Visually explaining 3D-CNN predictions for video classification with an adaptive occlusion sensitivity analysis

Tomoki Uchiyama [1] Naoya Sogi [1] Koichiro Niinuma[2] Kazuhiro Fukui[1]
[1]University of Tsukuba [2]Fujitsu Research of America
{uchiyama, sogi}@cvlab.cs.tsukuba.ac.jp, kniinuma@fujitsu.com, kfukui@cs.tsukuba.ac.jp

Abstract

This paper proposes a method for visually explaining the decision-making process of 3D convolutional neural networks (CNN) with a temporal extension of occlusion sensitivity analysis. The key idea here is to occlude a specific volume of data by a 3D mask in an input 3D temporal-spatial data space and then measure the change degree in the output score. The occluded volume data that produces a larger change degree is regarded as a more critical element for classification. However, while the occlusion sensitivity analysis is commonly used to analyze single image classification, it is not so straightforward to apply this idea to video classification as a simple fixed cuboid cannot deal with the motions. To this end, we adapt the shape of a 3D occlusion mask to complicated motions of target objects. Our flexible mask adaptation is performed by considering the temporal continuity and spatial co-occurrence of the optical flows extracted from the input video data. We further propose to approximate our method by using the first-order partial derivative of the score with respect to an input image to reduce its computational cost. We demonstrate the effectiveness of our method through various and extensive comparisons with the conventional methods in terms of the deletion/insertion metric and the pointing metric on the UCF101. The code is available at: `https://github.com/uchiyama33/AOSA`.

1. Introduction

This paper proposes an adaptive occlusion sensitivity analysis for visualizing and understanding the decision-making process of 3D convolutional neural networks (CNN) [15, 33] for video classification. Our occlusion sensitivity analysis provides informative sensitivity maps that indicate which parts of an input video are more important for explaining 3D-CNN predictions.

With increased attention to the high ability of deep neural networks (DNN), how to explain the DNN predictions has become a fundamental problem in both theoretical study and practical applications, particularly in serious tasks directly related to human life such as medical diagnosis and accident investigation of an autonomous vehicle [32]. Many types of methods have been proposed to provide such an explanation, mainly for CNN taking a single image as input [26, 35, 22, 9], as will be surveyed in the next section.

In this paper, of these, we focus on a method using the occlusion sensitivity analysis (OSA) [35], which has been widely used for visually explaining CNN predictions due to its simple idea and expandability. Our basic idea motivated by the OSA is to occlude a specific volume of data by a 3D mask in an input 3D temporal-spatial data space and then measure the change degree in the output score as shown in Fig.1. We regard an occluded volume that produces a more significant change in the score as more critical data for the predictions.

This idea is simple and easy to implement. However, it is not straightforward to incorporate the idea of OSA into the 3D-CNN network architecture since OSA has been originally developed to explain the prediction process of the standard 2D CNN that takes not a video but a single image as an input. For example, one may come up with a way of extending a rectangle mask along the temporal direction to a cuboid mask in an input 3D spatio-temporal space as a simple extension of the OSA. However, this naive approach does not work as expected when a target object moves, as a fixed cuboid mask cannot continue to occlude the moving object throughout the video.

To address this issue, we propose to change the shape of a 3D occlusion mask adaptively to the complicated motions of a target object, considering the temporal continuity and spatial co-occurrence of the motions. To this end, we first extract the optical flows [7, 5, 21] from an input video and then move each occlusion mask set to an attention region detected in the first frame of the video. In this way, the movement trace of an occlusion mask forms a blended tube mask with a rectangular shape, as shown in Fig. 2. We call this 3D occlusion mask a spatio-temporal occlusion mask Ω. This 3D mask is the base of our method.

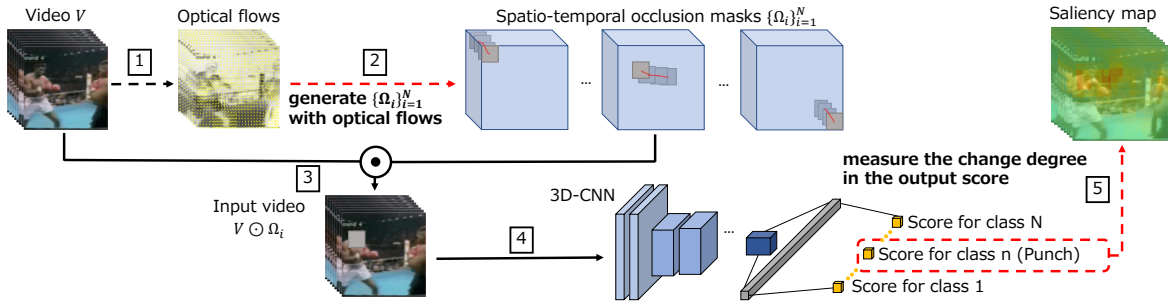

Figure 1: The framework of Adaptive Occlusion Sensitivity Analysis (AOSA). Our approach tracks objects using optical flow so that a spatio-temporal occlusion mask can occlude the same object throughout the video. A saliency map is generated by measuring the change degree in the output score when the occluded video is input to the 3D-CNN.

We need to consider multiple spatio-temporal occlusion masks to handle multiple objects with more complicated motions simultaneously. For this purpose, we group multiple occlusion masks with similar positions and motions into the same class and then integrate a set of masks into one mask $\hat{\Omega}$ with larger volume size. In this way, we generate a more effective 3D mask. We call this new analysis *Adaptive Occlusion Sensitivity Analysis* ($AOSA$).

Furthermore, we discuss two valid options for further enhancing our analysis method. First, we describe applying the conditional sampling method [38] to effectively fill the intensity values in each occlusion mask instead of using a constant value. Next, we describe the computational cost of our method. Although OSA is essential for constructing our method, it causes an additional problem of high computational cost due to conducting OSA over a whole video many times. We introduce an approximation computation of the change score to reduce the cost. Considering a 3D-CNN as a certain function that transforms an input video to a class score, we approximate the function in the first order using its Taylor expansion, where the first-order partial derivative of the function with respect to an input image and the derivate can be obtained by the automatic differentiation through back-propagation.

In the experiment section, we first conduct a qualitative evaluation of our sensitive map in comparison with that of various conventional methods. Next, we compare the effectiveness of our method with the conventional methods in terms of the deletion/insertion metric and the pointing metric [22], which have been widely used as valid indexes, on the video classification on UCF101 [29].

Our main contributions are summarized as follows.

(1) We propose an adaptive occlusion sensitivity analysis (AOSA) for explaining 3D CNN predictions.

(2) We introduce an approximation computation for calculating the change degree in the class score to reduce the high computational cost of conducting the sensitivity analysis.

(3) We demonstrate the effectiveness of our method through an extensive comparison with the conventional methods in terms of the deletion/insertion metric and the pointing metric on the UCF101.

The paper is organized as follows. In Section 2, we describe the related methods. In Section 3, we describe the proposed method. First, we describe the basic idea of our adaptive occlusion sensitivity analysis. Then, we construct a method for explaining 3D CNN. Further, we introduce an approximate computation of occlusion sensitivity analysis using Taylor expansion. In Section 4, we demonstrate the effectiveness of our method through evaluation experiments. Finally, Section 5 concludes the paper.

2. Related Work

2.1. Explanation of 2D CNN predictions

Most methods provide the information for explaining the CNN predictions, that is, the decision-making process by a saliency map indicating which elements/regions of an input image contribute to the output score. For 2D-CNN taking a single image as an input, many types of methods for generating such a saliency map have been proposed [28, 23, 19, 17, 8, 1, 2, 24, 36]. These methods are categorized into three types of methods: perturbation-based methods, activation-based methods and gradient-based methods.

As a simple yet effective perturbation-based method, we review a method using the occlusion sensitivity analysis (OSA). The idea of the OSA-based method is straightforward. First, it measures the slight variation of the class score to occlusion in different regions of an input image using small perturbations of the image. Then, the resultant variation of each region is summarized as a saliency map called a sensitivity map of the input image. In the sensitivity map, the local image regions with significant variation are emphasized as the part that positively contributes to

the class score. Accordingly, the occlusion sensitivity map can provide helpful information for understanding what image features contribute to a final decision and further implies why the network fails the classification. Meaningful Perturbation [10] and Extremal Perturbation [9] generate a saliency map opposite the OSA. These methods occlude the elements/regions with fewer contributions while remaining the elements/regions with significant contributions.

Grad-CAM [26] has been well known as one of the popular activation-based methods. The Grad-CAM generates a saliency as the weighted sum of the convolutional feature maps, where the gradient is used as the weight of each feature map.

Although the Grad-CAM has been widely used for explaining CNN predictions, it has a limitation in spatial resolution since the spatial resolution of the Grad-CAM can be determined by the low resolution of the last layer. For example, the resolution of Grad-CAM is only 7-by-7 pixels when using the GooglNet. Thus, the resolution of a saliency map from Grad-CAM is usually much lower than an occlusion map.

For the gradient-based methods, Deep LIFT [27] and Guided Backprop [30] have been proposed for explaining CNN predictions. These methods obtain the contribution of each input element by applying back-propagation. They can provide a pixel-wise fine saliency map. However, it is often difficult to visually understand the meaning of the map.

2.2. Extensions for 3D-CNN predictions

There are a few methods for explaining the decision-making process of 3D-CNN taking videos as input. A naive approach is applying methods such as OSA and Grad-CAM to 3D-CNN without any modification just by replacing 2D-matrix with 3D-tensor data as an input. For example, an extension of Grad-CAM, Grad-CAM++ [6], has been applied to 3D-CNN for explaining the process of action recognition. However, such naive approaches cannot work well, as will be demonstrated in the experimental section, as they do not have a mechanism for explicitly handling the temporal relationship in video data.

Several methods considering the temporal relationship have been proposed. Saliency tube [31] generates a spatio-temporal saliency map as an extension of CAM [37], where the feature map of the last layer is weighted with the coefficients from the prediction layer. It has been shown that the separation of the information from Grad-CAM into spatial and temporal information works effectively for a more detailed explanation [14]. SWAG-V [13] enhances the framework SWAG [12] by averaging and smoothing a saliency map at the super-pixel level.

Like our approach, Saptio-Temporal Extremal Perturbation (STEP) [18] is also a perturbation-based method. This method tackles how to handle the temporal relationship effectively, unlike other methods, which simply apply existing techniques designed for 2D-CNN. STEP extends the framework of Extremal Perturbation [9] to handle the task of explaining 3D-CNN predictions. The key idea here is to solve an optimization problem with the smoothness constraint on successive temporal saliency maps. Although STEP can use the temporal information well, its computational cost is high due to the heavy optimization. Besides, STEP tends to be affected by some noise to output an unclear map.

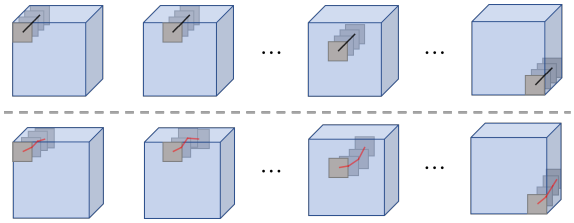

Figure 2: A simple extension of OSA for 3D (top) and the proposed method (bottom). The simple 3D OSA occludes a cuboid. On the other hand, the proposed method occludes a curved tubular rectangular shape (spatio-temporal occlusion mask) along the displacement vectors of the optical flows.

3. Proposed Method

In this section, we extend the idea of OSA to a visual explanation of 3D-CNN predictions. In a 3D spatio-temporal data composed by an input video, the temporal direction is simultaneously occluded in addition to the spatial direction.

3.1. Generating masks with optical flows

As mentioned in Section 1, a simple extension of OSA is to occlude a part of the 3D spatio-temporal data with a fixed cuboid, as shown in the top of Fig. 2. Unfortunately, this method occludes a different object in each frame when the objects are moving. Therefore, it is difficult to produce a meaningful visualization map. To overcome this limitation, the proposed method generates occlusion masks while following the motion of the target object based on the optical flow [7, 5, 21], as shown in the bottom of Fig. 2.

Let an input color video with T frames be $V \in \mathbb{R}^{T\times H\times W\times 3}$ and $I_t \in \mathbb{R}^{H\times W\times 3}$ be the t frame image. Occlusion masks $\{M_i^t\}$ are generated by the following process:

(1) N anchor points $\{p_i^{(1)} \in \mathbb{R}^2\}_{i=1}^N$ for I_1 are equally spaced vertically and horizontally every s pixels across the entire screen.

(2) A rectangle mask $M_i^1 \in \{0,1\}^{H\times W\times 3}$ is set up to occlude a $h \times w$ region centered at each anchor point $p_i^{(1)}$. The masked region of M_i^t is set to 0 and the other regions are set to 1.

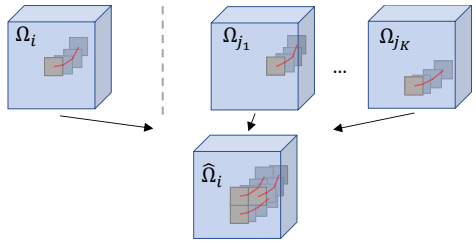

Figure 3: Generation of integration mask. The highly co-occurring masks with a masks Ω_i are selected by Eq 1. Then, the integrated mask $\hat{\Omega}_i$ is calculated by the element-wise product among Ω_i and $\{\Omega_{j_k}\}_{j_k \in \mathcal{K}_i}$.

(3) In the image of the second frame I_2, each anchor point $p_1^{(2)}, \ldots, p_N^{(2)}$ is moved by the optical flow.

(4) A rectangle mask $M_i^2 \in \{0,1\}^{H \times W \times 3}$ is set up to occlude a $h \times w$ region centered at each anchor point $p_i^{(2)}$.

By repeating the above process until the final frame T, we obtain $T \times N$ rectangle masks $\{M_i^t\}_{i,t}$. Each spatio-temporal occlusion mask $\Omega_i \in \{0,1\}^{T \times H \times W \times 3}$ is generated by arranging rectangle masks $\{M_i^t\}_t$ over the temporal domain, resulting in that N masks $\{\Omega_i\}_{i=1}^N$ being obtained. A spatio-temporal occlusion mask represents the moving of an anchor point or an object.

When the output score changes significantly by applying a spatio-temporal occlusion mask Ω_i, the corresponding occluded regions are visualized as important for a 3D-CNN decision.

3.2. Consideration of co-occurrence

In the previous Section 3.1, N spatio-temporal occlusion masks $\{\Omega_i\}$ are applied independently, and the change degrees in the output scores are measured without considering their relationship. However, spatio-temporal occlusion masks capturing the same object's motion have co-occurrence, which is essential information for the classification. Therefore, the influence of highly co-occurred motions on the classification should be investigated. To this end, we measure the degree of co-occurrence and then integrate masks with highly co-occurred motions into a single integrated mask $\hat{\Omega}$, as shown in Fig.3.

■ Co-occurrence between occlusion masks: There are many ways to measure degrees of co-occurrence between spatio-temporal occlusion masks. Since the proposed method uses optical flow for generating masks while considering motion, we measure co-occurrence based on the variation pattern of displacement vectors $\{v_i\}$ obtained by optical flow, i.e., movement of anchor points $\{p_i^{(t)} \in \mathbb{R}^2\}_t$.

We define the co-occurrence between i' and xth spatio-temporal occlusion masks $\Omega_{i'}, \Omega_x$ as follows:

$$Co(\Omega_{i'}, \Omega_x) = v_{i'} \cdot v_x / (\|v_{i'}\| \|v_x\|), \tag{1}$$

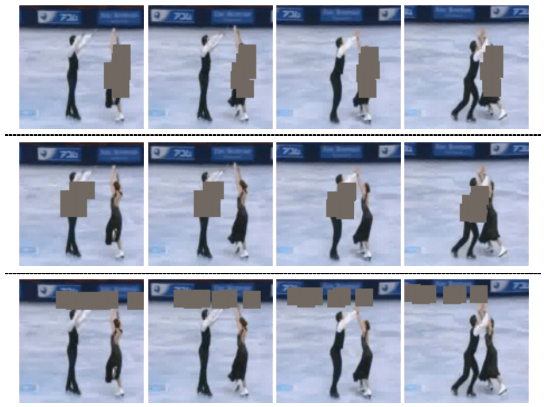

Figure 4: A video with integration masks ($K = 5$). Each row shows the occluded frames by an integrated mask.

where $v_{i'} = [(p_{i'}^{(2)} - p_{i'}^{(1)})^\top, \ldots, (p_{i'}^{(T)} - p_{i'}^{(T-1)})^\top]^\top \in \mathbb{R}^{2(T-1)}$ is a displacement vector.

■ Mask integration: We integrate spatio-temporal occlusion masks by the co-occurrence Co, i.e., we integrate the ith mask Ω_i with the K highly co-occurring masks $\{\Omega_j\}_{j \in \mathcal{K}_i}$. We apply the element-wise product among Ω_i and $\{\Omega_j\}_{j \in \mathcal{K}_i}$ to generate the integration mask $\hat{\Omega}_i$, such that all target regions by the masks are occluded.

Finally, we apply occlusion by calculating the element-wise product to the integrated mask $\hat{\Omega}_i$ and the input video V, i.e. $\hat{\Omega}_i \odot V$. Then, we input the occluded video into a classification model and measure the output score of the target class. We estimate that the occluded regions are important for the classification if the output score becomes low. With reference to [22], the visualization map S is given as the weighted sum of the mask $\hat{\Omega}_i$ with weights by the output score $f(\hat{\Omega}_i \odot V)$ as follows:

$$S = \frac{1}{N} \sum_{i=1}^{N} f(\hat{\Omega}_i \odot V) \cdot \hat{\Omega}_i, \tag{2}$$

where f is a classification model.

In the following, we refer to this method as Adaptive Occlusion Sensitivity Analysis (AOSA).

Fig. 4 shows examples of a video with integration masks. We can see that the same objects like people can be occluded, by using the co-occurrence.

3.3. Dealing with frame-outs

The proposed method uses optical flows to track the anchor points, but the anchor points sometimes disappear from the screen in the middle of the video. The main examples are situations in which the camera moves or the attention object moves off the screen. Therefore, if an anchor point moved by optical flow is outside the screen, the tracking of the anchor point is stopped at that frame, and no further occlusion is made.

3.4. Conditional sampling

Instead of filling up occluded regions with a uniform value, conditional sampling [38] introduced the calculation method of replaced values v of occluded regions to OSA. The replaced value is randomly sampled from the normal distribution, whose parameters are estimated from patches around the occluded region.

In the conditional sampling, it first extracts patches $\{\hat{\mathbf{x}}_i^j\}$ from around an anchor point p_i, where each patch includes the pixel indicated by the p_i, and its size is same as the occluded regions. Then, the mean μ and variance Σ are calculated from the patches without original occlusion patch. A replaced value is randomly sampled from $\mathcal{N}(\mu, \Sigma)$. A visualization map is obtained by using an average output score with multiple randomly sampled values. An average score can be written as $\sum_{v_i} q(v_i|\mu, \Sigma) f(x')$, where $q(v_i|\mu, \Sigma)$ is the probability density function, x' is the occluded input with a replaced value $v_i \in \mathbb{R}^{h \times w}$, and f is a neural network.

Conditional sampling make sensitivity analysis more stable [38]. However, if this process is applied to the proposed method directly, the computational cost dramatically increases, as the number of inference of a network increases in proportion to the number of random sampling. Thus, we reduce the computational cost by approximating the inference of the network.

3.5. Speedup by approximation

The disadvantage of sensitivity analysis and the conditional sampling is that the computational cost is high because it is necessary to prepare many occluded inputs and use a neural network to infer them. Here, we assume that the change to the input caused by occlusion is small, as occluded region is typically small. This assumption motivates us to approximate the inference of the neural network by a first-order Taylor expansion. This enables fast computation of the output values of the deep neural network, resulting in that the computational cost of OSA and Conditional sampling can be reduced. As shown later, this approximation makes the computational cost of Conditional sampling almost the same as that of OSA.

3.5.1 First-order approximation of OSA

Let $f : \mathcal{X} \to \mathbb{R}$ be a function from the input space $\mathcal{X}$ to the output score $\mathbb{R}$ by a neural network. In the following, we assume that the input is a vector image x of dimension $H \times W$. When a small value δ_x is added to the input x, the first-order approximation by Taylor expansion of $f(x + \delta_x)$ in the neighborhood of x is as follows:

$$f(x + \delta_x) \simeq f(x) + \left(\frac{\partial f}{\partial x}|_x\right)^T (x + \delta_x - x). \quad (3)$$

From equation (3), we can approximate the inference of a neural network by computing $f(x)$ and the partial derivative $\frac{\partial f}{\partial x}|_x$ of f with respect to x. $\frac{\partial f}{\partial x}|_x$ is obtained by back-propagating [25] the output score to the input x after $f(x)$ is computed by the neural network. Thus, the approximation of $f(x + \delta_x)$ can be computed with a single forward $f(x)$ and the corresponding backward pass. This back-propagation can be easily computed using the automatic differentiation with deep learning libraries such as PyTorch [20].

Next, we explain how to compute the approximate inference of the occluded input by the neural network using equation (3). Let m be the mask, and v be a matrix including replaced values. The occluded input can be represented as $g(x) = x \odot m + (1 - m) \odot v$. Therefore, the inference of the occluded input by the neural network can be represented as $f(g(x))$. From the equation (3), the first-order approximation of $f(g(x))$ is as follows:

$$f(g(x)) \simeq f(x) + \left(\frac{\partial f}{\partial x}|_x\right)^\top (g(x) - x) = \hat{f}(g(x)). \quad (4)$$

Finally, the importance S_m of the occluded region by m is the difference from the output score of the original input as follows:

$$S_m = f(x) - \hat{f}(g(x)). \quad (5)$$

3.5.2 Approximation using conditional sampling

According to [38], the equation (5) for the conditional sampling is as follows:

$$S_m = f(x) - \sum_{v_i} q(v_i) f(g(x; m, v_i)). \quad (6)$$

Here, since $g(x; m, v_i) - x = (1 - m) \odot (v_i - x)$, the first-order approximation of equation (6) is as follows:

$$\begin{aligned}
&f(x) - \Sigma_{v_i} q(v_i) f(g(x; m, v_i)) \\
&\simeq f(x) - \left\{ f(x) + \sum_{v_i} q(v_i) J_x^\top (g(x; m, v_i) - x) \right\} \\
&= -J_x^\top \sum_{v_i} \{q(v_i)(1 - m) \odot (v_i - x)\} \\
&= -J_x^\top (1 - m) \odot \{\sum_{v_i} q(v_i) v_i - x \sum_{v_i} q(v_i)\} \\
&= J_x^\top (1 - m) \odot (x - \mu), \quad (7)
\end{aligned}$$

where $\sum_{v_i} q(v_i) v_i$ is an expectation μ of the probability distribution, and $\sum_{v_i} q(v_i) = 1$. Therefore, the number of the neural networks' inferences does not relate to the number of random sampling. This makes the computational cost of the conditional sampling almost the same as the OSA.

3.5.3 Adjustment of partial derivatives

In the above methods, the first-order approximation in the neighborhood of the original image x has been used. However, from the definition of the Taylor expansion, if difference of $\hat{f}(g(x))$ in Eq. 4 with the original output score $f(x)$ is large, it may have large approximation error. To alleviate this issue, we first find masks having a significant effect on the corresponding outputs, and then adjust the importance of the found masks.

Specifically, we find masks to be adjusted by the following simple outlier detection algorithm. We calculate difference amount $f(x) - \hat{f}(g(x))$ for all masks. Then, we select masks, where the difference amount of each mask is not in the 1.5 $\times$ the interquartile range (IQR).

After that, we recalculate importance scores of the found masks by using the first order approximation at an other partial derivative $J_{g(x)}$. For the masks whose difference amounts are over 1.5 $\times$ IQR, we utilize the occluded input by the mask with the largest difference amount, to calculate $J_{g(x)}$. On the other hand, for the masks whose difference amounts are less than 1.5 $\times$ IQR, we utilize the occluded input by the mask with the smallest difference amount, to calculate $J_{g(x)}$. Note that the smallest value means that the output score significantly smaller than the original score.

This still keeps the number of inferences of the network small but deals with large variations in output scores. We refer to the proposed method, which introduces the above approximation and stabilization techniques, as AOSA-approx.

4. Evaluation Experiments

In this section, we evaluate the proposed methods through the comparison with the conventional explanation methods. To this end, we conduct qualitative evaluation by visualization, and quantitative evaluation using deletion and insertion metrics [22] and S-PT [18].

4.1. Experiments setting

In this experiment, we evaluate the effectiveness of the proposed method through qualitative and quantitative evaluations. We use ResNet50-based R3D [11] and R(2+1)D [34] as classification models. Saliency maps are generated after fine-tuning the networks [16], which is pretrained on the Kinetics-700 [4]. We use the action recognition dataset UCF101 [29] for evaluation. The input videos are clipped and resized to the size $16 \times 112 \times 112$. The evaluation is performed on 3783 videos in the UCF101 test set.

For AOSA, we place the anchor points equally spaced every $s = 8$ pixels in the first frame. In this case, the total number of anchor points is $N = 196 (= (112/8)^2)$. Then, a rectangle occlusion of 16×16 pixels is placed around each anchor point. To verify the effect of mask integration, we also compare the performance of AOSA with a single mask and integrated masks. In the results below, AOSA with a single mask is referred to as AOSA_{SGL}. In addition, for AOSA-approx, patches are extracted from the 36×36 pixels region centered at each anchor point.

We compare the proposed method with Grad-CAM [26], occlusion sensitivity analysis (OSA) [35], and STEP [18]. Grad-CAM and OSA are applied by extending the 2D matrix data of images to the 3D tensor data of videos. For Grad-CAM, we utilize feature maps of the final convolutional layer. For OSA, we set the occlusion size to $8 \times 16 \times 16$ pixels and the interval between occluded regions to every 8 pixels in the spatial direction and every 2 pixels in the temporal direction. For STEP, the visualization method of the video recognition network, we use the default parameters [18].

4.2. Evaluation metrics

We use two types of metrics from different viewpoints for the quantitative evaluation.

First, we use deletion and insertion metrics [22], which evaluates how faithfully the saliency map represents the inferences of the model. Deletion evaluates performance based on how quickly the model prediction probability reduces when pixels are deleted in the order of importance in the saliency map. Conversely, insertion evaluates performance based on how quickly the model prediction probability increases when pixels are inserted in order of importance. Specifically, the both metrics use the area under the curve (AUC) where the horizontal axis is the percentage of pixels deleted or inserted, and the vertical axis is the output probability. In this experiment, pixel deletion and insertion are performed in the same 28 iterations as in [13].

Second, we also use the spatial pointing game (S-PT) metric [18, 3] that evaluates how well the explanation matches the human interpretation. For this evaluation, we use UCF101-24 [29], which is part of UCF101 annotated with bounding boxes indicating the area in which humans act. In this experiment, following [18], one hit is recorded when a 7-pixel radius circle centered at the maximum value of the saliency map intersects the bounding box in each frame. The hit rate in the entire dataset is defined as the evaluation score of S-PT. Assuming that the model learns the humans acting as a feature, a good explanation will have a high score value

4.3. Qualitative results

Fig. 5 shows the visualization results of the saliency maps using R3D and R(2+1)D. We select frames 1, 5, 9, 13, and 16 from 16 frames of the video. The left side of the figure shows the results of the SkateBoarding class video in R3D. The right side of the figure shows the results of the CliffDiving class video in R(2+1)D.

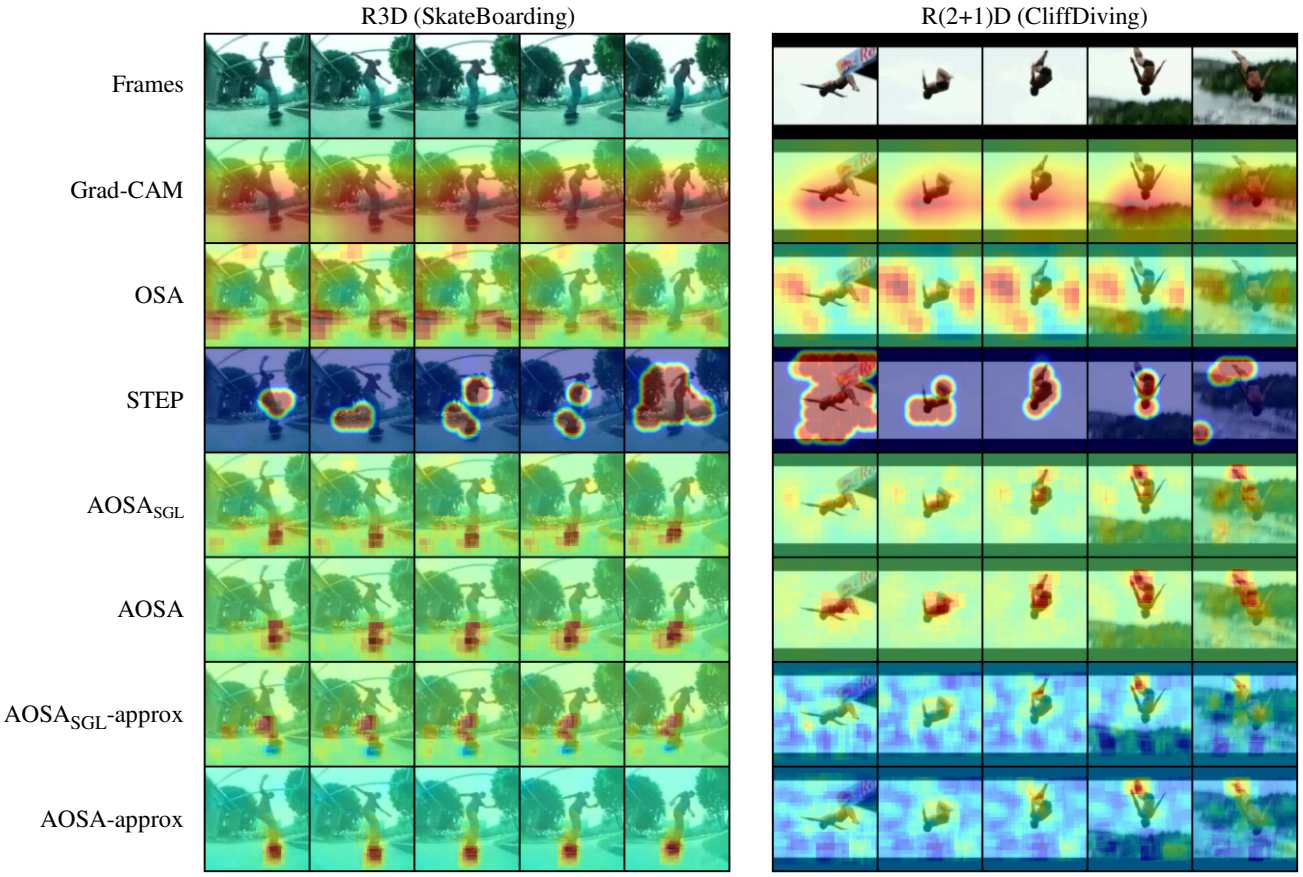

Figure 5: Visualozatoion results of saliency maps. The proposed methods generate stable and smooth saliency maps.

Grad-CAM and OSA cannot represent time-series variations because they do not consider the time series structure of the video. In particular, since the maps from Grad-CAM are resized from a resolution of $1 \times 4 \times 4$, it is not possible to understand the points of focus of the network in detail from those maps. STEP shows some characteristic regions along the time series, but we can see that STEP contains noise.

The proposed method accurately captures class-specific features such as the skateboard and the diving person, and the time-series variation of the maps is smooth. Furthermore, by integrating the masks, the noisy elements are reduced in the map of the SkateBoarding class video. This suggests that the integration of the highly co-occurring masks could fuse meaningful regions, such as the skateboard. region. Besides, we can see that approximate computations are able to indicate important regions, although the responses are somewhat different from those of the original computations. Further examples are shown in the supplementary material.

4.4. Quantitative results

Table 1 shows the evaluation results using the deletion and insertion metrics. According to [13], in deletion, the saliency map that accurately captures important individual pixels is better evaluated, while in insertion, the saliency map that presents cohesive regions that are important is better evaluated. We confirm the same tendency in this experiment. STEP performs well in deletion, and Grad-CAM performs well in insertion. STEP's poor performance in insertion is considered to be that STEP generates saliency maps that are not coherent in space and time. The proposed method shows competitive results in both deletion and insertion metrics than the conventional method. These results show that the proposed method explains the basis of the network predictions accurately and understandably. However, the evaluation by deletion and insertion worsened due to the reduced stability when we introduced approximate calculations.

Then, we show the results for S-PT in Table 2. The proposed methods achieve the best performance among the compared methods in S-PT. In particular, the proposed

Table 1: Deletion and insertion scores on UCF101. For deletion, lower is better (↓). For insertion, higher is better (↑).

	Deletion (↓)		Insertion (↑)	
Method	R3D	R(2+1)D	R3D	R(2+1)D
Grad-CAM [26]	0.203	0.241	0.656	0.687
OSA [35]	0.149	0.204	0.631	0.658
STEP [18]	**0.145**	**0.147**	0.555	0.597
AOSA_{SGL}	0.161	0.189	0.662	0.697
AOSA	0.155	0.180	**0.671**	**0.702**
AOSA_{SGL}-approx	0.181	0.203	0.633	0.633
AOSA-approx	0.196	0.215	0.618	0.624

Table 2: S-PT scores on UCF101-24 for R3D and R(2+1)D.

Method	R3D	R(2+1)D
Grad-CAM [26]	0.688	0.676
OSA [35]	0.659	0.720
STEP [18]	0.705	0.692
AOSA_{SGL}	0.722	0.783
AOSA	0.733	**0.785**
AOSA_{SGL}-approx	**0.735**	0.741
AOSA-approx	0.718	0.736

method performs better than the OSA which is the basis of the proposed method. This result shows that the proposed method can properly capture the variations in the time series. In addition, the approximate computation of the proposed method also performed well. This indicates that the proposed method is able to capture the most important points even in the approximate computation.

4.5. Comparison of generation time

In Table 3, we show the mean computational time to generate a saliency map. In the proposed method, we experimented with $s = 8$ ($N = 196$) and $s = 4$ ($N = 784$) for the interval s of anchor points that controls the map resolution. Although the proposed method requires more processing time than the Grad-CAM, the proposed method is faster than STEP. This is because that although both the proposed method and STEP need hundreds to thousands of inference by the model, STEP requires the gradient computation of the network during optimization unlike the proposed method. Moreover, by approximating the inference of the network, the proposed method can generate a saliency map even faster. This is because the numbers of gradient computations and inferences are significantly small thanks to the approximation compared with STEP.

4.6. Effects of approximate stabilization

In this section, we confirm the effects of conditional sampling and adjustment of partial derivatives introduced in Section 3.5.2 and Section 3.5.3, respectively. Fig. 6 shows the results of AOSA-approx with and without conditional sampling and adjustment of partial derivatives. Each of the two techniques contributes to stabilizing the approximation.

Table 3: Mean computation time for generating a saliency map. These are the results measured in R3D.

Method	Computation time (sec)
Grad-CAM [26]	**0.021**
OSA [35]	4.759
STEP [18]	18.706
AOSA (s=8)	0.728
AOSA-approx (s=8)	0.436
AOSA (s=4)	2.548
AOSA-approx (s=4)	1.624

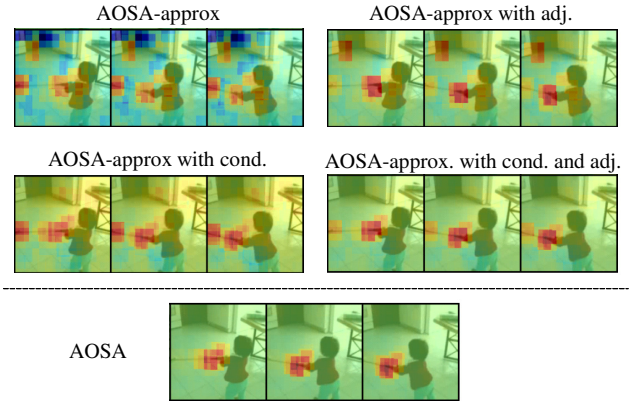

Figure 6: Saliency maps of AOSA-approx with and without conditional sampling and adjustment of partial derivatives.

5. Conclusion

In this paper, we have proposed an adaptive occlusion sensitivity analysis for visually explaining the decision-making process of 3D-CNN in video classification. Our sensitivity analysis taking a video as input is a temporal extension of the previous occlusion sensitivity analysis taking a single image. The novelty of our sensitivity analysis is to change the shape of 3D occlusion map adaptively to the complicated optical flows extracted from an input video. Moreover, we have introduced a method for reducing the computational cost of the sensitivity analysis through its first-order approximation. The results of the evaluation experiment have demonstrated that the proposed method is quantitatively advantageous over the conventional method and qualitatively provides clear explanations that are easy for users to understand.

References

[1] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[2] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

[3] Sarah Adel Bargal, Andrea Zunino, Donghyun Kim, Jianming Zhang, Vittorio Murino, and Stan Sclaroff. Excitation backprop for rnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1440–1449, 2018.

[4] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

[5] Liu Ce. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[6] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, 2018-Janua:839–847, 2018.

[7] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.

[8] Raphael Féraud and Fabrice Clérot. A methodology to explain neural network classification. *Neural networks*, 15(2):237–246, 2002.

[9] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:2950–2958, 2019.

[10] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *IEEE international conference on computer vision*, pages 3429–3437, 2017.

[11] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 3154–3160, 2017.

[12] Thomas Hartley, Kirill Sidorov, Christopher Willis, and David Marshall. SWAG: Superpixels Weighted by average gradients for explanations of CNNs. *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*, pages 423–432, 2021.

[13] Thomas Hartley, Kirill Sidorov, Christopher Willis, and David Marshall. SWAG-V: Explanations for Video using Superpixels Weighted by Average Gradients. *Proceedings - 2022 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022*, pages 1576–1585, 2022.

[14] Liam Hiley, Alun Preece, Yulia Hicks, Supriyo Chakraborty, Prudhvi Gurram, and Richard Tomsett. Explaining Motion Relevance for Activity Recognition in Video Deep Learning Models. 2020.

[15] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.

[16] Hirokatsu Kataoka, Tenga Wakamiya, Kensho Hara, and Yutaka Satoh. Would mega-scale datasets further enhance spatiotemporal 3d cnns? *arXiv preprint arXiv:2004.04968*, 2020.

[17] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 06–11 Aug 2017.

[18] Zhenqiang Li, Weimin Wang, Zuoyue Li, Yifei Huang, and Yoichi Sato. Towards visually explaining video understanding networks with perturbation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1120–1129, 2021.

[19] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, pages 4765–4774. 2017.

[20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Autodiff Workshop*, 2017.

[21] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:6024–6033, 2017.

[22] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *British Machine Vision Conference*, 2018.

[23] Sylvestre-Alvise Rebuffi, Ruth Fong, Xu Ji, and Andrea Vedaldi. There and back again: Revisiting backpropagation saliency methods. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8848, 2020.

[24] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[25] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[26] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via

gradient-based localization. In *IEEE international conference on computer vision*, pages 618–626, 2017.

[27] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.

[28] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*, 2014.

[29] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[30] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015.

[31] Alexandros Stergiou, Georgios Kapidis, Grigorios Kalliatakis, Christos Chrysoulas, Remco Veltkamp, and Ronald Poppe. Saliency tubes: Visual explanations for spatio-temporal convolutions. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1830–1834. IEEE, 2019.

[32] Erico Tjoa and Cuntai Guan. A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813, 2021.

[33] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 International Conference on Computer Vision, ICCV 2015:4489–4497, 2015.

[34] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann Lecun, and Manohar Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[35] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[36] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021.

[37] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:2921–2929, 2016.

[38] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. 2017.