Streaming Weak Submodularity: Interpreting Neural Networks on the Fly

Elenberg, E., Dimakis, A. G., Feldman, M., & Karbasi, A. (2017).

Presented by: Ronghao Zhang, Wenting Song

Introduction

- Sparse Explanations (feature selection)
- Formulate combinatorial optimization problem
    - weak submodular maximization
- Design streaming algorithms to obtain approximate solutions
    - provable, data dependent performance guarantees

Preliminaries

Interpretability as Subset Selection

- Subset Selection: Optimizing set functions

Given a set $\mathcal{N} = \{1, 2, \ldots, N\}$ and set function $f : 2^{\mathcal{N}} \mapsto \mathbb{R}_{\geq 0}$

$$\underset{\mathrm{S}:|\mathrm{S}| \leq k}{\operatorname{argmax}} f(\mathrm{S})$$

- Interpretability: Select image segments which maximize label's likelihood

$$\max_{|S| \leq k} \text{softmax\_score}(\text{Image}_S)$$

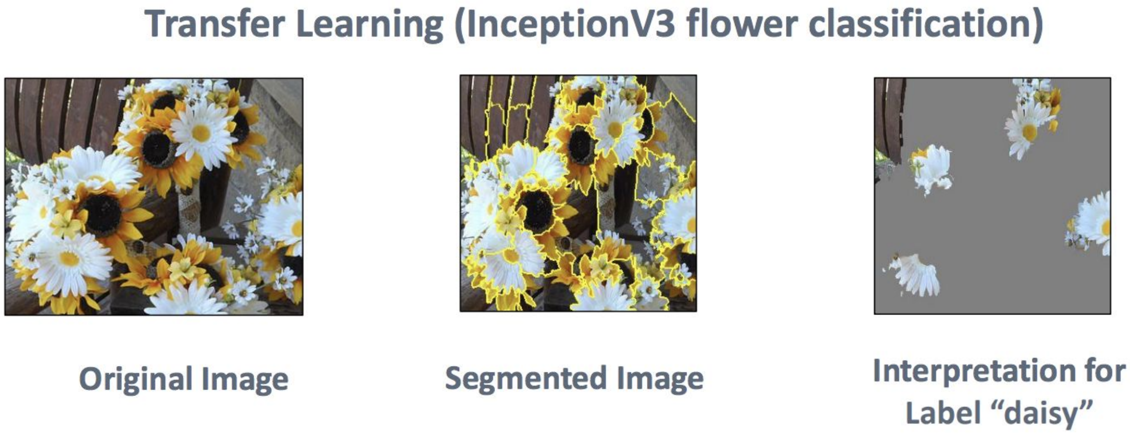
$\sigma_{\texttt{sunflower}}(\{1, 3, 8, 19, 27\}) = 10^{-4}$

$\sigma_{\texttt{sunflower}}(\{25, 28\}) = .49$

$\sigma_{\texttt{sunflower}}(\{21, 25, 27, 28, 30\}) = .79$

Interpretability as Subset Selection

- Find a subset of image segments which contribute the most to its output label.



Weak Submodularity

- Given a set function f and two sets A and B, **Discrete Derivative** is defined as,

  $$f(B \mid A) \triangleq f(A \cup B) - f(A).$$

- Set function is **monotone** if $f(x|A) \geq 0 \quad \forall A, \{x\}$

  Set function is **submodular** if $\forall A, B, \{x\} : A \subseteq B, f(x|A) \geq f(x|B)$

Weak Submodularity

Definition 3.1 (Weak Submodularity, adapted from Das and Kempe [2011]). *A monotone nonnegative set function $f : 2^{\mathcal{N}} \mapsto \mathbb{R}_{\geq 0}$ is called γ-weakly submodular for an integer r if*

$$\gamma \leq \gamma_r \triangleq \min_{\substack{L,S \subseteq \mathcal{N}: \\ |L|,|S \setminus L| \leq r}} \frac{\sum_{j \in S \setminus L} f(j \mid L)}{f(S \mid L)} ,$$

where the ratio is considered to be equal to 1 *when its numerator and denominator are both* 0.

f is submodular if and only if $\gamma_{|\mathcal{N}|} = 1$

To simplify notation, we use γ in place of γ_k in the rest of the paper.

Streaming Algorithms

Streaming Algorithms

- Streaming Optimization
  - one pass over the N items in data
  - Keep or throw away forever
  - Maintain sublinear number of elements in memory, ideally constant.
- Approximation Guarantees

Definition 3.2 (Approximation Ratio). *A streaming maximization algorithm* ALG *which returns a set S has approximation ratio $R \in [0, 1]$ if $\mathbb{E}[f(S)] \geq R \cdot f(OPT)$, where OPT is the optimal solution and the expectation is over the random decisions of the algorithm and the randomness of the input stream order (when it is random).*

Assumption	Algorithm	Approx. Ratio
None	Exhaustive Search	1
Submodular	Greedy	$1 - e^{-1}$
Submodular, Streaming	SIEVE-STREAMING	$\frac{1}{2} - \varepsilon$
γ-Weakly Submodular	Greedy	$1 - e^{-\gamma}$
γ-WS, Streaming	???	???

THRESHOLD GREEDY

Algorithm 1 THRESHOLD GREEDY(f, k, τ)

Let $S \leftarrow \varnothing$.
while there are more elements **do**
  Let u be the next element.
  if $|S| < k$ and $f(u \mid S) \geq \tau / k$ **then**
    Update $S \leftarrow S \cup \{u\}$.
  end if
end while
return: S

Input: Set function f, sparsity parameter k, threshold τ, in range $[0, a\gamma \cdot f(\breve{OPT})]$

compute the **discrete derivative** of adding u to S.

If it exceeds the threshold, add u to S.

- The expected value of the set produced by THRESHOLD GREEDY is at least $\tau \cdot (\sqrt{2 - e^{-\gamma/2}} - 1)$
- Independent of k and N(number of elements in streams)
- Good approximation ratio if $\tau \approx a(\gamma) \cdot f(OPT)$

STREAK

Compute running maximum singleton $f(u_m) = m$

Algorithm 2 $\textsc{Streak}(f, k, \varepsilon)$

Let $m \leftarrow 0$, and let I be an (originally empty) collection of instances of Algorithm 1.
while there are more elements **do**
  Let u be the next element.
  if $f(u) \geq m$ **then**
    Update $m \leftarrow f(u)$ and $u_m \leftarrow u$.
  end if
  Update I so that it contains an instance of Algorithm 1 with $\tau = x$ for every $x \in \{(1-\varepsilon)^i \mid i \in \mathbb{Z}$ and $(1-\varepsilon)m/(9k^2) \leq (1-\varepsilon)^i \leq mk\}$, as explained in Section 5.2.
  Pass u to all instances of Algorithm 1 in I.
end while
return: the best set among all the outputs of the instances of Algorithm 1 in I and the singleton set $\{u_m\}$.

Run and update $\mathcal{O}(\varepsilon^{-1} \log k)$ instances of **ThresholdGreedy**, with exponentially spaced thresholds

$$\tau \in \{(1-\varepsilon)^i \mid i \in \mathbb{Z} \text{ and } (1-\varepsilon)m/(9k^2) \leq (1-\varepsilon)^i \leq mk\}$$

Return the output of best instance or the best singleton

$$\max\{S_{I^*}, u_m\}$$

accuracy parameter $\varepsilon \in (0, 1)$.

Streaming Thresholding Algorithms

Algorithm	ThresholdGreedy	Streak
Approximation Ratio	$\tau \cdot (\sqrt{2 - e^{-\gamma/2}} - 1)$	$(1-\varepsilon)\gamma \cdot \frac{3 - e^{-\gamma/2} - 2\sqrt{2 - e^{-\gamma/2}}}{2}$
Memory	$\mathcal{O}(k)$	$\mathcal{O}(\varepsilon^{-1} k \log k)$
Running Time	$\mathcal{O}(Nf)$	$\mathcal{O}(Nf\varepsilon^{-1} \log k)$

Experiment 1:
Sparse Regression with Pairwise Features

RandomSubset vs. STREAK vs. LocalSearch

RandomSubset vs. STREAK vs. LocalSearch

Preperations:

- 2000 training and 2000 test observations from the Phishing dataset (UC Irvine ML Repository)
- This setup is known to be weakly submodular under mild data assumption

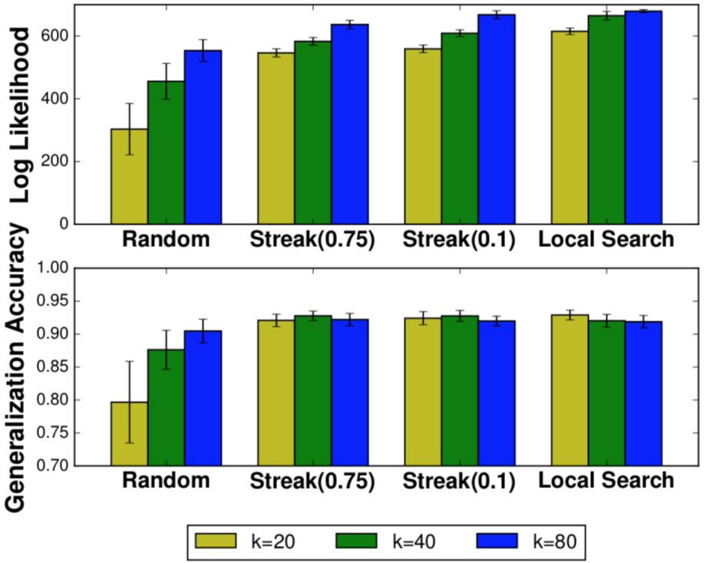
Steps:

- Categorical features are one-hot encoded, increasing the feature dimension to 68
- All pairwise products are added for a total of N = 4692 features. (68 choose 2 + 68) * 2 = (2278+68)*2=4692
- Feature products are generated and added to the stream on-the-fly as needed.

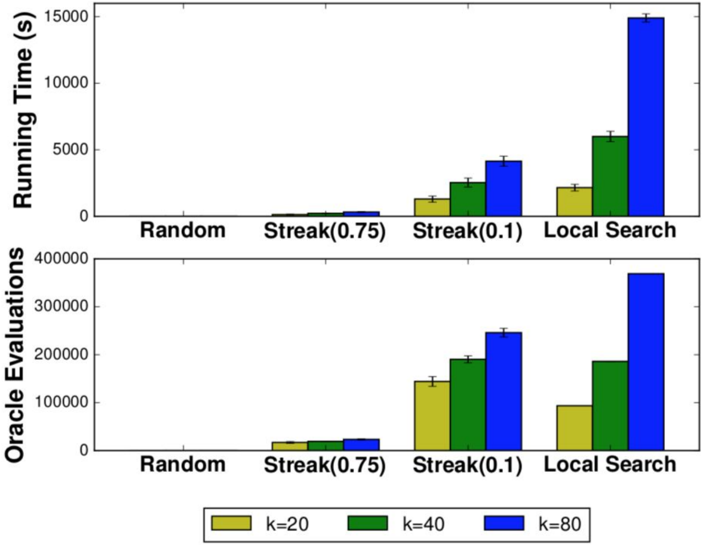
RANDOMSUBSET: selects the first k features from the random stream.

LOCALSEARCH: first fills a buffer with the first k features, and then swaps each incoming feature with the feature from the buffer which yields the largest nonnegative improvement.

Performance and Cost Analysis



(a) Performance

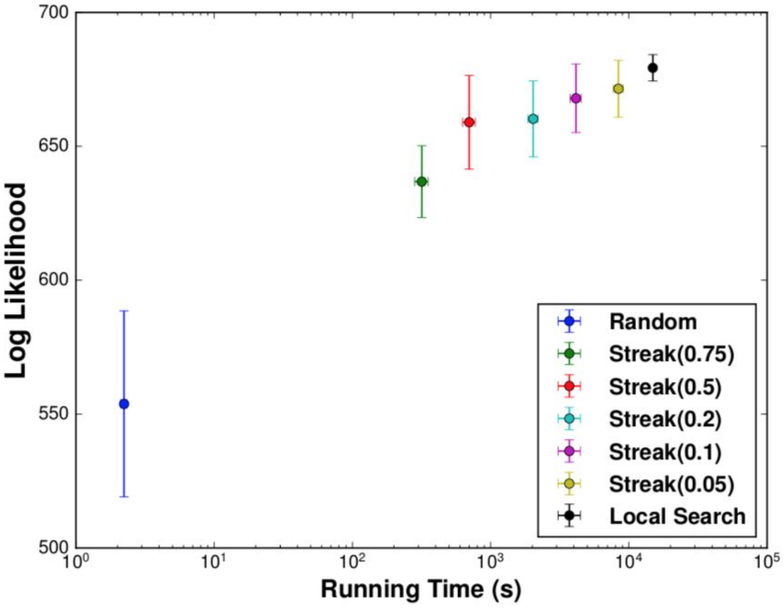


(b) Cost

Figure (a) shows both the final log likelihood and the generalization accuracy for RANDOMSUBSET, LOCALSEARCH, and our STREAK algorithm for accuracy parameter = {0.75, 0.1} and number of features = {20, 40, 80}.

Figure (b) shows two measures of computational cost: running time and the number of oracle evaluations (regression fits).

Log Likelihood vs. Running Time



(a) Sparse Regression

k = 80 and precision in range [0.05, 0.75]

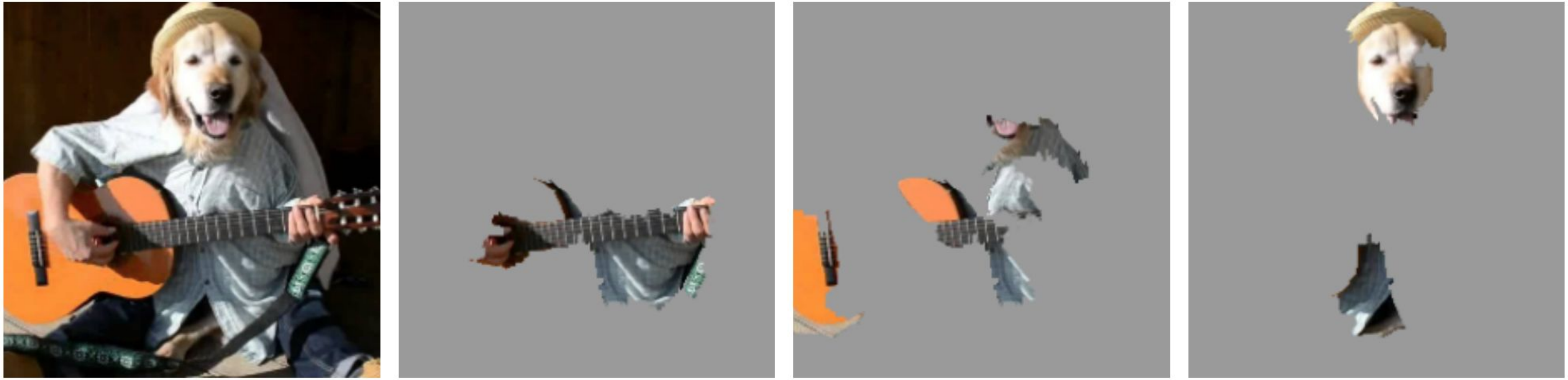
By varying the precision, we achieve a gradual tradeoff between speed and performance.

This shows that STREAK can reduce the running time by over an order of magnitude with minimal impact on the final log likelihood.

Experiment 2:
Black-Box Interpretability

LIME vs STREAK

Black Box Interpretability Explained



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

Experiment Background

Objective and Preparation:

- Interpreting the predictions of black-box machine learning models
- Inception V3 deep neural network trained on ImageNet
- Classifying 5 types of flowers via transfer learning

Procedure to Interpret the Model:

- This is done by adding a final softmax layer and retraining the network
- Find the set of superpixels that contribute the most to the final classification

LIME: perturb images by hiding superpixels (hide guitar, model think it’s labrador)

STREAK: greedy maximization algorithm (Stream)

LIME Type Application

Linear Regression

SLIC image segmentation algorithm: features (superpixels)

Image → Features → Feature Selection

Feature selection methods are supplied by:

Perturb Dataset

1. Highest Weights: fits a full regression and keep k features w/ largest coefficients.

2. Forward Selection: standard greedy forward selection.

3. Lasso: Least Absolute Shrinkage and Selection Operator

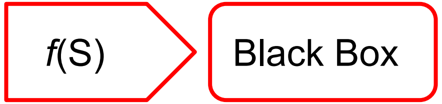
Perturb and fit a k-sparse linear regression in the space of interpretable features.

STREAK Type Application

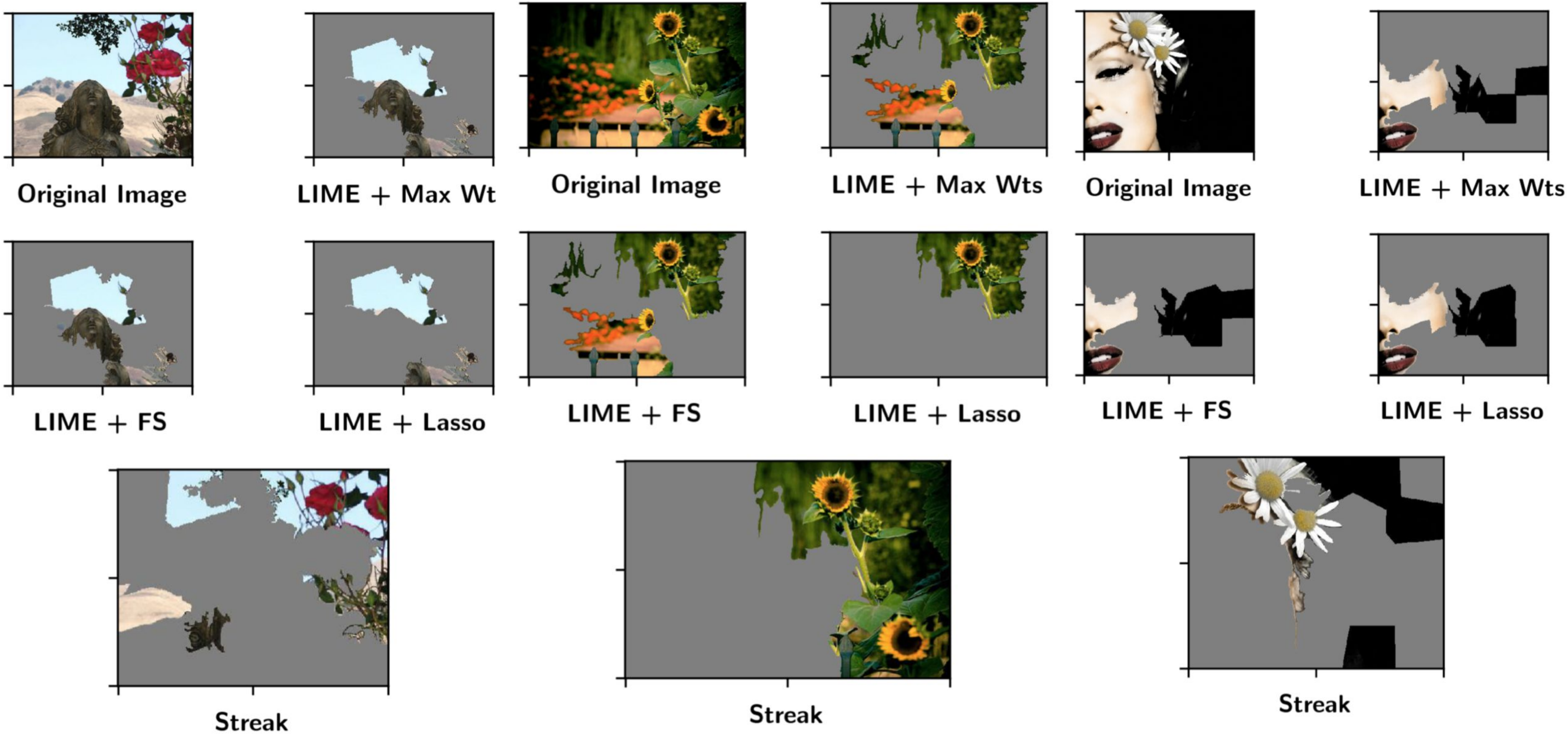
We introduce a novel method for black-box interpretability that is similar to but simpler than LIME. As before, we segment an image into N superpixels. Then, for a subset S of those regions we can create a new image that contains only these regions and feed this into the black-box classifier. For a given model M, an input image I, and a label $\mathbf{L}_1$ we ask for an explanation: why did model M label image I with label $\mathbf{L}_1$. We propose the following solution to this problem. Consider the set function $f(S)$ giving the likelihood that image $I(S)$ has label $\mathbf{L}_1$. We approximately solve

$$\boxed{\max_{|S| \leq k} f(S) \ ,}$$

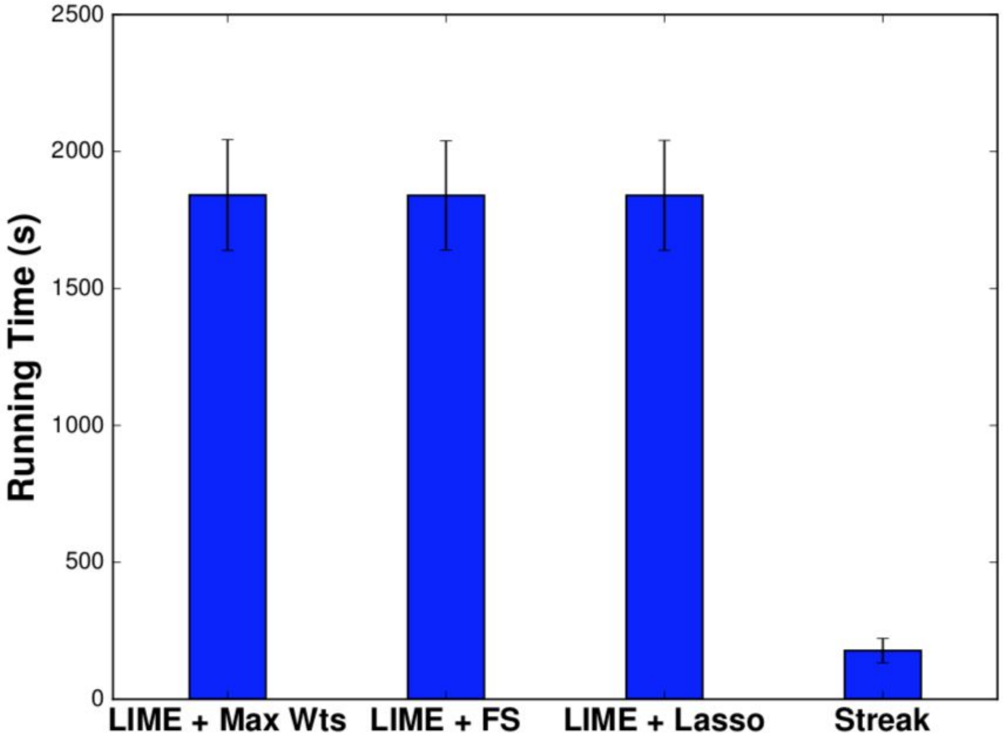
using STREAK. Intuitively, we are limiting the number of superpixels to k so that the output will include only the most important superpixels, and thus, will represent an interpretable explanation. In our experiments we set $k = 5$.



Result Analysis



Runtime Analysis

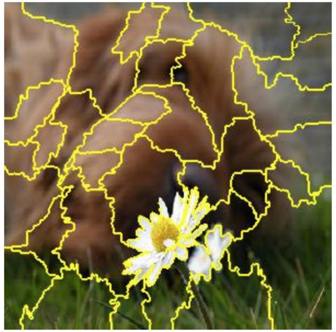


Running times of interpretability algorithms on the Inception V3 network, N = 30, k = 5. Streaming maximization runs 10 times faster than the LIME framework. Results averaged over 40 total iterations using 8 example explanations, error bars show 1 standard deviation.

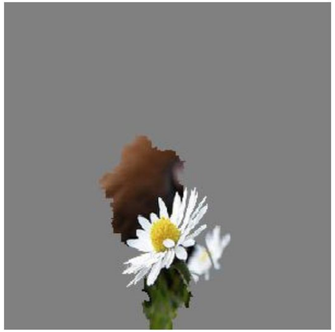
Thank you.

STREAK Application Example

Original Image (top label: daisy)



Segmented Image



Interpretation for daisy

Original Image (top label: daisy)

Segmented Image

Interpretation for daisy

Given a black-box neural network and a test image, the algorithm finds a sparse explanation for the network's prediction.

1. Segment the image into regions
2. Rerun the network with most of the image regions replaced by a gray reference image, record the output
3. The algorithm returns a sparse set of regions that collectively still activate the network's top label