Multimodal Explanations by Predicting Counterfactuality in Videos

Atsushi Kanehira[1], Kentaro Takemoto[2], Sho Inayoshi[2], and Tatsuya Harada[2,3]

[1]Preferred Networks, [2]The University of Tokyo, [3]RIKEN

Abstract

This study addresses generating counterfactual explanations with multimodal information. Our goal is not only to classify a video into a specific category, but also to provide explanations on why it is not categorized to a specific class with combinations of visual-linguistic information. Requirements that the expected output should satisfy are referred to as counterfactuality in this paper: (1) Compatibility of visual-linguistic explanations, and (2) Positiveness/negativeness for the specific positive/negative class. Exploiting a spatio-temporal region (tube) and an attribute as visual and linguistic explanations respectively, the explanation model is trained to predict the counterfactuality for possible combinations of multimodal information in a post-hoc manner. The optimization problem, which appears during training/inference, can be efficiently solved by inserting a novel neural network layer, namely the maximum subpath layer. We demonstrated the effectiveness of this method by comparison with a baseline of the action recognition datasets extended for this task. Moreover, we provide information-theoretical insight into the proposed method.

1. Introduction

The visual cognitive ability of machines has significantly improved mostly due to the recent development of deep learning techniques. Owing to its high complexity, the decision process is inherently a black-box, and therefore, much research has focused on making the machine explain the reason behind its decision to verify its trustability.

The present study particularly focuses on building a system that not only classifies a video, but also explains why a given sample is **not** predicted to one class but another, in spite of almost all existing research pursuing the reason for the positive class. Concretely, we intend to generate the explanation in the form "X is classified to A not B because C and D exists in X."[1] This type of explanation is referred to as *counterfactual explanation* [29] in this paper. It may aid us in understanding why the model prediction is different from what we think, or when discrimination is difficult between two specific classes. The explanation is valuable especially for long videos because it provides information efficiently on the content of what humans cannot see immediately.

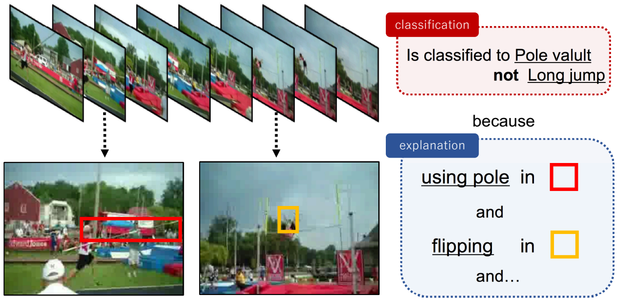

Figure 1: Our model not only classifies a video to a category (Pole vault), but also generates explanations why the video is not classified to another class (Long jump). It outputs several pairs of attribute (e.g., using pole) and spatio-temporal region (e.g., red box) as an explanation.

We first need to discuss the desired output. This work treats an explanation as the one satisfying two conditions as follows:

(A) The output should be interpretable for humans,

(B) The output should have fidelity to the explained target.

Related to (A), one natural way of obtaining interpretable output would be to assign an importance to each element in the input space and visualize it. Although this can help us to perceive the important region for the model prediction, the interpretation leading to it cannot be uniquely determined. We enhance the interpretability of the explanation by leveraging not only parts of visual input but also linguistics, which is compatible with the visual information similar to the previous work [2]. More specifically, dealing with a spatio-temporal region of the target video and (the existence of) an attribute as elements, we concatenate them

This work is done at the University of Tokyo.

[1] rephrased by "X would be classified as B not A if C and D not in X."

to generate explanations. An example is shown in the Fig. 1.

To realize (B) while satisfying (A), the expected output of the visual-linguistic explanation should have the following two properties:

(1) Visual explanation is the region which retains high positiveness/negativeness on model prediction for specific positive/negative classes,

(2) Linguistic explanation is compatible with the visual counterpart.

The score to measure how the requirements above are fulfilled is hereafter referred to as the *counterfactuality score*, or simply *counterfactuality*.

The above-listed requirements cannot be achieved by naively exploiting output of the existing method, which considers only positiveness for explanation, such as in [2], where the authors attempt to generate visual-linguistic explanations of positive class. This is mainly because positiveness/negativeness need to be considered simultaneously for specific positive/negative classes in the same region.

To build a system that generates explanations satisfying both (1) and (2), we propose a novel framework for generating counterfactual explanations based on predicting counterfactuality. The outline of the framework is depicted in two steps: (a) Train a classification model that is the target of the explanation, (b) Train an auxiliary explanation model in a post-hoc manner by utilizing output and mid-level features of the target classifier after freezing its weights to prevent output change. An explanation model predicts counterfactuality scores for all negative classes. It is trained by exploiting the fact that supervised information "X is classified to category A." can be translated into "X is not classified to any category B except A."

The proposed explanation model holds a trainable classifier that predicts simultaneous existence of the pair of class and attribute. Counterfactuality for a specific visual-linguistic explanation (or region-attribute) can be simply calculated by subtracting classifier outputs corresponding to positive/negative classes of interest. When the system outputs the explanation, several pairs of [attribute, region] are selected, whose counterfactuality is large for input positive/negative classes.

Maximization (or minimization) of the prediction score with regard to the region is required during the training/inference process, which is computationally intractable in a naive computation. Under the natural assumption that candidate regions are tube-shaped, the maximum (or minimum) value and corresponding region path can be efficiently computed by dynamic programming. We construct the algorithm such that it can be implemented as a layer of a neural network with only standard functions (e.g., max pooling, relu), pre-implemented in most deep learning libraries [19, 1, 7, 14, 25] by changing the computation order, which enables combining it easily with Convoluional Neural Networks (CNNs) on GPUs.

The proposed simple and intuitive framework for predicting counterfactuality is justified as the maximization of the lower bound of conditional mutual information, as discussed later, providing an information-theoretical point of view toward it.

Moreover, we assigned additional annotations for existing action-recognition datasets to enable quantitative evaluation for this task, and evaluated our method utilizing the created dataset.

The contributions of this work are as follows:

- Introduce a novel task, which is generating counterfactual explanations with spatio-temporal region,
- Propose a novel method based on simultaneous estimations of visual-linguistic compatibility and discriminativeness of two specific classes,
- Propose a novel neural network layer for efficiently solving the dynamic programming problem which appears during training/inference procedures,
- Derive a connection between the proposed method and conditional mutual information maximization, providing better understanding of the proposed model from the information-theoretical viewpoint,
- Propose a metric as well as extending datasets for this task, enabling quantitative evaluation,
- Demonstrate the effectiveness of the proposed approach by experiment.

2. Related Work

We divide existing research on visual explanation into two categories, i.e., justification, or introspection.

Methods for justification are expected to explain as humans do, by projecting input to *correct* reason obtained from the outside [20]. Methods exploiting textual [13] or multimodal [18, 2, 16] supervision belong to this category. True explanations, as by humans, are expected to be generated regardless of the type of model, and evaluation is performed by comparison with ground-truth supervision.

In the latter category, the main goal is to know where the model actually "looks" in the input by propagating the prediction to the input space [22, 4, 30, 21, 31, 9, 32], or by learning instance-wise importance of elements [6, 8] with an auxiliary model. As opposed to the methods of the former category, it is important to show the region where the model focuses for prediction rather than whether the prediction is true for humans. The evaluation is often performed by the investigating before/after output of the model when the element considered to be important is changed.

Although almost all previous research pursues the reason for positiveness, we attempt to provide the reason for negativeness as well. While our work is categorized to the latter, it also has an aspect of the former; The important region for the model prediction is outputted, while exploiting the linguistic attribute for enhancing interpretability for human.

[2] conducted research similar to the present study, stating an application for grounding visual explanation to counterfactual explanation, where the textual explanations (without the region) are generated by comparing the output of generated explanations for target sample and the nearest sample to it. Because their work is in the former category where negativeness cannot be well-defined, no quantitative evaluation was provided. The main differences between our work and this previous study are:

- We set generating counterfactual explanations as the main goal, and propose a method which utilizes multi-modal information specifically for this task,
- Our work belongs to the latter category where negativeness can be well-defined by model output, and therefore, quantitative evaluation is possible,
- We provide quantitative evaluation with the metric.

3. Method

We describe the details of our proposed method in this section. The main goal of this work is to build a system that not only classifies a video, but also explains why a given sample is not predicted to one class but another. As stated earlier, we utilize the combination of the spatio-temporal region (tube) and the attribute as the element of explanations. The expected output explanation should have following two properties:

(1) Visual explanation is the region which retains high positiveness/negativeness on the model prediction for specific positive/negative classes.

(2) Linguistic explanation is compatible to the visual counterpart.

First, we formulate the task addressed in subsection 3.1, and describe the outline of the framework as well as the actual training/inference process of the explanation model from subsection 3.2 to 3.4. In the subsequent subsection 3.5 and 3.6, we elucidate the method and its implementation for efficiently solving the optimization problem in the training/inference step. Theoretical background of our method will be discussed in subsection 3.7.

3.1. Task formulation

Notations used throughout this paper and formulation of the task we deal with are described in this subsection.

Let $\mathbf{x} \in \mathbb{R}^{W' \times H' \times T' \times 3}$ be the input video where W', H', T' are width, height, and the number of frames of the video, respectively. We denote the class and the attribute by $c \in \mathcal{C}$ and $s \in \mathcal{S}$, respectively. We assume that attributes are assigned to each sample used for training, and that the assigned set of attributes is represented as $\mathcal{S}(\mathbf{x}) \subset \mathcal{S}$. $\mathbf{R} \subset \mathcal{R}$ denotes the spatio-temporal region used for visual explanation, and its element $[i, j, t, \mathrm{scale}, \mathrm{shape}] \in \mathbf{R}$ is the spatio-temporal coordinate, scale, and shape of the element of visual region, respectively. $\mathcal{R}$ is a possible set of $\mathbf{R}$. In this work, we particularly limit $\mathcal{R}$ to the set containing all possible tubes. In other words, $\mathbf{R}$ contains at most one element corresponding to the time step t, and all the elements are spatially and temporally continuous.

We build an explainable model, having the following two functions: (a) Classify the video $\mathbf{x}$ to a specific class $c_{\mathrm{pos}} \in \mathcal{C}$, (b) Explain the reason for specific negative class $c_{\mathrm{neg}} \in \mathcal{C} \backslash c_{\mathrm{pos}}$ by the combination of attribute s (linguistic) and spatio-temporal tube $\mathbf{R}$ (visual). Our model predicts several pairs of $(s, \mathbf{R})$ for specific class pair $c_{\mathrm{pos}}, c_{\mathrm{neg}}$, and simply puts them together as final output.

3.2. System pipeline

We outline the pipeline of the proposed method in Fig. 2. Our model holds two modules, namely, the classification module and the explanation module. The outline of the framework follows two steps: (a) Train a classification model, which is the target of the explanation, (b) Train an auxiliary explanation model in a post-hoc manner as in existing research (e.g.,[13]) by utilizing output and mid-level activation of the target classifier after freezing its weights to prevent change in output. Specifically, we explicitly represent feature extraction parts in the pre-trained classification network as

$$p(c|\mathbf{x}) = f(\mathbf{h}(\mathbf{x})),\ \mathbf{h}(\mathbf{x}) \in \mathbb{R}^{W \times H \times T \times d}, \tag{1}$$

where W, H, T indicate width, height, and the number of frames in the mid-level feature, respectively. The d-dimensional feature vector corresponding to each physical coordinate is denoted by $\mathbf{h}(\mathbf{R})[i, j, t] \in \mathbb{R}^d$.

We introduce an auxiliary model $\mathbf{g}$, which is responsible for the explanation. It predicts *conterfactuality*, which measures (1) the positiveness/negativeness of the region $\mathbf{R}$ on the model prediction $p(c|\mathbf{x})$ for a specific pair of $c_{\mathrm{pos}}, c_{\mathrm{neg}}$, and (2) the compatibility of linguistic explanation s to the visual counterpart $\mathbf{R}$. By fixing the parameter of the feature extraction part $\mathbf{h}(\cdot)$, we obtain

$$y_{c_{\mathrm{pos}}, c_{\mathrm{neg}}, s, \mathbf{R}} = \mathbf{g}(\mathbf{h}(\mathbf{x})) \in [0, 1]^{(|\mathcal{C}|-1) \times |\mathcal{S}| \times |\mathcal{R}|} \tag{2}$$

which holds a counterfactuality score corresponding to one combination of $c_{\mathrm{pos}}, c_{\mathrm{neg}}, s, \mathbf{R}$ in each dimension. Positive class c_{pos} is sampled from $p(c|\mathbf{x})$ during training, and $c_{\mathrm{pos}} = \arg\max_c p(c|\mathbf{x})$ is applied in the inference step. Any remaining class $c_{\mathrm{neg}} \in \mathcal{C} \backslash c_{\mathrm{pos}}$ is regarded as negative.

We consider the element of $\mathbf{R}$ in the space of $\mathbf{h}(\mathbf{x})$. In other words, the coordinate (i, j, t) of $\mathbf{R}$ corresponds to that of $\mathbf{h}(\mathbf{x})$. The shape of $\mathbf{R}$ is fixed to $[W/W', H/H']$ for the sake of simplicity. The extension to multiple scales and the aspect ratio will be discussed in subsection 3.6.

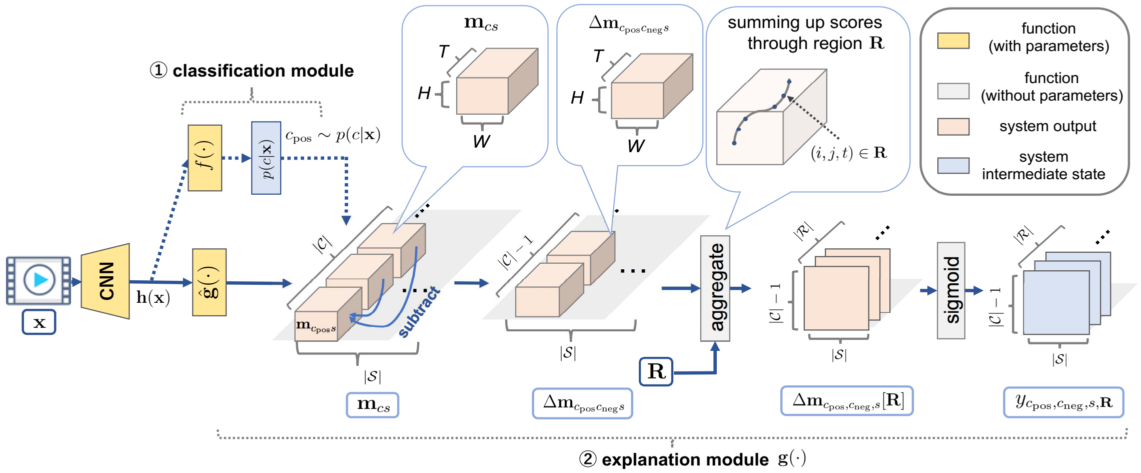
① classification module
$f(\cdot)$
$p(c|\mathbf{x})$
$c_{\text{pos}} \sim p(c|\mathbf{x})$
$\mathbf{x}$
CNN
$\mathbf{h}(\mathbf{x})$
$\hat{\mathrm{g}}(\cdot)$
$\mathbf{m}_{cs}$
T
H
W
$\Delta\mathbf{m}_{c_{\text{pos}}c_{\text{neg}}s}$
summing up scores through region $\mathbf{R}$
$(i,j,t) \in \mathbf{R}$
function (with parameters)
function (without parameters)
system output
system intermediate state
$|\mathcal{C}|$
$\mathbf{m}_{c_{\text{pos}}s}$
subtract
$|\mathcal{S}|$
$|\mathcal{C}|-1$
aggregate
$\mathbf{R}$
$|\mathcal{R}|$
sigmoid
$\Delta\mathbf{m}_{c_{\text{pos}},c_{\text{neg}},s}[\mathbf{R}]$
$y_{c_{\text{pos}},c_{\text{neg}},s,\mathbf{R}}$
② explanation module $\mathrm{g}(\cdot)$

Figure 2: Pipeline of the proposed method. Our model holds two modules, the classification and explanation module. The outline of the framework follows two steps: (a) Train a classification model to be explained, (b) Train an auxiliary explanation model in a post-hoc manner by utilizing output and mid-level features of the target classifier after freezing its weights.

3.3. Predicting counterfactuality

Our explanation model predicts counterfactuality, that is, (1) how much the region $\mathbf{R}$ retains high positiveness/negativeness on the $p(c|\mathbf{x})$ for $c_{\text{pos}}, c_{\text{neg}}$, and (2) how much s is compatible to $\mathbf{R}$. The counterfactuality score is predicted in the following steps.

A target sample $\mathbf{x}$ is inputted to obtain the mid-level representation $\mathbf{h}(\mathbf{x})$ and the conditional probability $p(c|\mathbf{x})$. The explanation model holds classifiers $\hat{\mathbf{g}}_{cs}$ for each pair of (c, s). These classifiers are applied to each element feature of $\mathbf{h}(\mathbf{x})$, and predict simultaneous existence of (c, s) as

$$\mathbf{m}_{cs} = \hat{\mathbf{g}}_{cs}(\mathbf{h}(\mathbf{x})) \in \mathbb{R}^{W \times H \times T}, \tag{3}$$

is obtained for each pair of (c, s). For simplicity, we utilize a linear function that preserves geometrical informatiton, that is, the convolutional layer as classifier $\hat{\mathbf{g}}_{cs}$.

To measure how likely the region element is considered to be c_{pos}, not c_{neg}, with linguistic explanation s, we element-wise subtract the value of (3) for all $c_{\text{neg}} \in \mathcal{C} \backslash \{c_{\text{pos}}\}$ and s as

$$\Delta\mathbf{m}_{c_{\text{pos}},c_{\text{neg}},s} = \mathbf{m}_{c_{\text{pos}}s} - \mathbf{m}_{c_{\text{neg}}s} \tag{4}$$

To obtain the score for the region $\mathbf{R}$, we define the procedure of aggregating scalar values through $\mathbf{R}$ in the 3 dimensional tensor $\Delta\mathbf{m}$ as

$$\Delta\mathbf{m}_{c_{\text{pos}},c_{\text{neg}},s}[\mathbf{R}] = \sum_{(i,j,t)\in\mathbf{R}} \Delta\mathbf{m}_{c_{\text{pos}},c_{\text{neg}},s}[i,j,t]. \tag{5}$$

Please note $\Delta\mathbf{m}_{c_{\text{pos}},c_{\text{neg}},s}[\mathbf{R}] \in \mathbb{R}$. By applying a sigmoid activation function to the output, we obtain counterfactuality $y_{c_{\text{pos}},c_{\text{neg}},s,\mathbf{R}} \in [0,1]^{(|\mathcal{C}|-1)\times|\mathcal{S}|}$ as

$$y_{c_{\text{pos}},c_{\text{neg}},s,\mathbf{R}} = \sigma(\Delta\mathbf{m}_{c_{\text{pos}},c_{\text{neg}},s}[\mathbf{R}]) \tag{6}$$

where $\sigma(a) = \frac{1}{1+\exp(-a)}$.

3.4. Training and inference

We illustrate the loss function optimized in the training step and the procedure in the inference step.

Loss function: The supervised information "A sample is classified to category c_{pos}." can be translated to "A sample is not classified to any category $c_{\text{neg}} \in \mathcal{C} \backslash c_{\text{pos}}$." By utilizing this, the model is trained to enlarge the counterfactuality score corresponding to class pairs $c_{\text{pos}}, c_{\text{neg}}$ and attributes $s \in \mathcal{S}(\mathbf{x})$. The output obtained after sigmoid activation in (6) can be interpreted as a probability, and its negative log likelihood is minimized. Because computing output $y_{c_{\text{pos}},c_{\text{neg}},s,\mathbf{R}}$ for all pairs of $c_{\text{neg}} \in \mathcal{C} \backslash c_{\text{pos}}$, $s \in \mathcal{S}$ and $\mathbf{R} \in \mathcal{R}$ is not feasible, only $\mathbf{R}$ maximizing the loss is utilized for each pair of c_{neg}, s while training. Formally, for a given $\mathbf{x}$ and c_{pos}, the loss

$$\ell(\mathbf{x}, c_{\text{pos}}) = \frac{1}{|\mathcal{S}(\mathbf{x})|} \sum_{s\in\mathcal{S}(\mathbf{x})} \sum_{c_{\text{neg}}\in\mathcal{C}\backslash c_{\text{pos}}} -\log \hat{y}_{c_{\text{pos}},c_{\text{neg}},s}$$
$$\text{where } \hat{y}_{c_{\text{pos}},c_{\text{neg}},s} = \min_{\mathbf{R}} y_{c_{\text{pos}},c_{\text{neg}},s,\mathbf{R}} \tag{7}$$

is calculated. As stated in the next subsection 3.5, $\hat{y}_{c_{\text{pos}},c_{\text{neg}},s}$ can be efficiently computed by dynamic programming under the condition where $\mathcal{R}$, a possible set of $\mathbf{R}$, is limited to the set of all spatio-temporal tubes.

The overall loss function to be optimized is obtained by taking the expectation of (7) over $\mathbf{x}$ and c_{pos} as

$$\mathcal{L} = \mathbb{E}_{p(\mathbf{x})p(c_{\text{pos}}|\mathbf{x})}\left[\ell(\mathbf{x}, c_{\text{pos}})\right]. \tag{8}$$

$p(\mathbf{x})$ indicates the true sample distribution, and $p(c_{\text{pos}}|\mathbf{x})$ is the pre-trained network in (1). Empirically, the expectation over $\mathbf{x}$ is calculated by summing up all training N samples, and that over c_{pos} is achieved by sampling from the conditional distribution $p(c_{\text{pos}}|\mathbf{x})$ given $\mathbf{x}$.

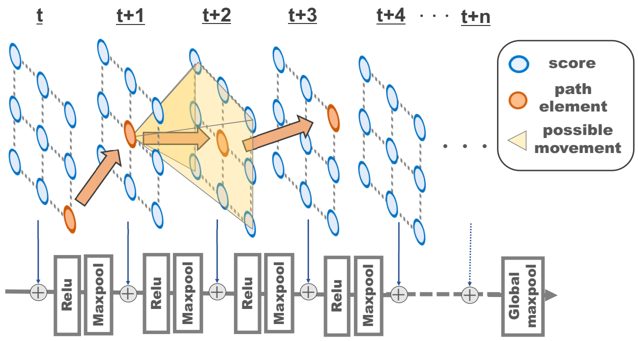

Figure 3: The illustration of maximum subpath pooling. Finding the subpath in the 3d tensor whose summation is maximum can be implemented by sequentially applying the elementwise sum, relu, and 2d max pooling in the time direction, following global 2d max pooling.

Inference: During inference, provided with positive and negative class $c_{\text{pos}}, c_{\text{neg}}$ as well as input $\mathbf{x}$, pairs of the attribute s and the region $\mathbf{R}$ are outputted whose score is the largest. Formally,

$$s^\star, \mathbf{R}^\star = \arg\max_{s,\mathbf{R}} \; y_{c_{\text{pos}},c_{\text{neg}},s,\mathbf{R}} \tag{9}$$

is calculated as the element of explanation. For computing k multiple outputs, we compute $\max_{\mathbf{R}} \; y_{c_{\text{pos}},c_{\text{neg}},s,\mathbf{R}}$ for all s and pick k pairs whose scores are the largest. Minimization for $\mathbf{R}$ is also efficiently calculated as is the case in the training step.

3.5. Maximum subpath pooling

We describe in detail the maximization (minimization) problem for $\mathbf{R}$ appearing in (7) and (9) in this subsection. We limit $\mathcal{R}$, a possible set of $\mathbf{R}$, to the set containing all possible tubes. A tube can be expressed as a path in the 3d tensor, starting at one spatial coordinate (i, j) in time t, and move to the neighbor $\{(i + l, j + m) \mid -k \leq l, m \leq k\}$ in time $t + 1$ where k controls how much movement of the spatial coordinate (i, j) is allowed when the time changes from t to $t + 1$. The path can start and end at any time. The $\mathbf{R}$ consists of coordinates (i, j, t) satisfying the path condition.

With this limitation, the maximization problem with regard to $\mathbf{R}$ can be cast as an extension of finding a subarray whose summation is maximum, and it can be efficiently solved by the algorithm proposed in [28] (shown in the supplementary material), which is an extension of the Kadane's algorithm [5]. Although [28] utilized it only for the inference, we need to train the parameters, especially by back-propagation on GPUs to combine CNNs. To realize this, we construct the algorithm such that it can be implemented as a layer of a neural network with only standard functions pre-implemented in most deep learning libraries [19, 1, 7, 14, 25]. Interestingly, as shown in Fig. 3, the same result can be achieved by sequentially applying relu, 2d maxpooling, and element-wise summation in the time direction followed by global 2d maxpooling. The kernel size of maxpooling is a hyper-parameter corresponding to k mentioned above. We fixed it to 3×3, which means that $k = 1$. The computational cost of this algorithm is $O(WHT)$, which can be solved by a single forward path, since the iteration for W and H can be parallelized on GPU without significant overhead. In the case of minimization of the objective, the same algorithm can be applied just by inverting sign of input and output.

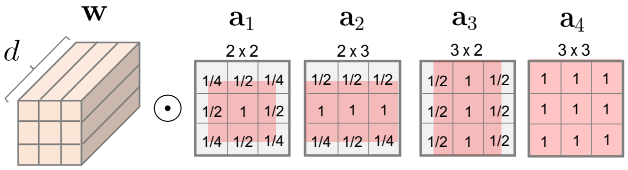

Figure 4: The spatial weights multiplied with the parameters of the convolutional layer. Each element of the weight has a value proportional to the overlap to the outputted shape (in red). These values are normalized such that the summation equals 1.

To acquire the path $\mathbf{R}^\star$ whose summation is maximum, we simply need to calculate the partial derivative of the maximum value with regards to each input. Because

$$\frac{\partial \Delta\mathbf{m}_{c_{\text{pos}},c_{\text{neg}},s}[\mathbf{R}^\star]}{\partial \Delta\mathbf{m}_{c_{\text{pos}},c_{\text{neg}},s}}[i,j,t] = \begin{cases} 1 & ((i,j,t) \in \mathbf{R}^\star) \\ 0 & (\text{otherwise}) \end{cases}$$

we can obtain the path corresponding to the maximum by extracting the element whose derivative is equal to 1. Implementation is likewise easy for the library, which has the function of automatic differentiation.

This procedure can be interpreted as a kind of pooling. To observe this, we denote the aggregated feature after applying sum pooling to mid-level local features throughout the region $\mathbf{R}^\star$ by $\text{pool}(\mathbf{h}(\mathbf{x}), \mathbf{R}^\star)$, and redefine $\Delta\mathbf{w} = \mathbf{w}_{c_{\text{pos}}s} - \mathbf{w}_{c_{\text{neg}}s}$, where $\mathbf{w}_{cs}$ is the parameter of the convolutional layer $\hat{\mathbf{g}}_{cs}$. The summation of the score inside the sigmoid function in (9) can be written as

$$\max_{\mathbf{R}} \Delta\mathbf{m}_{c_{\text{pos}},c_{\text{neg}},s}[\mathbf{R}] = \sum_{(i,j,t)\in\mathbf{R}^\star} \Delta\mathbf{w}^\top \mathbf{h}(\mathbf{x})[i,j,t]$$

$$= \Delta\mathbf{w}^\top \sum_{(i,j,t)\in\mathbf{R}^\star} \mathbf{h}(\mathbf{x})[i,j,t] = \Delta\mathbf{w}^\top \text{pool}(\mathbf{h}(\mathbf{x}), \mathbf{R}^\star)$$

We refer to the sequence of this process as the maximum subpath pooling layer.

3.6. Multiple scales and aspect ratio

So far, we only considered the situation where the scale and aspect ratio of $\mathbf{R}$ is fixed to $[W/W', H/H']$, $1 : 1$. We modify the algorithm to treat different scales and shapes of the region. As described in 3.5, the input of the optimization algorithm $\Delta\mathbf{m}_{c_{\text{pos}},c_{\neg},s}$ (defined in (4)) is a 3d tensor,

dataset	video	class	attribute	bbox
Olympic	783	16	39	949
UCF101-24	3204	24	40	6128

Table 1: statistics of dataset used in the experiment

which corresponds to each physical coordinate in the region. We expand the input from 3d to 5d by expanding $\mathbf{m}_{c,s}$ (defined in (3)), taking scales and shapes into consideration. To obtain the 5d tensor, we prepare multiple parameters $\mathbf{w}$ of the convolutional layer $\hat{\mathbf{g}}_{cs}$ (3) for each scale and shape. (For simplicity, the subscripts c, s of $\mathbf{w}$ will be omitted below). partial To treat different scales, we extract mid-level representations from different layers of the target classifier to construct $\mathbf{h}(\mathbf{x})$. After convolution is applied separately, they are appended for the scale-dimension.

When considering different shapes at each scale, we prepare different parameters by multiplying the significance corresponding to the shape of a region. Formally, $\mathbf{w}_i = \mathbf{w} \odot \mathbf{a}_i$ is computed, where $\mathbf{a}_i$ has the same window size as $\mathbf{w}$ and consists of the importance weight for each position of parameter, which is determined by the overlap ratio between the region and each local element. They are normalized to satisfy $|\mathbf{a}_i| = 1$. Different $\mathbf{w}_i$ are applied separately and output is appended to the shape-dimension of the tensor. Concretely, we compute the scores corresponding to four kinds of shapes $(2 \times 2, 2 \times 3, 3 \times 2, 3 \times 3)$ from 3×3 convolution for each scale as in Fig. 4. The optimization problem can be solved by applying the same algorithm described in subsection 3.5 to the obtained 5d tensor.

3.7. Theoretical background

To demonstrate that linguistic explanation s for region $\mathbf{R}$ is strongly dependent on the output of the target classifier c by minimizing the loss function proposed above, we reveal the relationship between the loss function and conditional mutual information. Conditional mutual information of the distribution parameterized by our model is denoted by $\mathrm{MI}(c, s|\mathbf{x}, \mathbf{R})$ and can be bounded as

$$\begin{aligned}\mathrm{MI}(c, s|\mathbf{x}, \mathbf{R}) &= \mathbb{E}_{p(c,s,\mathbf{x},\mathbf{R})}\left[\log \frac{\hat{\mathrm{p}}(\mathrm{c}, \mathrm{s}|\mathbf{x}, \mathbf{R})}{\hat{\mathrm{p}}(\mathrm{c}|\mathbf{x}, \mathbf{R})\hat{\mathrm{p}}(\mathrm{s}|\mathbf{x}, \mathbf{R})}\right] \\ &\geq \mathbb{E}_{p(c,s,\mathbf{x},\mathbf{R})}\left[\log \hat{p}(c|s, \mathbf{x}, \mathbf{R})\right] \qquad (10)\end{aligned}$$

(10) is derived from $\mathrm{H}(c|\mathbf{x}, \mathbf{R}) \geq 0$ and $\mathrm{KL}[p|q] \geq 0$ for any distribution p, q where $\mathrm{H}(\cdot)$ and $\mathrm{KL}[\cdot|\cdot]$ indicate the entropy and KL divergence, respectively. In our case, we parameterize the joint distribution as

$$\hat{p}(c, s|\mathbf{x}, \mathbf{R}) = \frac{\exp(\mathbf{m}_{\mathrm{cs}}[\mathbf{R}])}{\sum_{c' \in C, s' \in S} \exp(\mathbf{m}_{\mathrm{c's'}}[\mathbf{R}])} \qquad (11)$$

(10) is further bounded as

$$(10) \geq \mathbb{E}_{p(c,s,\mathbf{x})}\left[\min_{\mathbf{R}} \log \frac{\exp(\mathbf{m}_{\mathrm{cs}}[\mathbf{R}])}{\sum_{\mathrm{c's'}} \exp(\mathbf{m}_{\mathrm{c's'}}[\mathbf{R}])}\right] \qquad (12)$$

$$\geq \mathbb{E}_{p(c_{\mathrm{pos}},s,\mathbf{x})}\left[\sum_{c_{\mathrm{neg}} \in \mathcal{C} \setminus c_{\mathrm{pos}}} \min_{\mathbf{R}} \sigma(\Delta \mathbf{m}_{c_{\mathrm{pos}},c_{\mathrm{neg}},s}[\mathbf{R}])\right] \qquad (13)$$

$$= \mathbb{E}_{p(c_{\mathrm{pos}},s,\mathbf{x})}\left[\sum_{c_{\mathrm{neg}} \in \mathcal{C} \setminus c_{\mathrm{pos}}} \hat{y}_{c_{\mathrm{pos}},c_{\mathrm{neg}},s}\right] \qquad (14)$$

$\sigma(\cdot)$ is the sigmoid function and (12) is derived from the fact $\mathbb{E}_a[\mathrm{f}(a)] \geq \min \mathrm{f}(a)$. On the bound (13), we utilize the relationship $(1 + \sum_i a_i) \leq \prod_i (1 + a_i)$ [3].

Finally, by decomposing as $p(s, c|\mathbf{x}) = p(s|c, \mathbf{x})p(c|\mathbf{x})$ and setting $p(c|\mathbf{x})$ as the target classifier and $p(s|c, \mathbf{x}) = [s \in \mathcal{S}(\mathbf{x})]/|\mathcal{S}(\mathbf{x})|$ following the inversion of sign, (8) is obtained. The minimization of the loss function can be justified as the maximization of the lower bound of conditional mutual information. It may be beneficial to investigate the relationship with other methods proposed for the explanation task, such as [6], based on mutual information maximization.

4. Experiment

We describe experiments to demonstrate the effectiveness of proposed method, in particular the explanation module, which is the main proposal for this task. After the details of experimental settings including datasets and metrics used for quantitative evaluation are described in subsection 4.1, the obtained results are discussed in subsection 4.2.

4.1. Setting

Given an input video $\mathbf{x}$, and a pair of positive/negative class $c_{\mathrm{pos}}, c_{\mathrm{neg}}$, the explanation module outputs several pairs of attribute/region $\{(s_i, \mathbf{R}_i)\}_{i=1}^{k}$. We separately evaluate each pair of output.

Dataset: Two existing video datasets for action recognition: Olympic Sports [17] and UCF101-24 categories [15] were used in the experiments. The Olympic Sports dataset consists of 16 categories for sports action. The UCF101-24 categories is a subset of the UCF101 dataset [24] extracting 24 out of 101 general action classes. We utilized the original train/test split provided by the datasets. We additionally assigned these datasets with Amazon Mechanical Tutk (AMT) to make the evaluation possible as follows: (a) Assign a set of attributes to all videos in the dataset, (b) Assign a bounding box of assigned attributes for samples in the test split. Statistics of the dataset are shown in Table. 1, and a few examples of the annotations are provided in the supplementary material.

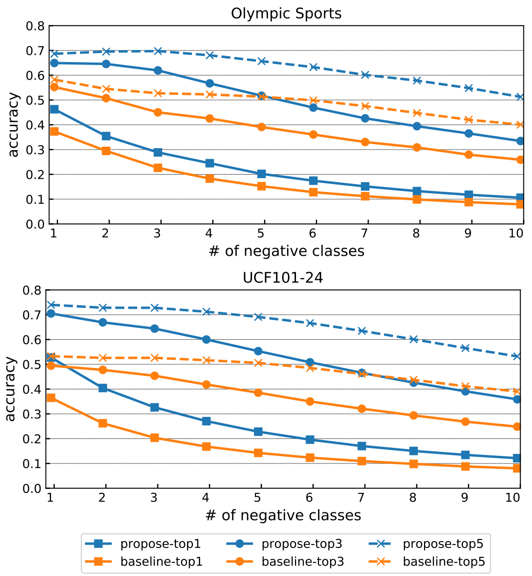

Figure 5: The negative class accuracy on the Olympic Sports dataset (above) and the UCF101-24 dataset (below). The y-axis depicts the mean accuracy and the x-axis denotes the number of negative classes used for averaging, whose prediction value is maximum.

Metric: As stated earlier, the method for this task is expected to satisfy the following two requirements:

(1) Visual explanation is the region which retains high positiveness/negativeness on the model prediction for specific positive/negative classes,

(2) Linguistic explanation is compatible to the visual counterpart,

and methods are evaluated based on them. To make the quantitative evaluation possible, we propose two metrics, both of which are based on the accuracy.

As for (1), we need to evaluate whether the obtained region is truly an explanation of the reason for "the target classifier predicts a sample to not c_{neg} but c_{pos}". More specifically, we would like to confirm whether the region explains the specific negative class c_{neg}, not other negatives $\hat{c}_{\text{neg}} \in \mathcal{C} \backslash \{c_{\text{pos}}, c_{\text{neg}}\}$. To evaluate this quantitatively, we investigate how the output of the target classifier changes corresponding to c_{neg} when region $\mathbf{R}$ is removed from the input. A mask $\mathbf{z} = \{0, 1\}^{W' \times H' \times T' \times 3}$ is prepared, which takes the value of 0 if the corresponding pixel's location is contained to $\mathbf{R}$ restored on the input space, otherwise it takes the value of 1. Applying again the masked sample to the target classifier, the difference from the original output $f(\mathbf{h}(\mathbf{x} \odot \mathbf{z})) - f(\mathbf{h}(\mathbf{x}))$ is calculated for all negative classes where $\odot$ denotes the Hadamard product. We calculate the accuracy, i.e., we pick the largest values out of the obtained difference scores, and examine if c_{neg} exists within this set. We refer to this metric as negative class accuracy.

As for (2), we assess how the region $\mathbf{R}$ makes the concept s identifiable by humans for each output pair $(s, \mathbf{R})$.

method	Olympic	UCF101-24
baseline	0.76	0.68
propose	0.89	0.88

Table 2: The ratio of the probability $p(c_{\text{pos}}|\mathbf{x})$ for the positive class c_{pos} decreasing after the region is masked out.

	Olympic Sports			UCF101-24		
method	top1	top3	top5	top1	top3	top5
baeline	0.02	0.07	0.12	0.02	0.07	0.13
propose	0.13	0.38	0.59	0.14	0.41	0.65

Table 3: The concept accuracy on the Olympic Sports dataset and the UCF101-24 dataset.

To quantify this, we exploit the bounding boxes assigned for each attribute in the test set, and compute the accuracy as follows. IoU (intersection over union) is calculated between given $\mathbf{R}$ and all bounding boxes $\mathbf{R}'$, which corresponds to attribute s'. We measure the accuracy by selecting the attribute s' with the largest IoU score, and checking its consistency with s, which is the counterpart of $\mathbf{R}$. This metric is referred to as concept accuracy in the following parts.

Detailed settings In the classification module, the output of convolutional layers was used as $\mathbf{h}(\cdot)$. Fc layers following convolutional layers were considered as $f(\cdot)$.

Specifically, we dealt with C3D-resnet [27, 11] as the target classifier in the experiments, which is based on the spatio-temporal convolution [26] with residual architecture [12]. Our network for classification consists of nine convolutional layers and one fully connected layer accepting a $112 \times 112 \times 16$ size input. Relu activation was applied to all layers, except the final fc layer. We selected the outputs of the last and the 2nd to last convolutional layers to construct $\mathbf{h}(\cdot)$. Moreover, we replaced 3d max pooling to 2d max pooling to guarantee $T = T'$ for all activations. The target classifier was trained with SGD, where learning rate, momentum, weight decay, and batch size, were set to 0.1, 0.9, 1e-3, and 64 respectively. To train the model in Olympic Sports, we pre-trained it with the UCF101-24.

For the training of the explanation module, all the settings (e.g., learning rate) were set to the same as in the training of classification module, except that the batch size was 30. We decomposed the weight of the convolutional layer $\mathbf{w}_{cs}$ corresponding to the pair of (c, s) to $\mathbf{w}_{cs} = \mathbf{w}_c \odot \mathbf{w}_s + \mathbf{w}_c + \mathbf{w}_s$ to reduce the number of parameters where $\mathbf{w}_s$ and $\mathbf{w}_c$ are the parameter shared by the same attribute and class respectively.

4.2. Identifiability of negative class

To assess whether the obtained region $\mathbf{R}$ is truly an explanation of the reason for a specific negative class, we compared the negative class accuracy with a baseline. Because there is no previous work for this task, we employed a simple baseline. For the UCF101-24 dataset, we exploited the bounding box for the action detection provided in [23],

	Olympic Sports			UCF101-24		
method	fc1	fc2	fc3	fc1	fc2	fc3
baeline	0.45	0.44	0.46	0.45	0.43	0.44
propose	0.62	0.64	0.59	0.64	0.62	0.61

Table 4: The top3 negative class accuracy on the Olympic Sports dataset and the UCF101-24 dataset averaged over 3 negative classes whose prediction probability is the largest, by changing the number of fully-connected layers.

which provides an upper limit of the performance for the action detection task. As bounding boxes are not provided for the Olympic Sports dataset, we cropped the center (32×32) from all frames. This forms a simple but strong baseline because the instance containing category information usually appears in the center of the frame in this dataset.

The results of the negative class accuracy for the Olympic Sports and the UCF101-24 datasets are shown in Fig. 5. The accuracy averaged over negative classes having largest $p(c|\mathbf{x})$ is calculated and the x-axis of the figures depicts the number of used negative classes. Our method performed consistently better than the baseline on both datasets, demonstrating the generalization of identifiability of negativeness in unseen samples. The gap between the accuracy of our method and that of baseline is decreased when negative classes having small $p(c|\mathbf{x})$ are included. We conjecture the reason is that such a *easy negative* class, which is highly dissimilar to the positive class, does not have common patterns to identify them. For example, for positive class 'pole vault', the negative class 'high jump' is considered to be more similar than 'bowling' for the classifier, and detecting the negativeness for such a *hard negative* 'pole vault' is relatively easy (e.g. the region of 'pole'), although detecting it is difficult for the *easy negative* class. We believe the low-accuracy for such an *easy negative* class does not have significant impact because in real applications, we may be interested in the reason for the negativeness of the *hard negative* class, which is difficult to discriminate. In addition, we also report the ratio of the probability $p(c_{\text{pos}}|\mathbf{x})$ for the positive class c_{pos} decreasing after the region is masked out in Table. 2. From these results, we claim that our method can find both better of negativeness/positiveness on the negative/positive classes.

4.3. Identifiability of concept

To evaluate whether the obtained region makes the concept (linguistic explanation) identifiable by human, we measured concept accuracy described above for the case where category prediction is correct. The same baseline was applied for region selection as in the previous subsection 4.2, and the attribute is randomly selected. Results are shown in Table 3. In both datasets, our method is consistently better than the baseline. Finding the region by which a specific concept can be identified is the significantly challenging task, where methods need to identify small objects or atomic-actions [10]. Although it is conceivable that there is still room for improvement of this metric, we believe that our simple method can serve as a base for future work regarding this novel task.

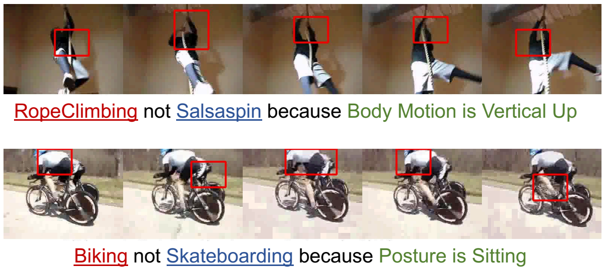
RopeClimbing not Salsaspin because Body Motion is Vertical Up
Biking not Skateboarding because Posture is Sitting

Figure 6: Example output from our system for the samples of the UCF101-24 dataset.

4.4. Influence of the complexity of classifier

To investigate the influence of the complexity of the classifier module on the generalization ability of the explanation module, we measured the negative class accuracy by changing the number of fc-layers of $f(c|\mathbf{x})$ in $1 \sim 3$. The other settings remain the same as those in subsection 4.2. The top3 accuracy averaged on 3 negative classes is shown in Table. 4 (Other results are shown in supplementary material). In both datasets, the gap between the baseline and the proposed method is consistent regardless of the number of fc-layers, demonstrating the robustness of the proposed method to the complexity of classifiers to be explained.

4.5. Output examples

We show a few output examples in Fig. 6 for the sample videos of the UCF101-24 dataset. As observed in the figures, our model is considered to appropriately localize the area compatible with the linguistic explanation. Other examples are shown in the supplementary material.

5. Conclusion

In this work, we particularly focused on building a model that not only categorizes a sample, but also generates an explanation with linguistic and region. To this end, we proposed a novel algorithm to predict *counterfactuality*, while identifying the important region for the linguistic explanation. Furthermore, we demonstrated the effectiveness of the approach on two existing datasets extended in this work.

6. Acknowledgement

This work was partially supported by JST CREST Grant Number JPMJCR1403, Japan, and partially supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) as "Seminal Issue on Post-K Computer." Authors would like to thank Kosuke Arase, Mikihiro Tanaka, Yusuke Mukuta for helpful discussions.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 2, 5

[2] L. Anne Hendricks, R. Hu, T. Darrell, and Z. Akata. Grounding visual explanations. In *ECCV*, 2018. 1, 2, 3

[3] M. T. R. AUEB. One-vs-each approximation to softmax for scalable estimation of probabilities. In *NIPS*, 2016. 6

[4] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. 2

[5] J. Bentley. Programming pearls: algorithm design techniques. *Communications of the ACM*, 27(9):865–873, 1984. 5

[6] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *ICML*, 2018. 2, 6

[7] F. Chollet. keras. `https://github.com/fchollet/keras`, 2015. 2, 5

[8] P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers. In *NIPS*, 2017. 2

[9] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, 2017. 2

[10] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 8

[11] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018. 7

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[13] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations. In *ECCV*, 2016. 2, 3

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 2, 5

[15] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/`, 2014. 6

[16] A. Kanehira and T. Harada. Learning to explain with complemental examples. In *CVPR*, 2019. 2

[17] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010. 6

[18] D. H. Park, L. A. Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, and M. Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *CVPR*, 2018. 2

[19] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. 2, 5

[20] A. S. Ross, M. C. Hughes, and F. Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. 2017. 2

[21] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 2

[22] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2

[23] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin. Online real time multiple spatiotemporal action localisation and prediction. 2017. 7

[24] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6

[25] S. Tokui, K. Oono, S. Hido, and J. Clayton. Chainer: a next-generation open source framework for deep learning. In *Workshop on Machine Learning Systems (LearningSys) in NIPS*, 2015. 2, 5

[26] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 7

[27] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri. Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*, 2017. 7

[28] D. Tran, J. Yuan, and D. Forsyth. Video event detection: From subvolume localization to spatio-temporal path search. 2014. 5

[29] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. 2017. 1

[30] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *ECCV*, 2016. 2

[31] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 2

[32] B. Zhou, Y. Sun, D. Bau, and A. Torralba. Interpretable basis decomposition for visual explanation. In *ECCV*, 2018. 2