ObjectMix: Data Augmentation by Copy-Pasting Objects in Videos for Action Recognition

Jun Kimata
Nagoya Institute of Technology
Japan

Tomoya Nitta
Nagoya Institute of Technology
Japan

Toru Tamaki
Nagoya Institute of Technology
Japan

ABSTRACT

In this paper, we propose a data augmentation method for action recognition using instance segmentation. Although many data augmentation methods have been proposed for image recognition, few of them are tailored for action recognition. Our proposed method, ObjectMix, extracts each object region from two videos using instance segmentation and combines them to create new videos. Experiments on two action recognition datasets, UCF101 and HMDB51, demonstrate the effectiveness of the proposed method and show its superiority over VideoMix, a prior work.

CCS CONCEPTS

• **Computing methodologies** → **Activity recognition and understanding**.

KEYWORDS

action recognition, data augmentation, instance segmentation

ACM Reference Format:
Jun Kimata, Tomoya Nitta, and Toru Tamaki. 2022. ObjectMix: Data Augmentation by Copy-Pasting Objects in Videos for Action Recognition. In *ACM Multimedia Asia (MMAsia '22), December 13–16, 2022, Tokyo, Japan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3551626.3564941

1 INTRODUCTION

In recent years, there has been a lot of research on video recognition, which are used in various applications. One of the problems in developing action recognition models is the cost of constructing datasets [19, 27, 35]. In actual application scenarios, practitioners often need to prepare a new dataset for their tasks, but the annotation cost of labeling a large number of videos is inherently high, and for some applications it may not be possible to collect many videos in the first place. In which cases, training on a small dataset is inevitable.

There are three approaches to the small dataset issue. The first is to synthesize various images with 3D models, for example, pose estimation [31] and flow estimation [2, 4, 22]. This approach has the advantage of being able to generate as many images as possible, while it can only be applied to tasks where images can be easily synthesized. For action recognition, it is necessary to consider various objects in scenes, motions and actions of people, but currently no methods are available to generate such realistic action videos.

The second is to use GAN [11, 23] to generate images. Once a GAN model has been trained, it can generate any number of realistic images. This approach has been used as data generation in medical image processing where it is not possible to collect data at a large scale [6, 18]. GANs are good at generating images with specific structures such as faces, human bodies, animals, and indoor images. But it is still difficult to generate videos of various scenes [28, 30] such as those appear in videos for action recognition.

The third is data augmentation [20, 25, 37], which is widely used because of its simplicity. Data augmentation refers to applying various image processing to images such as flipping, rotation, contrast change, and adding various noises [1, 17]. This means that even the dataset size is small, the generalization performance of the model is expected to be as good as when trained on a large dataset. Recently, mix-type methods [3, 39, 41] have been proposed that apply operations such as cropping a portion of an image, pasting the portion onto another image, and blending these two images and their corresponding labels. In addition, task-specific methods have also been proposed, such as for monocular depth estimation [16], super-resolution [38], object detection [5], and instance segmentation [10]. However, few are specific to action recognition [15]. Commonly used data augmentation for action recognition is simply applying the same geometric and photometric augmentation to all frames at once, except vertical flipping because usually videos are not recorded upside down. Also, for some datasets horizontal flipping is also not used since some actions distinguish left and right; for example, in something-something v2 (SSv2) [12], "move from right to left" and "move from left to right" are different categories. An exception is VideoMix [40], a recently proposed mix-type data augmentation method specialized for action recognition. However, this is a simple application of CutMix [39] to the spatio-temporal 3D volume of video frames, and does not take into any consideration the temporal and spatial continuity of the video contents.

In this study, we propose a method that extends Copy-Paste [10] to action recognition. Copy-Paste is a mix-type method that generates new images by extracting object regions from two images by semantic segmentation, and then pasting the objects onto each other using the other image as a background. The proposed method performs it to videos, in other words, object regions from each of the two video frames and paste them into each other's video frames to create new ones. The contributions of this paper are as follows.

- We propose a new mix-type method of data augmentation for action recognition. It is an extension of Copy-Paste, and unlike VideoMix, it is possible to generate video frames that

MMAsia '22, December 13–16, 2022, Tokyo, Japan

ACM ISBN 978-1-4503-9478-9/22/12…$15.00
https://doi.org/10.1145/3551626.3564941

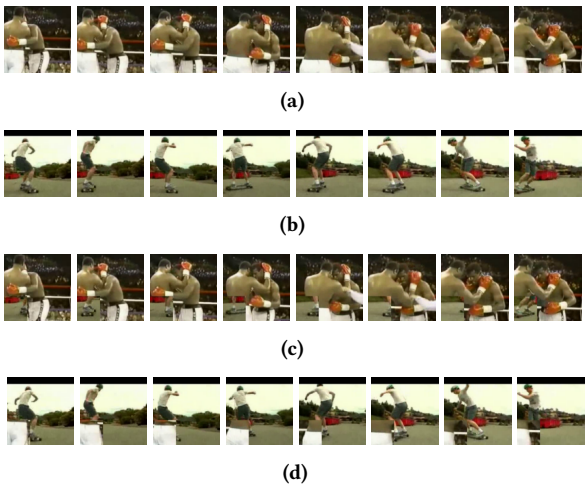


Figure 1: Example of video generation with VideoMix. (a)(b) Two original videos and (c)(d) two generated Videos.

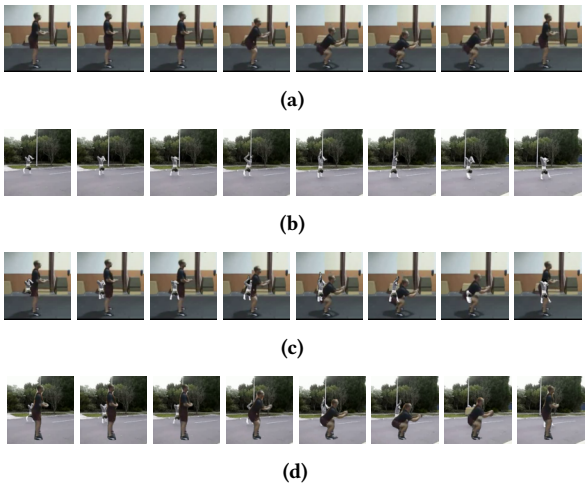


Figure 2: Example of video generation with ObjectMix. (a)(b) Two original videos and (c)(d) two generated Videos.

take into consideration the temporal and spatial continuity of objects in the videos.
- The proposed method is applicable to any existing action recognition models. This allows the recognition performance of existing models to be improved using the proposed method.
- In experiments using two action recognition datasets, we show that the performance of the proposed method is better than that of VideoMix.

2 RELATED WORKS

Data augmentation has been widely used to improve the performance by augmenting training samples with various transformations [20, 25, 37]. In addition to simple image processing such as rotation, translation, noise [1, 17], there are also a number of mix-type methods (Mixup [41], Cutout [3], CutMix [39]), and task-specific methods (CutDepth [16], CutBlur [38], Cut-Paste-Learn [5]). Copy-Paste [10] is a simple method for the task of instance segmentation: for two images, it cuts out only the instance region of in one image, and randomly pastes it onto the other image. Advantages of this method include the fact that spatial continuity is guaranteed since the whole instances are always pasted, and that further augmentation, such as scaling, can be performed on the pasted instances.

Action recognition is the task of identifying human actions in a video [15]. Unlike image recognition, action recognition requires to model temporal information. For example, Two-Stream types [26] use optical flow as input as the temporal information, and 3D CNN methods such as 3D ResNet [13], X3D [7], and SlowFast [8] perform 3D convolution in spatio-temporal volumes. Usually, models pre-trained on large datasets (Kinetics [19]) are transferred to small datasets (UCF101 [27] and HMDB51 [35]). However, even when fine-tuning on small datasets, the training set should be diverse, and in such cases, data augmentation would also be important to ensure better generalization performance.

VideoMix. The above data augmentation methods were proposed for image recognition tasks. Few methods exist for action recognition, with the sole exception of VideoMix [40], whose example is shown in Figure 1. This method is a direct extension of CutMix [39] to 3D video volumes, whereby a cube is cut from one video volume and pasted to another. The problem here is the discontinuity of objects in the video. Since the location of the cube is randomly selected, the entire regions of objects and humans may not be shown in the pasted video, or even worse, no objects might appear in the cube. Furthermore, the objects that appear in the beginning may disappear in the middle of the video. In contrast, as shown in Figure 2, the proposed method does not break the continuity of the objects and humans to be pasted, and keeps the objects shown for all the frames.

3 METHOD

This section describes an overview of the proposed method that consists of the following processes.

(1) Preparing two source videos v_1 and v_2. Let y_1, y_2 be the labels of each.
(2) Extracting object regions from each video and generating object masks M_1 and M_2.
(3) Pasting the masked region of one video onto the other video to create new videos v_{12} and v_{21}.
(4) Using the mask information to generate labels y_{12} and y_{21}.

Examples of videos generated by the proposed method are shown in Figures 2 (c) and (d).

3.1 Video preparation

The source videos $v_1, v_2 \in \mathbb{R}^{T\times C\times H\times W}$ are video clips consisting of T frames $v_1(t), v_2(t) \in \mathbb{R}^{C\times H\times W}$ for $t = 1, \ldots, T$, where H, W are height and width of the frames. Let $y_1, y_2 \in \{0, 1\}^{L_a}$ be the corresponding one-hot encoded labels, where L_a is the number of categories.

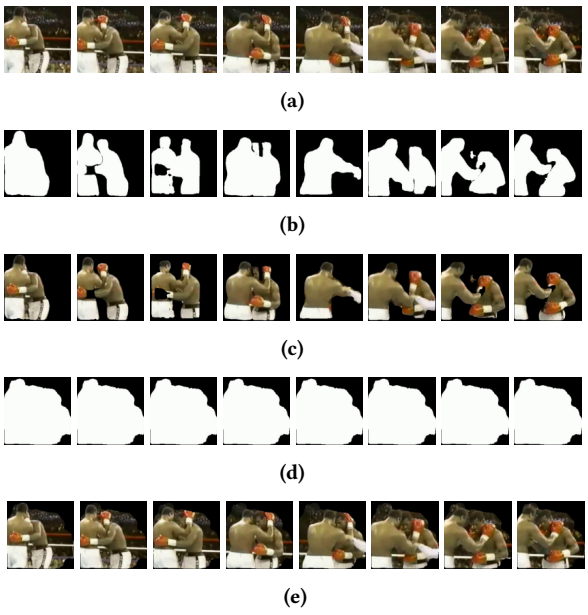


Figure 3: Example of extracted object masks. (a) Original video. (b) Mask M'_1 and (c) corresponding objects. (d) Mask M''_1 and (e) corresponding objects.

3.2 Extracting object masks

For each frame $v_k(t)$ for $k = 1, 2$, we apply instance segmentation to generate masks $M_k(t) \in \{0,1\}^{N_k(t)\times H\times W}$ for $t = 1, \ldots, T$ where $N_k(t)$ is the number of instances detected in $v_k(t)$.

If multiple instances are extracted (i.e., $N_k(t) > 1$), they are aggregated into a single-channel mask $M'_k(t) \in \{0,1\}^{1\times H\times W}$ by logical OR as follows;

$$M'_k(t) = \bigcup_{n=1}^{N_k(t)} M_{k,n}(t), \tag{1}$$

where $M_{k,n}(t)$ is the n-th channel of $M_k(t)$. Examples of this mask and extracted objects are shown in Figure 3(b). We used Detectron2 [36] that are pre-trained on the COCO [21] dataset with 80 classes (therefore $0 \le N(t) \le 80$). In this study, all extracted instances are used for mask generation, regardless of the relevance of the extracted instance categories to the action categories.

3.3 Temporal aggregation of masks

Masks are extracted from each frame, however, the temporal continuity of the masks would be lost if the instance segmentation fails to detect objects in a certain frame as shown in fifth column of Figure 3(b). Therefore, we propose to aggregate the masks of each extracted frame by logical OR in the temporal direction as well.

$$M''_k(t) = \bigcup_{t=1}^{T} \bigcup_{n=1}^{N_k(t)} M_{k,n}(t). \tag{2}$$

The masks M''_k are the same for all frames, however, even in frames where detection fails, the mask of the object can still be extracted. An example of this mask M''_k is shown in Figure 3(d). In our experiments, we refer to the proposed method with M'_k as ObjectMix, and the version with M''_k as ObjectMix+or.

3.4 Video and Label Composition

Next, we generate new videos using the generated masks M'_1, M'_2 (or M''_1, M''_2). In this case, two videos can be generated, that is, the object extracted with mask M'_1 from one video v_1 is pasted onto the other video v'_2, and vice versa.

$$v_{12}(t) = v'_1(t) \odot M'_1(t) + v'_2(t) \odot (1 - M'_1(t)) \tag{3}$$

$$v_{21}(t) = v'_1(t) \odot (1 - M'_2(t)) + v'_2(t) \odot M'_2(t), \tag{4}$$

where $\odot$ is the element-wise product.

To define weights for label composition, we use the fraction of pixels with non-zero values in the generated object masks. First, we define weights as $\lambda_k = \frac{|M'_k|}{THW}$, where $|M'_k|$ is the sum of non-zero values in the binary mask M'_k. Then, we composite the labels as follows;

$$y_{12} = \lambda_1 y_1 + (1 - \lambda_1) y_2 \tag{5}$$

$$y_{21} = (1 - \lambda_2) y_1 + \lambda_2 y_2. \tag{6}$$

This is similar to CutMix [39], however the weights of CutMix are fixed in advance, and the rectangle whose ratio of the rectangle's area to the entire image matches the weight is randomly selected. In contrast, the weights of the proposed method are dynamically adjusted according to the area of objects in the mask.

3.5 Loss

To train a model, we compute the cross-entropy (CE) loss. Let model predictions be $\hat{y}_{ij}, \hat{y}_{ji}$ for videos v_{ij}, v_{ji} generated from source videos v_i, v_j. The CE losses $L_{\mathrm{CE}}(\hat{y}_{ij}, y_{ij})$ or $L_{\mathrm{CE}}(\hat{y}_{ji}, y_{ji})$ need to be computed with an appropriate weight.

A typical implementation of mix-type augmentation computes the loss on a batch basis. Assuming that videos v_i, v_j are in the batch of size B, the loss for the batch is calculated as follows;

$$\lambda \sum_{i=1}^{B} L_{\mathrm{CE}}(\hat{y}_{ij_i}, y_i) + (1 - \lambda) \sum_{i=1}^{B} L_{\mathrm{CE}}(\hat{y}_{ij_i}, y_{j_i}), \tag{7}$$

where $j_1, \ldots, j_B$ is a certain permutation of $1, \ldots, B$.

However, in the case of the proposed method, and the weights are the relative area of the masks and different for each sample in the batch. Therefore, we compute the following loss;

$$\sum_{i=1}^{B} \lambda_i L_{\mathrm{CE}}(\hat{y}_{ij_i}, y_i) + (1 - \lambda_i) L_{\mathrm{CE}}(\hat{y}_{ij_i}, y_{j_i}). \tag{8}$$

4 EXPERIMENTAL RESULTS

In this section we report experimental results with two action recognition datasets to evaluate the performance of the proposed method and compare it with VideoMix.

4.1 Datasets

The following two datasets were used.

UCF101 [27] has 101 classes of human actions, consisting of a training set of about 9500 videos and a validation set of about 3500

Table 1: The top-1 performance of ObjectMix (OM) on UCF101 and HMDB51 validation set. The $p = 0.0$ is the baseline without applying the proposed method.

	UCF101		HMDB51	
method (p)	top-1	top-5	top-1	top-5
OM (0.0)	93.58 ± 0.03	99.23 ± 0.01	69.86 ± 0.39	91.89 ± 0.26
OM (0.2)	94.68 ± 0.23	**99.65** ± 0.05	70.59 ± 0.37	93.18 ± 0.54
OM (0.4)	93.63 ± 0.24	99.40 ± 0.08	**71.47** ± 0.30	92.77 ± 0.31
OM (0.6)	**95.12** ± 0.15	99.46 ± 0.07	71.44 ± 0.59	**93.66** ± 0.28
OM (0.8)	93.94 ± 0.23	99.22 ± 0.05	70.74 ± 0.29	92.38 ± 0.26
OM (1.0)	93.45 ± 0.21	98.96 ± 0.07	69.43 ± 0.46	92.49 ± 0.28

videos. Each video was collected from Youtube, with an average length of 7.21 seconds. There are three splits for training and validation, and we report the performance of the first split as it is usually used.

HMDB51 [35] has 51 classes of human actions, consisting of a training set of 3570 videos and a validation set of 1530 videos. Each video is collected from movies, Web, Youtube, etc., and the average length is 3.15 seconds. There are three splits for training and validation, and we used the first split.

4.2 Experimental Settings

We used X3D-M [7], a 3D CNN-based action recognition model, pretrained on Kinetics400 [19]. In training, we randomly sampled 16 frames per clip from the video, and randomly determined the short side of the frame in the range of [224, 320] pixels and resized it while maintaining the aspect ratio, and randomly cropped a 224×224 pixel patch, then flipped horizontally with a probability of 50%. No photometric augmentation were used. The optimizer was Adam with the learning rate of 0.0001 and the batch size of 16. Training epochs were set to 10 for UCF101 and 20 for HMDB51, so that the top-1 performance for the training set would roughly converge.

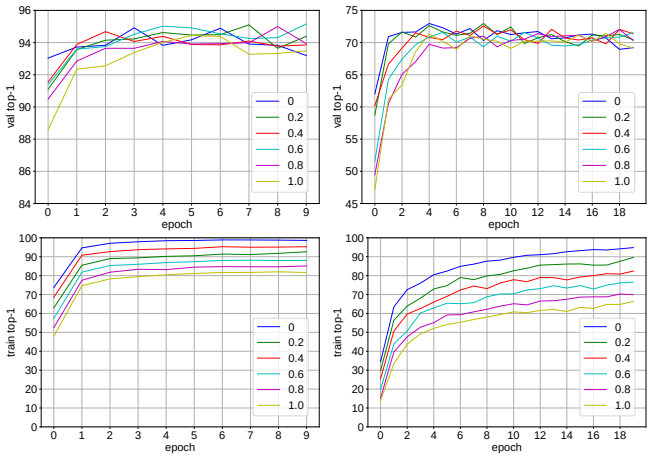
We used a single view test for validation (i.e., one clip was randomly sampled from a single video) instead of the multi-view test [34]. Frames were resized so that the short side of the frame is 256 pixels while maintaining the aspect ratio, and the central 224×224 pixels of the frame were cropped. To take into account the randomness of the clip sampling, we report the mean and standard deviation of 10 results.

Augmentation was randomly applied to each batch with the probability $0 \leq p \leq 1$. Note that $p = 0$ is equivalent to the case where no augmentation is applied. In the following experiments, performances are reported for $p = 0, 0.2, \ldots, 1.\ 0.2$.

4.3 Results of ObjectMix

First, we show the comparison of ObjectMix ($p > 0$) with no augmentation ($p = 0$) in Table 1. As p increases, the performance tends to have a peak at $p = 0.6$ or 0.4, then deteriorates for larger values. This is probably due to the fact that the original videos are used less for training for large values of p. In other words, the original and generated videos should be balanced.

The performances over training epochs are shown in Figure 4. The validation results show that the performance is worse than the case with $p = 0$ in the early stages of training, whereas it becomes



(a) UCF101 **(b) HMDB51**

Figure 4: Performance of ObjectMix for different p. The top row shows the top-1 performance on the validation set and the bottom row shows that on the training set.

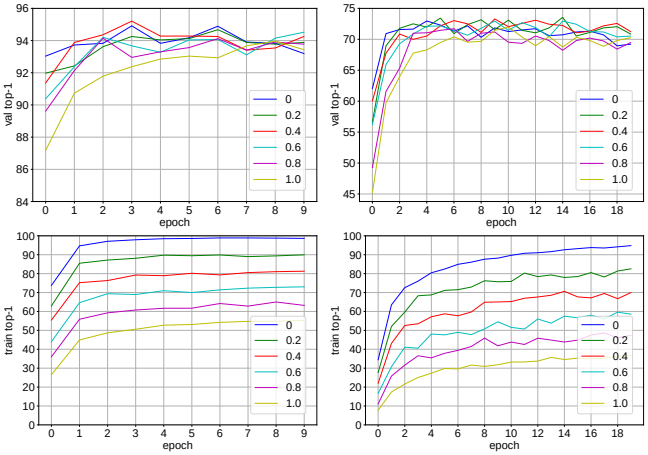
Table 2: The top-1 performance of ObjectMix+or (OM+or) on UCF101 and HMDB51 validation set.

	UCF101		HMDB51	
method (p)	top-1	top-5	top-1	top-5
OM+or (0.0)	93.58 ± 0.03	99.23 ± 0.01	69.86 ± 0.39	91.89 ± 0.26
OM+or (0.2)	94.08 ± 0.18	**99.54** ± 0.07	70.86 ± 0.53	92.49 ± 0.28
OM+or (0.4)	94.19 ± 0.09	99.49 ± 0.06	**71.70** ± 0.62	92.56 ± 0.22
OM+or (0.6)	**94.28** ± 0.22	99.31 ± 0.06	70.88 ± 0.54	**93.20** ± 0.29
OM+or (0.8)	93.67 ± 0.25	99.22 ± 0.05	68.93 ± 0.43	91.75 ± 0.25
OM+or (1.0)	93.25 ± 0.23	99.00 ± 0.07	69.88 ± 0.45	92.08 ± 0.33

equal or better as the training progresses, regardless of the value of p. On the other hand, for $p = 0$, the validation performance begins to deteriorate in the middle stage of the training even though the performance of the training set is consistently high and continues to increase, indicating that overfitting occurs due to the lack of data augmentation. On the other hand, the proposed method suppresses overfitting even with a small amount of augmentation with $p = 0.2$. The training performance degrades as p increases, but this does not have much impact on the validation performance.

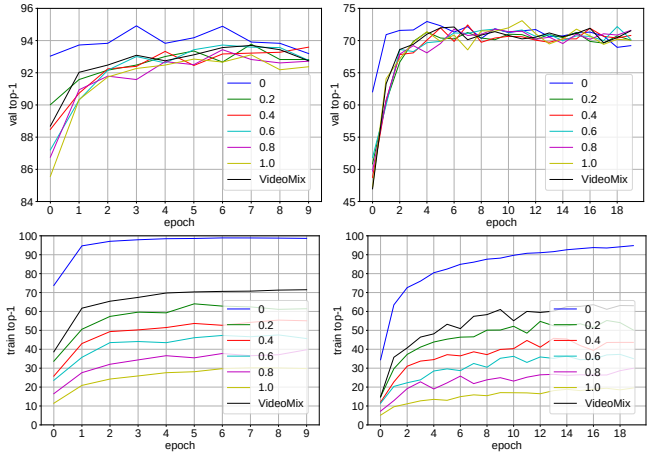
4.4 Results of ObjectMix+or

Next, we show results of ObjectMix+or in Table 2, and the performances over training epochs in Figure 5. This result shows a similar trend of ObjectMix for both data sets; the performance have a peak as p increases. Compared to ObjectMix, the performance on the training set is significantly lower, indicating that ObjectMix+or plays a greater role in regularization as a data augmentation. The validation performances are however similar (or slightly inferior) to those of ObjectMix, which may shows that frame-wise segmentation masks of ObjectMix might be enough even with failure makes in some frames.



(a) UCF101 (b) HMDB51

Figure 5: Performance of ObjectMix+or for different p.



(a) UCF101 (b) HMDB51

Figure 6: Performance comparisons of ObjectMix for different p and VideoMix.

4.5 Comparison with VideoMix

Here we report the effect of using the proposed method in combination with VideoMix. The settings followed the original paper [40], and a patch with center coordinates (w_c, h_c) was sampled as follows;

$$\lambda \sim \mathrm{Beta}(\alpha, \alpha), \quad \alpha = 1 \tag{9}$$

$$w_c \sim \mathrm{Unif}(0, W), \quad W = 224 \tag{10}$$

$$h_c \sim \mathrm{Unif}(0, H), \quad H = 224 \tag{11}$$

$$w_1 = \max\left(0, w_c - \frac{W\sqrt{\lambda}}{2}\right), \quad w_2 = w_c + \frac{W\sqrt{\lambda}}{2} \tag{12}$$

$$h_1 = h_c - \frac{H\sqrt{\lambda}}{2}, \quad h_2 = h_c + \frac{H\sqrt{\lambda}}{2}. \tag{13}$$

We used S-VideoMix, which uses the same spatial patch across all frames, and set the probability to $p = 1$ according to the original paper. The results are shown in Table 3 and Figure 6.

The combination of ObjectMix and VideoMix shows a significant performance degradation compared to using ObjectMix alone. One reason might be the fact that the size of the extracted mask region can bee too large by merging two regions from both ObjectMix and VideoMix. A possible improvement is to search reasonable parameters of VideoMix, taking into account the mask size issue when used combined with ObjectMix.

The performance of ObjectMix alone outperforms that of VideoMix alone shown here for any p value, indicating that the proposed method is more effective.

Table 3: The top-1 performance of the combination of ObjectMix+or (OM+or) and VideoMix (VM) on UCF101 and HMDB51 validation set.

	UCF101		HMDB51	
method (p)	top-1	top-5	top-1	top-5
OM+or (0.0)	93.58 ± 0.03	99.23 ± 0.01	69.86 ± 0.39	91.89 ± 0.26
VM	93.34 ± 0.07	99.38 ± 0.00	70.90 ± 0.34	92.84 ± 0.24
OM+or (0.2) +VM	93.15 ± 0.20	99.26 ± 0.06	69.83 ± 0.47	92.10 ± 0.35
OM+or (0.4) +VM	**93.77** ± 0.16	**99.41** ± 0.09	70.49 ± 0.54	92.09 ± 0.25
OM+or (0.6) +VM	92.32 ± 0.26	99.07 ± 0.06	70.74 ± 0.56	**92.68** ± 0.32
OM+or (0.8) +VM	92.92 ± 0.25	99.19 ± 0.09	**71.35** ± 0.29	92.54 ± 0.17
OM+or (1.0) +VM	92.72 ± 0.19	99.04 ± 0.08	70.15 ± 0.46	92.52 ± 0.24

Table 4: Summary of the proposed method (fixed to $p = 0.6$) and comparisons of data augmentation (upper rows) with self-supervised methods (lower rows).

	UCF101	HMDB51
method	top-1	top-1
no augmentation	93.58 ± 0.03	69.86 ± 0.39
OM (0.6)	95.12 ± 0.15	71.44 ± 0.59
OM+or (0.6)	94.28 ± 0.22	70.88 ± 0.54
VideoMix [40]	93.4	66.9
CMD [14]	85.7	54.0
MoCo+BE [33]	87.1	56.2
CVRL [24]	94.4	70.6
VideoMAE [29]	96.1	73.3
ρBYOL [9]	96.3	75.0

5 CONCLUSION

In this paper, we proposed ObjectMix, a data augmentation for action recognition, which uses object masks extracted from input video frames. The proposed method differs from mix-based VideoMix; ObjectMix creates new videos by extracting objects rather than cutting and pasting random rectangles. Experiments using UCF101 and HMDB51 have confirmed that the proposed method is effective to suppress overfitting. The reasonable value of p was about 0.5 for both datasets, which might be a good balance of the original and generated video samples, and we will verify the results by using other datasets. An obvious limitation of the proposed method is its computational cost. While random rectangle cropping is nearly zero-cost, applying instance segmentation is computationally expensive in both space and time, hindering an efficient model training. A less accurate but lighter model could be used to speed up mask extraction because the accuracy of segmentation might have a

small impact on the classification performance. Therefore, a future work includes verification of the trade-off between performance and cost with such an efficient model.

Another topic of the future work is the comparison with self-supervised learning for video representation. Table 4 summarizes our results and compares the performance with the recent self-supervised methods. This result shows that self-supervised learning is becoming more effective than supervised learning with data augmentation. Future work includes further investigation of how to combine the proposed segmentation-based augmentation method with self-supervised learning [32].

ACKNOWLEDGMENTS

This work was supported in part by JSPS KAKENHI Grant Number JP22K12090.

REFERENCES

[1] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. 2020. Albumentations: Fast and Flexible Image Augmentations. *Information* 11, 2 (2020). https://doi.org/10.3390/info11020125

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. 2012. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV) (Part IV, LNCS 7577)*, A. Fitzgibbon et al. (Eds.) (Ed.). Springer-Verlag, 611–625.

[3] Terrance Devries and Graham W. Taylor. 2017. Improved Regularization of Convolutional Neural Networks with Cutout. *CoRR* abs/1708.04552 (2017). arXiv:1708.04552 http://arxiv.org/abs/1708.04552

[4] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. 2015. FlowNet: Learning Optical Flow with Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV)*. http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15

[5] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. 2017. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[6] Hajar Emami, Ming Dong, Siamak P. Nejad-Davarani, and Carri K. Glide-Hurst. 2021. SA-GAN: Structure-Aware GAN for Organ-Preserving Synthetic CT Generation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, Marleen de Bruijne, Philippe C. Cattin, Stéphane Cotin, Nicolas Padoy, Stefanie Speidel, Yefeng Zheng, and Caroline Essert (Eds.). Springer International Publishing, Cham, 471–481.

[7] Christoph Feichtenhofer. 2020. X3D: Expanding Architectures for Efficient Video Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. SlowFast Networks for Video Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[9] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. 2021. A Large-Scale Study on Unsupervised Spatiotemporal Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3299–3309.

[10] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. 2021. Simple Copy-Paste Is a Strong Data Augmentation Method for Instance Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2918–2928.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[12] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. 2017. The "Something Something" Video Database for Learning and Evaluating Visual Common Sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[13] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2018. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6546–6555.

[14] Lianghua Huang, Yu Liu, Bin Wang, Pan Pan, Yinghui Xu, and Rong Jin. 2021. Self-Supervised Video Representation Learning by Context and Motion Decoupling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 13886–13895.

[15] Matthew S. Hutchinson and Vijay N. Gadepally. 2021. Video Action Understanding. *IEEE Access* 9 (2021), 134611–134637. https://doi.org/10.1109/ACCESS.2021.3115476

[16] Yasunori Ishii and Takayoshi Yamashita. 2021. CutDepth: Edge-aware Data Augmentation in Depth Estimation. *CoRR* abs/2107.07684 (2021). arXiv:2107.07684 https://arxiv.org/abs/2107.07684

[17] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. 2020. imgaug. https://github.com/aleju/imgaug. Online; accessed 01-Feb-2020.

[18] Euijin Jung, Miguel Luna, and Sang Hyun Park. 2021. Conditional GAN with an Attention-Based Generator and a 3D Discriminator for 3D Medical Image Generation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, Marleen de Bruijne, Philippe C. Cattin, Stéphane Cotin, Nicolas Padoy, Stefanie Speidel, Yefeng Zheng, and Caroline Essert (Eds.). Springer International Publishing, Cham, 318–328.

[19] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The Kinetics Human Action Video Dataset. *CoRR* abs/1705.06950 (2017). arXiv:1705.06950 http://arxiv.org/abs/1705.06950

[20] Nour Eldeen Mahmoud Khalifa, Mohamed Loey, and Seyedali Mirjalili. 2022. A comprehensive survey of recent trends in deep learning for digital images augmentation. *Artificial Intelligence Review* 55, 3 (2022), 2351–2377. https://doi.org/10.1007/s10462-021-10066-4

[21] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. *CoRR* abs/1405.0312 (2014). arXiv:1405.0312 http://arxiv.org/abs/1405.0312

[22] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. 2016. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. http://lmb.informatik.uni-freiburg.de/Publications/2016/MIFDB16 arXiv:1512.02134.

[23] Yingxue Pang, Jianxin Lin, Tao Qin, and Zhibo Chen. 2021. Image-to-Image Translation: Methods and Applications. *IEEE Transactions on Multimedia* (2021), 1–1. https://doi.org/10.1109/TMM.2021.3109419

[24] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. 2021. Spatiotemporal Contrastive Video Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6964–6974.

[25] Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* 6 (2019), 60. https://doi.org/10.1186/s40537-019-0197-0

[26] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2014/file/00ec53c4682d36f5c4359f4ae7bd7ba1-Paper.pdf

[27] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR* abs/1212.0402 (2012). arXiv:1212.0402 http://arxiv.org/abs/1212.0402

[28] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N. Metaxas, and Sergey Tulyakov. 2021. A Good Image Generator Is What You Need for High-Resolution Video Synthesis. In *International Conference on Learning Representations*. https://openreview.net/forum?id=6puCSjH3hwA

[29] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training. *CoRR* abs/2203.12602 (2022). https://doi.org/10.48550/arXiv.2203.12602 arXiv:2203.12602

[30] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. 2018. MoCoGAN: Decomposing Motion and Content for Video Generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[31] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. 2017. Learning from Synthetic Humans. In *CVPR*.

[32] Julius von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. 2021. Self-Supervised Learning with Data Augmentations Provably Isolates Content from Style. In *Advances in Neural Information Processing Systems*, Vol. 34. https://proceedings.neurips.cc/paper/2021/hash/8929c70f8d710e412d38da624b21c3c8-Abstract.html

[33] Jinpeng Wang, Yuting Gao, Ke Li, Yiqi Lin, Andy J. Ma, Hao Cheng, Pai Peng, Feiyue Huang, Rongrong Ji, and Xing Sun. 2021. Removing the Background by Adding the Background: Towards Background Robust Self-Supervised Video Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11804–11813.

[34] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-Local Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[35] David S Wishart, Yannick Djoumbou Feunang, Ana Marcu, An Chi Guo, Kevin Liang, Rosa Vázquez-Fresno, Tanvir Sajed, Daniel Johnson, Carin Li, Naama Karu, et al. 2018. HMDB 4.0: the human metabolome database for 2018. *Nucleic acids research* 46, D1 (2018), D608–D617.

[36] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019. Detectron2. https://github.com/facebookresearch/detectron2.

[37] Mingle Xu, Sook Yoon, Alvaro Fuentes, and Dong Sun Park. 2022. A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. *CoRR* abs/2205.01491 (2022). https://doi.org/10.48550/arXiv.2205.01491 arXiv:2205.01491

[38] Jaejun Yoo, Namhyuk Ahn, and Kyung-Ah Sohn. 2020. Rethinking Data Augmentation for Image Super-resolution: A Comprehensive Analysis and a New Strategy. *arXiv preprint arXiv:2004.00448* (2020).

[39] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[40] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, and Jinhyung Kim. 2020. VideoMix: Rethinking Data Augmentation for Video Classification. *CoRR* abs/2012.03457 (2020). arXiv:2012.03457 https://arxiv.org/abs/2012.03457

[41] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=r1Ddp1-Rb