MDPI

Multiple Image Splicing Dataset (MISD): A Dataset for Multiple Splicing

Kalyani Dhananjay Kadam [1], Swati Ahirrao [1],* and Ketan Kotecha [2],*

1   Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, India; kalyanik@sitpune.edu.in
2   Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International (Deemed University), Pune 412115, India
*   Correspondence: swatia@sitpune.edu.in (S.A.); head@scaai.siu.edu.in (K.K.)

Abstract: Image forgery has grown in popularity due to easy access to abundant image editing software. These forged images are so devious that it is impossible to predict with the naked eye. Such images are used to spread misleading information in society with the help of various social media platforms such as Facebook, Twitter, etc. Hence, there is an urgent need for effective forgery detection techniques. In order to validate the credibility of these techniques, publically available and more credible standard datasets are required. A few datasets are available for image splicing, such as Columbia, Carvalho, and CASIA V1.0. However, these datasets are employed for the detection of image splicing. There are also a few custom datasets available such as Modified CASIA, AbhAS, which are also employed for the detection of image splicing forgeries. A study of existing datasets used for the detection of image splicing reveals that they are limited to only image splicing and do not contain multiple spliced images. This research work presents a Multiple Image Splicing Dataset, which consists of a total of 300 multiple spliced images. We are the pioneer in developing the first publicly available Multiple Image Splicing Dataset containing high-quality, annotated, realistic multiple spliced images. In addition, we are providing a ground truth mask for these images. This dataset will open up opportunities for researchers working in this significant area.

1. Summary

With the advent of advanced image-editing software in the 21st century, the time, cost, and efforts required to tamper with images have been drastically affected. These images are utilized for communication and have a social impact. As an example, during the Brazilian presidential election, around 13.1% of WhatsApp posts were fake [1]. Cybersecurity researchers [2] found that hackers can access 3-D medical scans of patients and can append or remove images of malignant tumors. According to a recent survey, radiologists were misled by scans that had been manipulated with AI techniques, potentially resulting in a significant risk of misdiagnosis and insurance fraud. It has been observed that in online terrorist communications, specific kinds of images are likely to be reused, and in some cases forged, and spread across social media platforms [3–6]. Developing convincing, realistic fake images has become considerably easier than in the past because of the prevalence of user-friendly image editing softwares and advances in Deep Learning technologies like Generative Adversarial Networks (GAN) [7].

Therefore, there is an urgent need for efficacious Forgery Detection techniques [8–10]. To validate these techniques, standard datasets are required. The existing literature on

Image Forgery Detection employed standard datasets. However, these are popularly used for detection of image splicing forgeries. Similarly, custom datasets are also used for detection of image splicing forgeries. This research work presents the development of a Multiple Image Splicing Dataset used to detect multiple image splicing forgery. The dataset is constructed by collecting various images from the CASIA V1.0 image splicing dataset and combining various images from CASIA V1.0 to create a single image. This dataset includes the collection of realistic natural color images. It contains authentic as well as multiple spliced images. There are 618 authentic images of JPG format and 300 multiple spliced images in JPG format with a dimension of 384 × 256 pixels. The authentic images are taken from the CASIA V1.0 dataset [11]. The authentic images are divided into eight categories: scene, animal, architecture, character, plant, art, nature, indoor, and texture. The spliced images are constructed using Figma Software by performing various manipulation operations. Table 1 shows the specifications of the Multiple Image Splicing Dataset.

Table 1. Specifications of Multiple Image Splicing Dataset.

Subject Area	Computer Vision
More precise area	Image forgery detection, Image forensics
File type	JPG
Image Size (in pixels)	384 × 256
Number of classes in images	8 (scene, texture, architecture, character, plant, art, nature, indoor and animal)
Authentic Images	618
Spliced Images	300
Total ground truth masks	300

Figure 1 shows the process of construction of the Multiple Image Splicing Dataset.
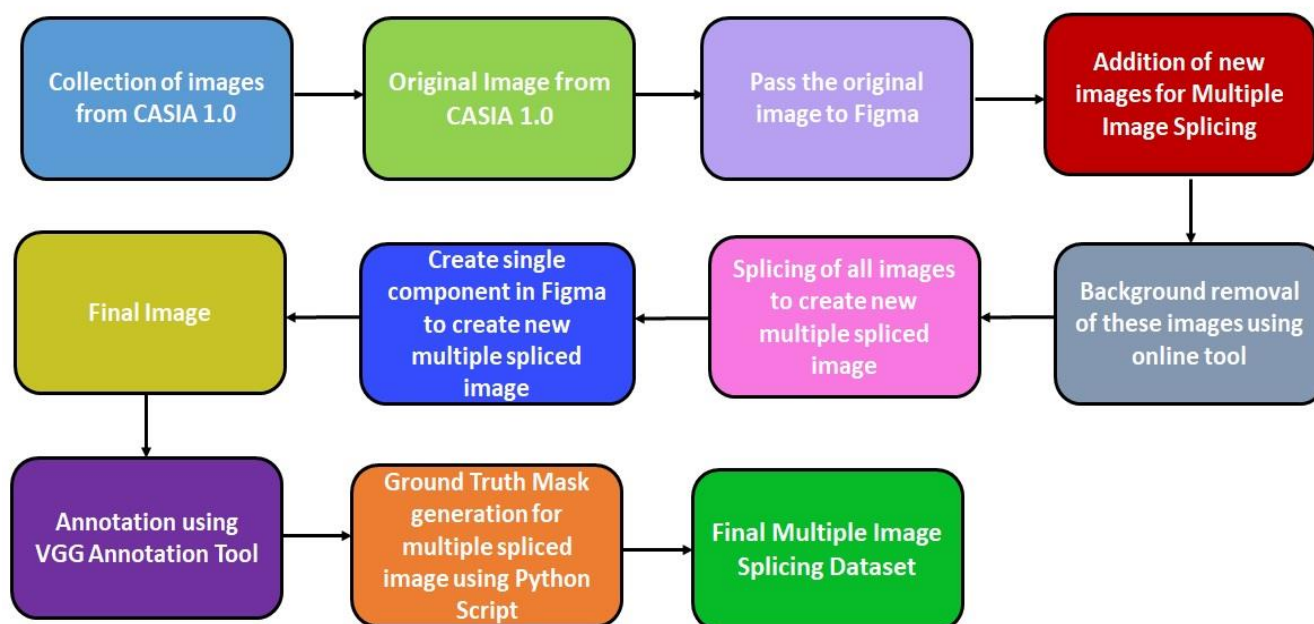
Figure 1. Multiple Image Splicing Dataset Creation Process.

The major contribution of this research work are as follows:

- Detailed process flow for the creation of a Multiple Image Splicing Dataset.
- Development of a Multiple Image Splicing Dataset which contains both authentic and realistic spliced images.
- Comparative analysis of existing dataset with the Multiple Image Splicing Dataset.
- To provide a ground truth mask for multiple spliced images.

Goal of MISD

- To provide high-quality, realistic multiple spliced images by applying various post-processing operations such as rotation, and scaling to the researchers working in the computer vision area.
- To provide generalized multiple spliced images from various categories so that the researchers can use a Deep Learning model which will be trained on these generalized images. Such a model can be used for the detection of multiple spliced forgeries in various domains such as forensics, courts, military, and healthcare, where the images are used for communication.
- To provide the ground truth masks for the multiple spliced images for effortless identification of spliced objects in the multiple spliced images.

2. Related Work

This section discusses the dataset and techniques used for the detection of image splicing forgery.

2.1. Datasets

Existing work on image splicing forgery detection uses standard and custom datasets that contain spliced images constructed from two images. It lacks multiple spliced images in standard and custom datasets.

There are several publicly available datasets for detection of image splicing forgery. The datasets are Columbia Gray [12], Columbia Color [13], CASIA 1.0 and 2.0 [11], DSO-1 [14], DSI-1 [14], Wild Web [15], and AbhAS [16].

The Columbia Gray [12] dataset contains image blocks of gray color that are extracted from the CalPhotos dataset [17]; spliced images in this dataset are constructed using a basic cut-paste operation without applying any post-processing operations. The Columbia Gray dataset does not contain ground truth masks or color images. The images in the dataset are not realistic and can be easily recognized. It does not contain multiple spliced images.

The Columbia Gray dataset's limitations are addressed by the Columbia Uncompressed Image Splicing Detection Evaluation dataset (CUISDE) [13]. The authentic images in this dataset were taken with cameras like the canong3, nikond70, canonxt, and kodakdcs330. Authentic images are edited using Adobe Photoshop to construct the spliced images of this dataset. The images in the authentic category contain images of indoor and outdoor scenes. The CUISDE dataset contains images of limited categories and also considers only two images for creating the spliced images in the dataset.

CASIA 1.0 and 2.0 [11] have been extensively employed in image splicing detection studies. However, this dataset includes both copy-move and spliced images. The limitation of this dataset is that it does not provide ground truth masks. It also does not contain multiple spliced images.

DSO -1 and DSI-1 [14] dataset is the facial image composites dataset. The dataset was also elected for the Forensic Challenge competition sponsored by the IEEE Information Forensics and Security Technical Committee (IFS-TC) [18]. The limitation of DSO-1 and DSI-1 is that it contains only facial image splicing. It does not provide multiple spliced images.

The images in the WildWeb [15] dataset were retrieved from the internet and these images have undergone compression before being uploaded to various Internet websites. The limitation of this dataset is that the images in this dataset are clearly recognizable as spliced and ground truth masks are not provided for some of the spliced images.

The AbhAS [16] dataset contains few images which are captured by the camera as well as from internet sources. The limitation of this dataset is that it does not contain facial splices. Few splicing images in this dataset are easily recognizable, and splicing images are constructed by combining only two images.

A study of existing standard and custom datasets reveals that splicing images are constructed by combining only two images and do not contain multiple spliced images.

The details of the image splicing datasets are shown in Table 2.

Table 2. Image Splicing Datasets.

Sr. No	Name of Dataset Used in Image Splicing	Total Number of Images including Authentic and Spliced Images	Dimension of Image in Pixel	Image Format	Ground Truth Mask
1	**Columbia Gray** [12]	1845 image blocks (Authentic—933; Spliced—912)	128 × 128	BMP	Not Available
2	**CUISDE** [13]	363 (Authentic—183; Spliced—180)	757 × 568 to 1152 × 768	TIFF	Available
3	**CASIA 1.0** [11]	1725 (Authentic—800; Spliced—925)	384 × 256	JPG and TIFF	Not Available
4	**CASIA 2.0** [11]	12,614 (Authentic—7491; Spliced—1849, and remaining images are of type copy move)	320 × 240 and 800 × 600	JPG and TIFF	Not Available
5	**DSO-1** [14]	200 (Authentic—100; Spliced—100)	2048 × 1536 and 1536 × 2048	JPG and TIFF	Available
6	**DSI-1** [14]	100 (Authentic—25; Splice—25)	Different sizes	PNG	Available
7	**WildWeb** [15]	10,666 (Authentic—100; Spliced—9666)	122 × 120 to 2560 × 1600	PNG	Available
8	**AbhAS** [16]	93 (Authentic—45; Spliced—48)	278 × 181 to 3216 × 4288	JPG	Available

2.2. Deep Learning Techniques for Image Splicing Detection

This section describes various deep learning techniques used for image splicing detection. Table 3 shows the deep learning techniques for the detection of image splicing.

Table 3. Deep Learning Techniques for image splicing forgery detection.

Paper Reference	Deep Learning Technique Used for Image Splicing Forgery Detection	Dataset Used for the Experiments
[19]	CNN	CASIA V1.0 [11], CASIA V2.0 [11], Columbia Gray [12]
[20]	Deep Neural Network	CUISDE [13]
[21]	Auto Encoder Decoder, LSTM	NIST'16, IEEE Forensics Challenge Dataset, MS-COCO [22]
[23]	Mask R-CNN	Columbia Gray [12]
[24]	Mask R-CNN with backbone network as ResNet-conv	Computer-generated dataset where forged images have been generated using COCO [22] 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.
[25]	CNN	CASIA V1.0 [11], CASIA V2.0 [11]
[26]	CNN, Dense classifier network	CASIA V2.0 [11]

This research work [19] utilized CNN to learn the hierarchical features of the input image. The weights of the first layer of the model were initialized to the value of the basic high-pass filter, then these weights were used for the calculation of residual maps in the SRM. After that, the patch descriptor was used to capture dense features from the test images using the pre-trained CNN. Then the final discriminative features for SVM classification were generated using a feature fusion method. In [20], a deep neural network was utilized in association with a conditional random field to detect the spliced region in an image without the need for any prior data. To improve the accuracy, three different FCN variations were used. In this work, small altered objects were difficult to find because downsampling reduced the dimensions of the real image, making smaller objects much harder to spot. The future work of this research includes handling the overfitting issue and optimization of the network.

In [21], patches were used to partition the network image. These patches were then used to extract resampling features. The Hilbert curve determines the order in which patches were

delivered to LSTM. The correlation between patches was detected by the sampling features. The encoder was used to figure out a spatial location of a modified region. The spatial features from the encoder and the out features from LSTM were combined to interpret the kind of manipulation in an image. The decoder provided a finer representation of the spatial map, revealing the image's altered region. An improved Mask R-CNN [23] along with the Sobel filter was used for image splicing detection. The Sobel filter uses edge information to identify the modified boundaries and pixel-wise information is used to train the model.

A solution based on the ResNet-Conv deep neural network was proposed in [24]. To create an initial feature map from the RGB image, two versions of ResNet-Conv (ResNet-50 and ResNet-101) were used. Mask-RCNN was also proposed by the authors for locating a forgery. Another research work [25] uses CNN with the weight combination approach. To distinguish splicing manipulation, the proposed technique used three types of features: YCbCr features, edge features, and photo response non-uniformity (PRNU) features, which were merged according to weight by the combination strategy. These weight parameters were automatically modified during the CNN training process, unlike the other approaches, until the best ratio was attained. Experiments demonstrated that the suggested technique was more accurate than existing CNN-based methods and the proposed method's CNN depth was substantially lower than the other methods.

A CNN-based technique was proposed in [26] for automated feature engineering. The feature vector was then loaded into a dense classifier network to assess if an image is authentic or spliced. On CASIA v2.0, the proposed model was trained, validated, and finally tested. This model classified the image as spliced or not, but it could not locate spliced regions of an image. Another research work [27] used Mask R-CNN with MobileNet V1 as a backbone network for multiple image splicing detection. This model was trained and tested on the MISD dataset (Supplementary Materials). To overcome the overfitting issue, k-fold cross-validation was used for performing the experiments.

3. Data Description

Our MISD is the first publicly available Multiple Image Splicing Dataset. It consists of 918 images classified as Authentic or Spliced. Table 2 shows 618 Authenticate images of type JPG and 300 Multiple Spliced JPG images.

4. Methods

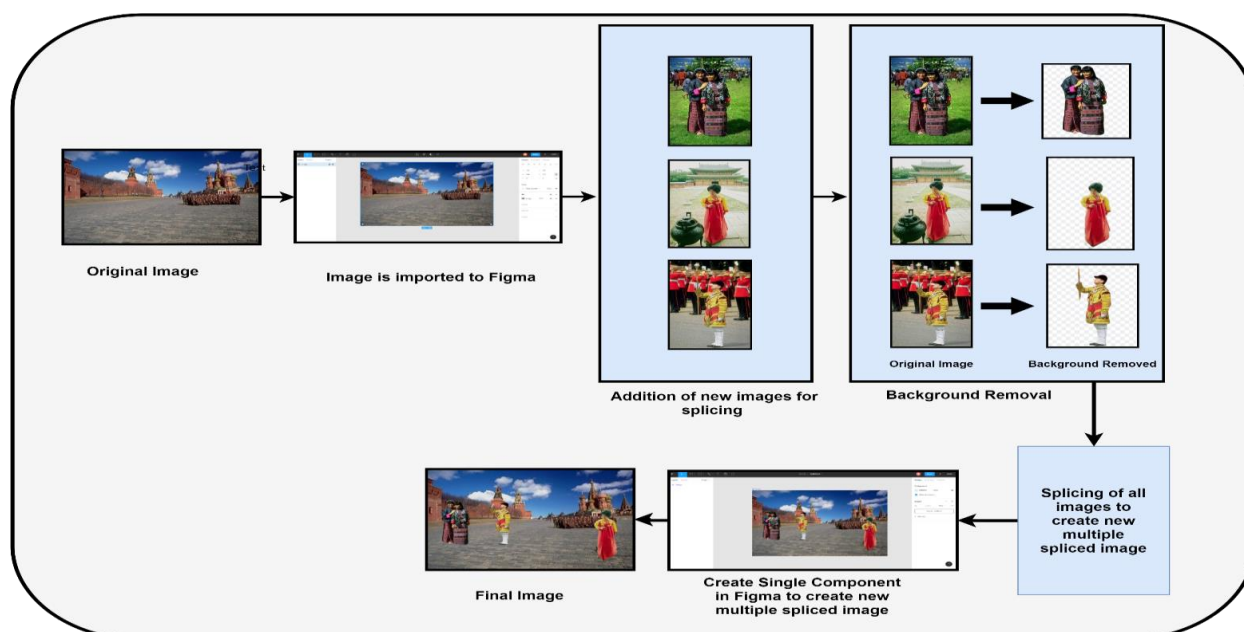Figure 2 shows the complete process for the construction of MISD.

Figure 2. Multiple Image Splicing Dataset Creation.

4.1. Data Acquisition

The authentic images were collected from CASIA 1.0 standard dataset. We extracted 618 authenticate images to construct 300 multiple spliced images. The authentic images extracted are from various categories, including scene, animal, architecture, character, plant, art, nature, indoor, and texture, to develop the generalized dataset. The number of authentic images under each category are: animal—167, architecture—35, art—76, character—124, indoor—7, nature—53, plant—50, scene—74, and texture—32. Table 4 shows the dataset description.

Table 4. Dataset Description.

	Data	Number of Images Per Category		Image Size	Type of Image	Total Number of Images
Authenticate Images	Image	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	Image	Images of all categories	300	384×256	JPG	300

Image manipulation requires a photo editing tool. We used Figma software for the construction of the MISD dataset. Figma software is preferred over to Adobe Photoshop as Adobe Photoshop requires more processing time and is expensive due to its subscription charges. Various image manipulation operations in Figma are useful for development of the MISD dataset. The spliced images in our dataset were constructed using Figma Software [28] by performing various manipulation operations on the authentic images. Figma is a cloud-based vector graphics editor and interface designer with robust features for creating spliced images. It is mainly a browser-based application, but there are desktop versions available for MacOS and Windows. It is the only prototyping tool that works on any operating system. In addition, it has vector tools for proficient illustrations and code generation. Figma is used mainly for UI/UX designing and prototyping. However, it can also be used for image manipulation. Figma supports image imports in the following formats: PNG, JPEG, GIF, TIFF, or WEBP. Figma allowed us to resize, crop, adjust colors, and apply filters like contrast, shadows, mirror, blur, exposure, highlight, etc., to the images.

The spliced images were constructed by considering the following criteria:

1. Spliced images were constructed using images from the authenticating images. The spliced region(s) come/comes from two or more authentic images or the same authentic image. The shape of spliced regions in the Figma Software can be customized

2. Spliced regions were created by cropping the regions from the original images. Then, these regions were modified using various operations such as rotation, scaling, and other manipulation operations.

3. Before pasting, the cropped image region(s) can be modified using scaling, rotation, or other manipulation operations to create a spliced image.

4. Various spliced regions such as small, medium, and large were considered for creating multiple spliced images. Furthermore, in the authentic images, there are various texture images. Therefore, we generated spliced texture images using one or two authentic texture images. As a result, we created a batch of spliced texture images by arbitrarily cropping a portion of a texture image (in a regular or arbitrary shape) and pasting it to the same or a different texture image.

Steps to construct the multiple spliced image are as follow:

1. First, an authentic image is imported into Figma [28]. This image serves as the base for other objects to be added to it.
2. To cut the objects from the other authenticate images, a background removal tool like removing bg [29] is used for cutting the objects. This tool is used for the instant removal of the background of the images. An image with a clear high contrast distinction between the image's subject and background is preferred to get the best possible results. After background removal, the resulting image is inserted on the base image in Figma. Then, various manipulation operations are applied to the inserted objects such as the transformation, rotation, color, and scaling, to make the spliced images look more authentic and difficult to detect.
3. Finally, all the inserted objects and the base image are selected and merge as a single component, and exported as a single image.
4. This process is repeated using several other authenticate images, and multiple objects are inserted on the base image.

Sample images from this dataset are shown in Figure 3. In this dataset, the filename of multiple spliced image indicates the number of images used to create a single multiple spliced image. Table 5 shows the naming convention of authentic image and multiple spliced image.
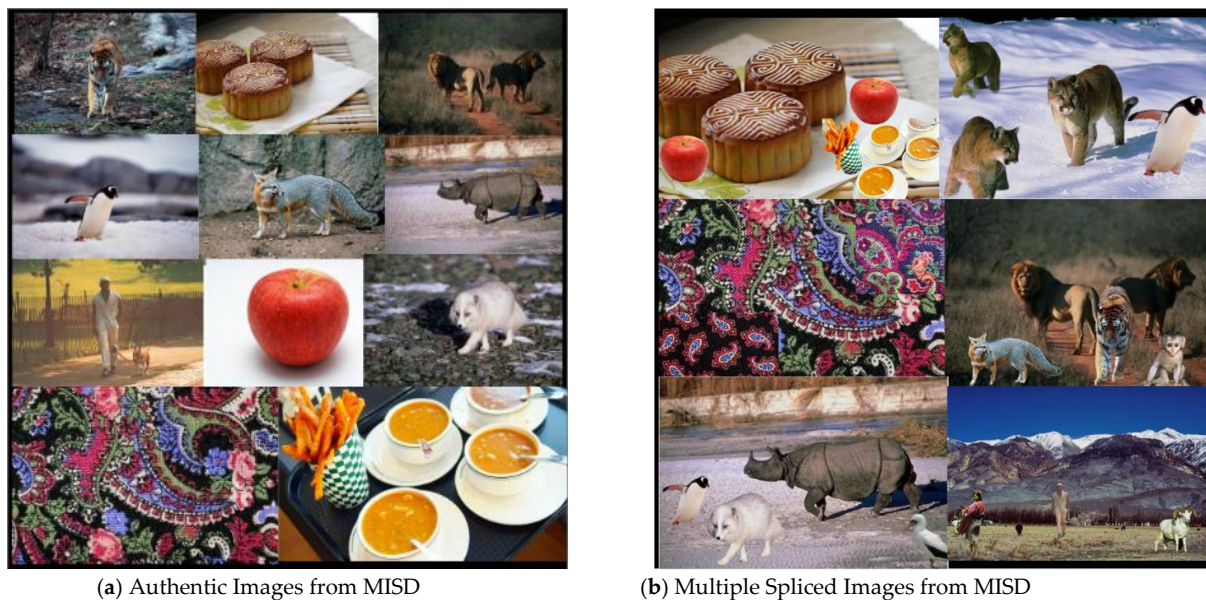

(**a**) Authentic Images from MISD   (**b**) Multiple Spliced Images from MISD

Figure 3. Sample Images for MISD.

Table 5. The naming convention for authentic and multiple spliced images [11].

Naming Convention of Authentic Image	Naming Convention of Multiple Spliced Image
Au_ani_00088.jpg where, ● Au: Authentic ● ani: animal category ● 00088: Authentic image ID	Sp_D_nat_30165_ani_00058_art_30625_ani_10192_82.png Sp: Splicing D: Different (means the region was copied from the different images) nat_30165: the source image ani_00058: the target image art_30625: the target image ani_10192: the target image 82: Multiple Spliced Image ID Here nat, ani, and art represents the category of the image
Other categories are: ● nat: nature ● arc: architecture ● art: art ● cha: characters ● ind: indoor ● pla: plants ● sec:scene ● xt: texture	

Statistical Summary of MISD

Table 6 shows a statistical summary of our MISD.

Table 6. Statistical Summary of MISD.

Size of Images	37 KB Minimum 60 KB Average and 129 KB Maximum
Image Quality	200–300 dpi
Softwares used during MISD construction	● Figma ● Background Remover (https://www.remove.bg/ (accessed on 13 July 2021)), ● VGG Annotation Tool, ● Python script for generating ground truth mask
Number of labels of the dataset	2
Number of images for each label	Authentic—618
	Spliced—300
Average Ground truth mask per image	3
Percentage of images in each category	Animal 27%
	Architecture 5.7%
	Art 12%
	Character 20.2%
	Indoor 1.1%
	Nature 8.6%
	Plant 8.1%
	Scene 12%
	Texture 5.3%

4.2. Data Annotation

Image annotation is crucial in developing Machine Learning and Deep Learning models for object detection, classification, and segmentation [30,31]. We manually annotated the Multiple Spliced Images using the VGG image annotation tool [32] shown in Figure 4. To annotate and label the images, we needed to develop a cost-effective and high-quality data annotation methodology. The annotation process consisted of three main tasks:

i.   Labelling the spliced object in the image
ii.   Locating and marking all the labelled objects
iii.   Segmenting each of the spliced instances
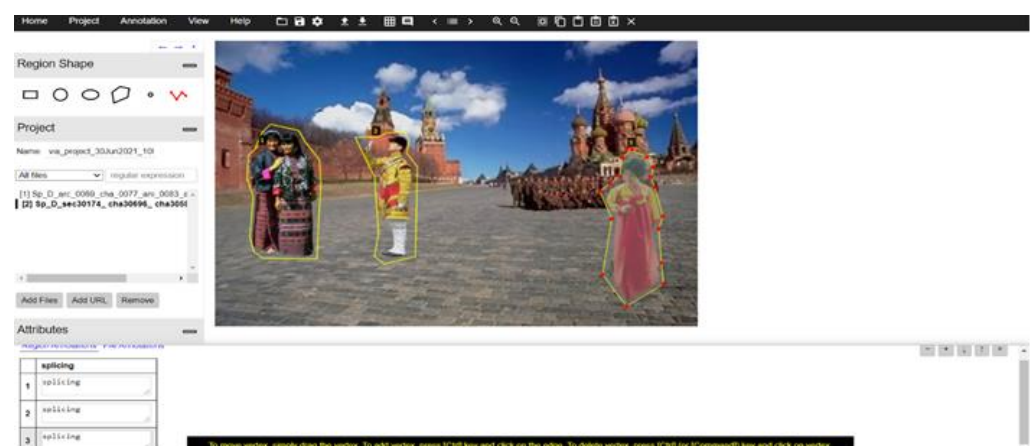
Figure 4. Annotation of the image using VGG annotation tool.

To annotate multiple spliced images, the Polygon Annotation method was employed. Based on the concept that the objects do not always have a rectangular shape, the Polygon method was employed instead of rectangles to specify the shape and location of an object with greater precision. VGG is a simple and standalone image, audio, and video annotation software. This annotation tool is a web-based program that requires no installation or configuration and is less than 400 KB in size. It is an open-source project based on HTML, JavaScript, and CSS. There are few annotation formats available, e.g., COCO, Pascal VOC, and YOLO. COCO [22] is the most popular object detection dataset at present, due to its simplicity. It is commonly used to evaluate the performance of computer vision algorithms. Several annotation types are available in COCO for object detection, keypoint detection, segmentation, and image captioning. In COCO, JSON format is used to save the annotations. The structure of the JSON file is given below:

i.   Filename: Contains the name of the image file
ii.   Size: Contains the size of the image in pixels
iii.   Regions: Contains the x and y coordinates for the vertices of polygon around any object instance for the segmentation masks and the form of the bounding box (rectangle, polygon)
iv.   Region_attributes: Contains the category of the object
v.   File_attributes: Contains other information about the image

Any model when provided with inaccurate, imperfect, and wrong labels will not be able to learn the features well, and the results will be far from good. Thus, it is important for us to create high-quality ground truth labels for our dataset. To aid the annotation process, the annotations were created by using the zoom-in functionality of VGG to maintain the efficiency of handling tricky curve boundaries and small spliced objects.

4.3. Ground Truth Mask Generation Using Python Script

This newly developed dataset contains ground truth masks for all multiple spliced images. In building any AI/ML network, huge volumes of training data are required. Along with the training dataset, a ground truth dataset is needed most of the times. A ground truth dataset is just like a normal dataset, but with annotations present in it. Training data must be properly categorized and labelled. For example, trying to identify image forgery in an image, the ground truth dataset would have images of forged objects along with the bounding boxes or binary masks where the regions have been forged in the image, as well as the labels that indicate any information about the forgery in the box. Another use of ground truth images is for reducing the time for computing the region of interest (ROI). Using ground truth boxes, the Intersection over union (IoU) for all the predicted regions can be computed using the equation:

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} \tag{1}$$

To dissociate the segmentation evaluation and the accuracy of the detection, we standardized the segmentation quality. If a model predicts the segmentation mask of a spliced object, it is important to find out how well the predicted segmentation mask matches our ground truth image. As a standard for correct detection, we imposed the basic requirement that only if the IoU for a particular region is greater or equal to 0.5 is it considered as a region of interest, otherwise it is discarded.

We have programmatically created the ground truth masks for each of the spliced instances using the annotations created as, in our dataset, splicing is concerned over multiple areas of the forged image. Thus, for each spliced instance in the image, we created multiple different ground masks to reflect them separately. The ground truth mask for images in the dataset were generated using a Python script with the help of a JSON file. First, the script reads the JSON file, extracts the polygon coordinates for each mask object, and generates the masks. As the JSON file does not include the size of the masks, the

script reads each of the image files to get the height and width of the mask. The mask is converted into an image using zeroes. The polygon is drawn on the image using the polygon coordinates. Finally, the Spliced object in the image is represented using white color with a black background. For each image in the directory, the script makes a new directory with the name of the image, and this directory contains sub-directories of both the authenticate image and generated mask files. Figure 5 gives the example of ground truth mask for the multiple spliced images.
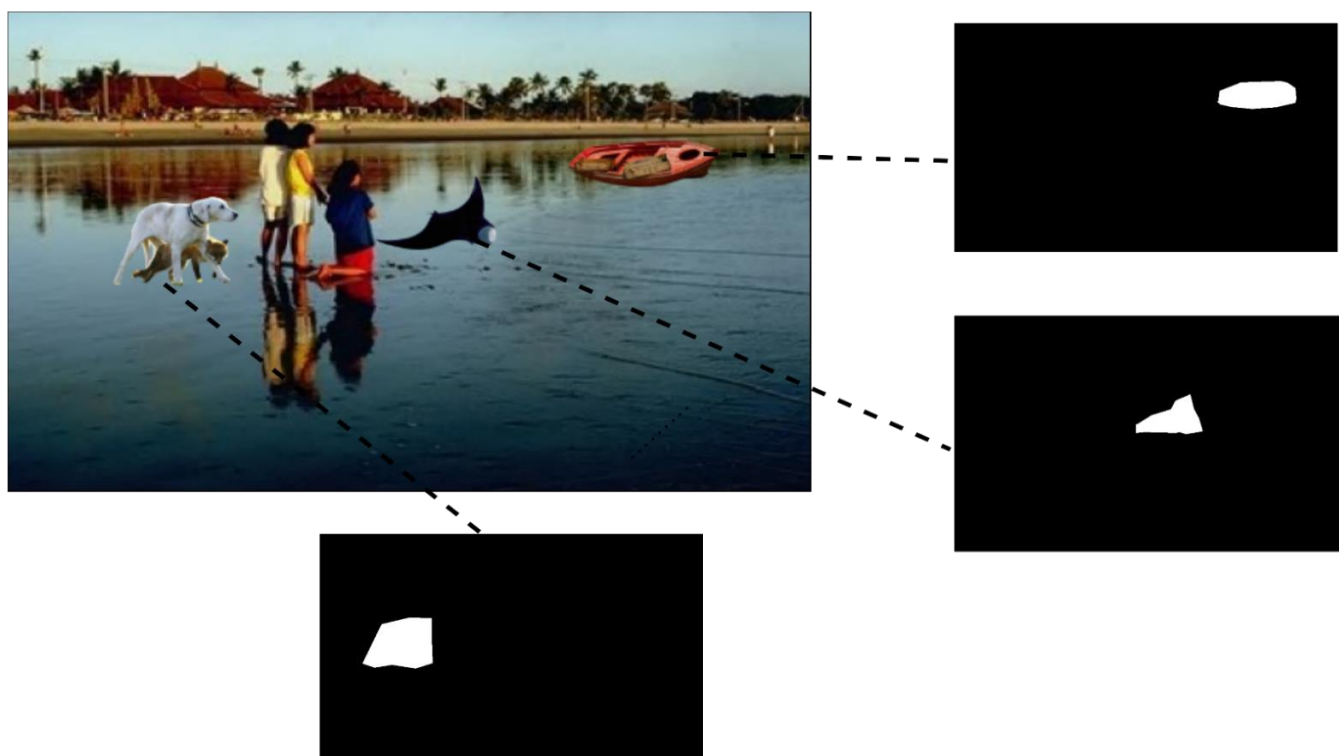
Figure 5. Ground truth masks of multiple spliced image.

5. Conclusions

This research work presents a Multiple Image Splicing Dataset consisting of 618 authentic and 300 realistic multiple spliced images by applying various processing operations such as rotation and scaling. It also contains generalized images from various categories such as animal, architecture, art, scene, nature, plant, texture, character, and indoor scene. This generalized dataset will be used for building a deep learning model for the detection of multiple image splicing forgeries. Ground truth masks are also provided in our dataset for the multiple image spliced images. This newly developed dataset is the first publicly available realistic Multiple Image Splicing Dataset to the best of our knowledge. This research work also presents the statistical summary of the proposed Multiple Image Splicing Dataset. It highlights the process flow for the creation of the Multiple Image Splicing Dataset. This dataset opens up opportunities for researchers working in the area of computer vision.

6. Future Work

1. Size of MISD—Our goal is to increase the size of the MISD. A few more categories can be included in the future to construct a versatile dataset.
2. Pretrained Neural Networks—Pretrained neural networks, such as ResNet or MobileNet, can be used to evaluate the performance of the MISD.
3. Addition of more post-processing operations—A few more post-processing operations can be applied to the images in the MISD.

References

1. Machado, C.; Kira, B.; Narayanan, V.; Kollanyi, B.; Howard, P. A Study of Misinformation in Whats App groups with a focus on the Brazilian Presidential Elections. In Proceedings of the Companion proceedings of the 2019 World Wide Web conference, San Francisco, CA, USA, 13 May 2019; pp. 1013–1019.
2. THE WEEK. Available online: https://www.theweek.in/news/health/2019/04/04/Researchers-hack-CT-scansto-create-fake-cancers-in-imaging.html (accessed on 13 July 2021).
3. O'Halloran, K.L.; Tan, S.; Wignell, P.; Lange, R. Multimodal Recontextualisations of Images in Violent Extremist Discourse. In *Advancing Multimodal and Critical Discourse Studies: Interdisciplinary Research Inspired by Theo van Leeuwen's Social Semiotics*; Zhao, S., Djonov, E., Björkvall, A., Boeriis, M., Eds.; Routledge: New York, NY, USA; London, UK, 2017; pp. 181–202.
4. Tan, S.; O'Halloran, K.L.; Wignell, P.; Chai, K.; Lange, R. A Multimodal Mixed Methods Approach for Examining Recontextualition Patterns of Violent Extremist Images in Online Media. *Discourse Context Media* **2018**, *21*, 18–35. [CrossRef]
5. Wignell, P.; Tan, S.; O'Halloran, K.L.; Lange, R.; Chai, K.; Lange, R.; Wiebrands, M. Images as Ideology in Terrorist-Related Communications. In *Image-Centric Practices in the Contemporary Media Sphere*; Stöckl, H., Caple, H., Pflaeging, J., Eds.; Routledge: New York, NY, USA; London, UK, 2019; pp. 93–112.
6. Students Expose Fake Social Media Reports on Israel Gaza to Set Record Straight. Available online: https://www.timesofisrael.com/students-expose-fake-social-media-reports-on-israel-gaza-to-set-record-straight/ (accessed on 13 July 2021).
7. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
8. Farid, H. Image forgery detection a survey. *IEEE Signal Process. Mag.* **2009**, *26*, 16–25. [CrossRef]
9. Kadam, K.; Ahirrao, S.; Kotecha, K. AHP validated literature review of forgery type dependent passive image forgery detection with explainable AI. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 5. [CrossRef]
10. Walia, S.; Kumar, K. Digital image forgery detection: A systematic scrutiny. *Aust. J. Forensic Sci.* **2019**, *51*, 488–526. [CrossRef]
11. Dong, J.; Wang, W.; Tan, T. CASIA Image Tampering Detection Evaluation Database. In Proceedings of the 2013 IEEE China Summit and International Conference on Signal and Information Processing, Beijing, China, 6–10 July 2013. [CrossRef]
12. Ng, T.T.; Chang, S.F.; Sun, Q. *A Data Set of Authentic and Spliced Image Blocks*; ADVENT Technical Report; Columbia University: New York, NY, USA, 2004; pp. 203–2004.
13. Hsu, Y.F.; Chang, S.F. Detecting Image Splicing Using Geometry Invariants and Camera Characteristics Consistency. In Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, 9–12 July 2006.
14. De Carvalho, T.J.; Riess, C.; Angelopoulou, E.; Pedrini, H.; de Rezende Rocha, A. Exposing digital image forgeries by illumination color classification. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1182–1194. [CrossRef]
15. Zampoglou, M.; Papadopoulos, S.; Kompatsiaris, Y. Detecting image splicing in the wild (web). In Proceedings of the 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Turin, Italy, 29 June–3 July 2015.
16. Gokhale, A.L.; Thepade, S.D.; Aarons, N.R.; Pramod, D.; Kulkarni, R. AbhAS: A Novel Realistic Image Splicing Forensics Dataset. *J. Appl. Secur. Res.* **2020**, 1–23. [CrossRef]

17. CalPhotos. Available online: https://calphotos.berkeley.edu/ (accessed on 13 July 2021).
18. IEEE. IFS-TC Image Forensics Challenge. Retrieved 25 May 2020. 2013. Available online: https://signalprocessingsociety.org/newsletter/2013/06/ifs-tc-image-forensics-challenge (accessed on 13 July 2021).
19. Rao, Y.; Ni, J. A deep learning approach to detection of splicing and copy-move forgeries in images. In Proceedings of the 2016 IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi, United Arab Emirates, 4–7 December 2016; pp. 1–6. [CrossRef]
20. Liu, B.; Pun, C.M. Locating splicing forgery by fully convolutional networks and conditional random field. *Signal Process Image Commun.* **2018**, *66*, 103–112. [CrossRef]
21. Bappy, J.H.; Simons, C.; Nataraj, L.; Manjunath, B.S.; Roy-Chowdhury, A.K. Hybrid LSTM and Encoder-Decoder Architecture for Detection of Image Forgeries. *IEEE Trans. Image Process.* **2019**, *28*, 3286–3300. [CrossRef] [PubMed]
22. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014. [CrossRef]
23. Wang, X.; Wang, H.; Niu, S.; Zhang, J. Detection and localization of image forgeries using improved mask regional convolutional neural network. *Math. Biosci. Eng.* **2019**, *16*, 4581–4593. [CrossRef] [PubMed]
24. Ahmed, B.; Gulliver, T.A.; alZahir, S. Image splicing detection using mask-RCNN. *Signal Image Video Process* **2020**, *14*, 1035–1042. [CrossRef]
25. Wang, J.; Ni, Q.; Liu, G.; Luo, X.; Jha, S.K. Image splicing detection based on convolutional neural network with weight combination strategy. *J. Inf. Secur. Appl.* **2020**, *54*, 102523. [CrossRef]
26. Nath, S.; Naskar, R. Automated image splicing detection using deep CNN-learned features and ANN-based classifier. *Signal Image Video Process.* **2021**, *15*, 1601–1608. [CrossRef]
27. Kadam, K.; Ahirrao, D.; Kotecha, D.; Sahu, S. Detection and Localization of Multiple Image Splicing Using MobileNet V1. *arXiv* **2021**, arXiv:2108.09674.
28. Learn Design with Figma. Available online: https://www.figma.com/resources/learn-design/ (accessed on 26 February 2021).
29. Remove Background from Image—Remove.bg. Available online: https://www.remove.bg/ (accessed on 26 February 2021).
30. Image Data Labelling and Annotation—Everything You Need to Know. Available online: https://www.towardsdatascience.com/image-data-labelling-and-annotation-everything-you-need-to-know-86ede6c684b1/ (accessed on 26 February 2021).
31. What Is Image Annotation? Available online: https://medium.com/supahands-techblog/what-is-image-annotation-caf4107601b7 (accessed on 26 February 2021).
32. VGG Image Annotator. Available online: https://www.robots.ox.ac.uk/~{}vgg/software/via/ (accessed on 26 February 2021).