IET Journals
The Institution of Engineering and Technology

Hand gesture recognition with better interpretability

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Yonghao Zhuang[1] *Haichen Dong*[1] *Jiaxin Lu*[1]*

[1] *Shanghai Jiao Tong University, Shanghai, 200240, People's Republic of China*
* *E-mail: lujiaxin@sjtu.edu.cn*

Abstract: Visual-based hand gesture recognition is of high importance in computer vision. Many previous works have presented a satisfying performance for hand gesture classification in videos. However, such methods like Conv3D may be difficult to visualize because of the mixture in the time dimension and the difficulty for upsampling caused by 3D pooling layers. To tackle these problems, we propose a network based on 2-dimensional structures. This network outperforms the commonly used C3D model in the recognition task. Moreover, we develop a method called Seq-Grad-CAM++ that obtains a significantly better performance in the objective evaluation and human trust and thus demonstrates that it enjoys better interpretability which enables its further reality use.

1 Introduction

Hand, as a powerful part of the human body, gives us access to interact with real-world objects, and also, as an auxiliary of our communication. Vision-based hand gesture recognition is still of high importance in computer vision, especially for applications such as hand rehabilitation [1], sign language [2], object manipulation in virtual/augmented reality [3], and human-computer interaction (HCI). Learning how we recognize a hand gesture is also instructive to constructing a virtual hand model, human pose recognition, well as for robots to learn to grasp and imitate the human pose.

With the evolution of sensors, cameras, and open-source software, many approaches have been raised to have access to hand gesture data in different settings, including first/second/third-person view of RGB images and videos. Among these settings, first-person action recognition [4] has shown its potential, especially in the case of wearable smart device interaction where gestures are captured by egocentric cameras mounted on some users. Preferably, it enables a human-centering perspective of the visual world, while embodies some unique characteristics: 1) Hands in close range: given that the camera is attached in a very short distance from the hand, one may find the hand is partly or totally out of the field-of-view of the frame. 2) Movement of both hands and camera: since the camera is mounted on a user, camera motion can be significant and thus may result in the drop of quality of the images gathered, which is more applied to the real-world situation.

Many works [5–8] have developed several kinds of algorithms based on computer vision methods using knowledge such as skin color, motion, depth, and appearance. Currently, many works [9–12] use 3D hand model or skeleton-based model as shown in Fig. 1 to detect and recognize hand gesture. However, these methods specify a large number of parameters and require a vast amount of images to formulate the characters of handshape in avoidance of its setback in multi-view. Meanwhile, these methods would be extensively affected by the data quality and function badly in the unclear view or the corruption of hand in one image. Moreover, the time consumption to build and match models as well as the demand for prior knowledge in the human body makes it hard for universal use.

Therefore, we apply Deep Learning, the trending solutions in this field. However, compared to the model-based methods stated above, CNN's lack of decomposability into individually intuitive components leads to hardship in interpretation. Consequently, even the network achieves superior performance, it cannot establish enough trust for users. Nor can it help to find insights of hand gesture representation for future research.

To tackle these distrust and find how different parts of the hand work in gesture cognition rather than using the existing hand models, we modify a method from the field of Deep Learning Interpretability: Grad-CAM++ [13]. The field of interpretability focuses on giving human interpretable results, mostly visible pictures, using the output or intermediate result in the interfacing process.

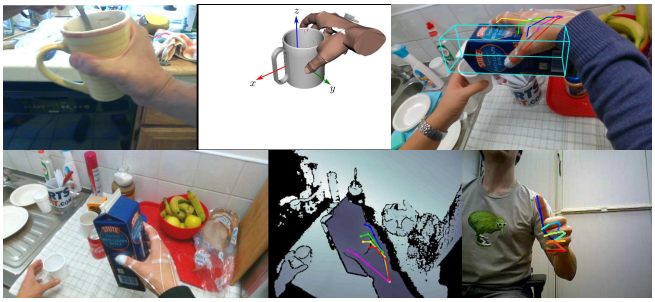

Fig. 1: Methods [9–12] using skeleton and 3D hand models

To extract enough information from RGB data in a period of time, we initially select C3D [14] model which is mainly constructed by 3D convolution layers and 3D pooling layers. However, We notice that although the classification accuracy of C3D is acceptable, the model is not eligible to accept Grad-CAM++ method based on reasons elaborated in 3.3.

We discover that the problem is mainly introduced by the property of the 3D input and output of convolution layers. Since the input has only few photos, the location of object is likely to shift between snapshots. Hence, the time dimension is purposed to have difference from the other two space dimensions, since the Grad-CAM++ algorithm tends to focus on a continuous area on the whole space, no matter the dimension number is 2 or 3. Consequently, the activated area in different photos drags each other, making the performance poor.

So we turn back to 2D convolution neural network and Grad-CAM++ for 2D, since the object is in a continuous area in a photo. We propose Seq-Grad-CAM and Seq-Grad-CAM++ which is a modification of the 2D convolution neural network and the Grad-CAM++ to make them fit the 3D input. Details of the modification are elaborated in 3.3.

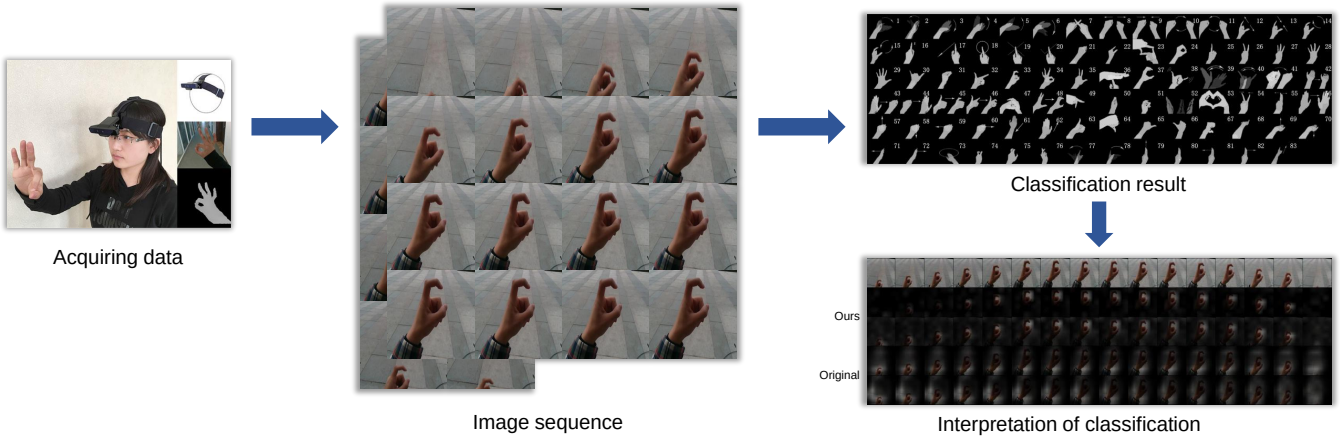

Fig. 2: Task Settings: The data acquiring process is shown in [4]. With given image sequence, we will first give classification result of the gesture in the sequence and then generate visualization of the important part in our architecture's decision-making.

In the experiments, we achieve high accuracy(about 90%) in hand gesture recognition. We also conduct human studies on the visualization of CNN results and it shows that our Seq-Grad-CAM++ outperforms the state-of-art methods in interpretability.

In a nutshell, the contribution of this paper can be summarized as follows:

- We design a novel architecture based on CNN for first-person hand gesture recognition. This architecture can handle complicated situation that the hand gesture is given in image sequence and can be corrupted, achieving a promising accuracy for real-world application.
- We introduce a class-discriminative localization technique, Seq-Grad-CAM++, that provides visualization of explanation for the result of our hand gesture recognition network. It shows us how each part of our hand is considered in the process of gesture recognition.
- While several existing methods visualize CNN decisions, namely, Deconvolution, CAM, Grad-CAM, and Grad-CAM++, their assessment on a sequence of images is of low quality, especially for the ones that the hand is moving. Our method, Seq-Grad-CAM++, shows superior performance under such circumstance, instilling greater trust in human studies.

In the next section, we review related work on hand gesture recognition and some models on interpretation and visualization of the CNN networks. The following sections explain the full setting of our tasks, our architectures for gesture recognition, our improvement on the interpretability of CNN networks, and the detailed implementation of these methods we use. Finally, we analyze the accuracy of our architecture on hand gesture recognition and evaluate the quality of our explanation map.

2 Related Work

Hand gesture recognition: In the early stage of gesture recognition, wearable glove-based sensors that should be attached directly to the hand are widely employed. With these sensors, information including coordinates of palms and fingers, overall movement, and finger bending would all be detected [15–17], and gestures could be recognized accordingly [17, 18]. Though these methods show promising results, they hold various limitations in real-world application.

With the rapid development of computer vision, camera vision-based approaches are raised for this problem and soon wins attention given it is contactless, convenient, and general for use [19].

Many approaches break down the whole process into two stages. The first stage is always hand detection or segmentation. Methods including skin color-based detection and appearance-based recognition are conduct for this preprocess. [6, 20] discussed skin color detection based on RGB channels in detail, while [21, 22] shows skin tone detection based on Y-Cb-Cr is superior given its ability to avoid the interference of lighting. From a different perspective, feature extracting is used to model hand appearance. Haar-like features which describe hand posture pattern and context-free grammar which can analyze postures' syntactic structure are used in [8]. More similar approaches including extracting features of fingers, palms, edges can be found in [7, 23, 24].

With features or hand segments above, methods based on the skeleton, 3D model, depth, or a mixture of models listed above, are proposed for gesture recognition. Specifically, [25] developed a method to extract hand skeleton joints' positions. Then to specify the skeleton points, Laplacian-based contraction was used in [26]. Depth information is introduced given the improvement in technologies considering cameras. [27] combines depth and color information to improve depth threshold segmentation for gesture recognition. A combination of depth and skeletal dataset used on dynamic hand gesture recognition can be found in [28]. Moreover, with depth parameter added, [29] proposed an architecture on 3D hand pose recognition with the use of the self-supervision neural network. A study by [30] developed a method based on point-to-point regression to estimate 3D gesture in single depth images. Even with 2D RGB input images, [31] managed to understand the interaction between objects and 3D hands.

Meanwhile, numerous deep-learning-based methods have been proposed for gesture recognition. Some [32–34] applied deep convolutional neural network (CNN) including Alex Net and VGG Net after utilizing skin color techniques listed above, while some [35] developed adapted deep neural network (ADCNN) together with feature extraction. Researchers also proposed a double-channel convolutional neural network (DC-CNN) in [36] where preprocessed hand edge information will be fed into the network. [37] has shown an architecture based on SPD manifold learning applied in skeleton-based gesture recognition.

However, these methods all used professional and prior knowledge or some localization techniques which could be highly affected or even unfunctional in harsher conditions with complex background, extreme lighting, or blurred view. Therefore, we proposed a learning-based model without utilizing any localization, skeleton, or 3D models. Some related work refers to [38, 39] which also consider the whole image without any region proposal algorithm, though they still have room for further improvement in accuracy, especially when input is a serial of images.

Visualizing CNNs: Many previous works [40, 41] have shown several approaches with pixel-space gradient visualization. Such methods visualize CNN prediction by highlighting pixels considered crucial for prediction. Specifically, [41] proposed a deconvolution view for understanding deep CNNs, especially to study what higher layers of a given CNN have learned. This method could

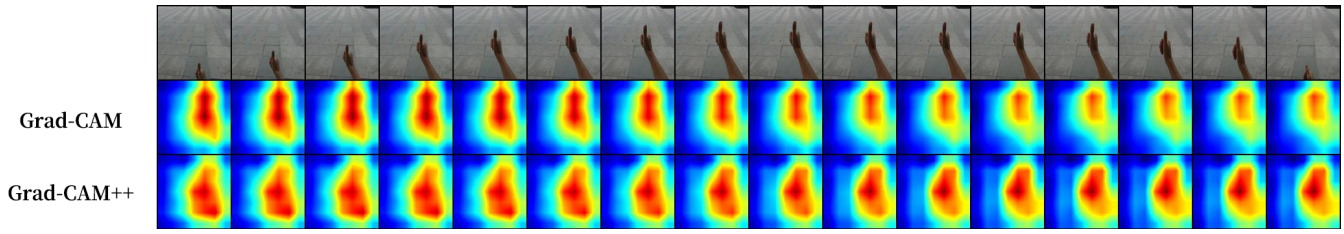

Fig. 3: Heatmap of Grad-CAM and Grad-CAM++ on C3D

highlight parts of the image which strongly activate neurons in the higher layer. Guided Backpropagation [40] is raised based on this work that allows an extended understanding of each neuron's impact with respect to the input image in a given network. These methods of interpretation are compared in [42]. Though they are high-resolution and produce fine-grained visualizations, they are not class-discriminative [43]. That is, their visualizations are almost identical for different classes.

To tackle these drawbacks, [44] proposed a visualization method that synthesizes images that cause high activation of a specific unit in a network so that we can have visualization of the functionality of these units. [45] developed a more guided approach to maximally activate a neural unit by synthesizing input images. They performed gradient ascent in pixel space to generate class-specific saliency map which functions to understand how one CNN models a class. Other methods [42, 46] that invert a latent representation are also used to generate class-discriminative interpretation for deep networks. However, these methods still suffer not specific to different input images [43] and thus provide undesirable visualizations.

Another perspective is to use the local interpretation methods [47–50] which look into the predictions of local perturbations aiming to visualize the critical features for an input instance. [51] proposed LIME (Local Interpretable Model-Agnostic Explanations) which learned a local approximation to the complex decision surface around the input instance, showing the relevance of a feature in a particular prediction.

Most relevant approaches to ours are Class Activation Mapping (CAM) [52] and two methods developed upon it, Grad-CAM [43] and Grad-CAM++ [13]. CAM [52] replaces fully-connected layers in CNNs with convolutional layers and global average pooling to identify discriminative regions of each image. However, this modification requires retraining for multiple classifiers and has limited interpretability prowess to CNNs. [43] proposed a generalization of CAM [52], Grad-CAM [43], by flowing class-specific gradients from upper layers into the final convolutional layer of a CNN, thus addressed the issues above. [13] built a more generalized version called Grad-CAM++ to improve the interpretability of the entire object (instead of bits and parts of it) and multiple objects in one image. These previous works don't apply well when given an image sequence and therefore we give our modification to make them more suitable for our task.

3 Method

In this section we will first give the settings of our task. We break down our task into two stages and give our analysis on each stage, well as their methods. Finally, we will show the implementation details of our methods.

3.1 Task Settings

3.1.1 Definitions and Notations: We first give some basic notations following from [9].

Image and image sequence: Let $\mathbf{X} \in \mathbb{R}^{H\times W\times 3}$ be an RGB image of a hand gesture. Let $\mathcal{S} = \{\mathbf{X}_i\}$ be an image sequence of a hand gesture which collected from a 30fps video. Note that images in an image sequence are in order. Let $|\mathcal{S}|$ denote the length of the image sequence.

Hand Gesture: Let $\mathbf{HG} \in \mathbb{R}^{83}$ describe the hand gesture in the camera frame which is one-hot encoded and each dimension represents a gesture listed in [4].

Explanation map: Let $\mathbf{E} \in \mathbb{R}^{1\times H\times W}$ be the explanation map of an image $\mathbf{X}$. It is normalized into $(0, 1)$ and should highlight the regions that are most relevant for decision-making. It is visualized as a heat map, and we will use such representation in the following experiments. Let $\mathcal{M} = \{\mathbf{E}_i\}$ be a sequence of explanation maps generated from an image sequence $\mathcal{S} = \{\mathbf{X}_i\}$.

3.1.2 Task Overview: Formally, our task becomes: with a sequence of images representing a hand gesture, one aims to first recognize the hand gesture, then gives visual explanation to demonstrate which parts of the input were looked at by the recognition algorithm for assigning a label.

More specifically, the input of our task is an image sequence $\mathcal{S}$ with $|\mathcal{S}| = 16$, each of which represents a frame of a 30fps video and two adjacent of which have the same time gap. In the first stage, we should recognize $\mathbf{HG}$, the gesture that the image sequence represents. Then, in the second stage, we generate a sequence of explanation maps $\mathcal{M}$ which explains key points in each image for the recognition task.

The whole process is shown in Fig. 2.

3.2 Hand Gesture Recognition

Consider images into the deep CNN for hand gesture recognition. Since we are given an image sequence, it is important for us to properly use the information in different images jointly. Therefore, we use one network to extract the hidden feature in every image of given sequence. Note that in this step we consider each image in one sequence separately, so that the network could have a complete understanding of the input images. Then we put the features into a classification network which could uniformly consider the contribution of images to the recognition.

This whole process lowers the number of parameters in the training while remains the consideration of each image and the image sequence as a whole. For the betterment of recognition quality, we choose ResNet50 [53] as our main CNN network in this stage. Detailed implementation is in 3.4.

3.3 Explanation of Recognition

Now we have a network to classify hand gestures, we still need to explain how our network does the recognition. We propose modified 3D version of Grad-CAM [43] and Grad-CAM++ [13] called Seq-Grad-CAM and Seq-Grad-CAM++ so it could apply to the given task.

Let Y^c be the score of a particular class c, in original Grad-CAM++ there is:

$$Y^c = \sum_k \{\sum_a \sum_b \alpha_{ab}^{kc} \cdot relu(\frac{\partial Y^c}{\partial A_{ab}^k})\}[\sum_i \sum_j A_{ij}^k] \quad (1)$$

where (i, j) and (a, b) are iterators over the same activation map A^k.

In the 3D case, the activation maps with feature k have three dimensions, representing the activation map at different time. If we apply the commonly used architecture C3D [14] in the classification process and then use Grad-CAM++ to explain the result, the output feature map of the last 3D convolution layer, named A^k, has three dimensions. Hence, there is:

$$Y^c = \sum_k \{\sum_a \sum_b \sum_t \alpha_{abt}^{kc} \cdot relu(\frac{\partial Y^c}{\partial A_{abt}^k})\}[\sum_i \sum_j \sum_l A_{ijl}^k] \tag{2}$$

This has no difference with the original equation but one more iteration over the third dimension.

However, the method mentioned above has two shortcomings: the first is the low performance on the salience map, which is mainly caused by the weighted average on the dimension of time. The result is that the calculated attention of the photo at a time contains the attention at another time. Moreover, to visualize the salience map, we use upsampling, but due to the architecture of C3D, with a few 3D pooling layers, the size of the third dimension of activation maps and the salience map is only 1 or 2. Thus, upsampling to 16 photos is a hard task, and the performance is low.

For instance, in Fig. 3, we can notice that in all 16 frames, the heatmap of the last 6 frames are almost the same, while the hand shifts from the top to the bottom of the frame.

To solve the two problems above, we modify the original 3D Grad-CAM++ method mentioned above and propose our version of Seq-Grad-CAM++. Our network architecture, as mentioned in section3.2, is based on a 2D convolution network. The features in different time are collected together after they are isolated processed by the 2D network and then used as the input of a classification network.

In our model, the activation map of feature k in time t is denoted by 2D A^{kt} rather than 3D A^k in original C3D network, and the equation turns out to:

$$Y^c = \sum_k \sum_t \{\sum_a \sum_b \alpha_{ab}^{ktc} \cdot relu(\frac{\partial Y^c}{\partial A_{ab}^{kt}})\}[\sum_i \sum_j A_{ij}^{kt}] \tag{3}$$

where (i, j) and (a, b) hold the same meaning.

Taking partial derivative w.r.t. A_{ij}^{kt} and a further partial derivative w.r.t. A_{ij}^{kt} on both sides, there is:

$$\frac{\partial^2 Y^c}{(\partial A_{ij}^{kt})} = 2\alpha_{ij}^{ktc} \frac{\partial^2 Y^c}{(\partial A_{ij}^{kt})^2} + \sum_a \sum_b A_{ab}^{kt} \{\alpha_{ij}^{ktc} \frac{\partial^3 Y^c}{(\partial A_{ij}^{kt})^3}\} \tag{4}$$

After rearranging terms there is:

$$\alpha_{ij}^{ktc} = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^{kt})^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^{kt})^2} + \sum_a \sum_b A_{ab}^{kt} \{\alpha_{ij}^{ktc} \frac{\partial^3 Y^c}{(\partial A_{ij}^{kt})^3}\}} \tag{5}$$

Substituting the Eq. 5 into Eq. 3, we get the weight that:

$$w_{kt}^c = \sum_i \sum_j [\frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^{kt})^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^{kt})^2} + \sum_a \sum_b A_{ab}^{kt} \{\alpha_{ij}^{ktc} \frac{\partial^3 Y^c}{(\partial A_{ij}^{kt})^3}\}}] \cdot relu(\frac{\partial Y^c}{\partial A_{ij}^{kt}}) \tag{6}$$

Evidently, let $k' = k \cdot T + t$ where $t = 1, 2 \dots T$, our Seq-Grad-CAM++ reduces to the formulation for Grad-CAM++. Thus shows the theoretical guarantee of our methods.

With the weight of different features and classes, the class-specific salience map L^c is then calculated as:

$$L_{ij,t}^c = \sum_k w_{kt}^c A_{ij}^{kt} \tag{7}$$

Here $L_{ij,t}^c$ directly correlates with the importance of a particular coordinate (i, j) on the feature map at time t for a particular class c, as well as functions as a visual explanation of the prediction of the network. To regress to the original input frames, the salience maps are upsampled and fused by point-wise multiplication.[40]

It should be mentioned that with this method, the weighted average operation is only among the two dimensions of pixels without the dimension of time. This feature preserves the independence between photos in different time. Hence, the first problem of Grad-CAM++ on the C3D architecture is solved by isolating photos actively. In addition, pooling operation in the 2D CNN network has nothing to do with the scale of time, so we do not need to consider the rough upsample problem on this dimension. Moreover, since the kernel of 2D convolution layer is much smaller than those of 3D convolution layer, with the same number of parameters as well as the same size of the model, our method can have more layers, which can depict more details of the data space and be more accurate.

3.4 Implementation Details

3.4.1 Data Pre-processing: Rather than directly using the given image sequence, we first conduct sampling on the given data. From the given image sequence $\mathcal{S}$, we construct our sampled image sequence $\hat{\mathcal{S}}$ with $\left|\hat{\mathcal{S}}\right| = 16$. Images in $\hat{\mathcal{S}}$ are randomly chosen from the given 32 images while remain their time order.

Such operation is designed to enhance the stability of our model. By randomly sampling half of the images, we could have a larger various of data for each scenery and gesture, which also helps to reduce the overfitting problem.

In order to apply our architecture, we also compress the given images into size 224×224. We apply direct compress instead of randomly selecting an area in the image which allows us to keep all the information in one image and ensures that the moving of the hand won't be eliminated.

3.4.2 Architecture of Hand Gesture Recognition (modified from ResNet50 [53]): The modified ResNet architecture is shown in Fig. 4.

Convolutional Layer: Our architecture uses 53 convolutional layers of different number and sizes of kernels. These convolutional layers follows that the layers have the same number of filters for the same output feature map size, while the number of filters should be doubled in order to capture more features if the output feature map size is halved. We also perform downsampling directly by convolutional layers that have stride 2.

Pooling Layer: To handle the overfitting problem, we use a global average pooling layer at the end of the convolutional layers.

Feature Layer: The feature of each frame is extracted by a 83-way fully-connected layer. Finally, the 83-dimensional feature of each frame is concatenated in time order, and is fed into another 83-way fully-connected layer with softmax. We also implement an 83-dimensional Long Short-Term Memory structure with output size 83 to gather together the information from every frame.

3.4.3 Seq-Grad-CAM and Seq-Grad-CAM++: Due to the same concern on the requirement about smoothness of Y^c as Grad-CAM++ has, we use the exponential function, that is to let $Y^c = \exp(X^c)$. So the activation is calculated by:

$$\alpha_{ij}^{ktc} = \frac{(\frac{\partial X^c}{\partial A_{ij}^{kt}})^2}{2(\frac{\partial X^c}{\partial A_{ij}^{kt}})^2 + \sum_a \sum_b A_{ab}^{kt} \{\alpha_{ij}^{ktc} (\frac{\partial X^c}{\partial A_{ij}^{kt}})^3\}}$$

And the saliency map is then calculated as a linear combination with a $relu$ layer:

$$L_{ij}^{tc} = relu\{\sum_k [\sum_a \sum_b \alpha_{ab}^{ktc} \cdot relu(\frac{\partial X^c}{\partial A_{ij}^{kt}} X^c)] A_{ij}^{kt}\}$$

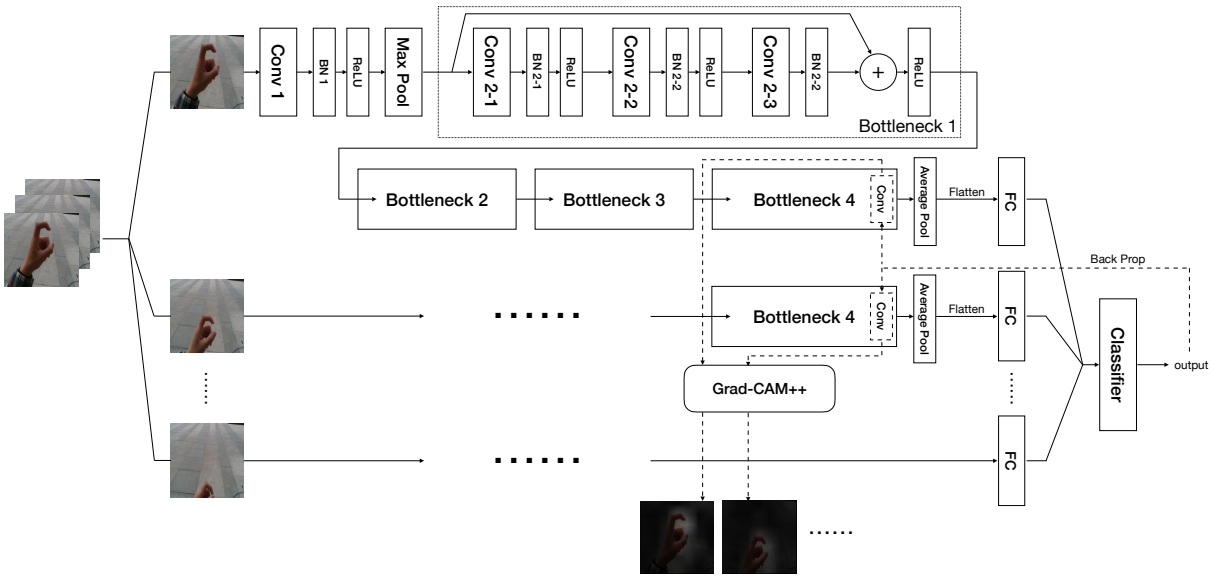

Fig. 4: Structure of the model: 2D Resnet for Seq-Grad-CAM++

It should be mentioned that, the idea can be applied in Grad-CAM as well, since channels in it is also different from

4 Experiments

In this section, we explain our experimental settings and the dataset used for gesture classification, then present the result of visualization generated by the proposed method as well as the accuracy of the classification, comparing with the state-of-art work.

4.1 Implementation settings

The backbone we adopt in our work is as follows: firstly, the 13 convolution layers from ResNet50 is adopted to extract the features per frame. After that, the 83-dimensional frame-specific feature obtained from fc layer of ResNet50 is then fed into an uni-directional Long Short-Term Memory Structure with 83 hidden-layers.

We use PyTorch framework to implement our method. Adam optimization algorithm is used to train our network, with learning rate of 10^{-4} and weight decay of 5×10^{-4}. We train the network with batch size 4, with each input to be some frame segments a video from the dataset. All layers except the last fully-connected layer adopted from ResNet50 is initialized with parameters of the one pretrained on the ImageNet dataset, while all other fc layers initialized with random Gaussian distribution with mean to be 0 and standard deviation to be 0.01. For Long Short-Term Memory layers, all weights and bias conforms to uniform distribution $\mathcal{U}(1/\sqrt{k}, 1/\sqrt{k})$ where $k = \frac{1}{83}$.

4.2 Datasets

EgoGesture Dataset [4] is a multi-modal large-scale dataset for egocentric hand gesture recognition, consisting of 2081 RGB videos with 24161 gesture samples and 2953224 frames. All data is marked with labels from 83 classes of static or dynamic gestures.

Our work is designed to be practical under real world conditions, thus only the RGB information from the dataset is used for training. Videos are resized to size of 224×224 using Antialias Filter. It is possible that one gesture contains more than 16 frames, which is how many the model accepts, we randomly sample 16 frames in the frame sequence while keeping the relative order.

4.3 Objective evaluation for interpretability

We follow the method in Grad-CAM++ [13] to evaluate the reliability of the explanations generated for our task: the hand gesture recognition in a video, which can be identified as a kind of classification. For every frame of the input video, named I, generate a corresponding explanation map by pointwise multiplication with the saliency map calculated by Grad-CAM++. That is:

$$E^c = L^c \circ I$$

where $\circ$ is the Hadamard product, c is the predicted class label. E^c is the explanation map under such prediction and L^c is the saliency map with respect to label c. An example of the final result in the form of the explanation map is given above in Fig. 5.

Then We test the performance of our method with three objective metrics [13]: 1. Average drop percentage; 2. Percentage increase in confidence; 3. Average increase percentage. We briefly describe them as below:

1. Average Drop Percentage [13]: A good explanation map should highlight the most relevant region in the process of decision-making. Also notice that the removal of parts of an image will cause the drop of the confidence of the model, compared to its confidence with the full image as input. Such drop should be less significant if we maintain the most relevant part of decision-making and only remove those irrelevant parts. With these observations, a better explanation map should preserve a higher confidence if we change the input of the model from the full image to the explanation map. That is, for comparison, the confidence drop between the setting with full image as input and the setting with the explanation map as input could be used to evaluate whether the visual explanation includes more of what is relevant when making decisions.

Follows from [13], the Average Drop Percentage is defined as:

$$\text{Average Drop Percentage} = \frac{100}{N}\left(\sum_{i=1}^{N}\frac{\max\{0, Y_i^c - O_i^c\}}{Y_i^c}\right)$$

where Y_i^c is the model's confidence for class c on the i-th full image and O_i^c is the same model's confidence in class c when only explanation map is used as input. Note that the drop percentage is bounded to 0 even if $O_i^c > Y_i^c$. This score is first computed per image and then averaged over the dataset.

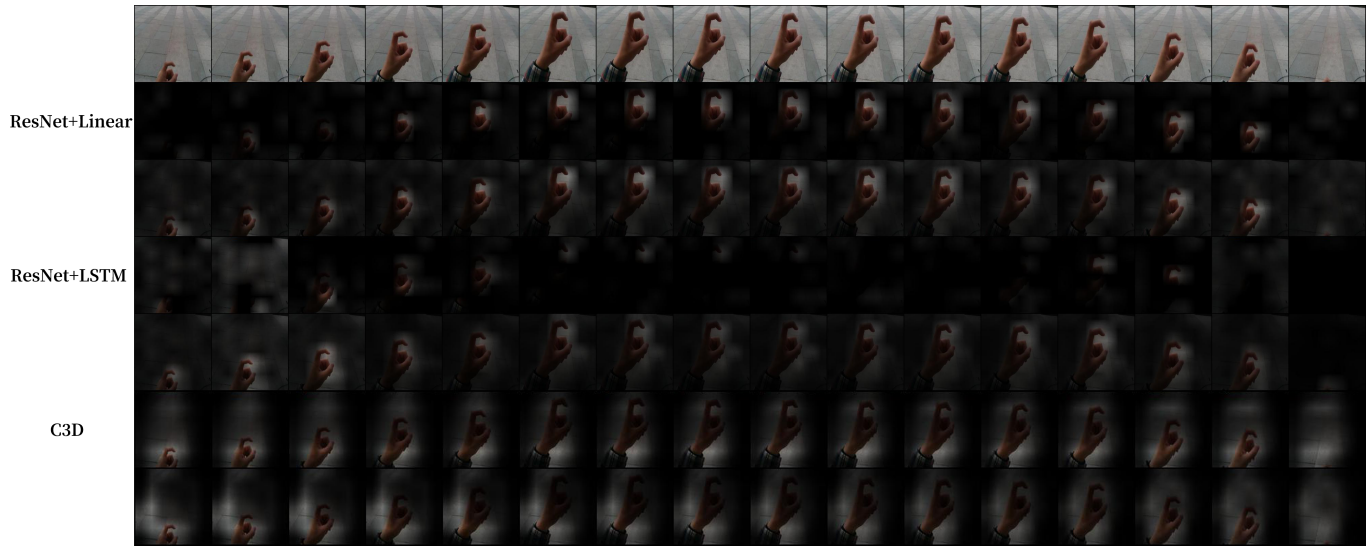

Fig. 5: Explanation map generated by three models. For each model, the first line is the result of (Seq-)Grad-CAM and the second line is the result of (Seq-)Grad-CAM++. In C3D model, the mentioned area keeps the same in most frames, and this is apparent in the last frame, where The center of the frame is marked but the hand actually disappears from the frame.

2. Percentage Increase in Confidence [13]: Though we can evaluate how model confidence would be affected if we replace the input full image with explanation map, we still ignored the cases when the confidence of the model increases. Consequently, as complement, we assess the number of times when such increase happens over the whole dataset. Formally, follows from [13], the Percentage Increase in Confidence is defined as,

$$\text{Percentage Increase in Confidence} = \frac{100}{N}\left(\sum_{i=1}^{N}\mathbf{1}_{O_i^c > Y_i^c}\right)$$

where $\mathbf{1}_{O_i^c > Y_i^c}$ is the indicator function takes 1 if $O_i^c > Y_i^c$ and 0 otherwise. Other notations follows the definition in the previous metric.

3. Average Increase Percentage: Close to the meaning of average drop percentage, we introduce the average increase percentage to evaluate the improvement when setting the explanation map as input. The average increase percentage is defined as:

$$\frac{1}{N}(\sum_{i=1}^{N}\frac{\max(0, O_i^c - Y_i^c)}{O_i^c})$$

where Y_i^c is the confidence with class c on the original picture and O_i^c is the corresponding one on the explanation map.

4.4 Human Trust Evaluation

We've given metrics used to demonstrate the faithfulness of proposed method in interpretability. Next, we'll conduct human studies to evaluate the human *trust* in our visualization. We generated explanation maps of images in EgoGesture Dataset, ranging from 83 classes with 500×16 images in total. C3D is the baseline model we used in this test. 20 human subjects who have no knowledge in hand gesture recognition, interpretability, deep learning, etc. are invited to our test. Explanation maps generated by our method and the baseline model C3D are shown to the subjects and they are asked to determine which explanation map earns more trust among the underlying models. The explanation algorithm that gains more votes is considered to invoke more human *trust*.

More specifically, for each image in image sequences used for interpretability test, we generate 6 explanation maps with original Grad-CAM/Grad-CAM++ on C3D, Seq-Grad-CAM/Seq-Grad-CAM++ on 2D-ResNet with Linear classifier and LSTM+Linear classifier. Examples of these explanation maps are presented in Fig. 5.

Then, we randomly arrange the order of explanation maps which corresponds to the same original image sequence and then show them with original image sequence as well as its caption to the subjects. The subjects are required to order all 6 explanation maps by the ability to exactly describe the given gesture. Then, they assign an score in $\{1, 2, 3, 4, 5, 6\}$ to each explanation map. The higher the score is, the better the explanation map interprets the gesture.

The subjects are also given the option to select *same* if they find no clear difference in two or more explanation maps. In this condition, Those with same performance get the same score which is the mean. Responses are normalized for each image sequence and then added which forms the score of human *trust*.

Moreover, we records the Top ratio, which means in how many testcases the explanation map generated by this method receives score 6.

4.5 Experimental results

1. Result on recognition accuracy: We evaluate the performance of different network structures on the EgoGesture Dataset as follows:

Table 1 Recognition accuracy and parameter size of the three models

Method	C3D	ResNet+Linear	ResNet+LSTM
Accuracy	0.864	0.895	0.912
# of trainable parameters	28335955	23788406	24048614

As the result mentions, although the ResNet is more complex, with same number of parameters, its performance is close to C3D. Moreover, with better classifier, the performance of our model is improved, so such method can be more explored.

2. Result on objective evaluation for interpretability: The results of experiments on EgoGesture as mentioned above is shown in Tabel 2 and Table 3.

As the two tables mention, our model performs better when we implement a more complex classifier. With an LSTM+Linear classifier, not only the recognition accuracy improves, but also

Table 2 Objective evaluation of ResNet with Linear classifier

Method	Seq-Grad-CAM++	Seq-Grad-CAM
Average Drop % (lower is better)	44.4	60.9
Average Increase % (higher is better)	10.7	4.2
% Increase in Confidence (higher is better)	5.2	7.7

Table 3 Objective evaluation of ResNet with LSTM+Linear classifier

Method	Seq-Grad-CAM++	Seq-Grad-CAM
Average Drop % (lower is better)	21.0	60.7
Average Increase % (higher is better)	5.3	3.0
% Increase in Confidence (higher is better)	22.8	6.5

the objective evaluation for interpretability with Grad-CAM++ is better(except for the average increase percentage).

3. Result on Human Trust: For each gesture, four explanation maps of each image are generated by using Grad-CAM, Grad-CAM++ on both our Res3D and C3D. Examples of some of these explanation maps are presented in Fig. 5. The class of each gesture is offered and the subjects are required to provide with an order of explanation maps.

Table 4 Human Trust Evaluation

Method		Score	Top ratio(%)
C3D	Grad-CAM	1.7	0.8
	Grad-CAM++	1.6	1.7
ResNet+ Linear	Seq-Grad-CAM	3.9	5.4
	Seq-Grad-CAM++	5.5	58.3
ResNet+ LSTM	Seq-Grad-CAM	3.2	3.3
	Seq-Grad-CAM++	5.2	30.5

As the table mentions, compared with original C3D, our model outperforms on the interpret task in human trust evaluation, while they have almost the same accuracy in classification task and the same number of parameters, which is an evaluation of the model's complexity.

In addition, as Fig. 5 shows, our model successfully solves the two problems mentioned in 3.3. In our model, 16 frames accurately focus on 16 areas of hands, instead of the condition that all frames share almost one area which is a mean of each.

5 Conclusion

In this study, we present a framework for gesture recognition and interpretation. We first design a network based on 2D ResNet to handle the recognition task, which slightly outperforms the commonly used C3D model. Then, we modify the original Grad-CAM and Grad-CAM++ algorithm in order to make it suitable for our network structure. The refinement solves both the mixture in time dimension and the difficulty for upsampling caused by 3D pooling layer. The objective evaluation as well as the human trust test demostrates that our framework significantly outperforms the C3D model in interpretation task with fewer parameters and even better recognition performance.

Acknowledgment

This work was advised and supported by our instructor Cewu Lu, Yong Yu, Weinan Zhang, and Junchi Yan. Our mentor Wenqiang Xu contributed equally to this work.

6 References

1 Allin, S. and Ramanan, D.: 'Assessment of post-stroke functioning using machine vision.'. MVA, 2007. pp. 299–302
2 Pansare, J.R., Gawande, S.H. and Ingle, M.: 'Real-time static hand gesture recognition for american sign language (asl) in complex background', , 2012,
3 Jang, Y., Noh, S.T., Chang, H.J., Kim, T.K. and Woo, W.: '3d finger cape: Clicking action and position estimation under self-occlusions in egocentric viewpoint', *IEEE Transactions on Visualization and Computer Graphics*, 2015, **21**, (4), pp. 501–510
4 Zhang, Y., Cao, C., Cheng, J. and Lu, H.: 'Egogesture: a new dataset and benchmark for egocentric hand gesture recognition', *IEEE Transactions on Multimedia*, 2018, **20**, (5), pp. 1038–1050
5 Yang, M.H., Ahuja, N. and Tabb, M.: 'Extraction of 2d motion trajectories and its application to hand gesture recognition', *IEEE Transactions on pattern analysis and machine intelligence*, 2002, **24**, (8), pp. 1061–1074
6 Jones, M.J. and Rehg, J.M.: 'Statistical color models with application to skin detection', *International Journal of Computer Vision*, 2002, **46**, (1), pp. 81–96
7 Zhou, Y., Jiang, G. and Lin, Y.: 'A novel finger and hand pose estimation technique for real-time hand gesture recognition', *Pattern Recognition*, 2016, **49**, pp. 102–114
8 Chen, Q., Georganas, N.D. and Petriu, E.M.: 'Real-time vision-based hand gesture recognition using haar-like features'. 2007 IEEE instrumentation & measurement technology conference IMTC 2007, 2007. pp. 1–6
9 Kokic, M., Kragic, D. and Bohg, J.: 'Learning to estimate pose and shape of hand-held objects from rgb images', *arXiv preprint arXiv:190303340*, 2019,
10 Garcia.Hernando, G., Yuan, S., Baek, S. and Kim, T.K.: 'First-person hand action benchmark with rgb-d videos and 3d hand pose annotations'. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018. pp. 409–419
11 Zimmermann, C. and Brox, T.: 'Learning to estimate 3d hand pose from single rgb images'. Proceedings of the IEEE international conference on computer vision, 2017. pp. 4903–4911
12 Yang, S., Liu, J., Lu, S., Er, M.H. and Kot, A.C.: 'Collaborative learning of gesture recognition and 3d hand pose estimation with multi-order feature analysis'. European Conference on Computer Vision, 2020. pp. 769–786
13 Chattopadhyay, A., Sarkar, A., Howlader, P. and Balasubramanian, V.N.: 'Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks', *CoRR*, 2017, **abs/1710.11063**. Available from: `http://arxiv.org/abs/1710.11063`
14 Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M.: 'Learning spatiotemporal features with 3d convolutional networks'. Proceedings of the IEEE international conference on computer vision, 2015. pp. 4489–4497
15 Garg, P., Aggarwal, N. and Sofat, S.: 'Vision based hand gesture recognition', *World academy of science, engineering and technology*, 2009, **49**, (1), pp. 972–977
16 Dipietro, L., Sabatini, A.M. and Dario, P.: 'A survey of glove-based systems and their applications', *Ieee transactions on systems, man, and cybernetics, part c (applications and reviews)*, 2008, **38**, (4), pp. 461–482
17 LaViola, J.: 'A survey of hand posture and gesture recognition techniques and technology', *Brown university, providence, ri*, 1999, **29**
18 Ibraheem, N.A. and Khan, R.Z.: 'Survey on various gesture recognition technologies and techniques', *International journal of computer applications*, 2012, **50**, (7)
19 Oudah, M., Al.Naji, A. and Chahl, J.: 'Hand gesture recognition based on computer vision: a review of techniques', *journal of Imaging*, 2020, **6**, (8), pp. 73
20 Brand, J. and Mason, J.S.: 'A comparative assessment of three approaches to pixel-level human skin-detection'. Proceedings 15th International Conference on Pattern Recognition. ICPR-2000. vol. 1, 2000. pp. 1056–1059
21 Chai, D. and Bouzerdoum, A.: 'A bayesian approach to skin color classification in ycbcr color space'. 2000 TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium (Cat. No. 00CH37119). vol. 2, 2000. pp. 421–424
22 Menser, B. and Wien, M.: 'Segmentation and tracking of facial regions in color image sequences'. Visual Communications and Image Processing 2000. vol. 4067, 2000. pp. 731–740
23 Kulkarni, V.S. and Lokhande, S.: 'Appearance based recognition of american sign language using gesture segmentation', *International Journal on Computer Science and Engineering*, 2010, **2**, (03), pp. 560–565
24 Licsár, A. and Szirányi, T.: 'User-adaptive hand gesture recognition system with interactive training', *Image and Vision Computing*, 2005, **23**, (12), pp. 1102–1114
25 Oyedotun, O.K. and Khashman, A.: 'Deep learning in vision-based static hand gesture recognition', *Neural Computing and Applications*, 2017, **28**, (12), pp. 3941–3951
26 Jiang, F., Wu, S., Yang, G., Zhao, D. and Kung, S.: 'independent hand gesture recognition with kinect', *Signal, Image and Video Processing*, 2014, **8**, (1), pp. 163–172
27 Ma, X. and Peng, J.: 'Kinect sensor-based long-distance hand gesture recognition and fingertip detection with depth information', *Journal of Sensors*, 2018, **2018**
28 De.Smedt, Q., Wannous, H., Vandeborre, J.P., Guerry, J., Saux, B.L. and Filliat, D.: '3d hand gesture recognition using a depth and skeletal dataset: Shrec'17 track'. Proceedings of the Workshop on 3D Object Retrieval, 2017. pp. 33–38
29 Tekin, B., Bogo, F. and Pollefeys, M.: 'H+ o: Unified egocentric recognition of 3d hand-object poses and interactions'. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. pp. 4511–4520
30 Ge, L., Ren, Z. and Yuan, J.: 'Point-to-point regression pointnet for 3d hand pose estimation'. Proceedings of the European conference on computer vision (ECCV), 2018. pp. 475–491
31 Wan, C., Probst, T., Gool, L.V. and Yao, A.: 'Self-supervised 3d hand pose estimation through training by fitting'. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. pp. 10853–10862

32 Chung, H.Y., Chung, Y.L. and Tsai, W.F.: 'An efficient hand gesture recognition system based on deep cnn.'. ICIT, 2019. pp. 853–858

33 Li, G., Tang, H., Sun, Y., Kong, J., Jiang, G., Jiang, D., et al.: 'Hand gesture recognition based on convolution neural network', *Cluster Computing*, 2019, **22**, (2), pp. 2719–2729

34 Lin, H.I., Hsu, M.H. and Chen, W.K.: 'Human hand gesture recognition using a convolution neural network'. 2014 IEEE International Conference on Automation Science and Engineering (CASE), 2014. pp. 1038–1043

35 Alnaim, N., Abbod, M. and Albar, A.: 'Hand gesture recognition using convolutional neural network for people who have experienced a stroke'. 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2019. pp. 1–6

36 Wu, X.Y.: 'A hand gesture recognition algorithm based on dc-cnn', *Multimedia Tools and Applications*, 2019, pp. 1–13

37 Nguyen, X.S., Brun, L., Lézoray, O. and Bougleux, S.: 'A neural network based on spd manifold learning for skeleton-based hand gesture recognition'. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. pp. 12036–12045

38 Molchanov, P., Gupta, S., Kim, K. and Kautz, J.: 'Hand gesture recognition with 3d convolutional neural networks'. Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015. pp. 1–7

39 Bao, P., Maqueda, A.I., del Blanco, C.R. and García, N.: 'Tiny hand gesture recognition without localization via a deep convolutional network', *IEEE Transactions on Consumer Electronics*, 2017, **63**, (3), pp. 251–257

40 Springenberg, J.T., Dosovitskiy, A., Brox, T. and Riedmiller, M.: 'Striving for simplicity: The all convolutional net', *arXiv preprint arXiv:14126806*, 2014,

41 Zeiler, M.D. and Fergus, R.: 'Visualizing and understanding convolutional networks'. European conference on computer vision, 2014. pp. 818–833

42 Mahendran, A. and Vedaldi, A.: 'Visualizing deep convolutional neural networks using natural pre-images', *International Journal of Computer Vision*, 2016, **120**, (3), pp. 233–255

43 Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D. and Batra, D.: 'Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization', *CoRR*, 2016, **abs/1610.02391**. Available from: `http://arxiv.org/abs/1610.02391`

44 Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. and Lipson, H.: 'Understanding neural networks through deep visualization', *arXiv preprint arXiv:150606579*, 2015,

45 Simonyan, K., Vedaldi, A. and Zisserman, A.: 'Deep inside convolutional networks: Visualising image classification models and saliency maps', *arXiv preprint arXiv:13126034*, 2013,

46 Dosovitskiy, A. and Brox, T.: 'Inverting convolutional networks with convolutional networks', *arXiv preprint arXiv:150602753*, 2015, **4**

47 Fong, R.C. and Vedaldi, A.: 'Interpretable explanations of black boxes by meaningful perturbation'. Proceedings of the IEEE International Conference on Computer Vision, 2017. pp. 3429–3437

48 Shrikumar, A., Greenside, P. and Kundaje, A.: 'Learning important features through propagating activation differences', *arXiv preprint arXiv:170402685*, 2017,

49 Smilkov, D., Thorat, N., Kim, B., Viégas, F. and Wattenberg, M.: 'Smoothgrad: removing noise by adding noise', *arXiv preprint arXiv:170603825*, 2017,

50 Sundararajan, M., Taly, A. and Yan, Q.: 'Axiomatic attribution for deep networks', *arXiv preprint arXiv:170301365*, 2017,

51 Ribeiro, M.T., Singh, S. and Guestrin, C.: '" why should i trust you?" explaining the predictions of any classifier'. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016. pp. 1135–1144

52 Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. and Torralba, A.: 'Learning deep features for discriminative localization'. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. pp. 2921–2929

53 He, K., Zhang, X., Ren, S. and Sun, J.: 'Deep residual learning for image recognition'. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. pp. 770–778