network until the conclusion, resulting in output resolutions of 1/4, 1/8, 1/16, and 1/32.

The DCT is a widely used technique in image and video compression. It converts spatial information (pixels) into frequency information (coefficients) using a mathematical transform. The DCT stream is designed to capture compression artifacts by analyzing the statistical distributions of Y-channel Discrete Cosine Transform coefficients. A quantization table is typically an 8X8 matrix table of values that determine the level of compression applied to each DCT coefficient. The input array of DCT coefficients is initially transformed into a binary volume representation using the quantization table. This is because CNNs cannot automatically learn the compression artifacts from raw DCT coefficients as the convolution assumes a translation invariant property and handles every coefficient the same. The output feature maps of the DCT stream have resolutions of 1/8, 1/16 and 1/32.

The results from both sets of feature maps are combined in the final fusion stage, which follows the HRNet architecture. This stage processes the combined features to produce the model's final output. This output is an array of predictions for two classes (authentic and tampered) at different spatial locations in the image. The dimensions of this array are $2 \times (H/4) \times (W/4)$, where H and W denote the height and width of the original image, respectively. These dimensions are divided by 4 to accommodate the down sampling that occurred during the feature extraction process.

Result and Analysis

Table 1

Score of different methods for manipulation detection

	Training accuracy	Training loss	Validation accuracy	Validation loss
CNN	0.6741	0.6054	0.672	0.68
JPEG compression + CNN	0.9782	0.0533	0.8886	0.5674
ELA + CNN	0.9512	0.4926	0.9356	0.5221

Table 2

Training and validation scores for manipulation localization

	Training accuracy	Training loss	Mean IU	avg p mIoU	Validation accuracy	Validation loss
CNN+DCT+RGB	0.889546	0.110454	0.7217	0.6261	0.8996	0.048

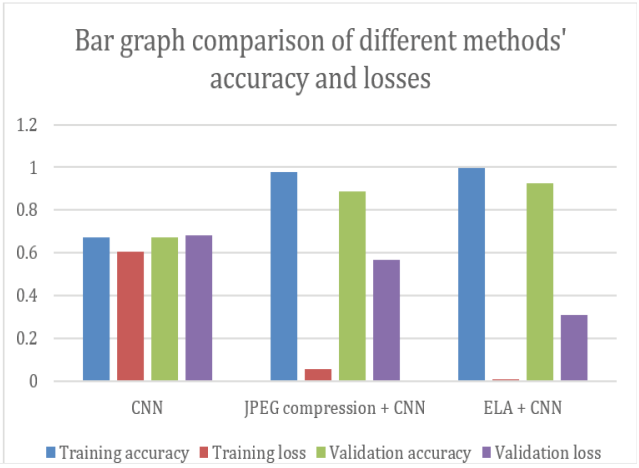


Figure 6: Bar graph comparison for different image manipulation detection methods

With the analysis of above bar graph, we get to know that CNN with ELA pre-processing has highest accuracy score, 0.9814, among all of the algorithms along with least mean squared error, 0.0186, whereas the least performing algorithm on the provided dataset was pure CNN with accuracy score 0.6741 and mean squares loss 0.6054. Hence, we use CNN with ELA for implementation of image manipulation.

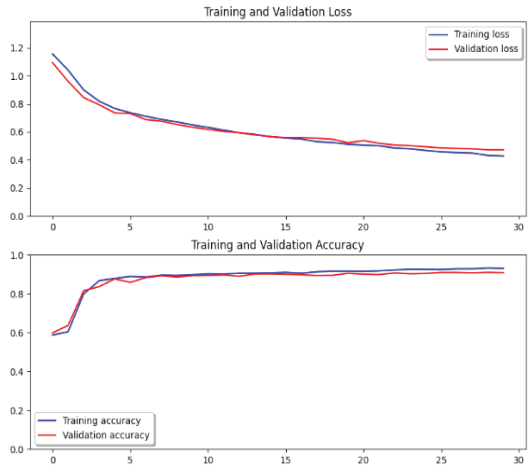


Figure 7: Training and validation accuracy and loss graph for CNN with ELA for manipulation detection

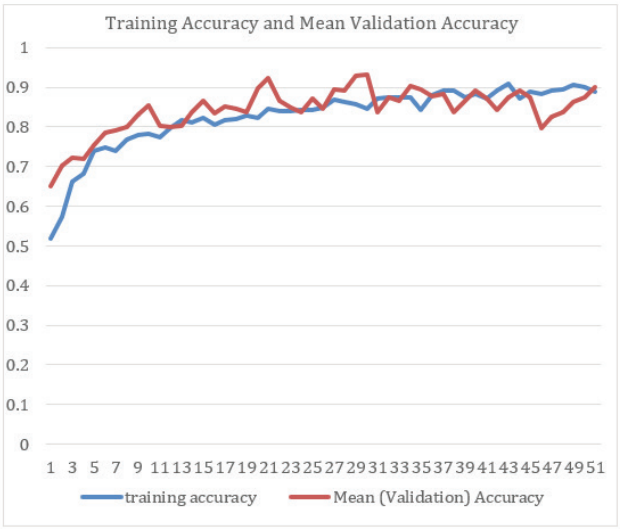


Figure 8: Training and validation accuracy graph for CNN with DCT and RGB analysis for manipulation localization

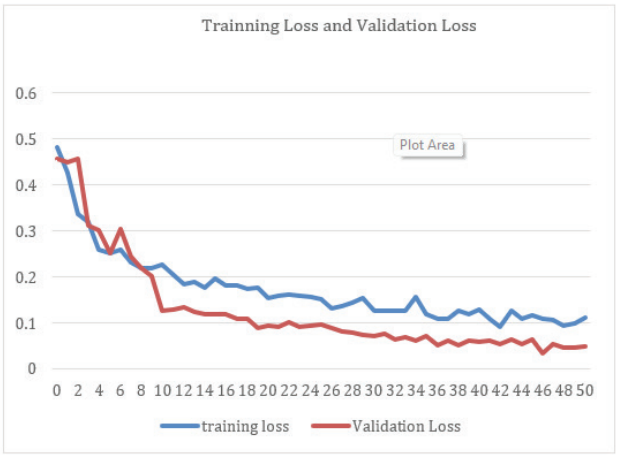


Figure 9: Training and validation loss graph for CNN with DCT and RGB analysis for manipulation localization

While overfitting concerns the model's adaptation to the training data, monitoring the performance on our validation set provides insights into its ability to generalize. A consistently high accuracy and low loss on the validation set indicate that the model is learning relevant patterns without solely memorizing the training data.

In addition to the aforementioned measures, our model's efficacy has been validated within the local context of Nepali images. The performance metrics, when applied to Nepali-specific data, exhibit promising results. This validation process enhances our confidence in the model's adaptability to diverse cultural and contextual nuances, reinforcing its reliability in real-world scenarios.

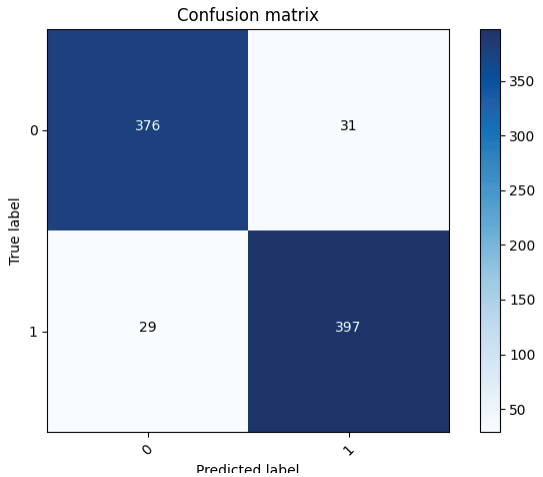


Figure 10: Confusion matrix for CNN with ELA

A confusion matrix is a tabular representation used in evaluation of the performance of a classification model. It provides a summary of the actual and predicted classifications made by the model on a dataset.

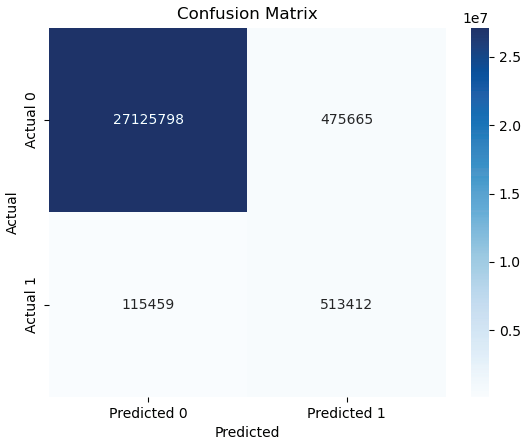


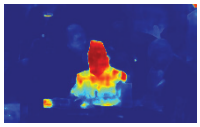
Figure 11: Confusion matrix for CNN with DCT and RGB

In assessing the performance of our Convolutional Neural Network (CNN) integrated with the Error Level Analysis (ELA) method, a comprehensive evaluation was conducted. The matrix provided below encapsulates the outcomes, illustrating 397 instances of true positives and 376 instances of true negatives.

Since the image localization occurs at the pixel level for CNN approach with DCT and RGB, our evaluation extends beyond image-level considerations. The assessment examines each individual pixel across the 106 test images, each image of size 512x512 pixels, discerning the accuracy of predictions concerning tampered or untampered status based on the ground truth. The confusion matrix thus offers

a granular perspective, detailing the count of true positive, false positive, false negative and true negative cases for pixel-wise predictions.

Evaluation

S.N	Raw Image	Result	DCT+RGB Localization
1.		Tampered 99.77	
2.		Authentic 99.68	
3.		Tampered 99.94	
4.		Authentic 100.00	
5.		Tampered 99.78	

For our validation process, we employed a custom dataset comprising a diverse range of images, incorporating cultural and regional contexts. The assessment revealed the robust capability of our system to accurately discern tampered and authentic images. The integration of Convolutional Neural Networks (CNN) with Error Level Analysis (ELA) methodology demonstrated a high level of accuracy in classifying images as tampered or genuine. It is noteworthy, however, that our system, while highly effective, does not consistently yield a 100% probability for tampered or real classifications.

Moreover, our localization component, employing CNN with Red-Green-Blue (RGB) and Discrete Cosine Transform (DCT) streams, exhibited proficiency in identifying tampered regions within images. The localization mechanism effectively highlights these areas using warmer color representations. These findings collectively underscore the competence of our system in detecting image tampering and providing valuable insights into the manipulated regions, despite not achieving absolute certainty in all cases.

Conclusion

By training various machine learning models, we got the best accuracy for CNN algorithm along with ELA pre-processing with a training accuracy of 0.9512 and validation accuracy of 0.9356. Hence, we used this algorithm for the manipulation detection part of our project.

For localization of manipulated region, we used a fusion of RGB and DCT streams for pre-processing and fed the data into a custom CNN model by following the implementation of HRNet architecture in order to localize the results in the form of a heat map. The training accuracy for this model was 0.889 and testing accuracy was found to be 0.899. The integration of image manipulation detection and localization methodologies yielded a proficient solution across a majority of cases.

References

1. https://blog.adobe.com/en/publish/2020/12/08/adobe-research-disinformation.
2. https://mteveresttoday.com/viral-images-of-a-snow-leopard-taken-by-american photographer-is-fake/.
3. https://www.theguardian.com/world/2016/aug/30/indian-couple-banned-from climbing-after-faking-ascent-of-everest
4. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, (pp. 1097–1105).
5. Verma, V., Singh, D., & Khanna, N. (2020). Block-level double JPEG compression detection for image forgery localization. arXiv preprint arXiv:2003.09393
6. Kwon, M.-J., Yu, I.-J., Nam, S.-H., & Lee, H.-K. (2021). CAT-Net: Compression Artifact Tracing Network for Detection and Localization of Image Splicing. Korea Advanced Institute of Science and Technology (KAIST).
7. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., & Xiao, B. (2019). Deep high-resolution representation learning for visual recognition. arXiv. https://doi.org/10.48550/arXiv.1908.07919