Dissertation on

Image Tampering Detection

Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

UE20CS390A – Capstone Project Phase - 1

Submitted by:

Anil Kumar MG	**PES1UG21CS804**
Divya M	**PES1UG21CS810**
Nagaratna Naik	**PES1UG21CS821**
Nandana CL	**PES1UG21CS822**

Under the guidance of

Dr. Sujatha R Upadhyaya
Professor

January - May 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100 Feet Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

Image Tampering Detection

is a bonafide work carried out by

Anil Kumar MG	**PES1UG21CS804**
Divya M	**PES1UG21CS810**
Nagaratna Naik	**PES1UG21CS821**
Nadana CL	**PES1UG21CS822**

in partial fulfilment for the completion of sixth-semester Capstone Project Phase - 1 (UE19CS390A) in the Program of Study - **Bachelor of Technology in Computer Science and Engineering** under rules and regulations of PES University, Bengaluru during the period Jan. 2022 – May. 2022. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 6th-semester academic requirements in respect of project work.

Signature	Signature	Signature
Dr. Sujatha R Upadhyaya	Dr Shylaja S S	Dr B K Keshavan
Designation	Chairperson	Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. ____________________________	____________________________
2. ____________________________	____________________________

DECLARATION

We hereby declare that the Capstone Project Phase - 1 entitled **"Image Tampering Detection"** has been carried out by us under the guidance of Dr Sujatha R Upadhyaya, Professor and submitted in partial fulfilment of the completion of the sixth semester of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester January – May 2022. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES1UG21CS804	**Anil Kumar MG**
PES1UG21CS810	**Divya M**
PES1UG21CS821	**Nagaratna Naik**
PES1UG21CS822	**Nandana CL**

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr Sujatha R Upadhyaya, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE19CS390A - Capstone Project Phase – 1.

We are grateful to the project coordinator, Dr Priyanka H., all the panel members & the supporting staff for organising, managing, and helping the entire process.

We take this opportunity to thank Dr Shylaja S S, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support we have received from her.

We are grateful to Dr M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro-Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, Dr B.K. Keshavan, Dean of Faculty, PES University for providing us with various opportunities and enlightenment during every step of the way.

Finally, this project could not have been completed without the continual support and encouragement we have received from our family and friends.

ABSTRACT

The increasing availability of digital image editing tools has made it easier to manipulate images, leading to an increase in tampered images online. These tampered images are being used for various malicious purposes, including spreading fake news or even defaming someone. Detecting tampered images manually is time-consuming and prone to errors, hence the need for an automated system to identify tampered images accurately and efficiently.

This project aims to develop a system that uses Convolutional Neural Networks (CNNs) to detect tampered images and identify the specific areas of tampering. We will be training the system on a dataset of both authentic and tampered images. The successful implementation of this project will contribute to the development of an automated system that can accurately detect image tampering, providing an essential tool for combating fake news and other malicious activities.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Image tampering is a common problem in the digital age, where the authenticity and integrity of images can be compromised easily. The increasing availability of image editing tools has made it easier for people with malicious intent to tamper with digital images for fraudulent or deceptive purposes. With the growing reliance on visual media for communication, the impact of manipulated images can be significant.

In this capstone project, we propose a CNN-based model for detecting image tampering. The model is trained on a dataset of tampered and authentic images. The trained model is then used to develop a website that allows users to upload an image and determine whether it has been tampered with. If the image is tampered with, the website provides information about the areas that have been tampered with.

The proposed solution has the potential to be used by media organisations, law enforcement agencies, and individuals to verify the authenticity of images. This project aims to contribute to the field of image tampering detection by proposing an accurate and user-friendly solution.

CHAPTER 2

PROBLEM STATEMENT

The widespread use of digital images for communication and documentation has made it increasingly easy for people to tamper with images using image editing tools. This has led to the rise of image tampering, which can have severe consequences in fields such as journalism, forensics, and scientific research. The detection of tampered images has thus become a critical task in computer vision.

Therefore, we propose a CNN-based model for detecting tampered images. The goal is to develop a user-friendly and scalable solution that can be easily used by anyone to identify if the input image is tampered with or not. Furthermore, the model can identify the areas that have been tampered with, providing additional information for further analysis. The proposed solution can be used by media organisations, law enforcement agencies, and individuals to check if the image is tampered or not and protect against the spread of false information.

CHAPTER 3

LITERATURE SURVEY

3.1 Learning Rich Features for Image Manipulation Detection

Abstract:

In this research paper, they proposed a 2-stream faster R-CNN network for image tampering detection.

The 2 streams are

1. RGB stream: Has extracted features from input RGB image.
2. Noise stream: Has extracted noise features.

Datasets used:

NIST: Contains spliced, copy-move, removal images

CASIA: Contains spliced, copy-move and removal images of various objects.

Columbia: Contains only spliced images

COVER: Contains only copy-move images

Methodology:

1. Using RGB stream extract features from RGB images.
2. Pass the RGB image through SRM(Steganalysis Rich Model) filter and obtain a noise feature map.
3. RPN network uses RGB features as input
4. Then features from both streams are selected by an RoI(Region of Interest) pooling layer.
5. Then using a bilinear pooling layer the spatial co-occurrence features are combined.
6. Then pass it to the softmax layer which determines whether the predicted area is tampered with or not

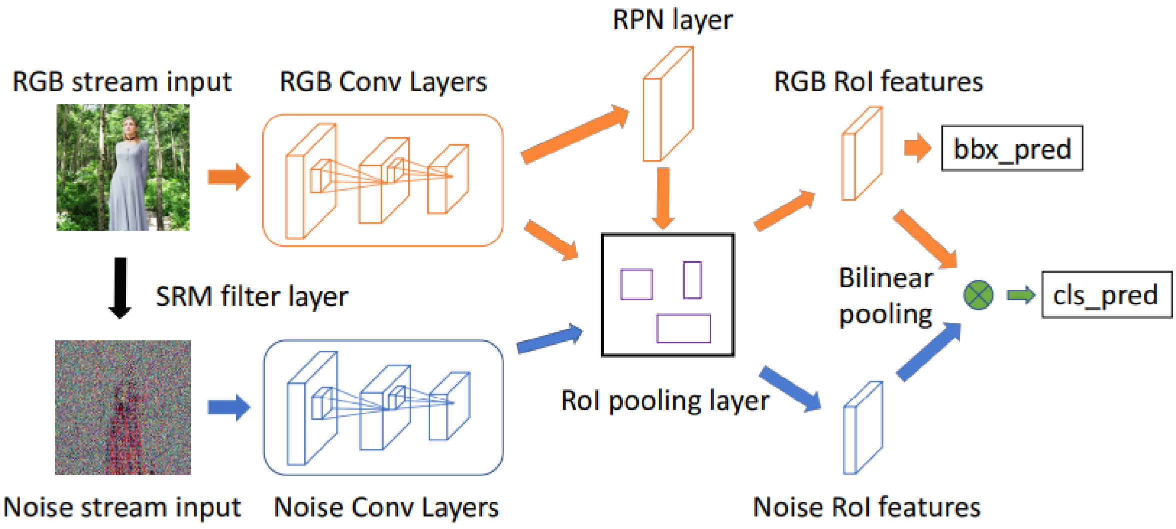
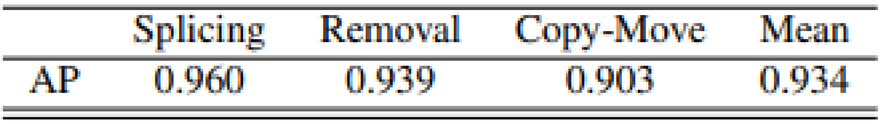


Figure 1: R-CNN Architecture

Accuracy:



	Splicing	Removal	Copy-Move	Mean
AP	0.960	0.939	0.903	0.934

Table 1: Accuracy comparison of R-CNN

3.2 Blind Forensics of Images using Higher Order Local Binary Pattern

Abstract:

In this research paper, they proposed an image tampering detection system which uses Local Binary Pattern and SVM.

Datasets used:

CASIA: Contains spliced, copy-move and removal images of various objects.

Columbia: Contains only spliced images.

Methodology:

1. Convert RGB image to YCbCr.
2. Extract the Cb(Chroma Blue) and Cr(Chroma Red) components of images.
3. LBP and LBP2(Local Binary Pattern) are calculated.
4. Histograms of LBP and LBP2 are calculated.
5. Concatenate the histograms of LBP and LBP2.
6. Normalize the feature vector.
7. Divide the feature vectors for training and testing.
8. Tampered images are labelled as 1 and authentic images are labelled as 2 in training.
9. SVM binary classifier is constructed.
10. Using SVM classify the images based on feature vectors.

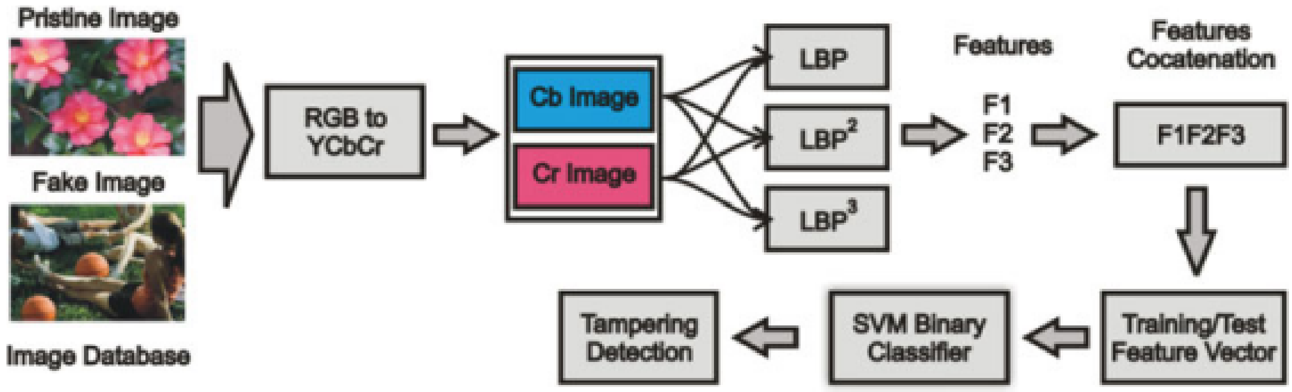
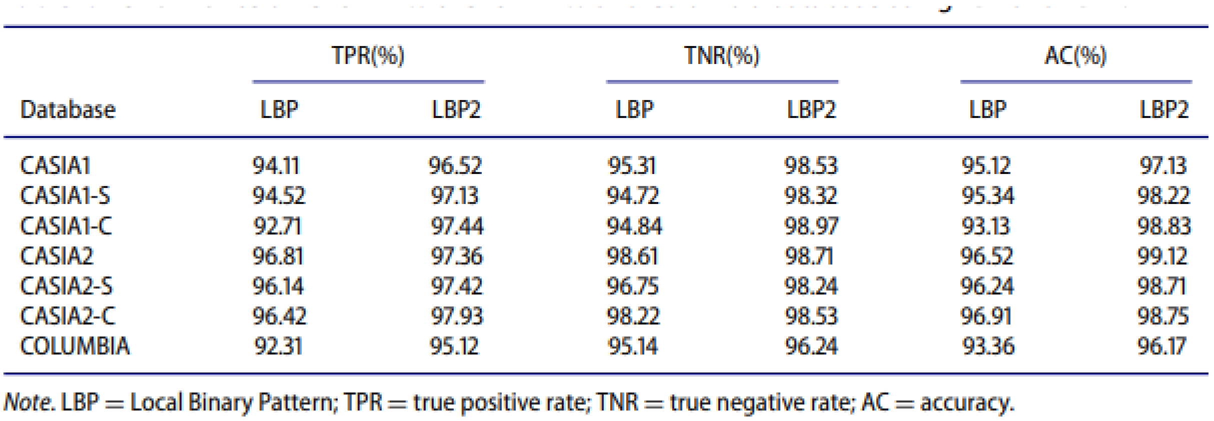


Figure 1: Higher Order LBP Architecture

Accuracy:



Database	TPR(%)		TNR(%)		AC(%)	
	LBP	LBP2	LBP	LBP2	LBP	LBP2
CASIA1	94.11	96.52	95.31	98.53	95.12	97.13
CASIA1-S	94.52	97.13	94.72	98.32	95.34	98.22
CASIA1-C	92.71	97.44	94.84	98.97	93.13	98.83
CASIA2	96.81	97.36	98.61	98.71	96.52	99.12
CASIA2-S	96.14	97.42	96.75	98.24	96.24	98.71
CASIA2-C	96.42	97.93	98.22	98.53	96.91	98.75
COLUMBIA	92.31	95.12	95.14	96.24	93.36	96.17

Note. LBP = Local Binary Pattern; TPR = true positive rate; TNR = true negative rate; AC = accuracy.

Table 1: Accuracy comparison of LBP

3.3 Detection of Image Tampering Using Deep Learning, Error Levels & Noise Residuals

Abstract:

The authors of the research paper presented a method to differentiate between authentic and tampered images by utilizing a dual-branch Convolutional Neural Network, in conjunction with Error Level Analysis and noise residuals obtained from the Spatial Rich Model.

Datasets used:

CASIA v2.0: Contains both spliced and copy-move images, but has more images

Methodology:

1. As we have already mentioned two pre-processing steps for further understanding of the learning of our CNN.

2. First the ELA and noise residual image for all images in the dataset.

3. To obtain noise residual images for their analysis, the researchers used 30 high-pass filters that were specifically created for the Spatial Rich Model (SRM).).

4. Then, we resize the pictures and include the labels in a corresponding manner. Later we divide the information into two sets,

5. The researchers utilized two separate data sets, where one was utilized to train the model and the other was used to assess and confirm its performance.

6. ELA resaves an image at a pre-determined compressed rate e.g. 95% and then checks the difference between the original and the resaved image.

7. Another technique that we added is to extract features from the noise residuals of an image using SRM.

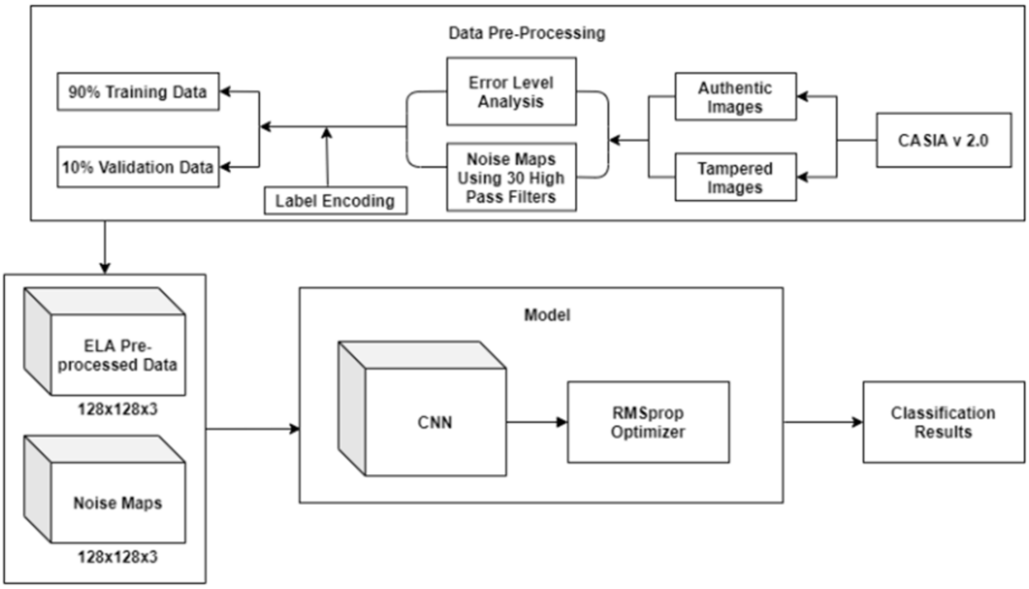
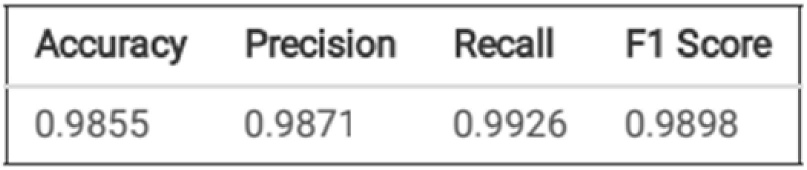


Figure 3: ELA and noise map Architecture

Accuracy:-



Accuracy	Precision	Recall	F1 Score
0.9855	0.9871	0.9926	0.9898

Table 3: Accuracy comparison of ELA and noise map

3.4 Deep Learning-based Technique for Image Tamper Detection

Abstract:

In this research paper, significant developments in passive image forensic analysis methods adopting deep learning techniques and Convolution Neural Networks (CNN)

Datasets used:

There are several image tampering datasets available for research purposes, including MICC, CASIA, and UCID, among others.

Methodology:

First, the image is segmented into different patches.

To identify duplicated regions even in small and smooth areas, a scale-invariant descriptor was utilized to extract features.

Various types of noise, such as compression and resampling, are often present in tampered images and can be effectively identified by analyzing resampling features.

However, using resampling features exhibit periodic correlations between the pixels; this is because of interpolation.

The CNN is robust to translation for capturing noisy information using resampling features and generating spatial maps among different segments of an image;

Thus both are utilized for localizing tampered segments. Thus, the resampling feature will be estimated for detecting inconsistencies in the estimated resampling factors using improved CNN architecture

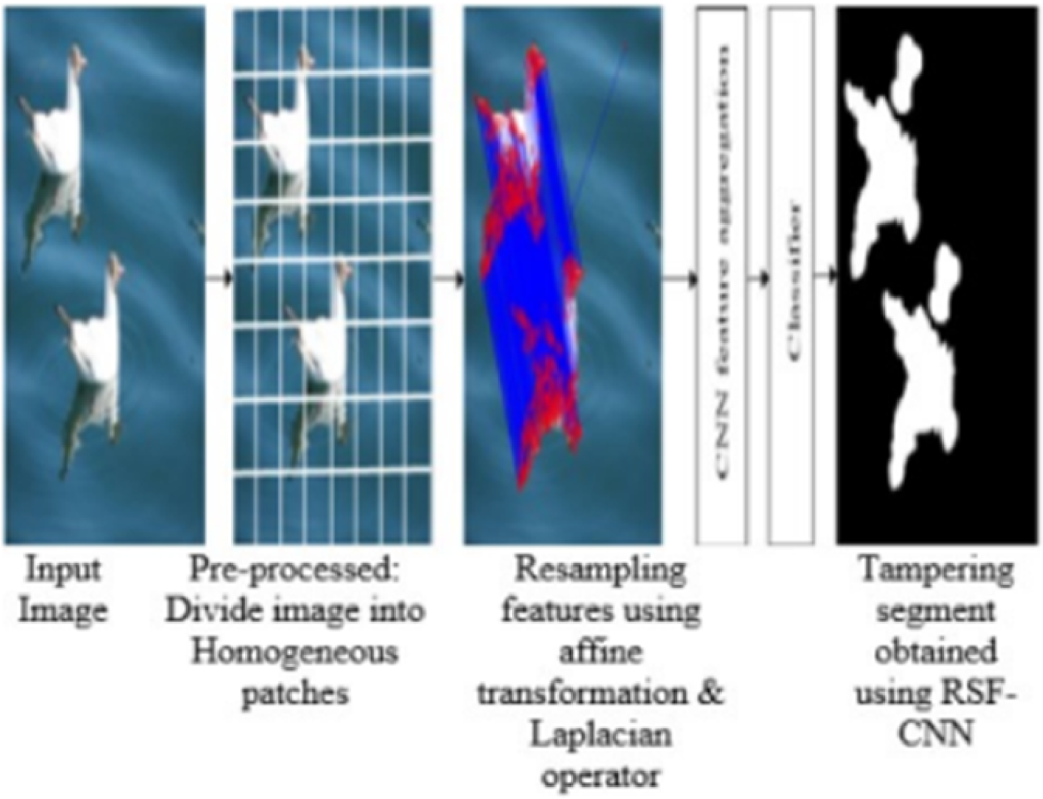
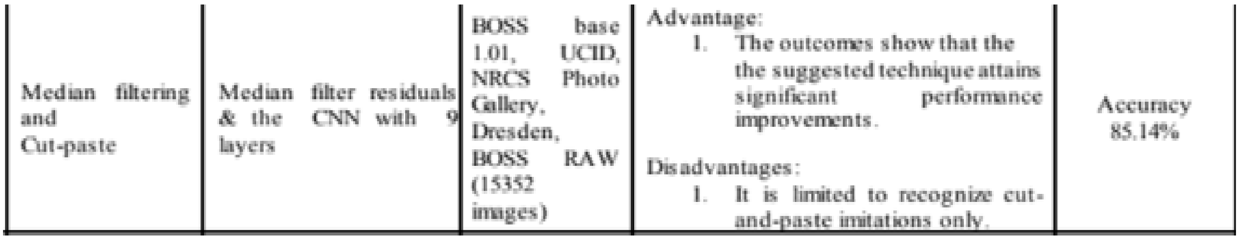

Input Image
Pre-processed: Divide image into Homogeneous patches
Resampling features using affine transformation & Laplacian operator
CNN feature aggregation
Classifier
Tampering segment obtained using RSF-CNN

Figure 4: RSF CNN

Accuracy:



Median filtering and Cut-paste	Median filter residuals & the CNN with 9 layers	BOSS base 1.01, UCID, NRCS Photo Gallery, Dresden, BOSS RAW (15352 images)	Advantage: 1. The outcomes show that the the suggested technique attains significant performance improvements. Disadvantages: 1. It is limited to recognize cut-and-paste imitations only.	Accuracy 85.14%

Table 4: Accuracy of RSF CNN

3.5 Scientific Image Tampering Detection Based On Noise Inconsistencies: A Method And Datasets

Abstract:

The research paper introduces a novel tampering detection approach specifically designed for scientific images that utilize noise inconsistencies. This method is able to learn and apply to various scientific fields, demonstrating its generalizability.

Datasets used:

The newly proposed method was trained and tested on a dataset comprising manipulated western blot and microscopy images.

Methodology:

At first, an input image will go through a variable amount of residual image generators

There are various methods for generating residual images, but the following ones are commonly used due to their versatility and suitability for a wide range of applications: (more information can be added if needed).

1. Steganalytic Filters:-Steganalysis (techniques used for detecting hidden messages in communications)

2. Error Level Analysis (ELA) ELA is an analysis technique that targets JPEG compression.

3. Median Filtering Residual Median filtering can suppress the noise of an image. When applying median filtering to a tampered image.

4. Wavelet Denoising Residual Wavelet denoising is a type of denoising method that represents an image in the wavelet domain and cancels the noise based on that representation.

Feature Extraction as our method is Patch-based which means it will generate a prediction for each path in the image Using patches instead of single pixels

Next the Feature Extraction Step will generate a corresponding feature vector for each path in the image.

Patch Reinterpretation reduces the complexity of image data, but they still have the same dimensionality as the original image

First an input image of size (h,w) will be divided into patches of size (m,n)

if the image is not divisible the images will be cropped

Then an image of size (h,w) will be divided into a patch matrix of size ([h/m],[w/n])

Then the patch matrix will be spilt into a rectangular patch gris of size (s,t) where each cell contains a certain number of patches

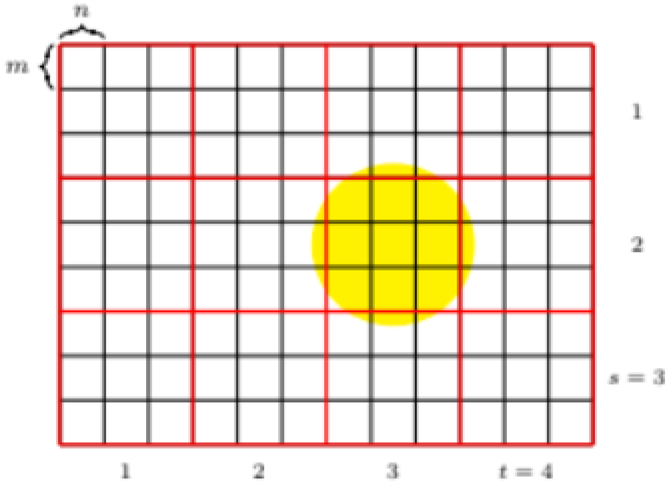


Figure 5: Patch matrix

In the image, the black blocks correspond to patches, the red blocks indicate grid cells, and the yellow region represents the tampered region. In this case, (s, t) = (3, 4).

Because the tampered region has a different residual pattern, and its contaminated patches concentrate in one of the cells, the outlier detector of that cell will learn a distinct decision boundary compared to other ones.

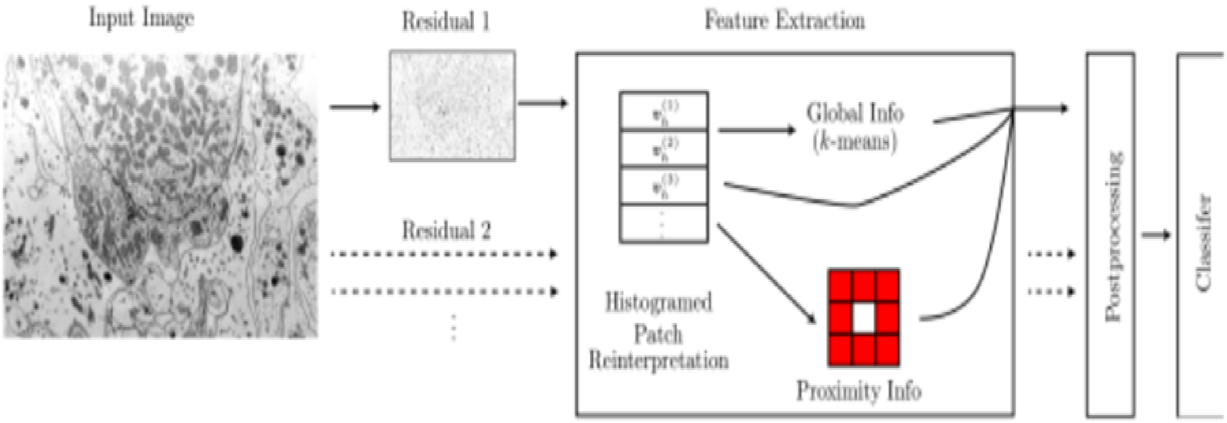
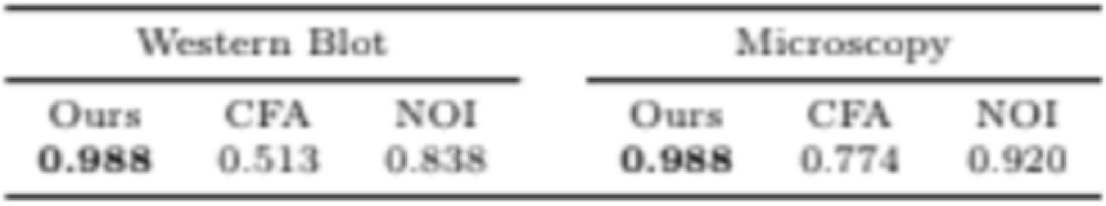


Figure 6: Noise Residual Architecture

Accuracy:



Western Blot			Microscopy		
Ours	CFA	NOI	Ours	CFA	NOI
0.988	0.513	0.838	**0.988**	0.774	0.920

Table 5: Accuracy comparison of Noise residual architecture

3.6 Digital Image Forensics An Affine Transform Robust Copy-Paste Tampering Detection

Abstract:

In this paper, copy-move tampering detection can be achieved by following 2 approaches. One is block-based and the other is key point based. and the proposed method is used.

Dataset Used:

CoMoFoD - it contains 260 forged images.

Methodology:

In this paper, they used 2 methods-block based and key based.

1. Block-based method:

The whole image is divided into overlapping or non-overlapping blocks and then they are processed to extract the features.

2. key point-based method:

The features are collected by calculating local key points for the whole image.

1. The positions of each block or key point are also stored in the feature vector. Then, feature matching is performed to find similar features within the same image.
2. Once that is done, the tampering localization is done by displaying the matched blocks or key points in colours corresponding to the locations of the matched features. Then, a proposed method is applied.
3. The proposed Copy-Paste tampering detection method is based on a block matching approach and uses Normalized Cross-Correlation.

1) Angle of Rotation and Scaling Detection

2) Coarse Scaled Rotated Tamper Detection (CSRTD)

3) Fine Scaled Rotated Tamper Detection (FSRTD)

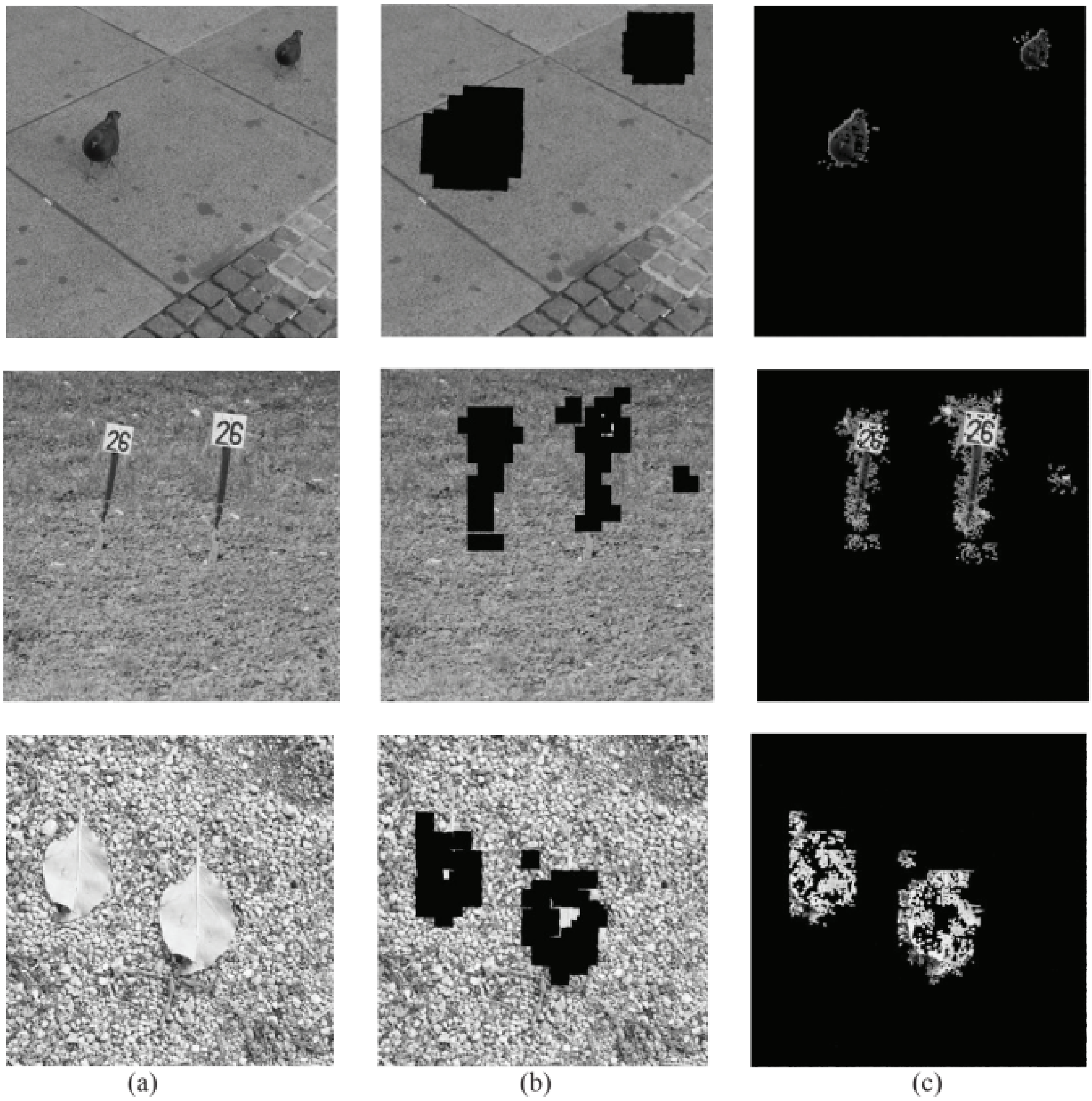


Figure 6.Original images from CoMoFoD [27] image database with scaled and rotated region tampering

3.7 Image Forgery Identification using Convolution Neural Network

Abstract:

In this research paper, they proposed a method for image tampering detection that uses the proposed method, SPAM, SRM, and SVM.

Dataset Used:

CASIA V1.0

CASIA V2.0

Methodology:

1. Convert RGB image to YCbCr.
2. Extract the Cb(Chroma Blue) and Cr(Chroma Red) components of images.
3. Feature is extracted by pre-processing.
4. Thresholding is optimized to separate foreground and background regions of an image.
5. Normalize the feature vector.
6. CNN is applied to extract features from the input of an image.
7. classifies the authentic and tampered image.
8. do post-processing and will get a detected region.

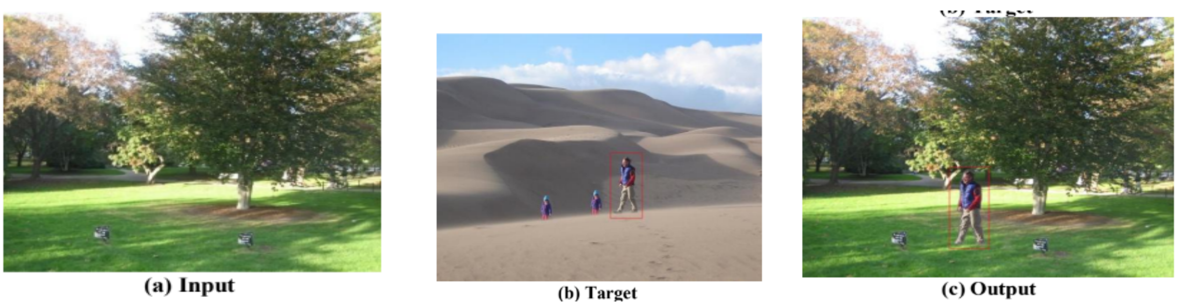
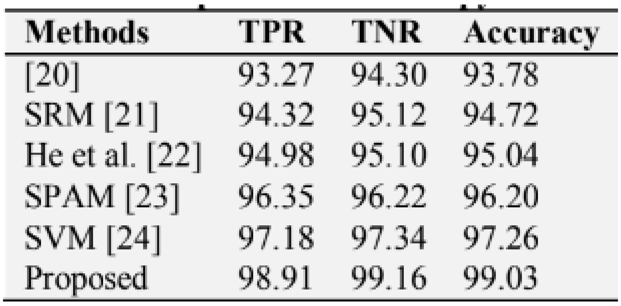

(a) Input (b) Target (c) Output

Figure7. Copy-move forgery detection

Accuracy:



Methods	TPR	TNR	Accuracy
[20]	93.27	94.30	93.78
SRM [21]	94.32	95.12	94.72
He et al. [22]	94.98	95.10	95.04
SPAM [23]	96.35	96.22	96.20
SVM [24]	97.18	97.34	97.26
Proposed	98.91	99.16	99.03

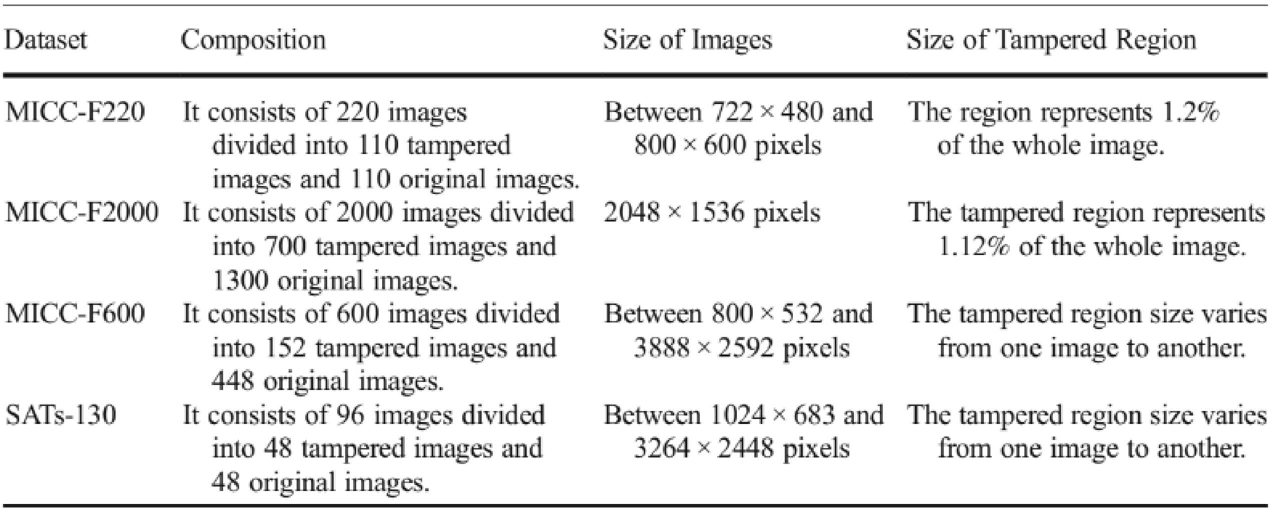
Table 7:Accuracy of copy-move forgery detection.

3.8 A novel deep learning framework for copy-move forgery detection in Images

Abstract

In this paper, a new, fast, easy, and efficient CMFD algorithm is configured and tested. The proposed algorithm is based on using CNN to develop a deep learning method to detect copy-move forgeries in a fast and robust manner.

Dataset



Dataset	Composition	Size of Images	Size of Tampered Region
MICC-F220	It consists of 220 images divided into 110 tampered images and 110 original images.	Between 722 × 480 and 800 × 600 pixels	The region represents 1.2% of the whole image.
MICC-F2000	It consists of 2000 images divided into 700 tampered images and 1300 original images.	2048 × 1536 pixels	The tampered region represents 1.12% of the whole image.
MICC-F600	It consists of 600 images divided into 152 tampered images and 448 original images.	Between 800 × 532 and 3888 × 2592 pixels	The tampered region size varies from one image to another.
SATs-130	It consists of 96 images divided into 48 tampered images and 48 original images.	Between 1024 × 683 and 3264 × 2448 pixels	The tampered region size varies from one image to another.

Methodology:

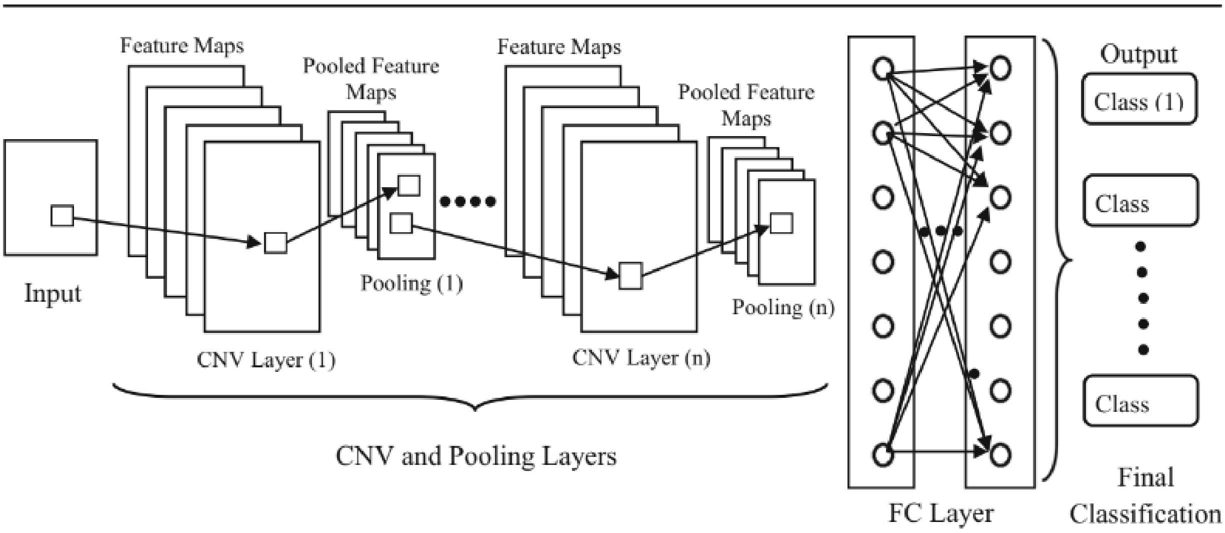


Fig 8:CNN Architecture

Accuracy:

87.12

CHAPTER 4

DATASET

4.1 CASIA

4.1.1 CASIA v1.0

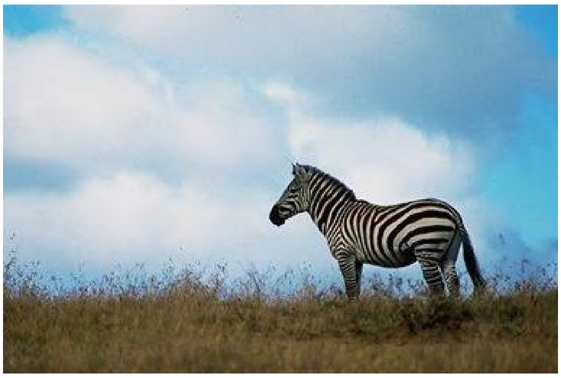
This dataset is an image manipulation dataset that was created by the Chinese Academy of Sciences' Institute of Automation (CASIA) to aid in the development of image tampering detection algorithms. The dataset contains 800 authentic images and 921 tampered images, where tampered images are those that have undergone various forms of manipulation, such as copy-move, splicing, and removal.

The tampered images were created using various techniques, including manual editing, copy-pasting, and compression. The dataset also includes ground truth annotations for each tampered image, indicating the specific region of the image that was tampered with.

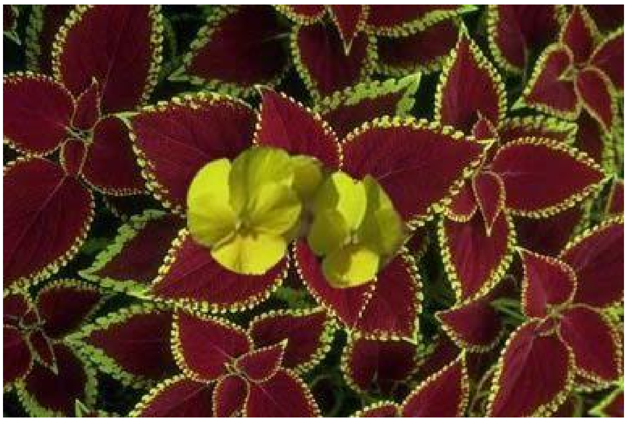
The CASIA v1.0 dataset is widely used by researchers in the field of image tampering detection to test the effectiveness of their algorithms. It's large size and diverse set of tampering techniques make it a valuable resource for developing robust image tampering detection methods.

Figure 9: Sample images from CASIA v1.0 dataset

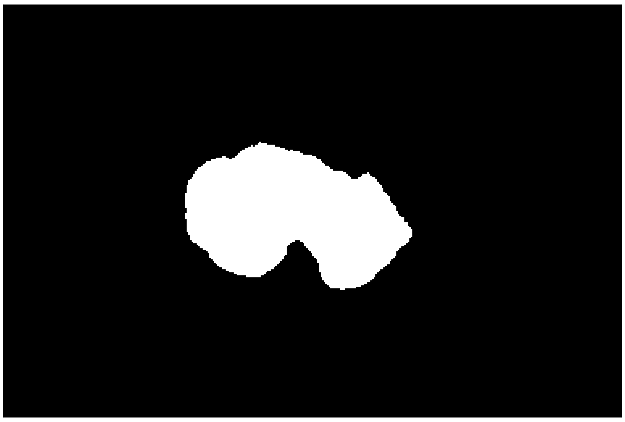
Authentic image



Tampered image



Ground truth mask



4.1.2 CASIA v2.0

This is a dataset of tampered digital images, which was developed by the Chinese Academy of Sciences' Institute of Automation (CASIA). The dataset contains a total of 7,492 authentic images and 5,124 tampered images, which were created using various image processing techniques such as copy-move, splicing, and retouching.

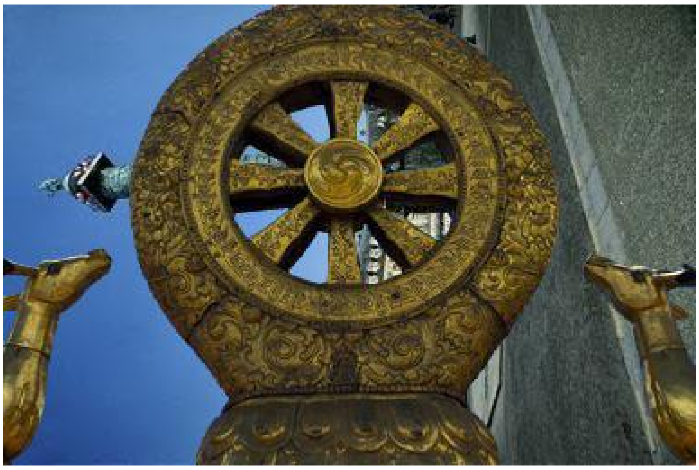
The images in the CASIA v2.0 dataset were collected from various sources, including the Internet and digital cameras, and they cover a wide range of scenes, objects, and lighting conditions. Each image in the dataset is labelled with a ground truth mask, which indicates the location of the tampered area.

The CASIA v2.0 dataset is widely used for evaluating and benchmarking the performance of image forgery detection algorithms. It has become a standard dataset in the field of digital image forensics and is used by researchers and practitioners to test the effectiveness of their algorithms.

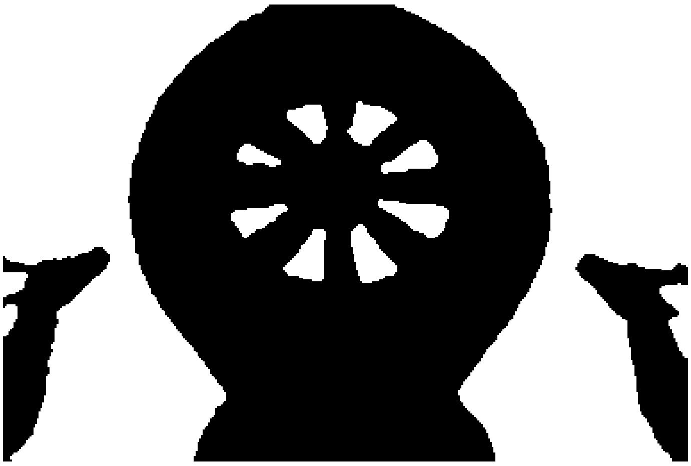
Figure 10: Sample images from CASIA v2.0 dataset

Authentic image

Tampered image



Ground truth mask



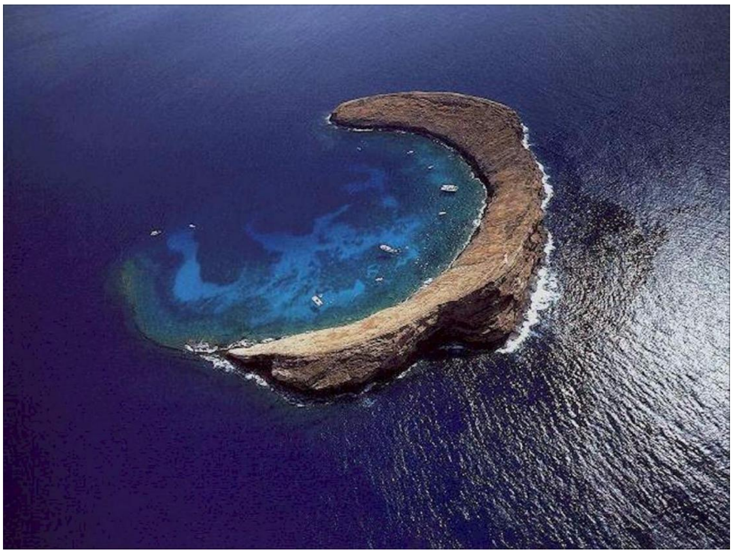
4.2 Wild Web Dataset

One of the main advantages of this dataset is that it covers a wide range of tampering techniques and scenarios, including copy-move, splicing, and removal, as well as different levels of manipulation complexity and image quality. Moreover, the availability of the original sources used to create the forgeries enables researchers to perform a more detailed and insightful analysis of the tampering process and its impact on image properties and characteristics.

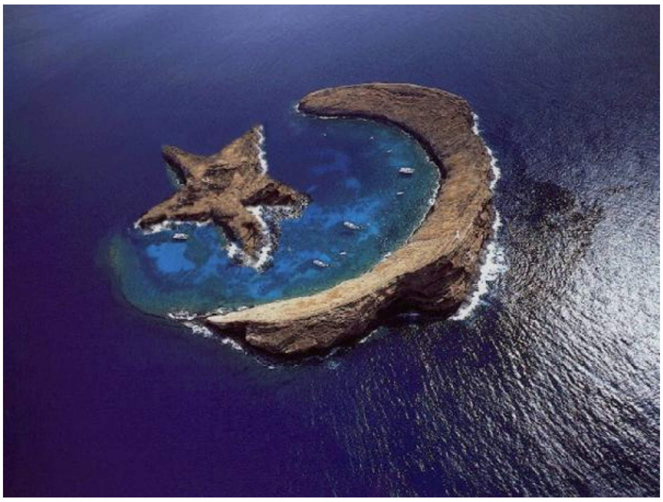
By using the Wild Web tampered image dataset, researchers can develop more robust and accurate image tampering detection and localization algorithms, which are essential for various applications, such as forensic investigation, journalism, and social media content moderation. Additionally, the dataset can also be used to raise awareness about the prevalence and potential harms of image manipulation and to promote the adoption of ethical and transparent practices in the digital media industry.

Figure 11: Sample images from Wild Web dataset

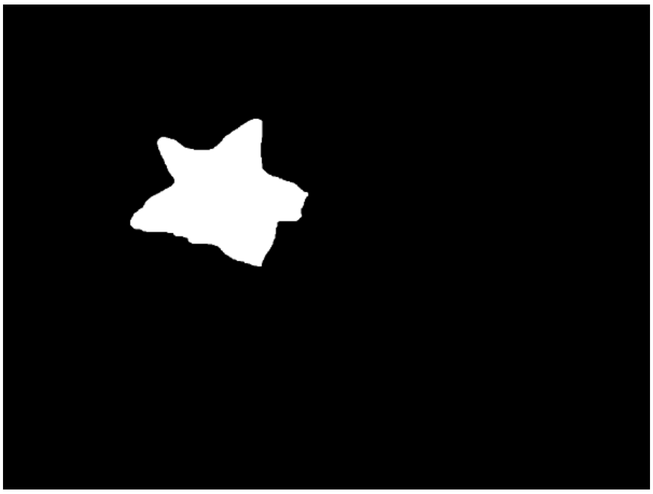
Authentic image



Tampered image



Ground truth mask



CHAPTER 5

PROJECT REQUIREMENT SPECIFICATION

Requirements of the project:

The primary requirements of this project are as follows:

- **Tampering Detection:** The system should be able to identify if an image has been tampered or not.

- **Localization of Tampered Regions**: The system should be able to accurately locate and highlight the tampered regions in the input image.

5.1 Project Scope

Input image: RGB
Category of tampering technique it detects: Copy-move, image splicing

Objectives

1. Develop a robust and accurate image tampering detection system that can identify various types of tampering.

2. Localize the tampered regions in the input image and generate a report indicating the extent and location of the tampering.

3. Verify the authenticity and integrity of the input image and report any discrepancies.

Goals

1. Provide a reliable and efficient solution for image tampering detection that can be used in various applications.

2. Enhance the credibility and trustworthiness of digital images and prevent the spread of misinformation.

3. Contribute to the development of advanced algorithms and techniques for digital image analysis and tampering detection.

The coverage of the system includes the detection and localization of a tampered image.

Limitations

1. Dependence on Image Quality: The effectiveness of image tampering detection systems may be limited by the quality of the input image. For example, low-quality images or images with high compression may be more difficult to analyze and may result in false positives or false negatives.

2. Dependence on Training Data: Image tampering detection systems rely on training data to identify patterns and anomalies in digital images. The effectiveness of these systems may be limited by the quality and quantity of the training data used.

3. False Positives and False Negatives: Image tampering detection systems may generate false positives or false negatives, which can lead to incorrect detection or failure to detect tampering. This can be due to limitations in the algorithms or techniques used, or due to the complexity of the tampering techniques employed.

5.2 Product Perspective

The context of the image tampering detection product is the increasing prevalence of image manipulation and tampering in today's digital world. With the widespread availability of powerful image editing tools, it has become easier than ever to alter images, leading to concerns about the credibility and authenticity of visual media.

The origin of the product can be traced to the need for a reliable and effective solution for detecting and preventing image tampering.

5.2.1 Product Features

The image tampering detection product contains several major features and functions that allow users to detect and locate instances of image manipulation or tampering. These features can be organized as follows:

1. **User Management:** The system includes features for user management, such as user registration and login.

2. **Image Upload:** The system allows users to upload digital images for analysis. This feature may include drag-and-drop functionality or a file browser.

3. **Tampering Detection:** The system uses advanced algorithms and techniques to analyze the uploaded images and identify any inconsistencies or anomalies that indicate tampering.

4. **Tampering Localization:** The system can localize the tampered regions in the input image and generate a report indicating the extent and location of the tampering. This feature allows users to identify the specific areas of the image that have been tampered with.

5. **Authenticity Verification:** The system can verify the authenticity and integrity of the input image and report any discrepancies. This feature can help users to ensure that the images they are analyzing are genuine and have not been altered or manipulated.

6. **Reporting:** The system can generate reports indicating the extent and location of any tampering detected, as well as the authenticity and integrity of the input image. This feature can help users to analyze and share the results of their analysis.

Overall, these features provide users with a reliable and efficient solution for detecting image tampering. The system allows for easy upload and analysis of digital images, detection and localization of tampered regions, and verification of image authenticity and integrity. The user management and reporting make the system a valuable tool for various applications, including journalism, forensics, and digital content creation.

5.2.2 Assumptions

1. **Quality:** It is assumed that the quality of the original image is good enough to allow for meaningful analysis and comparison.

2. **Format:** It is assumed that the image is in a standard format that is recognized by the forensic tools being used for analysis.

3. **Detection Techniques:** It is assumed that the detection techniques being used are effective and able to identify the specific type of tampering that has been performed.

5.3 Functional Requirements

The functional requirements of an image tampering detection system can be described as follows:

1. **Input Validation:** The system should validate the input image for format, size, and quality. The system should also be able to handle images of different sizes and resolutions.

2. **Image Analysis:** The system should analyze the input image to detect any signs of tampering. The analysis should include identifying inconsistencies in pixel values, detecting traces of image editing, and identifying inconsistencies in metadata.

3. **Tampering Detection:** The system should determine whether the input image is tampered with or not based on the analysis results. If the input image is tampered with, the system should provide details on the type of tampering and the specific areas of the image that are affected.

4. **Error Handling:** The system should handle errors gracefully and provide appropriate error messages to the user. For example, if the input image is of poor quality or is not in a compatible format, the system should notify the user and provide guidance on how to proceed.

5. **Output Generation:** The system should generate an output that provides details on the tampering detection results. The output should include information on the type of tampering, the specific areas of the image that are affected, and any other relevant information.

6. **User Interface:** The system should have a user-friendly interface that allows users to upload images, view analysis results, and download the output.

7. **Performance:** The system should be able to process input images quickly and efficiently. The system should also be scalable and able to handle a large volume of images if needed.

These functional requirements ensure that the image tampering detection system can process input images accurately, efficiently, and securely, providing users with reliable and actionable results.

5.4 External Interface Requirements

5.4.1 User Interfaces

The interface connecting the image tampering detection system and its users should have the following logical characteristics:

1. **User-Friendly:** The interface should be designed to be easy to use and understand, even for users who are not familiar with image tampering detection. The interface should be intuitive and visually appealing to encourage users to interact with the system.

2. **Responsive:** The interface should provide feedback to the user in real-time, such as showing the progress of the tampering detection process or highlighting the areas of the image that are suspicious.

3. **Secure:** The interface should be designed with security in mind to protect the user's privacy and prevent unauthorized access to the system. This can include features such as login and registration.

4. **Informative:** The interface should provide users with clear and concise information about the tampering detection process, including how the system works, what the results mean, and what actions the user can take based on the results.

5. **Feedback-Oriented:** The interface should provide users with feedback about their actions and the system's response. This can include progress bars, error messages, and confirmation messages.

5.4.2 Hardware Requirements

1. **CPU:** A multi-core CPU with a clock speed of at least 2 GHz is recommended for running CPU-intensive tasks such as model training and image processing.

2. **GPU:** A high-end GPU such as an Nvidia GeForce or AMD Radeon GPU is recommended for accelerating the training of the CNN model.

3. **RAM:** A minimum of 8 GB of RAM is recommended, but the amount of RAM required will depend on the size of the dataset and the complexity of your CNN model.

4. **Storage:** You will need enough storage space to store your dataset, trained models, and web-based interface.

5.4.3 Software Requirements

1. **Operating System**

   Windows 7 or above

2. **Programming Language**

   Python

3. **Web Development**

   Html, css, Javascript etc.

4. **Platforms**
   - Visual Studio Code
   - Google Colab
   - Jupyter Notebook

5. **Libraries**
   - OpenCV
   - Tensorflow
   - Keras
   - Pandas
   - Numpy
   - Sklearn

5.5 Non-Functional Requirements

5.5.1 Performance Requirement

The performance requirements for an image tampering detection system can be described as follows:

1. **Speed:** The system should be able to process images quickly and efficiently, providing tampering detection results in a timely manner. The response time should be within a reasonable time frame depending on the size and complexity of the input image.

2. **Accuracy:** The system should be able to accurately detect tampering in the input images. The detection results should be reliable and consistent, regardless of the image's quality or format.

3. **Scalability:** The system should be able to handle a large volume of input images without compromising its performance or accuracy. The system should be designed to scale up or down as needed, depending on the workload.

4. **Reliability:** The system should be reliable and consistent in its performance. The system should be able to handle errors and exceptions gracefully, without crashing or causing data loss.

5. **Robustness:** The system should be robust and resilient, able to handle unexpected or abnormal input data without compromising its performance or accuracy. The system should be able to recover from errors or failures without data loss or corruption.

6. **Availability:** The system should be available and accessible to users whenever they need it. The system should be designed to minimize downtime and ensure high availability, even during maintenance or updates.

7. **Usability:** The system should be user-friendly and easy to use. The system should provide clear and concise instructions to users and guide them through the process of uploading images and interpreting the results.

5.5.2 Safety Requirements

Backup and recovery mechanisms: The system should have backup and recovery mechanisms in place to ensure that data is not lost in the event of system failure or other disasters.

5.5.3 Security Requirements

1. **Authentication:** The system should require user authentication to prevent unauthorized access to the system and its data.

2. **Access controls:** The system should have access controls in place to ensure that users only have access to the data and features that they are authorized to use.

3. **Data privacy:** The system should be designed with data privacy in mind to ensure that user data is protected and not shared with unauthorized parties.

CHAPTER 6

HIGH-LEVEL DESIGN

6.1 Current System

The current systems just classify the images in a dataset as either original or tampered and localize the tampered area.

So we are developing an interactive software where the user can input the image themselves. Then the image we will identify if the image is tampered or not with a confidence score and then localize the tampered region.

6.2 Design Details

6.2.1 Novelty

We are creating an interactive web interface that allows users to upload images and visualize the results of the tampering detection and localization algorithm. This will make it easier for users to use the system and interpret the results, even if they have limited technical knowledge.

6.2.2 Innovativeness

Converting RGB to YCbCr and using those images for feature extraction and training using CNN along with NMS.

6.2.3 Interoperability

The system can be integrated with various applications and platforms, making it interoperable with different software systems.

6.2.4 Performance

The use of deep learning models and advanced image processing techniques ensures high performance in terms of tampering detection and localization accuracy.

6.2.5 Security

The system can help in ensuring the security of digital images by detecting tampering attempts and identifying tampered areas.

6.2.6 Reliability

The system's reliability depends on the accuracy of the trained model and the robustness of the image-processing techniques used.

6.2.7 Maintainability

The system can be maintained through periodic updates and model retraining to improve the accuracy of tampering detection and localization.

6.2.8 Portability

The system can be deployed on different platforms.

6.2.9 Legacy to Modernization

The system can be used to detect tampering in both legacy and modern digital images.

6.2.10 Reusability

The trained model can be reused for tampering detection and localization in different applications and systems.

6.2.11 Resource utilization

The system requires sufficient computational resources, especially during the training and testing of the deep learning model. However, the use of optimized algorithms can help in reducing resource requirements.

CHAPTER 7

SYSTEM DESIGN

7.1 Design Approach

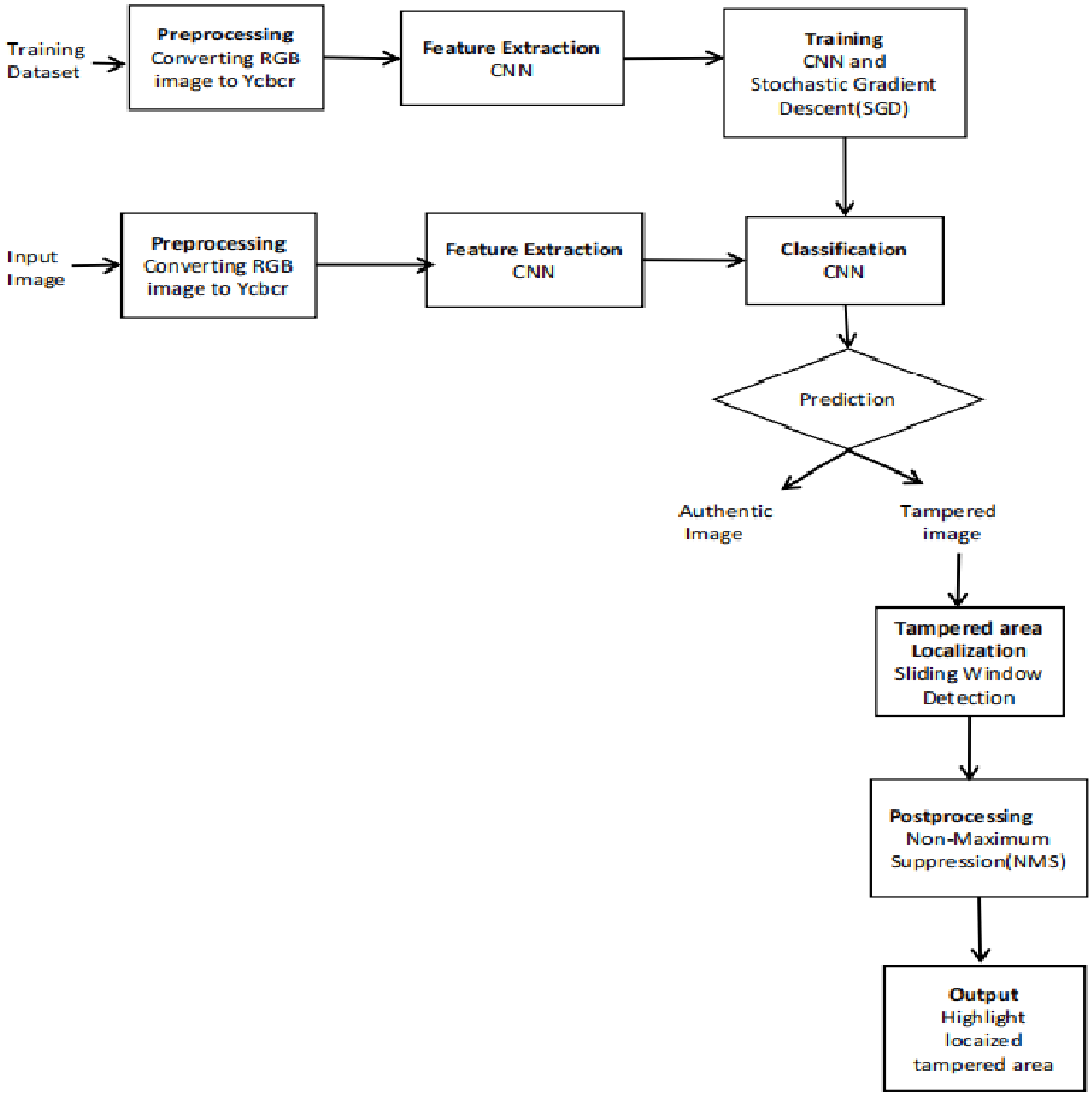

Training Dataset
Preprocessing Converting RGB image to Ycbcr
Feature Extraction CNN
Training CNN and Stochastic Gradient Descent(SGD)
Input Image
Preprocessing Converting RGB image to Ycbcr
Feature Extraction CNN
Classification CNN
Prediction
Authentic Image
Tampered image
Tampered area Localization Sliding Window Detection
Postprocessing Non-Maximum Suppression(NMS)
Output Highlight locaized tampered area

Figure 12: Design Approach

1. Preprocessing

The input RGB image is converted to the YCbCr colour space to separate the luminance (Y) and chrominance (Cb and Cr) components.

2. Feature Extraction using CNN

A convolutional neural network (CNN) is used to learn the features of authentic and tampered regions in the image. The CNN consists of multiple layers of filters that perform convolutions on the input image to extract different features at different scales.

3. Training with the CNN and SGD

The CNN is trained on a dataset of labeled images, including both authentic and tampered images. The training process involves feeding the labeled images into the CNN, which learns to extract features from the images and classify them as authentic or tampered. The training process is typically done using stochastic gradient descent (SGD) to adjust the weights and biases of the network to minimize the loss function and improve its accuracy in detecting tampered images..

4. Tampered Region Localization with Sliding Window Detection

Once the CNN has been trained to recognize tampered images, it can be used to localize the tampered regions within an image. This is done using a technique called sliding window detection, where a small window is moved across the image at different positions and scales, and the CNN is used to classify each window as authentic or tampered. The regions with the highest probability of being tampered are then considered as the tampered regions.

5. Post-processing with NMS

To improve the accuracy of the tampered region localization, post-processing techniques can be applied, such as non-maximum suppression (NMS), which removes overlapping regions and keeps only the regions with the highest probability of being tampered.

6. **Tampered Image Output**

   Finally, the system outputs whether the image is tampered or not and the tampered image with the localized tampered regions highlighted.

7.2 Benefits of the Proposed Design Approach

1. **Accurate tampered region detection:** The use of CNN and sliding window detection helps to accurately detect tampered regions in an image, even in cases where the tampering is subtle or difficult to detect.

2. **Flexibility:** The use of multiple layers of filters in the CNN allows the network to learn and extract features at different scales, making it adaptable to different types of tampering.

3. **Generalization:** The use of a large and diverse dataset for training the CNN helps to ensure that the network can generalize well to new, unseen images.

4. **Speed:** The use of sliding window detection and post-processing techniques such as NMS helps to improve the speed of tampered region detection, making it feasible for real-time applications.

7.3 Drawbacks of the Proposed Design Approach

1. **Complexity:** The use of CNNs and sliding window detection requires a high degree of computational complexity, making it difficult to implement on low-end devices.

2. **Training data requirements:** The accuracy of the tampered region detection depends on the quality and size of the training dataset, which can be time-consuming.

3. **Sensitivity to noise:** The CNN may be sensitive to noise or other artifacts in the input

image, which can lead to false positives or false negatives in tampered region detection.

7.4 Design Constraints

1. The CNN must be trained on a large dataset of labeled images, which requires significant computational resources and time.

2. The accuracy of the tampered region localization may be affected by the size and complexity of the image, as well as the quality and type of tampering.

3. The use of additional input channels (Cb and Cr) may increase the computational requirements and memory usage of the system.

7.5 Assumptions

1. The YCbCr color space provides a more effective representation of the image for tampering detection than the RGB color space.

2. The use of CNN is an effective approach for learning the features of authentic and tampered regions in an image.

3. The sliding window detection technique is an effective approach for localizing tampered regions within an image.

4. Post-processing techniques, such as non-maximum suppression, are effective in improving the accuracy of the tampered region localization.

7.6 Dependencies

1. **Data availability:** The design approach is highly dependent on the availability of a large and diverse dataset of tampered and untampered images. If the dataset is limited or biased, it can impact the accuracy and effectiveness of the tampered image detection system.

2. **Quality of input images**: The performance of the tampered image detection system is highly dependent on the quality of the input images. Poor image quality can affect the accuracy and effectiveness of the system.

3. **Accuracy of YCbCr color space conversion:** The design approach involves converting the input RGB image to YCbCr color space, and using the Y component as input to the CNN. The accuracy and reliability of this conversion process may impact the overall accuracy of the tampered region detection.

CHAPTER 8

PSEUDOCODE AND IMPLEMENTATION

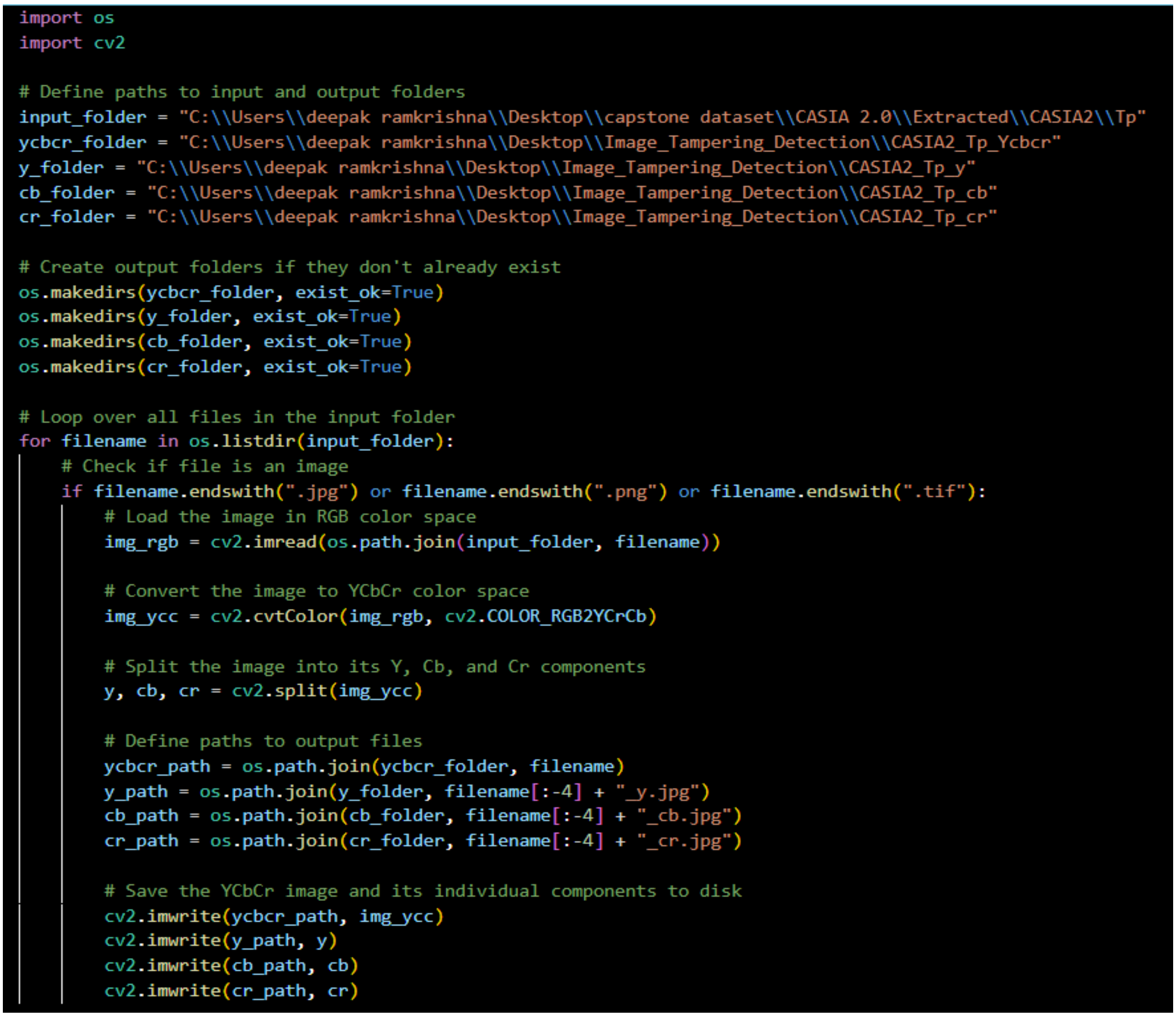
8.1 Pseudocode

```
# Image preprocessing
input_image = read_image()                                # read input image
yCbCr_image = convert_to_YCbCr(input_image)        # convert input image to YCbCr

# Image classification using trained model
cnn_model = load_cnn_model()                              # load the pre-trained CNN model
features = extract_features(yCbCr_image)                 # extract features using CNN
tamper_probability = classify_image(features, cnn_model) # classify image as tampered or not

# Tampered area localization and highlighting
if tamper_probability > threshold

    # find the tampered area using sliding window detection
    tampered_area = localize_tampered_area(input_image)

   # apply non-maximum suppression for post-processing
    tampered_area = apply_non_maximum_suppression(tampered_area)

  # highlight the tampered area and calculate the percentage of tampering
    highlighted_image, tampering_percentage = highlight_tampered_area(input_image, tampered_area)

 # display the highlighted image and  tampering percentage on the website
```

display_results(highlighted_image, tampering_percentage)

8.2 Implementation

Preprocessing: Converting RGB images in the dataset to Ycbcr

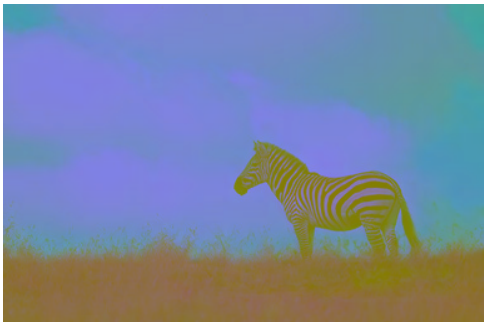


```
import os
import cv2

# Define paths to input and output folders
input_folder = "C:\\Users\\deepak ramkrishna\\Desktop\\capstone dataset\\CASIA 2.0\\Extracted\\CASIA2\\Tp"
ycbcr_folder = "C:\\Users\\deepak ramkrishna\\Desktop\\Image_Tampering_Detection\\CASIA2_Tp_Ycbcr"
y_folder = "C:\\Users\\deepak ramkrishna\\Desktop\\Image_Tampering_Detection\\CASIA2_Tp_y"
cb_folder = "C:\\Users\\deepak ramkrishna\\Desktop\\Image_Tampering_Detection\\CASIA2_Tp_cb"
cr_folder = "C:\\Users\\deepak ramkrishna\\Desktop\\Image_Tampering_Detection\\CASIA2_Tp_cr"

# Create output folders if they don't already exist
os.makedirs(ycbcr_folder, exist_ok=True)
os.makedirs(y_folder, exist_ok=True)
os.makedirs(cb_folder, exist_ok=True)
os.makedirs(cr_folder, exist_ok=True)

# Loop over all files in the input folder
for filename in os.listdir(input_folder):
    # Check if file is an image
    if filename.endswith(".jpg") or filename.endswith(".png") or filename.endswith(".tif"):
        # Load the image in RGB color space
        img_rgb = cv2.imread(os.path.join(input_folder, filename))

        # Convert the image to YCbCr color space
        img_ycc = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2YCrCb)

        # Split the image into its Y, Cb, and Cr components
        y, cb, cr = cv2.split(img_ycc)

        # Define paths to output files
        ycbcr_path = os.path.join(ycbcr_folder, filename)
        y_path = os.path.join(y_folder, filename[:-4] + "_y.jpg")
        cb_path = os.path.join(cb_folder, filename[:-4] + "_cb.jpg")
        cr_path = os.path.join(cr_folder, filename[:-4] + "_cr.jpg")

        # Save the YCbCr image and its individual components to disk
        cv2.imwrite(ycbcr_path, img_ycc)
        cv2.imwrite(y_path, y)
        cv2.imwrite(cb_path, cb)
        cv2.imwrite(cr_path, cr)
```

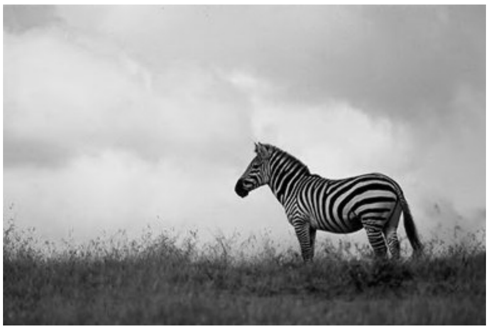
Authentic image before and after converting to Ycbcr

RGB Image

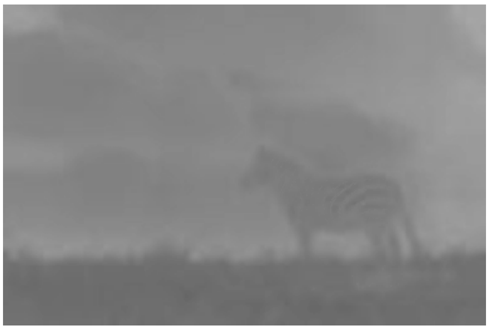
Ycbcr image



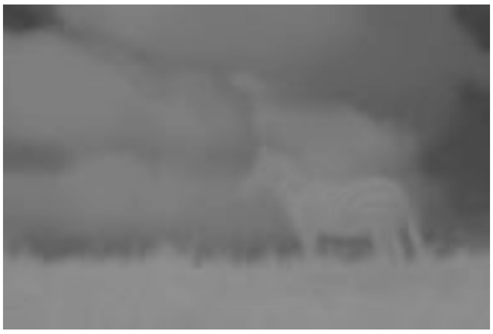
Y Component



Cb Component

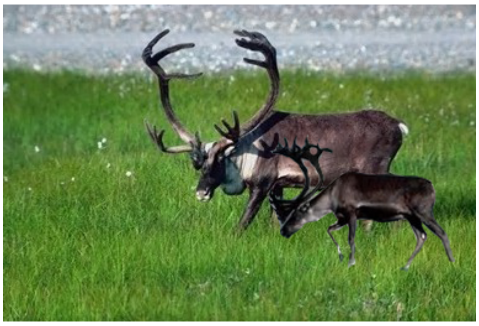


Cr Component

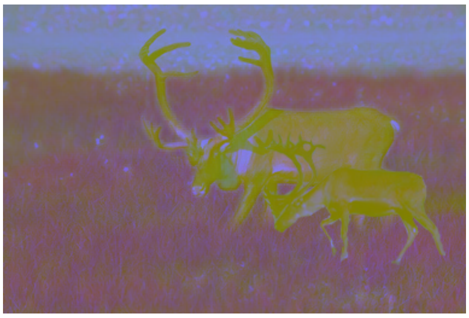


Tampered image before and after converting to Ycbcr

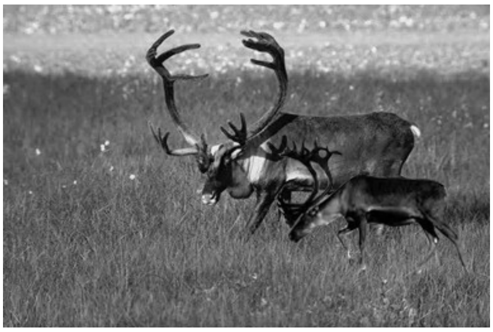
RGB Image



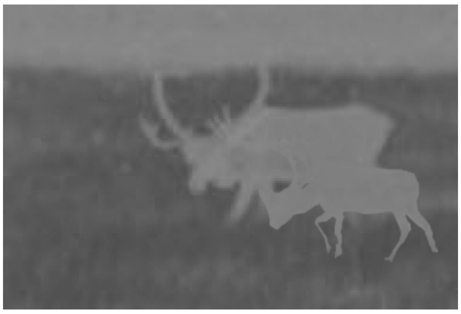
Ycbcr image



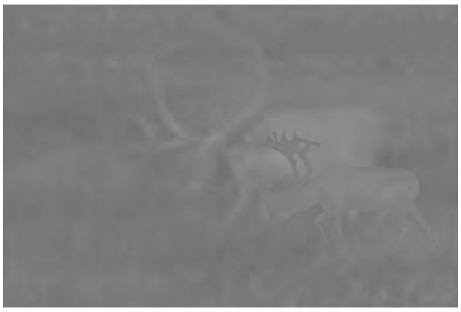
Y Component



Cb Component



Cr Component



CHAPTER 9

CONCLUSION OF CAPSTONE PHASE-1

In our Capstone Phase-1 we completed our problem statement selection and finalization. We identified the abstract, scope and feasibility study of our project. We also did a Literature survey and understood the already existing approaches and methodologies. This gave us some idea of what we should implement and do.

Then we finalized our design approach. We have also put together our Requirement Specification document and High-Level Design document. We have also done the Pre-processing part of our project. We have also completed the Project report for Capstone phase-1

CHAPTER 10

PLAN OF WORK FOR CAPSTONE PHASE-2

We will complete the implementation of our proposed methodology. Perform testing, experimentation and project report for Capstone Phase-2.

We will also start preparing our research paper.

REFERENCES

1. Peng Zhou, Xintong Han, Vlad I. Morariu and Larry S. Davis " Learning Rich Features for Image Manipulation Detection" in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 18-23 June 2018, doi: 10.1109/CVPR.2018.00116
2. Saurabh Agarwal & Satish Chand "Blind Forensics of Images using Higher Order Local Binary Pattern" in Journal of Applied Security Research, 8th March 2018, 13:2, 209-222, DOI: 10.1080/19361610.2017.1422367
3. Sunen Chakraborty, Kingshuk Chatterjee, and Paramita Dey are affiliated with the Government College of Engineering & Ceramic Technology, October 20th, 2022
4. Manjunatha. S Department of Information Science & Engineering.Global Academy of Technology Bengaluru, 560 098, India, Malini M Patil Department of Information Science & Engineering.J S S Academy of Technical Education Bengaluru, 560 060, India "Deep learning-based Technique for
5. Ziyue Xiang College of Engineering & Computer Science, Syracuse University, Daniel E. Acuna* School of Information Studies, Syracuse University "Scientific Image Tampering Detection Based On Noise Inconsistencies: A Method And Datasets" @2018
6. Rajiv V. Dharaskar Former Director Disha–DIMAT Raipur, India " Digital Image Forensics An Affine Transform Robust Copy-Paste Tampering Detection "2020
7. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1S4, June 2019

APPENDIX A: DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Definitions:

- Authentic/Original image: Refers to an image that has not been altered or manipulated.
- Tampered image: Refers to an image that has been altered or manipulated somehow.
- RGB: Red, Green, Blue colour space
- YCbCr: A colour space used for digital images that separate the image into three components: luminance (Y) and two chrominance components (Cb and Cr).
- CNN: CNN stands for Convolutional Neural Network. It is a type of artificial neural network commonly used for image and video analysis, classification, and processing. CNNs use convolutional layers to extract features from input images, which are then fed through fully connected layers for classification
- SGD: Stochastic Gradient Descent, a popular optimization algorithm for training neural networks.
- Sliding window detection: A technique used to classify regions of an image by moving a small window across the image at different positions and scales.
- NMS: Non-Maximum Suppression, a post-processing technique used to remove overlapping regions and keep only the regions with the highest probability of being tampered with.

Acronyms and Abbreviations:

- RGB: Red Green Blue (colour model)
- YCbCr: Luminance Chrominance (colour model)
- CNN: Convolutional Neural Network
- NMS: Mon Maximum Suppression