INTEGRATED GRAD-CAM: SENSITIVITY-AWARE VISUAL EXPLANATION OF DEEPCONVOLUTIONAL NETWORKS VIA INTEGRATED GRADIENT-BASED SCORING

Sam Sattarzadeh¹, Mahesh Sudhakar¹, K. N. Plataniotis¹, Jongseong Jang², Yeonjeong Jeong², Hyunwoo Kim²

1. The Edward S. Rogers Sr. Department of Electrical & Computer Engineering, University of Toronto
2. LG AI Research
Overview of the presentation

- Explainable AI: Motivation, Applications
- Problem statement
- Our proposed method: Integrated Grad-CAM (IG-CAM)
- Empirical results
- Conclusion
- References
Motivation

Explainable AI (XAI):
Understanding Convolutional Neural Networks (CNNs) is crucial for high-impact and high-risk applications in computer vision[1,2].

CNN-specific attribution methods:
Visualizing the input features responsible for CNN prediction. (A branch of post-hoc and local XAI algorithms)

Impactful in:
➢ Industrial Applications: Medicine, Autonomous Driving, Criminal Justice, Finance
➢ Research Fields: Object Recognition, Semantic Segmentation, Model Debugging, Dataset Bias Detection, etc.

Terminology:
Post-hoc: models the behavior of the target model after training has concluded.
Local: Illustrates the relationship between the outcome of the target model with the input

References:
Visual explanation algorithms:

- **Backpropagation-based methods**: Calculating the gradient of a model’s output to the input features or the hidden neurons (e.g., Vanilla Gradient, Integrated Gradient, SmoothGrad).

- **CAM-based methods**: Visualizing the features extracted in a single layer of the CNNs (e.g., Grad-CAM, Grad-CAM++, Score-CAM).

- **Perturbation-based methods**: Probing the model’s behavior using perturbed copies of the input image (e.g., RISE, Extremal Perturbation, Occlusion).

**Our focus: CAM-based methods**
Specialized for CNNs, utilized for interpretation and high-level feature visualization.
Problem Statement

CAM-based techniques for CNN interpretation:

- **Grad-CAM**\(^3\): Feature map-wise Gradient-based Weighting.
- **Grad-CAM++**\(^4\): Pixel-wise Gradient-based Weighting.
- **XGrad-CAM**\(^5\): Feature map-wise Axiom-based Weighting.

---

**Our approach: Integrated Grad-CAM**

- Addressing the limitations of backpropagation in explaining non-linear models.
- Solving the gradient limitations by employing gradients!

---


Integrated Grad-CAM
Integrated Grad-CAM: Intuition

Integrated Gradients\[^{6}\]:

- **Addressing the issues in the method “Vanilla Backpropagation”**.

- **Guarantees the Sensitivity axiom:**
  “For each pair of input and baseline differing only in one feature, an attribution method should highlight this difference by assigning different values corresponding to that feature.”

- **Idea:** Calculating the integral of gradient values in a path that links a specific baseline to the input.

- **Takeaways:**
  Enhanced clarity of explanations.
  Improved estimation of the features’ contribution in the model’s prediction.

Path Integral

Path Information:
\[ \gamma(\alpha) = I' + f(\alpha) \times (I - I') \quad (0 \leq \alpha \leq 1) \]
\[ f(\alpha) : \mathbb{R} \to \mathbb{R} : \text{Differentiable & Monotonically Increasing} \]
\[ f(0) = 0 \quad \& \quad f(1) = 1 \]

Integral Gradients:
For each pair of functions \((h(.), g(.))\):
\[ \text{PathIG}_{h,g}(I) = \int_{\alpha=0}^{1} \frac{\partial h(\gamma(\alpha))}{\partial g(\gamma(\alpha))} \left[ g(\gamma(\alpha)) - g(I') \right] d\alpha \]

Some paths linking \(I\) and \(I'\) in the image domain
(Input: \(I\) - Baseline: \(I'\))
Integrated Grad-CAM: Intuition

How “Integrated Gradients” can estimate the features’ importance more accurately than “Vanilla Gradient”?

Example: $i_1 = i_2 = 1 \rightarrow y = 1$

Importance of $i_1$ in the model’s prediction ($S(i_1)$):

Vanilla Gradient: $S(i_1) = \frac{\partial y}{\partial i_1} \mid i_1 = i_2 = 1 = 0$

Integrated Gradients:

\[
\begin{align*}
\gamma(\alpha) &= \left[0, \alpha \times \left[\frac{i_1}{i_2}\right] \right] = \left[\frac{\gamma_1}{\gamma_2}\right] \\
S(i_1) &= \int_{\alpha=0}^{\alpha=0.5} \frac{\partial y}{\partial y_1(\alpha)} \gamma_1(\alpha) \, d\alpha = 0.5
\end{align*}
\]

The same idea can be proposed to improve Grad-CAM!

Grad-CAM formulation

While feeding the CNN with the input image "I":
Feature maps in the convolutional layer "l":
\{A^{1l}(I), A^{2l}(I), ..., A^{Nl}(I)\}
Model’s confidence score for class "c": \(y_c(I)\)

Our modification: replacing gradient terms with integrated gradient terms

Grad-CAM Explanation map:

\[ M^c_{Grad-CAM} = \text{ReLU} \left( \sum_{k=1}^{N} \left( \sum_{i,j} \frac{\partial y_c(I)}{\partial A_{i,j}^{kl}(I)} A^{kl}(I) \right) \right) \]
Integrated Grad-CAM formulation

While feeding the CNN with the input image "I"

Feature maps in the convolutional layer "l":

\{A^{1l}(I), A^{2l}(I), ..., A^{Nl}(I)\}

Model’s confidence score for class "c": \(y_c(I)\)

Integrated Grad-CAM Explanation map:

\[ M_{IG-CAM}^c = \int_{\alpha=0}^{1} \text{ReLU}(\sum_{k=1}^{N}(\sum_{i,j} \frac{\partial y_c(y(\alpha))}{\partial A_{i,j}^{kl}(y(\alpha))})(A^{kl}(y(\alpha)) - (A^{kl}(I')))d\alpha \]

Grad-CAM Explanation map:

\[ M_{Grad-CAM}^c = \text{ReLU}(\sum_{k=1}^{N}(\sum_{i,j} \frac{\partial y_c(I)}{\partial A_{i,j}^{kl}(I)})A^{kl}(I)) \]
**Integrated Grad-CAM formulation**

**Limitation of our equation:**
The equation below is hard to implement.

**Solution:**
Approximating our equation with a summation.

- For simplicity, select a linear path between the input and the baseline.
- Use Reimann’s Approximation.

**Path** $P_2$: $\gamma(\alpha) = I' + f(\alpha) \times (I - I')$ \hspace{1cm} (0 ≤ $\alpha$ ≤ 1)

\[ f(\alpha) = \alpha \]

**Integrated Grad-CAM Explanation map:**
\[ M^C_{IG-CAM} = \int_{\alpha=0}^{1} \text{ReLU} \left( \sum_{k=1}^{N} \sum_{i,j} \frac{\partial y_c(\gamma(\alpha))}{\partial A_{i,j}^{kl}(\gamma(\alpha))} \left( A_{kl}^{\gamma(\alpha)} - (A_{kl}^{I'}(I')) \right) \right) \, d\alpha \]
Integrated Grad-CAM implementation

**Path** $P_2$: $\gamma(\alpha) = I' + \alpha \times (I - I')$ \hspace{0.5cm} (0 \leq \alpha \leq 1)

Reimann’s Approximation:
Sample $m$ points along the path $P_2$ with a constant interval.

Interval step: $\frac{1}{m}$ \hspace{0.3cm} (m $\in \mathbb{N}$)

Sampled points: $\alpha \in \{\frac{t}{m} | t = \{1, ..., m\}\}$

$$M_{IG-CAM}^{c} = \int_{0}^{1} \text{ReLU}\left( \sum_{k=1}^{N} \left( \sum_{i,j} \frac{\partial y_{c}(\gamma(\alpha))}{\partial A_{i,j}^{kl}(\gamma(\alpha))} \right) \left( A^{kl}(\gamma(\alpha)) - (A^{kl}(I')) \right) \right) d\alpha$$

$$M_{IG-CAM}^{c} \approx \sum_{t=1}^{m} \text{ReLU}\left( \frac{1}{m} \sum_{k=1}^{N} \left( \sum_{i,j} \frac{\partial y_{c}(\gamma\left(\frac{t}{m}\right))}{\partial A_{i,j}^{kl}(\gamma\left(\frac{t}{m}\right))} \right) \left( A^{kl}\left(\gamma\left(\frac{t}{m}\right)\right) - (A^{kl}(I')) \right) \right) d\alpha$$

Some paths linking $I$ and $I'$ in the image domain

(Input: $I$ - Baseline: $I'$)
IG-CAM can be modelled by applying Grad-CAM to translated copies of the input image.
PASCAL VOC 2007[5]:

- Purpose: Multi-label image classification, Object Detection.
- Containing 4963 test images in 20 classes, Bounding boxes provided.
- A VGG-16 model and a ResNet-50 model trained on this dataset are utilized[4].
- In out experiments, the number of intervals for IG-CAM is set to $m=20$.

Quantitative evaluation: metrics

Ground truth-based metrics
Verifying the meaningfulness of explanation methods, and their ability in feature visualization.
- Energy-based pointing game\[^8\] (The fraction of energy inside an explanation map captured in a bounding box.)
- Bounding box\[^9\] (Adaptive version of mean Intersection over Union (mIoU)).

Model truth-based metrics
Justifying the faithfulness and validity of the explanation maps from the perspective of the model.
- Drop rate\[^{10}\] (Measuring the average drop in the model's confidence score (if drops), when only the top 15% of the pixels are retained).
- Increase rate\[^{10}\] (Measuring the rate of increase in the model’s confidence score, when only the top 15% of the pixels are retained).

Empirical Results

Dataset: PASCAL VOC 2007

<table>
<thead>
<tr>
<th>Metric</th>
<th>Grad-CAM</th>
<th>Grad-CAM++</th>
<th>Integrated Grad-CAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EBPG(%)</td>
<td>55.44</td>
<td>46.29</td>
<td>55.94</td>
</tr>
<tr>
<td>Bbox(%)</td>
<td>51.7</td>
<td>55.59</td>
<td>55.6</td>
</tr>
<tr>
<td>Drop(%)</td>
<td>49.47</td>
<td>60.63</td>
<td>47.96</td>
</tr>
<tr>
<td>Increase(%)</td>
<td>31.08</td>
<td>23.89</td>
<td>31.47</td>
</tr>
<tr>
<td>ResNet-50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EBPG(%)</td>
<td>60.08</td>
<td>47.78</td>
<td>60.41</td>
</tr>
<tr>
<td>Bbox(%)</td>
<td>60.25</td>
<td>58.66</td>
<td>61.94</td>
</tr>
<tr>
<td>Drop(%)</td>
<td>35.80</td>
<td>41.77</td>
<td>34.49</td>
</tr>
<tr>
<td>Increase(%)</td>
<td>36.58</td>
<td>32.15</td>
<td>36.84</td>
</tr>
</tbody>
</table>

For each metric, the best is shown in bold.

Ground truth-based metrics
Verifying the meaningfulness of explanation methods, and their preciseness in feature visualization.

➢ Energy-based pointing game\(^8\) (The fraction of energy inside an explanation map captured in a bounding box.)
➢ Bounding box\(^9\) (Adaptive version of mean Intersection over Union (mIoU)).

Empirical Results

Model truth-based metrics
Justifying the faithfulness and validity of the explanation maps from the perspective of the model.

➢ **Drop rate\(^ {[10]}\)** (Measuring the average drop in the model’s confidence score (if drops), when only the top 15% of the pixels are retained).

➢ **Increase rate\(^ {[10]}\)** (Measuring the rate of increase in the model’s confidence score, when only the top 15% of the pixels are retained).

---

**Dataset: PASCAL VOC 2007**

<table>
<thead>
<tr>
<th></th>
<th>Metric</th>
<th>Grad-CAM</th>
<th>Grad-CAM++</th>
<th>Integrated Grad-CAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG16</td>
<td>EBPG(%)</td>
<td>55.44</td>
<td>46.29</td>
<td>55.94</td>
</tr>
<tr>
<td></td>
<td>Bbox(%)</td>
<td>51.7</td>
<td>55.59</td>
<td>55.6</td>
</tr>
<tr>
<td></td>
<td>Drop(%)</td>
<td>49.47</td>
<td>60.63</td>
<td>47.96</td>
</tr>
<tr>
<td></td>
<td>Increase(%)</td>
<td>31.08</td>
<td>23.89</td>
<td>31.47</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>EBPG(%)</td>
<td>60.08</td>
<td>47.78</td>
<td>60.41</td>
</tr>
<tr>
<td></td>
<td>Bbox(%)</td>
<td>60.25</td>
<td>58.66</td>
<td>61.94</td>
</tr>
<tr>
<td></td>
<td>Drop(%)</td>
<td>35.80</td>
<td>41.77</td>
<td>34.49</td>
</tr>
<tr>
<td></td>
<td>Increase(%)</td>
<td>36.58</td>
<td>32.15</td>
<td>36.84</td>
</tr>
</tbody>
</table>

For each metric, the best is shown in bold.


Complexity Analysis

Dataset: PASCAL VOC 2007

<table>
<thead>
<tr>
<th>Model</th>
<th>Grad-CAM</th>
<th>Grad-CAM++</th>
<th>IG-CAM ( (m=20) )</th>
<th>IG-CAM ( (m=50) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50</td>
<td>11.3 ms</td>
<td>12.2 ms</td>
<td>54.8 ms</td>
<td>108.08 ms</td>
</tr>
</tbody>
</table>

Average run-time on different models

Insights:

➢ The number of calls in IG-CAM (“m”) does not improve its performance significantly, if increased from 20.

➢ Though IG-CAM runs slower rather than Grad-CAM and Grad-CAM++, the modifications in IG-CAM do not slow this method down considerably.

➢ Though some perturbation-based methods may outperform IG-CAM, the satisfying speed of our method makes it a desired choice for real-world real-time applications.
**Takeaways**

**IG-CAM**

1. Circumvented the underestimations in Grad-CAM and Grad-CAM++.

2. Addressed the issues caused by backpropagation in the methods above.

3. Though slower than the conventional methods, offers an acceptable run-time to be used in real-world applications.

4. The takeaways of IG-CAM are verified through extensive experiments on the PASCAL VOC 2007 dataset.
Thank you. Questions?