Unsupervised computation of salient motion maps from the interpretation of a frame-based classification network

Meunier Etienne
etienne.meunier@inria.fr
Bouthemy Patrick
patrick.bouthemy@inria.fr

Inria
Centre Rennes - Bretagne Atlantique
Rennes, France

1 LRP Rules

Following description in [1, 2], we compute the importance score from the prediction output to the input layer $L = 0$, and use the attribution scores (R^0) at this layer as the LRP attribution map.

Z_+ rule, given in [1, 2], is applied for all linear and convolutional layers except the first :

$$z^+\text{-rule: } R_i^L = \sum_j \frac{x_i w_{ij}^+}{\sum_{i'} x_{i'} w_{i'j}^+} R_j^{L+1}.$$

Z_β rule, given in [1, 2], is applied for the first convolutional layer to deal with negative values in input flow :

$$z^\beta \text{ -rule: } R_i^L = \sum_j \frac{x_i w_{ij} - lw_{ij}^+ - hw_{ij}^-}{\sum_{i'} x_{i'} w_{i'j} - lw_{i'j}^+ - hw_{i'j}^-} R_j^{L+1},$$

where x_i is the input value at layer L and $R^L \in \mathbb{R}_+^I$ the relevance score associated to this input map. Weights in the layer L are denoted w_{ij}. We define $w_{ij}^+ = w_{ij} * Id\{w_{ij} > 0\}$ and $w_{ij}^- = w_{ij} * Id\{w_{ij} < 0\}$.

In contrast to image intensities, the components of the flow vectors are not restricted to a predefined bounded range. Thus, we have to adopt a different normalisation technique. For each input flow x, we compute l and h as the minimal and maximal values of the channel corresponding to each flow component.

2 Layers Permutation

Considering $X : \{x_1, x_2, \cdots x_n;\ x_i \in \mathbb{R}\}$ a local neighborhood of the input, the results of the computation using the network training order (Max Pool, Batch Norm, Relu) is :

$$\tilde{x} \triangleq relu(\gamma \frac{\max_{x_i \in X}(x_i) - E_r}{\sqrt{V_r}} + \beta),$$

Figure 1: Modification of the ordering of the network inner layers for the interpretation stage. Left: Original order used for training. Right: Order used for interpretation. Weights of each layer remain unchanged. For interpretation, Convolution and Batch Norm layers are merged into a functionally equivalent convolutional layer.

where $relu(x) = max(x,0)$. E_r and V_r are representing respectively the running estimate of the expectation and the variance accumulated during training phase and used as a fixed value during test phase, and γ, β are the scaling and shift parameters of the batch norm layer learned during training. Thus, if the condition $\gamma \geq 0$ holds, which we verified in all our experiments, we can write :

$$\tilde{x} = \max_{x_i \in X}(relu(\gamma \frac{x_i - E_r}{\sqrt{V_r}} + \beta)).$$

This second computation is corresponding to the "Interpretation order" (Batch Norm, Relu, Max Pool) described in the paper. In cases where the running estimate of the expectation and the variance are not computed, we can proceed a first forward step through the training network to retrieve the expectation and variance value for a batch, and then, use those values in the interpretation network. Note that this whole manipulation is not necessary if in the original training order the normalisation layer is already adjacent to the foregoing convolutional layer. In this case, we can directly proceed to the interpretation step.

After this step, the justification of the fusion between the Convolution and Normalisation layers is given in [3].

3 Model Randomization Test

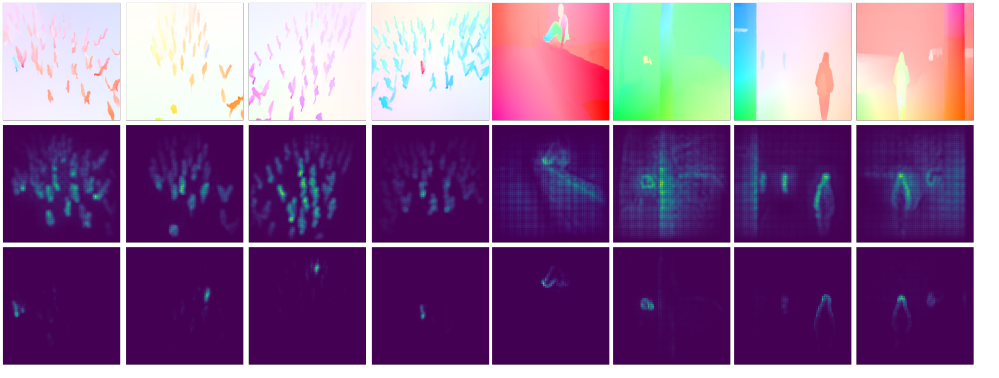

Figure 2: From top to bottom : Input optical flow displayed with the classical HSV color code, LRP attribution map from the random classification network and LRP attribution maps obtained using our trained classification network.

Interpretation methods can be sensitive to the structure of the network being analyzed and elements in the input images such as edges [4]. We want to ensure that our attribution (or interpretation) maps depend on the learned parameters of the network and not of intrinsic characteristics of our input flow maps only. Thus, we followed the recommendation in [4] and performed a model parameter randomization test. This test consists in comparing attribution maps obtained using our trained network and a random network. By visual inspection of Fig.2, we can observe that the attribution maps exhibit important differences. While attribution maps of the trained network are focusing only on points that exhibit salient motion, attribution maps of the random network highlight all points with an apparent motion indistinctively of its real saliency. This test confirms the pivotal role of training the network on a classification task to obtain meaningful attribution maps.

References

[1] Jindong Gu, Yinchong Yang, and Volker Tresp. Understanding individual decisions of CNNs via contrastive backpropagation. In *Asian Conf. on Computer Vision (ACCV)*, Perth, Australia, 2018.

[2] Grégoire Montavon et al. Layer-wise relevance propagation: An overview. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, vol.11700:193–209. Springer, 2019.

[3] Mathilde Guillemot et al. Breaking batch normalization for better explainability of deep neural networks through layer-wise relevance propagation. *arXiv:2002.11018*, Feb. 2020.

[4] Julius Adebayo et al. Sanity checks for saliency maps. *arXiv:1810.03292*, Nov. 2020.