IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 21, 2021, accepted November 7, 2021, date of publication November 23, 2021,
date of current version December 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3130342

Detection and Localization of Multiple Image Splicing Using MobileNet V1

KALYANI KADAM[1], SWATI AHIRRAO[1], KETAN KOTECHA[2], AND SAYAN SAHU[1]

[1]Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, Maharashtra 412115, India
[2]Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International (Deemed University), Pune, Maharashtra 412115, India

Corresponding authors: Swati Ahirrao (swatia@sitpune.edu.in) and Ketan Kotecha (head@scaai.siu.edu.in)

This work was supported by Research Support Fund of Symbiosis International (Deemed University).

ABSTRACT In modern society, digital images have become a prominent source of information and medium of communication. The easy availability of image-altering softwares have greatly reduced the expenses and expertise required to exploit visual tampering. Images can, however, be simply altered using these freely available image editing softwares. Two or more images are combined to generate a new image that can transmit information across social media platforms to influence the people in the society. This information may have both positive and negative consequences. Hence there is a need to develop a technique that will detect and locate a multiple image splicing forgery in an image. This research work proposes multiple image splicing forgery detection using Mask R-CNN, with a backbone as a MobileNet V1. It also calculates the percentage score of a forged region of multiple spliced images. The comparative analysis of the proposed work with the variants of ResNet is performed. The proposed model is trained and tested using the MISD (Multiple Image Splicing dataset), and it is observed that the proposed model outperforms the variants of ResNet models (ResNet 51,101 and 151). The proposed model achieves an average precision of 82% on Multiple Image Splicing Dataset, 74% on CASIA 1.0, 81% on WildWeb, and 86% on Columbia Gray. The F1-Score of the proposed method on MISD was 67%, 64% on CASIA 1.0 68% on WildWeb, and 61% on Columbia Gray, outperforming ResNet variants.

INDEX TERMS Image forgery, multiple image splicing forgery, deep learning, MobileNet V1, Mask R-CNN.

I. INTRODUCTION

The human brain has an exceptional capacity for processing visual information. Most people respond to images more quickly than they do to texts. An image is worth a thousand words. Images are used in almost every area for communication, such as social media, news channels, military, court, insurance, education sector, entertainment business, health sector, and many more. With the development in image editing software tools and technologies available on portable devices such as smartphones and laptops, it is now possible to easily manipulate images for various purposes. These forged images may have a significant impact on society and can influence the views of people.

These days, social media campaigning has become a new trend in elections all around the world. On a more positive side, digital visuals are extensively employed to raise election awareness. At the same time, forged images with misinformation have been seen being distributed across social media to influence the public. According to a study [3], roughly 13.1% of Whatsapp posts were fraudulent during the last Brazilian presidential election. Furthermore, several fraudulent images containing misinformation regarding the COVID-19 pandemic recently went viral on social media platforms [4].

Several ways are available for forging the image such as image splicing and copy move. Image splicing [1], [2] merges two images to create a spliced image. Copy move uses a single image; in this, one object is copied into the same image. As a result, forgery detection techniques must be developed to ensure the authenticity of such images. Many researchers have proposed passive and active forgery detection techniques to authenticate digital images in recent years such as [1], [2]. The active forgery detection technique detects forgery in an image with the help of statistical information of an image. On the other hand, the passive method doesn't require such information to detect forgeries. Instead, they detect the forgery using the features of an image.

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

In Computer Vision (CV), the techniques used in earlier days for image splicing forgery detection rely on the traditional feature extraction methods. These features are primarily selected to focus on specific image properties and are generated manually. Due to this, these feature extraction methods are also referred to as handcrafted feature extraction methods. Some of the prominent handcrafted features used in image splicing forgery detection are the DWT [5], LBP [6], CT [7], HHT [8], and DCT [6], Bi-coherence, camera response operation, invariant image moments. The limitation of handcrafted features is that they are not robust and computationally heavy due to high dimensions.

The Deep Learning (DL) techniques shows extraordinary performance in various areas such as image processing, digital image forensics [9], [10], fraud detection, self-driving cars, virtual assistance, and face recognition system. Recent developments ([11]–[15]) have focused on DL-based image splicing detection, as compared to hand-crafted feature-based image splicing detection techniques. DL-based techniques can learn more generic features from the input image in general. As a result, in recent years, DL-based image splicing forgery detection algorithms have grown in popularity.

There are various region-based CNN(Convolution Neural Network) such as R-CNN([16], Fast R-CNN [17], Faster R-CNN [18], and Mask R-CNN [19] are available for object detection and segmentation. The R-CNN [16] excerpts many RPs from the input image, then utilizes a CNN on each RP to excerpt its features, which are then used to predict the RP's class and bounding box. In R-CNN near around 2000 image proposals are send to CNN. As a result, utilizing R-CNN to train and test the image is computationally expensive. To address this issue, the Fast R-CNN architecture was created, which takes the entire image as input. It also introduces the area of interest pooling layer, which allows features of the same shape to be retrieved for different-shaped ROI. To improve object detection accuracy, the fast R-CNN [17] model must generate a large number of region recommendations in selective search. The Faster R-CNN [18] replaces selective search with RPN to reduce region proposals without compromising accuracy. The Mask R-CNN [19] is the improved version of Faster R-CNN. It provides a class and bounding box for each ROI and it also provides the mask, i.e., the pixel-wise position of the object using FCN.

Various CNN networks have been introduced in the computer vision field, including AlexNet [20], which won the ILSVRC in the year 2012, increasing classification accuracy by 10% above typical machine learning algorithms. The University of Oxford's Visual Geometry Group suggested VGGNet [21] in 2014, and GoogLeNet [22], and ResNet [23] in 2015. To obtain increased accuracy, several CNN networks in the CV listed above are growing increasingly complicated. The depth and parameters of the DL networks listed above growing exponentially, making them more reliant on computationally efficient graphical processing units (GPUs) [24]. To overcome the limitations of previous research, this research work proposes a MobileNet V1-based lightweight DL classification network [25]. This network is based on the Depthwise Separable Convolution (DSC) [25], [26], which reduces convolution processing complexity and network parameter values, resulting in a lightweight network. The research on image splicing forgery detection faces challenges below:

- Lack of publicly accessible standard and custom datasets for detection of Multiple Image Splicing forgeries.
- Lack of forgery detection techniques for the detection of multiple image splicing forgeries.
- Lack of lightweight models which estimate the percentage score of the forged region of a multiple spliced image

Contributions:

- Detection, localization, and identification of passive forgeries like multiple image splicing using Mask R-CNN with pre-trained backbone networks such as ResNet 51, ResNet 101, ResNet 151, and MobileNet V1.
- Evaluation of MISD using pre-trained networks such as ResNet 51, ResNet 101, ResNet 151 with MobileNet V1.
- Comparative analysis of various image splicing datasets such as CASIA 1.0, WildWeb, and Columbia Gray with Multiple Image Splicing Dataset.
- Comparative analysis of MobileNet V1 with variants of ResNet 51, ResNet 101, ResNet 151.
- Calculation of training and inference time of the proposed method for various backbone networks such as ResNet 51, ResNet 101, ResNet 151, and MobileNet V1.
- To find out the percentage score for a forged region of a multiple spliced image.

This research work is structured as: Section 1 presents an introduction, sections 2 covers related work, section 3 outlines the MISD (Multiple Image Splicing Dataset) information and creation process, section 4 represents proposed architecture for multiple image splicing detection, section 5 outlines experimental setup, section 6 shows Dataset Annotation, section 7 specifies results, section 8 gives limitation of proposed research work, and section 9 presents the conclusion.

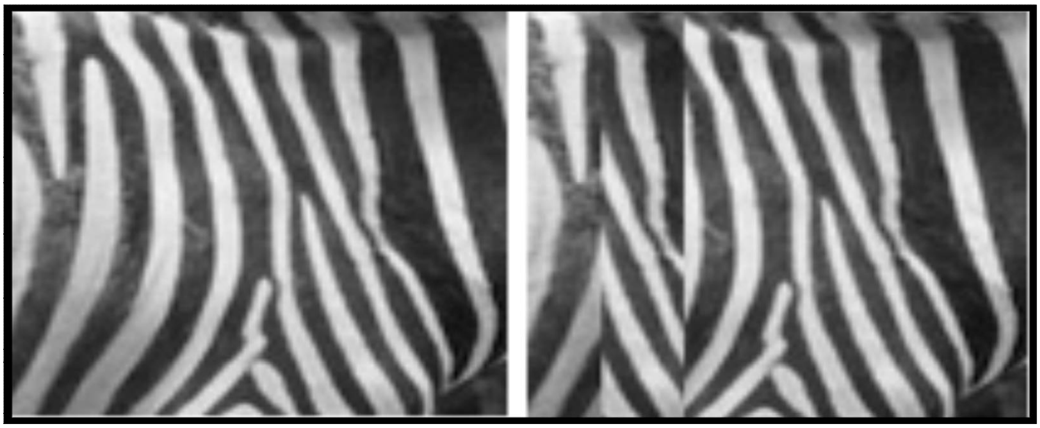
II. RELATED WORK

Existing work on image splicing forgery detection is explored with respect to the dataset and deep learning models. This section discusses the dataset employed by researchers for the detection of image splicing forgery. Table 1 shows a summary of image splicing datasets used for image splicing forgery detection. Figure 1,2,3 and 4 shows sample image from Columbia Color, CASIA 1.0, CASIA 2.0 and WildWeb dataset.

A. DATASETS FOR IMAGE SPLICING

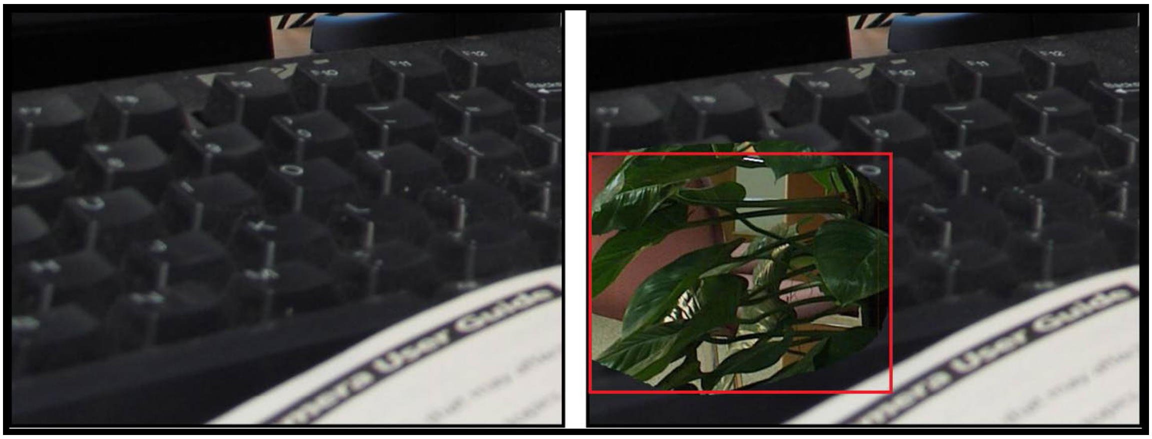
1) COLUMBIA GRAY [27]

This dataset contains 1845 image blocks, out of which 933 are AU, and 912 are SP. The AU and SP image blocks are having



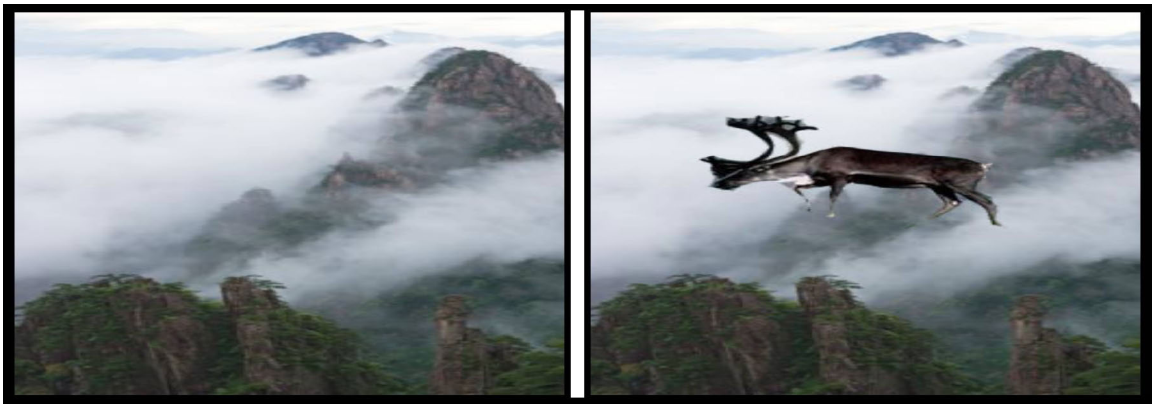
a) AU Image b) SP Image

FIGURE 1. AU and SP Image from Columbia gray.



a) AU Image **b) SP Image**

FIGURE 2. AU and SP Image from Columbia color dataset.



a) AU Image b) SP Image

FIGURE 3. AU and SP Image from CASIA 1.0 dataset.

a size of 128 × 128 pixels. These image blocks are in BMP image format with simple cut-paste operation without any post-processing operations. In this, the cut-paste operation is performed using Adobe Photoshop [28]. In this dataset, the image blocks are grayscale, retrieved from 322 photos, 10 are captured using a camera by the authors, and 312 are

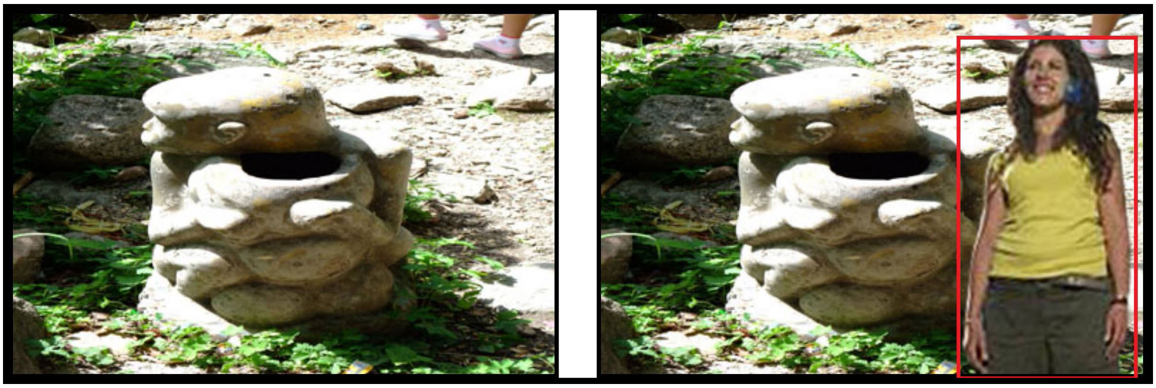


FIGURE 4. AU and SP Image from CASIA 2.0 dataset.

taken from the CalPhotos dataset [29]. The limitation of this dataset is it provides only a grayscale image block, not the color. It also does not provide, the ground truth masks for spliced image blocks.

2) COLUMBIA COLOR [30]

Columbia color dataset addresses the shortcomings of the Columbia Gray dataset. This dataset has 363 images, 183 are AU, and 180 are SP. All color images in this dataset are in TIFF image format, with image dimension range varying from 757 × 568 to 1152 × 768 in pixels. In this dataset, the authentic images are captured using cameras such as *canong3*, *nikond70*, *canonxt*, and kodakdcs330. The spliced images are constructed from authentic images using Adobe Photoshop. The images in the authentic category have indoor and outdoor scene images that contain various objects such as keyboards, books, tables, etc. For this dataset, edge masks are provided, which represent the spliced objects boundaries.

3) CASIA 1.0 [31]

The CASIA 1.0 dataset consists of 1725 images, 800 are Authentic (AU) and 925 are SP. All images in this dataset are of JPG image format with a dimension of 384 × 256. The SP images are constructed using Adobe Photoshop by performing copy and paste operations on AU images.

4) CASIA 2.0 [31]

The CASIA 2.0 dataset contains a total of 12614 images, 7491 of which are AU images, and 5123 are forged. This dataset contains both copy move and image splicing images. Thus, there are 3274 images of copy move and 1849 images of image splicing. The images are in JPEG and TIFF image formats. For image splicing images, 753 out of the 1849 SP images are in TIFF format, while 1096 are in JPG image format. The dimension of images in pixels ranges from 320 × 240 to 800 × 600. CASIA 1.0 and CASIA 2.0 are constructed using Adobe Photoshop CS3 version 10.0.1 on Windows XP. The images in these datasets are of various categories: *animal, architecture, art, indoor, nature, plant, scene, and texture*. But, both datasets do not provide ground truth mask information for copy move and spliced images.

5) DSO-1 [32]

This dataset contains 200 images, 100 AU, and 100 SP images, including indoor and outdoor images with image dimensions of 2048 × 1536 pixels. In this dataset, SP images are created by adding one or two people to the AU image. It applied post-processing operations to a few SP images, such as color and brightness modification to create more realistic images.

6) DSI-1 [32]

Carvalho *et al.* [32] constructed this dataset, and it contains a small set of popular image splicing categories acquired from the Internet. This dataset comprises 50 images, out of which 25 are AU, and 25 are SP of different dimensions.

7) WildWeb [33]

This dataset's images are gathered via Internet sources. There are a total of 9666 spliced images created from 82 categories. The majority of the images in the dataset are in JPEG format, and the remaining are of type PNG, GIF, and TIFF. The images inside this dataset are difficult for splicing localization as they have gone through post-processing operations such as re-save and resample. In addition, this dataset includes a ground truth mask for spliced images. But, the dataset is not publicly accessible. However, it is available to the authors upon request for study purposes.

B. CUSTOM DATASET

AbhAS [34]: This dataset contains 93 images, out of which 45 are AU, and 48 are SP. The images in this dataset are of JPG image format with dimensions ranging from 278 × 181 to 3216 × 4288). In this dataset, 19 authentic images are taken from a single source camera, and the remaining 26 images are taken from the Internet. The spliced images are created using Adobe Creative Cloud 2020 version with Photoshop. The ground truth masks are also available for these spliced images. A sample image from the AbhAS dataset is shown in Figure 5.

Challenges in the Existing Datasets:

I. *Standard Dataset:*

- All the standard dataset contains splicing images which merge only two images for splicing operation.
- There are no standard datasets available for the detection of multiple image splicing forgery.
- There are no datasets available that are containing ground truth masks for multiple spliced objects.
- Some of the existing datasets have used only cut and paste operations for the creation of datasets.

II. *Custom Dataset:*

- Very few images are available for image splicing. In addition, it does not contain multiple spliced images
- These datasets do not contain images from various categories.

Figure 6 shows the challenges in the existing standard as well as in the custom dataset.

C. DEEP LEARNING MODELS USED FOR IMAGE SPLICING

This work [11] uses DL based approach for image splicing detection. In this, CNN is used to learn hierarchical features of the input image. The weights of this network's first layer are set to the value of the basic high-pass filter. It is used for the calculation of residual maps in the SRM. Next, the pretrained CNN is utilized as a patch descriptor to collect dense features from the test images. Then a feature fusion method is employed to get the final discriminative features for SVM classification. This research work [12] proposed a solution based on the ResNet-Conv deep neural network. Two variations of ResNet-Conv, ResNet-50, and ResNet-101, were utilized to build an initial feature map from RGB images. The authors also offered the *Mask-RCNN* for locating a forgery.

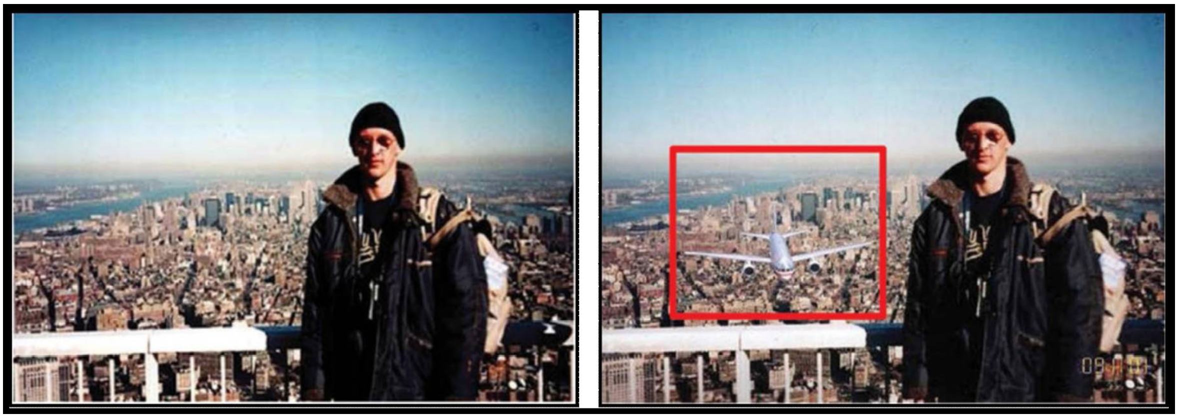
In this research work [35], two techniques such as combining resampling features and deep learning, are used to identify and locate image forgery. The first technique computes Radon transform of resampling features on overlapping image patches. A heatmap is then generated using deep learning classifiers and a Gaussian conditional random field model. Finally, a Random Walker segmentation approach is used to locate forged regions. In the second technique, resampling features obtained from overlapping image patches are passed to LSTM for classification and localization.

Handcrafted features are frequently used to detect manipulated areas in a synthetic image to uncover and locate splicing forgeries. However, for a given spliced image without prior information of image splicing, it is difficult to tell which feature will be effective to expose forgery. Furthermore, a particular handcrafted feature can only deal with one type of splicing forgery. This research work proposes [36] a technique based on *deep neural networks* and *conditional random* to overcome this issue. Three distinct *FCNs* plus a *condition random field* are used to achieve this. Each *FCN* is trained to deal with image scales of varying sizes. *CRF* combines the detection findings from these neural networks in an adaptive way. The trained *FCNs–CRF* can subsequently be utilized to perform image authentication and forecast pixel-to-pixel forgery. Thus, *FCNs–CRF* model outperforms compared to existing techniques that rely on handcrafted features. This research work [37] proposes two *FCN*, to identify image splicing. The initial network is a single-task network that primarily learns the attributes of surface labels. The next network, on the other hand, is a two-path multi-task network. This two-path network primarily learns the spliced region's edge or boundary.

The Conditional Generative Adversarial Network (cGAN) was used in this study [38] to detect spliced forgeries in satellite images. It detected and located the spliced objects with high accuracy. This research study used Mask R-CNN along with ResNet and Sobel filter for detection of copy move and image splicing forgeries. In this experiments were performed on Columbia Color standard dataset of image splicing and which shows that this model outperforms the state-of-art methods. Recently researchers published a DL-based image splicing technique [14] that employs a convolutional neural network and a weight combination mechanism. Three distinct features were used in this method: YCbCr features, edge features, and PRNU features. These three features were combined, and their weight settings were automatically modified during the CNN training process, unlike the other approaches, until the best ratio is attained.

This research study [40] uses color illumination, deep convolution neural networks, and semantic segmentation to detect and localize image splicing forgery. After the preprocessing step, color illumination is employed to apply the color map. The deep convolution neural network is used to train VGG-16 with two classes using the transfer learning approach. This research study determines whether or not a pixel is fake. To locate forged pixels, semantic segmentation is used to train these classed images using color pixel labels. This research work [9] uses ResNet 50 model and the 'Noiseprint' technique for image splicing forgery detection. Firstly, the input image is preprocessed using the 'Noiseprint' technique to obtain the noise residual, suppressing the image content. Then ResNet-50 network is deployed for feature extraction. Finally, using SVM classifier, the collected features are classified as SP or AU. For automatic feature selection, this study [41] uses a deep convolutional residual network. These features are then send into a classifier network, which determines if the image is authentic or spliced. This model was trained and evaluated on the CASIA 2.0 dataset. Experiments shows that this technique outperforms state-of-the-art techniques for image splicing identification, but it does not locate spliced regions.

Table 2. summarizes a few DL techniques for image splicing forgery detection proposed in the existing literature.



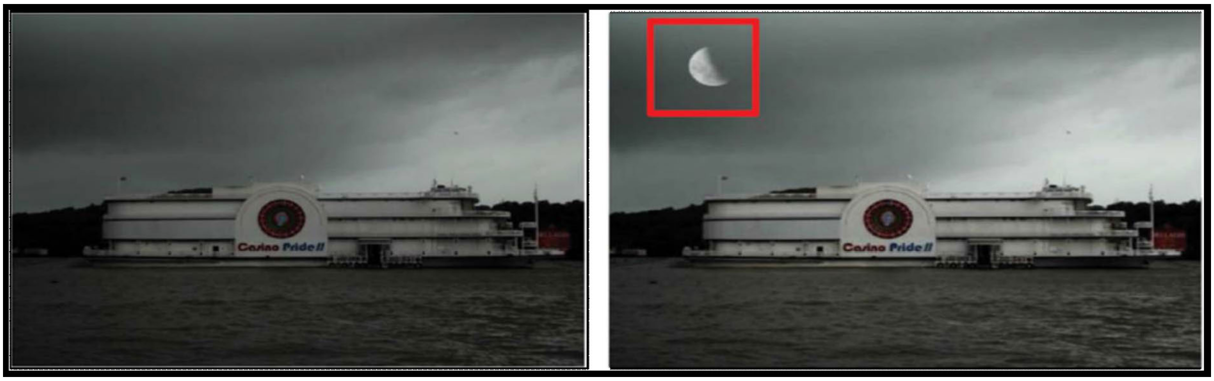
a) AU Image

b) SP Image

FIGURE 5. AU and SP Image from WildWeb dataset.

TABLE 1. Image splicing datasets.

Sr. No	Dataset Name	Type of Dataset	No of images	Size of image	Total No of Authentic images	Total No of Spliced images	Availability of ground truth mask	Post Processing Operations
1	Columbia Gray [27]	Standard Datasets	1845 image blocks	128 x 128	933	912	No	Cut paste
2	Columbia Color [30]	Standard Datasets	363	757 × 568 to 1152 × 768	183	180	Yes	Cut paste
3	CASIA 1.0 [31]	Standard Datasets	1725	384 × 256	800	925	No	scaling ,compession,rotation
4	CASIA 2.0 [31]	Standard Datasets	12614	320 ×240 and 800 × 600	7491	1849	No	scaling,compession,rotation
5	DSO-1 [32]	Standard Datasets	200	2048 ×1536 and 1536 × 2048	100	100	Yes	resizing ,exposure adjustment
6	DSI-1 [32]	Standard Datasets	100	Different sizes	25	25	Yes	resizing ,exposure adjustment
7	WildWeb [33]	Standard Datasets	10666	122 × 120 to 2560 × 1600	100	9666	Yes	resize,nose inclusion,compression
8	AbhAS [34]	Custom Dataset	93	278× 181 to 3216× 4288)	45	48	Yes	resizing,blur,



a) AU Image

b) SP Image

FIGURE 6. AU and SP Image from AbhAS dataset.

III. MISD DATASET (MULTIPLE IMAGE SPLICING DATASET) [44]

The construction of the Multiple Image Splicing dataset is described in detail in this section. The diagram depicts the entire procedure.

A. DATASET DESCRIPTION

This dataset contains 918 images, 618 of which are AU images, and 300 are multiple spliced images. The AU images are taken from CASIA 1.0 [31] dataset. Multiple spliced images are created by performing various image

TABLE 2. Summary of commonly used DL techniques for image splicing detection.

Paper	Technology	Dataset	Performance Metrics
[11]	CNN	CASIA V1.0[31],CASIA V2.0[31],Columbia Gray[12]	Accuracy on CASIA 1.0,CASIA 2.0 and Columbia Gray is 98.04%,97.83% and 96.38% respectively.
[12]	Mask R-CNN with backbone network as ResNet-conv	Computer-generated dataset where forged images have been generated using COCO and a set of objects with transparent backgrounds where 80,000 images are used for training and 40,000 for validation. The image size is 480 × 640 pixels.	AUC= 0.967
[35]	CNN,LSTM	UCID[42] and RAISE [43]	Classification = 94.86% and AUC=0.9138
[36]	FCN (Fully Convolution Network), CRF (Condition Random Field)	CASIA V2.0[31]	F1-Score of the proposed method without compression = 0.4795, JPEG Quality with 70 =0.4496, JPEG Quality with 50 = 0.4431 . F1-Score of the proposed method without noise = 0.4795 , SNR with 25 dB =0.4786, SNR with 20 dB =0.4811 , SNR with 15 dB =0.4719
[37]	Fully Convolutional Networks (SFCN,MFCN, Edge-enhanced MFCN)	CASIA V1.0[31],CASIA V2.0[31],Columbia Gray[11], Nimble 2016 SCI, Carvalho [14]	F1- Scores on CASIA V1.0=0.5410 ,Columbia = 0.6117, Nimble 2016 SCI = 0.5707 , Carvalho = 0.4795
[38]	cGAN	The Landsat Science Programme, which is administered jointly by NASA and the US Geological Survey (USGS), provided the images utilized in this study.	AUC=0.988 , Precision for localization of forgery = 0.953
[39]	Mask R-CNN with backbone network as ResNet 101	Columbia Color [19]	F1- Score on Columbia Color= 0.7825
[14]	Convolutional Neural Network (CNN)	CASIA V1.0[31],CASIA V2.0[31]	Detection Accuracy for CASIA 1.0 = 99.45 and for CASIA 2.0= 99.32 %
[40]	DCNN (deep convolution neural network), Semantic Segmentation	DVMM[19], BSDS300	100% classification accuracy. Pixel level forgery localization - DVMM has 98.69% accuracy
[9]	ResNet 50,SVM	Columbia Color [19]	Detection Accuracy = 97.24%
[41]	CNN,dense classifier network	CASIA V2.0[31]	Accuracy = 0.9645,Precision= 0.9669,Recall= 0.9415, F1 score= 0.9540

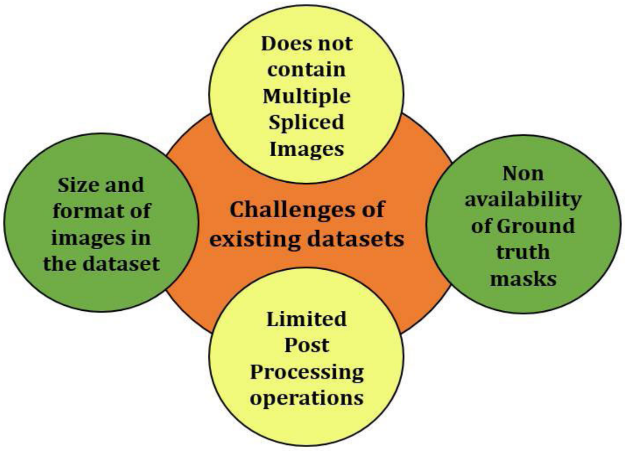

Does not contain Multiple Spliced Images
Size and format of images in the dataset
Challenges of existing datasets
Non availability of Ground truth masks
Limited Post Processing operations

FIGURE 7. Challenges of existing datasets.

editing operations with FIGMA software [45] on authentic images.

The images under AU are of nine categories: *animal, architecture, art, character, nature, indoor, scene, texture, and plant*. The ground truth masks are also available for these multiple spliced images. Table 3 gives the overall description of this dataset. Figure 6 shows the sample images for the Multiple Image Splicing dataset and figure 7 gives the different steps for the creation of this dataset.

Following steps are followed for the construction of this dataset.

a) Firstly, an image is uploaded into Figma software from authentic images. This uploaded image act as a source for the inclusion of various images.
b) A background removal software, such as removing bg [46], cuts the objects from other authentic images. This software is used to remove the background of images. An image with a clear high contrast differentiation between the image's subject and background is preferred to achieve the best potential results. The generated image after the background removal is pasted on top of the base image using Figma software. The inserted objects are then subjected to different manipulation procedures such as transformation, rotation, brightness adjustment, and scaling to create the spliced images that appear more real and tougher to identify.
c) Finally, all the added images/objects and the base image are selected, merged, and exported as a single image.
d) The process is repeated with various authenticated images, and multiple images with backgrounds removed are added to the base image.
e) Then these multiple spliced images are manually annotated using the VGG image annotation tool [47].

TABLE 3. Description of multiple image splicing dataset.

	Number of images per category		Image size	Type of Image	Total number of images
Authenticate Images	Animal	167	384 × 256	JPG	618
	Architecture	35			
	Art	76			
	Character	124			
	Indoor	7			
	Nature	53			
	Plant	50			
	Scene	74			
	Texture	32			
Multiple Spliced Images	Images of all categories	300	384 × 256	JPG	300

f) Lastly, ground truth masks are generated for each multiple spliced image with a Python script which helps in identifying the spliced objects inside multiple spliced image.

IV. PROPOSED ARCHITECTURE FOR MULTIPLE SPLICING

The proposed methodology is used for multiple splicing forgery detection as well as for calculating forged regions is explained. The percentage score of a forged region of a multiple spliced image is given by the formula 1.

P = number of pixels of the entire image
Q = number of pixels of the forged region
R = forged percentage of the region
S = Image dimension in pixels

$$R = \frac{(P - Q)}{S} \times 100 \tag{1}$$

The proposed system makes use of the Mask Regional Convolutional Neural Network (MASK R-CNN) framework, which is one of the futuristic object detection systems. In the proposed model MobileNet V1 backbone is used to extract the features of the forged region from the input image, and the Region of Interest (ROI) is quickly generated on the feature maps with the help of the RPN. ROIAlign is also used to maintain the exact spatial locations and to output the feature maps in fixed dimensions. Eventually, the network detects the location of the bounding box, the corresponding segmentation mask is generated on the forged region and the class of forged region using the Fully Convolutional Network (FCN). The key stages proposed in the network are discussed in detail in the below sections.

A. BACKBONE

The architecture used for image forgery detection [39] used Mask R-CNN and which was consists of two backbones that act as feature extractors: the deep residual networks (ResNets) and the feature pyramid networks (FPN) and each of them corresponds to a mask head architecture. To optimize the feature extractors and make the architecture lightweight, MobileNet is used in the proposed model. MobileNet is a lightweight neural network that can simplify the model, reduce the number of parameters significantly, and greatly increase the detection speed of the model without compromising on accuracy. The use of the other feature extractor: the FPN sometimes become time-consuming. To get a balance between the speed and accuracy of the model, MobileNet has been incorporated as part of the feature extractor of Mask R-CNN for the instance segmentation of forged regions in the image.

The architecture of MobileNet uses depth-wise separable convolution which segregates a conventional convolution into a pointwise and depth-wise convolution. The depth-wise convolution has just one convolution kernel for a corresponding input channel. The pointwise convolution makes use of a 1 × 1 convolutional kernel to linearly fuse the outputs from the depth-wise convolution. There are no pooling layers present in between the depth-wise separable layers. Both convolutions are succeeded by a batch normalization layer and a Rectified linear unit (ReLU6) activation function, which is like ReLU, but it prevents the activations from becoming large. MobileNet uses two hyperparameters: depth multiplier and resolution multiplier. The depth multiplier is used to change the number of channels in each layer and the resolution multiplier is used to control the resolution of the output image. These hyperparameters greatly optimize the computation speed and load.

B. FEATURE EXTRACTION USING FEATURE PYRAMID NETWORK

The Feature Pyramid Network (FPN) is used as a feature extractor to increase the accuracy. This acts as a replacement for feature extractors which is used in Faster R-CNN and constructs multi-scale feature maps for providing better information than the conventional feature pyramid. The detection of small objects is difficult therefore FPN is used. By using the pyramid of the same image this can be solved to detect the objects easily. In FPNs, several stages of feature maps are effectively fused. FPNs not only utilize deep but also shallow feature maps, which are very helpful for the detection of forged objects which are quite small. Improved Mask R-CNN by merging MobileNet and the FPN, constantly achieves better accuracy with much fewer parameters and faster speed than the conventional Mask R-CNN.

C. REGION PROPOSAL NETWORK (RPN)

After backbone, each of the feature image is sent to the RPN. It is a lightweight deep neural network. The Region of Interest (ROI) is generated directly on the feature map along with the Region Proposal Network (RPN). RPN has a distinctive architecture consisting of regressors and classifiers. An image having an unfixed size is given as an input to the RPN, which results in a set of rectangular object proposals in conjunction with an object score. It figures out whether the anchor belongs to the foreground or the background. For all the anchors belonging to the foreground, RPN performs the first coordinate correction. The RPN uses sliding windows on the convolutional feature maps to produce a fixed number of object boxes having a predetermined aspect ratio and a scale for each pixel. These are known as anchor boxes. An anchor is positioned at the sliding window and is correlated with an aspect ratio and scale. After that, these object proposals are then supplied to two related-connected layers: the Classifier - for object identification and the Regressor-for bounding box generation.

D. RoIAlign

The Region of Interest (ROI) is generated directly on the feature map along with the Region Proposal Network (RPN). The alignment of the ROI is crucial for dense feature map representation and for preserving regularity during convolving, predicting the class along with its segmentation mask and bounding box. But in the ROI pooling, the coordinates in the feature map suffer from quantization and the object location becomes misaligned. To avoid this and to accurately construct the ROI pool, Bilinear Interpolation called ROIAlign is proposed in Mask R-CNN replacing the ROIPooling in Faster R-CNN. In ROIPooling, there are two rounding procedures. One is the conversion of the coordinates and dimensions of the region proposal into integers for convenience and the other is to divide the boundary area of the region proposal into $k \times k$ cells and convert the boundary of each of the cells into integers. But after these two rounding operations each of the region proposals gets deviated from their original position. The improved version of ROI pooling i.e., Region of Interest Align (ROIAlign) makes use of Bilinear Interpolation instead of the rounding operation of the ROI pooling to get a smooth out interpolation and to get the coordinates in float, thus avoiding misalignments between the ROI and the extracted features. It is used to get the pixel-level segmentation of the images. The ROIAlign is also used to maintain the exact spatial locations and return the feature map to a fixed size.

E. FULLY CONVOLUTIONAL NETWORK

The Fully Convolutional Network (FCN) is applied to each ROI to predict the segmentation mask of the image tampering region in a pixel-to-pixel fashion. The FCN strategy arises from the traditional CNN network architecture but is also a little different from it. In CNNs, to get the feature vectors in fixed dimensions, the convolutional layer is connected to numerous full connection layers which results in an output that gives the numerical description of the input. The FCN uses CNN to transform image pixels into the number of pixel classes using a 1×1 convolutional layer. So, the FCN is quite like a CNN network, but the FCN network restores the output image size to that of the original image using deconvolution to up-sample the feature map, this is achieved by the transposed convolutional layer. After this, the classification output and the input image have a one-to-one association in pixel level. The dimension at any of the output pixel has the classification results for the corresponding input pixel at the same spatial location. Eventually, the FCN uses the Softmax classifier to predict the category of each of the pixels.

F. LOSS FUNCTION

The proposed model mainly consists of two stages, the first stage involves the RPN which proposes the candidate bounding boxes for the forged region and the second stage involves the feature extraction from each of the candidate boxes and then performing classification and calculation of the location of a bounding box. The multi-task loss function of Mask R-CNN for each proposal is calculated. This function includes the classification loss $\boldsymbol{L_{cls}}$, the segmentation loss $\boldsymbol{L_{mask}}$, and the bounding box location loss or the regression loss $\boldsymbol{L_{box}}$.

$$L = L_{cls} + L_{mask} + L_{box} \tag{2}$$

The classification and the regression loss need to be calculated using the formula shown below:

$$L_{cls} + L_{box} = \frac{1}{N_{cls}} \sum_{i} L_{cls}\left(p_i, p_i^*\right) + \lambda \frac{1}{N_{reg}} \sum_{i} p_i^* L_{reg}\left(t_i, t_i^*\right) \tag{3}$$

where i is the index of an anchor, p_i is the predicted probability of the anchor, t_i is the four coordinate parameters of the box, t_i^* is the four coordinate parameters of the ground truth box for the required positive anchor. for a positive anchor or else it is zero. The loss function must be minimized to optimize the model.

The architecture used in the proposed system is shown in Table 4. Each row in the table represents a sequence of one or more identical layers, which repeats n times. For a layer in a particular sequence, the output channel is c. Apart from the first layer in each sequence whose stride is s, all other layers use a stride of 1. All depth-wise separable convolutions use 3×3 kernels. MobileNet uses five convolutional layers Conv1-5 in the RPN network to get the ROIs. The architecture of MobileNet shown in the table uses depth-wise separable convolution, which segregates a conventional convolution into a pointwise and depth-wise convolution. The depth-wise convolution has just one convolution kernel for a corresponding input channel. The pointwise convolution uses a 1×1 convolutional kernel to fuse the outputs from the depth-wise convolution linearly. There are no pooling layers present in

TABLE 4. Overall architecture of MobileNet V1.

Layer Name	Input	Block Operator	c	n	s
Input Image	224 × 224 × 3	Conv2d	32	1	2
Conv 1	112 × 112 × 32 112 × 112 × 64	DepthWiseConv DepthWiseConv	64 128	1 1	1 2
Conv 2	56 × 56 × 128 56 × 56 × 128	DepthWiseConv DepthWiseConv	128 256	1 1	1 2
Conv 3	28 × 28 × 256 28 × 28 × 256	DepthWiseConv DepthWiseConv	256 512	1 1	1 2
Conv 4	14 × 14 × 512 14 × 14 × 512	DepthWiseConv DepthWiseConv	512 1024	5 1	1 2
Conv 5	7 × 7 × 1024 7 × 7 × 1024	DepthWiseConv -	1024 -	1 -	1 -

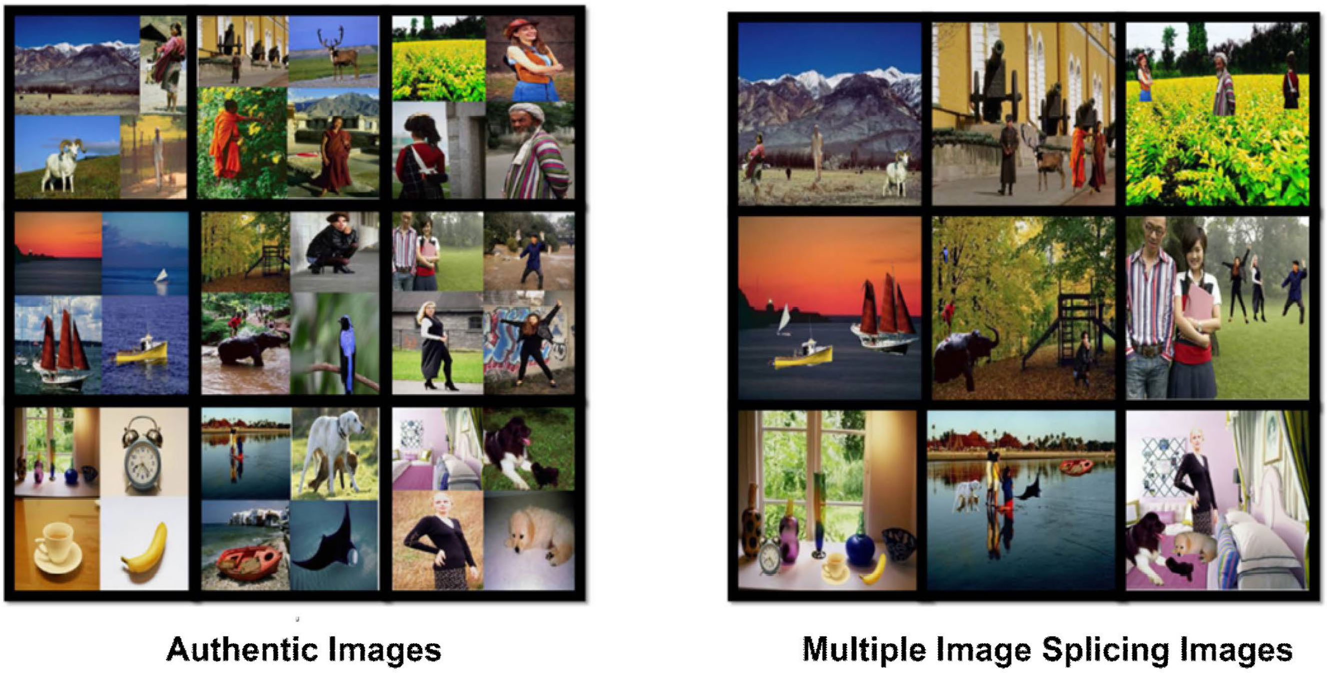


FIGURE 8. Authentic and spliced images from MISD dataset.

between the depth-wise separable layers. Both convolutions are succeeded by a batch normalization layer and a Rectified linear unit (ReLU6) activation function, like ReLU, but it prevents the activations from becoming large. MobileNet uses two hyperparameters: depth multiplier and resolution multiplier. The depth multiplier is used to change the number of channels in each layer, and the resolution multiplier is used to control the resolution of the output image. These hyperparameters greatly optimize the computation speed and load.

V. EXPERIMENTAL SETUP

This section gives the experimental setup for multiple image splicing forgery detection. Tables 5, 6 and 7 show the system specifications and parameters of the training environment. All experiments are conducted using *NVidia 1xTesla K80, compute 3.7, having 2496 CUDA cores with 12GB GDDR5 VRAM in google collaboratory*; the operating environment has *1xsingle core hyperthreaded Xeon Processors @2.3Ghz, i.e. (1 core, 2 threads) with 13 GB RAM. Tensorflow 1.8.0* is an open-source deep learning framework, and Python 3.7 is used as a programming language. COCO pre-trained network [48] was used as the starting point to train the model. Table 4 shows a few configuration parameters which were modified from the original Mask R-CNN. For the MISD dataset this experiment uses 232 images for training and 35 images for testing, and 35 images for validation, CASIA 1.0 dataset experiment uses 737 images for training and 92 images for validation, and 92 images for testing, the Columbia Gray dataset, uses 730 images for training and 91 images for validation, and 91 images for testing, WildWest dataset, uses 515 images for training and 64 images for

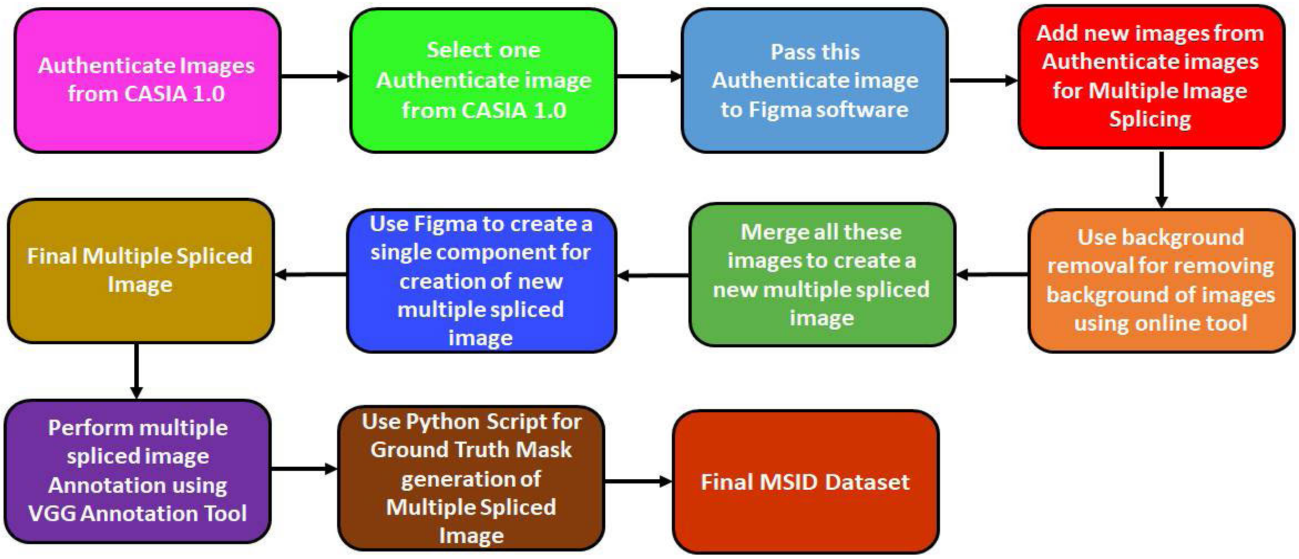

Authenticate Images from CASIA 1.0
Select one Authenticate image from CASIA 1.0
Pass this Authenticate image to Figma software
Add new images from Authenticate images for Multiple Image Splicing
Use background removal for removing background of images using online tool
Merge all these images to create a new multiple spliced image
Use Figma to create a single component for creation of new multiple spliced image
Final Multiple Spliced Image
Perform multiple spliced image Annotation using VGG Annotation Tool
Use Python Script for Ground Truth Mask generation of Multiple Spliced Image
Final MSID Dataset

FIGURE 9. MISD dataset creation steps.

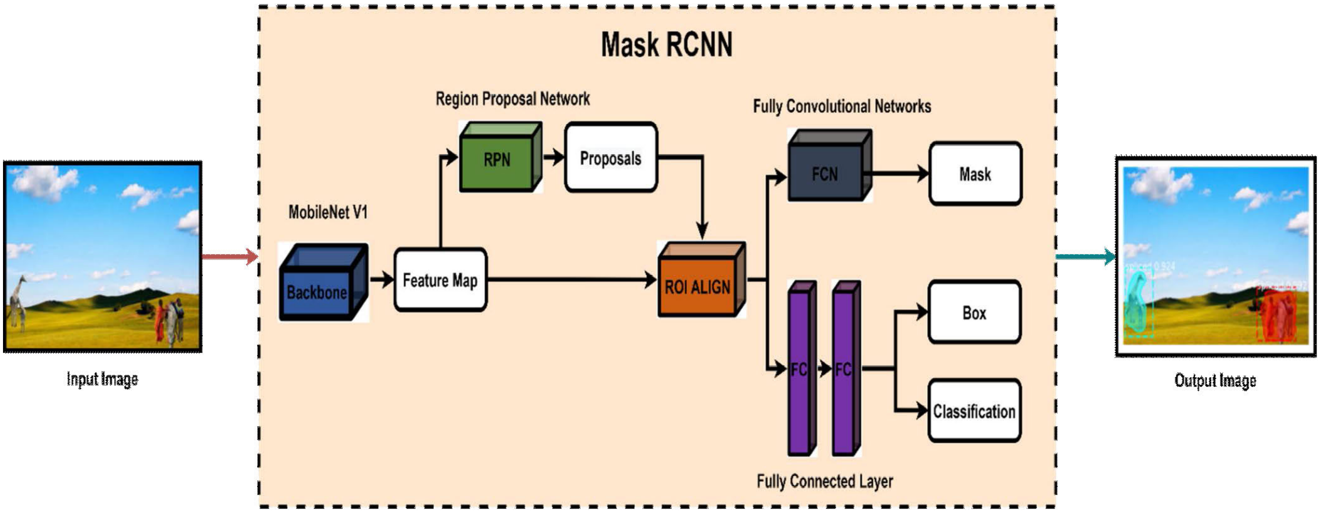


FIGURE 10. Architecture for detection of multiple image splicing.

validation and 64 images for testing, Columbia Color dataset uses 291 images for training and 36 images for validation and 36 images for testing. This experiment uses a total of 2505 images for training and 318 images for testing, and 318 images for validation purposes.

The training images were sized to maintain their aspect ratio. The mask size used is 28 × 28 pixels, and the size of the image is 512 × 512 pixels. This approach is different from the initial Mask R-CNN [19] approach, where the image resizing is done so that 800 pixels are the smallest size and 512 pixels are trimmed to the highest. Bbox selection is made by considering IOU, the ratio of expected bboxes to ground-truth boxes (GT boxes). Mask loss considers only positive ROI and is an intersection of ROI and its ground truth mask. Each mini-batch contains one image per GPU, where each image has an ROI of N samples and a 1:3 plus or minus ratio. The C4 backbone has a value of 64, while FPN has a value of 512. Images of batch size one were passed to the model on a single GPU unit. The model was trained for 360 iterations with an initial learning rate of 0.01, then modified to 0.003 at epoch 120 and 0.001 at epoch 240. Stochastic Gradient Descent (SGD) optimizer is used for optimization, with weight decay fixed to 0.0001 and momentum fixed to 0.9.

VI. DATASET ANNOTATION

Annotation is a process that performs the labelling of an image with a class. The forgery class or keyword vocabulary that was selected for the dataset was "splicing". To associate textual information with the forged regions in the image, all the images have been annotated and categorized. In annotation, the keyword has been associated with all the

TABLE 5. GPU specifications of the training environment.

Parameter	Specification
GPU	Nvidia K80 / T4
GPU Memory	12 GB
GPU Memory Clock	0.82GHz / 1.59GHz
Performance	4.1 TFLOPS / 8.1 TFLOPS
No. CPU Cores	2
RAM	12 GB

TABLE 6. CPU specifications of the training environment.

Parameter	Specification
CPU Model Name	Intel® Xeon®
CPU Freq.	2.30GHZ
CPU Family	Haswell
No. CPU Cores	2
RAM	12 GB

TABLE 7. Configuration parameters of proposed model.

Parameters	Values
BACKBONE	mobilenetv1
IMAGE MAX DIM	512
IMAGE META SIZE	15
IMAGE MIN DIM	800
IMAGE SHAPE	[512 512 3]
LEARNING RATE	0.01
MASK SHAPE	[28,28]
RPN_ANCHOR_SCALES	(8, 16, 32, 64, 128)
STEPS PER EPOCH	50
WEIGHT DECAY	0.0001

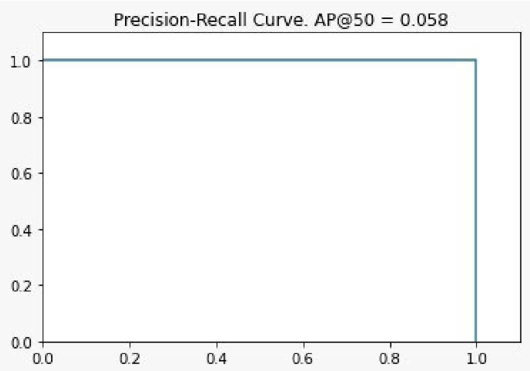


FIGURE 11. Precision-Recall plot on multiple image splicing dataset.

forged regions of the images in the dataset, and to categorize the images a predefined class has been assigned. To produce annotations for the images in JSON format, the VGG Image Annotator [47] was used. Here, the user can click the vertices of a polygon around the forged region and then enters the keyword which describes the type of forgery.

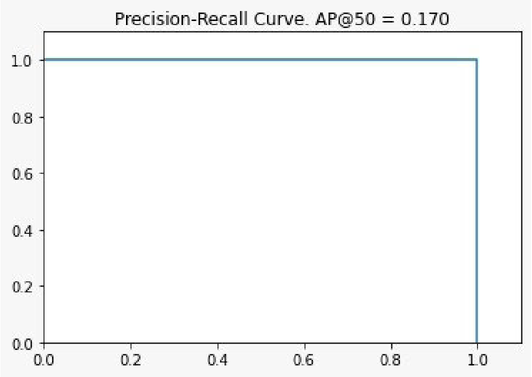


FIGURE 12. Precision-recall plot on CASIA 1.0 dataset.

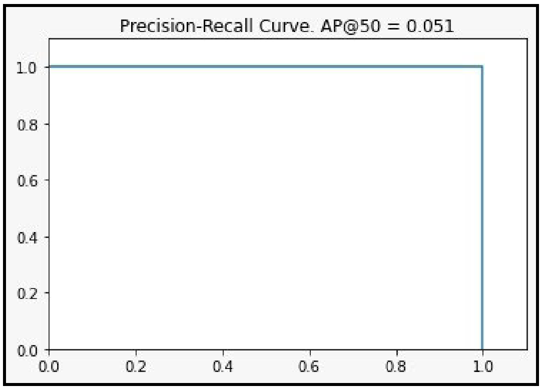


FIGURE 13. Precision-recall plot on WildWeb dataset.

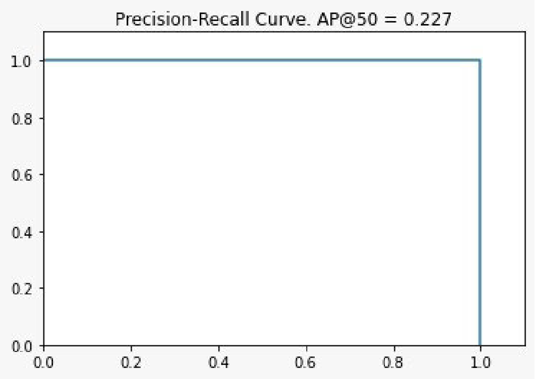
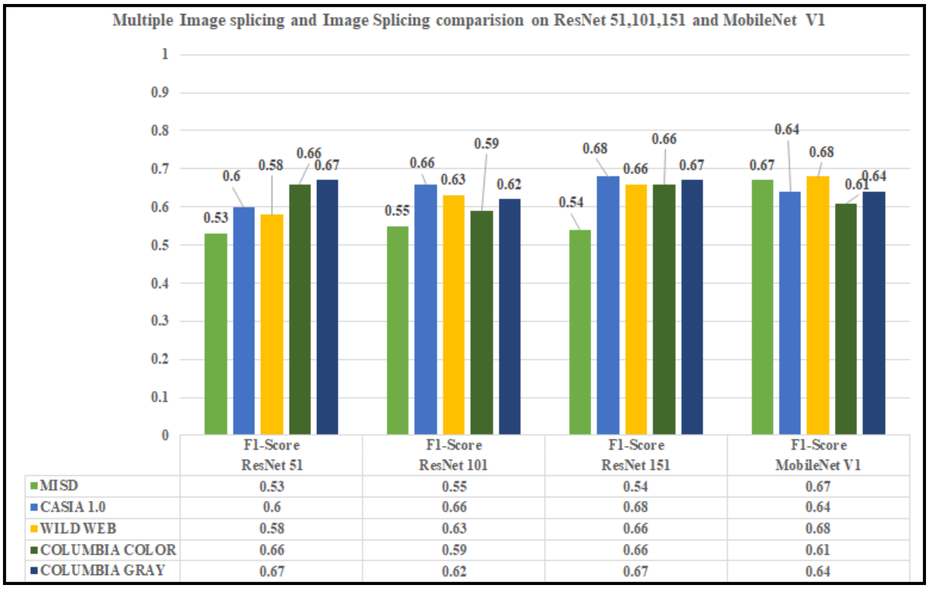


FIGURE 14. Precision-Recall plot on Columbia color dataset.

VII. RESULTS AND DISCUSSION

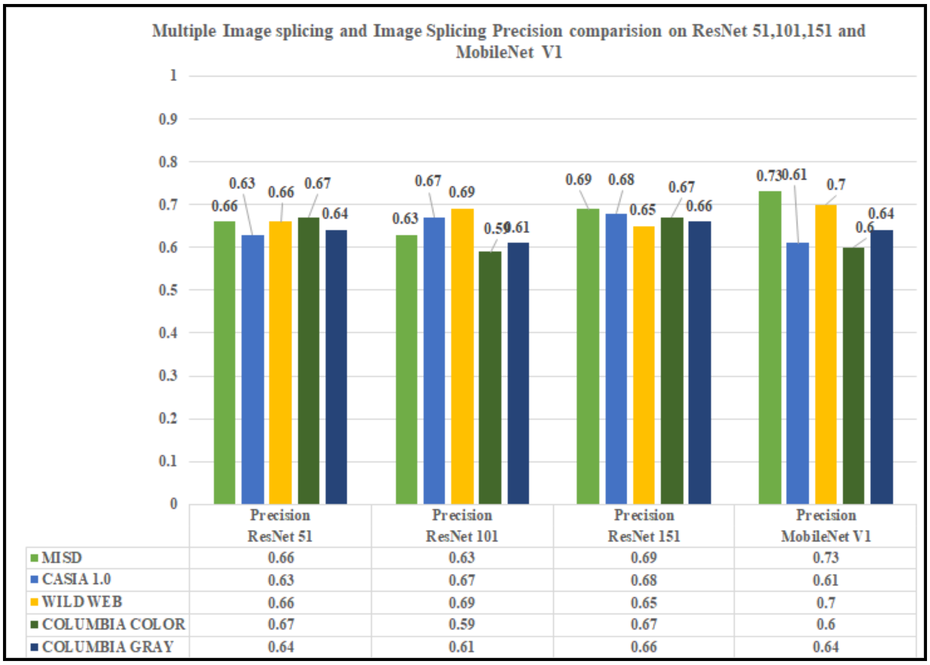
A. PRECISION-RECALL CURVES

The precision-Recall curve for the masks generated by the proposed model on the MISD, CASIA 1.0, WildWeb, and Columbia Gray datasets are shown in Figures 11-14. If the precision stays high as recall increases, the model is considered a good predictive model. The AP values are also shown that is the average across all recall values b/w 0 and 1 at various IOU thresholds. It can be interpreted as the area under the precision-recall curve. Changes in precision and recall are caused by different threshold values. A high recall means a



	F1-Score ResNet 51	F1-Score ResNet 101	F1-Score ResNet 151	F1-Score MobileNet V1
■ MISD	0.53	0.55	0.54	0.67
■ CASIA 1.0	0.6	0.66	0.68	0.64
■ WILD WEB	0.58	0.63	0.66	0.68
■ COLUMBIA COLOR	0.66	0.59	0.66	0.61
■ COLUMBIA GRAY	0.67	0.62	0.67	0.64

FIGURE 15. **Multiple image splicing and image splicing F1-Score comparison on ResNet 51,101,151 and MobileNet V1 using various datasets.**



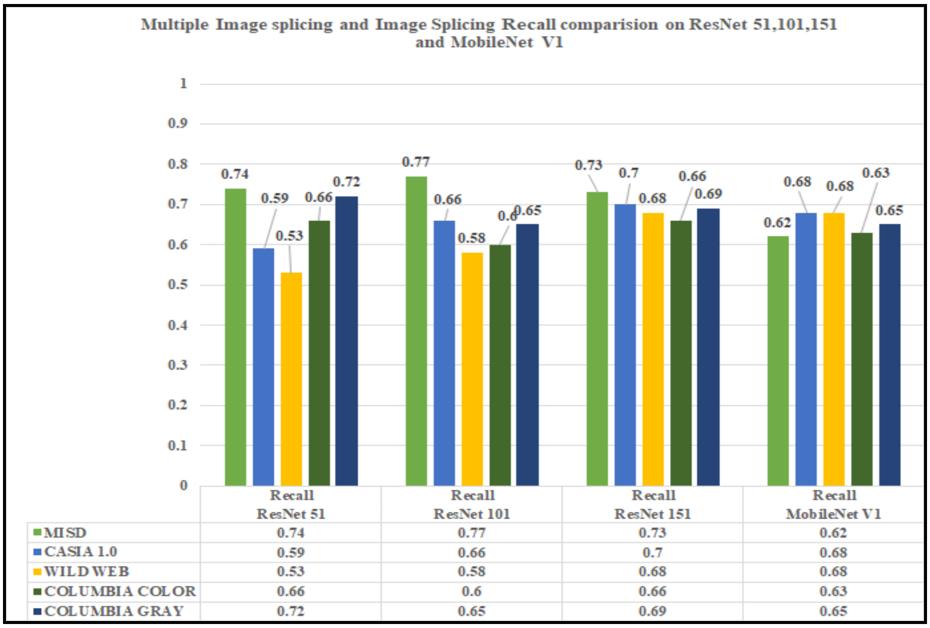
	Precision ResNet 51	Precision ResNet 101	Precision ResNet 151	Precision MobileNet V1
■ MISD	0.66	0.63	0.69	0.73
■ CASIA 1.0	0.63	0.67	0.68	0.61
■ WILD WEB	0.66	0.69	0.65	0.7
■ COLUMBIA COLOR	0.67	0.59	0.67	0.6
■ COLUMBIA GRAY	0.64	0.61	0.66	0.64

FIGURE 16. **Multiple image splicing and image splicing precision comparison on ResNet 51,101,151 and MobileNet V1 using various datasets.**

larger area under the curve with a minimal false +ve rate, which indicates incorrect pixel masking, and a minimal false −ve rate, which indicates the absence of masked pixels that should be present. For each dataset, precision, recall, and AP of each data point i.e., images are calculated and their mean is taken to plot the final PR curve. The value of AP i.e average precision at IOU threshold of 50% is different for all datasets. However, the PR curve of the model on various datasets seems similar as there is a very small difference between the precision-recall values for each dataset and each set of values follows a similar relationship.

B. COMPARISON OF RESULTS WITH MASK R-CNN USING VARIOUS DATASETS AND BACKBONE NETWORKS

This section specifies the results of the proposed model for multiple image splicing forgery detection. Tables 8 and 9 show the F1-Score, Precision, and Recall for ResNet (ResNet 51, ResNet 101, ResNet 151) and MobileNet V1.



	Recall ResNet 51	Recall ResNet 101	Recall ResNet 151	Recall MobileNet V1
🟩 MISD	0.74	0.77	0.73	0.62
🟦 CASIA 1.0	0.59	0.66	0.7	0.68
🟨 WILD WEB	0.53	0.58	0.68	0.68
🟩 COLUMBIA COLOR	0.66	0.6	0.66	0.63
🟦 COLUMBIA GRAY	0.72	0.65	0.69	0.65

FIGURE 17. Multiple image splicing and image splicing recall comparison on ResNet 51,101,151 and MobileNet V1 using various datasets.

From tables, it is observed that the F1-Score of the proposed model i.e. Mask R-CNN model with MobileNet V1 as backbone network outperforms the ResNet models [39] used for image forgery detection with less number of parameters. Figure 15-17 shows F1-Score, Precision, Recall using Mask RCNN with backbone networks such as MobileNet V1 and variants of ResNet (ResNet 51,101 and 151) for detection of multiple image splicing and image splicing on MISD, CASIA 1.0, WildWeb, and Columbia Gray datasets. The model with F1 scores Precision and Recall is represented on the X-axis, while the assessed metrics are represented on the Y-axis. The F1 score is low on different datasets because the proposed model automatically learns and extracts image manipulation characteristics directly from the input images without the help of complex preprocessing steps and hand design features that may add noise, which may interfere with image forgery artifacts. Additionally, our model is designed for detecting a specific kind of forgery, hence it may not have the necessary image clues needed for localization, which, reduces the performance.

Tables 10 and 11 show the AP, $AP_{0.5}$, and $AP_{0.75}$ for ResNet (ResNet 51, ResNet 101, ResNet 151) and MobileNet V1 over MISD dataset for $AP_{0.5}$, and $AP_{0.75}$, IOU is 0.5 and 0.75 respectively. Figure 18-20 shows AP, $AP_{0.5}$, and $AP_{0.75}$ for multiple image splicing and image splicing forgery detection on MISD and other datasets with backbone networks as ResNet (ResNet 51,101,151) and MobileNet V1. From tables, it is observed that the AP of the proposed model i.e. Mask R-CNN model with MobileNet V1 as backbone network outperforms the ResNet models [39] used for image forgery detection with less number of parameters. Here, the X-axis depicts the model with various average precision values, and Y-axis depicts evaluated metrics.

The proposed model produced relatively good results, but to avoid over-fitting K-fold cross-validation is used which is an indication of the true performance of the model. The proposed model was trained using the k-fold cross validation to evaluate the efficiency of the model. The value of K chosen was 5. The training dataset was divided into 5 subsets or folds randomly, and in each step one of the subsets was used as the validation set and the other 4 folds were used as the training set. The performance of the model is the average metric score obtained over the 5 times training. The average metric scores for the 5-fold cross-validation evaluation are shown in the table. During the 5-fold validation, as every image sample gets an opportunity to be a testing sample, unlike randomly picking up the training and testing data, it provided results comparable to the testing results. Tables 12 and 13 show the F1-Score, Precision, and Recall for ResNet (ResNet 51, ResNet 101, ResNet 151) and MobileNet V1 with k-fold cross validation. From tables, it is observed that the F1-Score of the proposed model i.e. Mask R-CNN model with MobileNet V1 as backbone network outperforms the ResNet models with less number of parameters. Figure 11 shows F1-Score, Precision, Recall using Mask RCNN with backbone networks such as MobileNet V1 and variants of ResNet (ResNet 51,101 and 151) for detection of multiple image splicing on MISD dataset. The model with F1 scores Precision and Recall is represented on the X-axis, while the assessed metrics are represented on the Y-axis.

Tables 14 and 15 show the AP, $AP_{0.5}$, and $AP_{0.75}$ for ResNet (ResNet 51, ResNet 101, ResNet 151) and MobileNet V1 over MISD dataset using k-fold cross validation for $AP_{0.5}$, and $AP_{0.75}$, IOU is 0.5 and 0.75 respectively. Figure 12 shows AP, $AP_{0.5}$, and $AP_{0.75}$ for multiple image splicing forgery detection on MISD with backbone networks as

TABLE 8. F1-Score, precision and recall for multiple image splicing and image splicing detection with backbone as ResNet with its variants.

Type of Forgery	Dataset	ResNet 51			ResNet 101			ResNet 151		
		F1-Score	*Precision*	*Recall*	*F1-Score*	*Precision*	*Recall*	*F1-Score*	*Precision*	*Recall*
Multiple Image Splicing	MISD	0.53	0.66	0.74	0.55	0.63	0.77	0.54	0.69	0.73
Image Splicing	CASIA 1.0	0.60	0.63	0.59	0.66	0.67	0.66	0.68	0.68	0.70
	WildWeb	0.58	0.66	0.53	0.63	0.69	0.58	0.66	0.65	0.68
	Columbia Color	0.66	0.67	0.66	0.59	0.59	0.60	0.66	0.67	0.66
	Columbia Gray	0.67	0.64	0.72	0.62	0.61	0.65	0.67	0.66	0.69

TABLE 9. F1-Score, precision, and recall for multiple image splicing detection and image splicing detection with backbone as MobileNet V1.

Type of Forgery	Dataset	MobileNetV1		
		F1-Score	**Precision**	**Recall**
Multiple Image Splicing	MISD	0.67	0.73	0.62
Image Splicing	CASIA 1.0	0.64	0.61	0.68
	WildWeb	0.68	0.70	0.68
	Columbia Color	0.61	0.60	0.63
	Columbia Gray	0.64	0.64	0.65

TABLE 10. Average precision results on multiple image splicing and image splicing detection with backbone as ResNet with its variants.

Type of Forgery	Dataset	ResNet 51			ResNet 101			ResNet 151		
		Avg. Precision	**Avg. Precion0.5 (AP0.5)**	**Avg. Precion 0.75 (AP0.75)**	**Avg. Precision**	**Avg. Precion0.5 (AP0.5)**	**Avg. Precion0.75 (AP0.75)**	**Avg. Precision**	**Avg. Precion0.5 (AP0.5)**	**Avg. Precion0.75 (AP0.75)**
Multiple Image Splicing	MISD	0.86	0.80	0.75	0.83	0.80	0.73	0.78	0.82	0.70
Image Splicing	CASIA 1.0	0.60	0.70	0.63	0.66	0.75	0.65	0.73	0.80	0.70
	WildWeb	0.70	0.83	0.76	0.73	0.88	0.80	0.70	0.80	0.74
	Columbia Color	0.72	0.84	0.76	0.77	0.90	0.82	0.70	0.82	0.78
	Columbia Gray	0.75	0.82	0.79	0.73	0.88	0.84	0.71	0.80	0.73

TABLE 11. Average precision results on multiple image splicing and image splicing detection with backbone as MobileNet V1.

Type of Forgery	Dataset	MobileNetV1		
		Avg. Precision	**Avg. Precion0.5 (AP0.5)**	**Avg. Precion0.75 (AP0.75)**
Multiple Image Splicing	MISD	0.85	0.88	0.73
Image Splicing	CASIA 1.0	0.72	0.76	0.73
	WildWeb	0.75	0.87	0.80
	Columbia Color	0.9	0.92	0.77
	Columbia Gray	0.87	0.89	0.80

ResNet (ResNet 51,101,151) and MobileNet V1 using k-fold cross validation. Here, the X-axis depicts the model with various average precision values, and Y-axis depicts evaluated metrics.

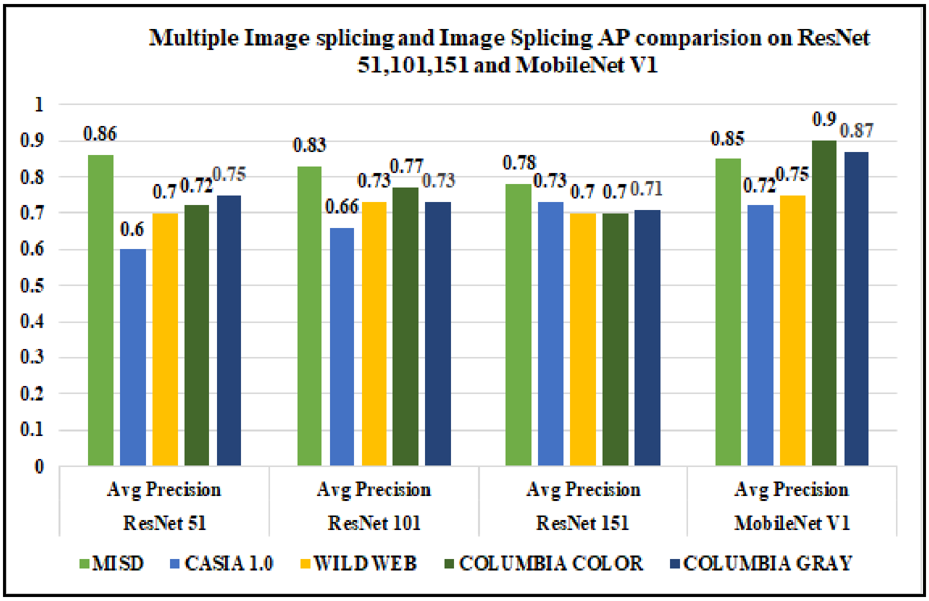


FIGURE 18. Multiple image splicing and image splicing AP comparison on ResNet 51,101,151 and MobileNet V1 using various datasets.

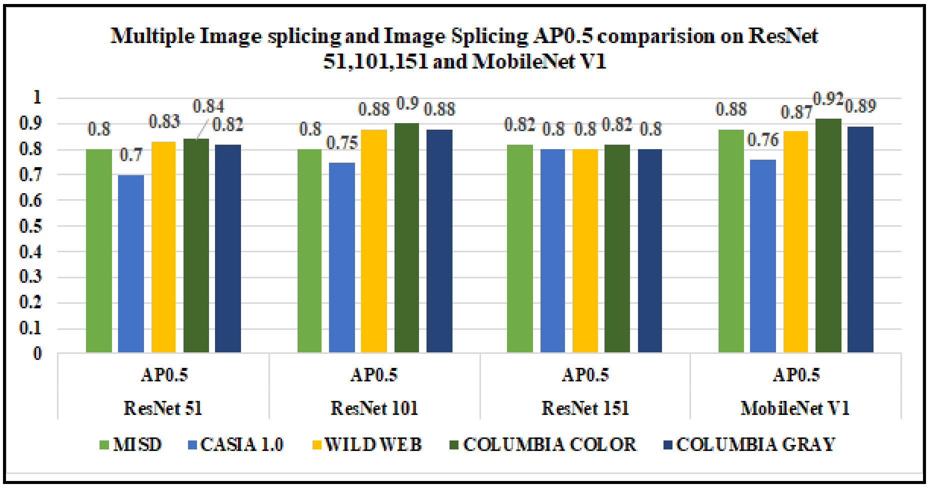


FIGURE 19. Multiple image splicing and image splicing AP0.5 comparison on ResNet 51,101,151 and MobileNet V1 using various datasets.

TABLE 12. F1-Score, precision and recall for multiple image splicing detection with backbone as ResNet with its variants with k-fold cross validation.

Type of Forgery	Dataset	ResNet 50			ResNet 101			ResNet 152		
		F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall
Multiple Image Splicing	MSID	0.55	0.66	0.70	0.55	0.61	0.70	0.51	0.62	0.70

Figure 13 shows the output from the proposed model includes original images, multiple spliced image, a mask for the multiple spliced objects. Table 16 shows that the Resnet has a greater number of parameters as compared to MobileNet. MobileNet uses DSC to reduce the model size (number of parameters) and complexity. A network that has many parameters or weights, can provide a better estimate for a large range of functions. The layer of a network stores the parameters or weights in the main memory. So, the fewer the parameters, the faster the model runs. MobileNet

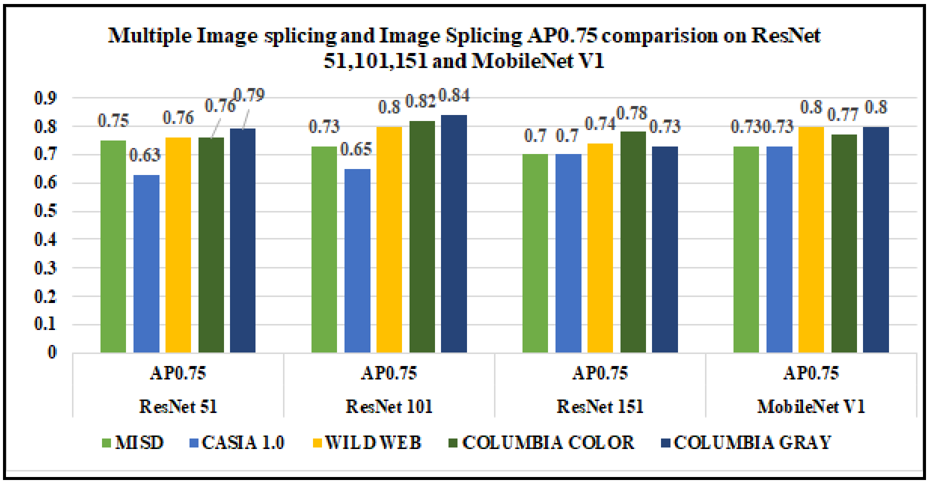


FIGURE 20. Multiple image splicing and image splicing AP0.75 comparison on ResNet 51,101,151 and MobileNet V1 using various datasets.

TABLE 13. F1-Score, precision, and recall for multiple image splicing detection with backbone as MobileNet V1 with k-fold cross validation.

Type of Forgery	Dataset	MobileNetV1		
		F1-Score	Precision	Recall
Multiple Image Splicing	MISD	0.68	0.75	0.62

TABLE 14. Average precision results on multiple image splicing detection with backbone as resnet with its variants with k-fold cross validation.

Type of Forgery	Dataset	ResNet 51			ResNet 101			ResNet 151		
		Avg. Precision	Avg. Precion0.5 (AP0.5)	Avg. Precion 0.75 (AP0.75)	Avg. Precision	Avg. Precion0.5 (AP0.5)	Avg. Precion0. 75 (AP0.75)	Avg. Precisi on	Avg. Precion0. 5 (AP0.5)	Avg. Precion0.75 (AP0.75)
Multiple Image Splicing	MISD	0.63	0.76	0.60	0.63	0.80	0.78	0.68	0.80	0.66

TABLE 15. Average precision results on multiple image splicing detection with backbone as MobileNet V1 variants with k-fold cross validation.

Type of Forgery	Dataset	MobileNetV1		
		Avg. Precision	Avg. Precion0.5 (AP0.5)	Avg. Precion0.75 (AP0.75)
Multiple Image Splicing	MISD	**0.60**	**0.87**	**0.77**

TABLE 16. Parameters, trainable and non-trainable parameters with backbone network as ResNet and its variants and MobileNet V1.

Method	Backbone	Parameters	Trainable	Non-Trainable
Mask R-CNN	ResNet 51	63,733,406	63,621,918	111,488
	ResNet 101	63,733,406	63,621,918	111,488
	ResNet 151	79,446,174	79,288,606	157,568
	MobileNet V1	**23,812,574**	**23,784,542**	**28,032**

offers similar performance as that of Resnet but with a much smaller network due to Depthwise Separable Convolution. In the table 17, TT indicates Training Time in minutes and IT indicates Inference Time in milliseconds. Table 17 shows that MobileNet V1 is the fastest according to the training and inference time. The average per image inference time

TABLE 17. Training time and inference time with backbone network as ResNet and its variants and MobileNet V1 on various datasets.

Model	Number of layers of Model	Datasets									
		MISD		CASIA 1.0		WildWeb		Columbia Gray		Columbia Color	
		TT	IT	TT	IT	TT	IT	TT	IT	TT	IT
ResNet 51	177	435.65	600	449.34	597	510.56	604	450.16	560	400.46	590
ResNet 101	347	514.45	687	520.78	684	620.15	695	564.20	670	504.40	660
ResNet 151	517	604.72	868	632.12	850	740.45	888	695.24	816	615.32	856
MobileNet V1.0	**92**	**265.15**	**434**	**280.10**	**436**	**350.78**	**421**	**302.30**	**430**	**295.20**	**450**

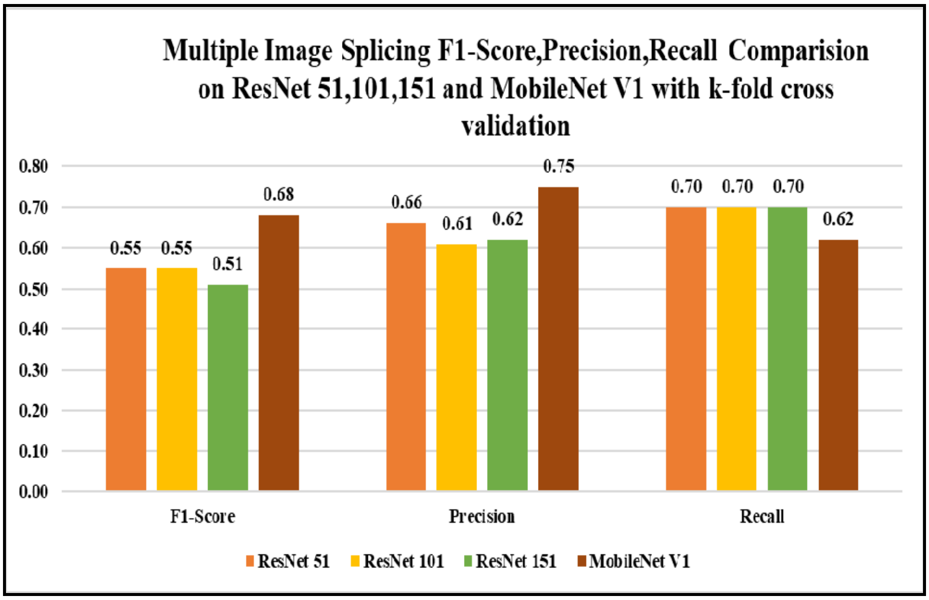


FIGURE 21. Multiple image splicing F1-Score, precision and recall comparison on ResNet 51,101,151 and MobileNet V1 with k-fold cross validation.

in terms of milliseconds over 10 runs for all the backbone models considered are stated in the table for batch size equal to one on Tesla k80. From the table, it is observed that all the backbone models can achieve excellent real-time performance on the Tesla k80 when the batch size of one is considered. MobileNet V1 has a smaller number of trainable parameters and is computationally less complex in terms of parameters space utilization and can make the best use of the available parameters. So, MobileNet V1 shows superior performance in both training and inference speeds. On, the contrary ResNet152 is by far the most expensive architecture-both in terms of computational requirements and number of parameters, that has been tested for this experiment. However, stacking up more layers and using more parameters didn't work in this case, as the model reached an inflection point where the complexity of the model started to outweigh the accuracy gains. Hence, the accuracy started to saturate at a particular inflection point. Overall, by comparing the computational speeds and the accuracy of the models, it seems that there is not a linear relationship between the model computational speeds and the accuracy for this experiment; for instance, the inference and training time of Resnet101 is greater than MobileNet V1, but their accuracy is quite similar.

VIII. LIMITATIONS

A. SIZE OF MISD DATASET

The size of the dataset is an important factor in determining the performance of the deep learning model. The proposed model training is heavily dependent upon the images with various post-processing operations performed on them. A limited number of images is one of the challenge of this research work.

B. ANNOTATION OF DATA

Image annotation is playing an important role in deep learning and machine learning models for image classification segmentation and object recognition. Manual annotation of

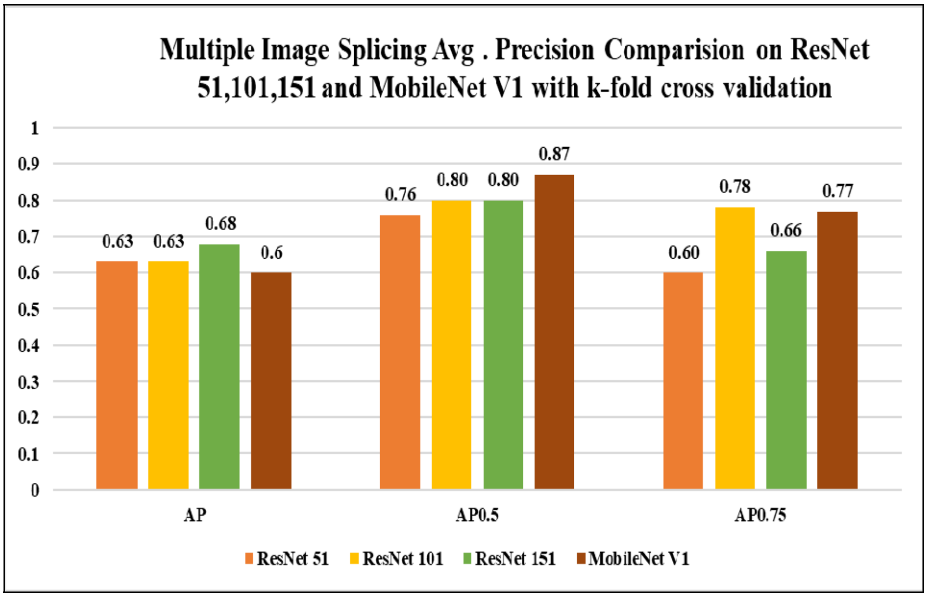


FIGURE 22. Multiple image splicing average precision comparison on ResNet 51,101,151 and MobileNet V1 with k-fold cross validation.

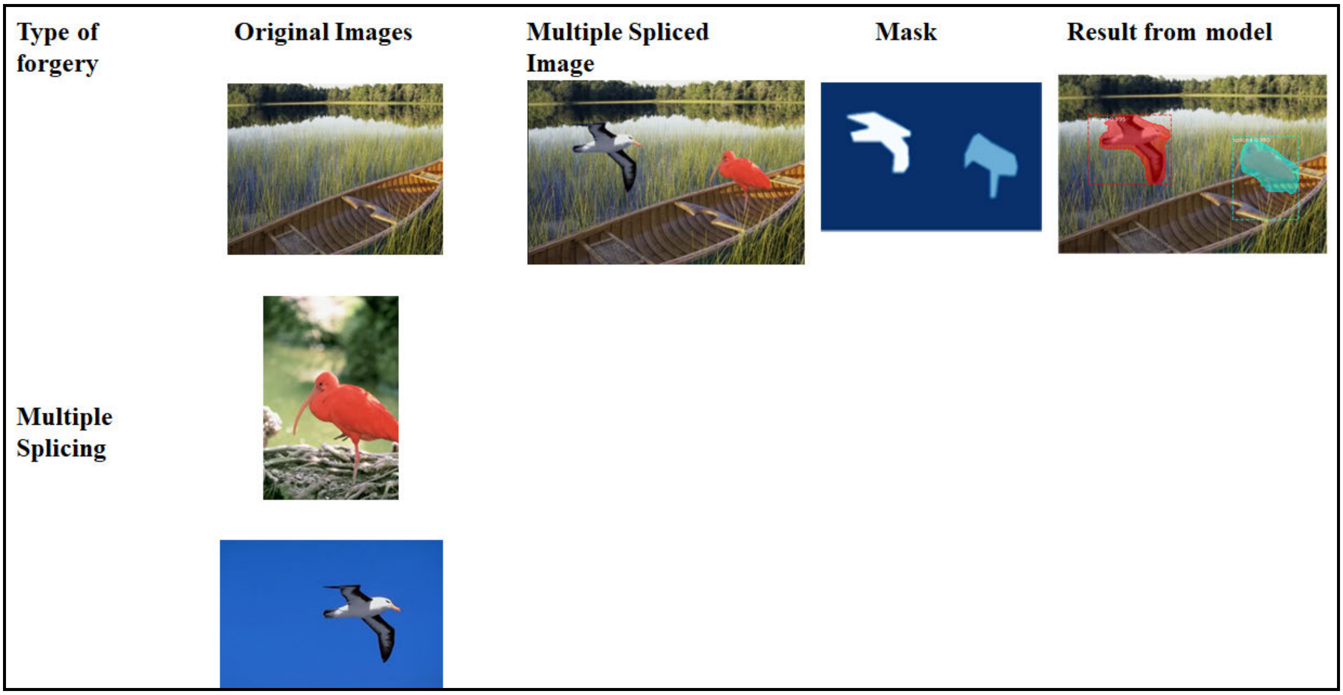


FIGURE 23. Multiple image splicing–original images, multiple spliced image, mask for a multiple spliced object, and result from proposed model.

forged images is reliant on the annotator's knowledge of the data labeling task.

IX. CONCLUSION

This research work presents Mask R-CNN with MobileNet V1 as a lightweight model for the detection of multiple image splicing forgery. It also provides a forged percentage score for multiple spliced images. The model specified in the literature [39] evaluated only one image splicing dataset. However, the proposed model is evaluated on the MISD dataset for multiple image splicing and image splicing on CASIA 1.0, WildWeb, Columbia Gray, and Columbia Color datasets. Also, comparative analysis of the proposed model is done with variants of ResNet such as ResNet 51,101, and 151. The proposed model achieves an average precision of 82% on the Multiple Image Splicing Dataset. The configuration of the proposed model is more efficient in terms of computing than variants of ResNet [39] used for the detection and identification of image splicing forgeries. The evaluation of the proposed model compared to variants of the ResNet [39] network shows that the proposed approach efficiently balanced efficiency and computational costs. The future work focuses on the use of various deep learning architectures such as GAN, MobileNet V2 with Mask-RCNN for detection and localization of multiple image splicing. Currently, the proposed model handle all the attacks/post-processing operations specified by table Image Splicing Dataset. However, in the future, we will try to evaluate our proposed model on more number of attacks and will compare evaluation results with and without attacks.

ABBREVIATIONS

DL	Deep Learning.
CV	Computer Vision.
CNN	Convolutional Neural Network.
FCN	Fully Convolutional Network.
SVM	Support Vector Machine.
RPN	Region Proposal Network.
ROIs	Regions of Interest.
Mask R-CNN	Mask Regional Convolutional Neural Network.
DSC	Depthwise Seperable Convolution.
DSCLs	Depthwise Seperable Convolution Layers.
bbox	bounding box.
NMS	Nonmax suppression.
IOU	Intersection Over Union.
DWT	Discrete wavelet transform.
LBP	Local Binary Pattern.
CT	Contourlet Transform.
HHT	Hilbert-Huang Transform.
DCT	Discrete Cosine Transform.
RP	region proposal.
ILSVRC	ImageNet Large Scale Visual Recognition Challenge.
AU	authentic.
SP	Spliced.
spatial rich model	SRM.

REFERENCES

[1] K. Kadam, S. Ahirrao, and K. Kotecha, "AHP validated literature review of forgery type dependent passive image forgery detection with explainable AI," *Int. J. Elect. Comput. Eng.*, vol. 11, no. 5, pp. 4489–4501, 2021.

[2] S. Walia and K. Kumar, "Digital image forgery detection: A systematic scrutiny," *Austral. J. Forensic Sci.*, vol. 51, no. 5, pp. 488–526, Sep. 2019.

[3] C. Machado, B. Kira, V. Narayanan, B. Kollanyi, and P. N. Howard, "A study of misinformation in WhatsApp groups with a focus on the Brazilian presidential elections," in *Proc. Companion World Wide Web Conf.*, 2019, pp. 1013–1019.

[4] Jessica McDonald. (2020). *Social Media Posts Spread Bogus Coronavirus Conspiracy Theory*. Accessed: Aug. 2, 2021. [Online]. Available: https://www.factcheck.org/2020/01/social-media-posts-spread-bogus-coronavirus-conspiracy-theory/

[5] Z. He, W. Lu, W. Sun, and J. Huang, "Digital image splicing detection based on Markov features in DCT and DWT domain," *Pattern Recognit.*, vol. 45, no. 12, pp. 4292–4299, 2012.

[6] Y. Zhang, C. Zhao, Y. Pi, S. Li, and S. Wang, "Image-splicing forgery detection based on local binary patterns of DCT coefficients," *Secur. Commun. Netw.*, vol. 8, no. 14, pp. 2386–2395, Sep. 2015.

[7] Q. Zhang, W. Lu, and J. Weng, "Joint image splicing detection in DCT and Contourlet transform domain," *J. Vis. Commun. Image Represent.*, vol. 40, pp. 449–458, Oct. 2016.

[8] X. Li, T. Jing, and X. Li, "Image splicing detection based on moment features and Hilbert-Huang transform," in *Proc. IEEE Int. Conf. Inf. Theory Inf. Secur.*, Dec. 2010, pp. 1127–1130, doi: 10.1109/ICITIS.2010.5689754.

[9] K. B. Meena and V. Tyagi, "A deep learning based method for image splicing detection," in *Proc. J. Phys., Conf.*, 2021, vol. 1714, no. 1, Art. no. 012038.

[10] K. B. Meena and V. Tyagi, "A deep learning based method to discriminate between photorealistic computer generated images and photographic images," in *Advances in Computing and Data Sciences* (Communications in Computer and Information Science), vol. 1244. 2020, pp. 212–223.

[11] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.

[12] B. Ahmed, T. A. Gulliver, and S. Alzahir, "Image splicing detection using mask-RCNN," *Signal, Image Video Process.*, vol. 14, no. 5, pp. 1035–1042, Jul. 2020.

[13] N. Y. Hussien, R. O. Mahmoud, and H. H. Zayed, "Deep learning on digital image splicing detection using CFA artifacts," *Int. J. Sociotechnol. Knowl. Develop.*, vol. 12, no. 2, pp. 31–44, Apr. 2020.

[14] J. Wang, Q. Ni, G. Liu, X. Luo, and S. K. Jha, "Image splicing detection based on convolutional neural network with weight combination strategy," *J. Inf. Secur. Appl.*, vol. 54, Oct. 2020, Art. no. 102523.

[15] E. I. Abd El-Latif, A. Taha, and H. H. Zayed, "A passive approach for detecting image splicing based on deep learning and wavelet transform," *Arabian J. Sci. Eng.*, vol. 45, no. 4, pp. 3379–3386, Apr. 2020.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[17] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[20] T. F. Gonzalez, *Handbook of Approximation Algorithms and Metaheuristics*. Boca Raton, FL, USA: CRC Press, 2007.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, and D. Anguelov, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[23] Z. Zhang, Y. Zhou, J. Kang, and Y. Ren, "Study of image splicing detection," in *Proc. 4th Int. Conf. Intell. Comput. (ICIC)*, Shanghai, China, Sep. 2008, pp. 1103–1110.

[24] S. Bahrampour, N. Ramakrishnan, L. Schott, and M. Shah, "Comparative study of deep learning software frameworks," 2016, *arXiv:1511.06435*.

[25] S.-H. Tsang. (2018). *Review: MobileNetV1—Depthwise Separable Convolution (Light Weight Model)*. Accessed: Aug. 2, 2021. [Online]. Available: https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69

[26] L. Sifre and S. Mallat, "Rigid-motion scattering for texture classification," 2014, *arXiv:1403.1687*.

[27] T. T. Ng and S. Chang, "A data set of authentic and spliced image blocks," Columbia Univ., New York, NY, USA, Tech. Rep. 203-2004-3, 2004, pp. 1–9.

[28] J. Evans and K. Straub, *Adobe Photoshop Lightroom Classic CC Classroom in a Book*. Boston, MA, USA: Pearson Education, 2018.

[29] *Calphotos*. University of California, Berkeley. Accessed: Aug. 2, 2021. [Online]. Available: https://calphotos.berkeley.edu/

[30] Y.-F. Hsu and S.-F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2006, pp. 549–552, doi: 10.1109/ICME.2006.262447.

[31] J. Dong, W. Wang, and T. Tan, "CASIA image tampering detection evaluation database," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process.*, Jul. 2013, pp. 422–426, doi: 10.1109/ChinaSIP.2013.6625374.

[32] T. J. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. Rocha, "Exposing digital image forgeries by illumination color classification," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 7, pp. 1182–1194, Jul. 2013, doi: 10.1109/TIFS.2013.2265677.

[33] M. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris, "Detecting image splicing in the wild (web)," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jun. 2015, pp. 1–6, doi: 10.1109/ICMEW.2015.7169839.

[34] A. L. Gokhale, S. D. Thepade, N. R. Aarons, D. Pramod, and R. Kulkarni, "AbhAS: A novel realistic image splicing forensics dataset," *J. Appl. Secur. Res.*, vol. 15, no. 4, pp. 1–23, 2020.

[35] J. Bunk, J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, B. S. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. Peterson, "Detection and localization of image forgeries using resampling features and deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1881–1889, doi: 10.1109/CVPRW.2017.235.

[36] B. Liu and C.-M. Pun, "Locating splicing forgery by fully convolutional networks and conditional random field," *Signal Process., Image Commun.*, vol. 66, pp. 103–112, Aug. 2018.

[37] R. Salloum, Y. Ren, and C.-C. J. Kuo, "Image splicing localization using a multi-task fully convolutional network (MFCN)," *J. Vis. Commun. Image Represent.*, vol. 51, pp. 201–209, Feb. 2018.

[38] E. R. Bartusiak, S. K. Yarlagadda, D. Guera, P. Bestagini, S. Tubaro, F. M. Zhu, and E. J. Delp, "Splicing detection and localization in satellite imagery using conditional GANs," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Mar. 2019, pp. 91–96.

[39] X. Wang, H. Wang, S. Niu, and J. Zhang, "Detection and localization of image forgeries using improved mask regional convolutional neural network," *Math. Biosci. Eng.*, vol. 16, no. 5, pp. 4581–4593, 2019.

[40] N. Jindal, "Copy move and splicing forgery detection using deep convolution neural network, and semantic segmentation," *Multimedia Tools Appl.*, vol. 80, no. 3, pp. 3571–3599, Jan. 2021.

[41] S. Nath and R. Naskar, "Automated image splicing detection using deep CNN-learned features and ANN-based classifier," *Signal, Image Video Process.*, vol. 15, no. 7, pp. 1601–1608, Oct. 2021.

[42] G. Schaefer and M. Stich, "UCID: An uncompressed color image database," *Proc. SPIE*, vol. 5307, pp. 472–480, Dec. 2003.

[43] G. B. Duc Tien Dang Nguyen, C. Pasquini, and V. Conotter, "RAISE: A raw images dataset for digital image forensics," in *Proc. 6th ACM Multimedia Syst. Conf.*, 2015, pp. 219–224.

[44] K. D. Kadam, S. Ahirrao, and K. Kotecha, "Multiple image splicing dataset (MISD): A dataset for multiple splicing," *Data*, vol. 6, no. 10, p. 102, Sep. 2021.

[45] FIgma. *Learn Design with FIgma*. Accessed: Aug. 2, 2021. [Online]. Available: https://www.figma.com/resources/learn-design/getting-started/

[46] *Upload an Image to Remove the Background*. Accessed: Aug. 2, 2021. [Online]. Available: https://www.remove.bg/upload

[47] A. G. Abhishek and A. Z. Dutta. *VGG Image Annotator*. Accessed: Aug. 2, 2021. [Online]. Available: https://www.robots.ox.ac.U.K./~vgg/software/via/

[48] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Lecture Notes in Computer Science), vol. 8693. Cham, Switzerland: Springer, 2014, pp. 740–755.

KALYANI KADAM is currently pursuing the Ph.D. degree from the Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune. She is also working as an Assistant Professor with the Symbiosis Institute of Technology, Pune. Her research interests include machine learning and deep learning.

SWATI AHIRRAO received the Ph.D. degree from Symbiosis International (Deemed University), Lavale, Pune, Maharashtra, India. She is currently working as Associate Professor with SIT, Pune. Her research interests include big data analytics, machine learning, deep learning, natural language processing, and reinforcement learning.

KETAN KOTECHA received the M.Tech. and Ph.D. degrees from IIT Bombay. He is currently the Head of the Symbiosis Centre for Applied AI (SCAAI), the Director of the Symbiosis Institute of Technology, and the Dean of the Faculty of Engineering, Symbiosis International (Deemed University). He has expertise and experience in cutting-edge research and projects in AI and deep learning for the last 25 years (more than).

SAYAN SAHU is currently pursuing the B.Tech. degree in computer science with the Symbiosis Institute of Technology, Symbiosis International (Deemed) University, Pune. His research interests include machine learning, deep learning, and computer vision.