

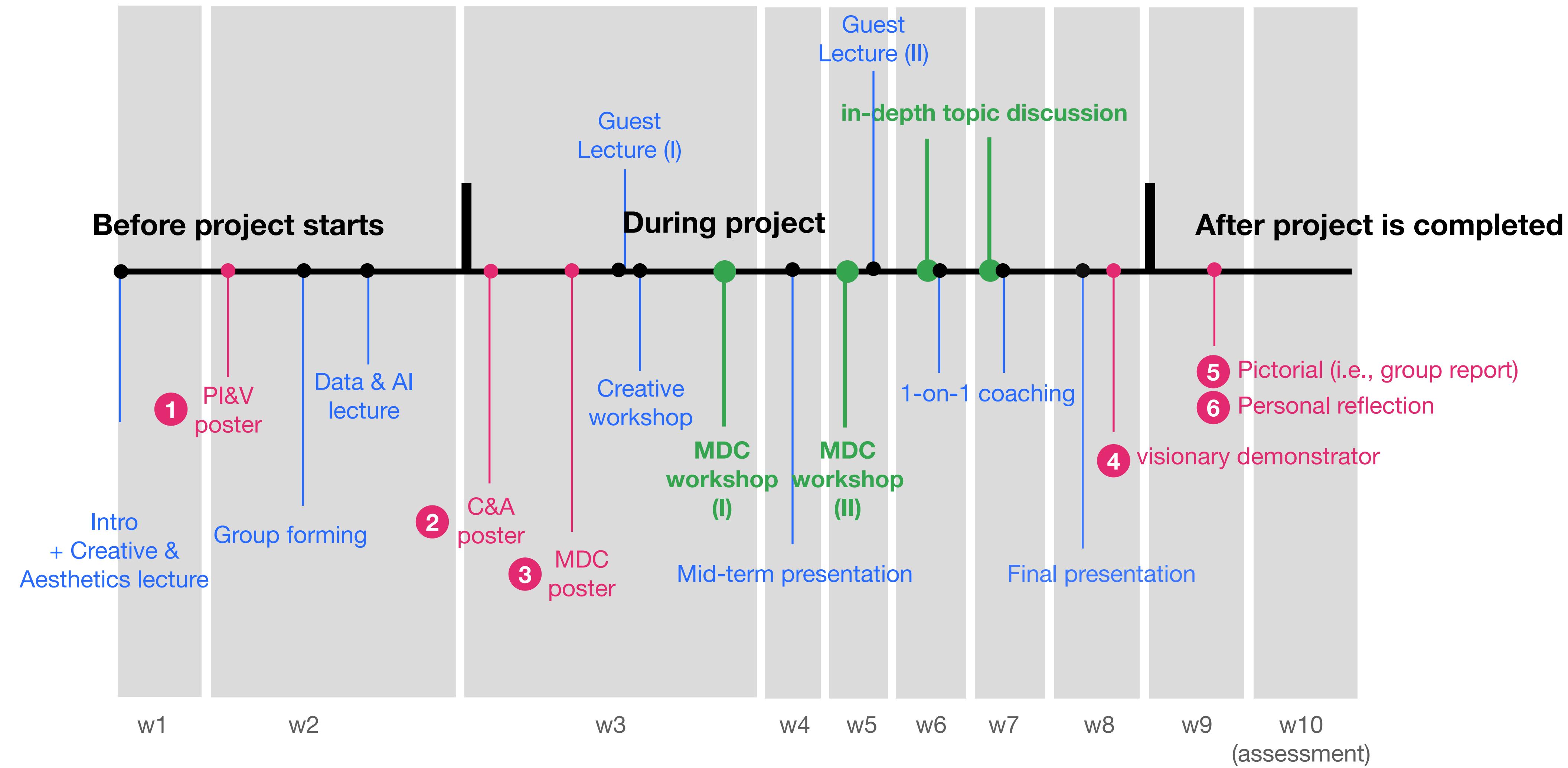
MDC Workshop (I)

Data, Algorithm & Computation, Creativity

Janet Huang

2023.03.02

2022/2023 Q3



MDC Workshops

MDC Workshop (I): Basic (3 hrs)

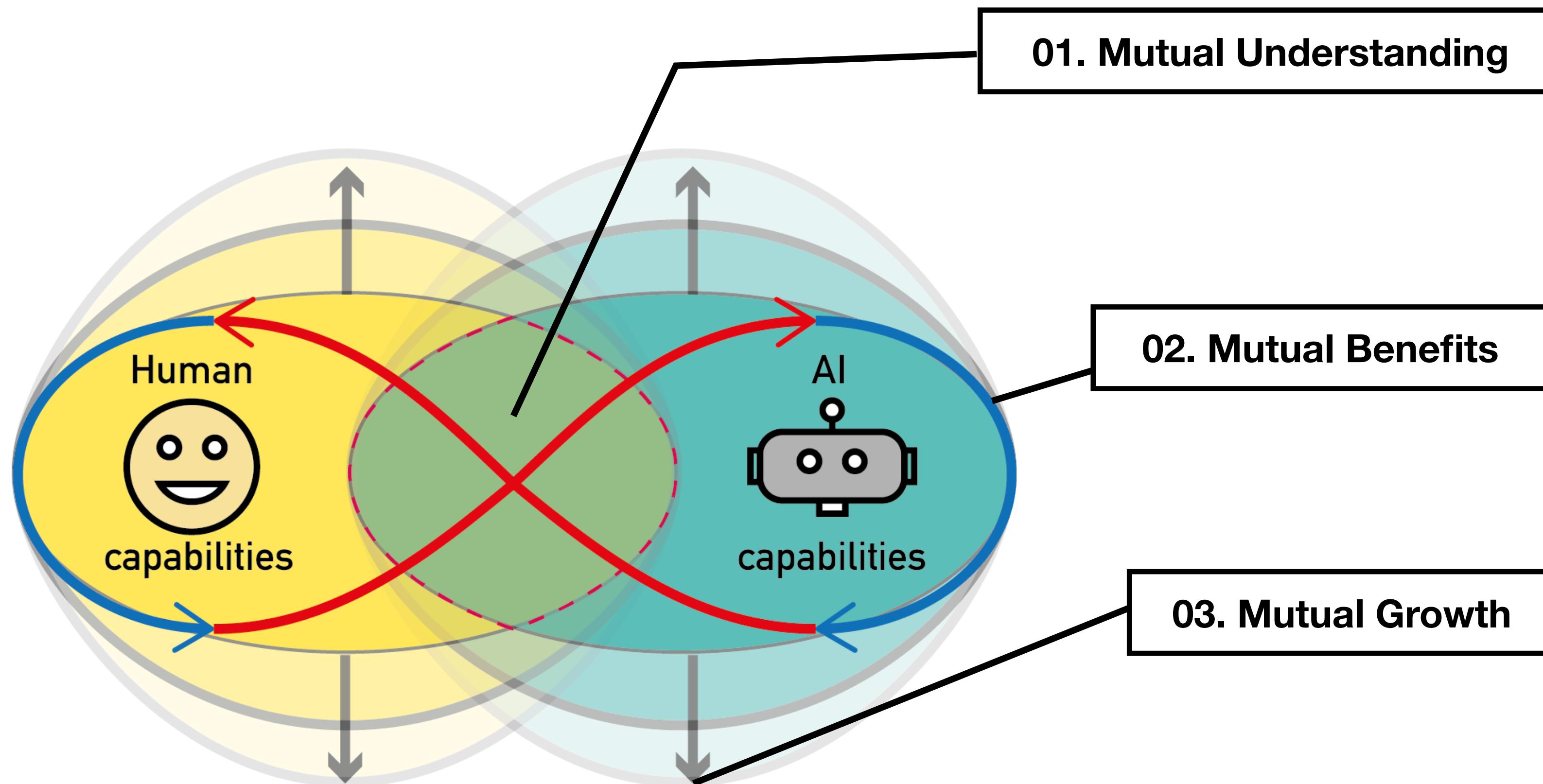
1. Introduction of thingCV (40 mins)
 - Introduction (10 mins)
 - Exercise 1: play with thingCV (20 mins)
 - Sharing (5-7 mins)
2. Collaborate with thingCV and generative AI (30 mins)
 - Introduction (5 mins)
 - Exercise 2: play with generative AI (20 mins)
 - Sharing (5 mins)
3. Build your thingCV from scratch (I) (90 mins)
 - Foundation: ml5.js, Data Foundry, observable
 - Exercise 3: object hunter
 - Exercise 4: data visualizer, data exploration

MDC Workshop (I): Advanced (2 hrs)

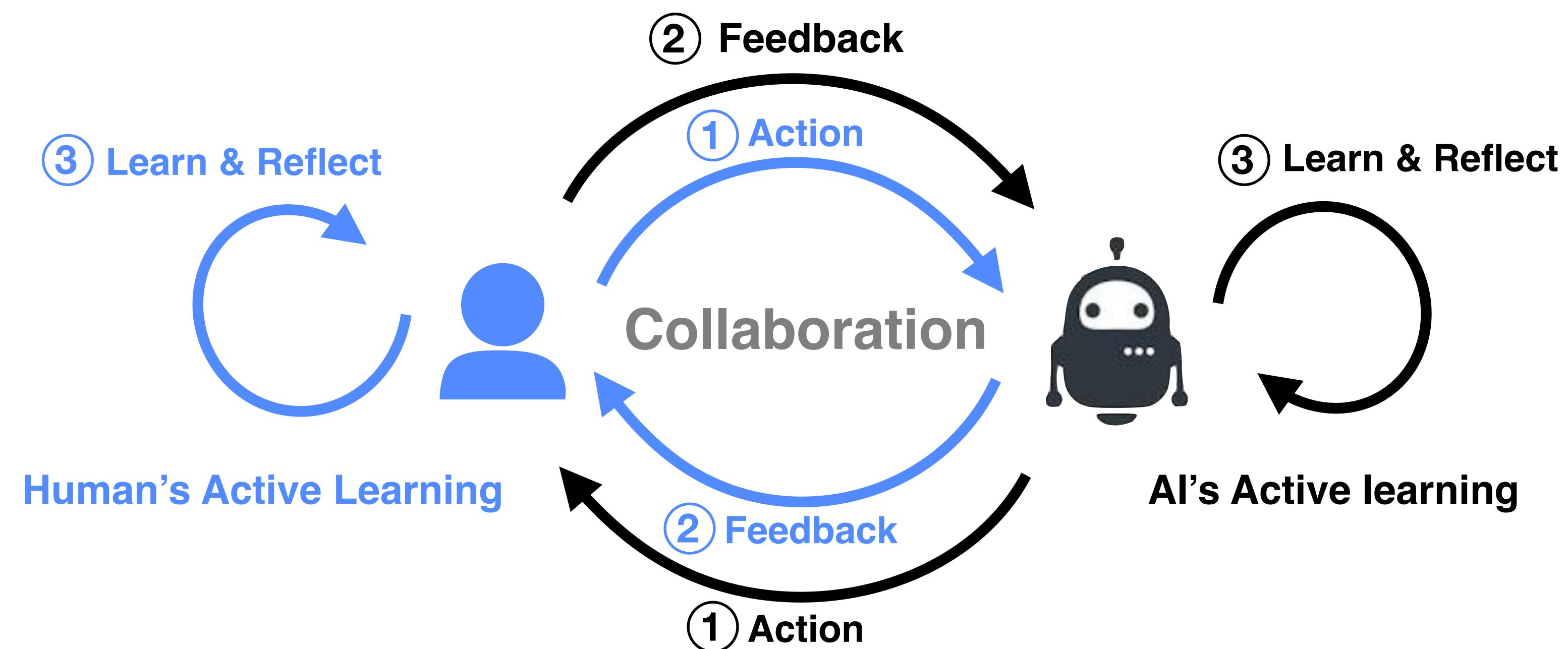
1. Build your thingCV from scratch (II): data, computation, algorithm
 - Foundation: Data Foundry, starboard, d3.js
 - Exercise 1: integration of object hunter, network builder, and data visualizer
2. Extend your thingCV with LLMs model:
 - Foundation: OpenAI api in Data Foundry
 - Exercise 2: thingCV with OpenAI api

Session I: Play with thingCV

Human-AI Co-Learning

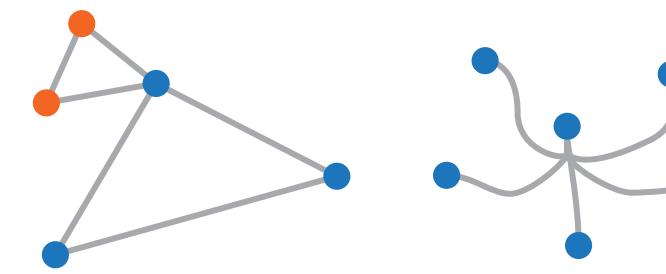


Human-AI Collaboration Framework

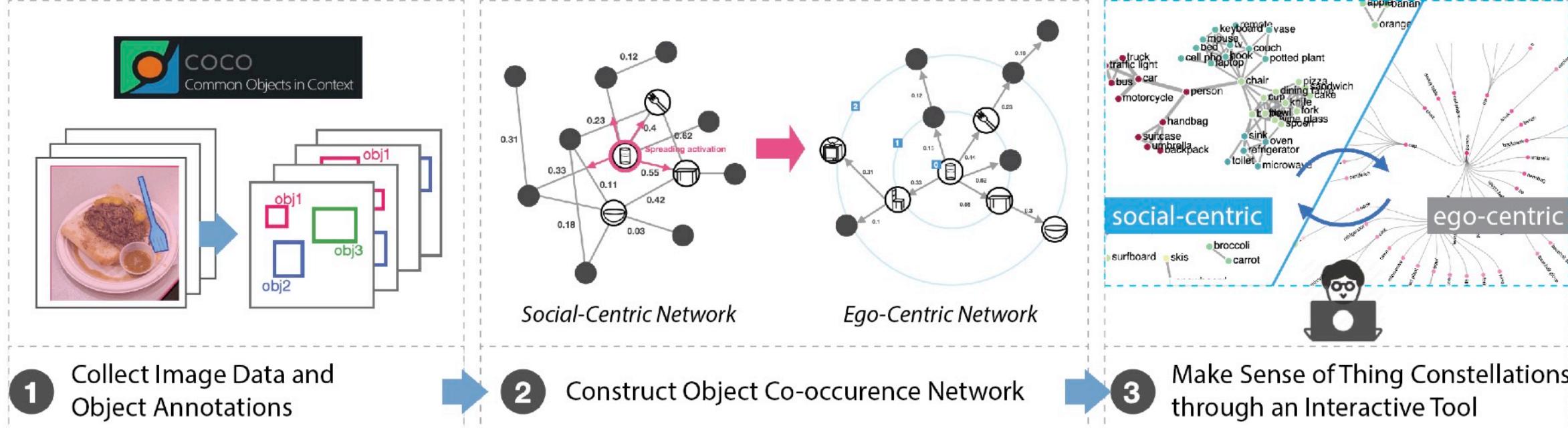


Thing Constellation Visualizer

Exploring Emergent Relationships of Everyday Objects



Data-Driven Design Exploration



Data-driven pattern + Human interpretation

Contributions

This work presents Thing Constellation Visualizer (ThingCV), a new approach and tool that empowers designers to use alternative perspectives to revisit everyday practice and contributes new insights from two workshops on in-depth understandings of emergent relationships among objects.

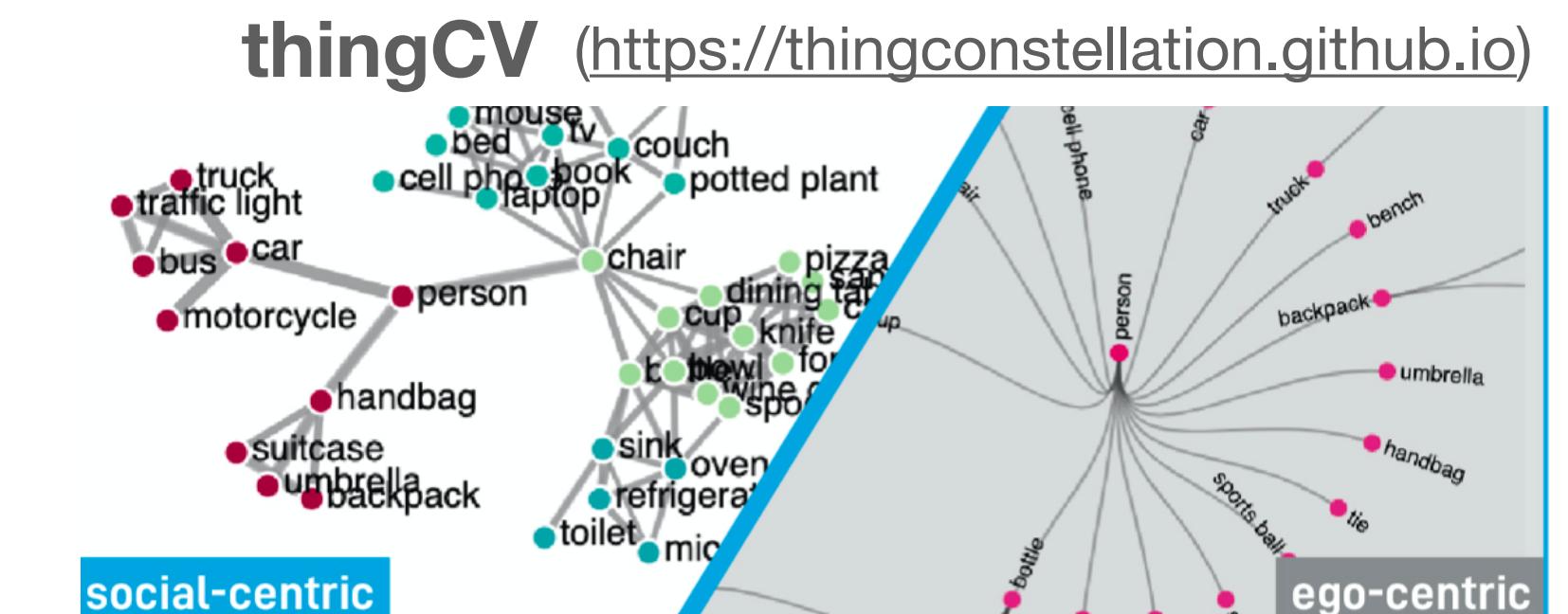
The approach, tool, and insights will contribute to the future design of IoT ecosystems.

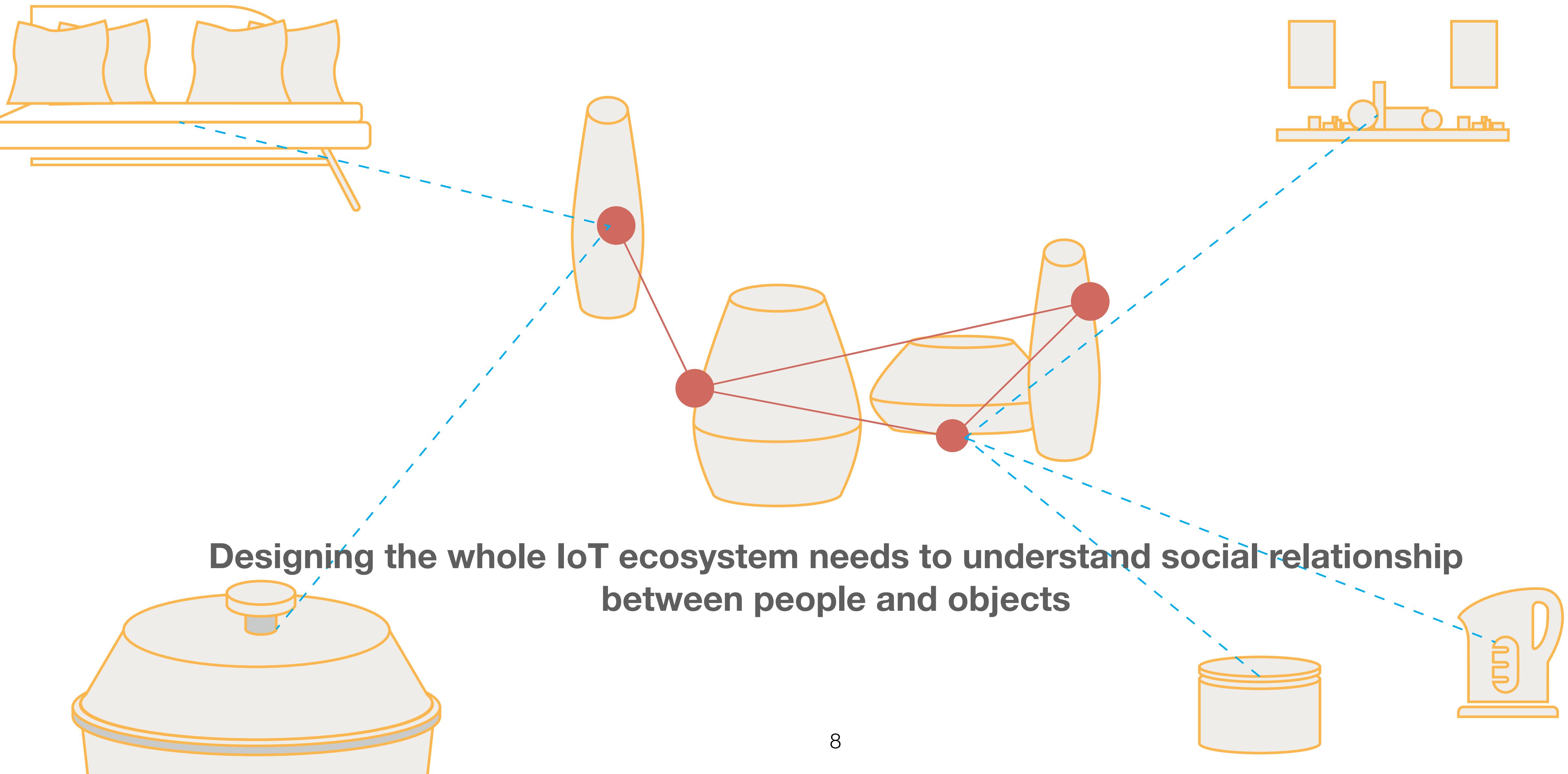


Janet Huang
TU Eindhoven

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Yi-Ching (Janet) Huang, Yu-Ting Cheng, Rung-Huei Liang, Jane Yung-jen Hsu, and Lin-Lin Chen. 2021. Thing Constellation Visualizer: Exploring Emergent Relationships of Everyday Objects. Proc. ACM Hum.-Comput. Interact. 5, CSCW2, Article 479 (October 2021), 29 pages. DOI:<https://doi.org/10.1145/3479866>





Challenges

- The emergent relationship is **hidden and barely recognized** by people. It requires **experienced designers or experts** to **extract patterns from empirical data** through an iterative sense-making process.
- As the **volume and complexity of data** increases, it is challenging even for experts to extract patterns **without appropriate support**.
- Prior work mainly focus on a single object or a few objects, limited tools support researchers to study **IoT ecosystems**.

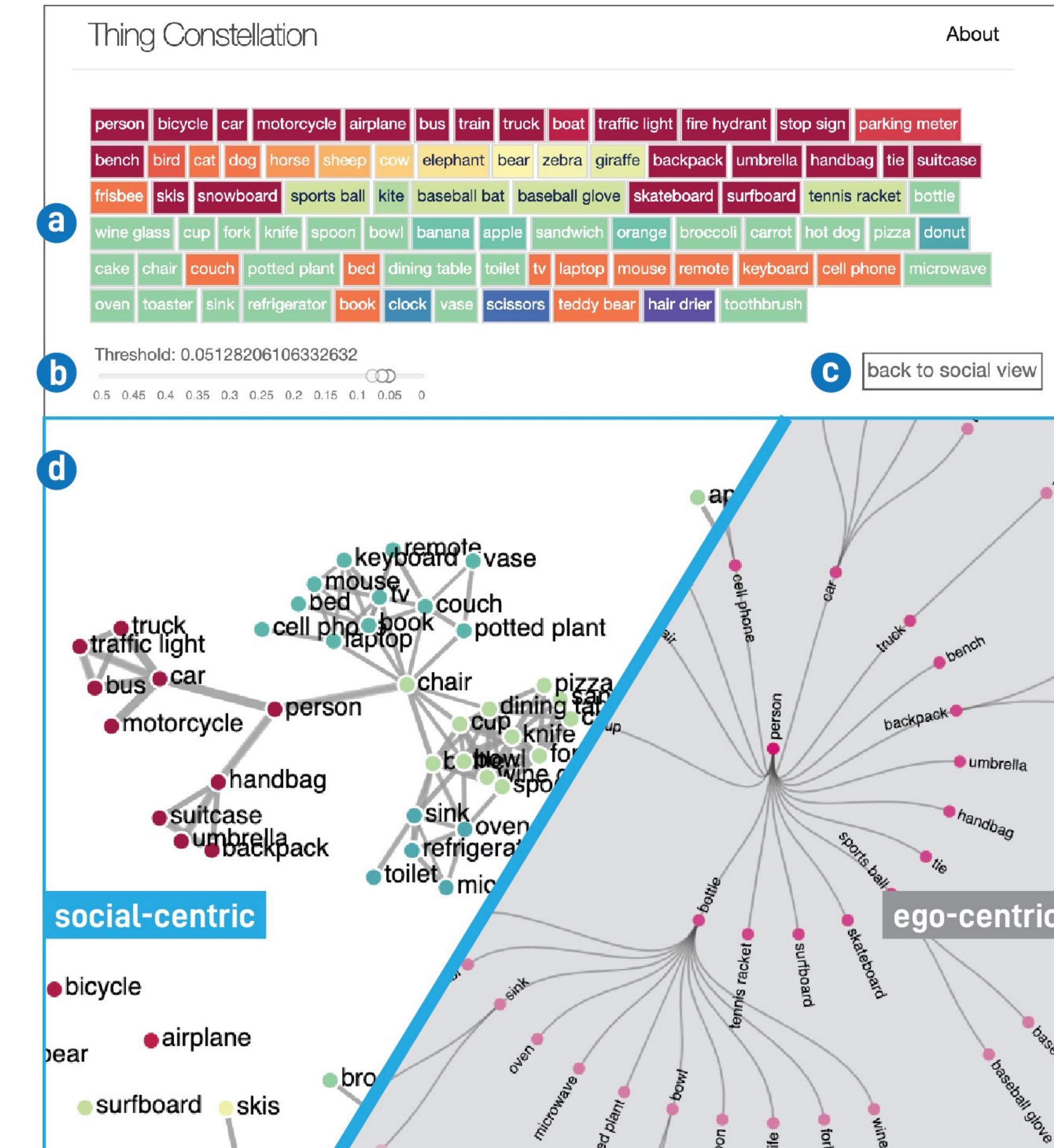
Data need further processing and need humans to intervene,
interact and interpret to make them “meaningful”

Data-driven pattern + Human interpretation

Research Goal

This work aims to develop a **novel tool with a computational approach** for supporting **researchers to understand the emergent relationships among everyday objects in practice**.

Thing Constellation Visualizer



a Object Panel

- objects in the same community are filled with the same colors.
- click any object to zoom into ego-centric view.

b Threshold Slider

- adjust threshold value to determine the density (i.e. numbers of links between nodes).

c Switch Button

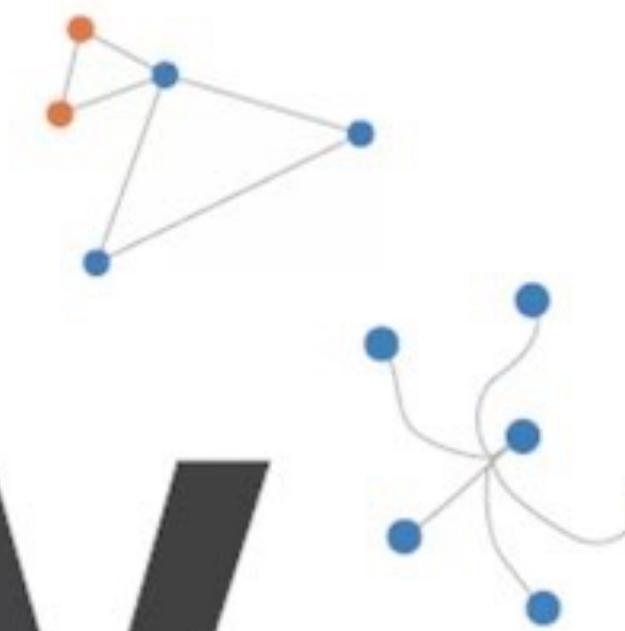
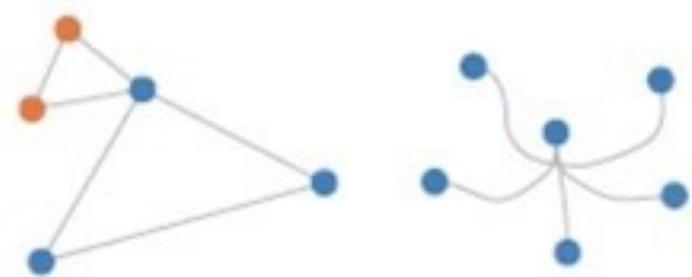
- switch two views under the same threshold value.

d Thing Constellation Visualisation

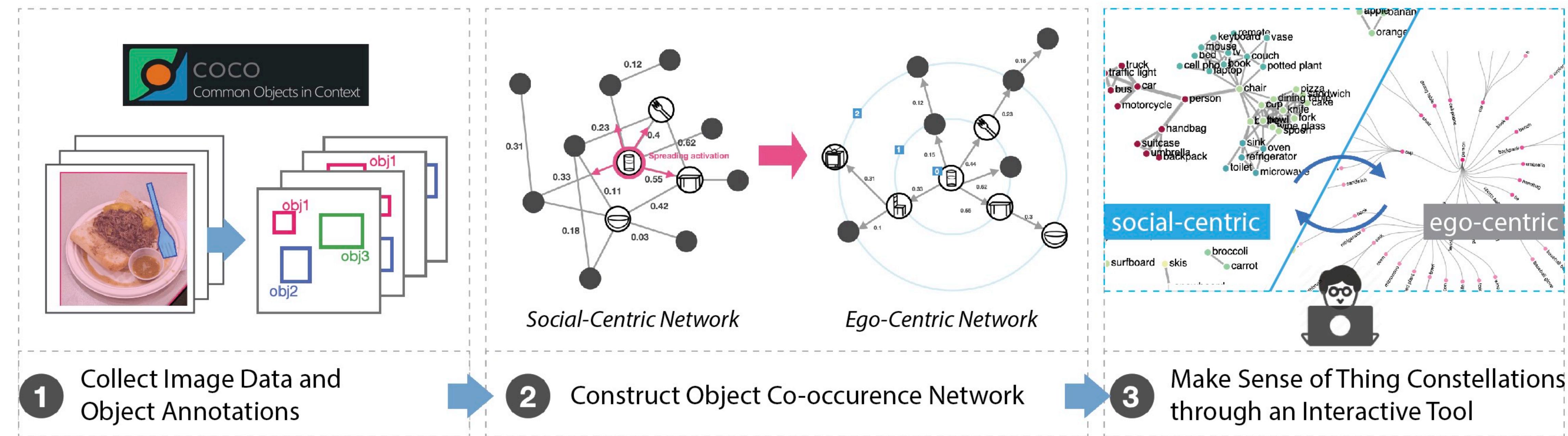
- based on the previous selections, here shows the result which can be social-centric or ego-centric constellation.

ThingCV

Thing Constellation Visualiser



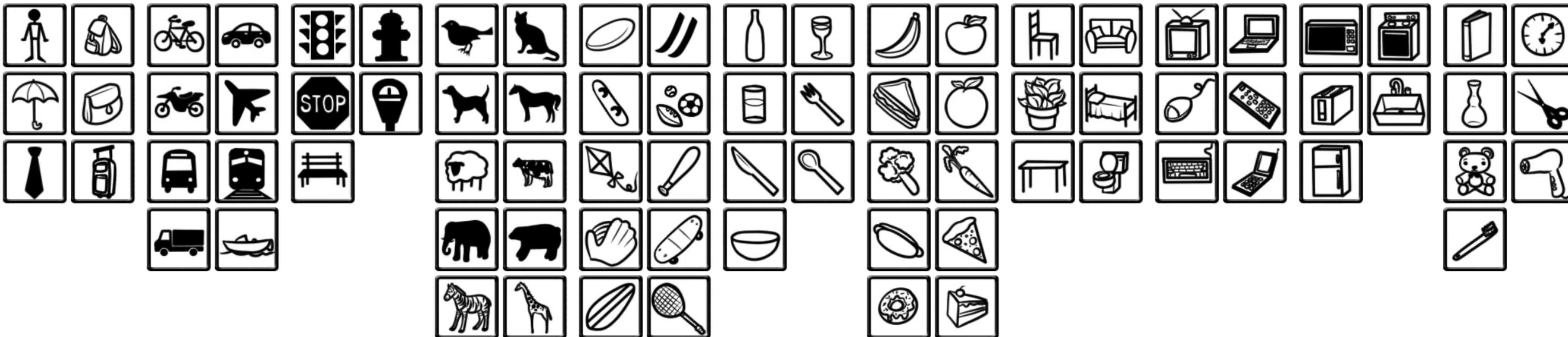
Data-Driven Design Exploration



Exercise 1: Play with thingCV

Task: Design Your AI Agents in a IoT ecosystem

Pickup one object and redesign it as an AI agent in a IoT ecosystem, and think about how this object interact with people.



Exercise 1-(a): Play with COCO dataset

The screenshot shows the COCO Explorer interface. At the top, there is a dark header bar with the COCO logo, the text "COCO Common Objects in Context", an email address "info@cocodataset.org", and a navigation menu with links "Home", "People", "Dataset", "Tasks", and "Evaluate". The "Dataset" link is highlighted with a green background.

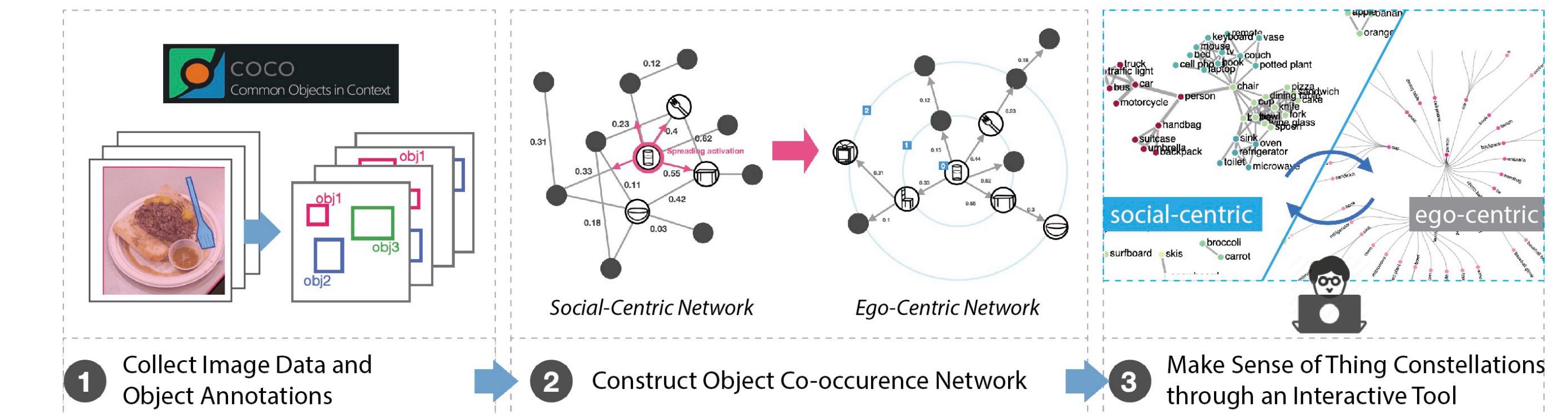
The main area is titled "COCO Explorer" and displays a message: "COCO 2017 train/val browser (123,287 images, 886,284 instances). Crowd labels not shown." Below this is a grid of 20x10 icons representing various objects and categories from the COCO dataset.

Below the grid, there is a search bar containing the text "dining table" with a clear button "x" and a "search" button. Underneath the search bar, it says "12338 results".

Below the search results, there is a row of small thumbnail images corresponding to the search term. One of these thumbnails is a large image of a room with a dining table, chairs, and a lamp, which is the result of the search.

Exercise 1-(b): Play with thingCV (20 mins)

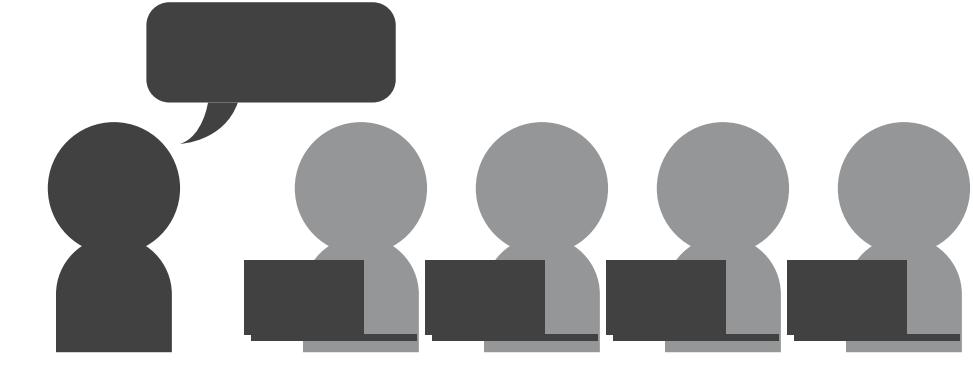
- Every group uses 20 mins to
 - play with an existing image dataset (i.e., COCO dataset)
 - use thingCV tool to explore the ecosystem of everyday objects
 - share your findings to everyone, including interesting things, weird things, supervise things, and so on.



Insights from two design workshops

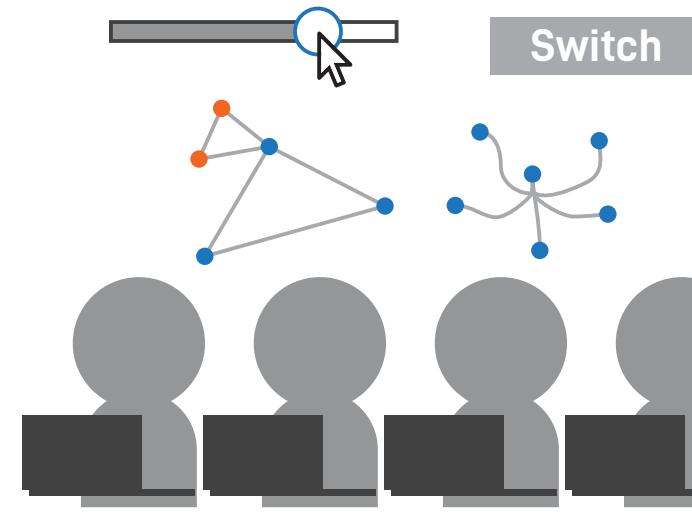
Design Workshops

1. Introduction



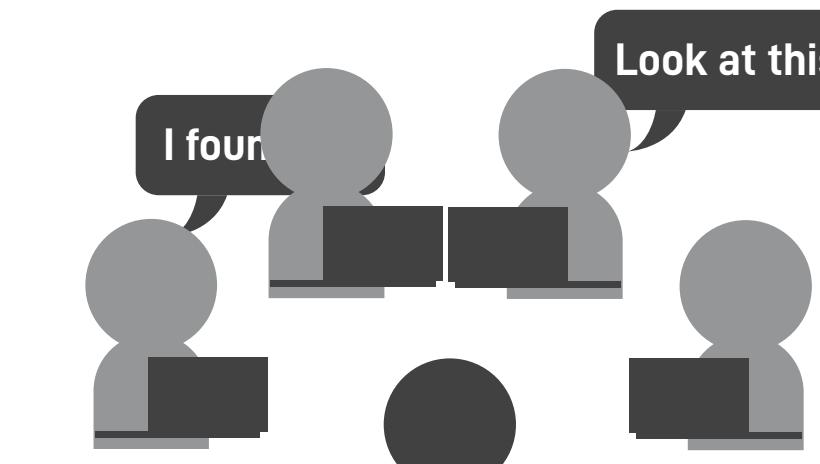
Facilitator Participants
(with their own laptops)

2. Observation



15-20mins

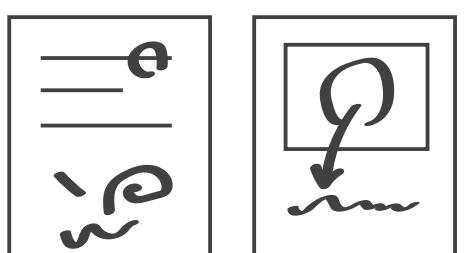
3. Group Discussion



50-60mins

4. Data Analysis

4-1. Raw Data



Audio Recording
Observation notes & Drawing
Screenshot of the constellation

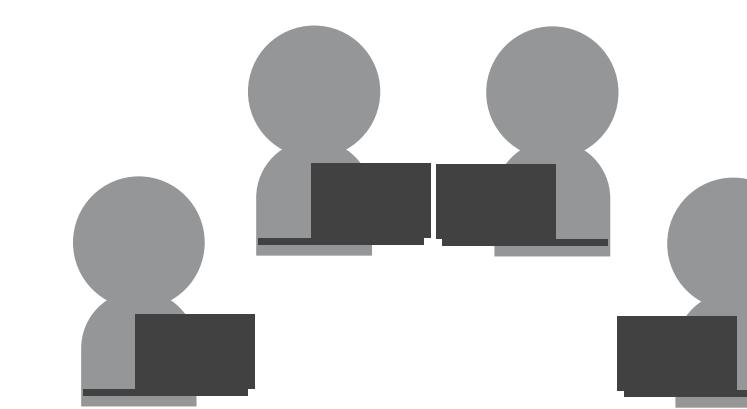
4-2. Annotating Raw Data



#tool usage process
& experience
#data interpretation
#general reflection

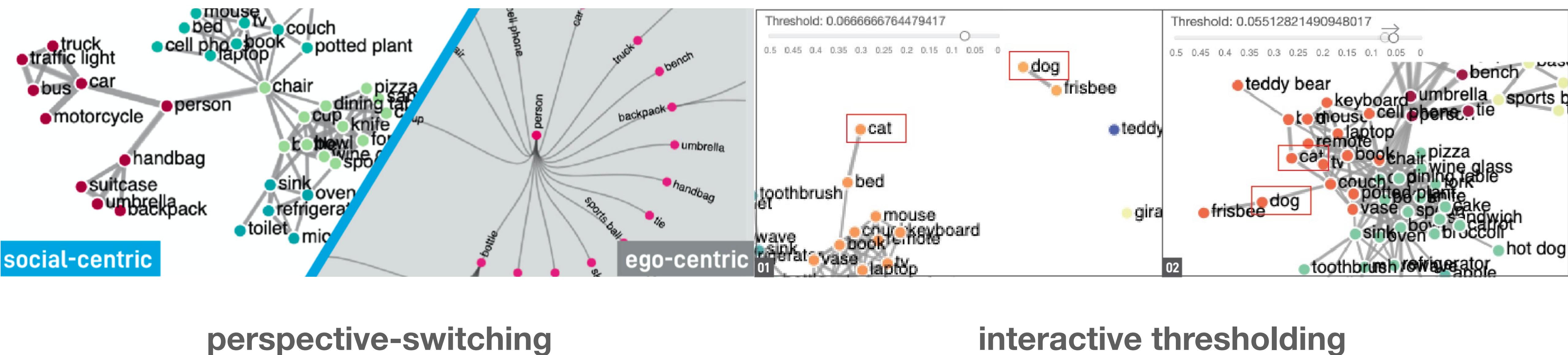
Researchers

4-3. Developing Themes & Sub-Themes



Experts (design x computer science)

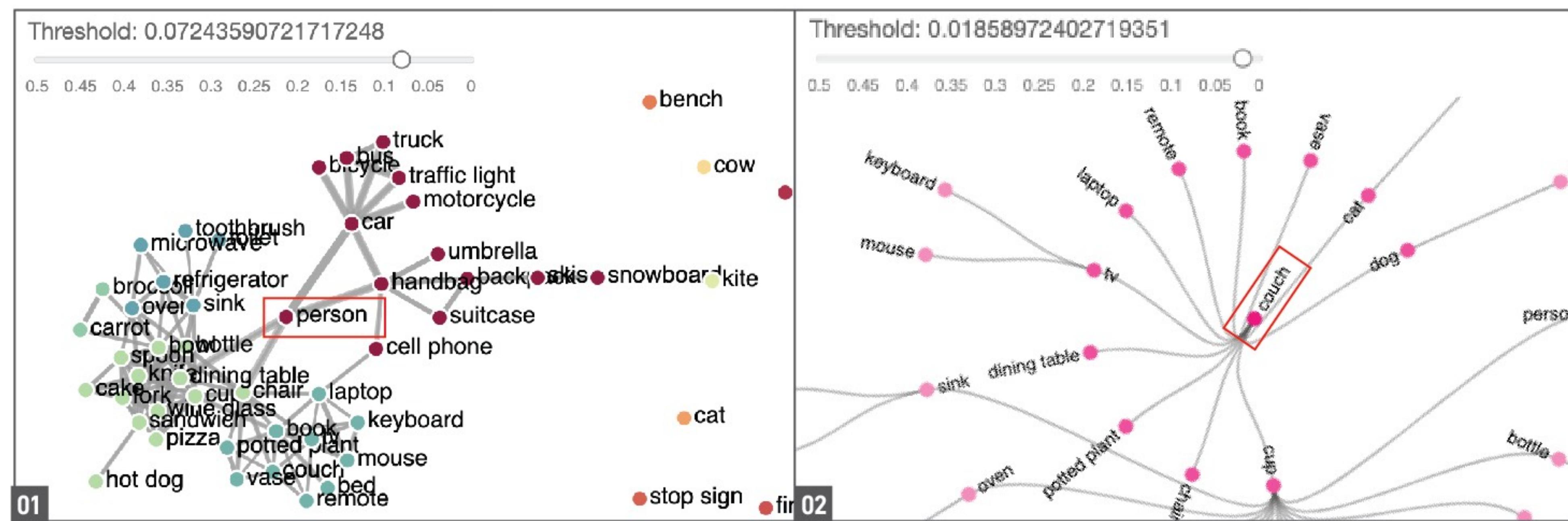
Finding 1: Observing Thing Constellation Flexibly



perspective-switching

interactive thresholding

Finding 2: Projecting Social Quality onto Things

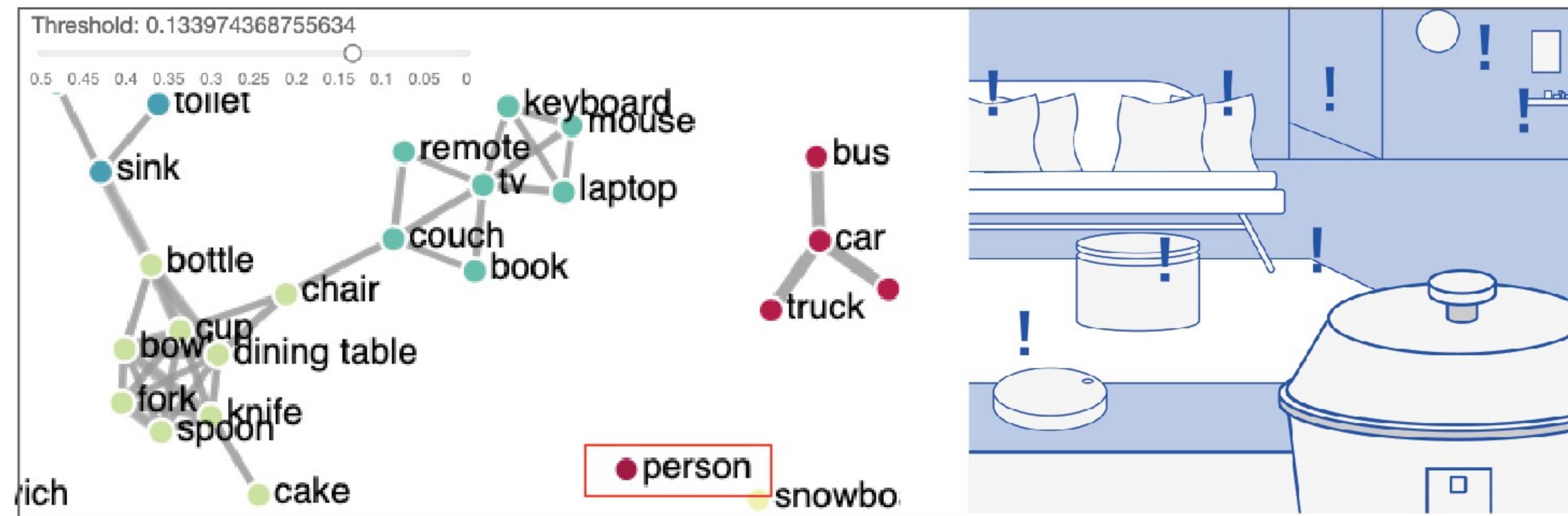


Person and couch are sociable and busy

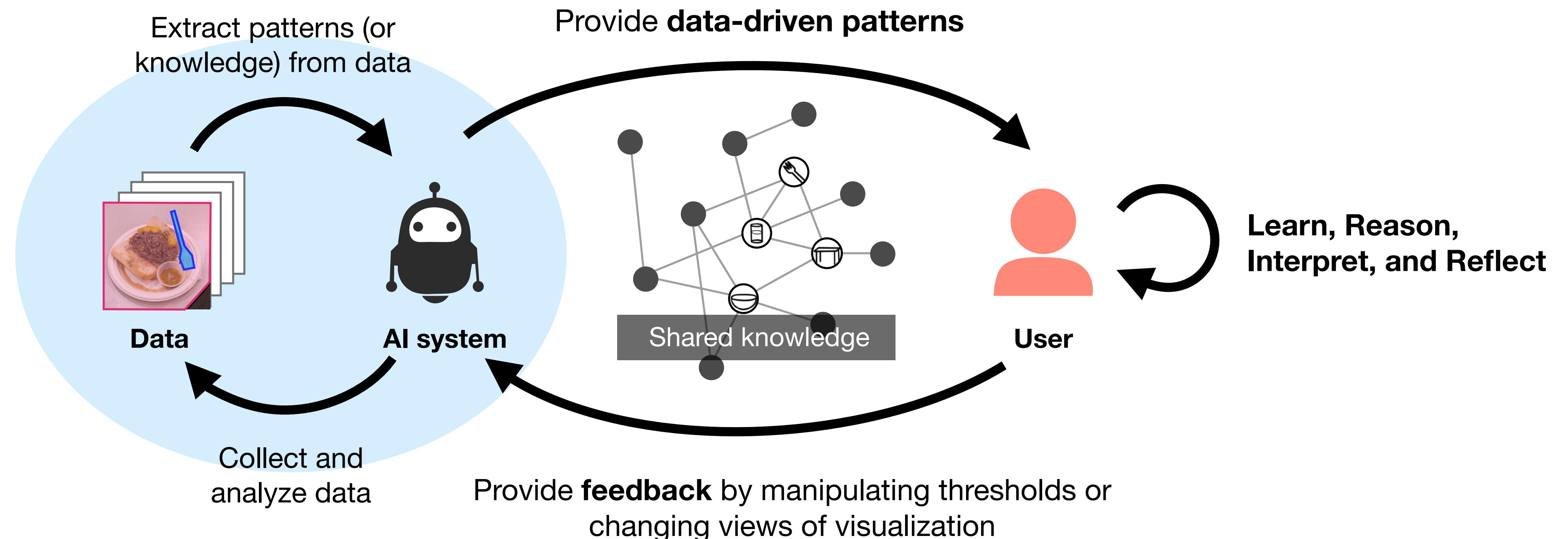
Finding 3: Discovering Emergent Diverse Contexts via Object Clusters



Finding 4: Changing Perspectives to Revisit Everyday Practice



Start imagining a different world in which everyday objects cooperate together



Data-driven pattern + **Human interpretation**

Example Datasets for thingCV

 COCO
Common Objects in Context info@cocodataset.org

Home People **Dataset** Tasks Evaluate

COCO Explorer

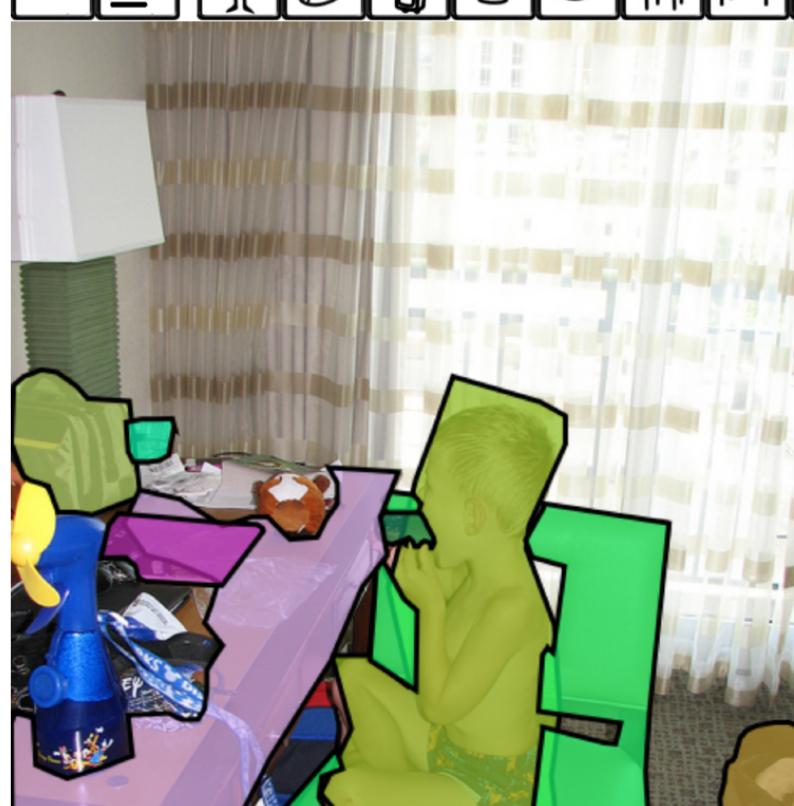
COCO 2017 train/val browser (123,287 images, 886,284 instances). Crowd labels not shown.



dining table

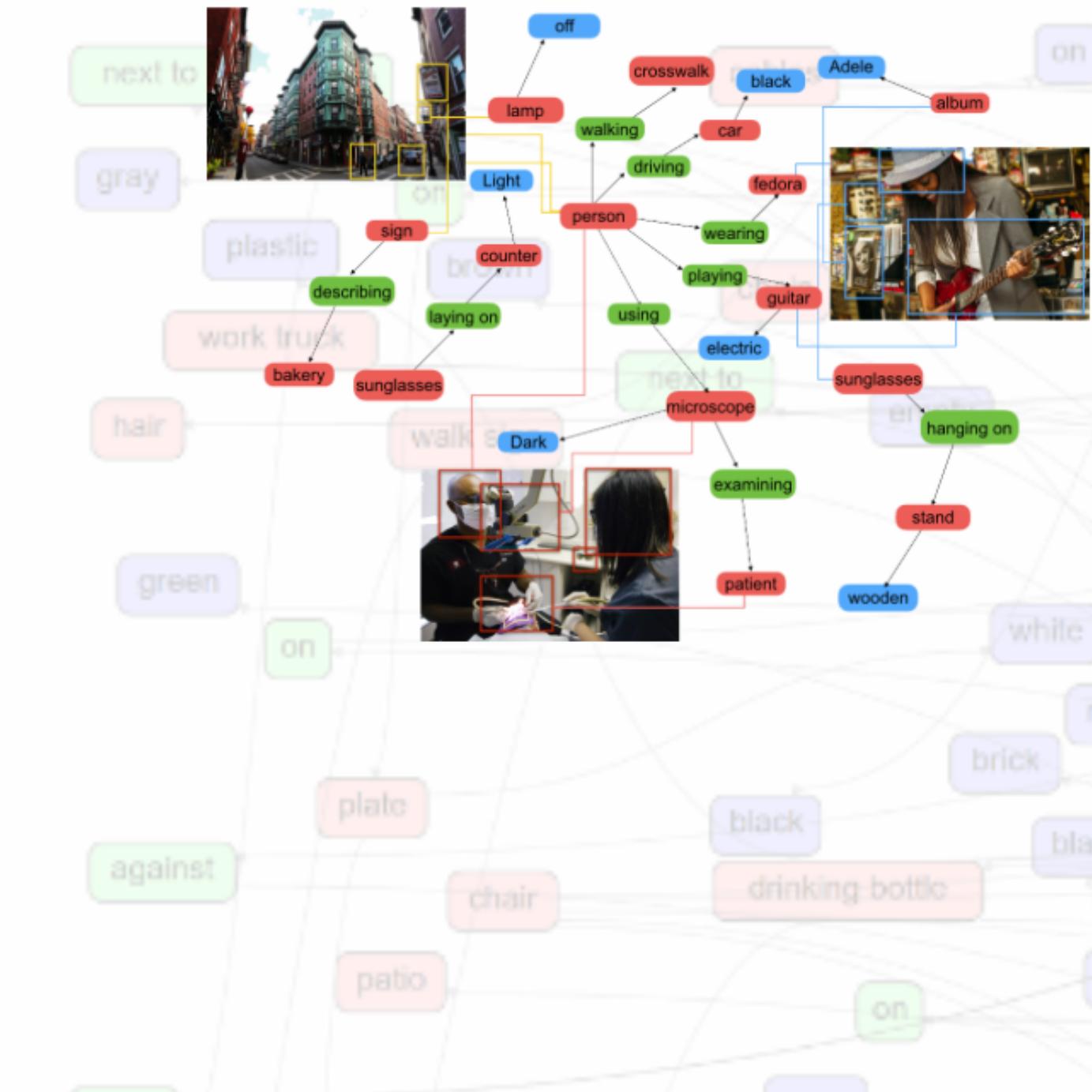
12338 results

URL          



 VISUALGENOME [About](#) [Download](#) [Data Analysis](#) [Paper](#) [Explore](#) [Internal](#)

Explore



Visual Genome is a dataset, a knowledge base, an ongoing effort to connect structured image concepts to language.

Explore our data:
throwing frisbee, helping, angry

108,077 Images
5.4 Million Region Descriptions
1.7 Million Visual Question Answers
3.8 Million Object Instances
2.8 Million Attributes
2.3 Million Relationships
Everything Mapped to Wordnet Synsets

Read our paper.

Inspired by Big Data

use existing datasets or public api to get big data

The screenshot shows the Kaggle Datasets homepage. On the left, there's a sidebar with navigation links like Home, Competitions, Datasets, Code, Discussions, Courses, and More. The main area has a search bar at the top. Below it, a section titled "Datasets" encourages users to explore, analyze, and share quality data. It features a "New Dataset" button and a "Trending Datasets" section with five cards. The cards include "COVID-19 Vaccine Articles" by Brandon Conrady, "Kaggle Algorithm" by Emirhan BULUT, "Emotion Prediction with Semi Supervised Learning" by EMIRHAN BULUT, "Github Organizations - Social Network Analysis" by Anshul Mehta, and "10 Years of Climate Science Denial on..." by Rick T. Each card shows the dataset title, author, last updated time, file count, and size.

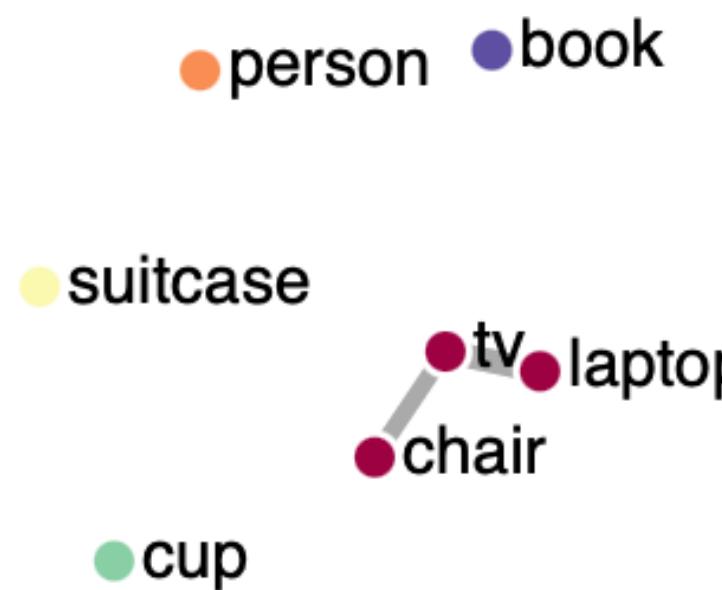
<https://www.kaggle.com/datasets>

The screenshot shows the GitHub repository page for "public-apis/public-apis". The repository has 3.5k watchers, 22k forks, and 190k stars. It's a public repository with 4,525 commits. The code tab is selected, showing files like README.md, CONTRIBUTING.md, LICENSE, and .github. The repository is described as a "collective list of free APIs for use in software and web development". It includes sections for Public APIs, Status (with green status indicators for tests), and Languages (Python 96.8%, Shell 3.2%). Contributors are listed with 1,269 contributors and 1,258 active ones.

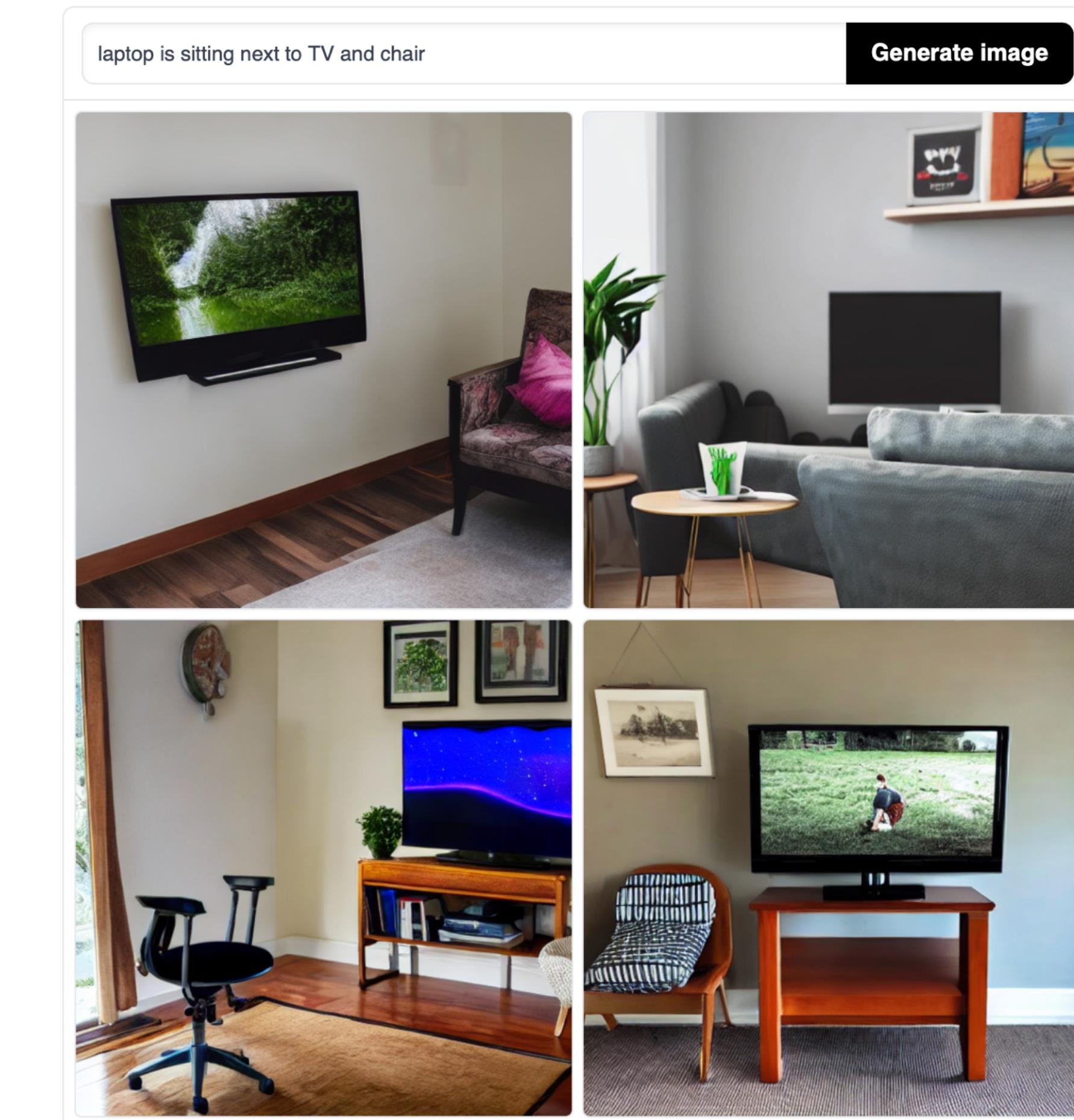
<https://github.com/public-apis/public-apis>

Session II: Collaborate with thingCV and Generative AI

Session II: Collaborate with thingCV and Generative AI



laptop is sitting next to TV and chair



Stable Diffusion: Text-to-Image AI

<https://huggingface.co/spaces/stabilityai/stable-diffusion>

The image shows a user interface for generating images using text prompts. On the left, there are two examples of generated landscapes: a mountain range at sunset and a valley with a bridge. Below them are two more landscape images, one with a prominent peak and another with a bridge over water. In the center, there is a configuration panel with sliders for Width (768), Height (512), Cfg Scale (7), Steps (50), Number of Images (1), and Sampler (K-LMS). A table titled "Compute Costs & Generation Counts Explained" provides a breakdown of costs for different image sizes and generation steps:

Steps	512x512	512x768	512x1024	768x768	768x1024	1024x1024
10	0.2	0.5	0.8	0.9	1.3	1.9
25	0.5	1.2	1.9	2.3	3.3	4.7
50	1.0	2.4	3.8	4.6	6.6	9.4
75	1.5	3.6	5.7	6.9	9.9	14.1
100	2.0	4.8	7.6	9.2	13.2	18.8
150	3.0	7.2	11.4	13.8	19.8	28.2

On the right side, a grid of generated images includes: an astronaut playing a piano in space; a white unicorn in a field; a seal wearing a top hat and leather jacket; a bear in an astronaut suit on a planet; a robot holding a drink; a futuristic cityscape; a green lizard shooting a beam of light; a red mushroom in a city; and an astronaut riding a tiger.

Image source: [link1, link2]

Introducing ChatGPT

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

[Try ChatGPT ↗](#)[Read about ChatGPT Plus](#)

There are 7 objects in the context, including person, book, suitcase, cup, tv, chair, and laptop. The tv, chair and laptop have a close relationship. Write a short scenario to describe how 7 objects interact with each other.



One evening, a person was sitting on a chair in their living room, with a book on their lap and a cup of tea on the nearby table. They were enjoying the peacefulness of the moment until they noticed that the TV wasn't turning on. They put down the book and got up from the chair to check the TV, but it seemed that it was unplugged. They quickly went to retrieve their laptop from the suitcase in the corner of the room, plugged it into the TV, and started to watch their favorite show.



As they were watching, they noticed that the sound was too low. They reached for the cup of tea to take a sip but accidentally knocked it over, spilling tea all over the table and the book. Frustrated, they paused the show and got up from the chair to clean up the mess. While cleaning, they noticed that their laptop battery was running low and they needed to plug it in. They went back to the suitcase to retrieve the charger and plugged it into the laptop.

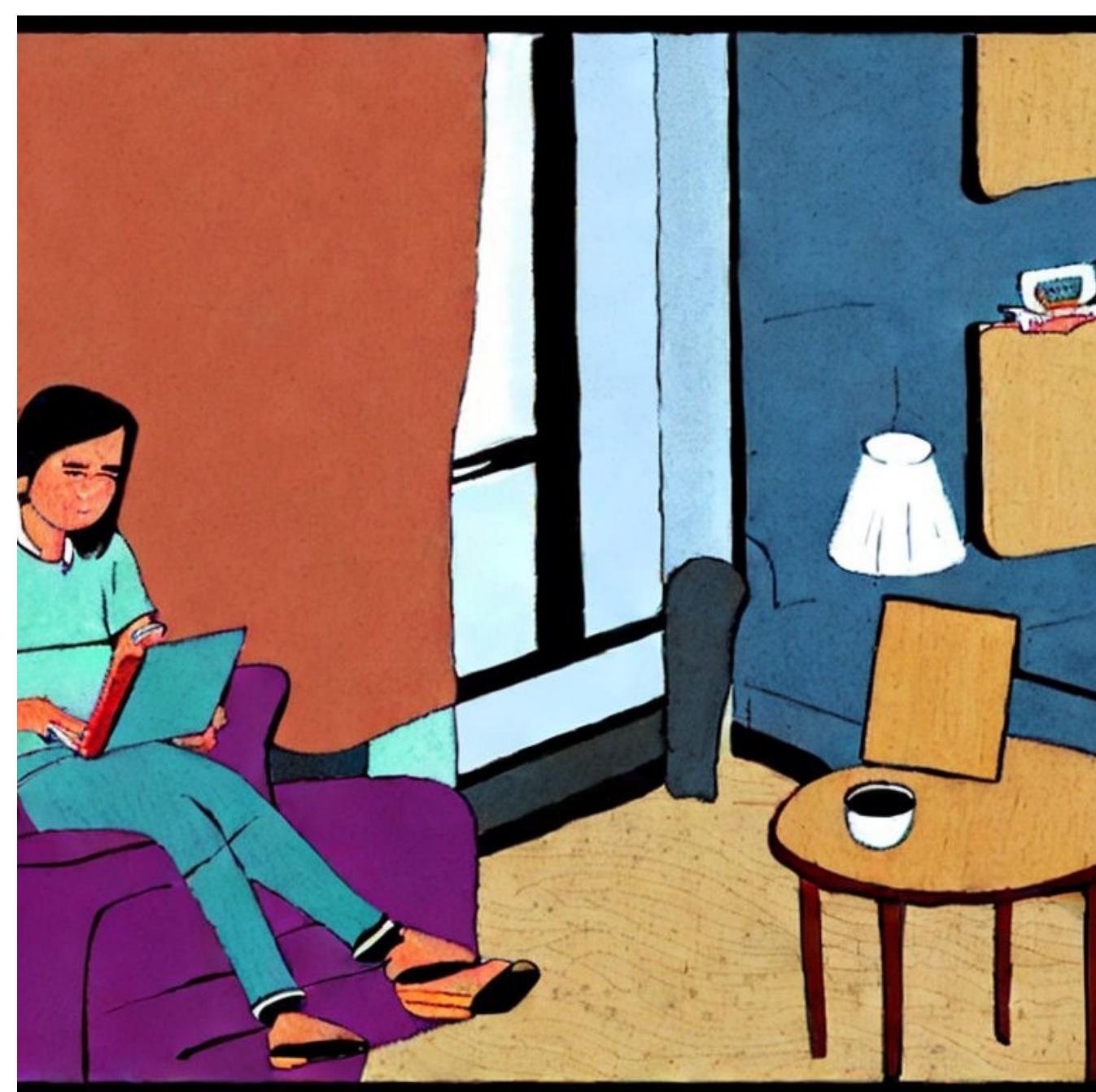
After cleaning up the mess, they sat back down on the chair and resumed watching the show on the TV, now with better sound. They were grateful for the close relationship between the TV, chair, and laptop, which provided them with the entertainment they needed to relax after a long day.

Entertainment Hub

A person is sitting on a chair in their living room, with a book on their lap and a cup of tea on the nearby table. Suddenly, they notice that the TV isn't turning on. They get up from the chair to check the TV and find that it's unplugged. They retrieve their laptop from the suitcase in the corner of the room, plug it into the TV, and start to watch their favorite show.

As they watch, they notice that the sound is too low. They accidentally knock over the cup of tea while reaching for it, spilling tea all over the table and the book. Frustrated, they pause the show and get up from the chair to clean up the mess. While cleaning, they notice that their laptop battery is running low and they need to plug it in. They retrieve the charger from the suitcase and plug it into the laptop.

After cleaning up the mess, they sit back down on the chair and resume watching the show on the TV, now with better sound. They feel grateful for the close relationship between the TV, chair, and laptop, which provide them with the entertainment they need to relax after a long day.



A person is sitting on a chair in their living room, with a book on their lap and a cup of tea on the nearby table. Suddenly, they notice that the TV isn't turning on. They get up from the chair to check the TV and find that it's unplugged. They retrieve their laptop from the suitcase in the corner of the room, plug it into the TV, and start to watch their favorite show.



As they watch, they notice that the sound is too low. They accidentally knock over the cup of tea while reaching for it, spilling tea all over the table and the book. Frustrated, they pause the show and get up from the chair to clean up the mess. While cleaning, they notice that their laptop battery is running low and they need to plug it in. They retrieve the charger from the suitcase and plug it into the laptop.

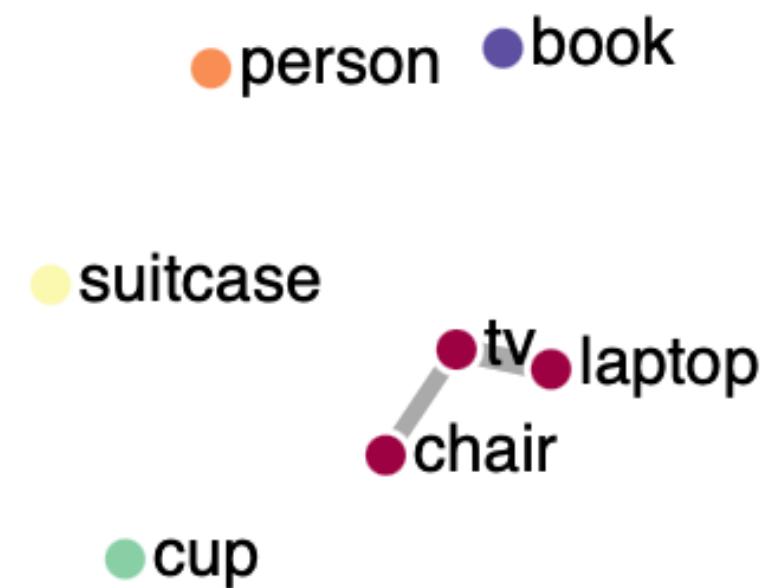


After cleaning up the mess, they sit back down on the chair and resume watching the show on the TV, now with better sound. They feel grateful for the close relationship between the TV, chair, and laptop, which provide them with the entertainment they need to relax after a long day.

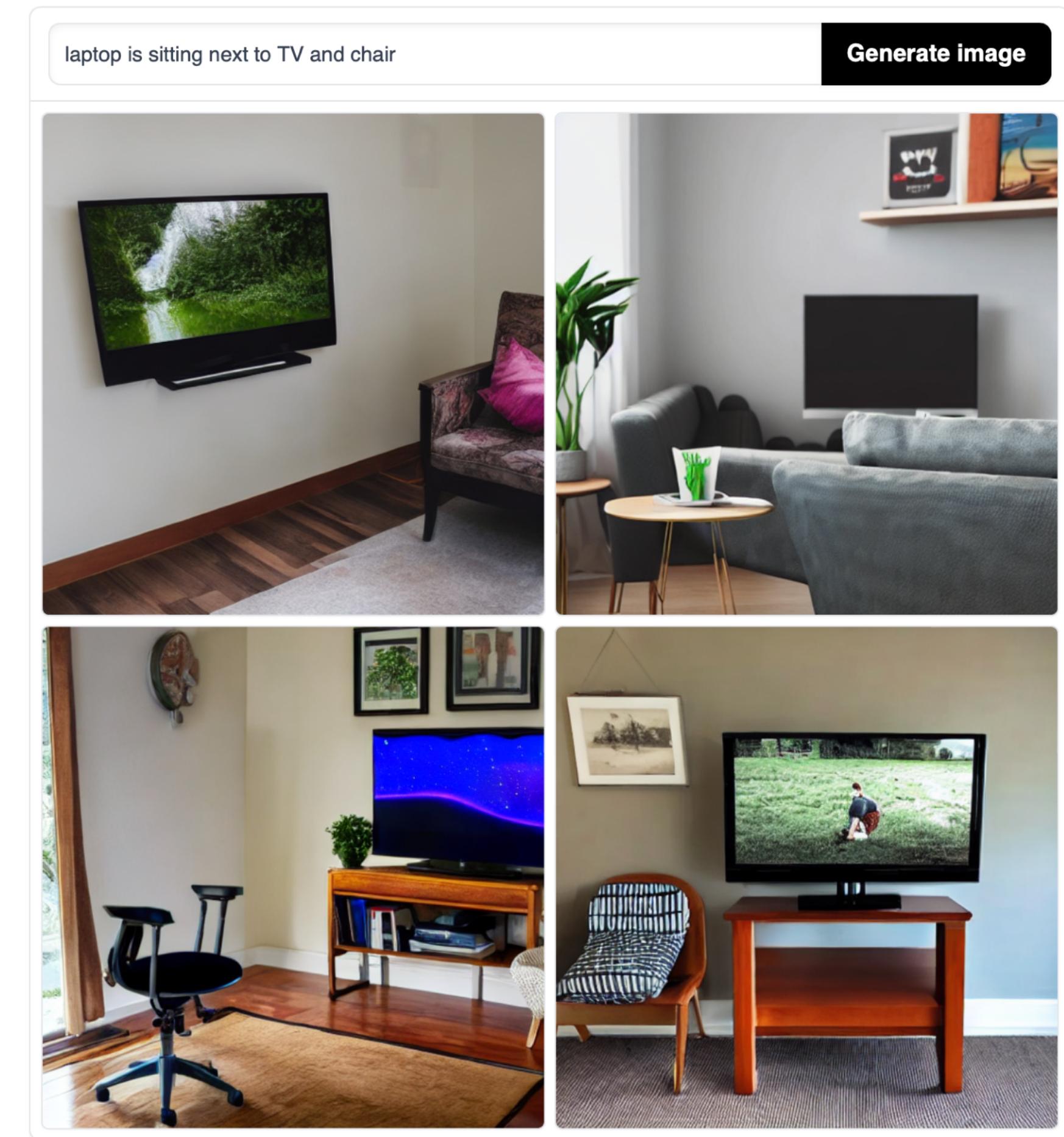


Exercise 2: Collaborate with generative AI (20 mins)

- Every group uses 20 mins to
 - use Stable Diffusion (and ChatGPT) to create a scenario for your collaborate AI
 - screenshot of thingCV
 - scenario (text)
 - AI images

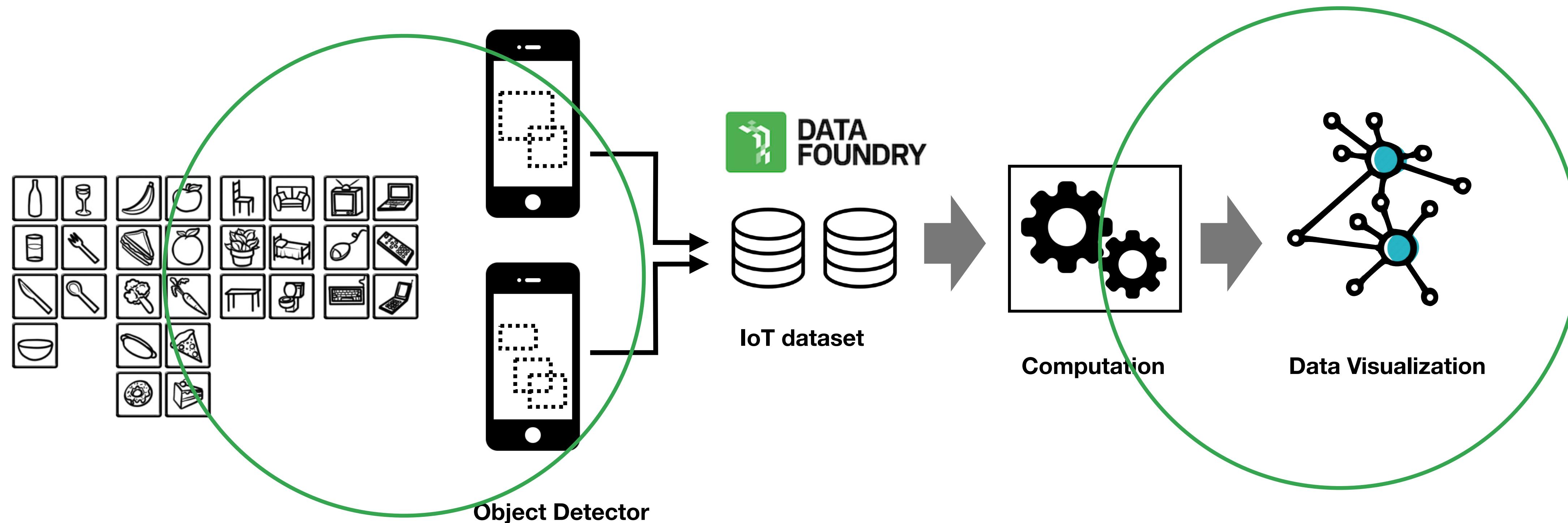


laptop is sitting next to TV and chair



Session III: Build your thingCV from Scratch

System Workflow



Object Detection

Classification



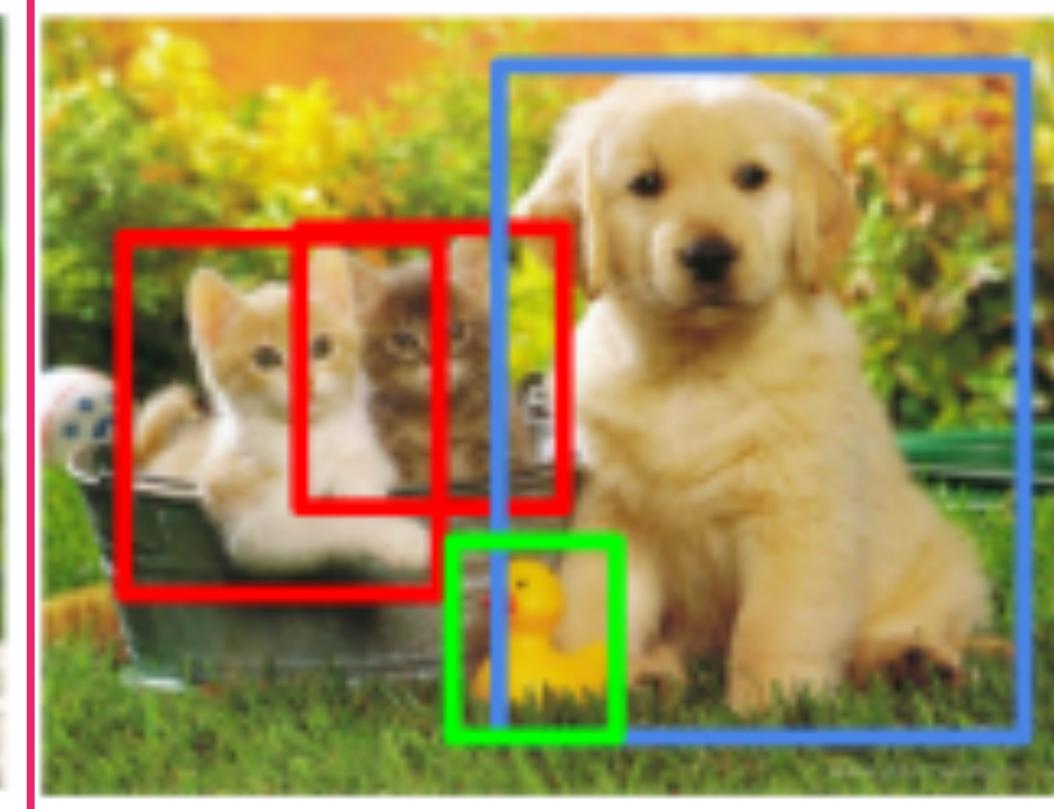
CAT

Classification + Localization



CAT

Object Detection



CAT, DOG, DUCK

Instance Segmentation



CAT, DOG, DUCK

Single object

Multiple objects

Object Detector using ML5.js



```
const video = document.getElementById('video');

// Create a ObjectDetector method
const objectDetector = ml5.objectDetector('cocossd', {}, modelLoaded);

// When the model is loaded
function modelLoaded() {
  console.log('Model Loaded!');
}

// Detect objects in the video element
objectDetector.detect(video, (err, results) => {
  console.log(results); // Will output bounding boxes of detected objects
});
```

p5.js

[Home](#)[Editor](#)[Download](#)[Donate](#)[Get Started](#)[Reference](#)[Libraries](#)[Learn](#)[Teach](#)[Examples](#)[Books](#)[Community](#)[Showcase](#)[Forum](#)

Hello!

Search p5js.org

p5.js is a JavaScript library for creative coding, with a focus on making coding accessible and inclusive for artists, designers, educators, beginners, and anyone else! p5.js is free and open-source because we believe software, and the tools to learn it, should be accessible to everyone.

Using the metaphor of a sketch, p5.js has a full set of drawing functionality. However, you're not limited to your drawing canvas. You can think of your whole browser page as your sketch, including HTML5 objects for text, input, video, webcam, and sound.

[Join the p5.js Discord!](#)

Start creating with the p5 Editor!

Community

We are a community of, and in solidarity with, people from every gender identity and expression, sexual orientation, race, ethnicity, language,



Friendly Machine Learning for the Web

A neighborly approach to creating and exploring artificial
intelligence in the browser.

What ml5.js can do?

image

ImageClassifier
PoseNet StyleTransfer
BodyPix pix2pix
UNET CVAE
Handpose SketchRNN
Facemesh ObjectDetector
FaceApi

sound

SoundClassification
PitchDetection

text

CharRNN
Sentiment
Word2Vec

helpers

NeuralNetwork
FeatureExtractor
KNNClassifier
Kmeans

p5*

File ▾ Edit ▾ Sketch ▾ Help ▾

English ▾ Hello, janetyc! ▾

Auto-refresh ImageClassification by ml5

Sketch Files < sketch.js Preview

1 images

2 index.html

3 sketch.js

```
8 Image classification using MobileNet and p5.js
9 This example uses a callback pattern to create the classifier
10 === */
11
12 // Initialize the Image Classifier method with MobileNet. A callback needs to be
13 // passed.
13 let classifier;
14
15 // A variable to hold the image we want to classify
16 let img;
17
18 function preload() {
19   classifier = ml5.imageClassifier('MobileNet');
20   img = loadImage('images/bird.jpg');
21 }
22
23 function setup() {
24   createCanvas(400, 400);
25   classifier.classify(img, gotResult);
26   image(img, 0, 0);
27 }
28
29 // A function to run when we get any errors and the results
30 function gotResult(error, results) {
31   // Display error in the console
32   if (error) {
33     console.error(error);
34   }
35   // The results are in an array ordered by confidence.
36   console.log(results);
37   createDiv('Label: ' + results[0].label);
38   createDiv('Confidence: ' + nf(results[0].confidence, 0, 2));
39 }
40
```

1 Call ml5 function and load the model

2 Apply the ml5 function to an input (e.g., image, video, text)

3 Do something with the output (e.g., value, labels, points, etc)

How to use ml5.js?

Image classification using MobileNet and p5.js



imageClassifier('MobileNet')



```
const classifier = ml5.imageClassifier('MobileNet');

classifier.classify(video, gotResult);

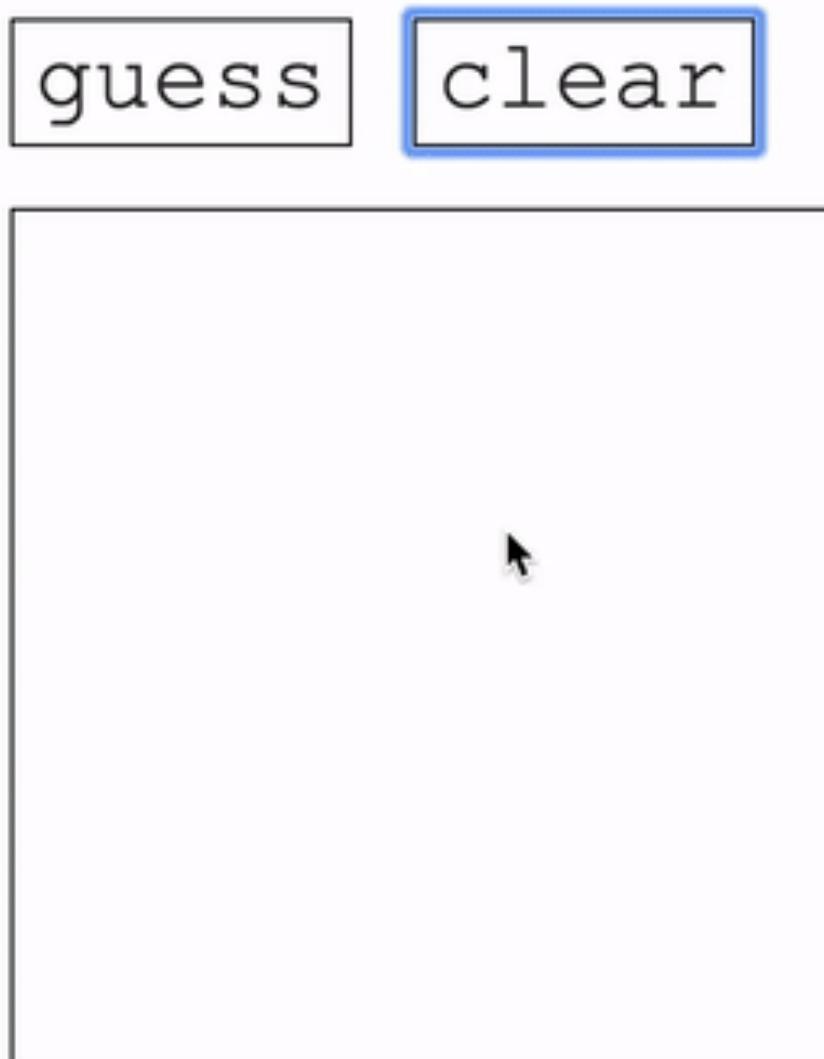
function gotResult(error, result) {
  console.log(result);
}
```

My guess is a toaster.

My confidence is 0.12.

imageClassifier('DoodleNet')

Doodle Classifier on 345 classes



<https://github.com/yining1023/doodleNet>

```
const classifier = ml5.imageClassifier('DoodleNet');

classifier.classify(canvas, gotResult);

function gotResult(error, result) {
  console.log(result);
}
```

https://editor.p5js.org/ml5/sketches/ImageClassification_DoodleNet_Canvas

PoseNet



[Image source]

```
const video = document.getElementById('video');

// Create a new poseNet method
const poseNet = ml5.poseNet(video, modelLoaded);

// When the model is loaded
function modelLoaded() {
  console.log('Model Loaded!');
}

// Listen to new 'pose' events
poseNet.on('pose', (results) => {
  poses = results;
});
```

BodyPix

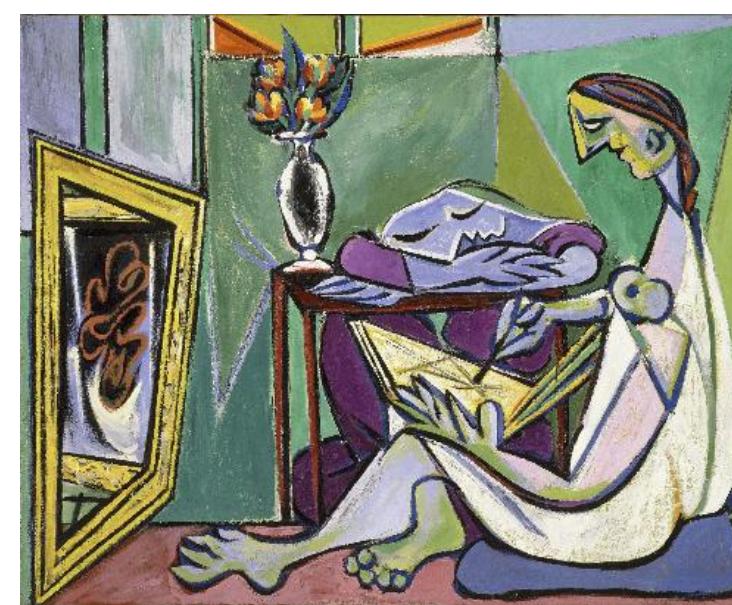


```
const bodypix = ml5.bodyPix(modelReady);

function modelReady() {
  // segment the image given
  bodypix.segment(img, gotResults);
}

function gotResults(error, result) {
  if (error) {
    console.log(error);
    return;
  }
  // log the result
  console.log(result.backgroundMask);
}
```

Style Transfer



[Image source]

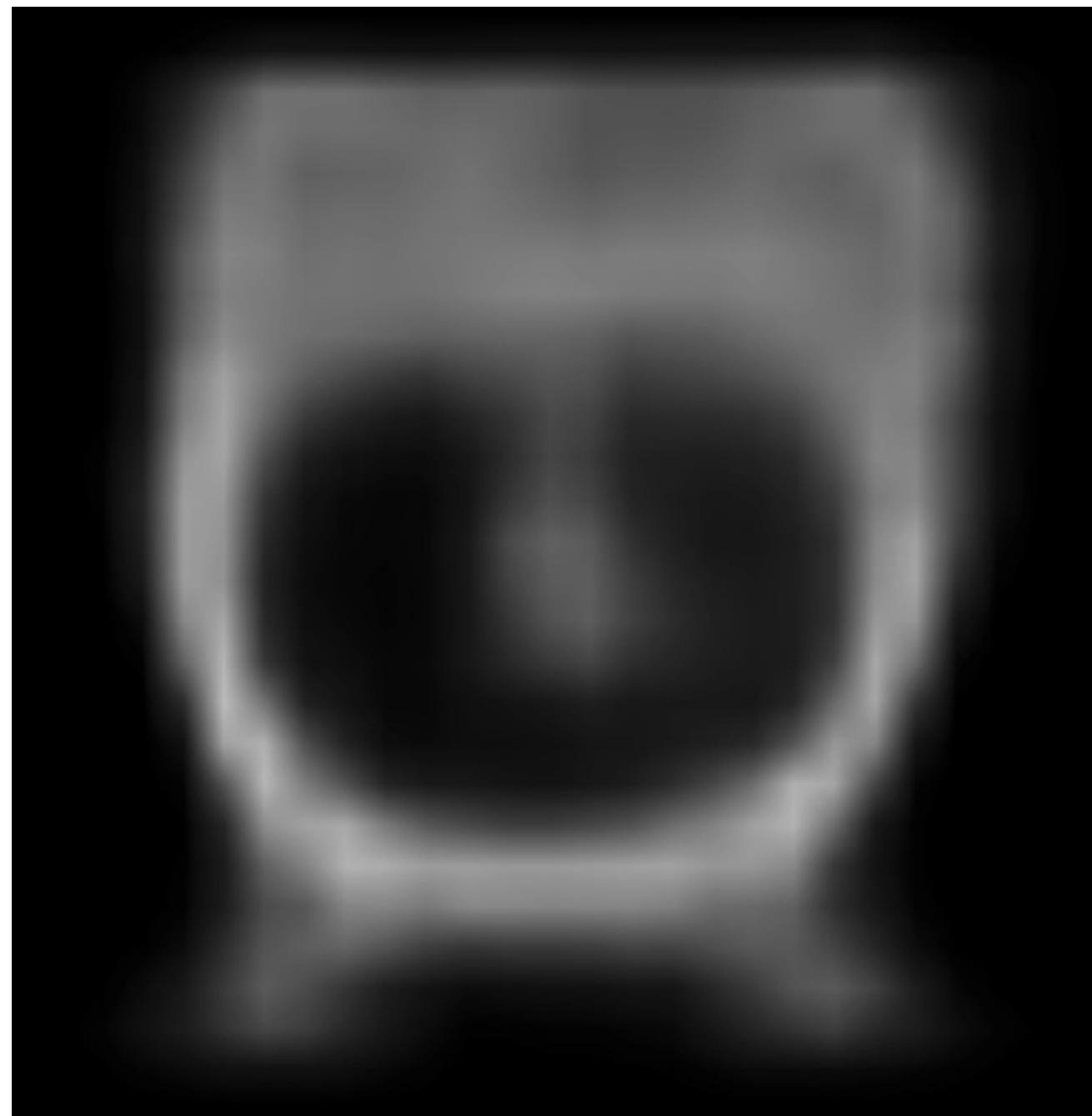
```
// Create a new Style Transfer Instance
const style = ml5.styleTransfer('data/myModel/', modelLoaded);

// When the model is loaded
function modelLoaded() {
  console.log('Model Loaded!');
}

// Grab a img element and generate a new image.
style.transfer(document.getElementById("img"), function(error, result)
{
  img.src = result.src;
});
```

CVAE

(conditional variational autoencoder)



[Image source]

```
const cvae = ml5.CVAE('model/quick_draw/manifest.json', modelReady);

function modelReady() {
  // generate an image of an airplane
  cvae.generate('airplane', gotImage);
}

function gotImage(error, result) {
  if (error) {
    console.log(error);
    return;
  }
  // log the result
  console.log(result);
}
```

CharRNN

This example uses a pre-trained model on a corpus of [Virginia Woolf](#)

seed text:

temperature: 0.5

Model Loaded

The sky was blue and

```
// Create the character level generator with a pre trained model
const rnn = ml5.charRNN('models/bolaño/', modelLoaded);

// When the model is loaded
function modelLoaded() {
  console.log('Model Loaded!');
}

// Generate content
rnn.generate({ seed: 'the meaning of pizza is' }, (err, results) => {
  console.log(results);
});
```

Sentiment Analysis

Sentiment Analysis Demo

This example uses model trained on movie reviews. This model scores the sentiment of text with a value between 0 ("negative") and 1 ("positive"). The movie reviews were truncated to a maximum of 200 words and only the 20,000 most common words in the reviews are used.

model loaded

sentiment score:

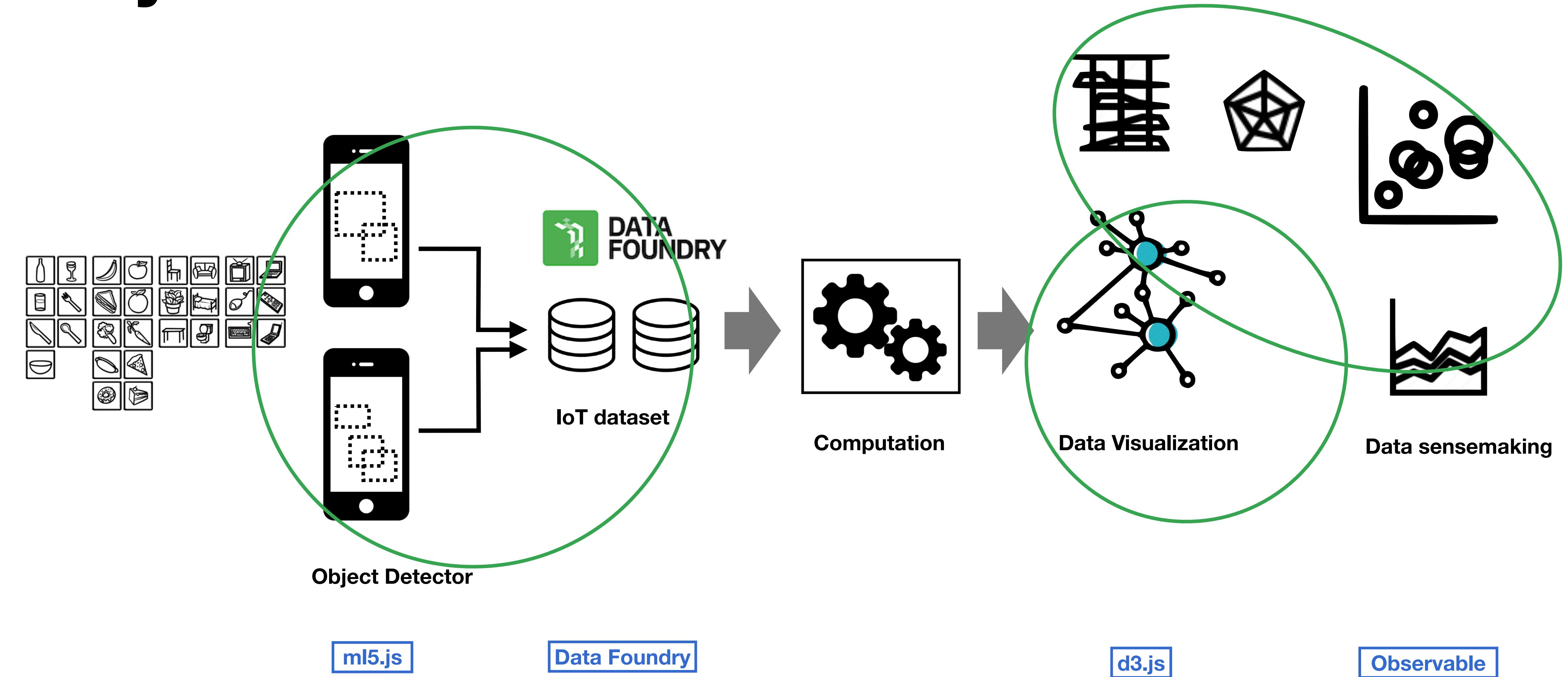
```
// Create a new Sentiment method
const sentiment = ml5.sentiment('movieReviews', modelReady);
```

```
// When the model is loaded
function modelReady() {
  // model is ready
  console.log('Model Loaded!');
}
```

```
// make the prediction
const prediction = sentiment.predict(text);
console.log(prediction);
```

Exercise 3: Build Object Detector

System Workflow



editor.p5js.org

gmail Research agent wiki Good Sites Class Tools Dict Design tool oTranscribe Time Zone academics - Dropbox TU/e 圖庫&Icon

p5* File Edit Sketch Help English Hello, janetycl!

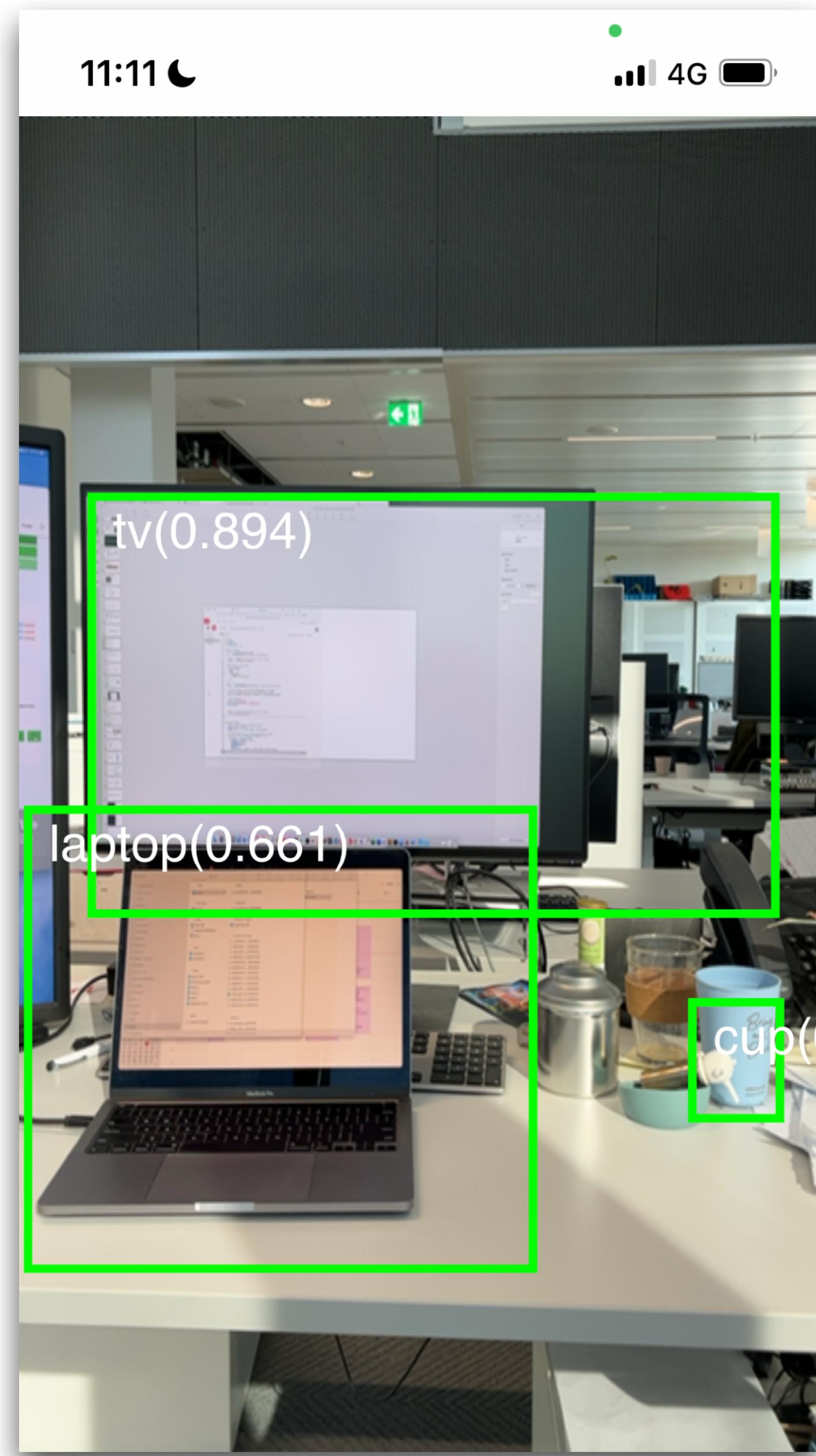
Auto-refresh MDC_Workshop_1_ObjectDetector_Webcam by janetyc

Sketch Files sketch.js

Saved: about 18 hours ago Preview

```
8
9 let cnv;
10 let video;
11 let detector;
12 let detections = [];
13
14 function setup() {
15 // cnv = createCanvas(500, 500);
16 cnv = createCanvas(windowWidth, windowHeight);
17
18 //Step 1: change the front camera to back camera
19 //video = createCapture(VIDEO);
20
21 //use phone's back camera
22 var constraints = {
23 audio: false,
24 video: {
25 facingMode: {
26 exact: "environment"
27 }
28 }
29 };
30 video = createCapture(constraints); //save the video results
31
32 // The line below + the videoLoadedCallback were added
33 // after the video was shot to fix compatibility issues.
34 video.elt.addEventListener('loadeddata', videoLoadedCallback);
35
36 // set video size
37 //video.size(500, 500);
38 video.size(windowWidth, windowHeight);
39 video.hide();
40
41
42 //Step 3: log data when people touch the screen
43 //cnv.touchEnded(logData);
44 }
45
46
47 function draw() {
48 //Step 2: draw video and detected objects
49 //draw video on canvas
50 image(video, 0, 0, video.width, video.height);
51
52 //draw bounding boxes for detected objects on canvas
53 for (let i = 0; i < detections.length; i++) {
54 let object = detections[i];
55 if(object.confidence > 0.6) {
56 stroke(0, 255, 0);
57 strokeWeight(4);
58 noFill();
59 rect(object.x, object.y, object.width, object.height);
}
}
}
}
```

Console Clear



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Getting Started with ml5.js</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- p5 -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/p5.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/addons/p5.sound.min.js"></script>
    <!-- ml5 -->
    <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>
  </head>

  <body>
    <script>
      function setup() {
        createCanvas(400, 400);
      }

      function draw() {
        background(200);
      }
    </script>
  </body>
</html>
```

```
9
10 let cnv;
11 let video;
12 let detector;
13 let detections = [];
14
15 function setup() {
16   cnv = createCanvas(windowWidth, windowHeight);
17   cnv.touchEnded(logData);
18   var constraints = {
19     audio: false,
20     video: {
21       facingMode: {
22         exact: "environment"
23       }
24     }
25   };
26   video = createCapture(constraints);
27
28 // The line below + the videoLoadedCallback were added
29 // after the video was shot to fix compatibility issues.
30 video.elt.addEventListener('loadeddata', videoLoadedCallback);
31
32 video.size(windowWidth, windowHeight);
33 video.hide();
34}
35
36
37 function draw() {
38   image(video, 0, 0, video.width, video.height);
39
40   for (let i = 0; i < detections.length; i++) {
41     let object = detections[i];
42     if(object.confidence > 0.6) {
43       stroke(0, 255, 0);
44       strokeWeight(4);
45       noFill();
46       rect(object.x, object.y, object.width, object.height);
47       noStroke();
48       fill(255);
49       textSize(24);
50       text(object.label+(object.confidence.toFixed(3)), object.x + 10, object.y + 24);
51     }
52   }
53 }
```

setup

draw

```
54
55 //solve loaded video issues
56 function videoLoadedCallback() {
57   print("Video Loaded");
58
59 //call model here
60 // Models available are 'cocossd', 'yolo'
61 detector = ml5.objectDetector('cocossd', modelReady);
62 }
63
64 function modelReady() {
65   detector.detect(video, gotDetections);
66 }
67
68 function gotDetections(error, results) {
69   if (error) {
70     console.error(error);
71   }
72   detections = results;
73   detector.detect(video, gotDetections);
74 }
```

objectDetector

Portfolio

My projects

Archive

Community

Collaborations

Subscriptions

Explore

Data tools

Guides

Support

Home

PROJECT STATUS

ADD PROJECT

PROJECTS

MY DIARY

post a diary via telegram bot

Janet Huang

EXISTING

TELEGRAM DIARY

Post a diary via telegram bot

Janet Huang

DIARY MEDIA

THING DIARY

collect object story

Janet Huang

DIARY MEDIA

ARTIFICE WORKSHOP

starboard with ml5.js

Janet Huang

EXISTING

ARTIFICE DEMO

demo

Janet Huang

EXISTING

DCB150 DATA WORKSHOP

data workshop

Janet Huang

EXISTING

[ARTIFICE] AI WORKSHOP: CUS...

Build your own thingCV using object detector

Janet Huang

IOT EXISTING

[ARTIFICE] OBJECT HUNTER

object hunter for 80 objects

Janet Huang

EXISTING

Data Foundry

Data Foundry is a platform that supports researchers to design intelligent interactive product.

It enables an easier way of collecting data, storing data, connecting data, and sharing data with other people. It also provides multiple useful tools for researchers to do rapid prototyping.

Data Foundry: IoT Dataset

The screenshot shows the Data Foundry web application interface. On the left, there is a sidebar with various navigation options: Portfolio, My projects, Archive, Community (which is highlighted in green), Collaborations, Subscriptions, Explore, Data tools, Guides, and Support. The main area displays a list of datasets under 'DATASETS (5)'. A modal window titled 'CHOOSE A DATASET TYPE' is open in the center, listing several dataset types:

- Script**: Script to automate stuff 🔧
- FitBit Dataset**: Data from connected FitBit devices 🏃
- GoogleFit Dataset**: Data from connected GoogleFit devices 🏃
- IOT Dataset**: Data from connected devices 🌐
- Entity Dataset**: JSON Database for variable data 🏛
- Form Dataset**: Data is collected by a simple form 📝
- Annotation data set**: Data by researcher as annotations 🎨
- Diary Dataset**: Data by participants as diary entries 📆

CONFIGURATION

[HTTP-POST](#)[OOCSI STREAM](#)[CSV/JSON TOKEN LINK](#)[OOCSI STREAM](#)

HTTP POST requests for this dataset (ID: 7642)

POST requests are easy to send from a wide variety of platforms and technologies. Find a list of possible options on the right side.

To allow these requests to store data in the dataset, you need to send them with a special token:

Token

MzlySFJHSFNqUjkzMHVWSmovSS9DMEhkMitEY25GV3FBM3VkaEZ5Rm9uaz0=

[GENERATE](#)[DELETE](#)

Generate a token to activate this inlet, delete the token to deactivate.

How to use this?

JavaScript

```
var data = { ... your data goes here ... }
var jsonBody = {
  activity: 'ACTIVITY',
  source_id: 'DEVICE_ID',
  data: JSON.stringify(data)
}
fetch('https://data.id.tue.nl/datasets/ts/record/7642/MzIySFJHSFNqUjkzMHVWSmovSS9DMEhkMitEY25GV3FBM3VkaEZ5Rm9u
      method: 'POST',
      mode: 'cors',
      cache: 'no-cache',
      headers: {
        'Content-Type': 'application/json'
      },
      redirect: 'follow',
      referrerPolicy: 'no-referrer',
      body: JSON.stringify(jsonBody)
});
```

Replace DEVICE_ID by the refid attribute of any device in the project. Use any ACTIVITY or leave empty. Provide data in JSON format, such as
 {"parameter1": 2, "parameter2": 5}

```
92
93   function logData() {
94     final_detections = detections
95     objectList = getDetectionObjects(final_detections);
96
97     let data= {
98       time: +(new Date),
99       detections: objectList
100    }
101
102   let jsonBody = {
103     activity: 'Customize_ThingCV',
104     data: JSON.stringify(data)
105   }
106
107   fetch('https://data.id.tue.nl/datasets/ts/record/7642/MzIySFJHSFNqUjkzMHVWSmovSS9DMEhkMitEY25GV3
108     method: 'POST',
109     mode: 'cors',
110     cache: 'no-cache',
111     headers: {
112       'Content-Type': 'application/json'
113     },
114     redirect: 'follow',
115     referrerPolicy: 'no-referrer',
116     body: JSON.stringify(jsonBody)
117   });
118
119
```

Portfolio

My projects

Archive

Community

Collaborations

Subscriptions

Explore

Data tools

Guides

Support

Home > [DCM210] AI Workshop: Customized your thingCV > Object Detector using ml5.js

id: 7641 EXISTING PUBLIC MIT

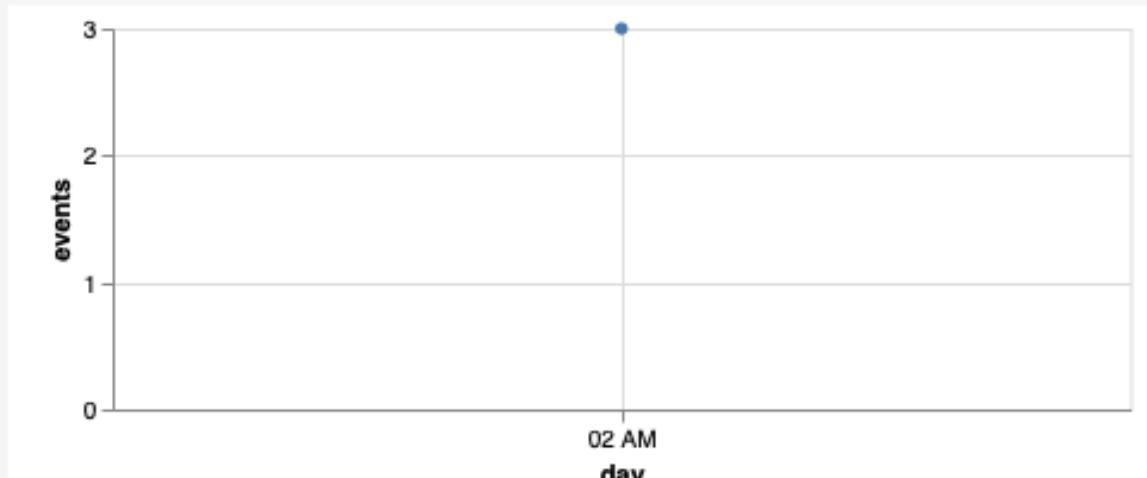
□ OBJECT DETECTOR USING ML5.JS

2023-03-01

2024-03-01

an object detector that can detect 80 objects through a camera

INFO	
License: MIT	

[VIEW DATA](#)[DOWNLOAD](#)[UPLOAD FILE\(S\)](#)

DATASET FILES

File name	Description	Uploaded	
sketch.js	object detector	Mar 01 at 14:27	edit delete
style.css	object detector	Mar 01 at 14:27	edit delete
index.html	object detector	Mar 01 at 14:27	edit delete

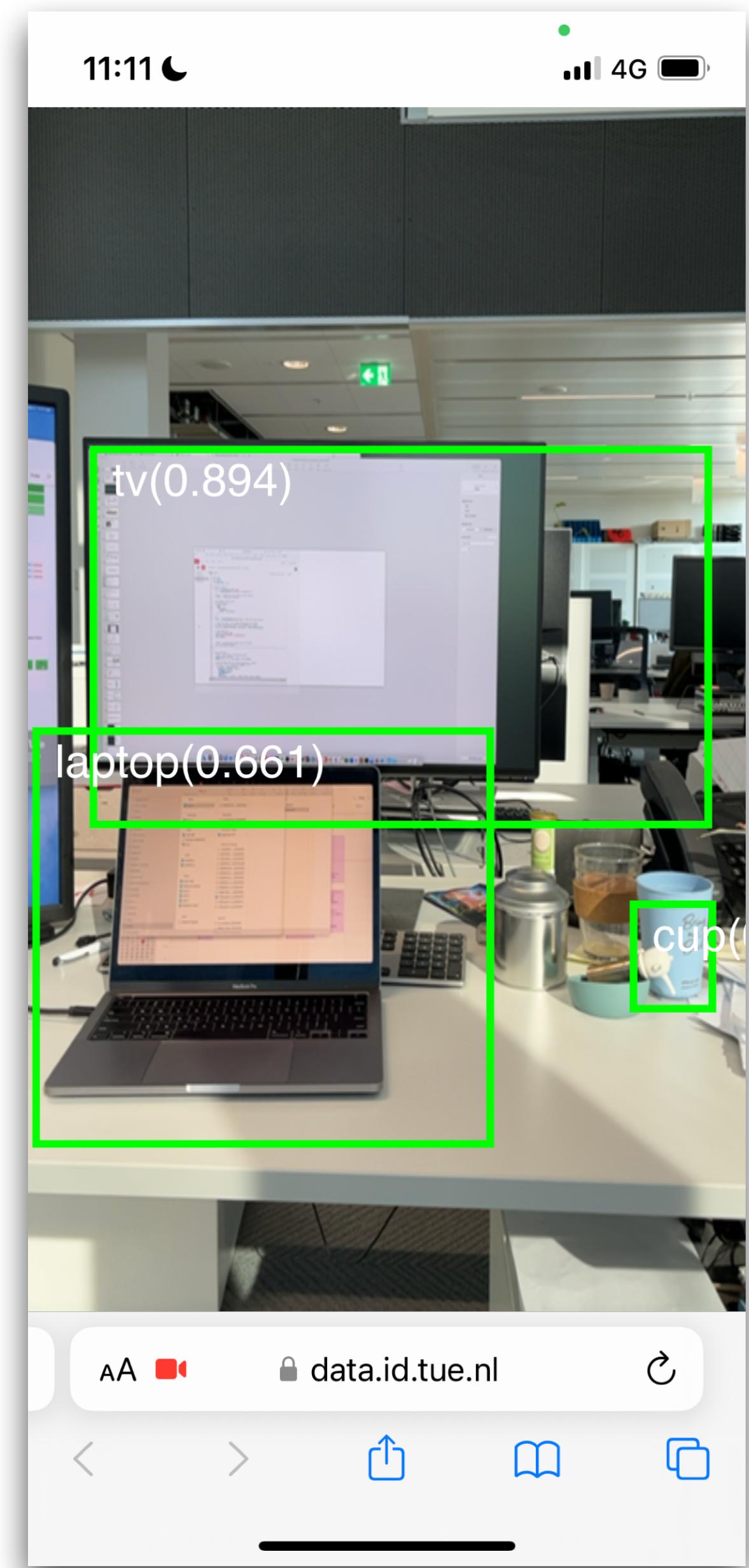
CONFIGURATION

 [CSV/JSON TOK...](#) [WEB-ACCESS](#) [OOCSI STREAM](#)

Access the data as a website

As the research team, you can always access the dataset as a [website](#), if the dataset is [active](#). (This is independent from public accessibility.)

This dataset is accessible as a public website with the following link:

<https://data.id.tue.nl/web/dStvaTU4ZGZKVVZLeGtPNU5FTVVlQWs1MFFidk1xM1ZQUVlp>[ACTIVATE WEB ACCESS](#)[DISABLE WEB ACCESS](#)

Portfolio

My projects

Archive

Community

Collaborations

Subscriptions

Explore

Data tools

Guides

Support

[Home](#) > [DCM210] AI Workshop: Customized your thingCV > Object Data > Data table

OBJECT DATA

[Show entries](#)

Search:

	id	ts	pp1	pp2	pp3	detections	time
	16	2023-03-01T14:37:24				tv	1677677844572
	17	2023-03-01T14:37:24				tv	1677677844684
	18	2023-03-01T14:37:28				laptop,tv	1677677848344
	19	2023-03-01T14:37:28				laptop,tv	1677677848518
	20	2023-03-01T14:37:28				laptop,tv	1677677848691

Showing 16 to 20 of 20 entries

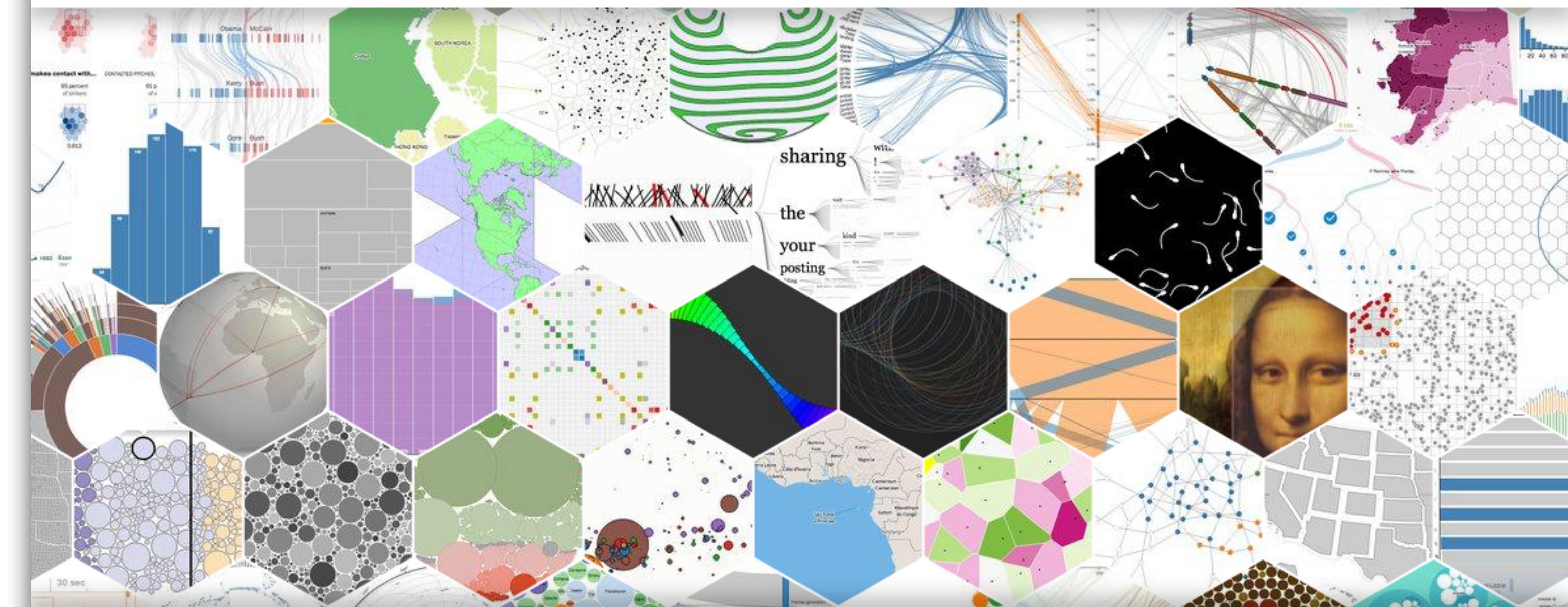
Previous

1

2

Next

Data Visualization



Like visualization and creative coding? Try interactive JavaScript notebooks in [Observable!](#)

D3.js is a JavaScript library for manipulating documents based on data. **D3** helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

Download the latest version (7.6.1) here:

- [d3-7.6.1.tgz](#)

To link directly to the latest release, copy this snippet:

```
<script src="https://d3js.org/d3.v7.min.js"></script>
```

The [full source and tests](#) are also available [for download](#) on GitHub.

- [See more examples](#)
- [Chat with the community](#)
- [Follow announcements](#)
- [Report a bug](#)
- [Ask for help](#)

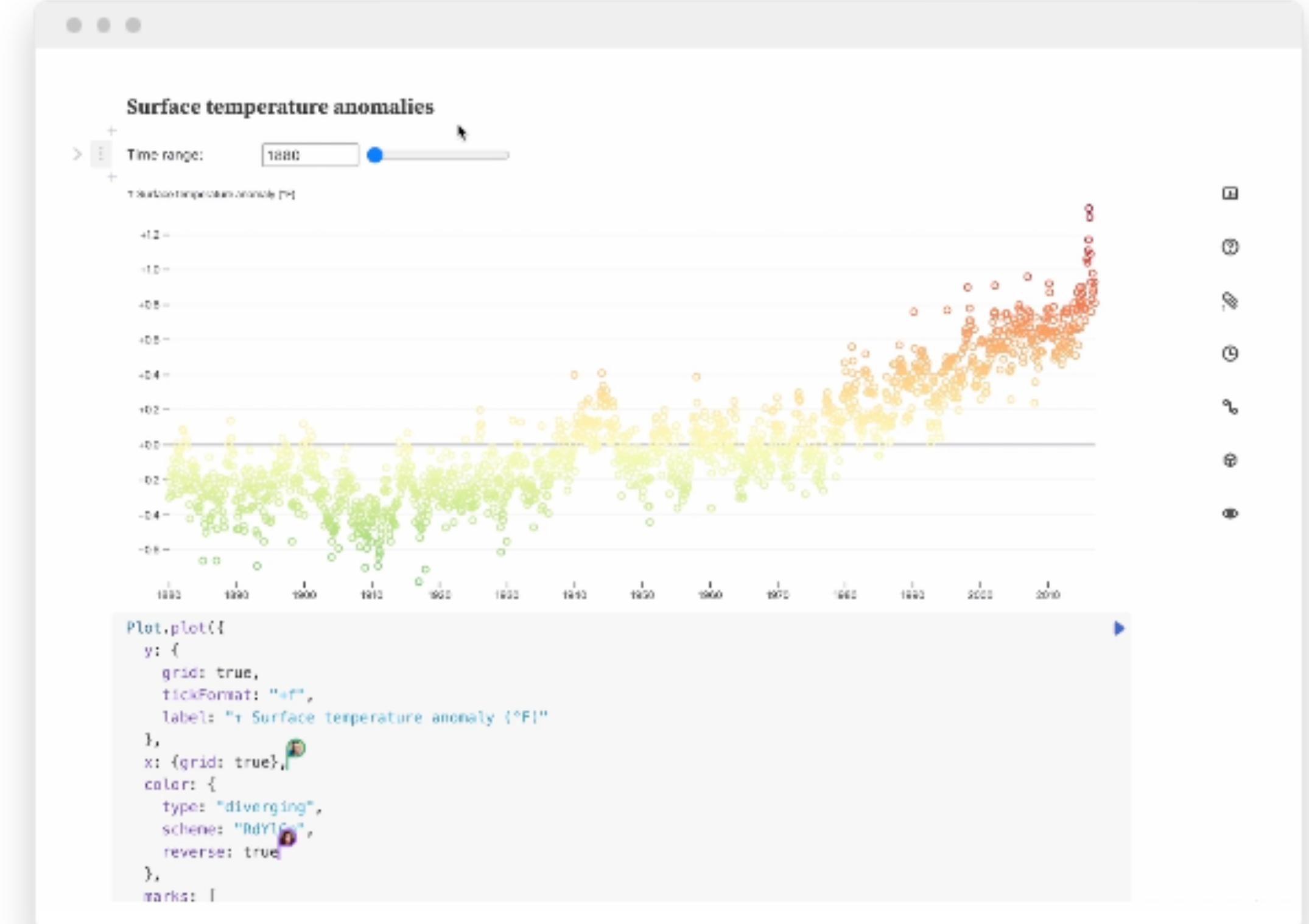
observablehq.com

gmail Research agent wiki Good Sites Class Tools Dict Design tool oTranscribe Time Zone academics - Dropbox TU/e 圖庫&Icon

Collaborative data platform and canvas | Observable

Product Pricing Solutions Explore Learn Community Search Sign in Sign up

Explore, analyze, and explain data.



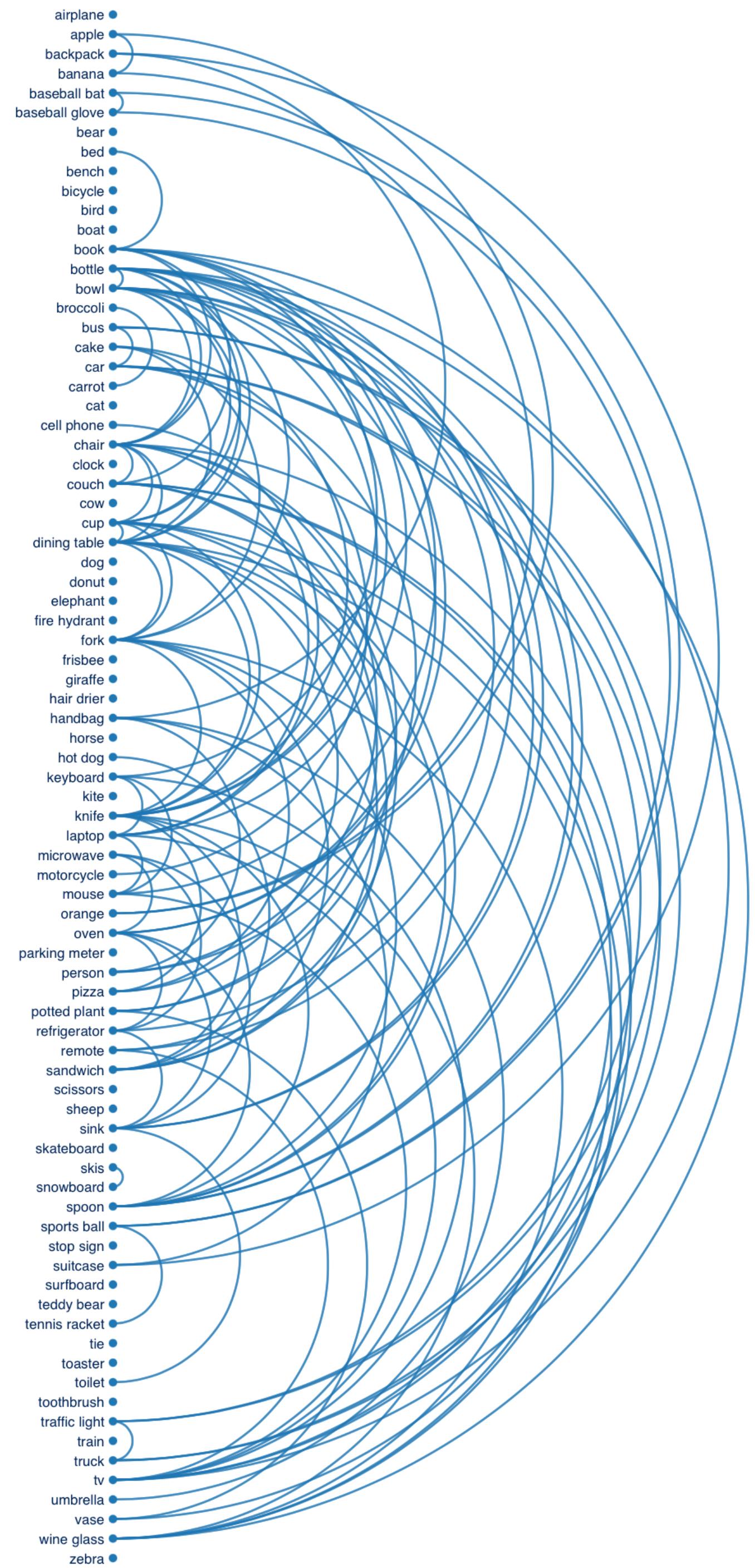
Uncover new insights, answer more questions, and make better decisions.

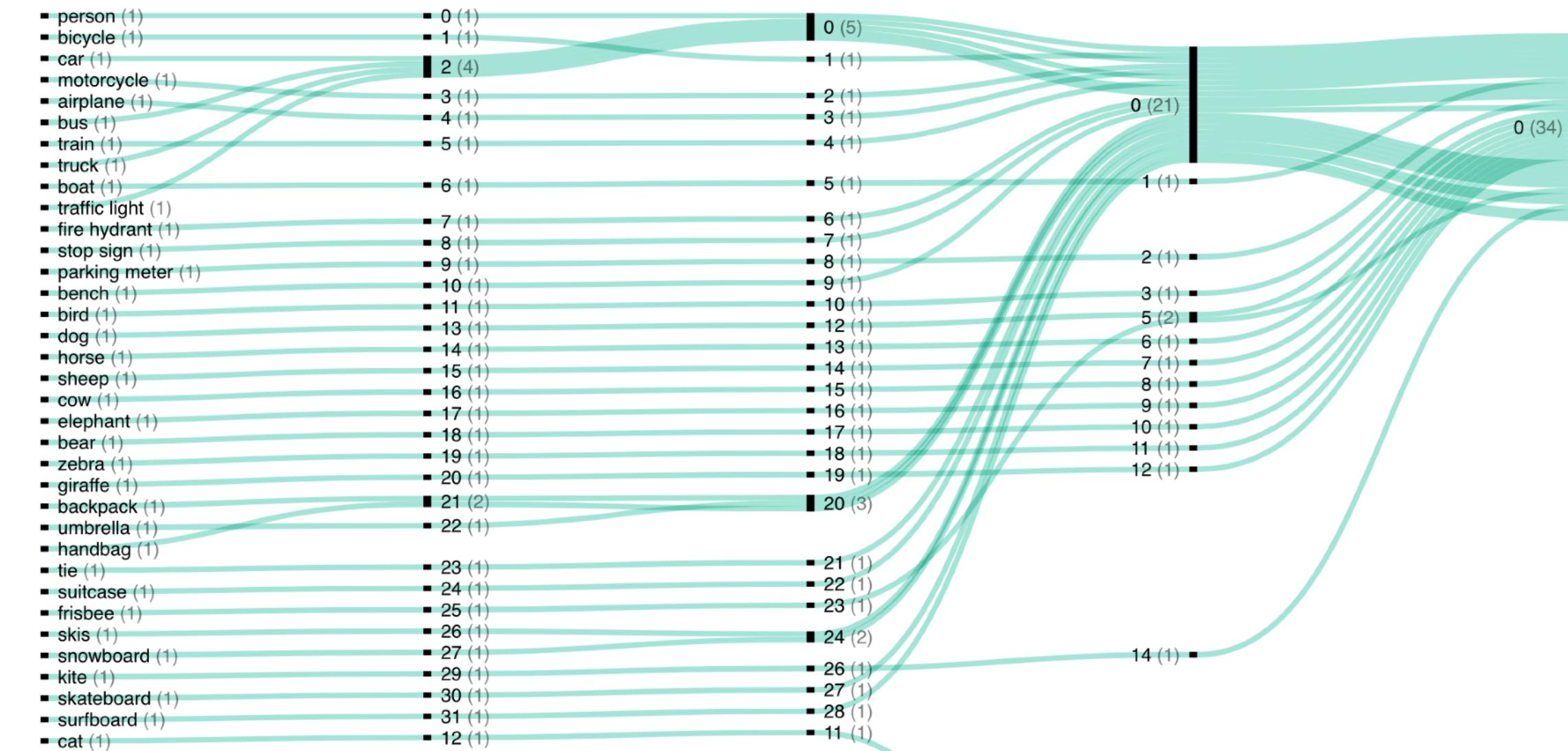
Sign up for free

<https://observablehq.com/>

Arc Diagram

This diagram places nodes in a horizontal or vertical line, with circular arcs for links. Unlike other network visualizations such as a [force layout](#), the appearance (and usefulness) of an arc diagram is highly dependent on the order of nodes. Hover over a node below to inspect its connections.

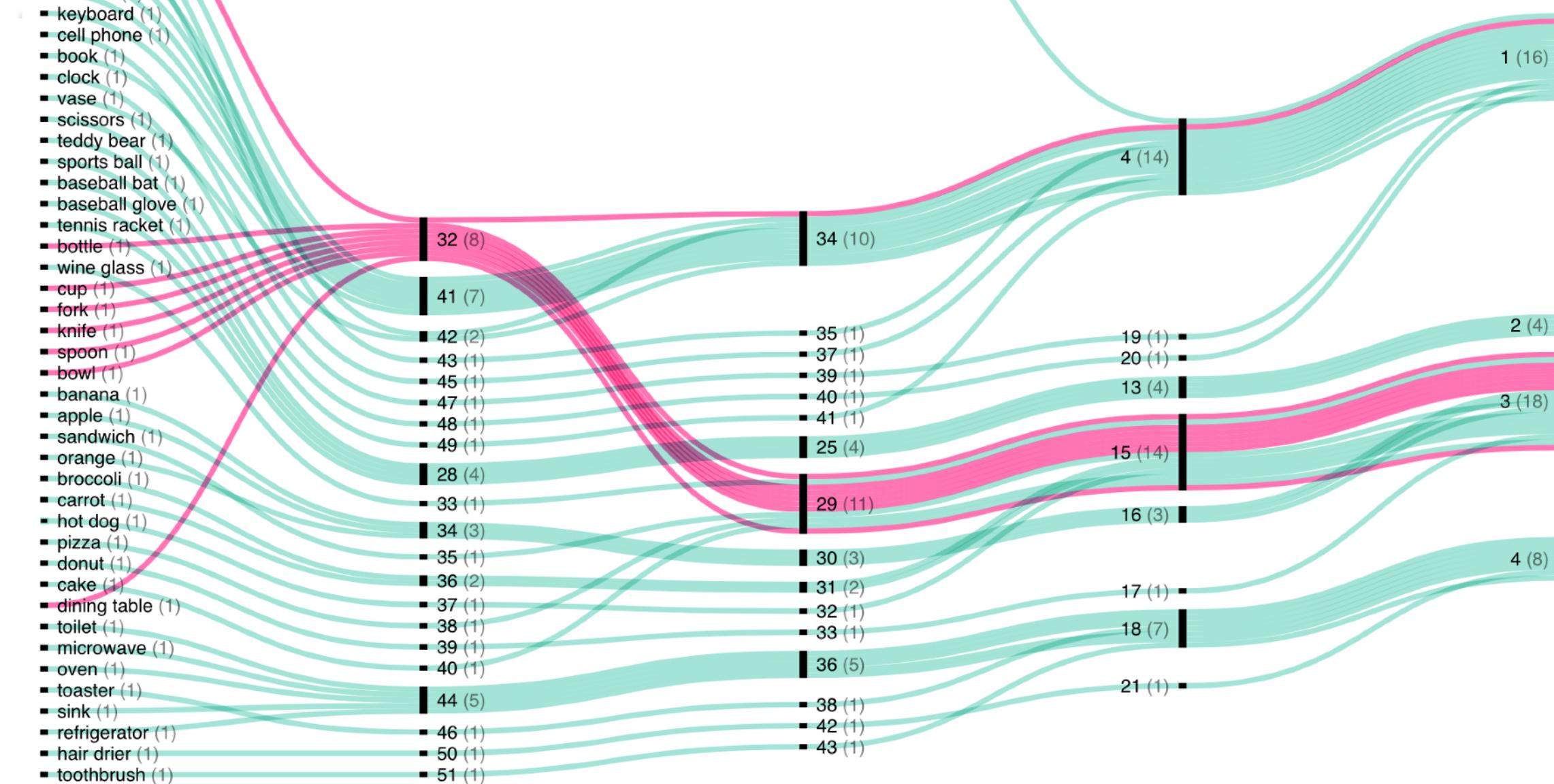




Visualization: Parallel sets

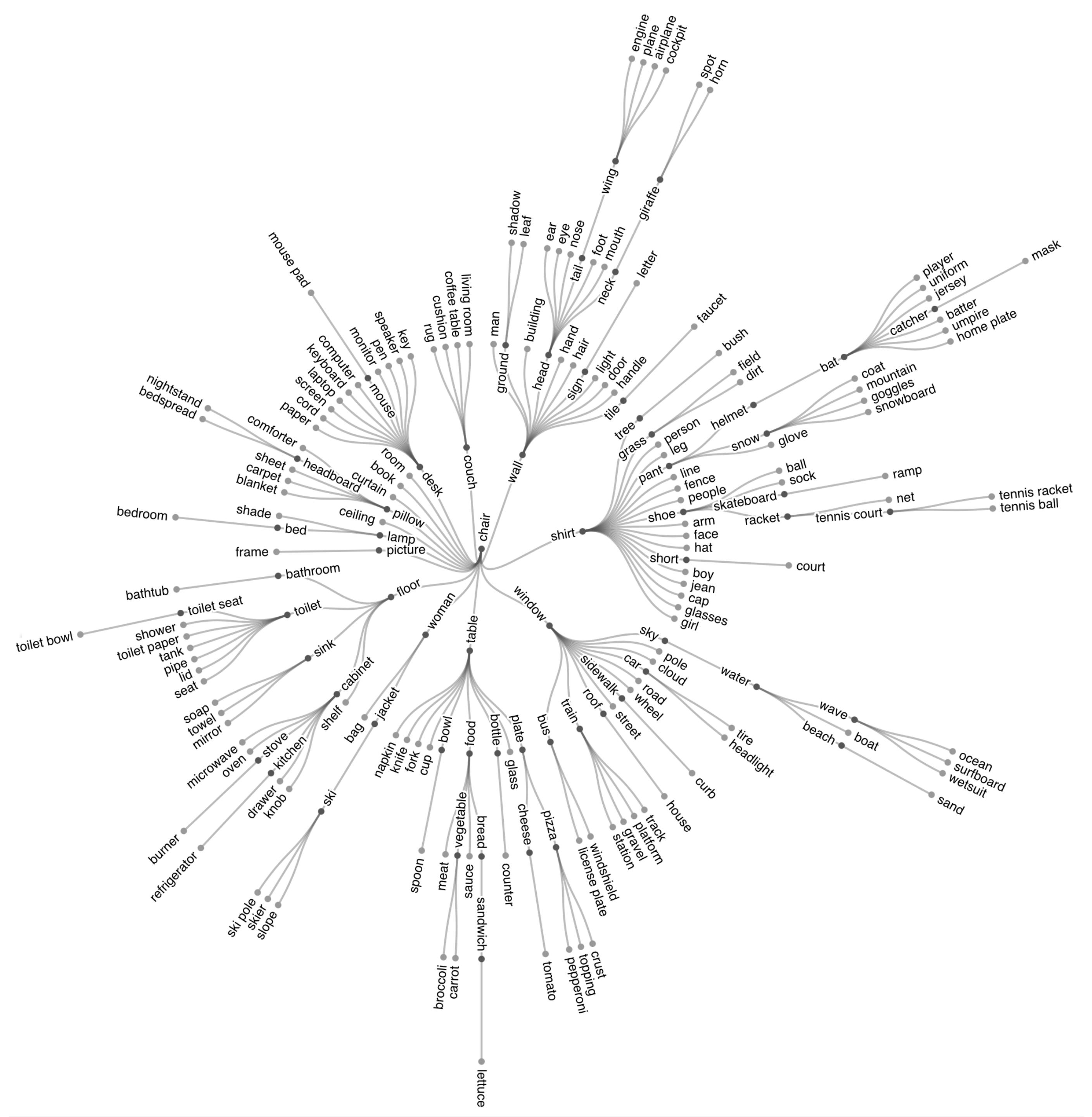
Parallel sets are like [parallel coordinates](#), but for categorical dimensions. The thickness of each curved line represents a quantity that is repeatedly subdivided by category.

This example looks at community flow of Everyday objects in thingCV



Visualization: Radial Tidy Tree

D3's [tree layout](#) implements the [Reingold–Tilford “tidy” algorithm](#) for constructing hierarchical node-link diagrams, improved to run in linear time by [Buchheim *et al.*](#). Tidy trees are typically more compact than [cluster dendograms](#), which place all leaves at the same level. See also the [Cartesian variant](#).



```
ego_network.json      own_graph.json  
1 {"nodes":  
2     [{"id": "person", "group": 1},  
3      {"id": "bicycle", "group": 1},  
4      {"id": "car", "group": 1},  
5      {"id": "motorcycle", "group": 1},  
6      {"id": "airplane", "group": 1},  
7      {"id": "bus", "group": 1},  
8      {"id": "train", "group": 1},  
9      {"id": "truck", "group": 1},  
10     {"id": "boat", "group": 1}], |  
11  "links":  
12    [{"source": 0, "target": 1, "weight": 0.1},  
13    {"source": 0, "target": 2, "weight": 0.3},  
14    {"source": 0, "target": 3, "weight": 0.02},  
15    {"source": 0, "target": 4, "weight": 0.01},  
16    {"source": 0, "target": 5, "weight": 0.0},  
17    {"source": 1, "target": 2, "weight": 0.5},  
18    {"source": 1, "target": 3, "weight": 0.03},  
19    {"source": 4, "target": 7, "weight": 0.6}],  
20 }  
|
```

Social-centric network

```
{"nodes": [  
    {"id": "object1", "group": 1},  
    {"id": "object2", "group": 1},  
    {"id": "object3", "group": 1}],  
"links": [  
    {"source": 0, "target": 1, "weight": 0.1},  
    {"source": 0, "target": 2, "weight": 0.5},  
    {"source": 0, "target": 3, "weight": 0.3},  
    {"source": 1, "target": 3, "weight": 0.2}],  
}|
```

```

1 {"id": "toaster",
2  "children":
3   [{"id": "microwave",
4    "children":
5      [{"id": "bottle",
6       "children":
7         [{"id": "person",
8          "children":
9            [{"id": "car",
10             "children":
11               [{"id": "bus"}, {"id": "truck"}, {"id": "traffic light"}]
12             },
13             {"id": "handbag",
14              "children":
15                [{"id": "backpack"}, {"id": "umbrella"}]
16             }
17           ]
18         },
19         {"id": "wine glass",
20          "children":
21            [{"id": "fork"}, {"id": "chair",
22              "children":
23                [{"id": "couch"}, {"id": "potted plant"}, {"id": "dining table"}, {"id": "book",
24                  "children":
25                    [{"id": "bed"}, {"id": "mouse"}, {"id": "remote"}, {"id": "keyboard"}]
26                  }
27                ]
28              }
29            ]
30          }
31        ]
32      }
33    ]
34  }
35 ]
36 },
37 {"id": "bowl",
38  "children":
39    [{"id": "apple",
40      "children": [{"id": "banana"}]}, {"id": "sandwich",
41      "children": [{"id": "hot dog"}]}, {"id": "orange"}, {"id": "broccoli"}, {"id": "carrot"}, {"id": "cake"}, {"id": "vase"}], {"id": "sink",
42      "children": [{"id": "toilet"}, {"id": "toothbrush"}]}]
43 },
44 {"id": "refrigerator"
45 },
46 {"id": "bedroom"
47 },
48 {"id": "kitchen"
49 },
50 {"id": "bathroom"
51 }

```

Ego-centric network

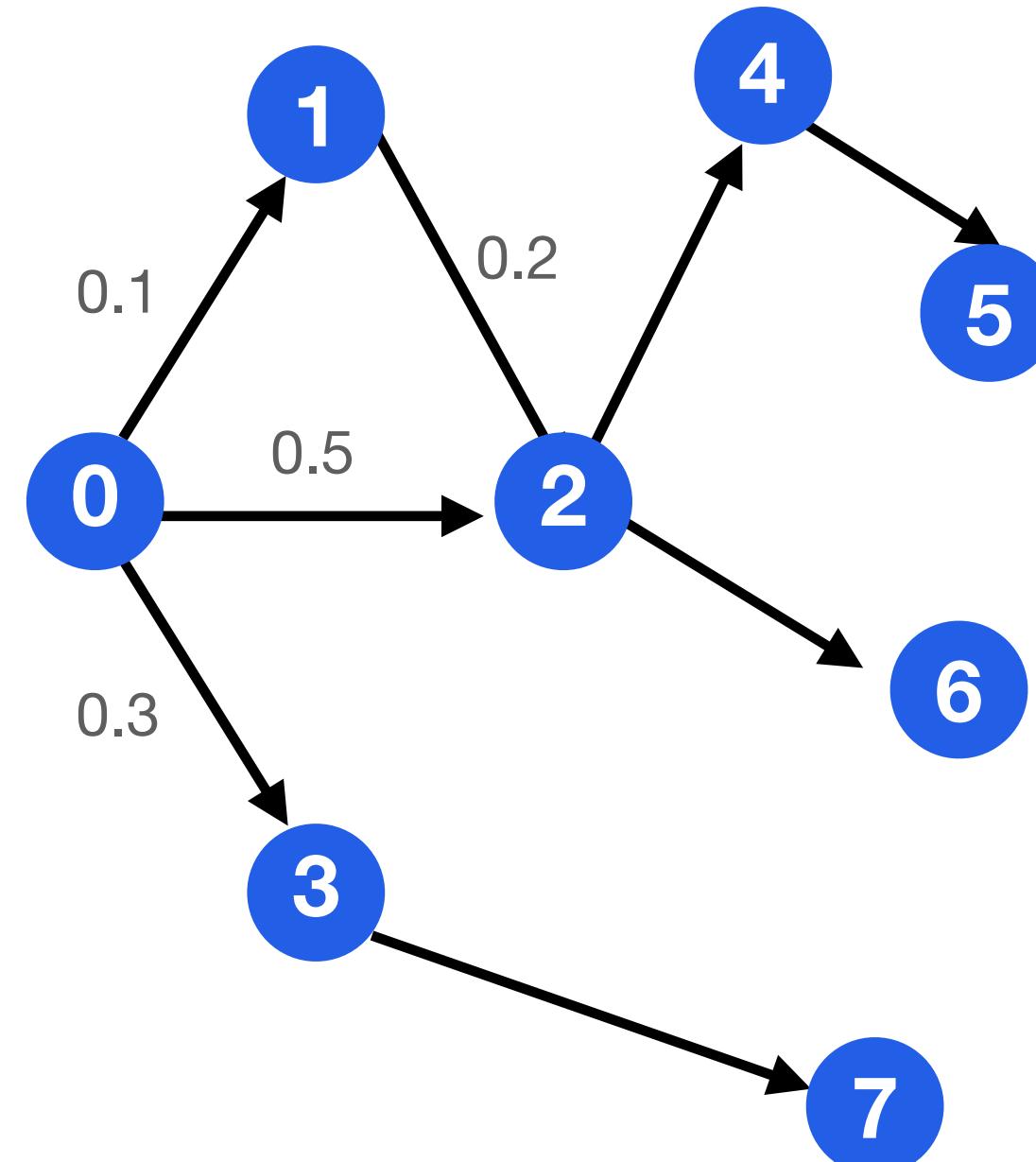
```

{"id": "objectId1",
  "children": [
    {"id": "objectId2",
      "children": [...]},
    {"id": "objectId3",
      "children": [..., ..., ...]},
    {"id": "objectId4",
      "children": [...]}]
}

```

Exercise 4: Data Visualizer

- Step 1: Try to upload data visualizer to Data Foundry
- Step 2: Every group uses 10 mins to create your own data and feed it into data visualization



```
{"nodes": [  
    {"id": "object1", "group": 1},  
    {"id": "object2", "group": 1},  
    {"id": "object3", "group": 1},  
    {"...."}],  
"links": [  
    {"source": 0, "target": 1, "weight": 0.1},  
    {"source": 0, "target": 2, "weight": 0.5},  
    {"source": 0, "target": 3, "weight": 0.3},  
    {"source": 1, "target": 2, "weight": 0.2}  
}]
```

```
298 }
299
300
301 //read data from json file
302 //var DFdata = readDataFromDF();
303 //DFdata.then(function(graph) {
304
305 ▼ d3.json('object_graph.json') function(error, graph) {
306   if (error) throw error;
307
308
309 ▼ function getByValue(map, searchValue) {
310   for (let [key, value] of map.entries()) {
311     if (value.id === searchValue)
312       return value.i;
313   }
314 }
315 //-----
316
317 ▼ var init_data = function (){
318   graph.links.forEach(function(d,i){
319     d.i = i;
320   });
321
322   //clear up nodeArea
323   d3.select('#nodeArea').html("");
324
325   graph.nodes.forEach(function(d,i){
326     var nodeArea = d3.select('#nodeArea').append('div').attr("id","node_"+i).attr("class","nodeBt")
327     .text(d.id)
328     .on("click", function(d){
329       if(this.className == "nodeBt active"){
330         d3.select("#status").text("").classed("statusBox", false);
331         d3.selectAll(".nodeBt").classed("active", false);
332         d3.select('#treeSVG').select("svg").remove();
333         d3.select("#globalSVG").style("display","block");
334         view_mode = 0;
335       }else{ //show ego-view
336         view_mode = 1;
337         d3.select("#status").text("back to social view").attr("class","statusBox");
338         d3.select("#globalSVG").transition().style("display","none");
339         d3.select('#treeSVG').style("display","block");
340         d3.selectAll(".nodeBt").classed("active", false);
341         this.className = "nodeBt active";
342       }
343     });
344   });
345 }
```

Put data filename here

Thank You

Q&A