# Advanced Data Visualization

CS 6965

Fall 2019

Prof. Bei Wang Phillips

University of Utah

Lecture 16

# Deep Learning and Vis
# Part II

HD

# Exploring Neural Networks with Activation Atlases

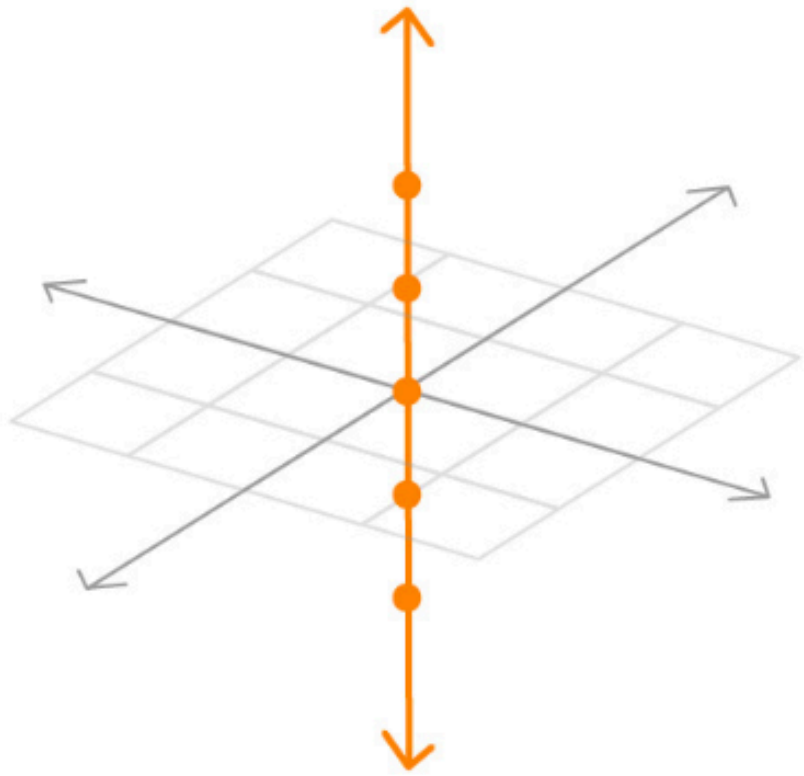
https://distill.pub/2019/activation-atlas/

# Activation Atlas

- Use feature inversion to visualize millions of activations from an image classification network
- An explorable activation atlas of features the network has learned which can reveal how the network typically represents some concepts

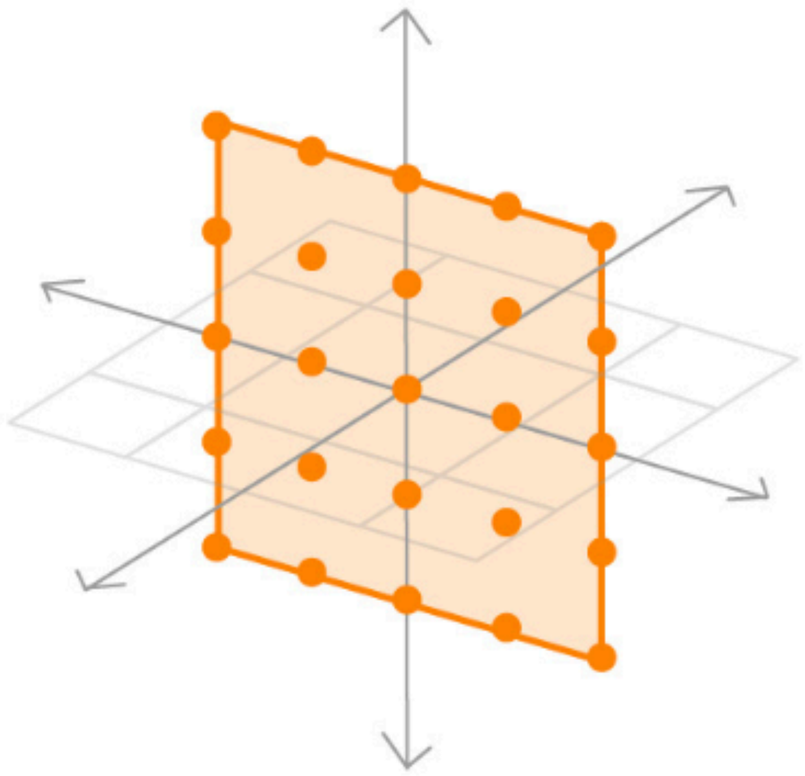

https://distill.pub/2019/activation-atlas/

## INDIVIDUAL NEURONS



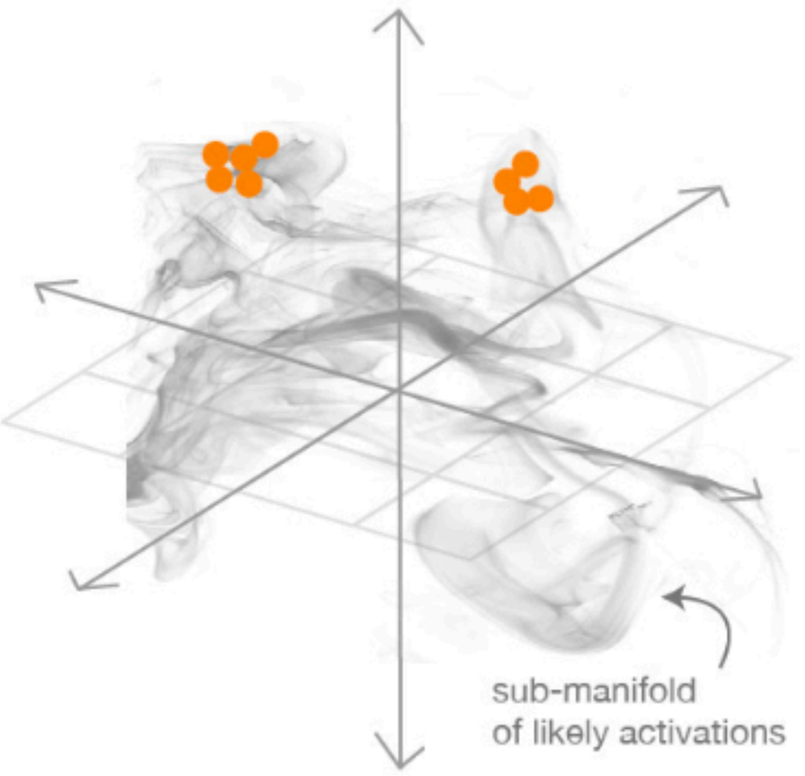
[Visualizing individual neurons] make hidden layers somewhat meaningful, but misses interactions between neurons — it only shows us one-dimensional, orthogonal probes of the high-dimensional activation space.

## PAIRWISE INTERACTIONS



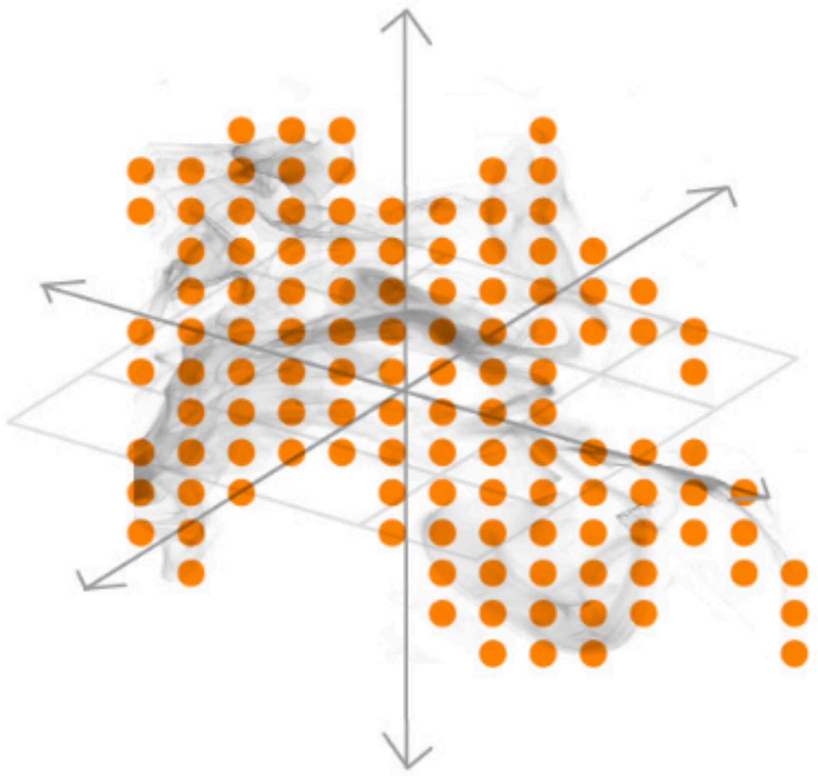
[Pairwise interactions] reveal interaction effects, but they only show two-dimensional slices of a space that has hundreds of dimensions, and many of the combinations are not realistic.

## SPATIAL ACTIVATIONS




[Spatial activations] show us important combinations of many neurons by sampling the sub-manifold of likely activations, but they are limited to those that occur in the given example image.

## ACTIVATION ATLAS



Activation atlases give us a bigger picture overview by sampling more of the manifold of likely activations.

| | | |
|---|---|---|
| **1.** | **grey whale** | **91.0%** |
| 2. | killer whale | 7.5% |
| 3. | great white shark | 0.7% |
| 4. | gar | 0.4% |

| | | |
|---|---|---|
| **1.** | **great white shark** | **66.7%** |
| 2. | baseball | 7.4% |
| 3. | grey whale | 4.1% |
| 4. | sombrero | 3.2% |

# InceptionV1: a convolutional network


InceptionV1 builds up its understanding of images over several layers (see overview from [2]). It was trained on ImageNet ILSVRC [11]. Each layer actually has several component parts, but for this article we'll focus on these larger groups.

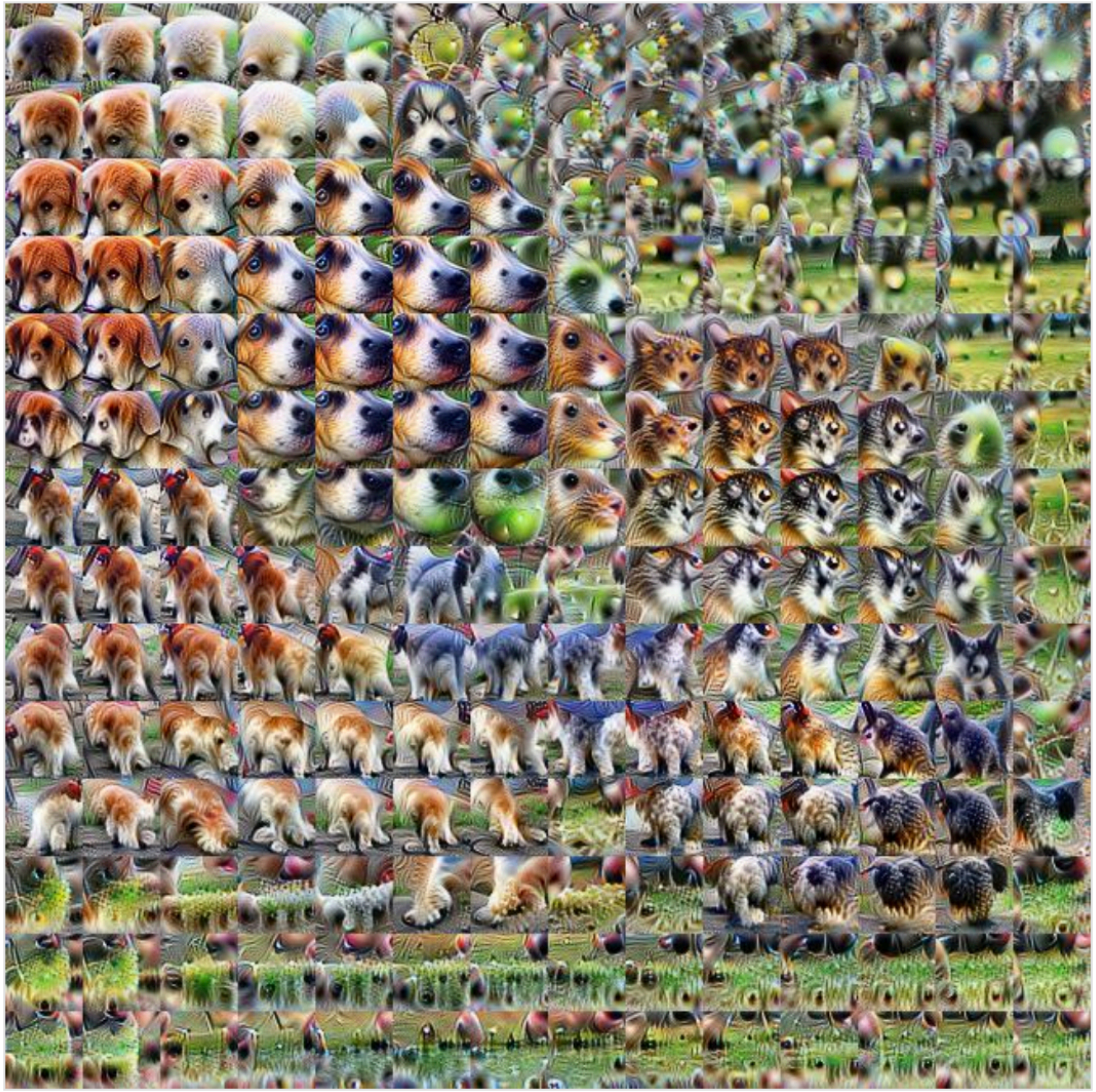
# Visualize how the network sees an image

- Feed the image into the network and run it through to the layer of interest.
- Collect the activations — the numerical values of how much each neuron fired. Positive activation value if a neuron is excited by the input.
- Use feature visualization that transform vectors of activation values to an idealized image of what the network thinks and sees.
- Starting with an activation vector at a particular layer, we create an image through an iterative optimization process.


https://distill.pub/2017/feature-visualization/appendix/

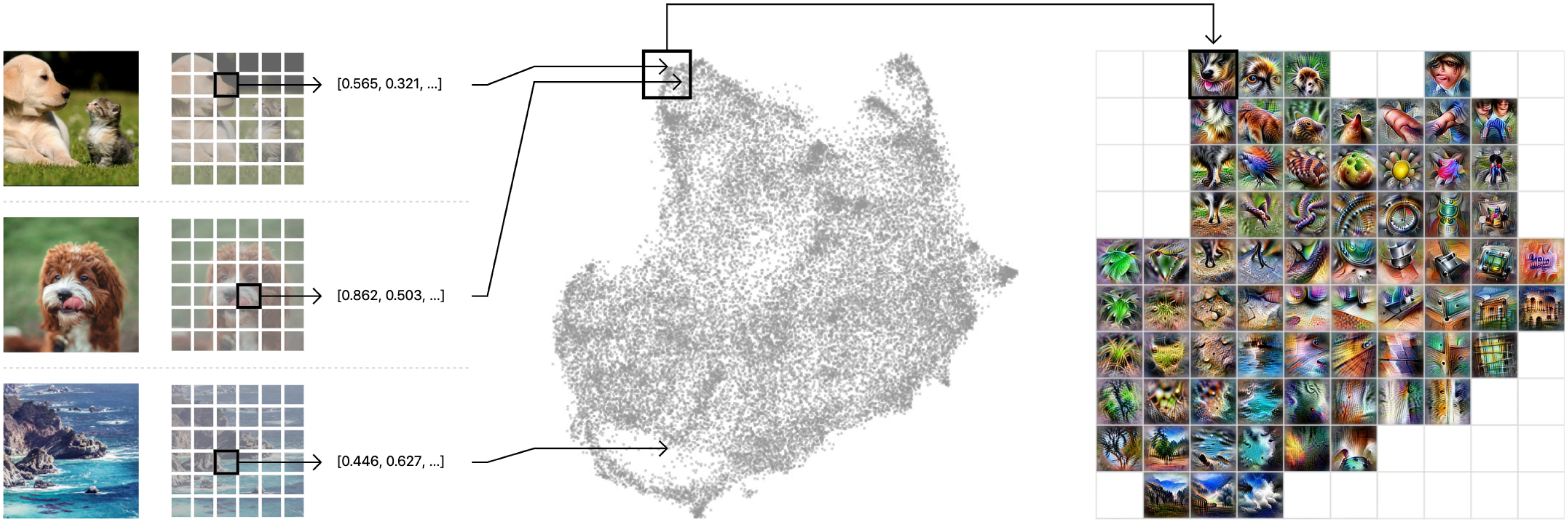
# How the network sees different parts of an image

Input image from ImageNet.



Activation grid from InceptionV1, layer mixed4d.

# Aggregation multiple images

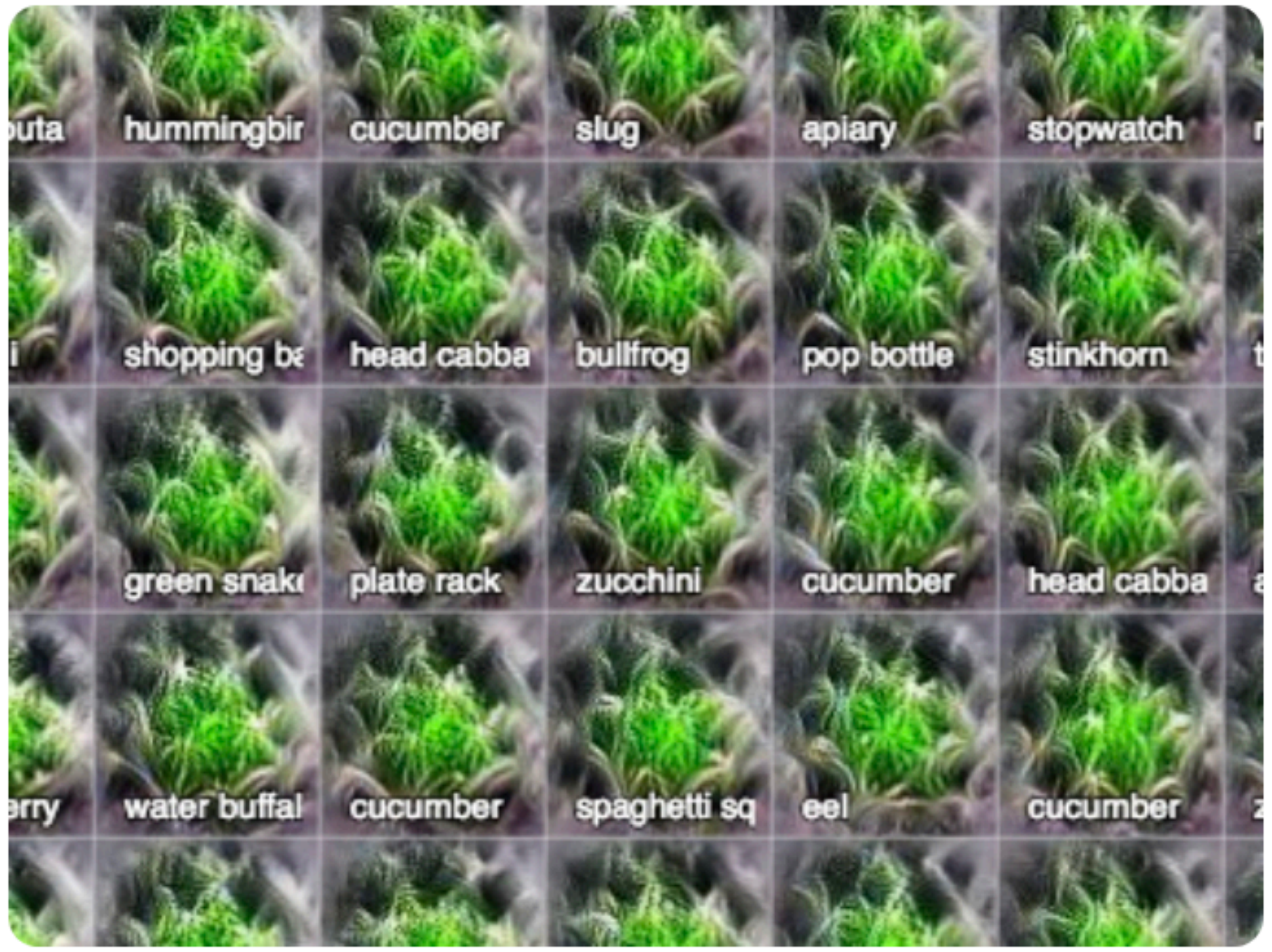



A randomized set of one million images is fed through the network, collecting one random spatial activation per image.

The activations are fed through UMAP to reduce them to two dimensions. They are then plotted, with similar activations placed near each other.

We then draw a grid and average the activations that fall within a cell and run feature inversion on the averaged activation. We also optionally size the grid cells according to the density of the number of activations that are averaged within.

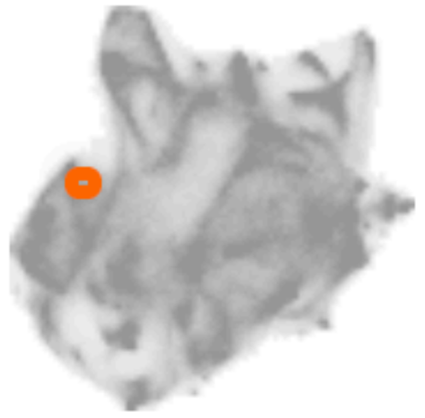
## MIXED3B




You'll immediately notice that the early layer is very nonspecific in comparison to the others. The icons that emerge are of patterns and splotches of color. It is suggestive of the final class, but not particularly evocative.

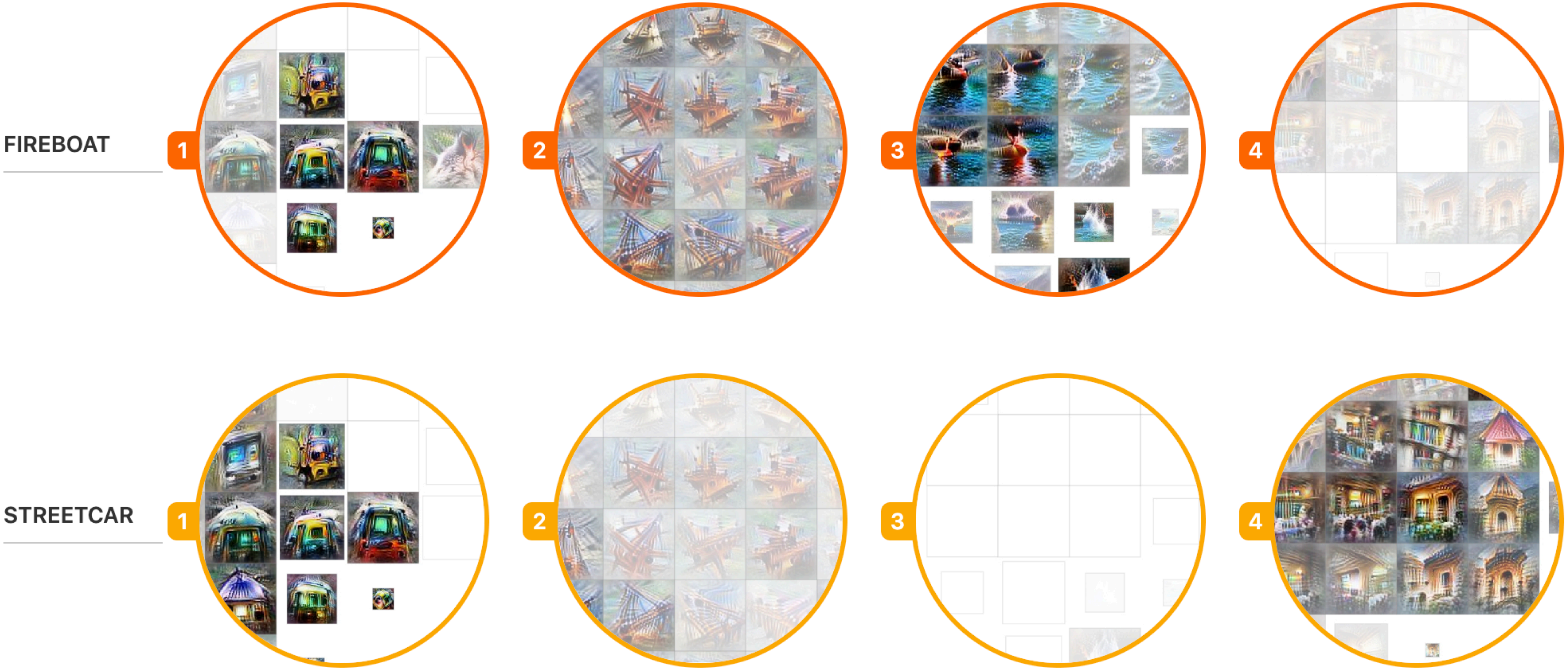
## MIXED4C


By the middle layer, icons definitely resemble leaves, but they could be any type of plant. Attributions are focused on plants, but are a little all over the board.



## MIXED5B


Here we see foliage with textures that are specific to cabbage, and curved into rounded balls. There are full heads of cabbage rather than individual leaves.




At mixed4d, Both "streetcar" and "fireboat" contain activations for what look like windows.

Both classes contain activations for crane-like apparatuses, though they are less prominent than the window activations.

"Fireboat" activations have much stronger attributions from water than "streetcar", where there is virtually no positive evidence.

The activations for "streetcar" have much stronger attributions from buildings than does "fireboat".

# SUMMIT: Scaling Deep Learning Interpretability by Visualizing Activation and Attribution Summarizations

https://arxiv.org/abs/1904.02323

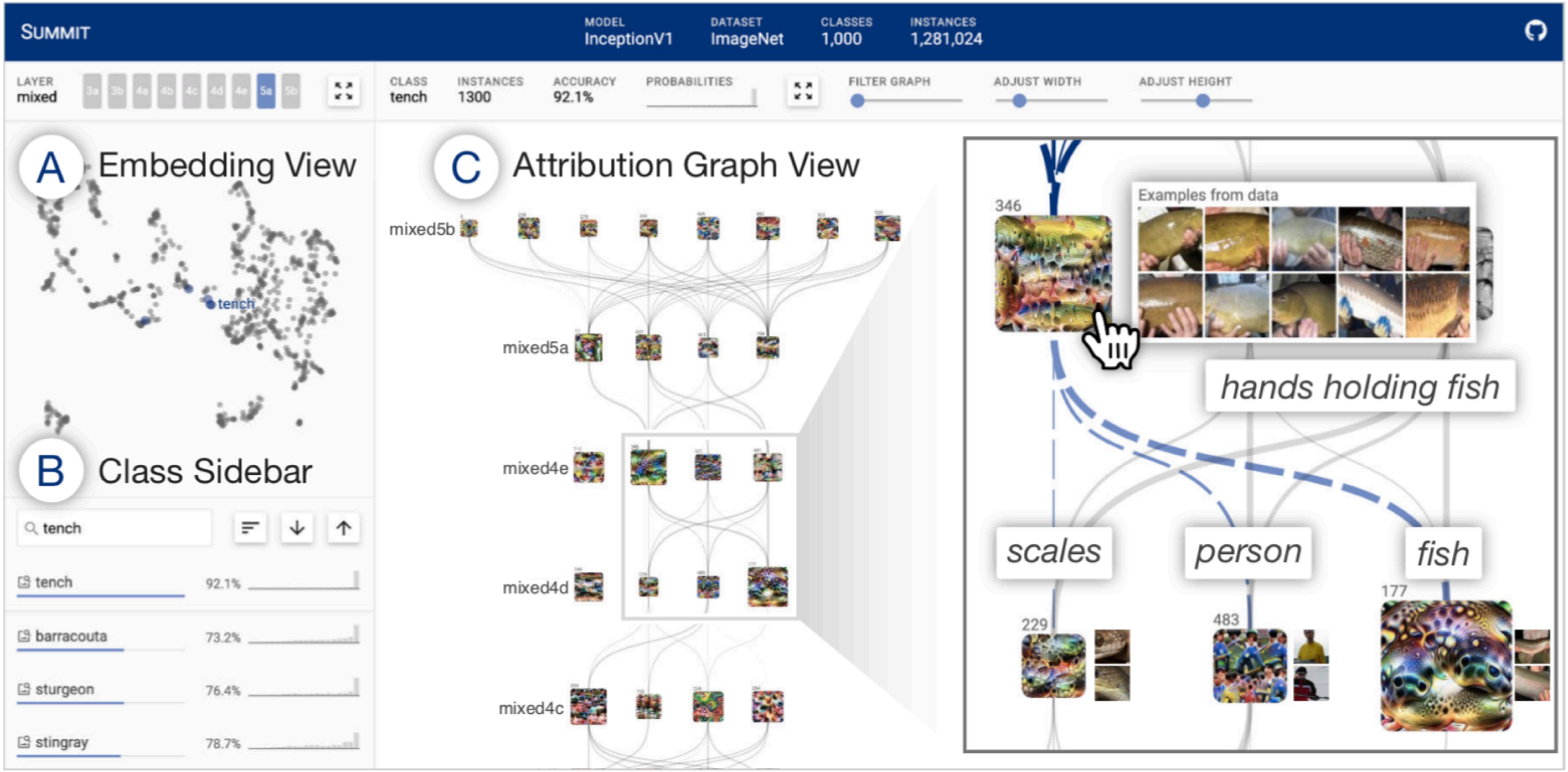



Fig. 1. With Summit, users can scalably summarize and interactively interpret deep neural networks by visualizing *what* features a network detects and *how* they are related. In this example, INCEPTIONV1 accurately classifies images of ***tench*** (yellow-brown fish). However, SUMMIT reveals surprising associations in the network (e.g., using parts of people) that contribute to its final outcome: the "tench" prediction is dependent on an intermediate "hands holding fish" feature (right callout), which is influenced by lower-level features like *"scales," "person,"* and *"fish"*. **(A) Embedding View** summarizes all classes' aggregated activations using dimensionality reduction. **(B) Class Sidebar** enables users to search, sort, and compare all classes within a model. **(C) Attribution Graph View** visualizes highly activated neurons as vertices ("scales," "fish") and their most influential connections as edges (dashed purple edges).

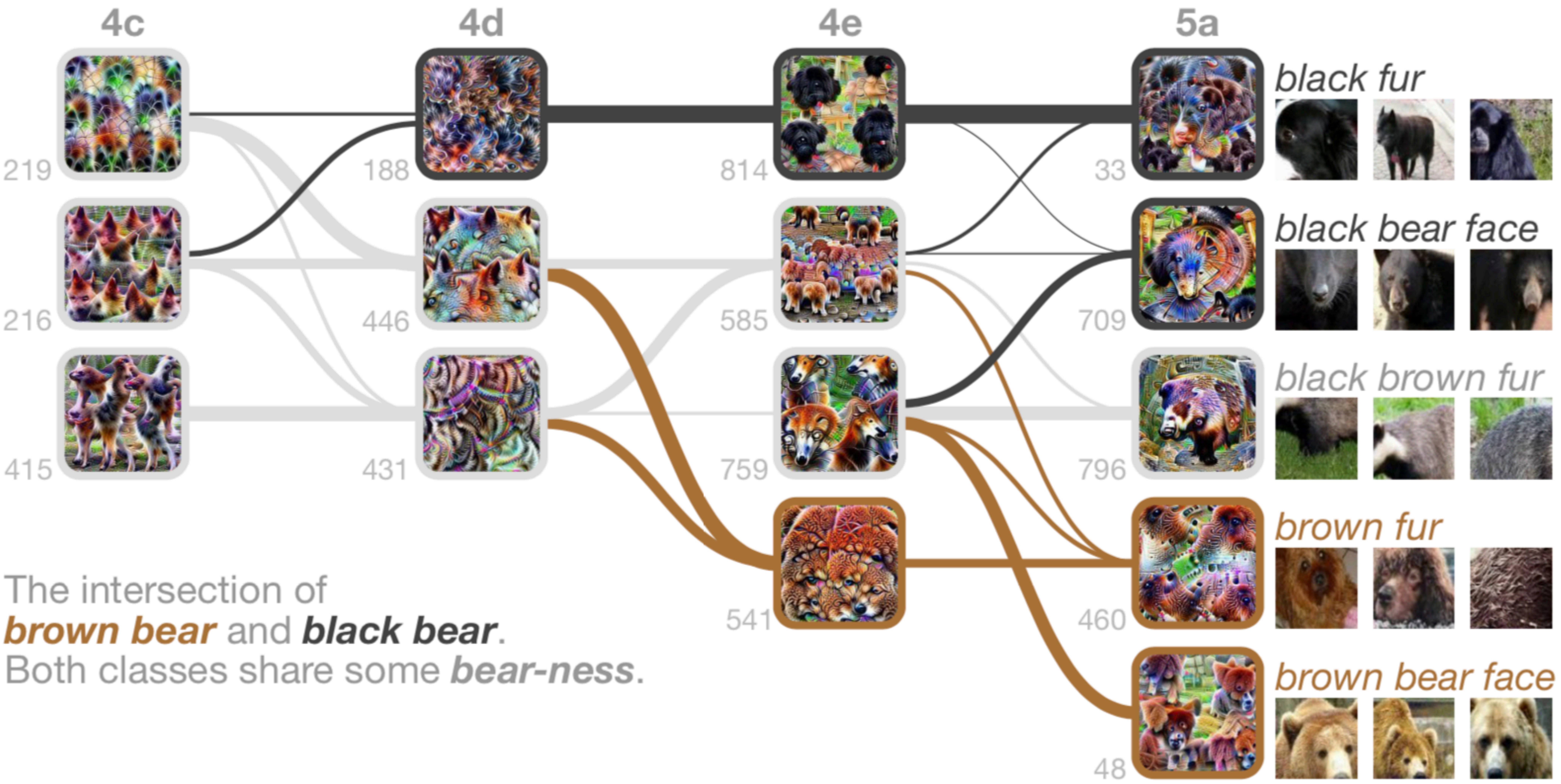
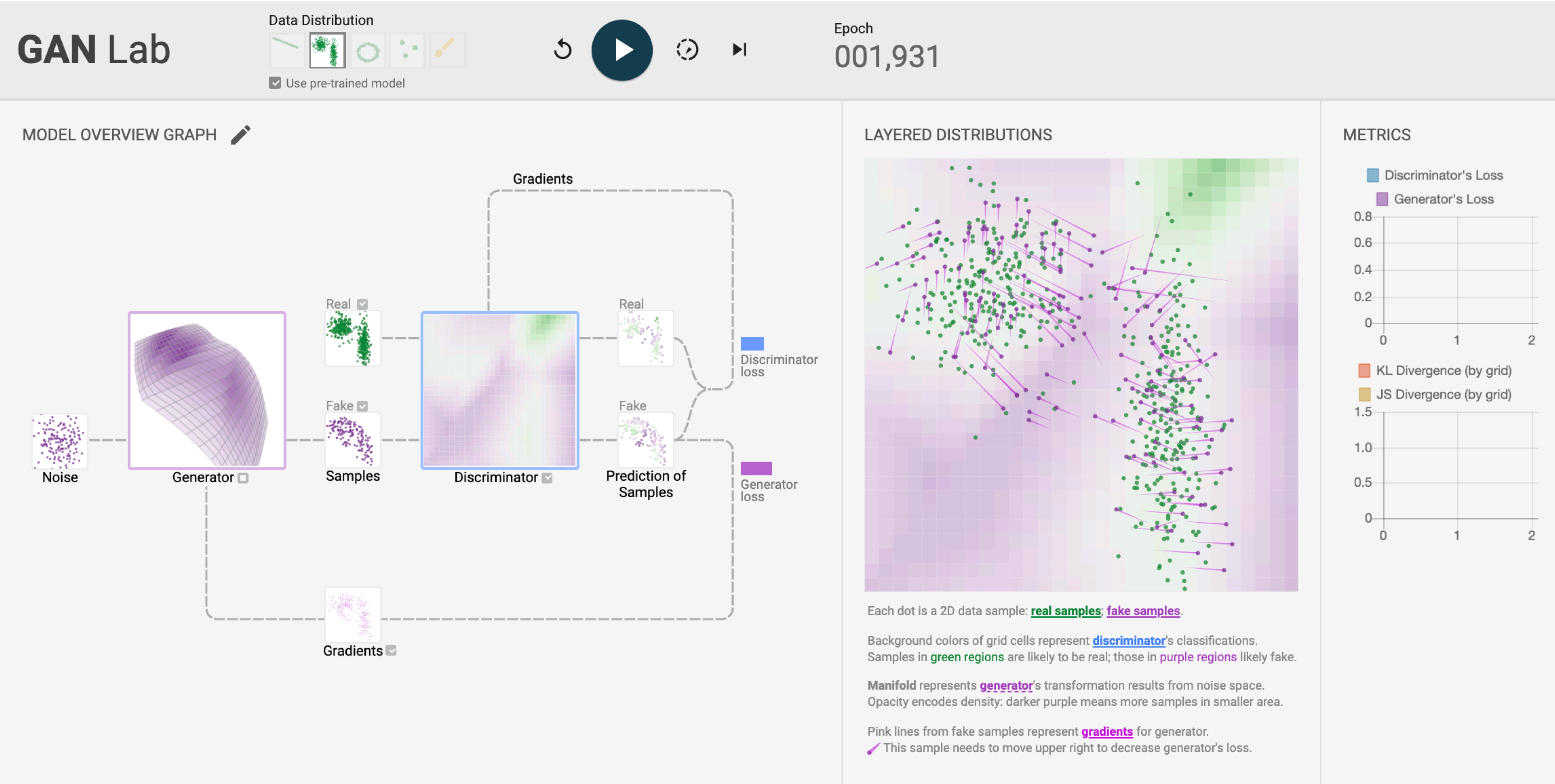



Fig. 9. With attribution graphs, we can compare classes throughout layers of a network. Here we compare two similar classes: ***black bear*** and ***brown bear***. From the intersection of their attribution graphs, we see both classes share features related to ***bear-ness***, but diverge towards the end of the network using fur color and face color as discriminable features. This feature discrimination aligns with how humans might classify bears.

# GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation

https://arxiv.org/abs/1809.01587




https://poloclub.github.io/ganlab/

https://www.youtube.com/watch?v=eTq9T_sPTYQ

# Visual Analytics in Deep Learning Survey

https://arxiv.org/abs/1801.06889

**§ 4** **Why do we want to visualize deep learning?**
Why and for what purpose would one want to use visualization in deep learning?

**§ 5** **Who wants to visualize deep learning?**
Who are the types of people and users that would use and stand to benefit from visualizing deep learning?

**§ 6** **What can we visualize in deep learning?**
What data, features, and relationships are inherent to deep learning that can be visualized?

**§ 7** **How can we visualize deep learning?**
How can we visualize the aforementioned data, features, and relationships?

**§ 8** **When can we visualize deep learning?**
When in the deep learning process is visualization used and best suited?

**§ 9** **Where is deep learning visualization being used?**
Where has deep learning visualization been used?

# Visual Analytics in Deep Learning | Interrogative Survey Overview

## §4 WHY

***Why would one want to use visualization in deep learning?***

Interpretability & Explainability
Debugging & Improving Models
Comparing & Selecting Models
Teaching Deep Learning Concepts

## §6 WHAT

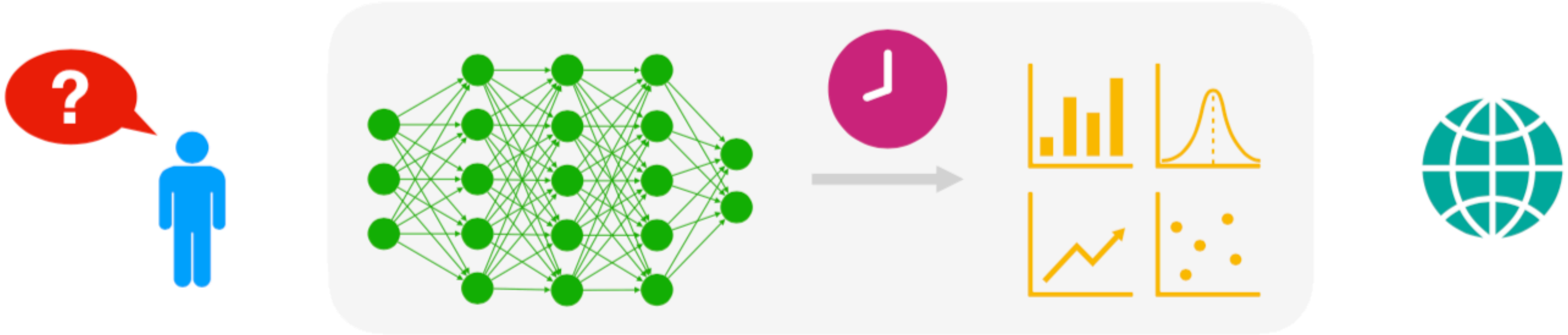
***What data, features, and relationships in deep learning can be visualized?***

Computational Graph & Network Architecture
Learned Model Parameters
Individual Computational Units
Neurons In High-dimensional Space
Aggregated Information

## §8 WHEN

***When in the deep learning process is visualization used?***

During Training
After Training




## §5 WHO

***Who would use and benefit from visualizing deep learning?***

Model Developers & Builders
Model Users
Non-experts

## §7 HOW

***How can we visualize deep learning data, features, and relationships?***

Node-link Diagrams for Network Architecture
Dimensionality Reduction & Scatter Plots
Line Charts for Temporal Metrics
Instance-based Analysis & Exploration
Interactive Experimentation
Algorithms for Attribution & Feature Visualization

## §9 WHERE

***Where has deep learning visualization been used?***

Application Domains & Models
A Vibrant Research Community

| Technical Term | Synonyms | Meaning |
|---|---|---|
| Neural Network | Artificial neural net, net | Biologically-inspired models that form the basis of deep learning; approximate functions dependent upon a large and unknown amount of inputs consisting of *layers* of *neurons* |
| Neuron | Computational unit, node | Building blocks of *neural networks*, entities that can apply *activation functions* |
| Weights | Edges | The trained and updated parameters in the *neural network* model that connect *neurons* to one another |
| Layer | Hidden layer | Stacked collection of *neurons* that attempt to extract features from data; a *layer's* input is connected to a previous *layer's* output |
| Computational Graph | Dataflow graph | Directed graph where nodes represent operations and edges represent data paths; when implementing *neural network* models, often times they are represented as these |
| Activation Functions | Transform function | Functions embedded into each *layer* of a *neural network* that enable the network represent complex non-linear decisions boundaries |
| Activations | Internal representation | Given a trained network one can pass in data and recover the *activations* at any *layer* of the network to obtain its current representation inside the network |
| Convolutional Neural Network | CNN, convnet | Type of *neural network* composed of convolutional *layers* that typically assume image data as input; these *layers* have depth unlike typical *layers* that only have width (number of *neurons* in a *layer*); they make use of filters (feature & pattern detectors) to extract spatially invariant representations |
| Long Short-Term Memory | LSTM | Type of *neural network*, often used in text analysis, that addresses the vanishing gradient problem by using memory gates to propagate gradients through the network to learn long-range dependencies |
| Loss Function | Objective function, cost function, error | Also seen in general ML contexts, defines what success looks like when learning a representation, i.e., a measure of difference between a *neural network's* prediction and ground truth |
| Embedding | Encoding | Representation of input data (e.g., images, text, audio, time series) as vectors of numbers in a high-dimensional space; oftentimes reduced so data points (i.e., their vectors) can be more easily analyzed (e.g., compute similarity) |
| Recurrent Neural Network | RNN | Type of *neural network* where recurrent connections allow the persistence (or "memory") of previous inputs in the network's internal state which are used to influence the network output |
| Generative Adversarial Networks | GAN | Method to conduct unsupervised learning by pitting a generative network against a discriminative network; the first network mimics the probability distribution of a training dataset in order to fool the discriminative network into judging that the generated data instance belongs to the training set |
| Epoch | Data pass | A complete pass through a given dataset; by the end of one *epoch*, a *neural network* will have seen every datum within the dataset once |

# WHY Visualize Deep Learning

- Interpretability & Explainability
- Debugging & Improving Models
- Comparing & Selecting Models
- Teaching Deep Learning Concepts

# Network Dissection: Quantifying interpretability of deep visual representations

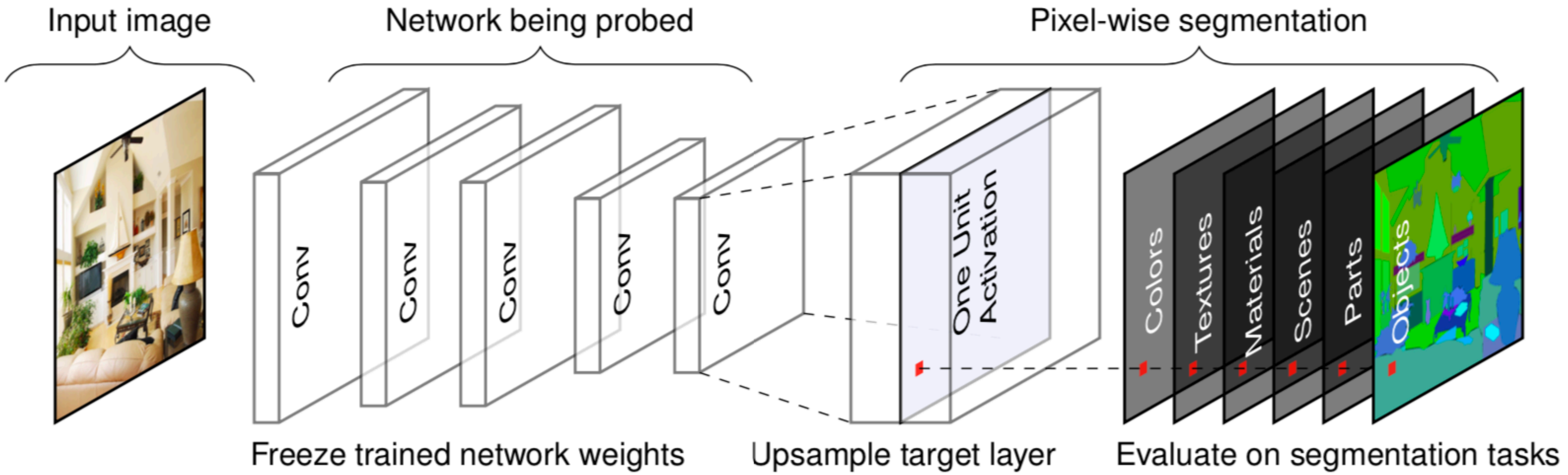



Figure 3. Illustration of network dissection for measuring semantic alignment of units in a given CNN. Here one unit of the last convolutional layer of a given CNN is probed by evaluating its performance on 1197 segmentation tasks. Our method can probe any convolutional layer.

https://arxiv.org/abs/1704.05796

https://www.youtube.com/watch?v=62O10xo4REA

https://www.youtube.com/watch?v=Xy6RcjXMa2c

# Visualization for Classification in Deep Neural Networks

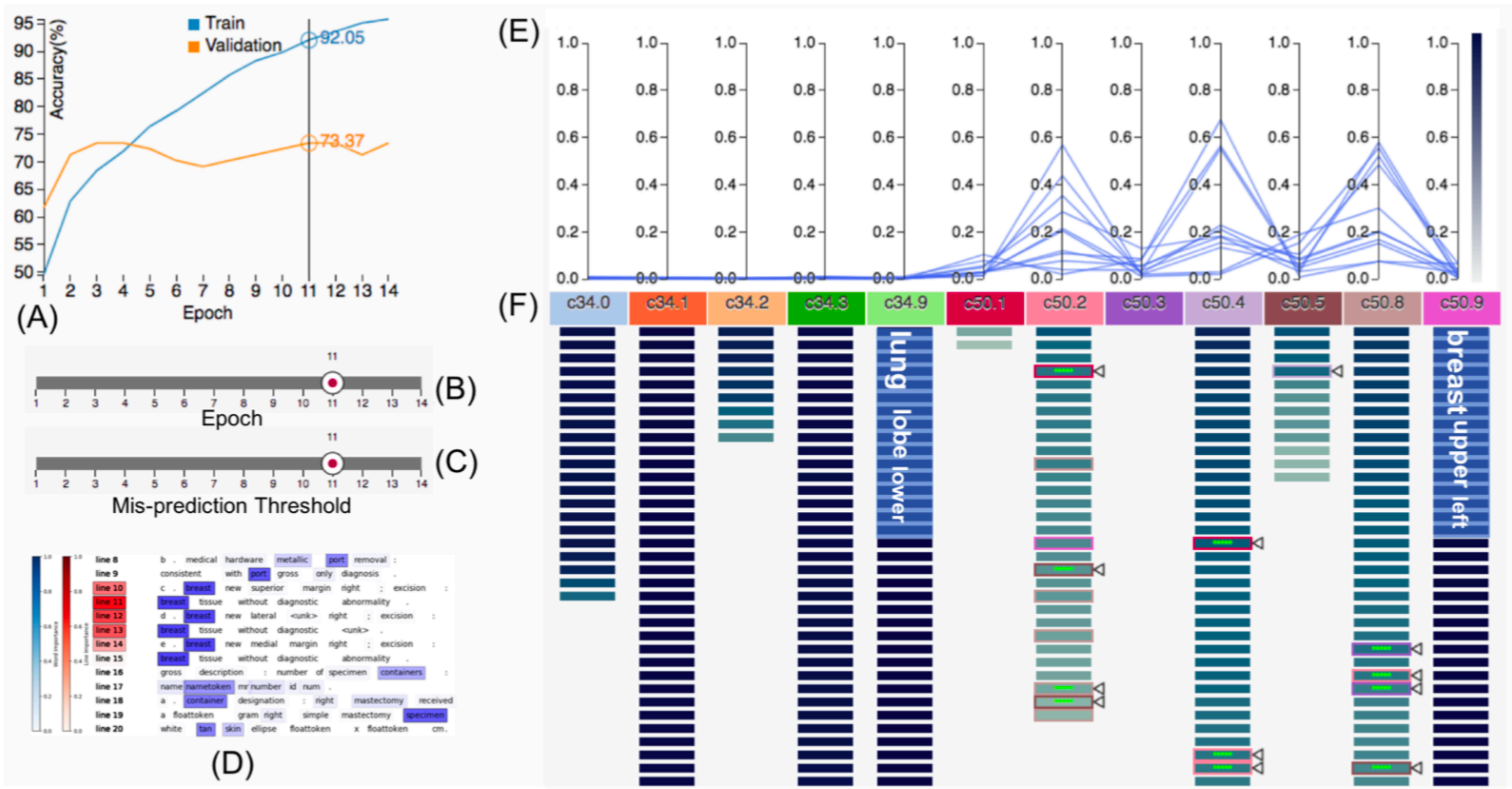



Figure 1: A visual analytics tool to understand classification results and suggest potential directions during the development of a Deep Neural Networks model.

https://vadl2017.github.io/paper/vadl0101_new.pdf

# Visualization for Classification in Deep Neural Networks

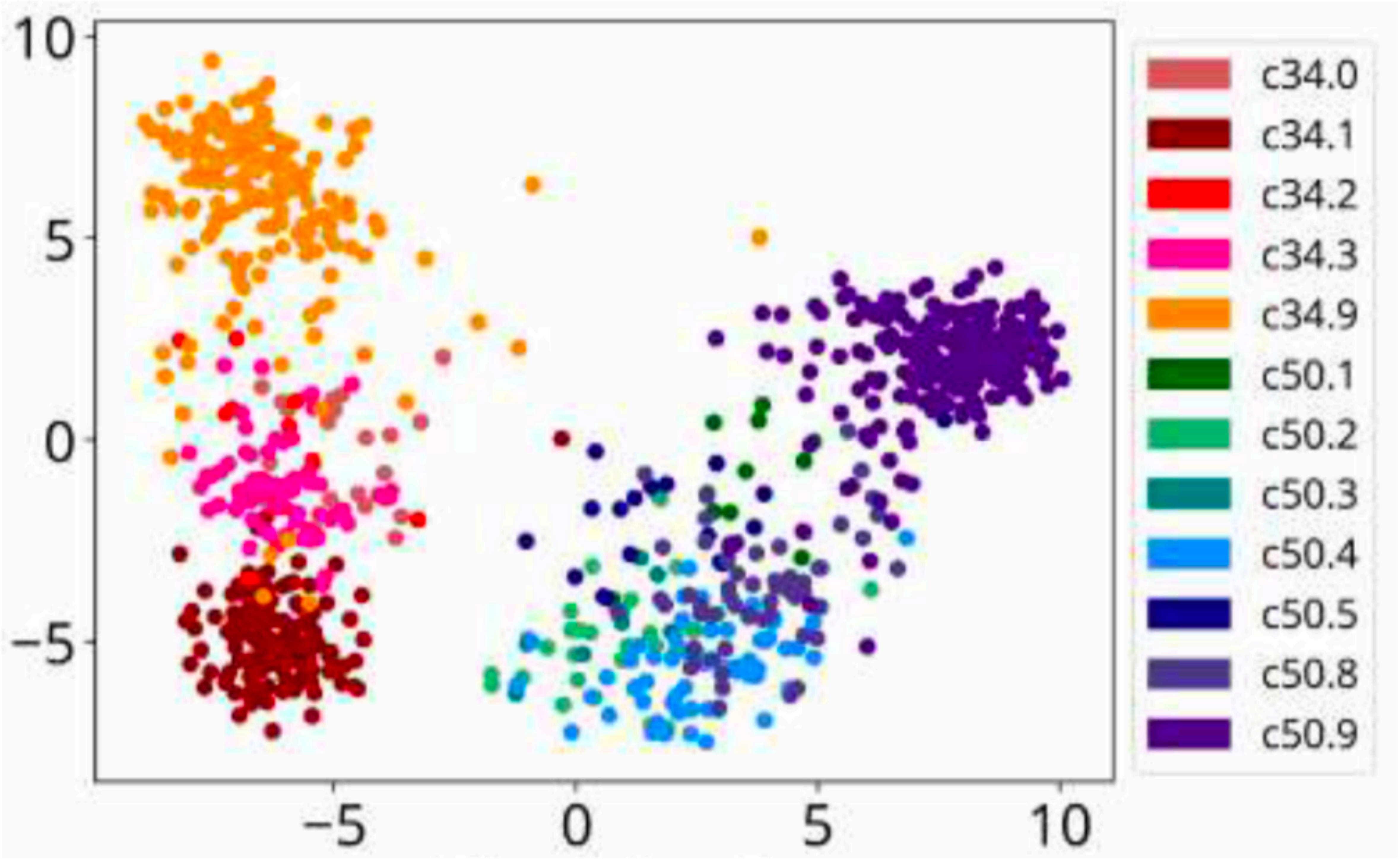



Figure 2: 2D-embedding of cancer pathology reports using PCA. The colors of the points denote their classes.

https://vadl2017.github.io/paper/vadl0101_new.pdf

# Visualization for Classification in Deep Neural Networks

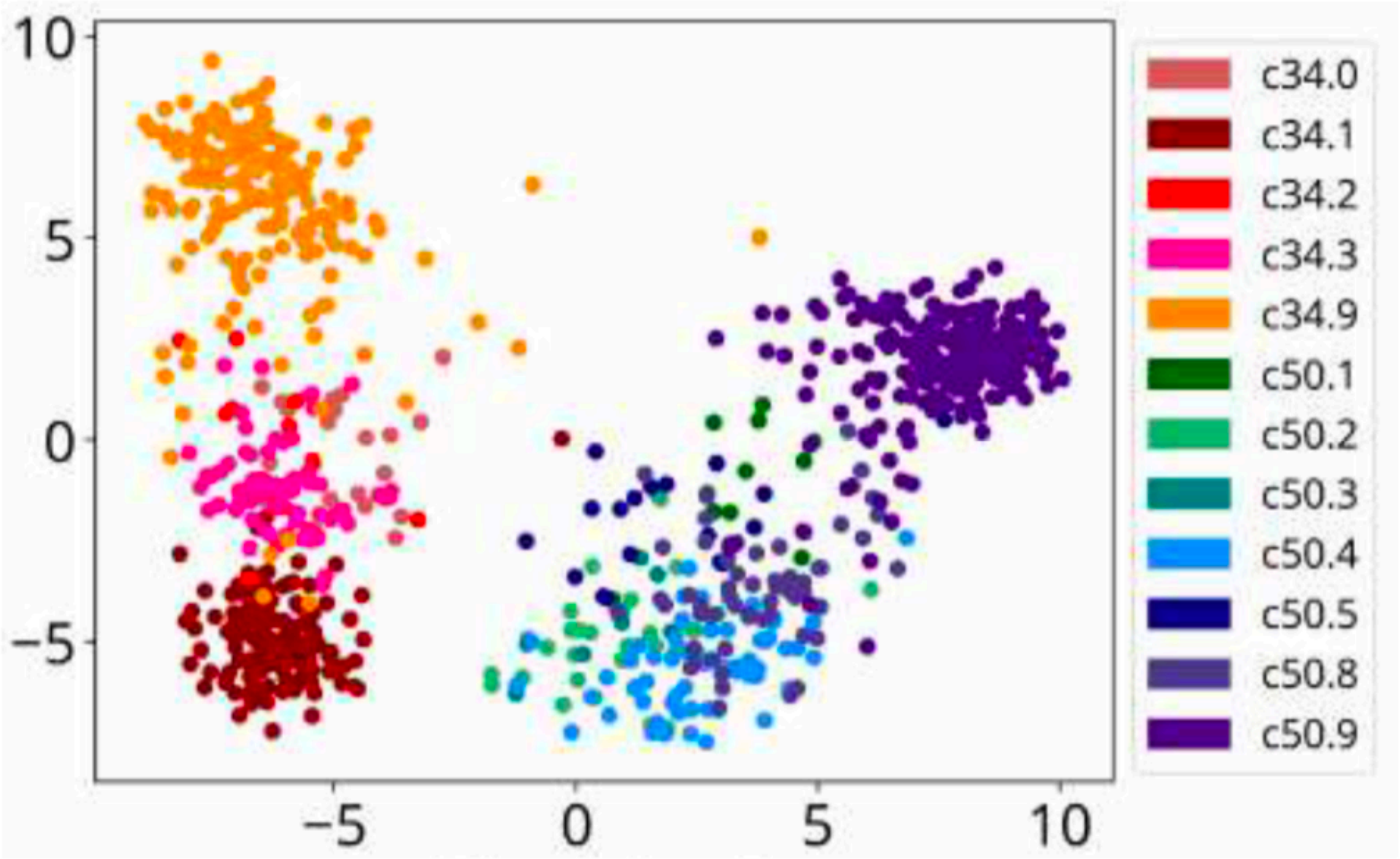



Figure 2: 2D-embedding of cancer pathology reports using PCA. The colors of the points denote their classes.

https://vadl2017.github.io/paper/vadl0101_new.pdf

# Visualization for Classification in Deep Neural Networks

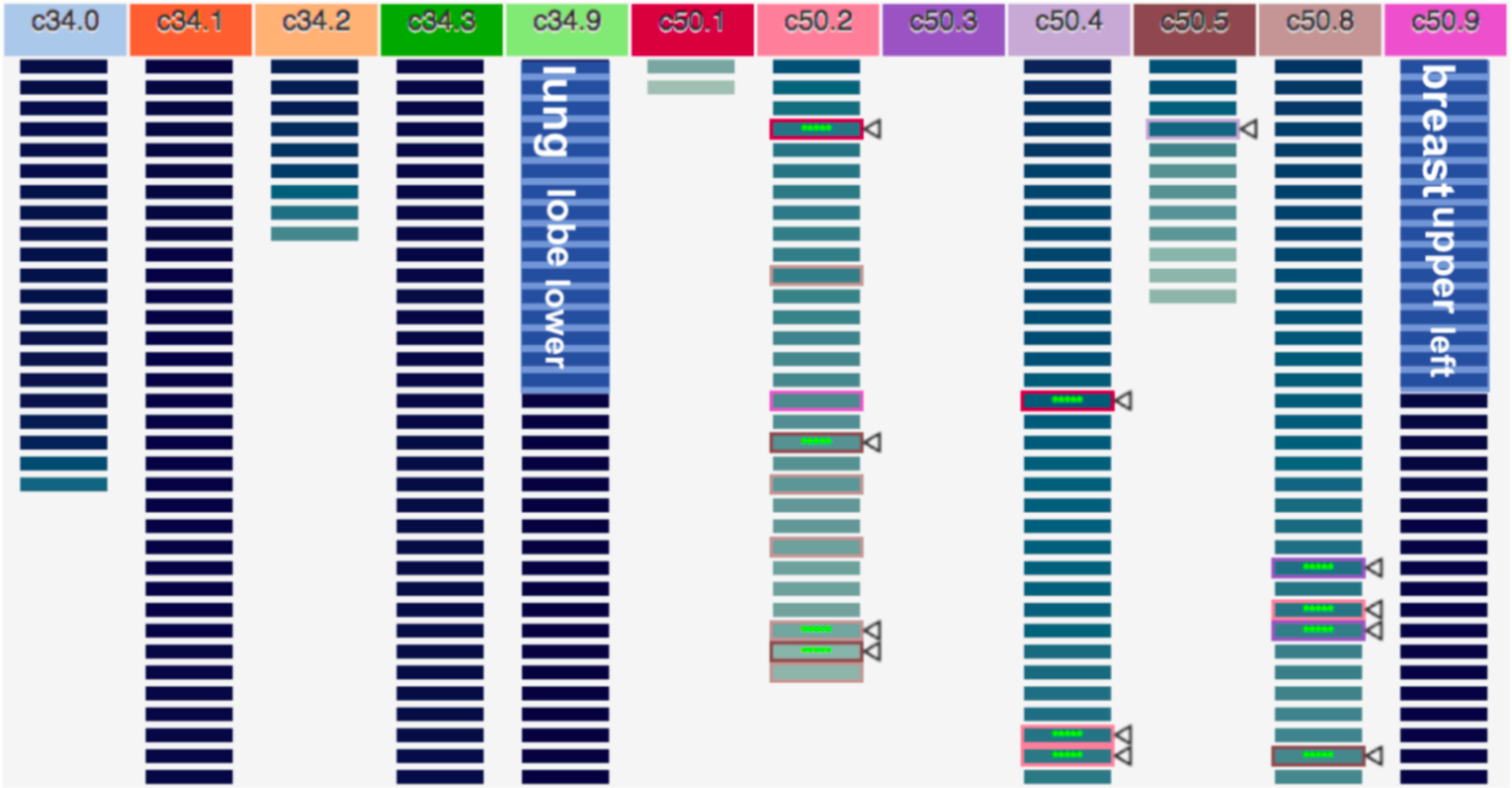
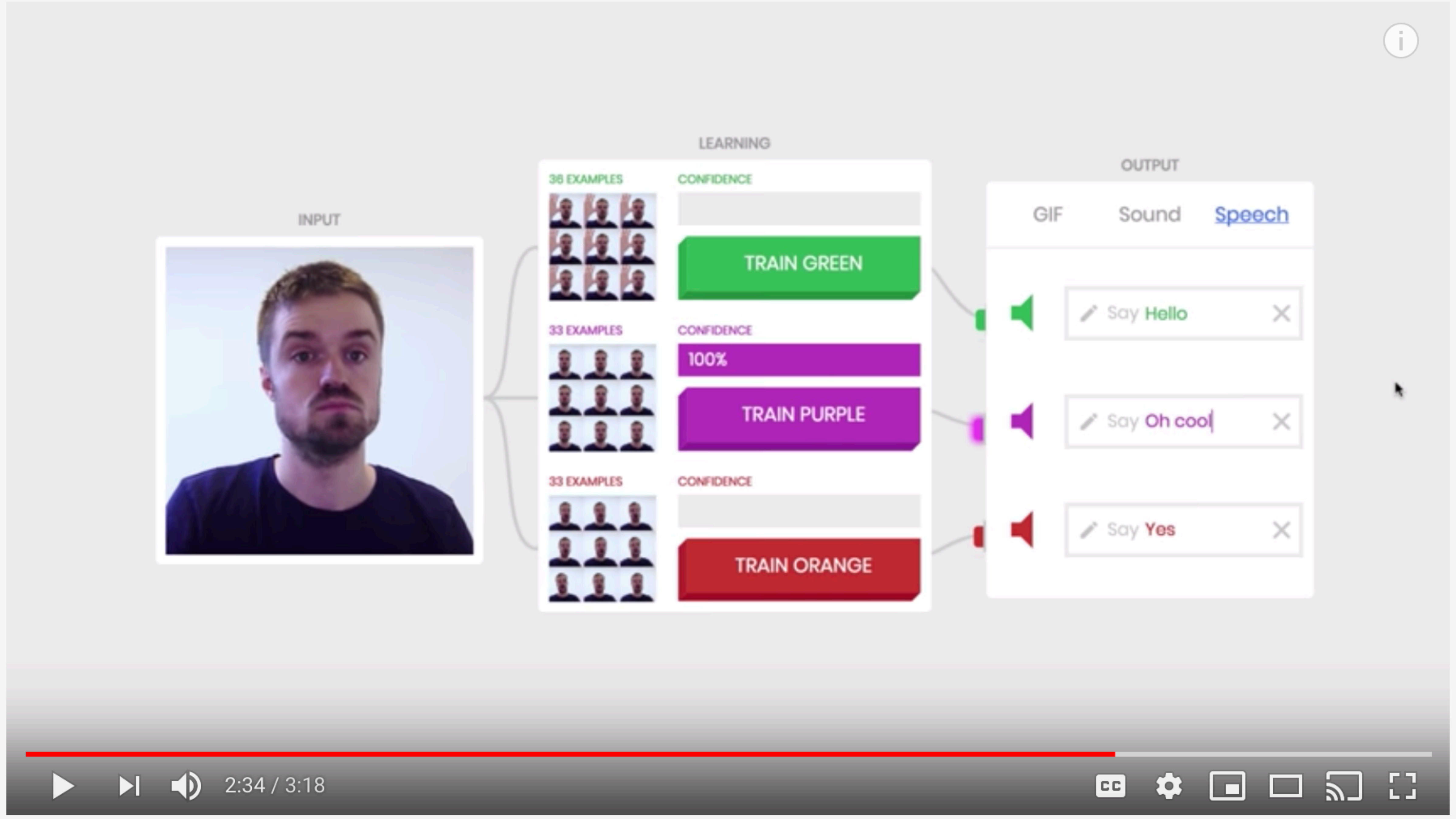



Figure 3: Classification View: Samples (small narrow boxes) are visualized according to their predicted classes. The box colors represent their predicted scores. Outlined boxes are incorrectly predicted samples. Small triangles denote the samples whose the misclassified number is more than mis-prediction threshold value.

https://vadl2017.github.io/paper/vadl0101_new.pdf

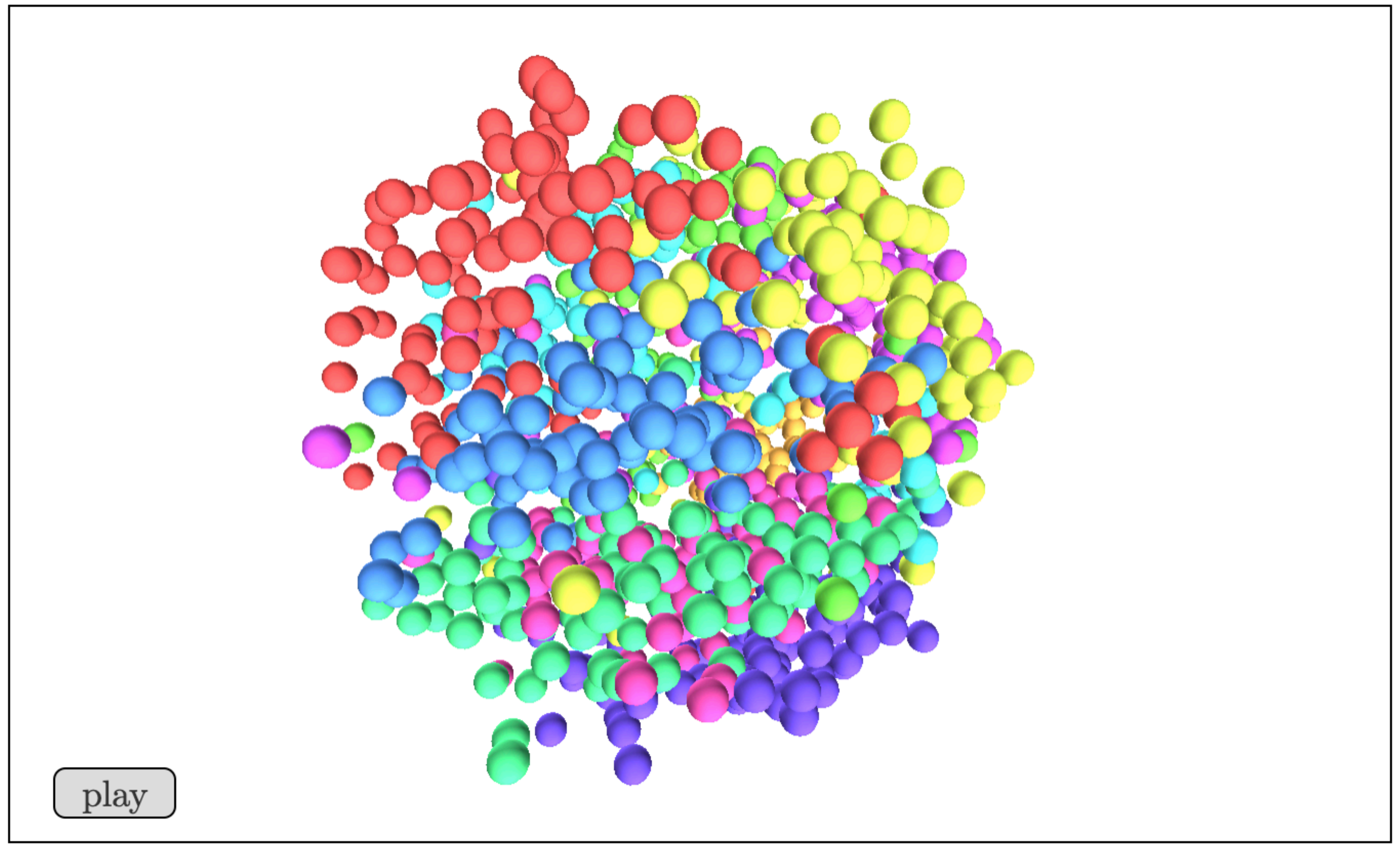
# Teachable machines




A.I. Experiments: Teachable Machine

https://www.youtube.com/watch?v=3BhkeY974Rg

# Visualizing MNIST

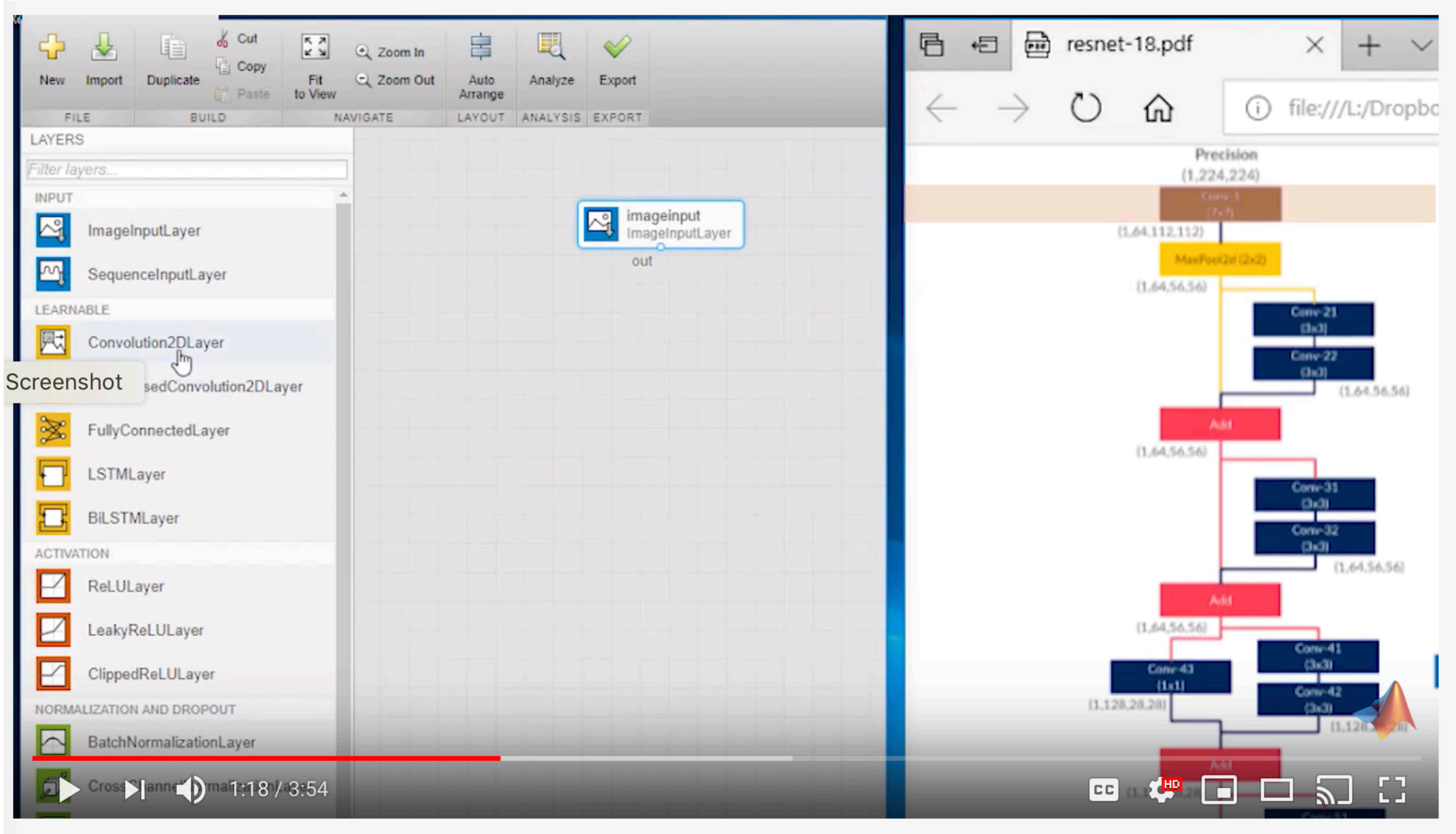



**Visualizing MNIST with MDS in 3D**
(click and drag to rotate)

http://colah.github.io/posts/2014-10-Visualizing-MNIST/

# From MATLAB (commercial tools)

## Interactively Build, Visualize, and Edit Deep Learning Networks




https://www.youtube.com/watch?v=vX9rw6KIMa8

# Thanks!

Any questions?

You can find me at: beiwang@sci.utah.edu

# CREDITS

Special thanks to all people who made and share these awesome resources for free:

- Presentation template designed by Slidesmash
- Photographs by unsplash.com and pexels.com
- Vector Icons by Matthew Skiles

# Presentation Design

This presentation uses the following typographies and colors:

## Free Fonts used:

http://www.1001fonts.com/oswald-font.html

https://www.fontsquirrel.com/fonts/open-sans

## Colors used