Visual Computing Group
Ulm University

Visualization-Based Neural Network Introspection

Alex Bäuerle

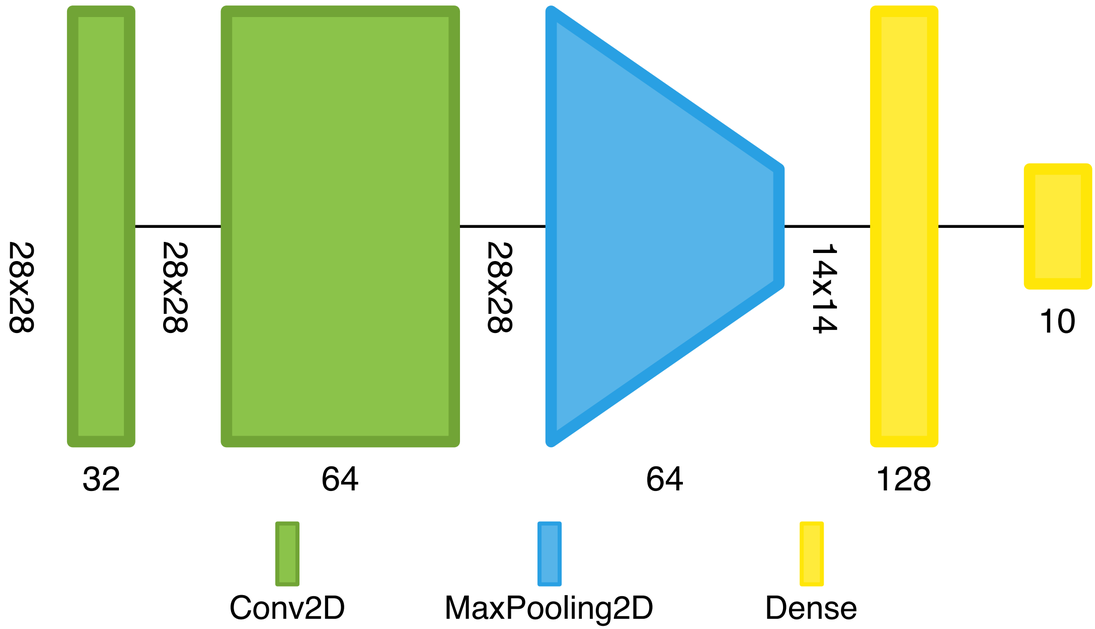
Defense, 21.12.2022

Potential

"Recent and rapid advances in AI research and technologies are widely expected to bring about pervasive and far-reaching social transformation on a global scale. The celebratory rhetoric that accompanied the emergence of these technologies several years ago [...] has recently become considerably more muted in the face of rising public anxiety about the possible adverse effects [...]"

Risk

Karen Yeung 2020, International Legal Materials, Recommendation of the council on artificial intelligence (OECD)



27,232 Parameters

AI bot ChatGPT writes smart essays — should professors worry?

Chris Stokel-Walker, Nature News Explainer, December 2022

ChatGPT and How AI Disrupts Industries

Ajay Agrawal, Joshua Gans, and Avi Goldfarb, Harvard Business Review, December 2022

How AI-Driven Technology Is Increasing Food Security, And Improving The Lives Of Farmers Worldwide

Forbes, August 2021

How Will AI Impact the Transportation Industry?

Maayan, Dataversity, April 2021

Here's how AI can help fight climate change

World Economic Forum, August 2021

How Machine Learning Will Transform Biomedicine

Goecks, et al. "How machine learning will transform biomedicine.", Cell, 2020

AI Is a Game-Changer in the Fight Against Hunger and Poverty. Here's Why

Bennington-Castro, NBCNews, June 2021

AI Art Is Challenging the Boundaries of Curation

Raphaël Millière, WIRED, July 2022

“The full cycle of a machine learning project is not just modeling. It is finding the right data, deploying it, monitoring it, feeding data back, showing safety [...]. [That goes] beyond doing well on the test set, which fortunately or unfortunately is what we in machine learning are great at."

Andrew Ng, IEEE Spectrum, May 2021

Introspection

What has a model learned?

How was my model trained?

What does my model do?

Where does my model fail?

Develop methods for ML introspection and make them accessible through visualization.

Visualization

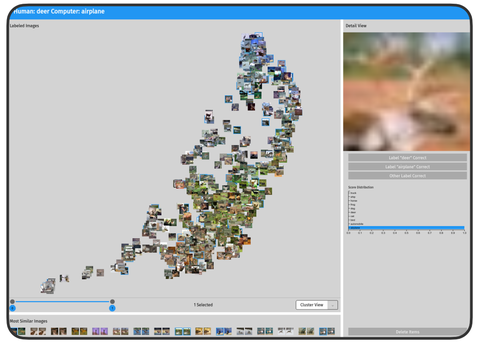
How do we communicate results?

How do me make introspection usable?

How do we explore data?

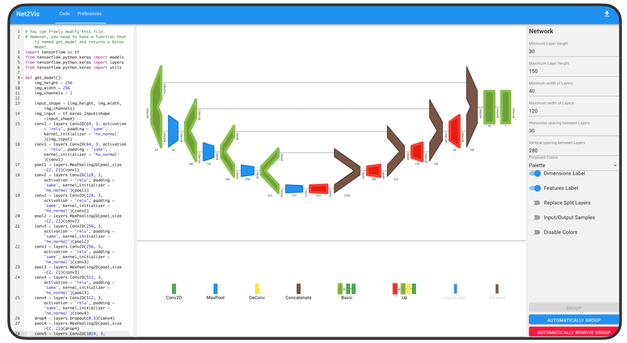
Can we see how a model works?

Quality Assurance



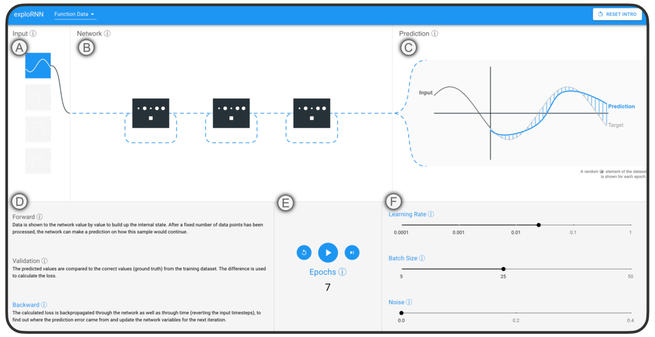
Bäuerle et al. 2020, CGF
Classifier-Guided Visual Correction of Noisy Labels for Image Classification Tasks

Communication

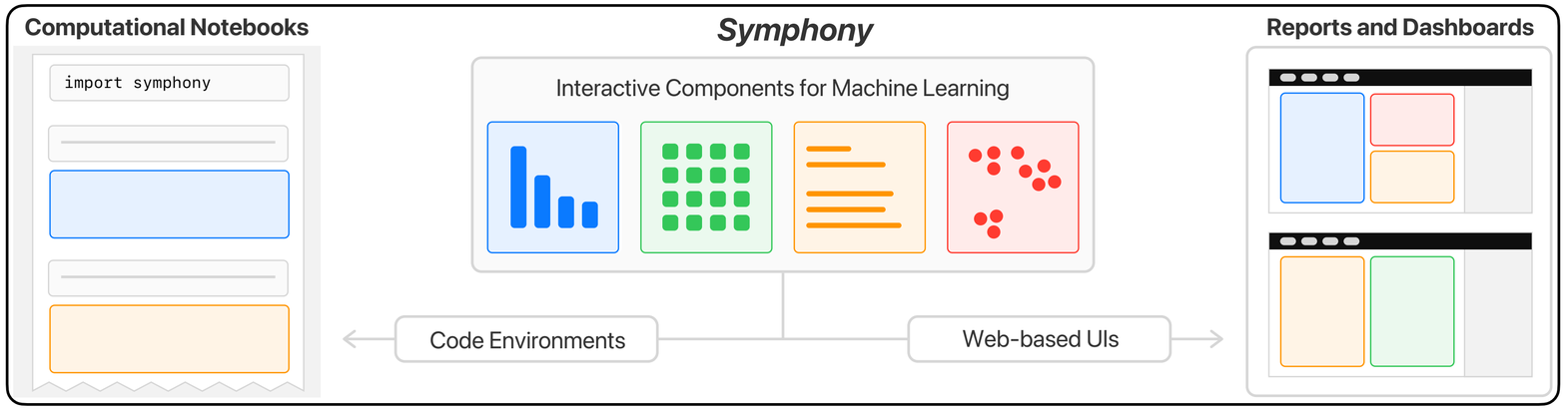


Bäuerle et al. 2021, TVCG
Net2Vis - A Visual Grammar for Automatically Generating Publication-Tailored CNN Architecture Visualizations

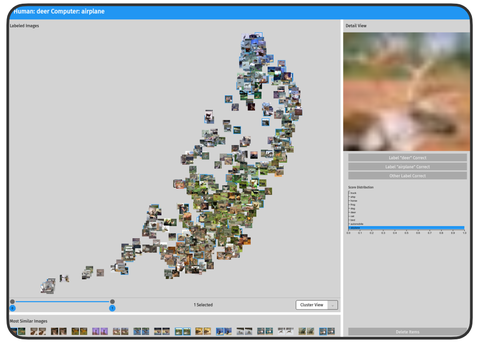
Education



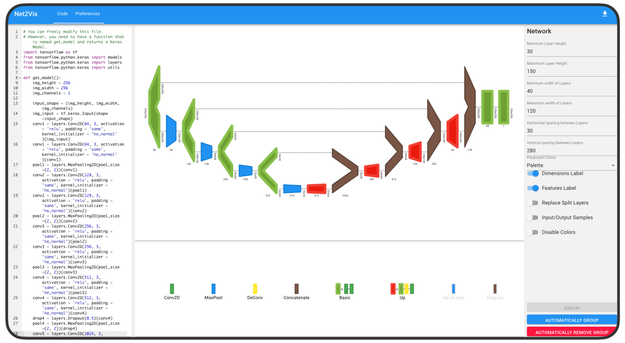
Bäuerle et al. 2022, The Visual Computer
exploRNN: Understanding Recurrent Neural Networks through Visual Exploration



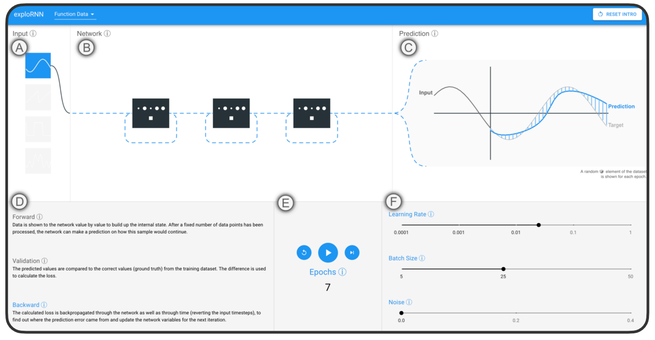
Bäuerle and Cabrera et al. 2022, CHI
Symphony: Composing Interactive Interfaces for Machine Learning



Bäuerle et al. 2020, CGF
Classifier-Guided Visual Correction of Noisy Labels for Image Classification Tasks

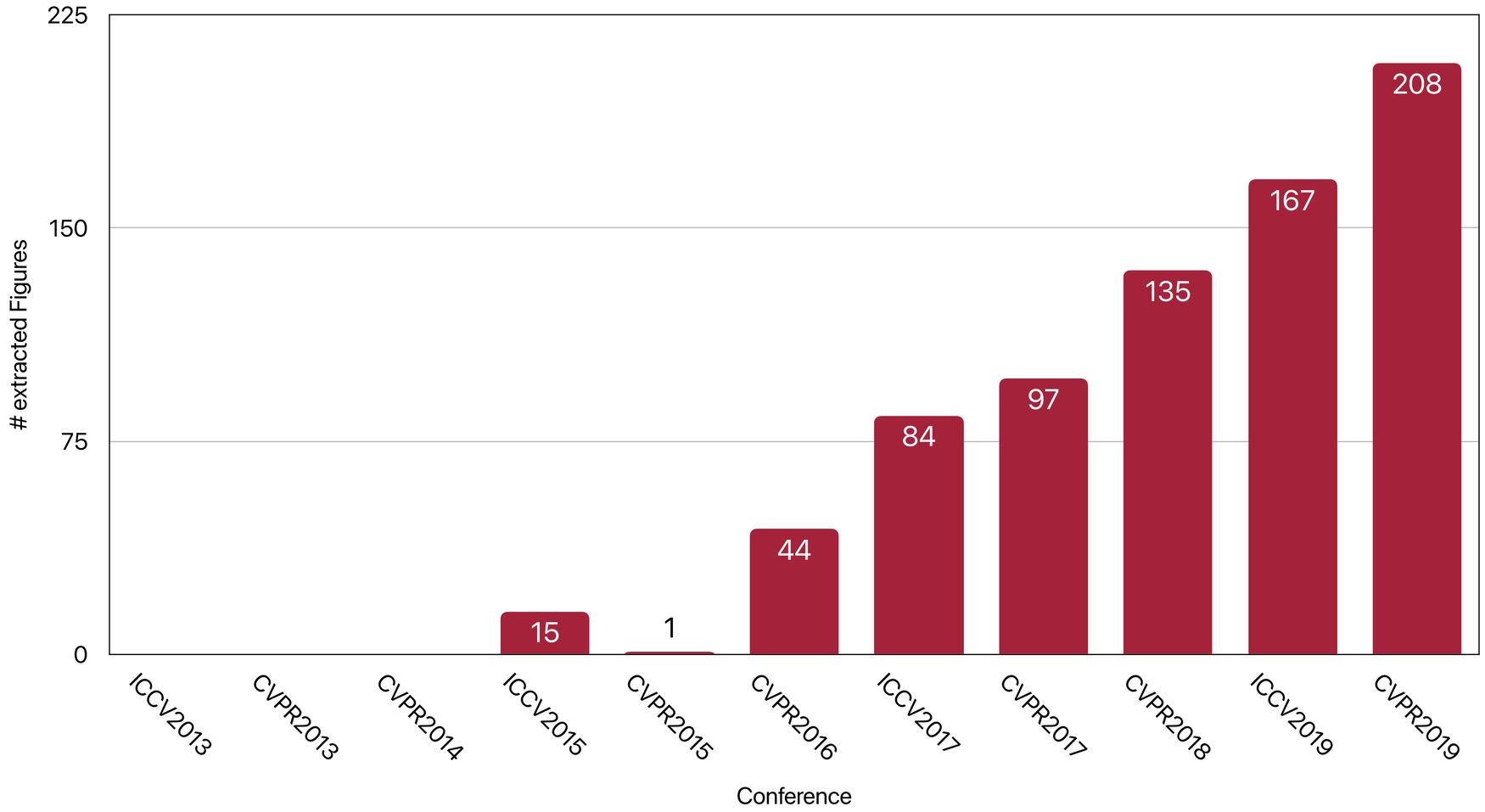


Bäuerle et al. 2021, TVCG
Net2Vis - A Visual Grammar for Automatically Generating Publication-Tailored CNN Architecture Visualizations

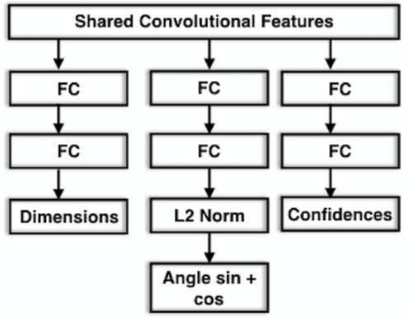


Bäuerle et al. 2022, The Visual Computer
exploRNN: Understanding Recurrent Neural Networks through Visual Exploration

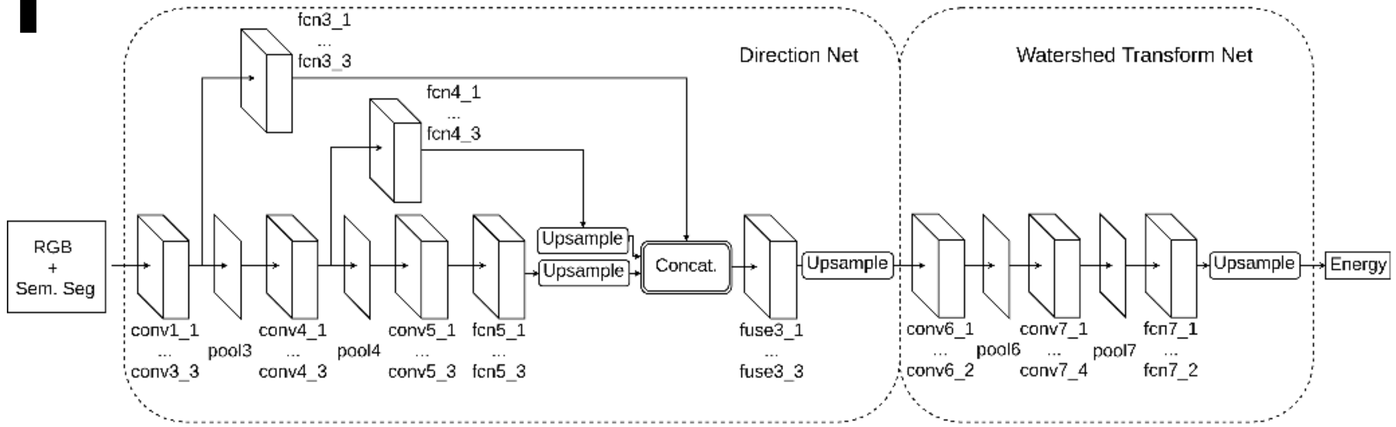
Architecture Figures in Publications



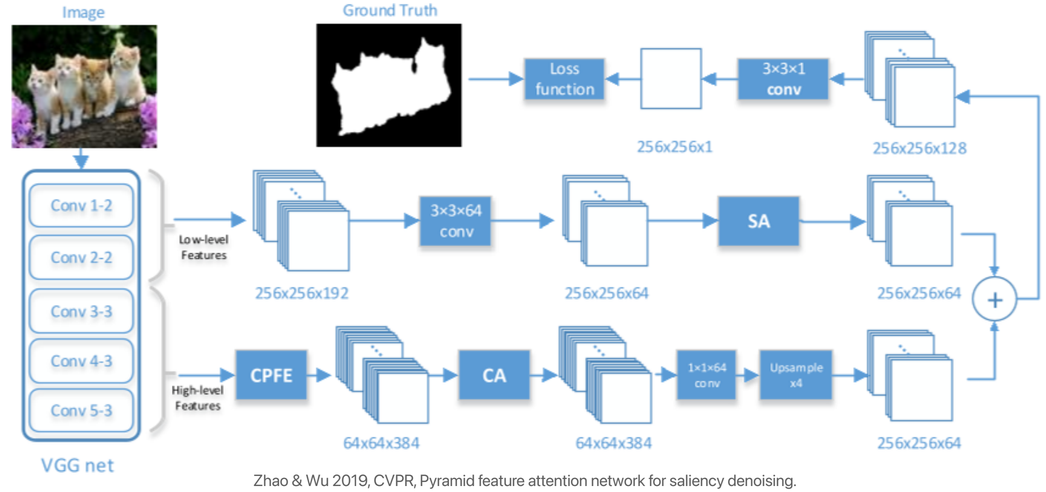
Current Situation



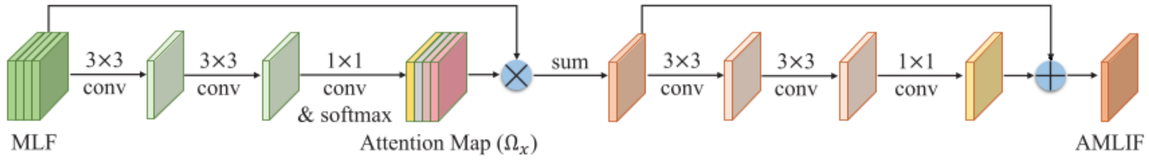
Mousavian et. al. 2017, CVPR, 3d bounding box estimation using deep learning and geometry.



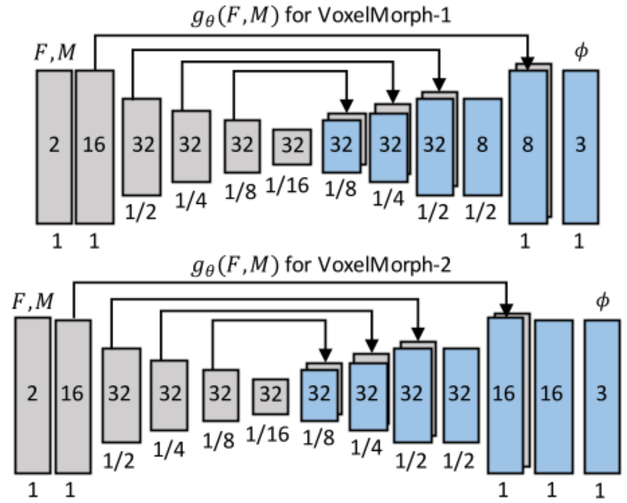
Bai & Urtasun 2017, CVPR, Deep watershed transform for instance segmentation.



Zhao & Wu 2019, CVPR, Pyramid feature attention network for saliency denoising.



Deng et. al. 2019, CVPR, Deep multi-model fusion for single image dehazing.



Balakrishnan et. al. 2018, CVPR, An unsupervised learning model for deformable medical image registration.

Net2Vis

Net2Vis – A Visual Grammar for Automatically Generating Publication-Ready CNN Architecture Visualizations

Alex Bäuerle, Christian van Onzenoodt, and Timo Ropinski

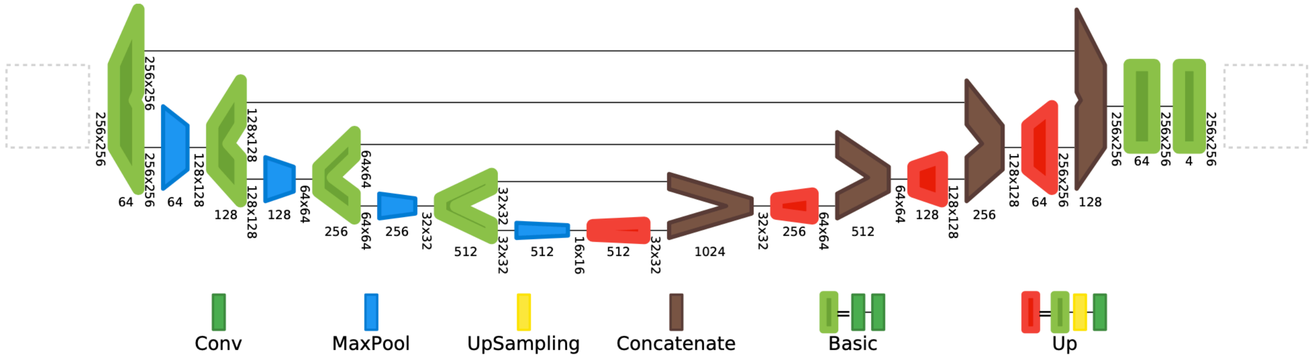
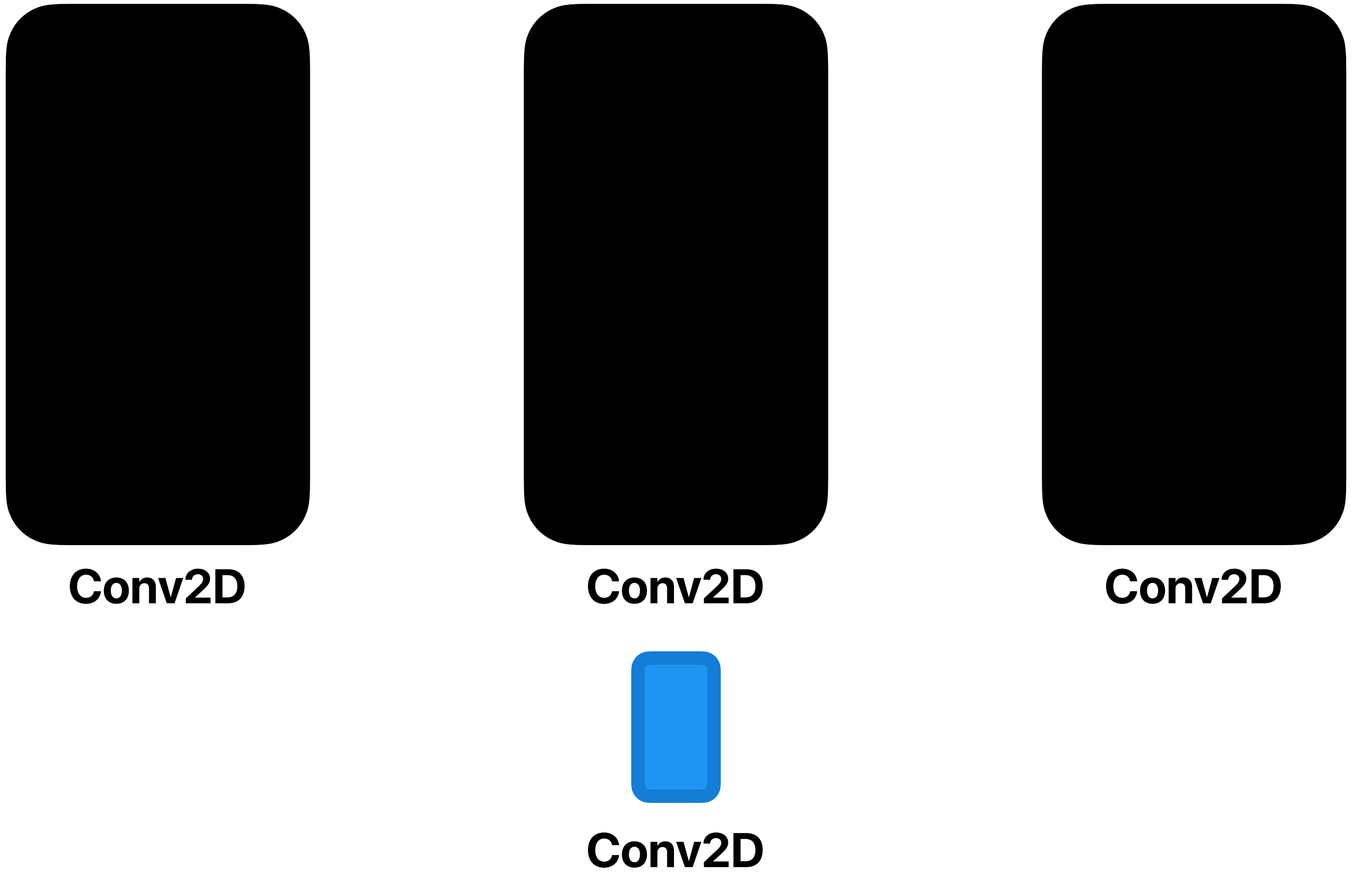


Fig. 1: Visualization of a U-Net variant automatically generated using our approach, based on the Keras code describing the architecture. Data flows from left to right. Glyphs represent layers or aggregates, while lines represent connections. Glyph widths communicate feature size, while heights communicate the spatial resolution. Both values are also given through labels, while dashed boxes on the left and right serve as placeholders to provide input and output samples. The legend communicates layer types and the composition of aggregates.

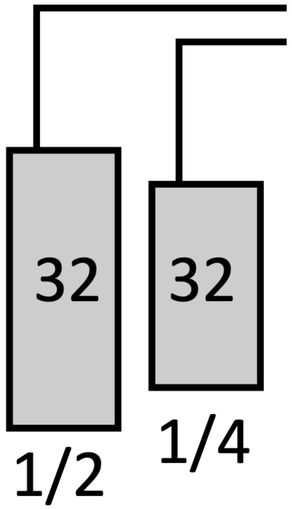
Abstract—To convey neural network architectures in publications, appropriate visualizations are of great importance. While most current deep learning papers contain such visualizations, these are usually handcrafted just before publication, which results in a lack

Visualization Analysis

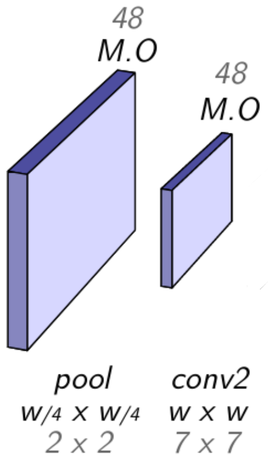
Layer Properties



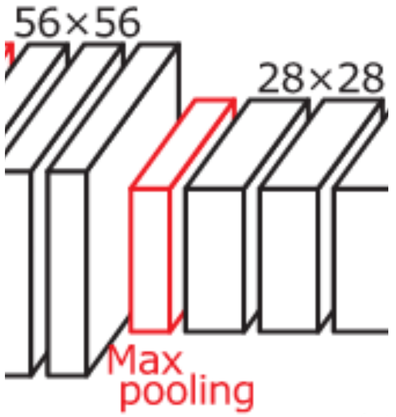
Layer Properties



Balakrishnan et. al., 2018, CVPR, An unsupervised learning model for deformable medical image registration.

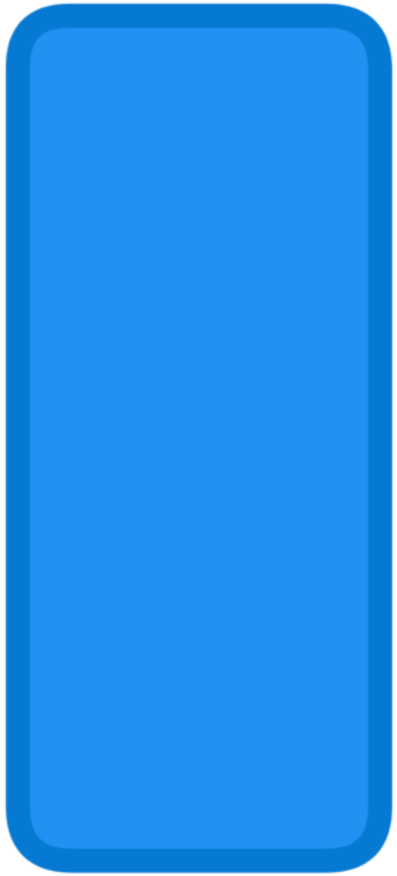


Teney & Martial 2016, ACCV, Leaning to extract motion from videos in convolutional neural networks.

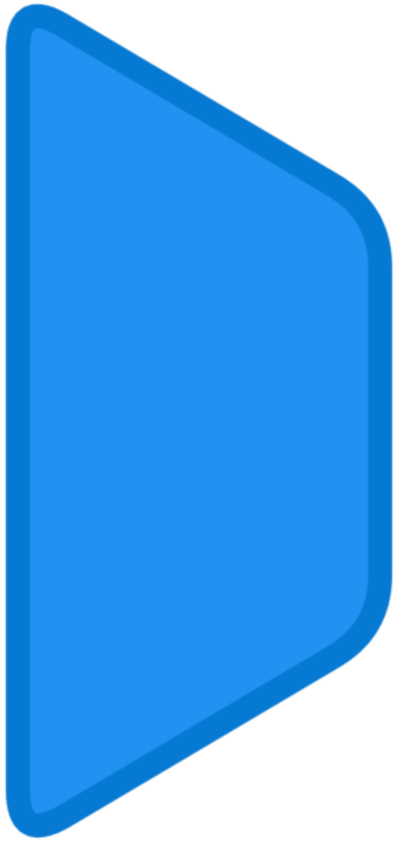


Noh et. al. 2015, ICCV, Learning deconvolution network for semantic segmentation.

Layer Properties

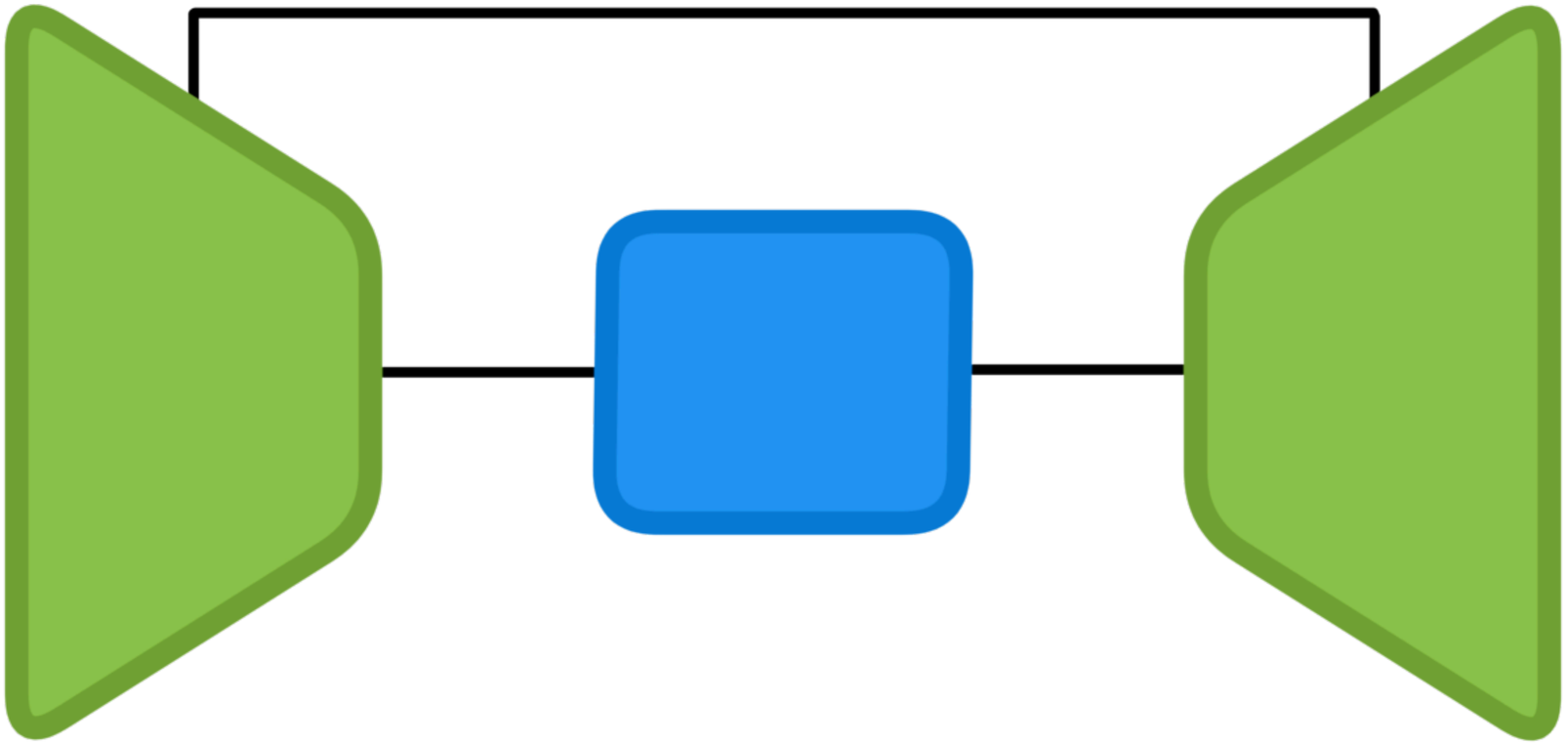


Layer Properties

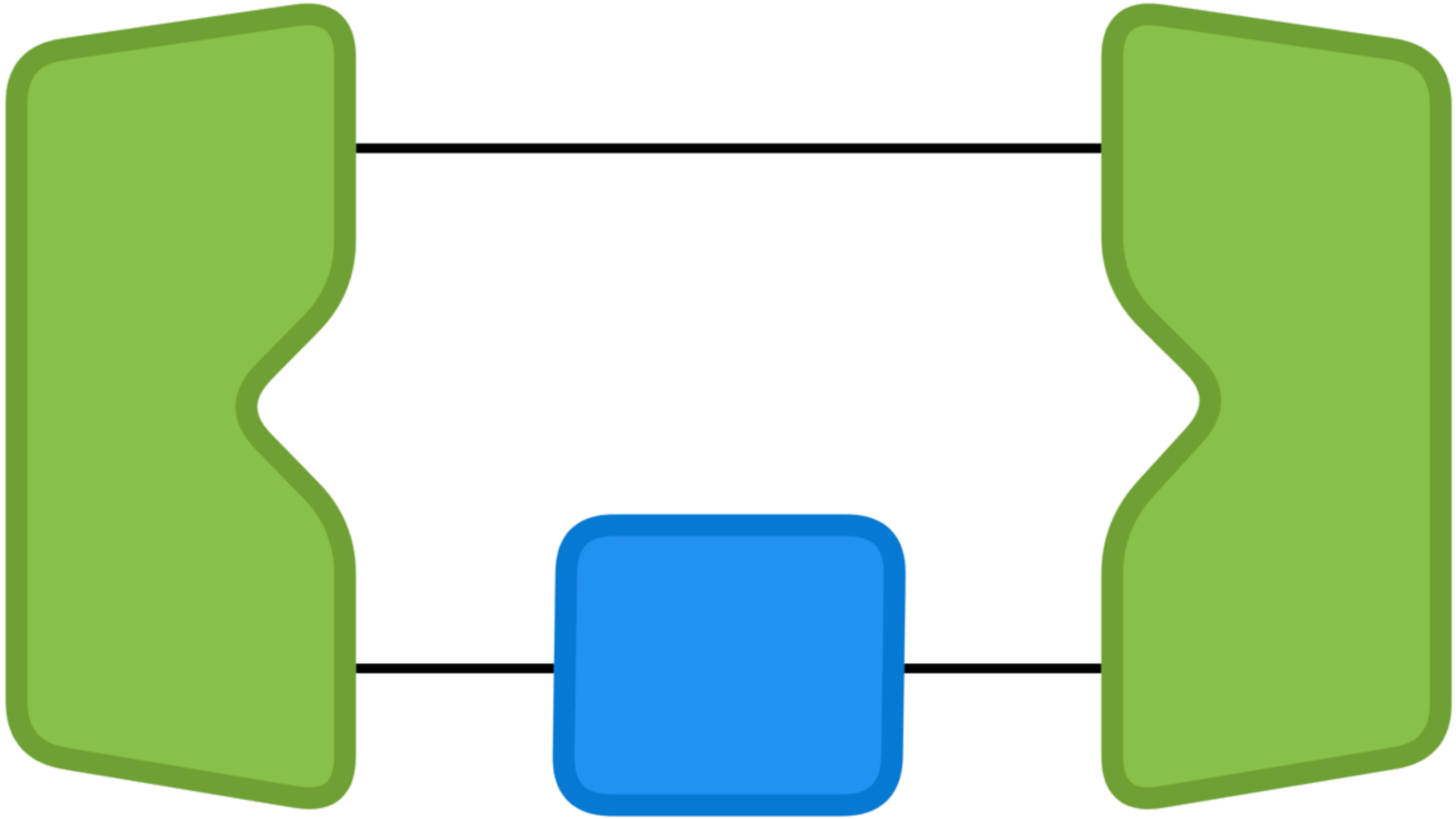


Model Properties

Model Properties



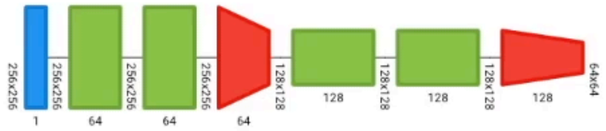
Model Properties



Net2Vis Code Preferences

```python
# You can freely modify this file.
# However, you need to have a function that is named get_model and returns a Keras Model.
import keras as k
from keras import models
from keras import layers
from keras import utils

def get_model():
    img_height = 256
    img_width = 256
    img_channels = 1

    input_shape = (img_height, img_width, img_channels)
    img_input = k.Input(shape=input_shape)
    conv1 = layers.Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initializer = 'he_normal')(img_input)
    conv1 = layers.Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initializer = 'he_normal')(conv1)
    pool1 = layers.MaxPooling2D(pool_size=(2, 2))(conv1)
    conv2 = layers.Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_initializer = 'he_normal')(pool1)
    conv2 = layers.Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_initializer = 'he_normal')(conv2)
    pool2 = layers.MaxPooling2D(pool_size=(2, 2))(conv2)

    model = models.Model(img_input, pool2)

    return model

```

UPDATE



Network

Minimum Layer Height
30

Maximum Layer Height
100

Minimum width of Layers
20

Maximum width of Layers
80

Horizontal spacing between Layers
20

Vertical spacing between Layers
200

Porposed Colors
Palette

- Dimensions Label
- Features Label
- Name Label
- Replace Split Layers
- Input/Output Samples
- Disable Colors

GROUP

AUTOMATICALLY GROUP

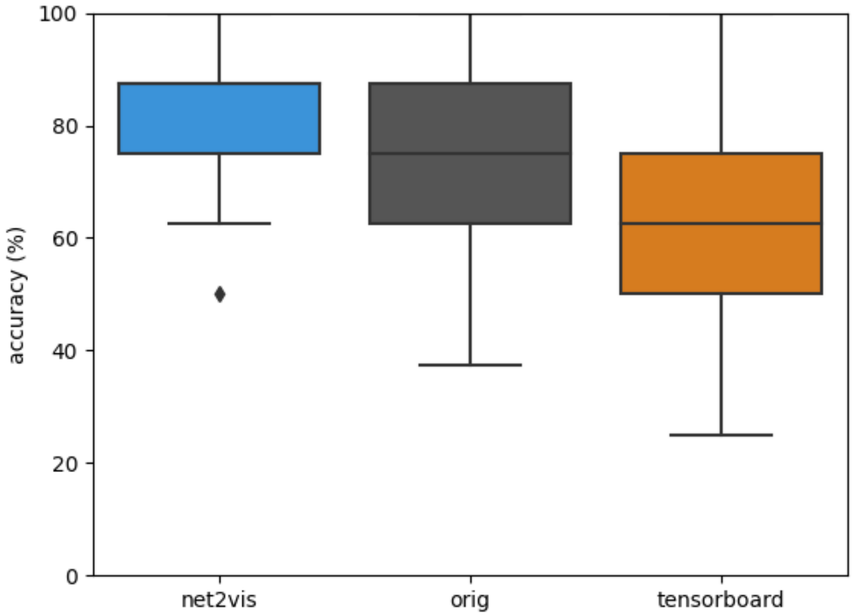
AUTOMATICALLY REMOVE GROUP

Quantitative User Study

10 students after DL course

Architecture visualizations from Net2Vis, TensorBoard, Paper

1. Architecture questions - best accuracy
2. Comparison questions - preferred
3. Own architecture - excellent usability



Adoption

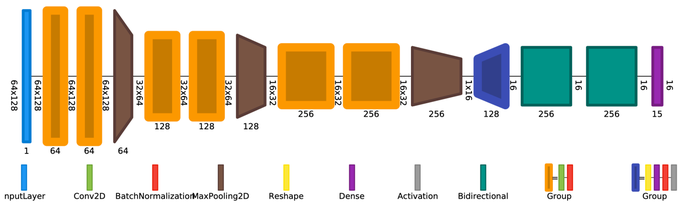


Figure 1: Architecture of the CNN-RNN-CTC. Each orange layer composes a 2D convolution, batch norm, and RELU activation. Kernel sizes are (7,7) and (5,5) for the first two layers and (3,3) for all subsequent layers. To translate from (2D × Features) to (timesteps × features), we use pooling with kernel (16,2) and a single dense layer with 128 neurons, batch norm, and RELU activation. The bidirectional layers use concatenation of 128 GRU units with 0.5 dropout. The output sequences are fed into a 15 output neurons softmax activation layer.

Mueller-Gastell et. al., "A Multi-digit OCR System for Historical Records (Computer Vision)", Stanford CS230, 2020

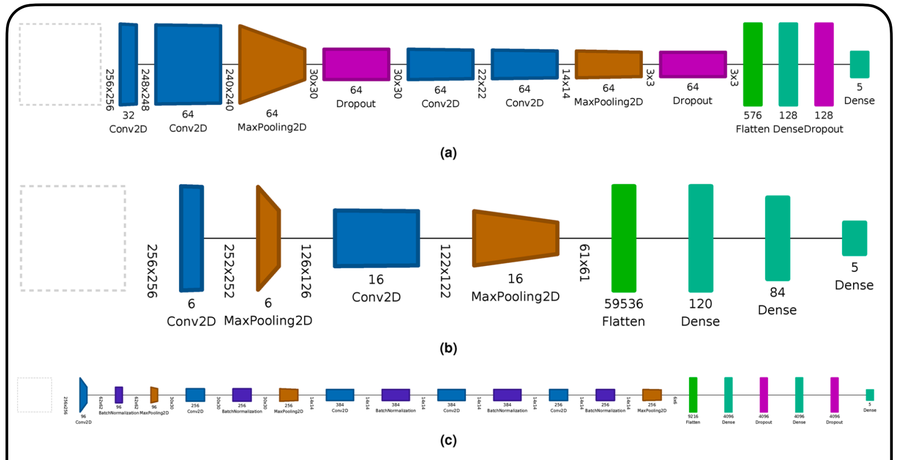


Figure 4. CNN architectures of (**a**) Custom, (**b**) LeNet, (**c**) AlexNet.

Álvarez-Aparicio et al. "Biometric recognition through gait analysis", Nature Scientific Reports, 2022

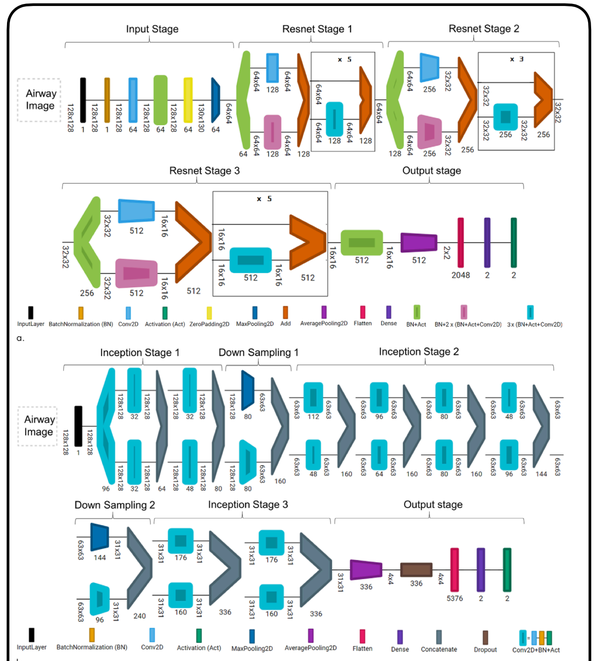


Figure 3: **(a)** Schematic of the ResNet-50 architecture generated using Net2Vis visualization tool [10]. For each ResNet stage, the layers in the boxes are repeated in series as indicated within the box. **(b)** Schematic of the mini-Inception architecture generated using Net2Vis visualization tool [10]. Add = addition operation, Average-Pooling2D = two-dimensional average pooling operation, Conv2D = two-dimensional convolution, MaxPooling2D = two-dimensional maximum pooling operation, Zero-Padding2D = two-dimensional padding with zeroes.

Somasundaram et al. "Automatic Detection of Inadequate Pediatric Lateral Neck Radiographs of the Airway and Soft Tissues using Deep Learning", Radiology: Artificial Intelligence 2.5, 2020

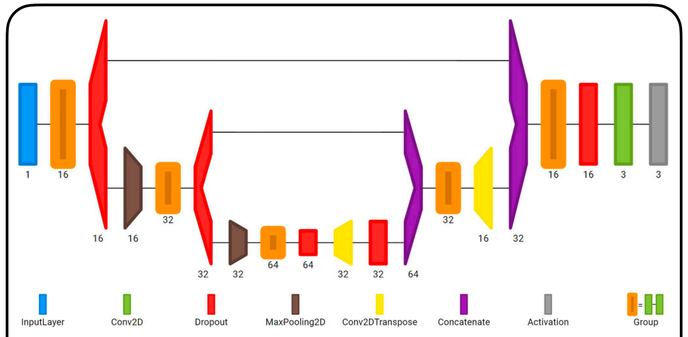


Figure 6. Architecture of proposed encoder-decoder convolutional neural network. It takes an image of 384×384 pixels resolution and process it in several convolutional, pooling, transposed convolutional and concatenation layers, before the final pixelwise semantic segmentation is performed with the "softmax" activation in the last classification layer.

Vernikouskaya et al. "Body fat compartment determination by encoder-decoder convolutional neural network: application to amyotrophic lateral sclerosis", Nature Scientific Reports, 2022

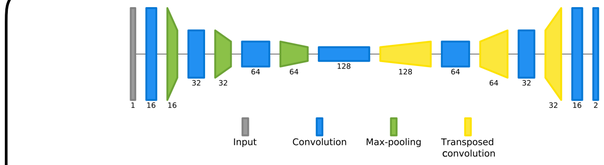


Fig. 6. The neural network architecture we use for segmenting fibril images. It transforms the input image into a 128-dimensional feature vector, for which the spatial resolution is reduced by a factor of eight. In the following, this vector is utilized to produce the two-dimensional segmentation map used as the crossover prediction. Due to its fully convolutional nature, the input and output sizes of the network are arbitrary.

Weber et. al., "Automatic identification of crossovers in cryo-EM images of murine amyloid protein A fibrils with machine learning", Journal of Microscopy, 2020

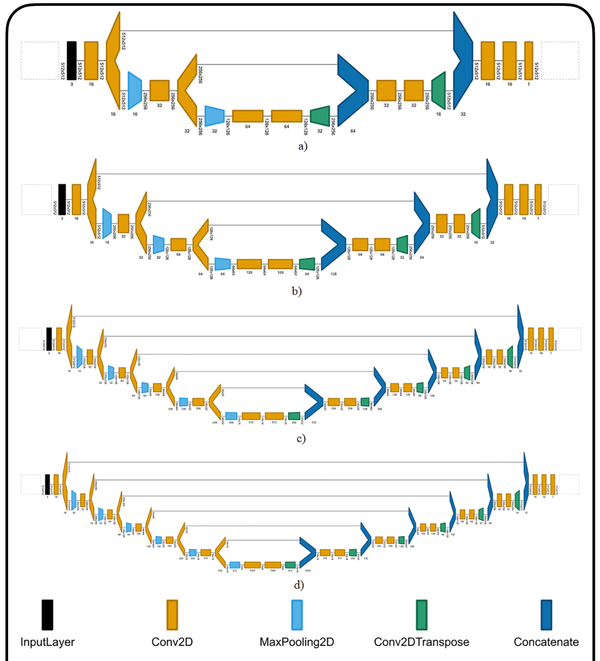


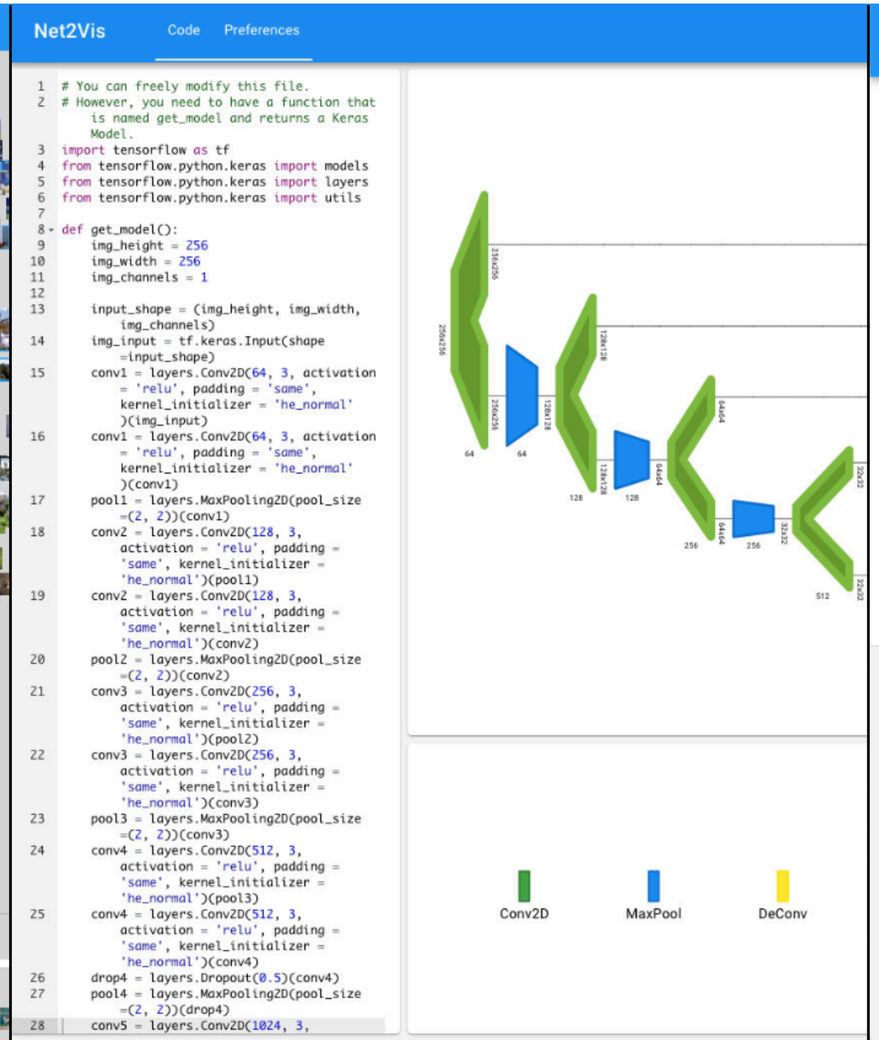
Fig. 6 **a)** U-Net shallower 2 architecture **b)** U-Net shallower 1 architecture **c)** U-Net deeper 1 architecture **d)** U-Net deeper 2 architecture (visualized using (Bauerle et al. 2021))

Sariturk et al. "Performance evaluation of shallow and deep CNN architectures on building segmentation from high-resolution images", Earth Science Informatics, 2022

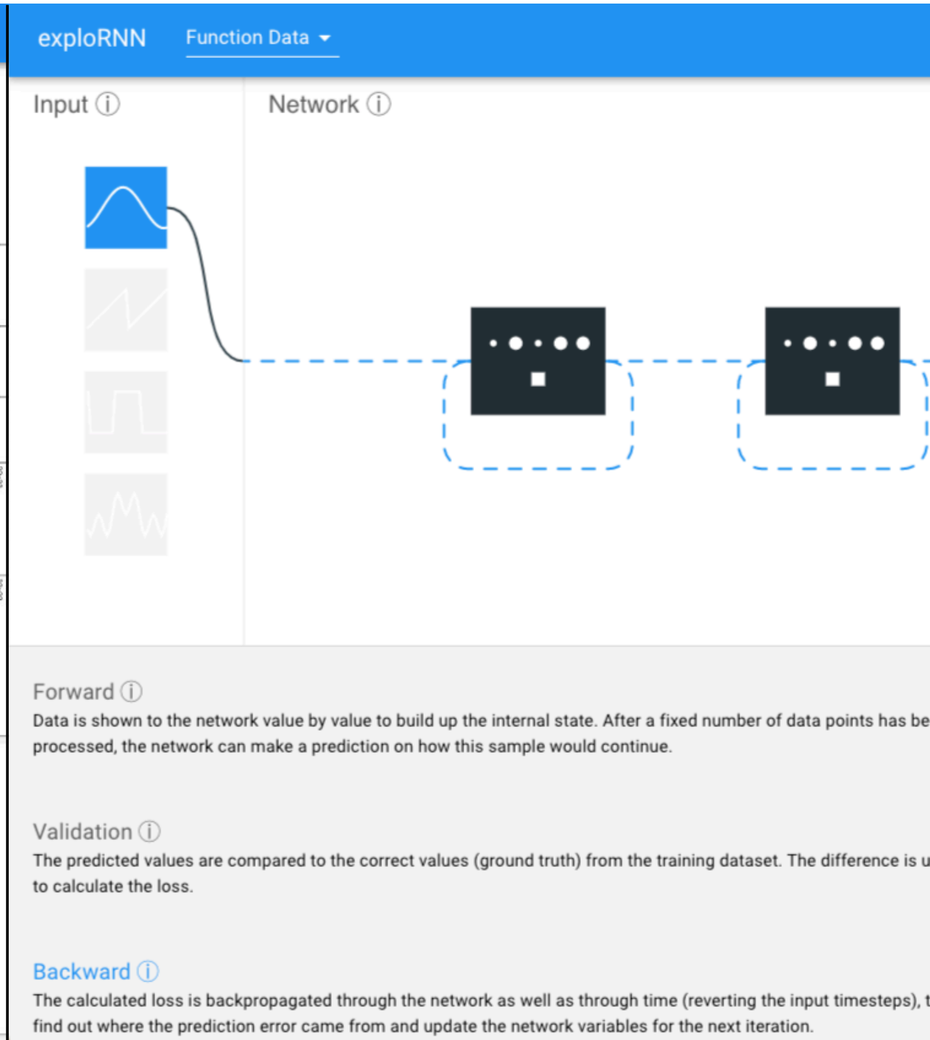
Improving research communication through automatic visualization-generation.

Quality Assurance

Communication



Education



How Are ML Interfaces Used Today?

Interviews with 9 Apple ML practitioners

What data reporting tools do you use?

What tools are you lacking?

Do you use visualizations?

Where do you store your data?

What would your ideal workflow be?

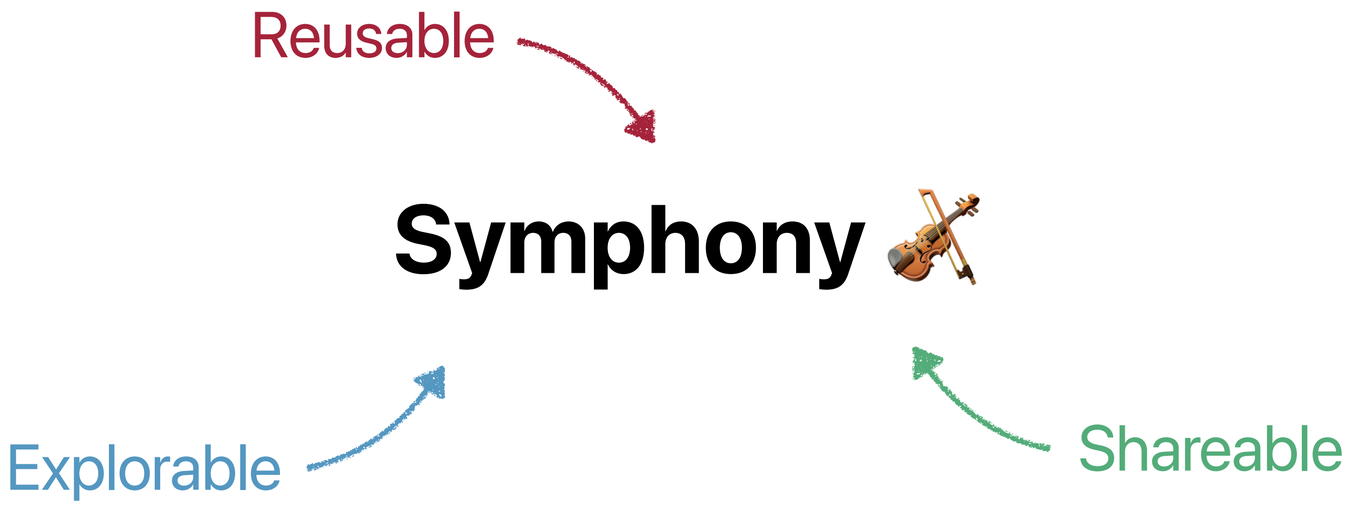
How are ML interfaces used today?

Ad-hoc tools and analyses

Limitations of existing ML interfaces

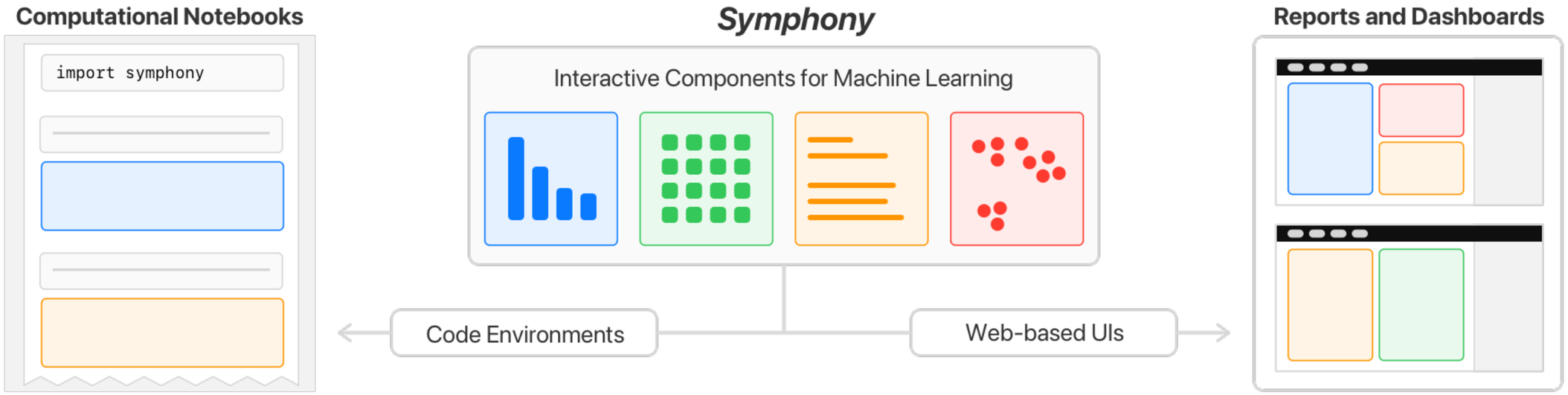
Lack of communication between stakeholders

ML interface framework with **reusable**, **explorable**, and **shareable** visualizations.


Reusable
Symphony
Explorable
Shareable

Symphony 🎻

Modular, Interactive, and Shareable Components



Cifar 10 Symphony

In [1]:
```
import pandas as pd

from symphony import Symphony
from symphony_list import SymphonyList
from symphony_summary import SymphonySummary
from symphony_familiarity import SymphonyFamiliarity
from symphony_scatterplot import SymphonyScatterplot
from symphony_duplicates import SymphonyDuplicates
from symphony_hierarchical_confusion_matrix import SymphonyHierarchicalConfusionMatrix
from symphony_fairvis import SymphonyFairVis
from symphony_markdown import SymphonyMarkdown
```

In [2]:
```
DATA_PATH = 'cifar/'
df = pd.read_parquet('table.parquet')
```

In [3]:
```
df.head()
```

Out[3]:

	id	label	dataset	output	duplicates_embedding	projection_embedding_x	projection_embedding_y	familiarity_embedding	splitFamiliarity_embedding_byAttr_label	splitFamiliarity_embedding
0	train/ship/image194.png	ship	train	ship	-1	0.409252	9.471022	24.430313	{'airplane': 12.975905777838438, 'automobile':...	{'test': 21.55013580
1	train/deer/image610.png	deer	train	deer	-1	15.026260	4.096645	5.905596	{'airplane': -29.83306951371854, 'automobile':...	{'test': 6.17149782(
2	train/truck/image2259.png	truck	train	truck	-1	-0.561973	5.591904	19.569766	{'airplane': 10.05607537711247, 'automobile': ...	{'test': 19.8754948
3	train/cat/image2898.png	cat	train	cat	-1	6.223715	0.279402	16.975622	{'airplane': -31.109959897103085, 'automobile'...	{'test': 10.8236136
4	test/dog/image447.png	dog	test	horse	-1	8.475064	3.314383	19.254085	{'airplane': 12.839210831518892, 'automobile':...	{'test': 25.41012778

Create and Show Visualizations

We can now explore the individual report widgets!

In [ ]:
```
report = Symphony(df, files_path='cifar/')
```

In [ ]:
```
report.widget(SymphonyMarkdown, page="Overview", width="M", content=open("README.md").read())
```

In [ ]:
```
report.widget(SymphonySummary, page="Overview", width="M")
```

In [ ]:
```
report.widget(SymphonyList, page="Overview")
```

In [ ]:
```
report.widget(SymphonyDuplicates, page="Data Analysis", width="M", height="L")
```

Description

Edit Download

Overview

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

Dataset Collection

The original tiny images dataset was collected by researchers at MIT and NYU. The researchers used keywords from the WordNet database to search for images on search platforms like Google, Flickr, and Altavista. They then removed perfect duplicates and images with a significant amount of white pixels that tended to be synthetic images. The images were then downsampled to 32x32 pixels.

To create the CIFAR-10 dataset, researchers at the University of Toronto took a subset of the tiny images dataset and labeled them across 10 object classes. The dataset has a total of 60,000 images with 6,000 images in each of the 10 classes.

Dataset Labeling

The dataset was labeled by paid students. The 10 labels are airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks.

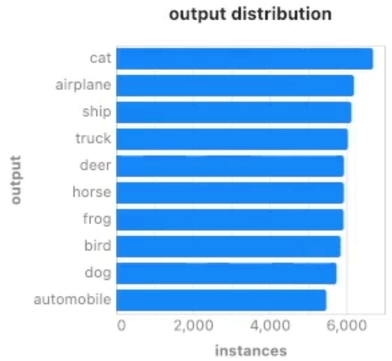
Accessing the Dataset

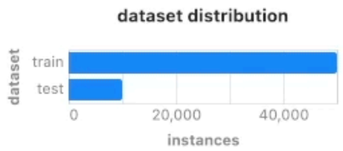
The dataset can be downloaded from the website of its oritinal authors.

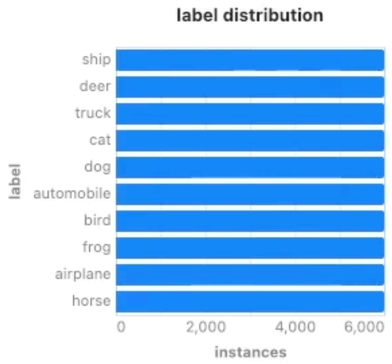
Summary Statistics

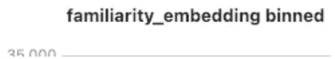
Edit Download

# of Instances: 60,000 **duplicates_embedding:** 1,434





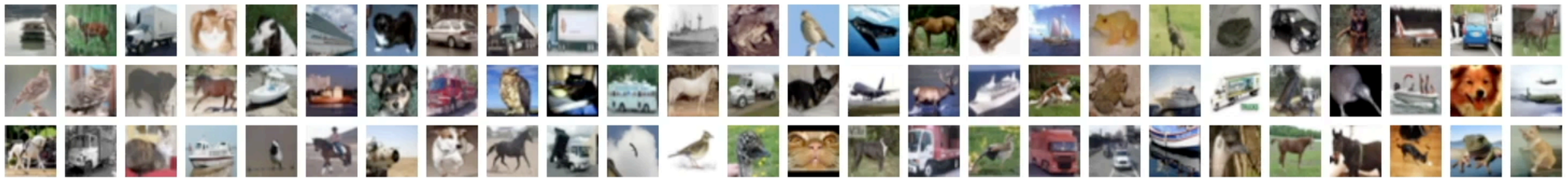




List View

Download

Instance List



Settings

Show unfiltered charts

Samples per page 150

Filter

Apply

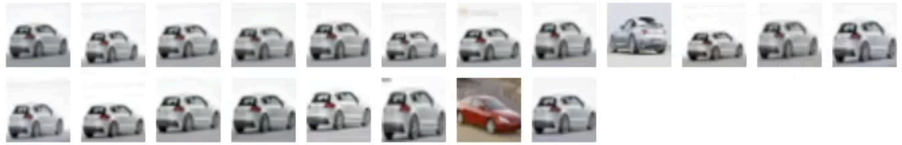
Group

Group

Duplicates

Download

Candidate Group 0



Show 15 more

Candidate Group 1

Show 8 more

Candidate Group 2

Show 2 more

Candidate Group 3

Candidate Group 4

Showing 1 to 9 of 77 results

Page 1 of 9

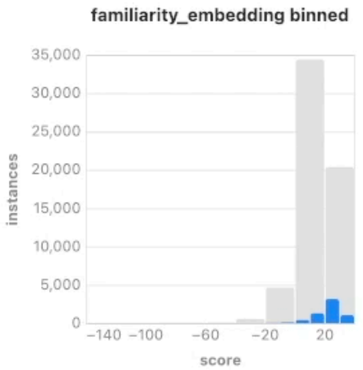
Familiarity

By Attribute

Download

Unfamiliar to Familiar

Least Familiar Instances



Mean: 22.44, STD: 8.86

Settings

Show unfiltered charts

Samples per page 150

Filter

```
d.label == 'automobile'
```

Clear Apply

Group

Group

Hierarchical Confusion Matrix

Download

Color probabilities

Dimensions

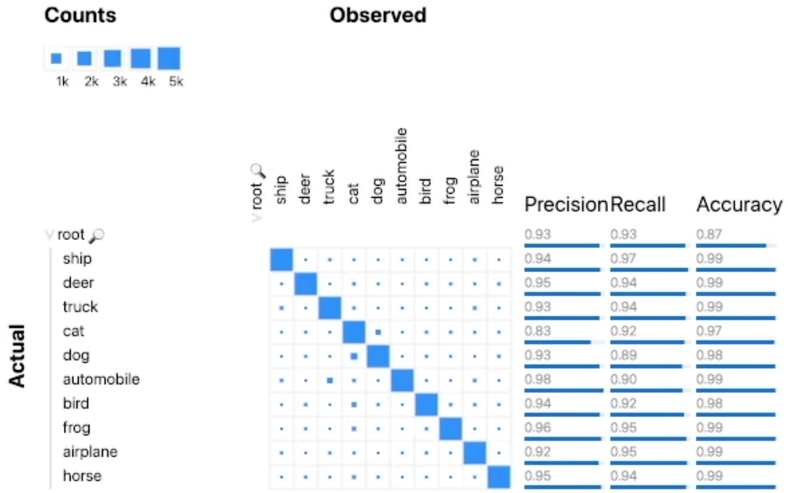
root ◁ ▷

Shelf Enable and disable different dimensions of the data. The order of dimension defines the nesting level.

All dimensions are already in use.

Where Condition the confusion matrix on the value of a given label.

Hover over cells to show more information.

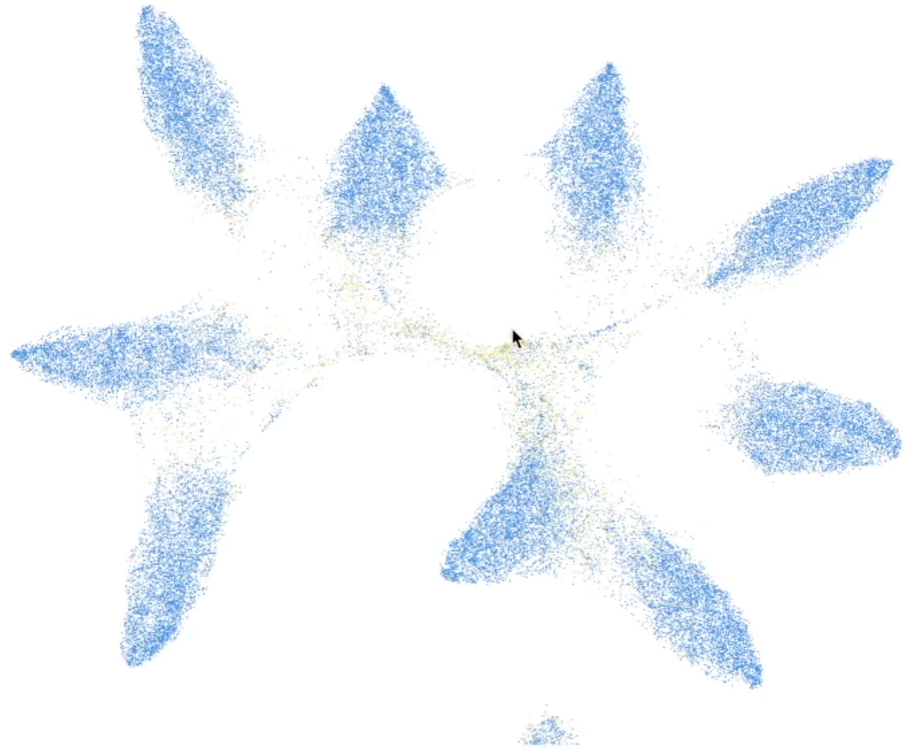


Scatterplot

Category column: correct

Download

Double-click to recenter. Shift-click and drag to lasso-select.



FairVis

Label column: label

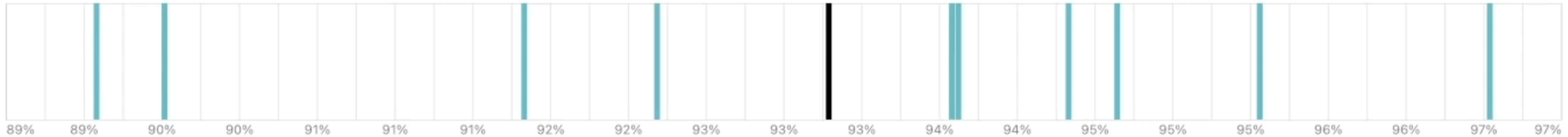
Prediction column: output

Plot type: Strip Plot

Download

Accuracy

Accuracy



Settings

Show unfiltered charts

Samples per page 150

Filter

Apply

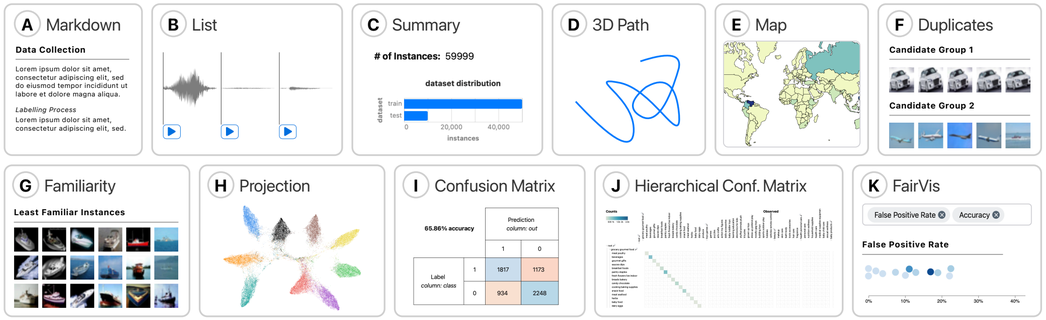
Group

label

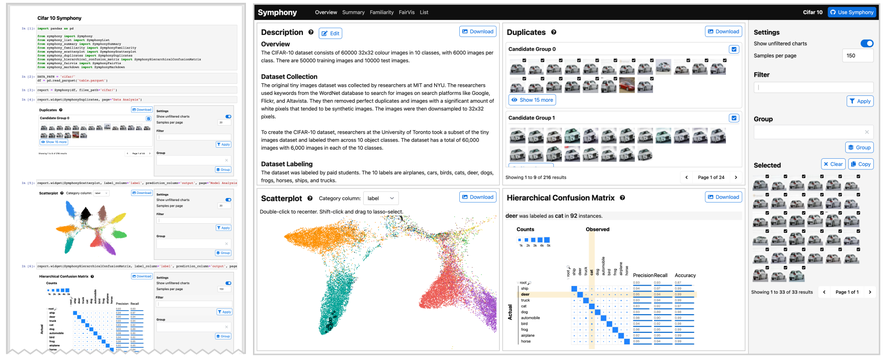
Clear Group

ship (6,000) deer (6,000)
truck (6,000) cat (6,000)
dog (6,000) automobile (6,000)
bird (6,000) frog (6,000)
airplane (6,000) horse (6,000)

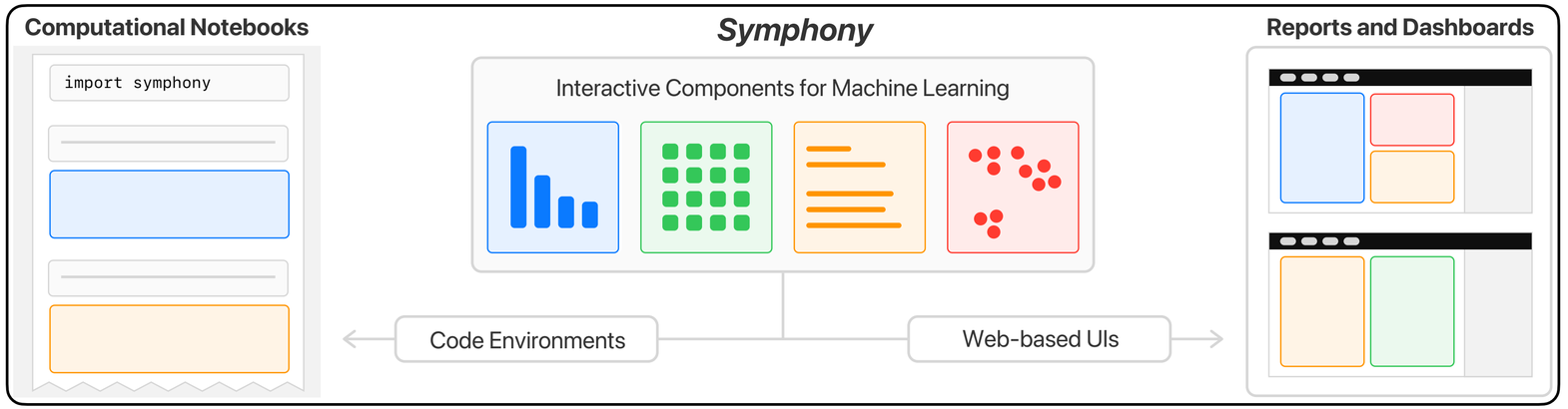
Reusable visualization components



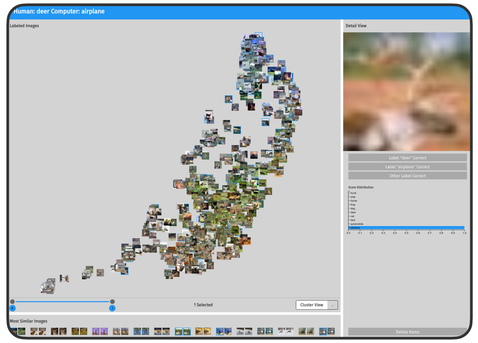
Across multiple environments



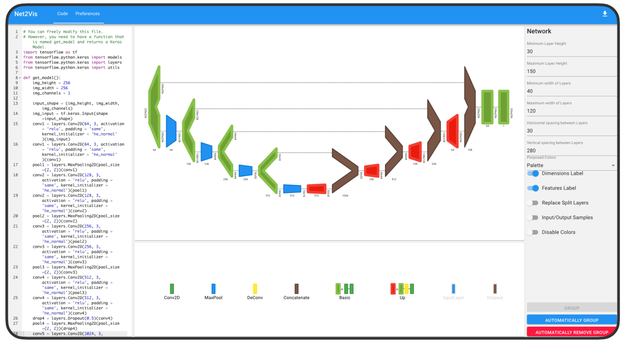
Shared organizational understanding of data and models encourages the creation of accurate, responsible, and robust AI products.



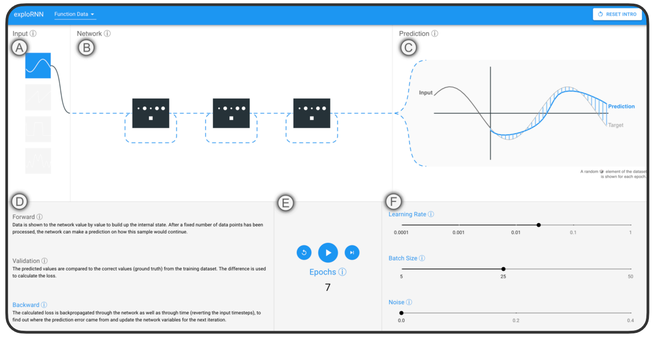
Bäuerle and Cabrera et al. 2022, CHI
Symphony: Composing Interactive Interfaces for Machine Learning



Bäuerle et al. 2020, CGF
Classifier-Guided Visual Correction of Noisy Labels for Image Classification Tasks



Bäuerle et al. 2021, TVCG
Net2Vis - A Visual Grammar for Automatically Generating Publication-Tailored CNN Architecture Visualizations



Bäuerle et al. 2022, TVC
exploRNN: Understanding Recurrent Neural Networks through Visual Exploration

Introspection Technique

Visualization Interface

1 Bäuerle et al. 2020, CGF
Classifier-Guided Visual Correction of Noisy Labels for Image Classification Tasks

Bäuerle et al. 2021, TVCG
Net2Vis - A Visual Grammar for Automatically Generating Publication-Tailored CNN Architecture Visualizations 2

Bäuerle et al. 2022, TVC
exploRNN: Understanding Recurrent Neural Networks through Visual Exploration 3

Bäuerle and Cabrera et al. 2022, CHI
Symphony: Composing Interactive Interfaces for Machine Learning 4

Introspection Technique

Visualization Interface

1 Bäuerle et al. 2020, CGF
Classifier-Guided Visual Correction of Noisy Labels for Image Classification Tasks

Bäuerle et al. 2021, TVCG
Net2Vis - A Visual Grammar for Automatically Generating Publication-Tailored CNN Architecture Visualizations 2

Bäuerle et al. 2022, TVC
exploRNN: Understanding Recurrent Neural Networks through Visual Exploration 3

Bäuerle and Cabrera et al. 2022, CHI
Symphony: Composing Interactive Interfaces for Machine Learning 4

5 Aka et al. 2021, AIES
Measuring Model Biases in the Absence of Ground Truth

Bäuerle et al. 2022, arXiv
Visual Identification of Problematic Bias in Large Label Spaces 6

7 Bäuerle and Wexler 2020, VISxAI
What does BERT dream of?

8 Weber et. al. 2020, Journal of Microscopy
Automatic identification of crossovers in cryo-EM images of murine amyloid protein A fibrils with machine learning

Bäuerle and van Onzenoodt et. al. 2022, CGF
Where did my Lines go? Visualizing Missing Data in Parallel Coordinates 9

10 Bäuerle and Jönsson et. al. 2022, arXiv
Neural Activation Patterns (NAPs): Visual Interpretability of Learned Concepts

Model attributes

Modern architectures

Introspection

Develop methods for ML introspection and make them accessible through visualization.

Visualization

Data types, tasks & users

Beyond research projects

Visual Computing Group
Ulm University

Visualization-Based Neural Network Introspection

Alex Bäuerle

Defense, 21.12.2022