**Introducing and assessing the explainable AI (XAI) method: SIDU**

<https://arxiv.org/pdf/2101.10710.pdf>

**0. Related Works – Visual Explanation**

|  |
| --- |
| **Saliency maps:** visual explanation을 위해 사용된다.  **Back-propagation methods:**  Feature signal을 output neuron으로부터 input neuron까지 single pass를 통해 전달한다. (Layer-wise Relevance Propagation, DeCovNet 등)  **Gradient-based methods:**  Backpropagation 알고리즘에 Gradient 또는 그것의 수정된 버전을 적용하여, CNN의 output의 detivative를 그것의 input에 대해 시각화한다. (Grad-CAM 등)   * Pixel space에서 gradient ascent를 적용하여 class-specific saliency map을 생성한다. * 이 synthesized image는 class-specific visualization으로 작용하며 주어진 CNN이 해당 class에 대해 어떻게 모델링되었는지 이해할 수 있게 한다.   **Perturbation-based methods:**  Output으로의 전체적인 변화를 트래킹하는 동안 Input이 perturb된다. (state-of-the-art RISE 등)   * 어떤 경우에는 모델의 중간 레이어에서 변화가 발생할 수 있다.   **Approximation-based method:**  Deep CNN을 보다 간단한 approximation으로 대체하며, visual explanation을 더 쉽게 생성할 수 있다. (LIME 알고리즘 등)   * Decision Tree도 같은 목적으로 이용할 수 있지만, visual input에 명시적으로 적용할 수 없다. |

**1. SIDU: Proposed Method**

|  |
| --- |
| **< Figure 2 >** |
| **NEXT PAGE** |

|  |
| --- |
| **PREVIOUS PAGE** |
| Gradient-based 모델과 perturbation explanation method에 의해 생성된 **visual explanation은 object class의 전체 salient region을 localize하는 데 실패**한다. 이를 극복하기 위해 본 논문에서는 SIDU 방법을 제안한다.   * 여기서는 **어떤 주어진 CNN 모델에 대해서도 계속적으로 더 좋은 explanation method를 제공할 수 있는 XAI method**를 제안한다. * 제안하는 방법은 **마지막 Convolution layer를 이용하여 mask**를 생성한다.   + 이들 mask로부터 Similarity Difference와 Uniqueness score가 계산되며, CNN 모델의 최종 결정에 대한 설명을 생성한다. * **< Figure 2 >** 는 SIDU에 대한 overview이다.   이 방법은 **총 3개의 step**으로 구성된다.   * 1. CNN으로부터 마지막 convolution layer를 추출하여, 그 layer를 이용하여 mask를 생성한다. * 2. 예측된 class에 대해 각 mask에 대한 Similarity Difference를 계산한다. * 3. 각 mask에 대한 weight을 계산하고, prediction을 설명할 수 있는 final map으로 합성한다. |

**2. Step 1. Generating Feature Activation Masks**

|  |
| --- |
| CNN model의 예측된 output에 대한 visual explanation을 생성하기 위해, 먼저 마지막 convolution layer로부터 feature image mask를 생성한다.   * 어떤 **deep CNN model F**에 대해서도, **크기가 n\*n\*N인 마지막 convolution layer**에 대해 다음을 수행한다. (n: conv layer의 size, N: feature의 개수)   + 각각의 feature activation map 는 binary mask 로 바뀐다.   + 이들 각각의 binary mask는 bi-linear interpolation을 이용하여 upsampling된다.   + Binary mask는 [0, 1] 사이의 값을 가지며 더 이상 binary하지 않다.   + Feature activation mask (up-sampled) 와 입력 이미지 I 간에 point-wise multiplication을 수행한다.   + Feature image mask 를 얻는다. |

**3. Step 2. Computing Similarity Differences and Uniqueness**

|  |
| --- |
| 마지막 convolutional layer의 size가 **n\*n\*N**이라고 할 때, **전체 feature activation mask의 개수는 N**이다.  1. class c의 모든 mask 는 각각 서로 같은 CNN 모델 F를 이용하여 feature image mask를 생성하는데, 이들 **mask 각각에 대해서 probability prediction score를 계산**한다.   * Feature activation image mask 에 대한 probability prediction score는 로 정의되며, 주어진 입력 이미지 I에 대한 probability prediction score는 로 정의된다. * Prediction score의 벡터 크기는 CNN 모델의 학습에 사용되는 class의 개수에 따라 달라진다. (예: 1000개의 object class일 때 의 크기는 1\*1000)   2. 모든 image mask와 입력 이미지 I에 대하여 prediction score를 계산한 후에는 각각의 ***와* 에 대해서 similarity difference를 계산**한다.   * 이 두 벡터에 대한 similarity difference는 해당하는 original input image와 feature image mask 간의 relevance를 나타낸다.   + Feature가 알려지지 않았을 때 예측이 얼마나 달라지는지를 측정하기 위함 * Feature activation image mask의 relevance value는 예측된 class와 비슷할수록 높다. * **Feature activation map 의 Similarity difference**는 다음과 같이 주어진다.   3. similarity difference를 측정한 다음에는 **feature image mask prediction score vector 간에 uniqueness measure 를 계산**한다.   * 다른 region들과 달리 눈에 띄는 image region은 특정한 관점에서 우리의 눈길을 사로잡는다. 이런 region들은 highly salient region이라고 불린다.   + False region을 low weight으로 압축하고 actual region을 higher weight으로 highlight하기 때문이다. * Uniqueness는 다음과 같이 계산된다.   4. **각 feature importance의 weight 를 계산**한다. 이때 다음과 같이 similarity difference와 uniqueness measure에 대한 dot product를 이용한다. |

**4. Step 3. Explanations for the prediction**

|  |
| --- |
| **Feature activation mask 와 이에 대응되는 feature importance weight 의 weighted sum**을 통해, **예측된 output class c에 대한 설명을 도출**한다.   * Weight은 와 같이 계산된다.     **<Figure 4>**    **<Figure 5>**  heatmap으로 표현된 **visual explanation map** , 즉 class discriminative localization map은 **<Figure 5>**와 같다.   * 예측된 Class c에 대한 Visual explanation은 다음과 같은 수식으로 나타낸다. |

**5. Evaluation**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Explanation map을 각 모델에 대해서 비교**하면 다음과 같다. (자료: natural images class of ImageNet, 가장 오른쪽이 SIDU)    **각 dataset과 model에 대해 XAI method의 성능을 비교한 결과**는 다음과 같다.   |  |  |  |  | | --- | --- | --- | --- | | Dataset | ImageNet validation set | ImageNet validation set | RFIQA | | Model | ResNet-50 | VGG-16 | ResNet-50 | | Result |  |  |  |   Eye-tracking data collection에서 주어진 object class를 **사람이** 인식한 결과는 다음과 같다. |
| **NEXT PAGE** |

|  |
| --- |
| **PREVIOUS PAGE** |
| 각 **XAI method의 saliency map (with eye fixation map)**은 다음과 같다.    각 XAI method에 대한 **visual explanation을 human visual explanation과 비교**하면 다음과 같다. (정가운데가 SIDU)    Medical RFIQA dataset에 대한 **Expert level evaluation**은 다음과 같다. |
| **NEXT PAGE** |

|  |
| --- |
| **PREVIOUS PAGE** |
| Eye fundus image에 대한 **Good/Bad quality image의 visual explanation**은 다음과 같다.   * RFIQA dataset에 대해서 **RISE, Grad-CAM과 SIDU를 각각 사용**하였다. * Good quality image는 위쪽 행, Bad quality image는 아래쪽 행이다.     **Noise를 적용했을 때의 성능**은 다음과 같다. |
| **NEXT PAGE** |

|  |
| --- |
| **PREVIOUS PAGE** |
| FGSM noise를 적용한 이미지에 대한 human visual explanation의 heatmap은 다음과 같다.     * 1번째, 2번째, 3번째 행은 각각 **SIDU, GRAD-CAM, RISE를 적용한 결과**이다. * (a)는 원래 이미지, (b)는 eye-tracker의 결과, (c), (d), (e), (f)는 각각 일 때의 heatmap이다.   **Adversarial attack에 의해 원래의 visual explanation heatmap이 틀리게 되는** adversarial example에 대한 visual explanation heatmap은 다음과 같다. |