Syntax-Guided Program Synthesis for Class Activation Mapping[1]

Alejandro Luque Cerpa, CSE, Chalmers, Sweden
Elizabeth Polgreen, School of Informatics, University of Edinburgh, UK
Hazem Torfah, CSE, Chalmers, Sweden (Supervisor)

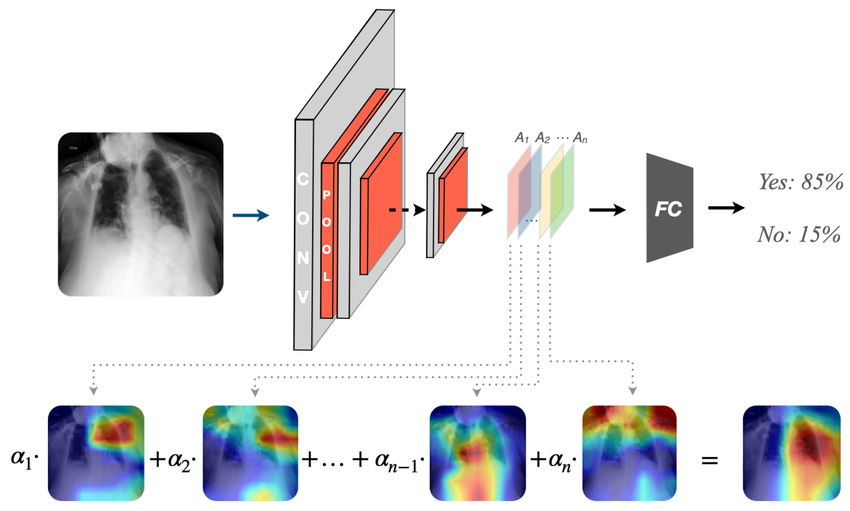
CHALMERS UNIVERSITY OF TECHNOLOGY

THE UNIVERSITY of EDINBURGH

Summary

Class activation mapping (**CAM**) methods explain the behavior of Convolutional Neural Networks (CNNs) using **saliency maps**. Choosing the right CAM method depends on **specific requirements** such as the desired **level of detail** in the visualization, **accuracy**, and **robustness**.
We introduce **SyCAM**, a framework for **synthesizing CAM expressions tailored** to a given **context**. We also **eliminate human biases** by synthesizing the expressions according to given evaluation metrics.

Problem Statement Let $\mathcal{M} = (\mathcal{I} \to \mathcal{R}^{|\mathcal{C}|})$ be a set of CNN-based classifiers defined over a space of images $\mathcal{I}$ and a set of classes $\mathcal{C}$. Given $M \in \mathcal{M}$, a set of images $I \subseteq \mathcal{I}$, a threshold function $\lambda\colon \mathcal{M} \times \mathcal{I} \to \mathcal{R}$, a set of CAM-weight expressions $\mathcal{E}$, and an evaluation function $\mu\colon \mathcal{F}_{\text{CAM}} \times \mathcal{M} \times \mathcal{I} \to \mathcal{R}$, synthesize an expression $e \in \mathcal{E}$ s.t. $\mu(L^c[e], M, I) > \lambda(M, I)$.



Structure of a CAM-based method. The saliency map L^c for class c is defined as:

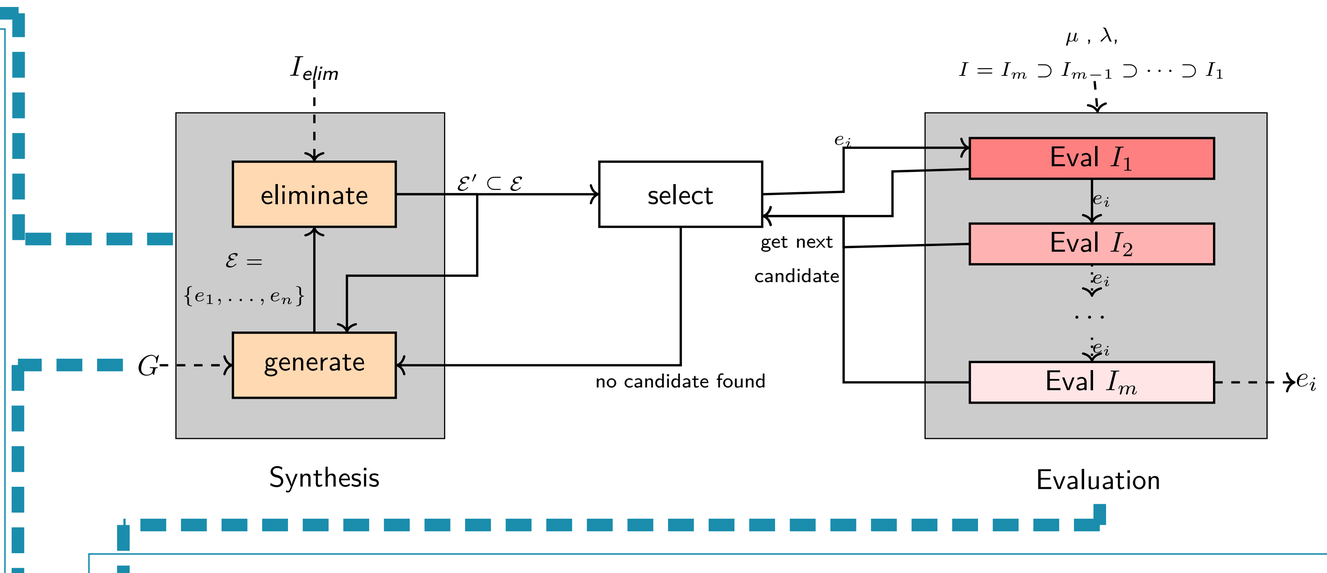
$$L^c = \sum_k \alpha_k^c A_k^l$$

The SyCAM framework

Synthesis

SyCAM uses a Bottom-Up search approach to enumerate expressions using the given grammar. Semantically equivalent expressions found are later discarded.
A naive **enumerative** approach can reject good expressions that perform well over subsets of images.
To prevent this, we use a **class-based decomposition approach** by expanding the grammar, allowing the generation of different expressions for different classes.

$$Expr := (Y_i^c = max(Y_1^c, \ldots Y_n^c))\,?\,Expr\ :\ Expr$$



An Example Grammar

$grads := \dfrac{\partial Y^c}{A_k}$ for any class c and activation map k.

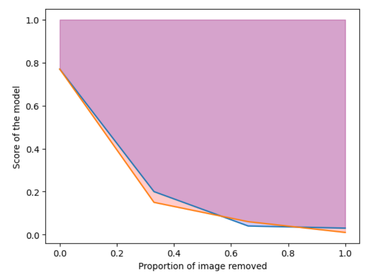
$$term := grads \mid top_5(grads) \mid top_{10}(grads) \mid top_{20}(grads) \mid top_{50}(grads)$$

$$Expr := term \mid 0.5 \cdot Expr + Expr \mid Expr + Expr \mid 2 \cdot Expr + Expr \mid Expr \cdot Expr \mid ReLU(Expr)$$

To generate expressions, SyCAM requires a grammar. We include the gradients inspired by other CAM-based[2] methods.

[2] Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, 2017 IEEE International Conference on Computer Vision (ICCV)

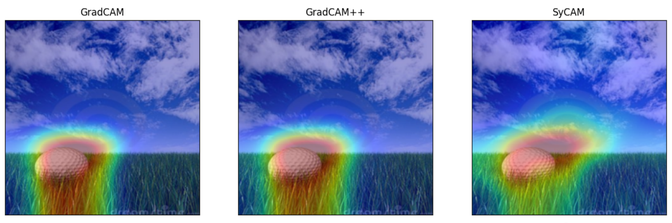
Evaluation



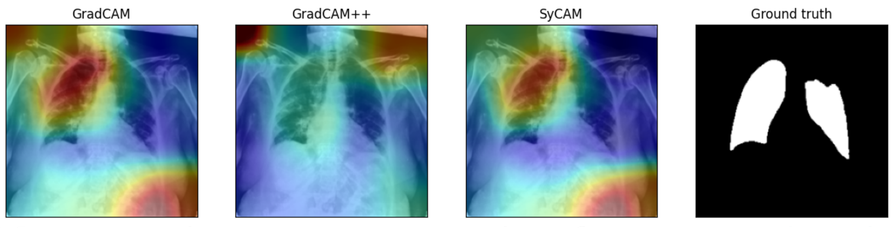
$$Deletion = avg_{\mathcal{I}}\left(\frac{1}{L+1}\sum_{k=0}^{L} f(x^{(0)}) - f(x^{(k)})\right)$$

Example of an evaluation metric. Portions of the image are iteratively removed, the resulting images are classified, and the drop in the classification score is measured. A higher area means that the more important pixels are removed first.

Experiments: Imagenette & COVID-19



SyCAM expressions generate more concise saliency maps. The Deletion metric scores are 0.1364, 0.1366, and 0.1378, respectively.



SyCAM can synthesize expressions that favor expert knowledge, in contrast to standard methods. The scores are 0.021, ∼0.000, and 0.038.

Enumerative vs. Class-based decomposition approach

	Deletion metric			Proportion of images		
Model	GradCAM	GradCAM++	SyCAM	Better	Worse	SyCAM Expression
ResNet50	0.3220	0.3143	**0.3222**	**30.7**	30.2	$2 \cdot Grads + top_{20}$
VGG-16	0.1545	0.1531	**0.1557**	48.3	**48.5**	$Grads + top_{10}$
VGG-19	0.1554	0.1522	Timeout	0	0	$Grads$

Using the enumerative approach, we may obtain timeouts (see VGG-19).

	Deletion metric			Proportion of images		
Class	GradCAM	GradCAM++	SyCAM	Better	Worse	SyCAM Expression
1.Tench	0.2029	0.1962	**0.2066**	**51.9**	45.7	$2 \cdot Grads + top_{20}$
2.English springer	0.2212	0.2167	Timeout	0	0	$Grads$
3.Cassette player	0.0818	0.0795	**0.0819**	47.1	50.4	$2 \cdot Grads + ReLU(Grads)$
4.Chain saw	0.2189	**0.2220**	**0.2220**	47.2	51.0	$ReLU(Grads)$
5.Church	0.1003	0.0876	Timeout	0	0	$Grads$
6.French horn	0.1852	0.1749	Timeout	0	0	$Grads$
7.Garbage truck	0.1072	0.1011	**0.1138**	**56.0**	44.0	$Grads + top_5$
8.Gas Pump	0.1277	0.1199	**0.1278**	**50.6**	49.4	$Grads + top_{20}$
9.Golf ball	0.2042	**0.2066**	**0.2066**	51.1	45.6	$ReLU(Grads)$
10.Parachute	0.1022	**0.1152**	**0.1152**	**26.4**	21.5	$ReLU(Grads)$
Average	0.1552	0.1520	**0.1581**	**36.1**	33.8	

The class-based decomposition approach prevents this problem.

Find our research group in https://starlab.systems
Email: luque@chalmers.se

[1] Syntax-Guided Program Synthesis for Class Activation Mapping
Alejandro Luque-Cerpa, Elizabeth Polgreen, Hazem Torfah
Under review

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM