A DEEP-LEARNING APPROACH FOR DETECTING SPLICING & COPY-MOVE IMAGE FORGERIES AND IMAGE RECOVERY

A PROJECT REPORT

Submitted by

ARAVIND J (2019115017)

KRISHNAN S (2019115047)

PRANAY VARMA (2019115067)

submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

DECEMBER 2022

ANNA UNIVERSITY

CHENNAI - 600 025

BONA FIDE CERTIFICATE

Certified that this project report titled ”A DEEP-LEARNING APPROACH FOR DETECTING SPLICING & COPY-MOVE IMAGE FORGERIES AND IMAGE RECOVERY” is the bona fide work of ARAVIND J (2019115017), KRISHNAN S (2019115047) and PRANAY VARMA (2019115067) who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE: Chennai
DATE: 21.12.2022

DR. K. INDRA GANDHI
Assistant Professor
PROJECT GUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI 600025

COUNTERSIGNED

DR.S.SRIDHAR
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

Digital picture usage has increased at a never-before-seen rate in our day and age, due to the proliferation of gadgets like smartphones and tablets. Furthermore, the development of user-friendly image manipulation software that is available at reasonable prices has made manipulating such content more effortless than ever. Some of these images are tampered with so that it is impossible for the human eye to detect. Moreover, social media platforms have made their distribution to the general public a simple task. It is hence very important to develop automated methods that can detect such forgeries.

In this project, we detect and localize splicing and copy-move image forgeries in images by using two different deep-learning techniques - Convolutional Neural Networks (CNN) and Unsupervised Self-Consistency Learning.

For the CNN based method, the network returns a feature representation which is passed on to an Support Vector Machine (SVM) classifier that predicts if an image is forged or authentic. If forgery is detected, the area of tampering is detected and returned. The Unsupervised Self Consistency Learning scheme uses the Exchangeable Image File Format (EXIF) metadata attributes of an image in order to detect and localize forgeries. An image that is identified as forged then undergoes segmentation to localize the spliced region.

ACKNOWLEDGEMENT

We express our sincere gratitude to our guide DR.K.INDRA GANDHI, Assistant Professor (Sl.Gr), Department of Information Science and Technology , College of Engineering, Guindy, Anna University, Chennai for her invaluable support, guidance and encouragement for the successful completion of this project. Her knowledge, attitude, commitment and spirit have inspired and enlightened us.

We would like to convey our gratitude to DR.S.SRIDHAR, Professor & Head of the Department, Department of Information Science and Technology, College of Engineering, Guindy, Anna University, Chennai for providing us the opportunity and infrastructure to carry out this project. We express our thanks to the panel of reviewers Dr. K. VANI, Professor, Dr. S. BAMA, Assistant Professor, Dr. P. GEETHA, Assistant Professor, Dr. M. DEIVAMANI, Teaching Fellow, Department of Information Science and Technology for their valuable suggestions.

And last, but not the least, we wish to thank our parents and family members for bearing with us throughout the project period and for having given us the opportunity to do this course in such a prestigious institution.

J ARAVIND

S KRISHNAN

PRANAY VARMA

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Networks
EXIF	Exchangeable Image File Format
SVM	Support Vector Machine
SRM	Spatial Rich Model
CMF	Copy Move Forgery
DCT	Discrete Cosine Transform
ELA	Error Level Analysis

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Malicious image manipulation, previously restricted to dictators and spy agencies, is now available to legions of common Internet trolls and Facebook conmen. It is now possible to create realistic image composites, fill in large image regions, generate plausible video from speech, and so on with only basic editing skills. One might have expected that these new methods for creating synthetic visual content would be accompanied by equally powerful techniques for detecting fakes, but this has not been the case thus far. Thus detecting such forgeries becomes very important to stop the spread of false information. In this project, we suggest both supervised and unsupervised methods to detect and localize image tampering.

Three of the most common image manipulation techniques are:

- **SPLICING**: In splicing a region from an authentic image is copied into a different image.
- **IMAGE INPAINTING**: In image inpainting, an image region is removed and the removed part is then filled in to complete the image.
- **COPY-MOVE**: A specific region from the image is copy pasted within the same image

1.2 OBJECTIVE

This project work aims to:

- Detect and localize splicing and copy-move image tamperings.
- Approximately recover the original image that was tampered with.
- Analyse the performance of different deep-learning approaches on different datasets.
- Develop a web application using Streamlit, which allows the users to upload test images and find the region of tampering if any.

1.3 PROBLEM STATEMENT

The goal of this study is to detect and localize Splicing and Copy-Move forgeries in images using both supervised and unsupervised deep learning techniques. To achieve this two deep-learning approaches CNN and unsupervised self-consistency learning have been implemented on various image forensics datasets like CASIA2[7], Dresden[8] and in the In-the-Wild Image Splice Dataset dataset [9] and the performance of image forgery detection for each approach is analysed based on the test sample difficulty.

This project work also aims to approximately recover the original image which is subject to splicing or copy-move tampering using an image tamper resilient generative scheme for image self-recovery.

1.4 SOLUTION OVERVIEW

CNN Approach: Various computer vision and deep-learning approaches have been suggested to detect image forgeries to date. Specifically, a few CNN-based architectures have managed to predict images with an accuracy of close to 98%. However, the tampering done in these images can also be easily recognized by humans. In this project, we are developing a CNN network that attempts to detect forgeries on more difficult samples and analyse its performance on such examples. This is a supervised approach.It is robust in detecting both copy move and splice forgeries.

Unsupervised Self-Consistency Learning: Standard supervised learning approaches, which have been extremely successful for many types of detection problems, are unsuitable for image forensics. This is due to the vast and diverse space of manipulated images, making it unlikely that we will ever have enough manipulated training data for a supervised method to fully succeed.

To overcome this, we are using an unsupervised methodology called self-consistency learning where with the help of EXIF metadata, we can identify if the image has been tampered with. EXIF tags are camera specifications that are digitally engraved into an image file at the time of capture. Thus, given two photographs, we can figure out from their EXIF metadata that there are a number of differences in the two imaging pipelines. This approach is well suited to identifying spliced images as the metadata for patches from different images is very likely to be different. However, copy move forgeries cannot be identified as the imaging pipeline for the portion that has been copied and reproduced would not be different. When compared to CNN, more complex forms of splicing forgeries can be identified.

1.5 ORGANIZATION OF THE REPORT

The organization of report is as follows: Chapter 1 gives the background information, objective and the problem statement of our project titled ”A Deep-Learning Approach for Detecting Splicing & Copy-Move Image Forgeries and Image Recovery”. Chapter 2 deals with the literature survey, the methods that can be implemented for detecting tampered images and image reconstruction. Chapter 3 elaborates on the technical architecture, function and working of each module. Chapter 4 presents us the information about the algorithms used in the project. Chapter 5 discusses about our results obtained and the performance analysis of the project. Chapter 6 gives the conclusion and the future work to be done for the next phase of our project.

CHAPTER 2

LITERATURE SURVEY

This chapter provides a review of the literature on reference papers. Section 2.2 deals with literature survey of image forgery detection using CNN. In section 2.3, related works of an unsupervised method known as Self-Consistency Learning has been discussed. The chapter also contains a survey on the reconstruction of tampered images in the section 2.4.

2.1 INTRODUCTION

The detection of a forged image is driven by the need for authenticity and to maintain the integrity of the image. There are cases when it is difficult to identify the edited region from the original image. Various computer vision and deep-learning approaches have been suggested to detect image forgeries to date. While some of these approaches had very high accuracy, they were tested on less challenging datasets. A literature survey was done to gain better insights into the existing solutions and their performance on different test samples. The limitations and the knowledge gained from the papers will help us to create a better system.

2.2 SUPERVISED LEARNING APPROACHES

The sections 2.2.1, 2.2.2 and 2.2.3 discusses the literature survey for the CNN approach.

2.2.1 AN EFFICIENT CNN MODEL TO DETECT COPY-MOVE IMAGE FORGERY [1]

This paper introduced an accurate deep CMF detection method. The traditional approach uses a block-based algorithm, whereas the CNN approach uses the entire image. The proposed method is divided into three stages: preprocessing, feature extraction, and classification. The input image is resized to enter the next stage without cropping any image parts in the preprocessing data stage. Three convolution layers are followed by a max-pooling layer in the feature extraction stage. At the end of this stage, a full connection layer connects all features to the dense layer. Finally, the classification stage is invoked to categorise the data into two groups (forged or original).

With an appropriate number of convolutional and max-pooling layers, the proposed architecture is computationally lightweight. The approach also offers a quick and accurate testing process that takes 0.83 seconds for each test. Many empirical experiments have been carried out to ensure the proposed model's efficiency in terms of accuracy and time. These tests were carried out on benchmark datasets and achieved very high accuracy.

However, the accuracy of classification in this approach is not so good when the test samples are challenging.

2.2.2 COPY MOVE AND SPLICING IMAGE FORGERY DETECTION USING CNN [2]

This paper presents an approach to detecting copy move and splicing image forgery using a CNN with three different models i.e. ELA (Error Level Analysis), VGG16 and VGG19. Two datasets of varying difficulty, CASIA v2.0 and NC2016 are used. The major components of the proposed methodology are,

pre-processing, error level analysis and CNN.

In the preprocessing stage, the dataset is resized to 128*128 pixels. The ELA stage involves resaving the preprocessed images, resulting in an increase in brightness. The resaved images are compared to the preprocessed ones, with forged images having a greater brightness in their modified components with respect to the original portions. Next, the image is resized with each RGB value normalized between 0 to 1. Finally, CNN-based training occurs with two architectures (VGG16 and VGG19) being utilized.

The experimental results validate that the classification performance decreases when the samples are more challenging. The implemented architecture does not easily generalize to datasets with different underlying distributions.

2.2.3 A DEEP LEARNING APPROACH TO DETECTION OF SPLICING AND COPY-MOVE FORGERIES IN IMAGES [3]

In this paper, a new deep learning-based image forgery detection method that uses a convolutional neural network (CNN) to automatically learn hierarchical representations from input RGB colour images has been presented. The proposed CNN is intended primarily for image splicing and copy-move detection applications. Rather than using a random strategy, the weights in the network's first layer are initialised with the basic high-pass filter set used in the calculation of residual maps in the Spatial Rich Model (SRM), which serves as a regularizer to efficiently suppress the effect of image contents and capture the subtle artifacts introduced by tampering operations. To extract dense features from the test images, the pre-trained CNN is used as a patch descriptor, and a feature fusion technique is used.

The proposed solution outperforms many state-of-the-art models, in terms of speed and accuracy, however, the performance of the model deteriorates for more challenging image forgery datasets.

2.3 UNSUPERVISED LEARNING APPROACHES

The sections 2.3.1 and 2.3.2 discusses the literature surveys related to unsupervised learning approaches for detecting image forgery.

2.3.1 FIGHTING FAKE NEWS: IMAGE SPLICE DETECTION VIA LEARNED SELF-CONSISTENCY [4]

In this paper, a learning algorithm has been proposed for detecting visual image manipulations that have been trained solely on a large dataset of real photographs. The algorithm employs the automatically recorded photo EXIF metadata as a supervisory signal for training a model to determine whether an image is self-consistent, or whether its content could have been produced by a single imaging pipeline. This self-consistency model is applied to the task of detecting and localising image splices. The insight explored in this paper is that patches from a spliced image are typically produced by different imaging pipelines, as indicated by the EXIF meta-data of the two source images.

Despite never seeing any manipulated images during training, the proposed method achieves state-of-the-art performance on several image forensics benchmarks.

However, the model is not well-suited to finding very small splices in images. Also, over and underexposed regions are sometimes flagged by the model to be inconsistent because they lack any meta-data signal.

2.3.2 DETECTING TAMPERED REGIONS IN JPEG IMAGES VIA CNN [5]

This paper's author, N. Takeda T. H.,et. al proposes a method for using CNN to detect the tampered region in a JPEG image. The DCT coefficients are provided as input, and the output is a binary segmented image in which the tampered and non-tampered regions are represented by contrasting white and black regions.

They have created a total of 45 types of CNN models and compared them. The best model had an accuracy of 0.63 in terms of the F-measure, which is based on SVM. In the experiment, the tampered part was found accurate to a great extent, and the F-measure of their method is approximately 2.3 times that of the MDBD method.

2.4 IMAGE RECONSTRUCTION

The section 2.4.1 discusses the literature survey related to image reconstruction of tampered images.

2.4.1 FROM IMAGE TO IMUGE: IMMUNIZED IMAGE GENERATION [6]

S. Li, et.al propose a system to produce tamper-resilient images. The steps involved in the approach involve training a U-Net backboned encoder, a tamper localization network and a decoder for image recovery. The objective is to convert normal images into immunized images and to conduct successful tamper localization and content recovery on them at the recipient's side. Imuge is designed to be robust against common attacks such as lossy compression,

image interpolation or cropping. One of the concerns of the authors S. Li, et.al is for the differences between the encoded or 'immunized' images and the original images to be imperceptible.

The encoded images are subjected to 5 types of attacks and 2 kinds of malicious tampering. The tampered image is then passed through a verifying layer, which predicts the tamper mask of the attacked image and generates the rectified image. Finally, the decoder generates the recovered image given the rectified image. The system performs well in detecting different kinds of tampering and reconstruction is satisfactory, both in terms of the quality of the image and accuracy with respect to the original version.

2.5 SUMMARY

Hence there are many methodologies we studied from the research papers. For the first phase of our project we will be implementing the image forgery detection by CNN and Unsupervised self consistency. In the second phase we will be reconstructing the tampered image to original image.

CHAPTER 3

SYSTEM DESIGN

This chapter presents us with information about the overall system design and module description in a detailed manner. The overall architecture of the individual approaches has been discussed in sections 3.2.2 and 3.2.3. The detailed module wise description can be found in sections 3.3 and 3.4.

3.1 INTRODUCTION

The approaches that have been implemented to detect copy-move and splicing tampering in images are CNN, a supervised learning approach and Self-Consistency Learning, an unsupervised learning method. Two different datasets of varying difficulty have been used to test both these approaches. The first dataset used is the 'CASIA 2 Image Forensics' dataset, which has 12,622 images, where the ratio of authentic to tampered images is 60:40. The tampering in this dataset is less challenging and can be recognized by humans.

The second dataset used is the 'Labels in the Wild' dataset which contains 201 tampered images and the masks of each tampered image. This dataset is relatively much more challenging than the CASIA 2 dataset and the tampering cannot be easily recognized by humans. Both approaches' classification accuracy is evaluated using these two datasets.

3.2 TECHNICAL ARCHITECTURE

This section discusses the architecture of the project in a two-level approach. The high level design in 3.2.1 provides insight into the overall flow, while the CNN and Unsupervised Self Consistency Learning approaches have been elaborated in 3.2.2 and 3.2.3 respectively .

3.2.1 HIGH LEVEL SYSTEM DESIGN

The proposed system's high-level architecture is depicted in Fig 3.1. The user interface has been developed with Streamlit, which is used to accept a test image from the user as well as the deep learning model of choice. The sequence of actions will be followed depending on the model selected, as discussed further in sections 3.2.2 and 3.2.3 for CNN, a supervised approach and Unsupervised Self Consistency respectively.

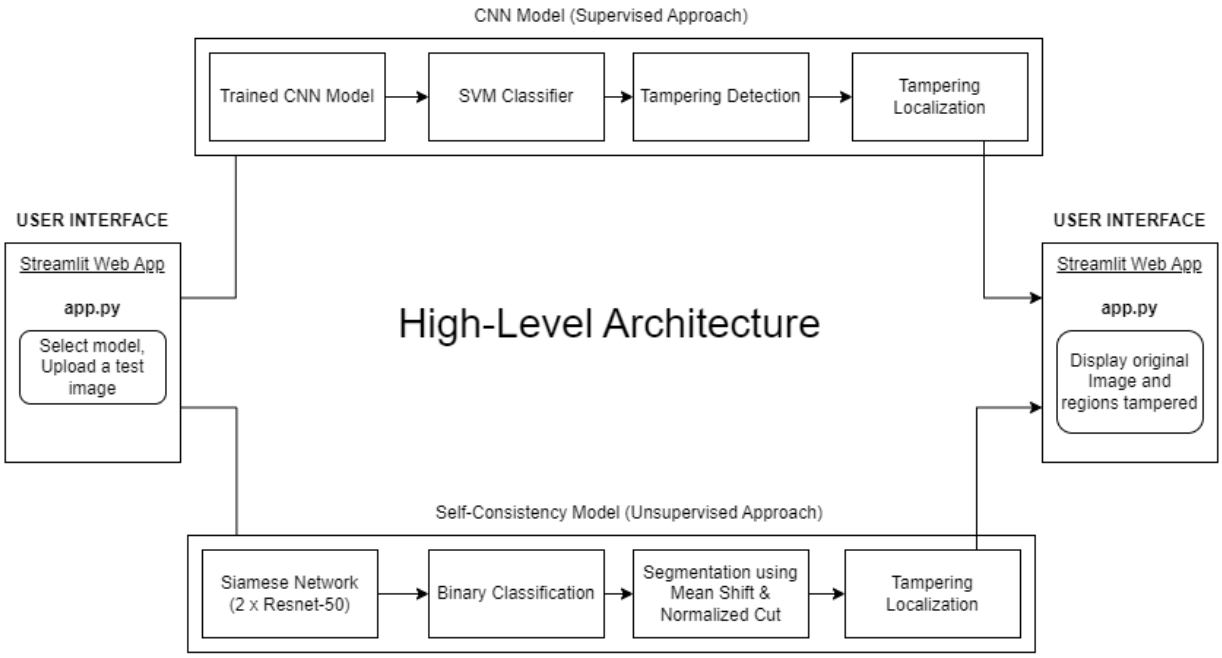

Figure 3.1: High-Level Architecture

Once the model has been executed, the web application will display the original image uploaded by the user as well as the parts of the image that have been tampered with (tampering localization).

3.2.2 CNN APPROACH

The overall architecture of the CNN approach is depicted in Figure 3.2. The images in the CASIA dataset's authentic and tampered classes are first preprocessed using techniques such as augmentation. Patches of 128 x 128 x 3 pixels are extracted from each image in both image classes.

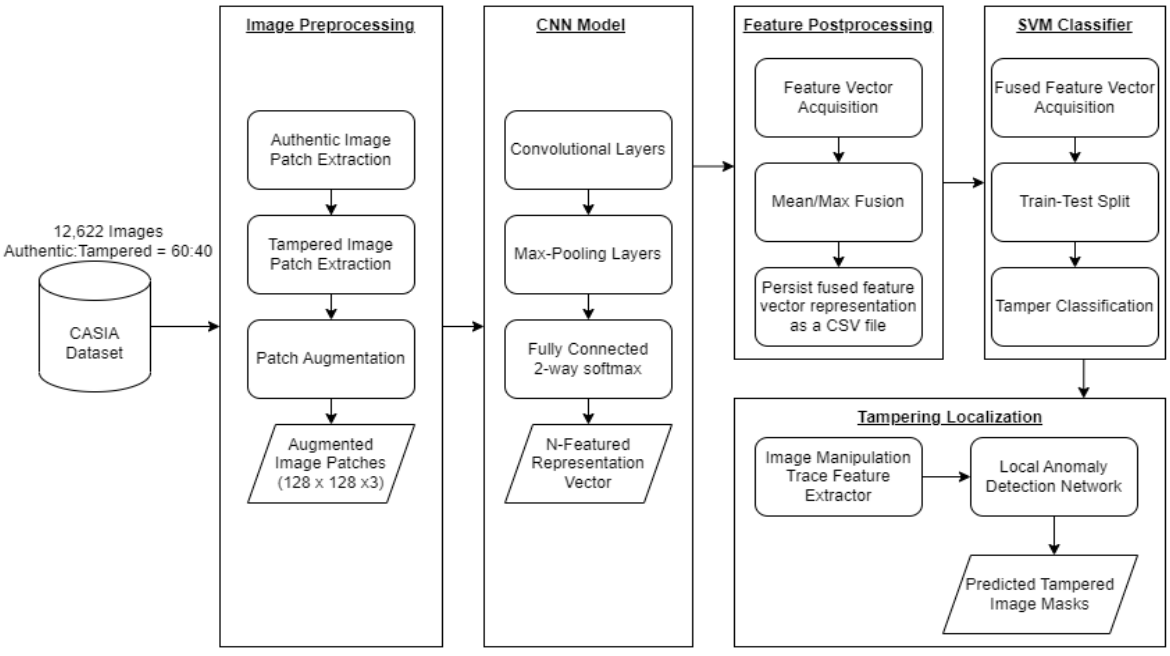

Figure 3.2: CNN Approach Architecture

These patches are then provided as input to the CNN model, producing an N-featured representation vector. Mean/Max fusion is then used to fuse this vector to a single feature vector. This fused vector is then fed into the SVM classifier, which determines whether or not the given image has been tampered with. An image manipulation trace feature extractor and a local anomaly detection network have been used to determine the region of tampering, which is further discussed in section 3.3.4.

3.2.3 UNSUPERVISED SELF-CONSISTENCY LEARNING

Architecture for training the Siamese Network in the unsupervised self-consistency approach is depicted in Figure 3.3.

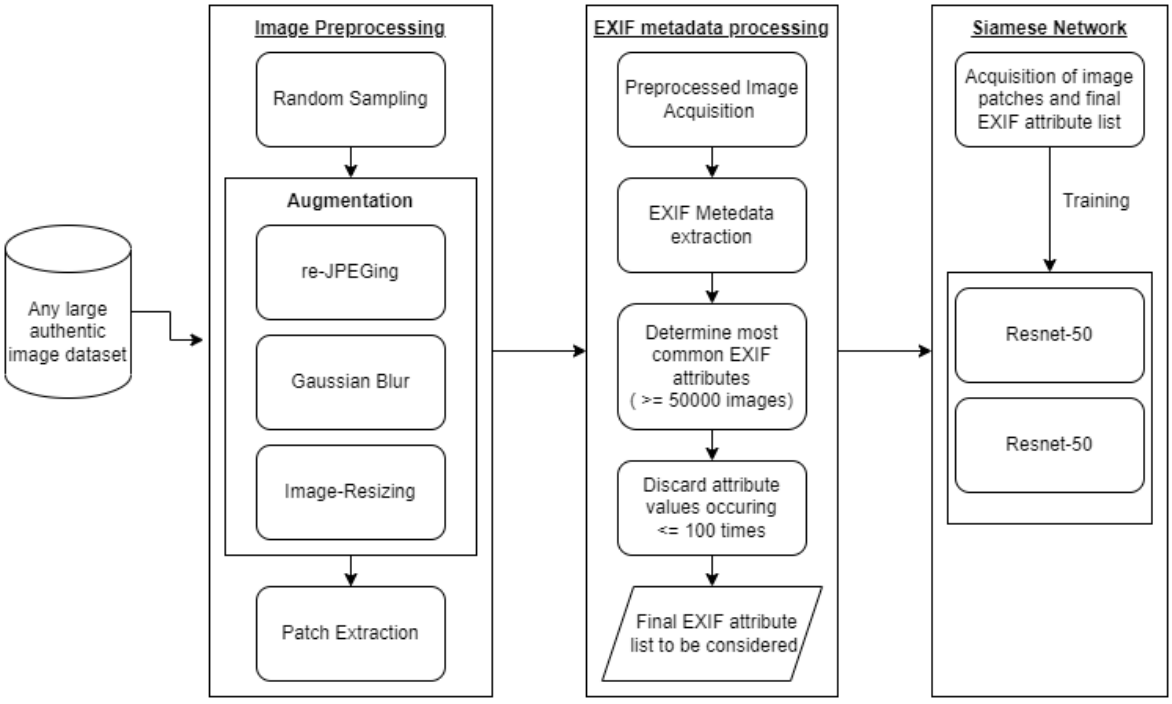

Figure 3.3: Siamese Network training architecture

For training the network, any large authentic image dataset can be used. In this project, the Flickr dataset has been used which has more than 400000 authentic images. First, the input dataset is preprocessed by applying random sampling and image augmentation techniques to get a subset of well-distributed, augmented images. The EXIF metadata is extracted from these images, and the most commonly occurring EXIF attributes are determined. The authentic image patches and the final EXIF attribute list are used for training the Resnet-50 model. The model determines the percentage of consistency between 2 image patches based on the EXIF values in those patches. The architecture for determining the tampered region in an image using self consistency learning is depicted in Figure 3.4.

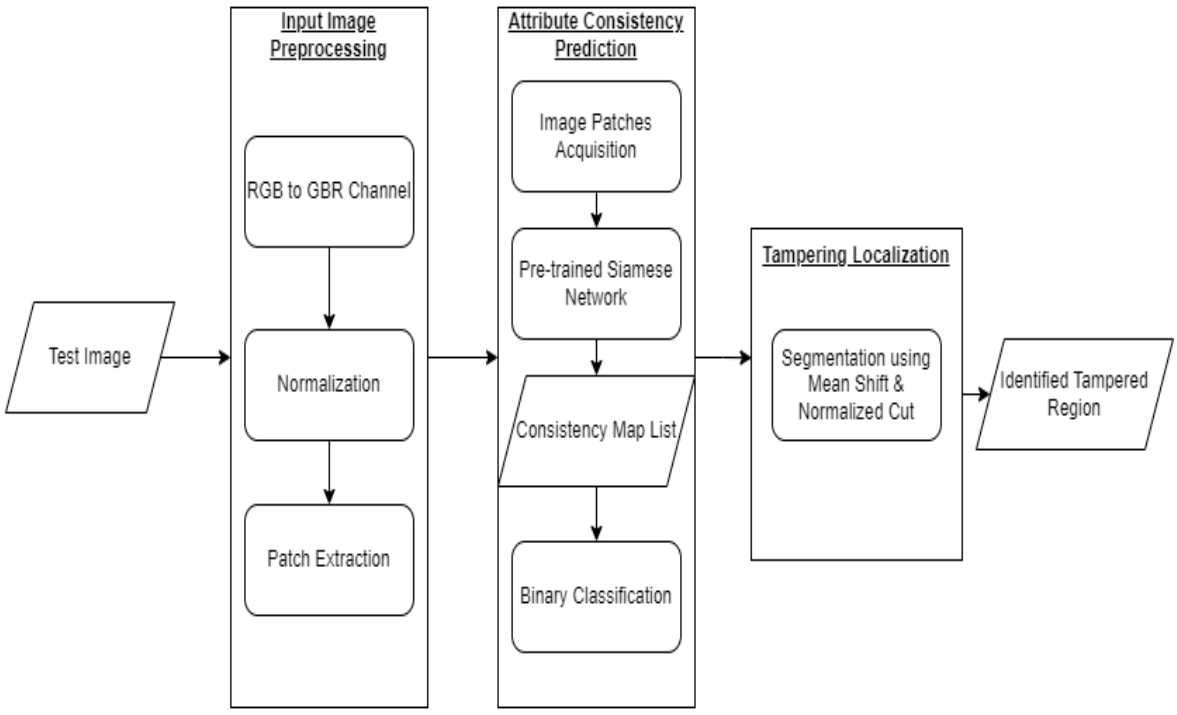

Figure 3.4: Self-Consistency Learning Localization Architecture

The input image is first preprocessed and patches are extracted as mentioned in section 3.4.1. These extracted image patches as passed as input to the pre-trained Siamese Network, which returns a consistency map list after performing a pairwise consistency check of all the patches of the image. The relative consistency percentage values to the first patch are contained in each element of the consistency map list. Finally, using the obtained consistency map list, segmentation methods such as Mean Shift and Normalized Cut are applied to the input image to determine the exact region of tampering.

3.3 CNN MODULES

The four sub-modules for CNN have been discussed in this section. The Patch Extractor produces patches that are passed to the CNN model, whose architecture and training process has been discussed in 3.3.2. CNN produces feature vectors that serve as inputs to the SVM Classifier in 3.3.3. An image that is identified as tampered undergoes Tampered Region Localization as described in 3.3.4.

3.3.1 PATCH EXTRACTOR

The flow of the patch extraction module is depicted in figure 3.5. For each image in the authentic and tampered class in the data set, a patch having a window size 128 x 128 x 3 (Width x Height x Number of Channels) is applied and the window is slid based on the stride value. Here, we are using a stride of 128 and 2 patches are extracted per image so that training of the CNN model becomes less computationally expensive.

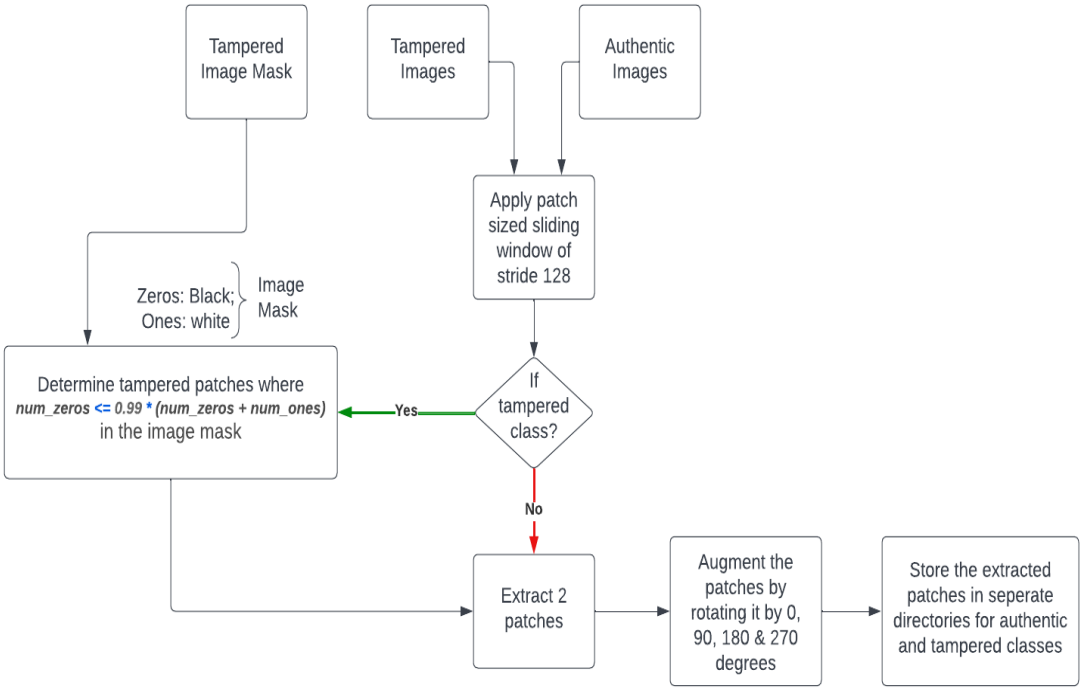

Figure 3.5: Patch Extraction

To identify the exact patch which is tampered with in the images of the tampered class, the masks of the image are analysed and the region where the number of zeros (the black portion in the image mask) is less than or equal to 99% of the total number of zeros and ones combined is flagged as a tampered patch.

Then the extracted patches are augmented by rotating them by 0, 90, 180 and 270 degrees, thereby resulting in 8 patches per image. The module programmatically creates directories for storing the extracted patches of the tampered and the authentic class.

3.3.2 CNN MODEL AND TRAINING

The model consists of 9 convolution and 2 pooling layers. 128 x 128 x 3 patches are passed through the network for the purpose of feature extraction, with the resultant feature vector serving as an input for the SVM classifier. The first convolution layer consists of 3 5x5 kernels with a stride of 1, while the first pooling layer has 30 2x2 filters of stride 2. The second convolution layer has 30 5x5 kernels of stride 2. The next 7 convolution layers all have 16 3x3 filters of stride 1, while the other pooling layer has 16 2x2 filters of stride 2. The CNN layers are depicted in the figure 3.6.

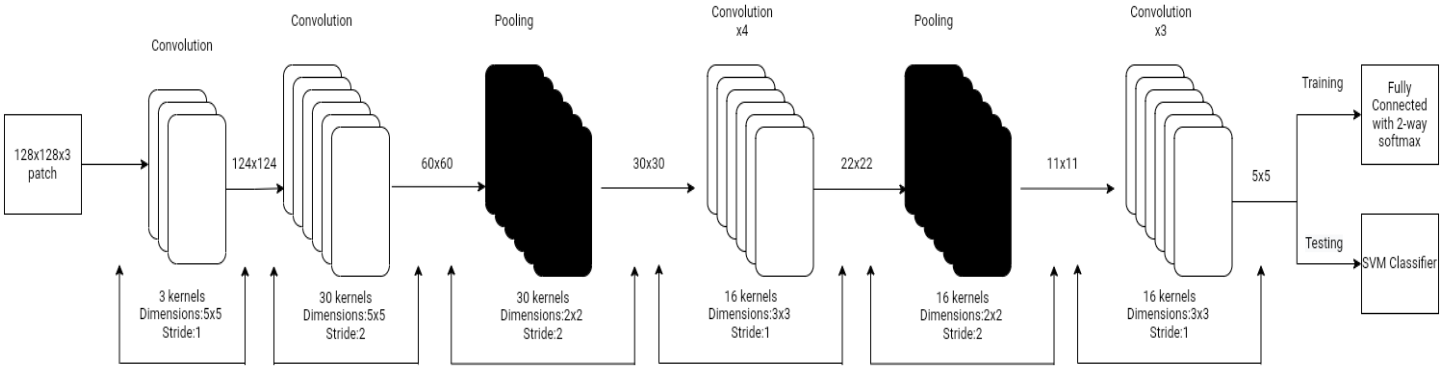

Figure 3.6: CNN Model and Training

The convolution layers extract features from the input matrices, while the pooling layers perform down-sampling or dimensionality reduction of the features. The ReLu activation function is used by each of the convolution layers. Local response normalization is applied to every feature map before the pooling operation to improve generalization. As far as training is concerned, two random tampered patches are selected per image as training a huge amount of extracted patches would be computationally expensive. The model is trained for 250 epochs.

3.3.3 FEATURE EXTRACTOR AND SVM CLASSIFIER

The flow of the feature extraction and svm classification module is depicted in Figure 3.7. The 128x128x3 patches are fed into the CNN model, which extracts a 400-D (5x5x16) feature representation of the patches. These features are then passed to a fully-connected layer with a 2-way softmax classifier in the training phase and the SVM model in the testing phase.

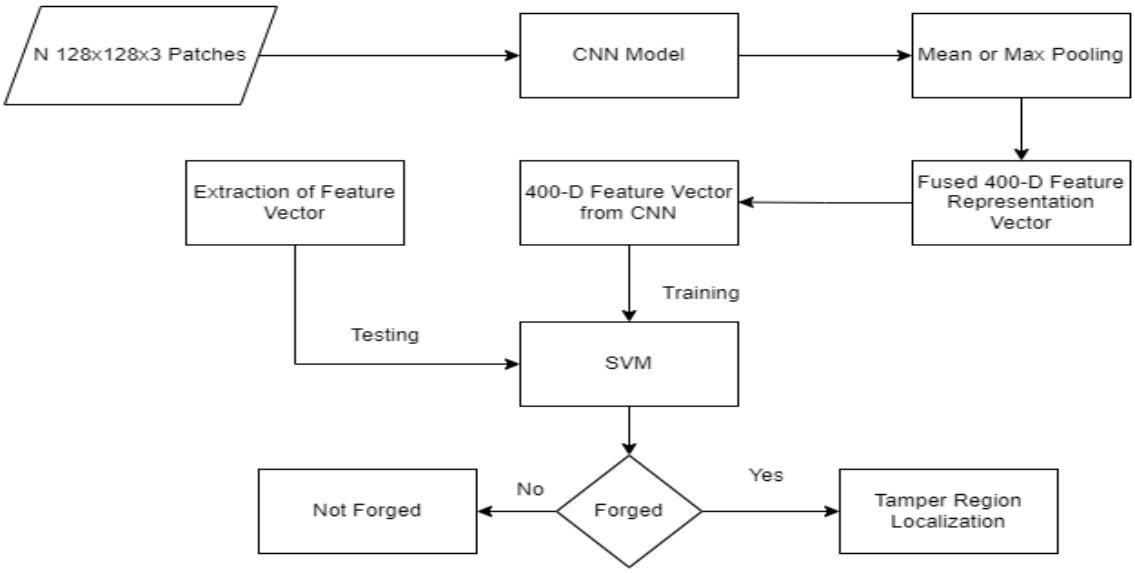

Figure 3.7: Feature extraction and SVM classifier

Before being fed into the SVM classifier, the (n x 400-D) feature representations for an image must be fused into a single feature vector. Mean or max pooling is applied on each dimension of the representation over all the n patches to obtain the resultant fused feature vector for an image, which has 400 features. This 400-D feature vector is used to train the SVM classifier. After training the SVM classifier, an input image can be given to the SVM model to predict whether the image has been tampered with or not.

3.3.4 TAMPERED REGION LOCALIZATION

MantraNet is a machine learning-based image forgery detection method that uses deep learning techniques, specifically a CNN, to analyze images and identify whether they have been manipulated or altered in any way. The architecture of ManTraNet is depicted in figure 3.8.

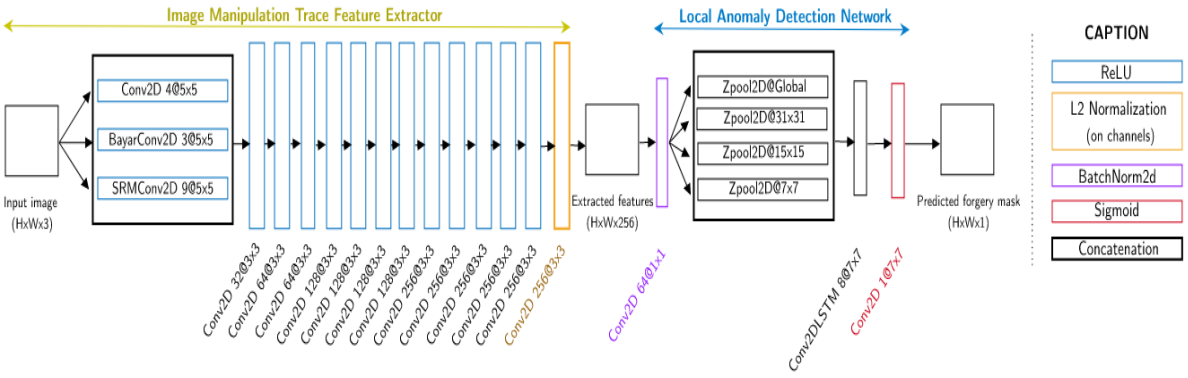

Figure 3.8: ManTraNet Architecture

A testing image is used as the input, and a pre-trained ”ManTraNet” model is used to predict a pixel-level forgery likelihood map as the output. It is made up of two smaller networks:

- The Image Manipulation Trace Feature Extractor is a feature extraction network for the purpose of classifying images that have been altered, and it encodes the altered image in a patch into a feature vector with a fixed dimension.
- The Local Anomaly Detection Network is a network that was created with the understanding that in order to effectively detect various

types of forgeries, we must evaluate our extracted characteristics more and more locally.

3.4 SELF CONSISTENCY LEARNING MODULES

This section covers the sub-modules involved in the Unsupervised Self Consistency Learning pipeline. 3.4.1 covers the preprocessing and consistency map extraction process. The Siamese network that identifies tampering has been discussed in 3.4.2, while the segmentation of tampered regions has been dealt with in 3.4.3.

3.4.1 INPUT IMAGE PREPROCESSING AND CONSISTENCY MAP EXTRACTION

The complete flow of the input preprocessing and consistency map extraction module is depicted in figure 3.9. The image is loaded from the specified input directory and the RGB channels are converted to GBR format as OpenCV reads the images in GBR format. The image is then normalised to restrict the pixel values between 0 and 1.

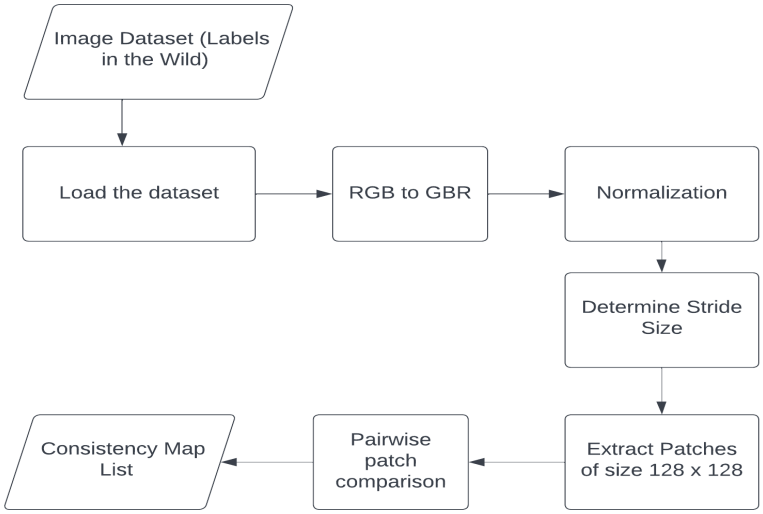

Figure 3.9: Input Preprocessing and Consistency Map Extraction

For the ease of processing these images by the pre-trained Siamese Network, the dimensions of the image are 'unsqueezed' from (w, h, 3) to (1, 3, w, h) so that it can easily pass through the model. The stride size is dynamically calculated based on the image dimensions. A patch-sized sliding window having the determined stride size is applied over the image to get patches of size 128 x 128. Then, all the consistency maps are generated by comparing each patch in the first patch list with each patch in the second patch list. These maps show relative values to the first patch.

3.4.2 SIAMESE NETWORK

The architecture of the Siamese Network is depicted in Figure 3.10. The ResNet 50 is the classical neural network used here. It is a predefined model available in pytorch which can be trained on the input dataset to predict the results.

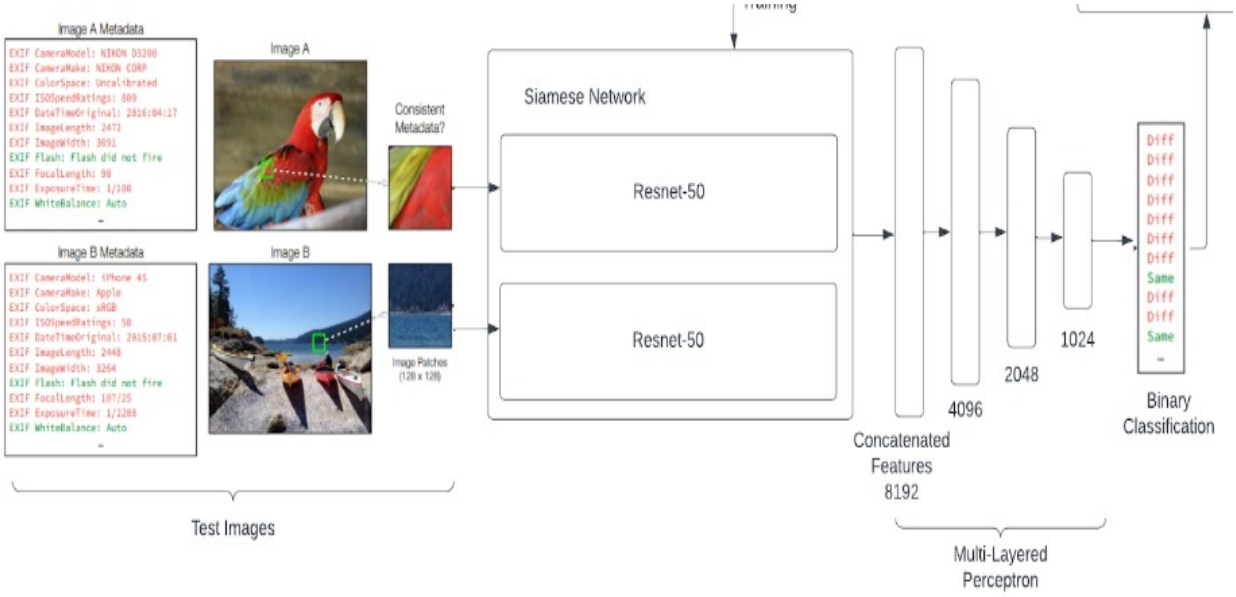

Figure 3.10: Siamese network architecture

The Siamese network is used to predict the probability that a pair of Each EXIF field in 128 128 image patches has the same value. attribute. It uses shared ResNet50 sub-networks, which have been pretrained. Each of the sub-networks produces a 4096-dimension feature vector. These vectors are concatenated and passed through a multilayer perceptron (4 layers) with 4096, 2048, and 1024 units, followed by the final output layer. The network predicts the probability that the images share the same value for each of the n metadata attributes.

3.4.3 IMAGE SEGMENTATION USING MEAN SHIFT AND NORMALIZED CUT

This module deals with the segmentation of an image into two parts (original and spliced). After the consistency maps for two patches in an image for all EXIF attributes are returned, the points in the resultant map are plotted and the mean shift is calculated. For the mean shift, points within the 10th percentile for distance from an individual point are considered. Similarly, the resultant map serves as an input for the normalized cut computation that makes use of the sklearn spectral clustering function to return the fit. If most of the image is of high probability, it is flipped. A sample output is shown in figure 3.11.

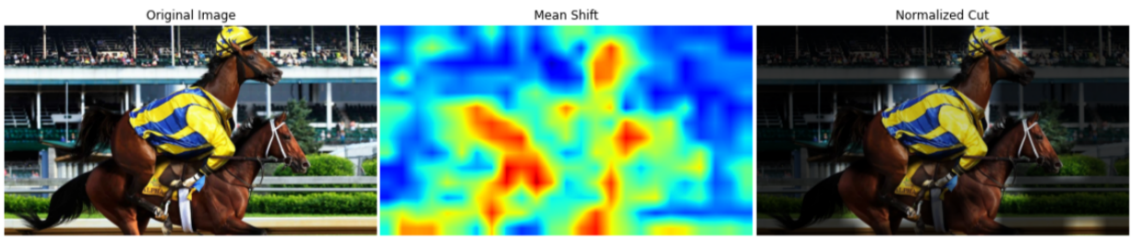

Figure 3.11: Localization of tampered region

Finally, the results for mean shift and normalized cut are resized (enlarged) and returned. The interpolation type is Inter Linear of the cv2 package.

CHAPTER 4

IMPLEMENTATION OF YOUR WORK

This chapter presents us with information about the algorithms used for implementing the modules discussed in chapter 3. Section 4.1 deals with the algorithms used in CNN approach and section 4.2 deals with the algorithms used in unsupervised self-consistency learning method.

4.1 CNN APPROACH

The algorithms involved in the CNN approach have been outlined in this section. 4.1 deals with the image patch extraction process, while 4.2 covers the extraction of features and identification of forgery.

Algorithm 4.1 Image Patch extractor

Input: input_path,output_path,patches_per_image,no_of_rotations,stride
Output: Rotated image patches

1: START
2: **for** each image in Tampered Images and Authentic Images **do**
3: Apply patch-sized sliding window of stride 128
4: **if** image belongs to Tampered Images **then**
5: Determine tampered patches where num_zeros ¡= 0.99 * (num_zeros + num_ones)
6: **end if**
7: Augment the patches by rotating them by 0, 90, 180 270 degrees.
8: GOTO 2
9: **end for**
10: Store the extracted patches in separate directories for authentic and tampered classes.
11: STOP

Algorithm 4.2 Feature Extraction and Forgery Classification

Input: 128x128x3 image patches
Output: 1 or 0 (Binary Classification)

1: START
2: The patches are fed into the CNN model, which extracts a 400-D feature representation for each patch.
3: The (n x 400-D) feature representations for an image must be fused into a single feature vector
4: These features are then passed to a fully-connected layer with a 2-way softmax classifier in the training phase and the SVM model in the testing phase.
5: The SVM model returns 1 if the image is tampered, and 0 otherwise.
6: STOP

4.2 UNSUPERVISED SELF-CONSISTENCY LEARNING

Algorithms 4.3 (preprocessing and consistency map extraction) and 4.4 (image segmentation) cover the major steps involved in the Unsupervised Self-Consistency Learning approach.

Algorithm 4.3 Input preprocessing and consistency map extraction

Input: Test Images
Output: Consistency Map List

1: START
2: Load the images to be tested.
3: Convert RGB to GBR colour scheme.
4: Unsqueeze the image's dimensions from (w, h, 3) to (1, 3, w, h).
5: Calculate stride size based on the image dimensions.
6: Apply patch-sized sliding window to extract patches of size 128 x 128.
7: Compare the obtained patches pairwise and get the probability score of consistency.
8: Obtain the consistency map list.
9: STOP

Algorithm 4.4 Image Segmentation

Input: Image,Image_Patches
Output: Segmented Images using Mean Shift and Normalized Cut

1: START
2: Compute the consistency map of a patch with respect to other patches considering each metadata attribute independently.
3: The resultant consistency map is used to plot the mean shift, taking the top 10 percentile of nearest points into consideration for a given point.
4: The normalized cut is obtained from the consistency maps using the spectral clustering method.
5: If most of the image is high probability, flip it.
6: The resultant images for mean shift and normalized cut are resized, showing the segments clearly.
7: STOP

CHAPTER 5

RESULTS AND PERFORMANCE ANALYSIS

This chapter contains the final output screenshots and performance analysis of this project - ”A Deep-Learning approach for detecting splicing & copy-move image forgeries and image self-recovery”. Section 5.1 discusses the performance of the CNN approach. Section 5.1.1 and 5.1.2 shows the results obtained for copy-move and splicing forgeries using CNN approach. Section 5.2 shows the outputs obtained using self-consistency learning method. Section 5.3 discusses about the UI of the web-app developed. Section 5.4 discusses the performance analysis of both the approaches.

5.1 CNN APPROACH

This section deals with the performance analysis of CNN approach. Section 5.1.1 discusses the accuracy achieved for training the CNN model, section 5.1.2 discusses the performance metrics related to SVM classification and section 5.1.3 shows the results achieved.

5.1.1 CNN - EPOCH VS TRAINING ACCURACY

The epoch vs training accuracy graph is depicted in figure 5.1. These results were obtained when the CNN is trained with the CASIA2 dataset.

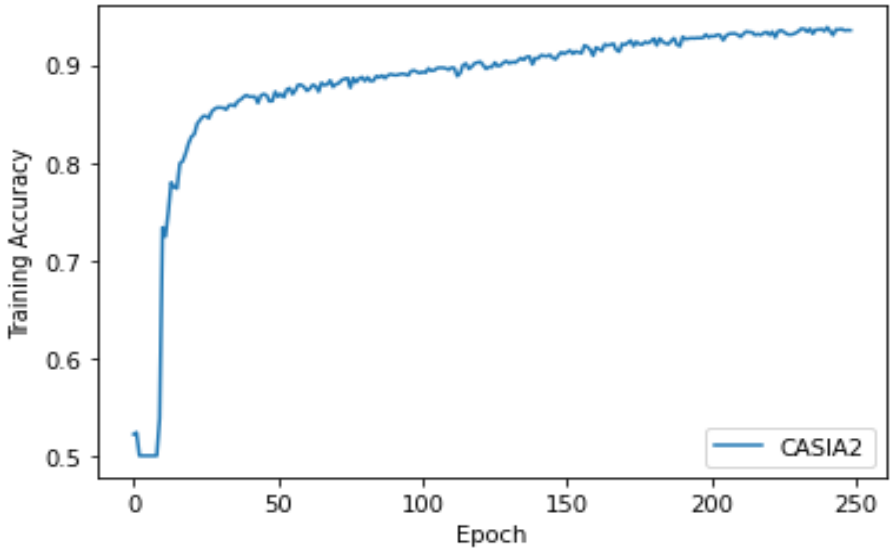

Figure 5.1: EPOCH VS TRAINING ACCURACY

As from the graph we can infer that as the Epoch increases the training accuracy also increases and reaches a saturation after which the training accuracy doesn't change much. So the number of epoch is stopped at 250 to prevent overfitting.

5.1.2 SVM PERFORMANCE

A confusion matrix is a tabular summary of the number of correct and incorrect predictions made by a classifier. It is used to measure the performance of a classification model. It can be used to evaluate the performance of a classification model through the calculation of performance metrics like accuracy, precision, recall, and F1-score. Figure 5.2 shows the confusion matrix when tested with the CASIA dataset.

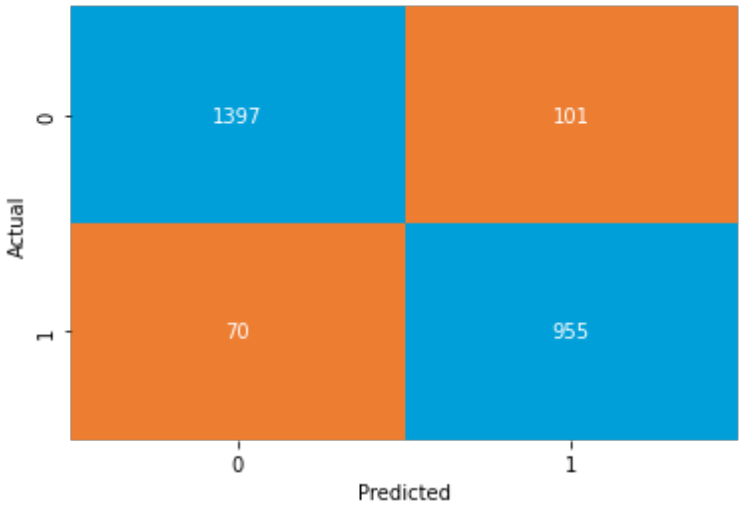

Figure 5.2: CONFUSION MATRIX

TP -Image is tampered and predicted as tampered
FP -Image is authentic but predicted as tampered
TN -Image is authentic and predicted as authentic
FN -Image is tampered but predicted as authentic

SVM performs binary classification. 0 indicates that the image is authentic and 1 indicates that the image is tampered.

PRECISION : Precision is defined as the ratio of correctly classified positive samples (True Positive) to a total number of classified positive samples. Our model has 90.43% of precision which depicts that almost 10% of tampered images are misclassified as original.

$$\text{Precision} = \frac{TP}{TP+FP}$$

RECALL : The recall is calculated as the ratio between the number of Positive samples correctly classified as Positive to the total number of Positive samples. The recall of our model is 93.1 so it means that 93.1% of tampered images are predicted correctly by our model.

$$\text{Recall} = \frac{TP}{TP+FN}$$

SPECIFICITY : Specificity is the metric that evaluates a model's ability to predict true negatives of each available category. In accordance to our project it refers to the percentage of original images predicted correctly.cOur model has a specificity of 93.25%.

$$\text{Specificity} = \frac{TN}{FP+TN}$$

F1 SCORE: The F1 score is defined as the harmonic mean of precision and recall. It is one of the most important evaluation metrics in machine learning. The F1 score of the model is 91.74%.

$$\text{F1} = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN}$$

SVM's performance with respect to the various metrics mentioned above has been summarized in the figure 5.3 below.

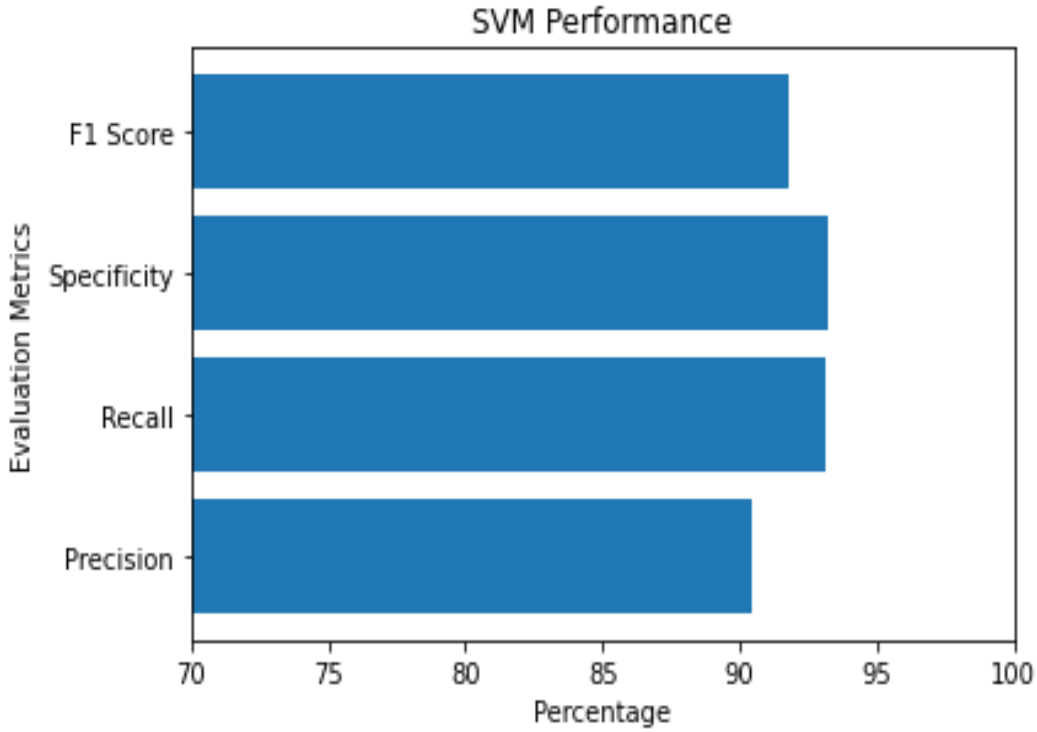

Figure 5.3: SVM PERFORMANCE METRICS

It performs well, with a score greater than 90% for each of the performance metrics considered. The results are thus found to be satisfactory.

The table 5.1 shows the Precision, Recall, Specificity and F1 Score of the SVM classifier.

Table 5.1: Performance analysis of SVM classifier

Precision	Recall	Specificity	F1 Score
90.43%	93.1%	93.25%	91.74%

5.1.3 CNN OUTPUT

Figure 5.4 shows the outputs of CNN approach when applied on copy-move tampered images.The images seen below have been taken from the CASIA dataset.

Figure 5.4: CNN outputs for localization (copy-move images)

The original image has been followed by its predicted forgery mask and the suspicious region. The final image shows the ground-truth of the tampered image.

Fig 5.5 shows the outputs of CNN approach when applied on spliced images. The images below for testing of spliced images are taken from CASIA and Labels in the wild.

Figure 5.5: CNN outputs for localization (spliced images)

The original image has been followed by its predicted forgery mask and the suspicious region. The final image shows the ground-truth of the tampered image.

5.2 SELF CONSISTENCY LEARNING OUTPUTS

Fig 5.6 shows the outputs of Self consistency learning approach.The below images are from Labels in the Wild which contains forgeries which are more difficult to predict compared to CASIA.

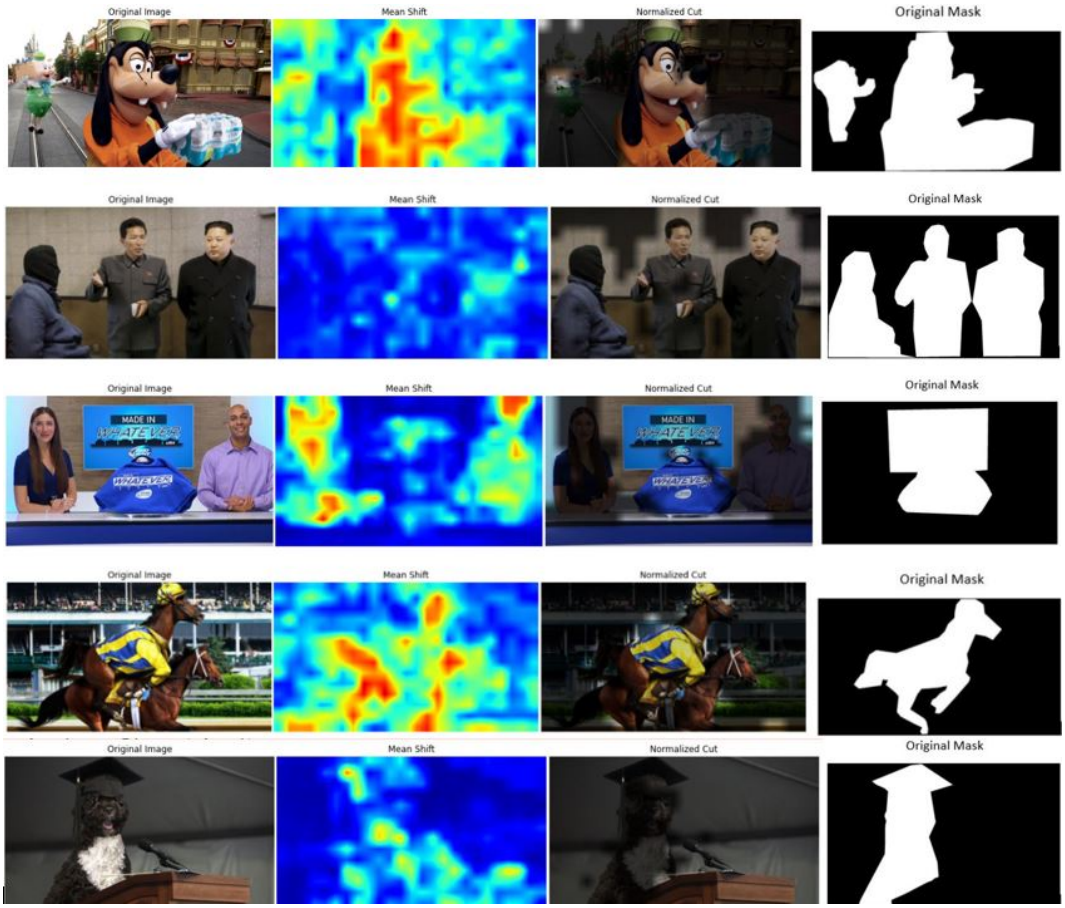

Figure 5.6: Self-consistency learning outputs

The original image has been followed by its corresponding Mean shift and Normalized cut based segmentation. The final image shows the ground-truth of the tampered image.

5.3 USER INTERFACE

The UI of the project is made using Streamlit, an open-source library available in python. Figure 5.7 is a screenshot of the web application developed.

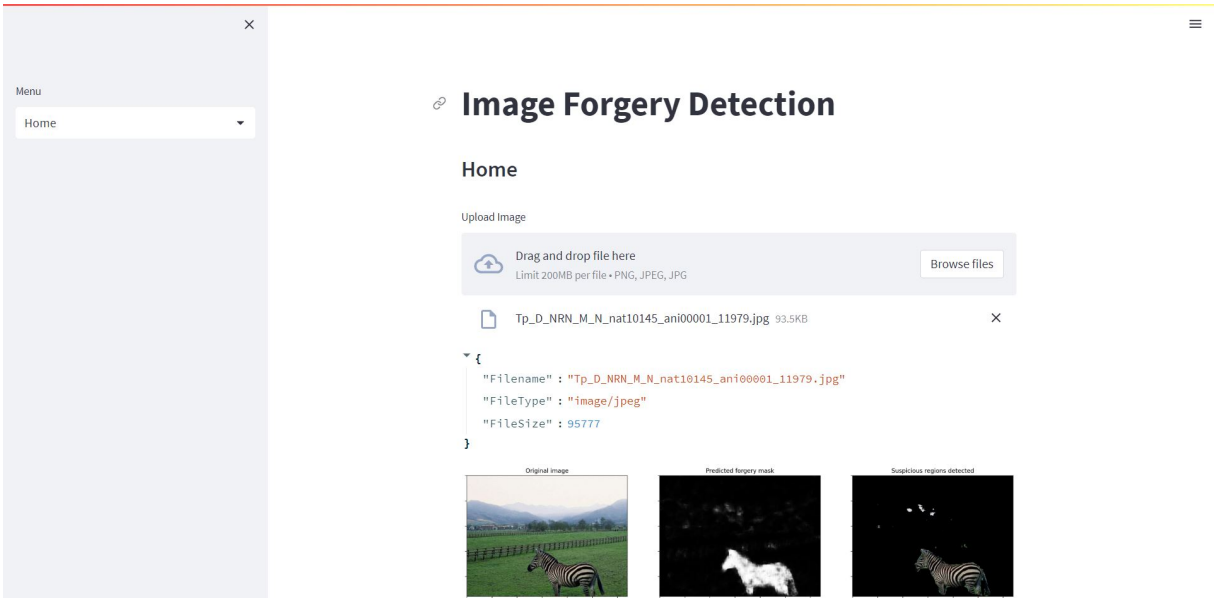

Figure 5.7: Tampering Localization in UI

The web app prompts the user to upload an image for testing. On image upload, the web app runs the deep learning model in the background to localize the exact region of tampering, if any in the uploaded image. Fig 5.7 shows a sample output of the same.

5.4 PERFORMANCE ANALYSIS

The CNN model performs well with obvious instances of tampering, in terms of recognizing whether an image has been tampered or not. Instances where the tampering is less obvious produce mixed results. The localization

of the tampered region (using MantraNet) produces satisfactory results when the training and testing data are from the same source. As it is a supervised CNN-based approach, the accuracy suffers when the domains for testing and training differ.

Self-consistency is more effective than CNN when it comes to identifying more subtle forms of tampering, but carries the downside of being unable to detect copy-move forgeries as the EXIF attributes of forged regions would match those of the original image. Factors like light exposure and the size of the region of splicing were found to impact the performance of the tamper localization. Over and under-exposed images often produced varying results. On testing with authentic images, the model sometimes flags tiny portions of the image that appear to have a different level of exposure as tampered and hence these portions appear as separate segments from the rest of the image.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter includes the final concluding remarks of our project, ”A Deep-Learning Approach for Detecting Splicing & Copy-Move Image Forgeries and Image Recovery,” in section 6.1 and future work to be done in section 6.2.

6.1 CONCLUSION

This project uses two deep learning approaches to detect and localise splicing and copy-move forgeries in images: CNN approach and Unsupervised Self-Consistency Learning. The project was developed using Python and its libraries like PyTorch, Pandas, and Matplotlib. A user interface was also developed using Streamlit, which allows a user to upload a test image and get the exact region of tampering in the image.

The CNN approach achieved a precision of 90.43%, Recall of 93.1%, Specificity of 93.25% and F1 Score of 91.74% for classification of forged or authentic images based on the CASIA dataset. The CNN approach was found to be more robust in detecting both splicing and copy-move image forgeries, whereas the self-consistency learning approach could only detect splicing image forgeries. However, when tested with the ’Label in the Wild’ dataset, it was discovered that the CNN approach did not perform well when the tamperings in the images were much more complex and difficult to be identified by the human eye. The Self-Consistency Learning approach, on the other hand, could detect much more complex splicing tamperings in images, but the total time taken to localise the region of forgery is significantly longer when compared to the CNN approach.

6.2 FUTURE WORK

The future work includes making the self-consistency approach more efficient so that it takes lesser time to localise the region of tampering. Future work will also include expanding the system to be able to reconstruct/recover the original image given a tampered image.

REFERENCES

[1] Mortda A.M. Fouda M.M. Hosny, K.M. and Lashin. An efficient cnn model to detect copy-move image forgery. 2022.

[2] Shaikh M. Gulhane A. Mallick, D. and Maktum. Copy move and splicing image forgery detection using cnn. 44:03052, 2022.

[3] Y. Rao and J. Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. *IEEE international workshop on information forensics and security (WIFS)*, pages pp. 1–6, 2016.

[4] Liu A. Owens A. Huh, M. and A.A. Efros. Fighting fake news: Image splice detection via learned self-consistency. *In Proceedings of the European conference on computer vision (ECCV)*, pages pp. 101–117, 2018.

[5] N. Takeda T. Hirose K. Taya, N. Kuroki and M. Numa. Detecting tampered regions in jpeg images via cnn. *18th IEEE International New Circuits and Systems Conference*, pages pp. 202–205, 2020.

[6] Qian Z. Zhou H. Xu H. Zhang X. Ying, Q. and S. Li. From image to imuge: Immunized image generation. *In Proceedings of the 29th ACM international conference on Multimedia*, pages pp. 3565–3573, 2021.

[7] Casia 2.0 image tampering detection dataset. `https://www.kaggle.com/datasets/divg07/casia-20-image-tampering-detection-dataset`.

[8] Dresden image dataset. `https://www.kaggle.com/datasets/hanjunyang1/dresden`.

[9] In-the-wild image splice dataset. *: The dataset consists of 201 images scraped from THE ONION, a parody news website, and REDDIT PHOTOSHOP BATTLES, an online community of users who create and share manipulated images*.

[10] Bhavsar A. Kumar, A. and R. Verma. Forgery classification via unsupervised domain adaptation. *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages pp. 63–70, 2020.

[11] W. AbdAlmageed Y. Wu and P. Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages pp. 9535–9544, 2019.

[12] K. J. Sani M. Kaya and S. Karakuş. Copy-move forgery detection in digital forensic images using cnn. *2022 7th International Conference on Computer Science and Engineering (UBMK)*, pages pp. 239–245, 2022.

[13] S. Rathod M. Baviskar and J. Lohokare. A comparative analysis of image forgery detection techniques. *2022 International Conference on Computing, Communication, Security and Intelligent Systems*, pages pp. 1–6, 2022.

[14] J. Y. Park S. I. Lee and I. K. Eom. Cnn-based copy-move forgery detection using rotation-invariant wavelet feature. *IEEE Access*, pages pp. 106217–106229, 2022.

[15] Y. H. Moon C. W. Park and I. K. Eom. Image tampering localization using demosaicing patterns and singular value based prediction residue. *IEEE Access*, pages pp. 91921–91933, 2021.

[16] A. Anjum G. Singh and S. Islam. Content prioritization based self-embedding for image restoration. *7th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages pp. 429–434, 2020.

[17] F. Rasouli and M. Taheri. A perfect recovery for fragile watermarking by hamming code on distributed pixels. *18th International ISC Conference on Information Security and Cryptology (ISCISC)*, pages pp. 18–22, 2021.

[18] D. Cozzolino and L. Verdoliva. Noiseprint: A cnn-based camera model fingerprint. *in IEEE Transactions on Information Forensics and Security, vol. 15*, pages pp. 144–159, 2020.