Article | Received 28 May 2023; Accepted 20 July 2023; Published 11 December 2023
https://doi.org/10.55092/pcs2023020022

Video forgery detection via deep learning semantic segmentation architecture

Muhammad Faiz Misman*, Azurah A Samah, Hairudin Abdul Majid, Zuraini Ali Shah, Zalmiyah Zakaria, Mohd Murtadha Mohamed and Siti Zaiton Hashim

Artificial Intelligence and Bioinformatics Group (AIBIG), Faculty of Computing, Universiti Teknologi Malaysia, Johor, Malaysia.

* Correspondence author; E-mail: faizmisman@gmail.com.

Abstract: Video forgery has recently emerged as a global problem due to the development of sophisticated and user-friendly video modification tools and software. This study introduces an end-to-end deep learning architecture for detecting the fabricated object in a video. The recent advancements in deep learning for semantic segmentation of images and videos served as inspiration for this architecture. To distinguish fake objects from background images, this research suggested a semantic segmentation technique. The suggested architecture, which combines the U-net and VGG19 architectures based on Convolutional Neural Networks (ConvNet), is capable of differentiating between a forged object and its background, even though the model was trained on a small sample size of data and decreased the number of channels in every network layer, which reduced the computational complexity of the suggested approach without compromising performance. On 10 videos, the chroma-key composition and splicing forgery methods were used to assess how well the proposed architecture performed. In lieu of traditional classification metrics, mean intersection over union (mIoU) was used to evaluate the performance of the proposed method. According to the experiment, the training and validation sets for the proposed method both scored 0.9343 for mIoU accuracy, which is the highest.

Keywords: video forgery; semantic segmentation; convolutional neural network; VGG19; U-net

1. Introduction

In a variety of fields, including the military, medical imaging, surveillance systems, law enforcement, criminal investigations, and many more fields, digital video has been extensively used as historical records and supporting evidence. But with simple and user-friendly video editing programs like Apple Final Cut Pro, Adobe After Effect, and Adobe Premiere Pro, altering a video sequence is now a simple task that only requires a few minutes

of work and no specialized high-end knowledge [1,2]. Therefore, developing a reliable method for detecting video forgery presents a significant challenge for digital forensic research.

The paper is structured as follows: In Section 2, works in video semantic segmentation and video forgery detection will be explained briefly. The proposed architecture will follow Section 3's description of the U-net and VGG19 architectures. In Section 4, the experimental apparatus will be described. Section 5 will discuss the experimental results, followed by Section 6's summary and future directions.

2. Related study on video semantic segmentation and forgery detection

In this section, research that is related to this paper will be covered. It is worth mentioning that, based on the literature, there is no similar research in image and video forgery detection using semantic segmentation via deep learning. This section instead focused on the application of deep learning to semantic segmentation and video forgery detection.

2.1. Segmentation of images and videos from a semantic perspective

A technique for classifying images includes semantic segmentation. Semantic segmentation categorises each pixel within the image into its assigned class, in contrast to conventional image classification, which considers the entire image as a designated class. In areas of computer vision like medical imaging, autonomous vehicles, robotics, and many more, this new method opens up a new path towards complete scene understanding because it provides comprehensive information on every image or video [3]. Conventional machine learning techniques like Support Vector Machines (SVM) [4,5], Random Forests [6,7], and Markov Random Fields [8] were widely used before deep learning methods. However, in order to improve accuracy, these traditional machine learning approaches heavily rely on hand-crafted domain knowledge and post-processing steps [9].

Researchers were motivated to evaluate ConvNet's ability to solve the semantic segmentation problem by leveraging the benefits of deep neural networks (or deep learning), specifically on ConvNet [10]. Fully convolutional neural networks (FCN) were initially used for semantic segmentation in 2015 by [11], who supplanted the fully connected layers with a deconvolutional layer to obtain feature maps as opposed to classification results. By winning the PASCAL VOC 2015 [12] challenge, FCN established a basis for the use of deep learning in semantic segmentation for years to come, outperforming traditional methods. FCN, however, has limitations when working with high-resolution and unstructured data because the deconvolutional process lacks global context information [3].

Researchers then improve the FCN by utilising cutting-edge methods like encoder-decoder architecture. The same number of layers separate the encoding and decoding parts of the encoder-decoder architecture found in SegNet [13] and U-net [14], which uses max-pooling to encode feature maps. By adding element-wise feature maps from the convolution process to the corresponding deconvolution feature maps, the encoder-decoder architecture takes global context information into account during the deconvolution feature maps. There are a few alternatives to the encoder-decoder architecture, such as the Conditional Random

Field (CRF) application as a post-processing step, which has been suggested by the FC-RCCN [15], DeepLab [16], and improved by the CRFasRNN [17]. In order to improve the FCN semantic segmentation problem, dilated convolutions are another option. This method uses a generalised Kronecker-factored convolutional filter with an exponentially dilated rate l, as opposed to a conventional convolutional filter where l=1 [18]. Examples of this method include DeepLab (improvised version) [19] and Enet [20]. One of the suggested improvements is multi-scale CNN, which combines multiple CNN prediction models created from various input sizes to produce a single output [21]. The works of ParseNet [22] and SharpMask [23], which combine multiple features from different layers (within the same network) and combine them into one feature before passing them to the next layer or classifier [3], bring about another observable improvement. Processing video is different from processing still images; while using an image segmentation algorithm and processing the video frame by frame is possible, it is still not financially viable because it requires taking into account spatial-temporal dependencies. There aren't many works on segmenting videos semantically, but one is clockworkFCN [24], which builds multiple CNN layers based on the number of frames and combines them into one additional layer or classifier. A work using 3D ConvNet for video semantic segmentation is from [25]. In addition, 3D ConvNet is used in video semantic segmentation because it has the ability to add spatial-temporal correlation during the convolutional process.

2.2. Video forgery detection

There are two methods—passive and active—for detecting video forgeries. During the pre-processing phase of an active approach, a digital signature, such as a watermark, is embedded during the construction of the video.Any changes to this digital signature are regarded as forged [26]. This method is not widely used in the community of digital forgery detection because it requires extremely complex software and has a tendency to lower the video quality. Unlike the active approach, the passive approach typically occurs during the post-processing stage, where it looks for any instances of the underlying statistical correlation being inconsistent for any given pixel or frame from a given video. Inter-frame forgery, double/multiple compression, and region tampering are three methods for detecting forged video.

A new path has recently opened up for improving performance in spotting fake objects in videos thanks to the introduction of deep learning methodologies. ConvNet is a popular architecture for detecting video forgeries, and [27] uses ConvNet to find copied and moved fake objects in video frames. In addition, [1] combined LSTM and ConvNet to detect splicing forged objects, but this time taking spatio-temporal correlation into account. Convolutional 3D Neural Networks (C3D) were used by [28] to detect frame dropping across many videos without affecting performance. To detect facial forgery, MesoNet [29] proposed a ConvNet-based architecture that is not only highly accurate but also computationally light. Recently, by combining forensic-based filters as a pre-process before passing them into the

architectures for feature extraction and classification, Zampoglou *et al*. (2019) enhanced well-known architectures like GoogleNet and ResNet [30].

3. The architectures

This section will start with brief explanation on the U-net and VGG19 architectures, which the proposed method inspired from.

3.1. Passive approach

2015 ISBI cell tracking challenge for cell segmentation in light microscopy images was won by Ronneberger's U-net [14]. U-net architecture originates from FCN and Autoencoder architecture, and the architecture was modified and upgraded due to the limitations of the ISBI Challenge 2012 winner's network architecture [31]. As depicted in Figure 1, U-net architecture is composed of three parts, the contracting part (left side), the bottleneck part (lower middle), and the expanding part (right side). U-net proved to be faster and more precise in segmentation tasks even with a few sample images.

The contracting part uses a normal ConvNet network architecture with 4 convolutional blocks, where each block has 3x3 unpadded convolutions, followed by a 2x2 max pooling operation with stride 2. After each max pooling operation, they doubled the number of feature channels, starting with 64 and ending up with 512 channels. Every ConvNet layer in this part is followed by a Rectified Linear Unit (ReLU) activation function. ReLU can be defined as

$$f(x) = max(0, x) \tag{1}$$

where you can see that ReLU only take the positive values of input x while the negative value will be set as 0. Due to its low processing load and quick convergence, ReLU is the most commonly employed activation function in neural networks [32].

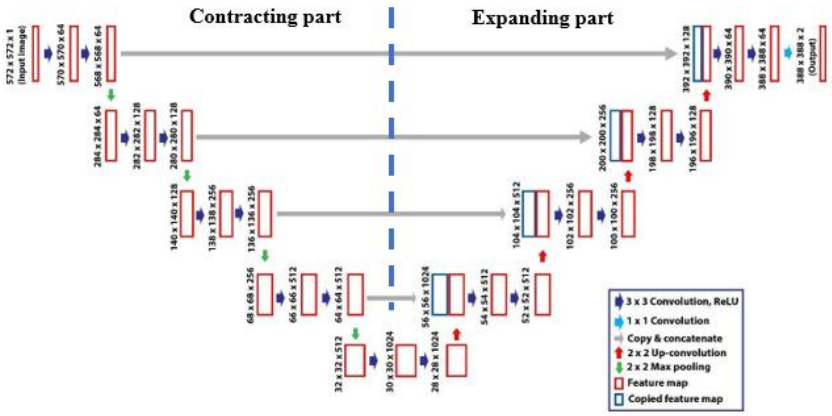

Figure 1. U-net architecture.

The bottleneck part is in between the contracting and expanding parts and consists of only 2 ConvNet layers with dropout. The purpose of the expanding part is to enable precise localization combined with contextual information from the contracting part. Every step in the expanding section consists of two 3x3 convolutions with ReLU activation, followed by a

2x2 up-convolution that doubles the feature map's size. U-net then grows the number of channels by copying and concatenating the cropped feature map from the contracting path. Each of the 64 feature vectors from the final layer is mapped to the required number of classes using a 1x1 convolution.

3.2. VGG-19 architecture (VGG19)

VGG19 [33] was developed by the Visual Geometry Group at the University of Oxford; its architecture is inspired by [34] and [35] deep ConvNet architectures and employs the same underlying principles. The main idea of this architecture is with smaller kernel size, but deeper ConvNet layer can provide more accurate in object detection simultaneously improve the image classification and localization.

3.3. Proposed architecture

Due to the advantages shown by the U-net and VGG19 architectures (Figure 2), the proposed architecture adopted the U-net with the ConvNet layers constructed based on the VGG19. The main contribution is building a ConvNet based architecture that not just with decent depth, but the architecture also consisted of a small number of weights to deliver good results in discriminating between the forged object and background image.

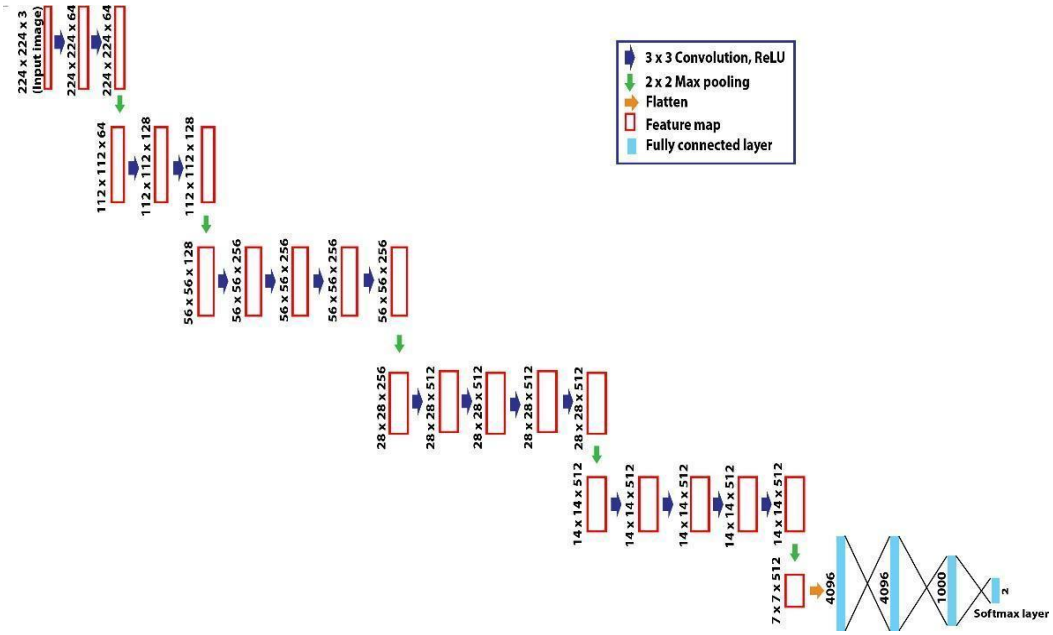

Figure 2. VGG19 architecture.

As shown in Figure 3, the proposed architecture consists of 36 layers of ConvNet (including 1 x 1 convolution at the last layer), deeper than the original U-net architecture. For each ConvNet layer (except the last layer), the proposed method used a 3 x 3 kernel size with stride 1 and zero padding to keep the size of output feature map until it downsizes by the max pooling layer. The convolution process is followed by the ReLU activation function and normalisation process for every layer.

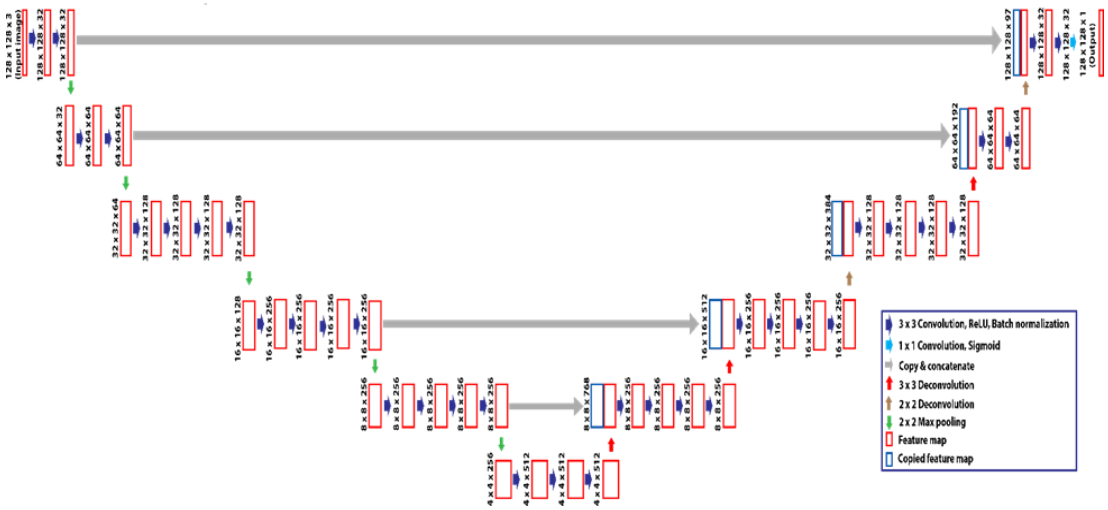

Figure 3. Proposed architecture.

Like U-net, the proposed architecture consisted of three parts. The contracting part follows typical VGG19 layers. It consisted of a total of 16 zero-padded ConvNet layers with stride 1, followed by ReLU activation and Batch normalisation for each layer. For feature map downsampling, five max-pooling layers are used, which follow some of the ConvNet layers (not all the ConvNet layers are followed by max-pooling; refer to Figure 3). Max-pooling is performed over a 2 x 2 window with stride 2. This research doubles up the number of channels C after every max-pooling layer. The proposed architecture starts with 32 channels and ends up with 512 channels for this contracting part. To make the VGG19 layers fit into the U-net architecture, this research converted all three fully connected layers from the original VGG19 into the ConvNet layers. All these 3 ConvNet layers are represented in the bottleneck layer, which makes the architecture have an extra ConvNet layer compared to U-net.

For the expanding part, 3 x 3 and 2 x 2 kernel windows of Deconvolutional networks [12] were used to reconstruct the output feature map at the same size as the ground truth image. This research doubled the number of channels after the deconvolution process by copying and concatenating the channels from the expanding part, which assists the network in propagating context information to higher resolution layers [14]. For the last layer, this research used 1 x 1 convolution with a sigmoid function to classify each pixel into its designated class. This research chose the sigmoid function because of its superior performance in binary-class classification. Binomial cross-entropy to calculate the loss error between the predicted image and the ground truth image. Different than the original VGG19, this research halved the number of channels for every layer. This decision significantly reduced the number of weight parameters and decreased the training time (will be discussed in Section 5).

4. Experiments

In this section, the experimental setup including data preparation will be described, hyper-parameters, evaluation metric, and training model in detail. This is for the purpose of research reproducibility.

4.1. Datasets

All the experiments were conducted using 10 short videos that only involved slicing forged objects. The video datasets consisted of original videos taken by the [1] itself with different cameras for each video, forged videos (uncompressed AVI), YouTube upload/download quality versions of the forged videos, and the ground truth. The videos were created using Adobe After Effects CC software. All the details for the videos are explained in [1] and available for download at www.grip.unina.it.

This research only focuses on uncompressed forged videos. All the videos were converted into picture frames, then pooled all the videos as training and validation sets with a 90:10 ratio and reserved the Girl video as a test set. This makes the total number of training sample images 2779, while validation images are 309, and 371 for the test set. The total number of frames and forged frames is as revealed in Table 1.

Table 1. Number of frames for each video.

No	Video name	No of frames	No of forged frames
1	Tank	335	191
2	Man	399	207
3	Cat	281	71*
4	Helicopter	488	292
5	Chicken	373	169
6	Lion	294	228
7	UFO	306	96
8	Tree	302	240
9	Girl	371	162
10	Dog	310	186
	Total frames	**3459**	**1842**

* Represents a video with the lowest forged frames. This shows that the model can be trained to recognise forged frames even with low samples.

Table 2. Arguments and values for data augmentation.

Argument	Value
Shear range	0.5
Rotation range	50
Zoom range	0.2
Width shift range	0.2
Height shift range	0.2
Fill mode	Reflect
Horizontal flip	True
Vertical flip	True
Seed	123

Data augmentation plays an important role in reducing the possibility of training models becoming overfit, especially in this research, which involved a low number of samples. For data augmentation, this research used the ImageDataGenerator class from Keras to provide real-time data augmentation for every sample image by batch. Table 2 shows the arguments and values used in this research for data augmentation. Figures 4 (c) and (d) show examples of the augmented image and ground truth, respectively.

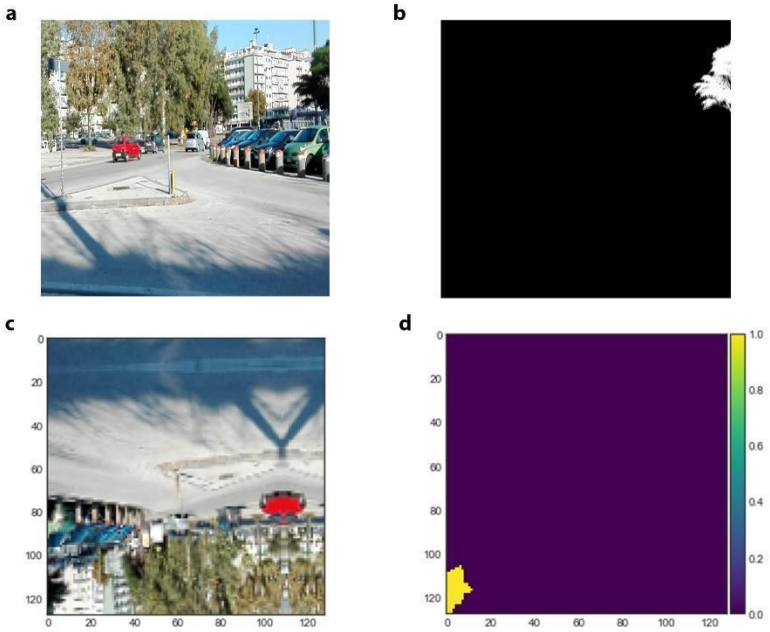

Figure 4. Example of data augmentation **(a)** Original image **(b)** Original ground truth **(c)** Augmented image **(d)** Augmented ground truth.

For the evaluation metric, this research used mean Intersection over Union (mIoU) accuracy to evaluate the performance of the methods. IoU accuracy has been widely used in image and video semantic segmentation compared to by-pixel classification accuracy because it has been proven to avoid the bias in a class imbalance between the forged object (non-background) and the image background [21]. For example, let's say the forged object in the video consists of 10 percent compared to the background with 90 percent of total pixels. The by-pixel classification will easily achieve 90 percent accuracy when the classifier classifies all the pixels as image background.

The IoU can surmount this limitation by measuring the similarity of the predicted forged object region with the actual forged object region in the ground truth image, which can be defined as the size (in pixels) of the intersection region between the predicted and actual objects divided by the union of both regions. Equations 2, 3, and 4 further explain the implementation of the mIoU.

$$IoU = \frac{TP}{TP + FP + FN} \tag{2}$$

where *TP*, *FP*, *FN* denoted as true positive, false positive, and false negative respectively. Therefore, the calculation of IoU can be as

$$IoU(y, y') = \frac{|\{y = c\} \cap \{y' = c\}|}{|\{y = c\} \cup \{y' = c\}|} \quad (3)$$

with y and y' is ground truth pixel and predicted pixel respectively with the class of $c = \{0,1\}$ which 0 is the background pixel and 1 is the forged object, which displays the intersection between the evaluated mask and the ground truth mask over their union as a ratio in [0, 1]. For the purpose of calculating the overall mean IoU (mIoU), the IoU score is calculated independently for each class as in Equation 4 below.

$$mIoU = \frac{1}{c}\sum_{i=1}^{c} IoU(y, y') \quad (4)$$

Since this research used binary classes (background and forged object). Rather than averaging the IoU with the number of classes, this research calculates the mIoU by setting certain thresholds for each IoU score on every sample image. From Equation 4, the formula for calculating mIoU is as follows:

$$\mathrm{mIoU} = \frac{1}{|\mathrm{t}|}\sum_{\mathrm{t}} \mathrm{IoU}(\mathrm{y}, \mathrm{y}') \quad (5)$$

where *t* is the number of thresholds with values that have a 0.05 step size and range from 0.5 to 0.95. To put it another way, a predicted object is said to have "hit" status at a threshold of 0.5 if its intersection over union with a ground truth object is bigger than 0.5. These thresholds are widely used in image semantic segmentation's challenge such Microsoft COCO challenge.

4.2. Environment and hyperparameters setup

This research developed and trained the proposed method using Tensorflow v1.10 [33] with Keras v2.2.2 API [34]. For model training, Workstation with Intel i7-8700 CPU processor, 32GB of RAM memory, and Nvidia GTX 1060ti GPU with 6GB of VRAM was used. The mid-range of GPU for training model was used, to show that the proposed method is using low memory and processing power. Table 3 provides a summary of the hyper-parameters used in this experiment. All methods run in this experiment were using the same hyperparameters.

Table 3. Hyper-parameters setup.

Hyper-parameter	Value
Number of epochs	1000
Batch size	16
Steps per epoch	97 (Number of samples/batch size)
Learning rate initializer	0.099

During model training, the model's weight was validated on the validation set for every epoch. The best model with the highest validation mean IoU is saved for further evaluation using the testing set. To avoid the model learning become stagnate during training, this

research use ReduceLROnPlateau function from Keras where it will reduce the learning rate to the factor of 2 until it reaches the minimum rate 0.0001.

5. Results and discussion

The main research focus is to develop a ConvNet-based deep learning architecture that is fast, less computationally burdensome, and provides significant accuracy in the identification and segmentation of forged objects. As shown in Table 2, the proposed architecture achieved the lowest running time compared to other benchmark architectures. The proposed architecture has a smaller number of parameters compared to others, but it has been proven that it won't jeopardise the performance since the architecture has enough depth in the ConvNet layer for the machine learning to learn. It can be proven by the proposed architecture that it achieved the highest mIoU accuracy and the second lowest loss error rate (Table 4).

Table 4. mIoU performance on Training dataset.

Methods	Mean IoU	Loss error	Running time
Unet	Training: 0.9266	Training: **0.0094**	12 hours 33 minutes
	Validation: 0.9266	Validation: 0.0341	
Unet-VGG19	Training: 0.4921	Training: 0.1842	13 hours 21 minutes
	Validation: 0.4925	Validation: 0.3251	
Unet-VGG19	Training: 0.9334	Training: **0.0094**	15 hours 38 minutes
(batch normalization)	Validation: 0.9334	Validation: **0.0177**	
Proposed method	Training: **0.9343**	Training: 0.0104	**7 hours 22 minutes**
	Validation**: 0.9343**	Validation: 0.0184	

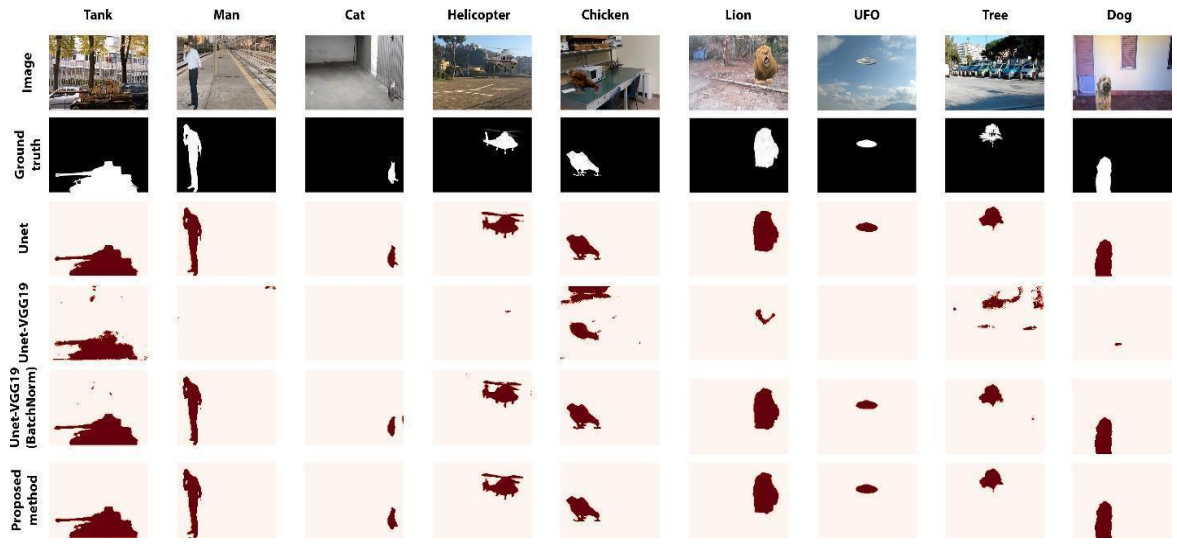

Figure 5. Predicted output for training data with the ground truth.

Figure 5 shows that the proposed architecture successfully predicts the forged object, especially on tree video. Unet-VGG19 has the worst result because, without batch

normalisation, it can produce sparsity in feature output [36]. Figure 5 also shows that the only architecture comparable to the proposed method is the U-net. This is because the U-net architecture was developed to tackle regular foreground and background segmentation problems [5]. This is the reason why this research chose U-net as the backbone of the proposed method.

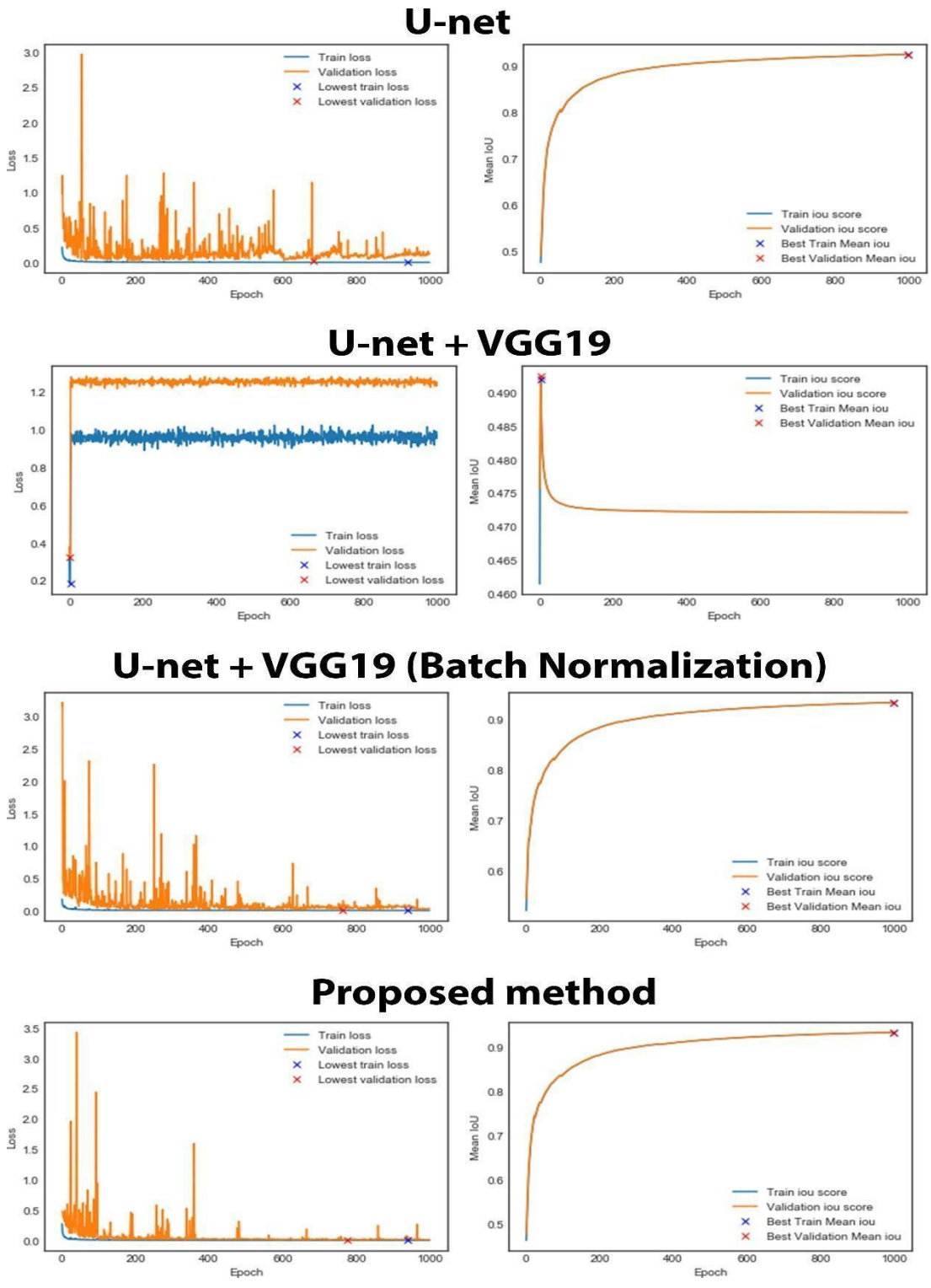

Figure 6. Loss error and Mean IoU accuracy graphs for 1000 epochs

In Figure 6, the graph of the mean IoU for 1000 epochs shows that only U-net + VGG19 suffers from a decline in accuracy, while the other 3 architectures show a constant improvement on every epoch. From the results, it shows none of the three architectures suffer from overfitting since they have similar IoU scores between the training set and validation set. The U-net and VGG19 have major overfitting from the large difference between the train error and the validation error, and the loss error also seems to increase over time. U-net shows some tendency to overfit where there is some slight difference between train and validation errors, and it happens after 500 epochs and can be avoided with an early stopping function. The proposed method and U-net + VGG19 with normalisation do not show any sign of overfitting from the loss error, but the proposed method is more stable.

Applying the trained model to testing data and expecting a good result is not easy, even if it shows good performances over training and validation data during the training process. In Figure 6, the proposed method successfully predicts the forged area (Figure 7). It shows that the trained model has flexibility over different types of data with different distributions. Even though it is not smooth and accurate, a few improvements are still needed for the proposed method.

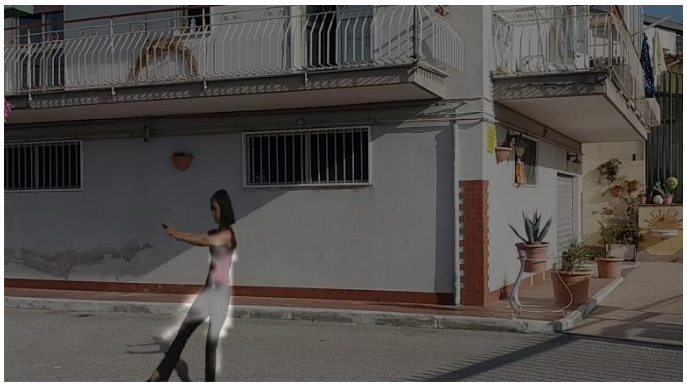

Figure 7. Predicted output on test data. The white mark shows the predicted output.

6. Conclusions

This research successfully developed a ConvNet-based architecture for detecting forged objects in the video. The proposed architecture had shown its superiority in mIoU accuracy compared to other benchmark architectures. However, with some flaws in the test dataset, more improvements need to be made. Maybe by considering spatial-temporal correlation in processing the video. Besides that, with the limitation of public datasets, the proposed method only focuses on splicing-type forgery; it can be extended to other types of forgery such as copy-move or double/multiple compression. In the future, detecting forged videos can become more sophisticated and complex as people start using deep learning approaches to develop forged images or videos, such as the Generative Adversarial Network (GAN). This will bring new challenges to the image and video research communities. However, this research has built some foundations for detecting forged objects on video via semantic segmentation and can be further improved to tackle the stated problems.

Acknowledgement

The authors would like to express their gratitude to the Universiti Teknologi Malaysia (UTM) for the financial support of this research through grants UTMFR (Q.J130000.2551.21H71) The research is additionally supported by the UTM Faculty of Computing.

Conflicts of interests

The authors declare no conflict of interests.

References

[1] D'avino D, Cozzolino D, Poggi G, Verdoliva L. Autoencoder with recurrent neural networks for video forgery detection. *arXiv* 2017, arXiv:1708.08754.
[2] Pandey RC, Singh SK, Shukla KK. Passive forensics in image and video using noise features: A review. *Digit. Investig.* 2016, 19:1–28.
[3] Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Martinez-Gonzalez P, *et al.* A survey on deep learning techniques for image and video semantic segmentation. *Appl. Soft Comput.* 2018, 70:41–65.
[4] Yang Y, Hallman S, Ramanan D, Fowlkes CC. Layered object models for image segmentation. *IEEE Trans. Pattern Ana.l Mach. Intell*. 2011, 34(9):1731–1743.
[5] Forsyth D. Object detection with discriminatively trained part-based models. *Comput.* 2014, 47(2):6–7.
[6] Chierchia G, Parrilli S, Poggi G, Sansone C, Verdoliva L. On the influence of denoising in PRNU based forgery detection. In *Proceedings of the 2nd ACM Workshop on Multimedia in Forensics, Security and Intelligence*, 2010, pp. 117–122.
[7] Richao C, Gaobo Y, Ningbo Z. Detection of object-based manipulation by the statistical features of object contour. *Forensic Sci. Int.* 2014, 236:164–169.
[8] Zhang Y, Brady M, Smith S. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Trans. Med. Imaging.* 2001, 20(1):45–57.
[9] Thoma M. A survey of semantic segmentation. *arXiv* 2016, arXiv:1602.06541.
[10] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc. IEEE* 1998, 86(11):2278–2324.
[11] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
[12] Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* 2010, 88:303–338.
[13] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 2017, 39(12):2481–2495.
[14] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 2015, pp. 234–241.
[15] Zhou L, Kong X, Gong C, Zhang F, Zhang X. FC-RCCN: Fully convolutional residual continuous CRF network for semantic segmentation. *Pattern Recognit. Lett.* 2020, 130:54–63.

[16] Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* 2014, arXiv:1412.7062.
[17] Zheng S, Jayasumana S, Romera-Paredes B, Vineet V, Su Z, *et al.* Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1529–1537.
[18] Zhou S, Wu JN, Wu Y, Zhou X. Exploiting local structures with the kronecker layer in convolutional networks. *arXiv* 2015, arXiv:1512.09194.
[19] Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* 2017, 40(4):834–848.
[20] Paszke A, Chaurasia A, Kim S, Culurciello E. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv* 2016, arXiv:1606.02147.
[21] Wang Y, Luo B, Shen J, Pantic M. Face mask extraction in video sequence. *Int. J. Comput. Vis.* 2019, 127:625–641.
[22] Liu W, Rabinovich A, Berg AC. Parsenet: Looking wider to see better. *arXiv* 2015 arXiv:1506.04579.
[23] Pinheiro PO, Lin TY, Collobert R, Dollár P. Learning to refine object segments. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 2016, pp. 75–91.
[24] Shelhamer E, Rakelly K, Hoffman J, Darrell T. Clockwork convnets for video semantic segmentation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, 2016, pp. 852–868.
[25] Zhang H, Jiang K, Zhang Y, Li Q, Xia C, *et al.* Discriminative feature learning for video semantic segmentation. In *2014 International Conference on Virtual Reality and Visualization*, 2014, pp. 321–326.
[26] Sitara K, Mehtre BM. Digital video tampering detection: An overview of passive techniques. *Digit. Investig.* 2016, 18:8–22.
[27] Yao Y, Shi Y, Weng S, Guan B. Deep learning for detection of object-based forgery in advanced video. *Symmetry* 2017, 10(1):3.
[28] Long C, Smith E, Basharat A, Hoogs A. A c3d-based convolutional neural network for frame dropping detection in a single video shot. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1898–1906.
[29] Afchar D, Nozick V, Yamagishi J, Echizen I. Mesonet: a compact facial video forgery detection network. In *2018 IEEE international workshop on information forensics and security (WIFS)*, 2018, pp. 1–7.
[31] Zampoglou M, Markatopoulou F, Mercier G, Touska D, Apostolidis E, *et al*. Detecting tampered videos with multimedia forensics and deep learning. In *MultiMedia Modeling: 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8–11, 2019, Proceedings, Part I 25*, 2019, pp. 374–386.
[31] Ciresan D, Giusti A, Gambardella L, Schmidhuber J. Deep neural networks segment neuronal membranes in electron microscopy images. *Adv. Neural Inf. Process. Syst.* 2012, 25.
[32] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 2012, 25.
[33] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, *et al*. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* 2016, arXiv:1603.04467.
[34] Chollet F and others. *Keras*. GitHub, 2015.
[35] Ciresan DC, Meier U, Masci J, Gambardella LM, Schmidhuber J. Flexible, high performance convolutional neural networks for image classification. In *Twenty-second international joint conference on artificial intelligence*, 2011.

[36] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 2015, pp. 448–456.