ChannelExplorer: Visual Analytics at Activation Channel's Granularity

Rahat Zaman
rahatzamancse@sci.utah.edu

Bei Wang
beiwang@sci.utah.edu

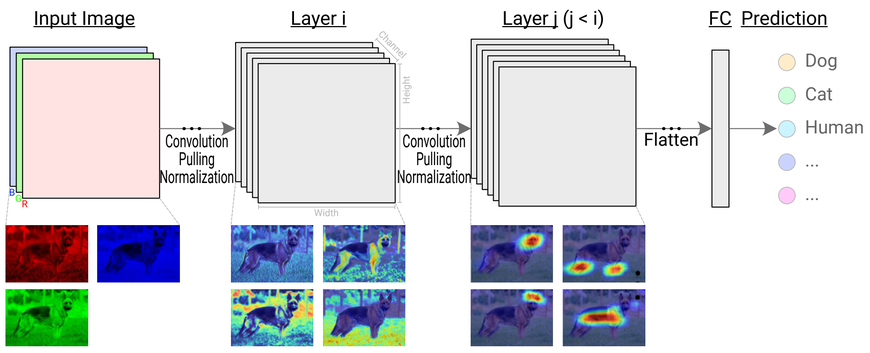
Paul Rosen
prosen@sci.utah.edu

Scientific Computing and Imaging Institute (SCI)
University of Utah

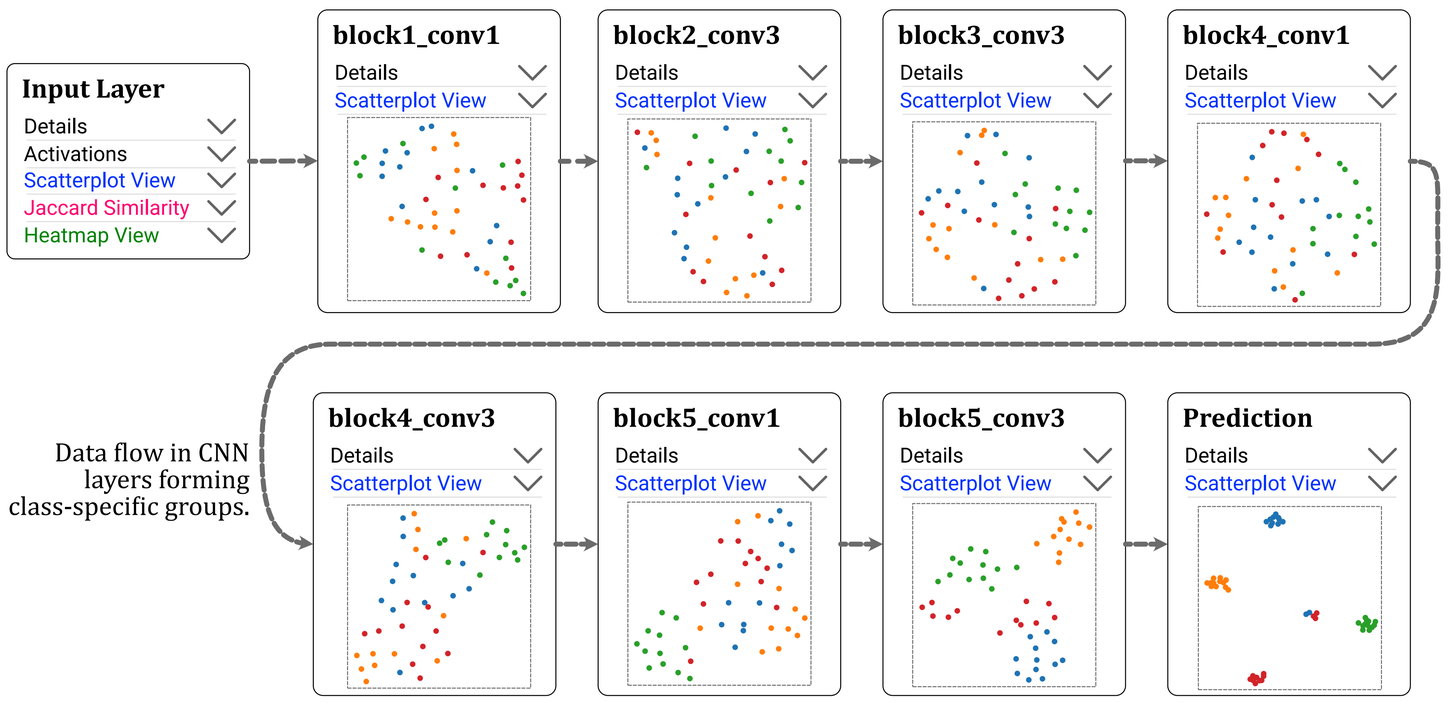
Introduction and Motivation

Deep Learning models are hard to debug and interpret, especially with the lack of generalized tools.

This poster introduces ChannelExplorer, a visualization tool that analyzes convolutional deep learning models. Each CNN layer output is divided into slices called channels that is used to produce the visualizations.



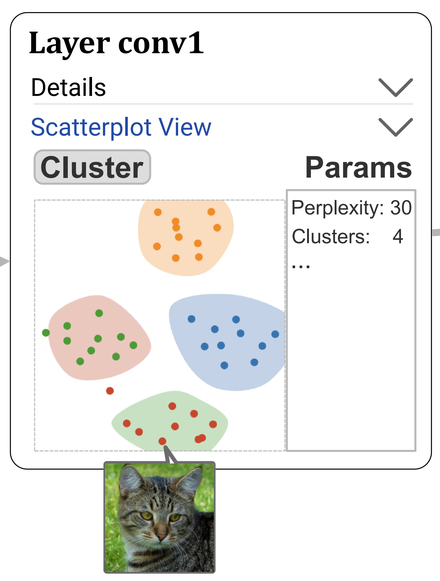
Earlier layers detect basic features; later layers detect complex features.



Data flow in CNN layers forming class-specific groups.

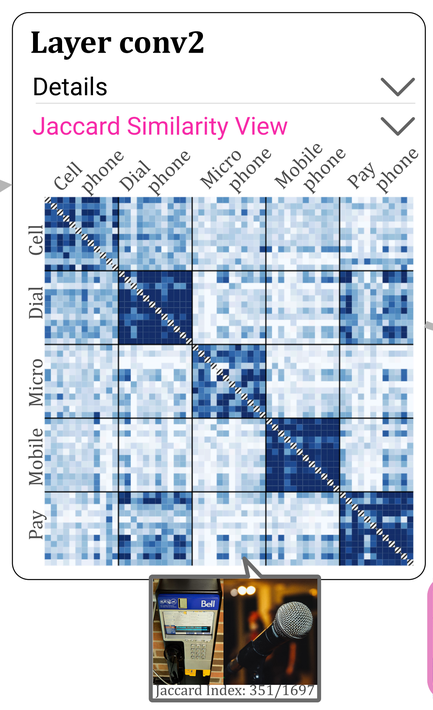
Methodology

ScatterPlot View



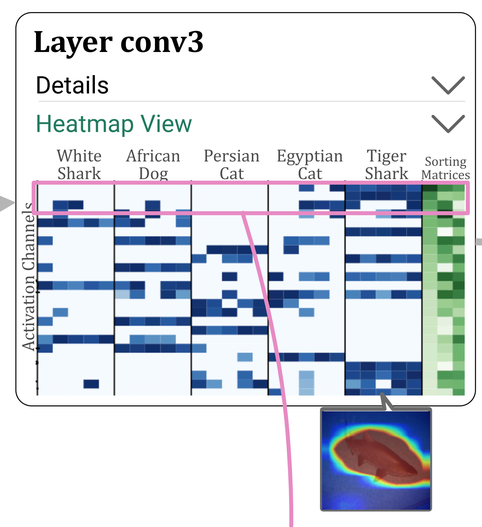
Inputs are projected using **Dimension Reduction** methods, showing different patterns in activation space for different types of inputs.

Jaccard Similarity View



Similarity between activation pattern for multiple inputs are visualized with Jaccard channel similarity.

Heatmap View

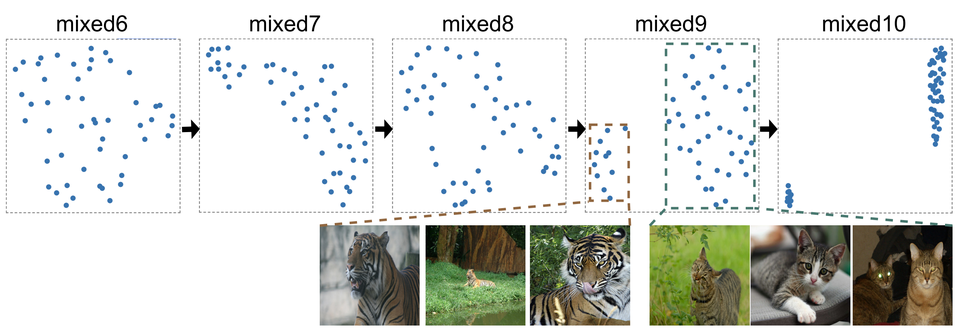


Individual channels for multiple inputs are sorted and visualized using heatmap. Bright stripes indicate filters targeting specific features.

Findings

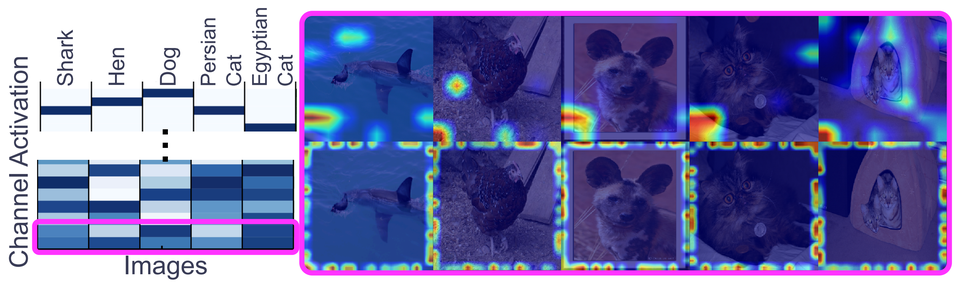
Mislabel Identification

Scatterplot view of ChannelExplorer shows that a classification dataset have mislabeled inputs. In ImageNet, cats are classified as Tiger Cat.



Unimportant Channels

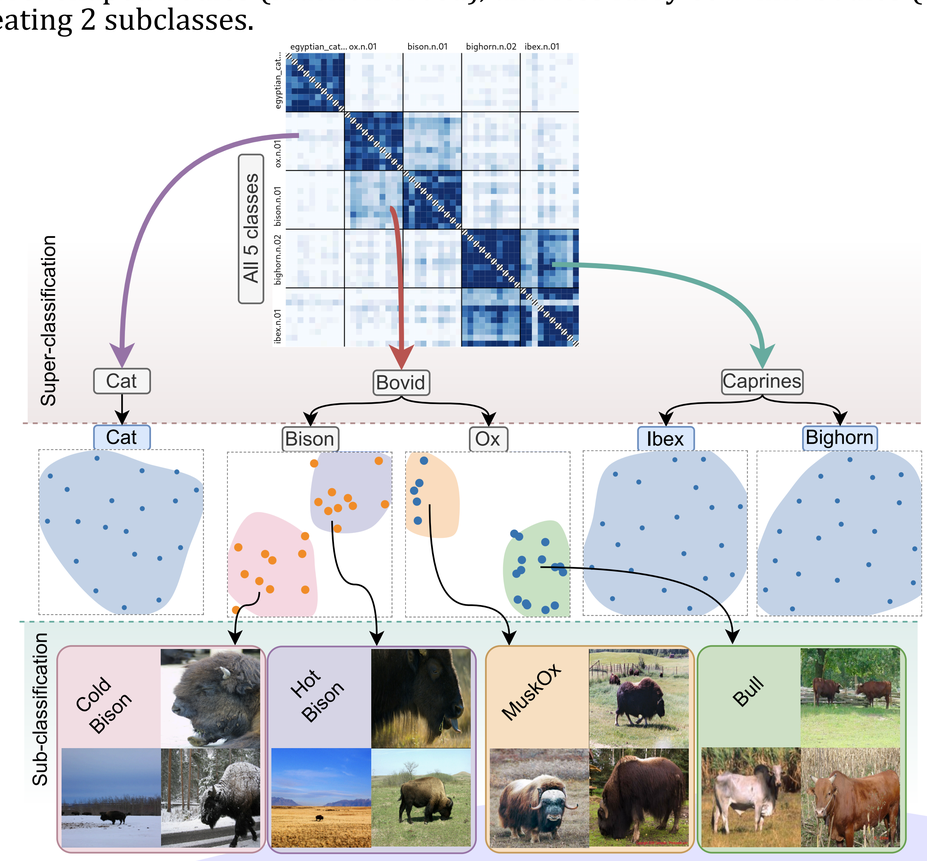
Channels activating for unimportant features - attending to borders of the image.



Use cases on InceptionV3 & ImageNet

Hierarchical Classification

Starting from 5 classes, 3 super-classes (Cat, Bovid, and Caprines) are created. In the Bison's Scatter View, two clusters show examples of Bison in Cold & Hot weather. Similarly, Oxen's Scatterplot View shows furry and barrel-shaped bodies (MuskOx breed), and less furry cow-like bodies (Bull); creating 2 subclasses.



Built with: Tensorflow, PyTorch, Scikit Learn, Sphinx, Sphinx, BearType, React, Typescript, React-Flow, Redux, FastAPI

* This work is partially supported by **NSF DMS-2134223**