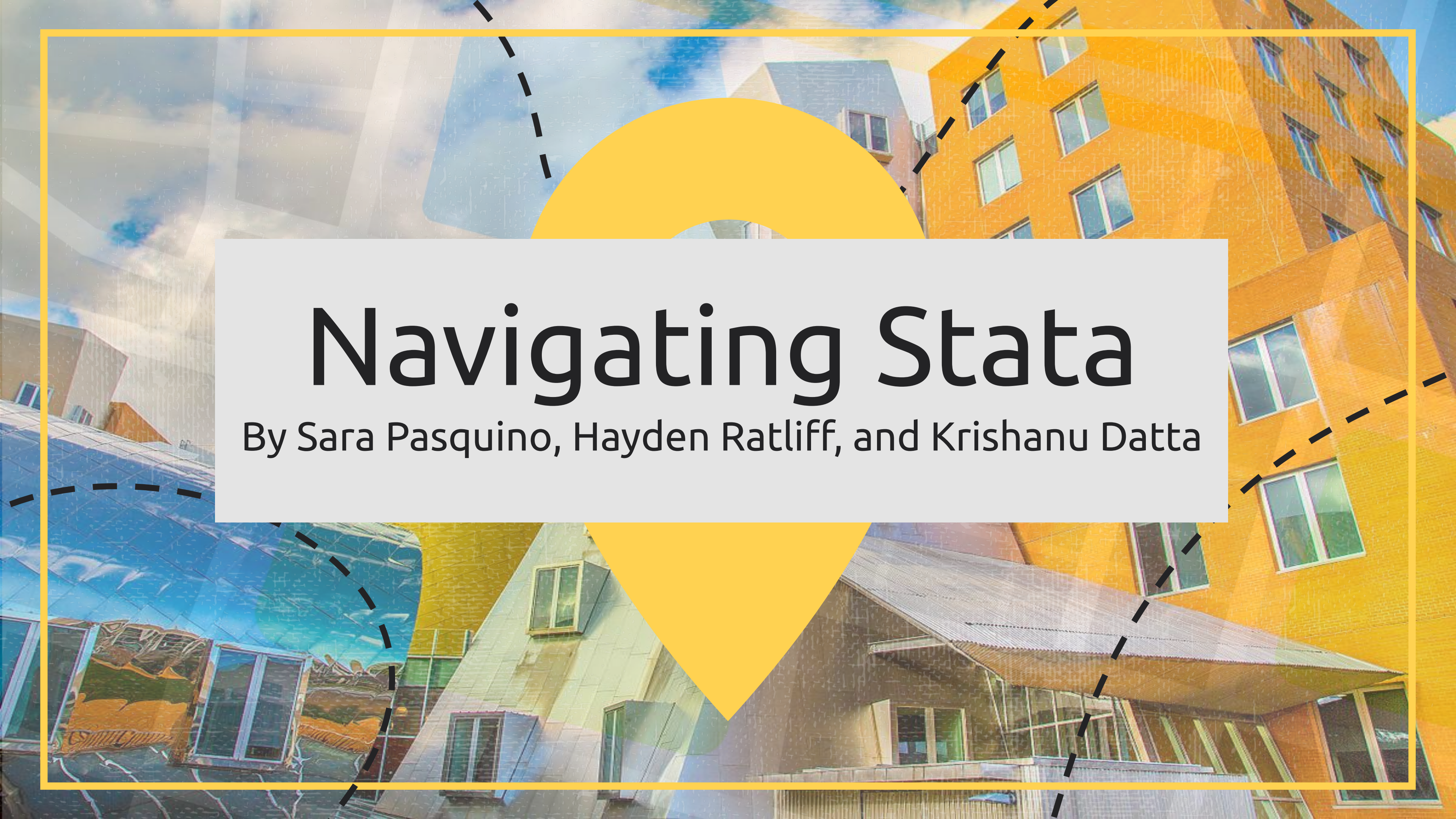


Navigating Stata

By Sara Pasquino, Hayden Ratliff, and Krishanu Datta

Project Overview

01 Introduction

02 Motivation and Related Work

03 Methodology

04 Results and Discussion

01

Introduction

Introduction

MIT’s Stata Center is a labyrinth.

Our plan:

1. Users input photo of surroundings
2. Use CV for location classification
3. Implement route optimization
4. Output guides users to destinations

02

Motivation and Related Work

Why does this matter?

- Streamline navigation within the Stata Center
- Enhances the visitor experience within complex indoor spaces
  - Airports, museums, shopping malls, etc.
- Opportunity for increased accessibility
  - Non native language speakers
  - Wheelchair-only routing integration

Literature Review

There are a couple approaches to image classification and image-based navigation in the literature that are relevant to our project.

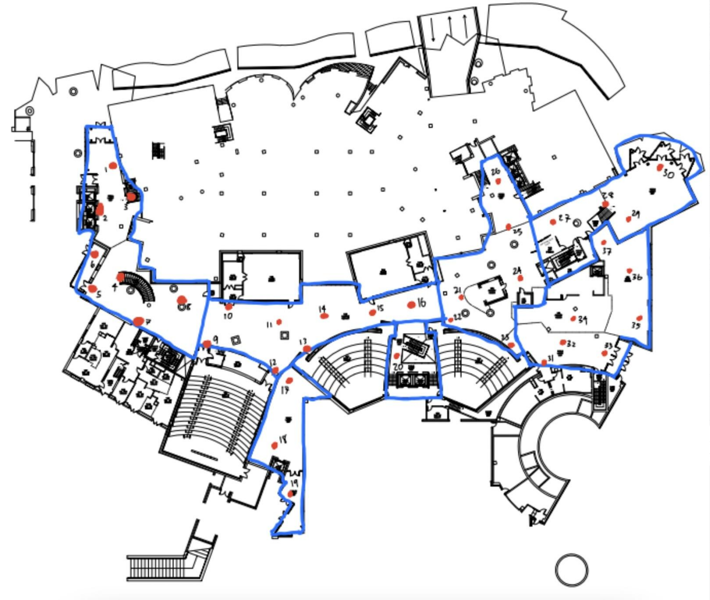
01 **Sensors (Morales et al. 2020)** explores the use of sensor-based systems in navigating complex buildings, highlighting the effectiveness of combining multiple data sources to improve location accuracy and system reliability.

02 **Meta-Explore (Hwang et al. 2023)** is a hierarchical vision-and-language navigation model that uses scene object spectrum grounding. This approach enables dynamic exploration and understanding of indoor environments, which is particularly relevant for navigating within structurally complex buildings like the Stata Center.

03

Methodology

Data Collection



- First floor of Stata
- 37 points to be labels of our model
    - Front of every classroom, restroom and exit/entrance
- 15s video at each location
- Extracted all frames from each video, and resized them yielding dataset of 10,189 points

CNN and Embedding Similarity Models

CNN

- Network depth and width
- Augmentation
  - Rotation
  - Flip
- Dropout layer

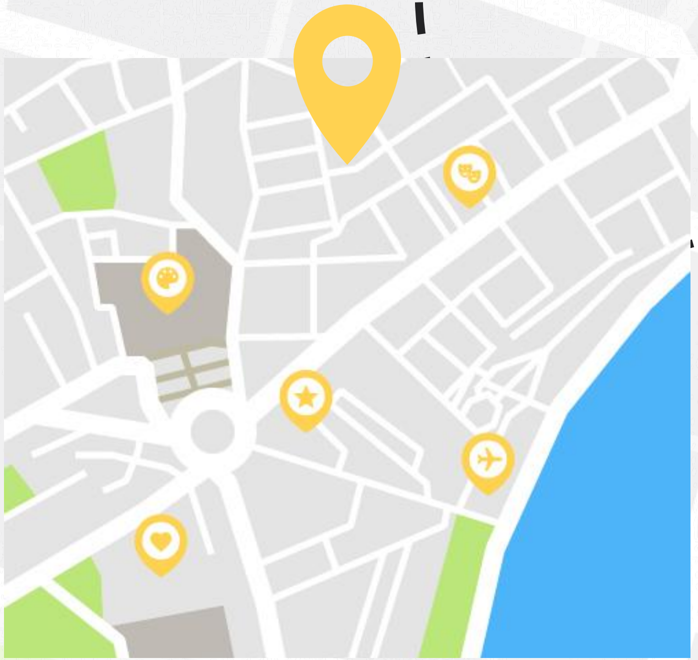
CLIP

Allows extraction of image embeddings capturing features such as edges, colors and shapes, as well as semantic meaning

- Max Score per Label, Train Data Vocabulary
- Max Score per Label, All Data Vocabulary
- Average Score per Label, Train Data Vocabulary
- Average Score per Label, All Data Vocabulary

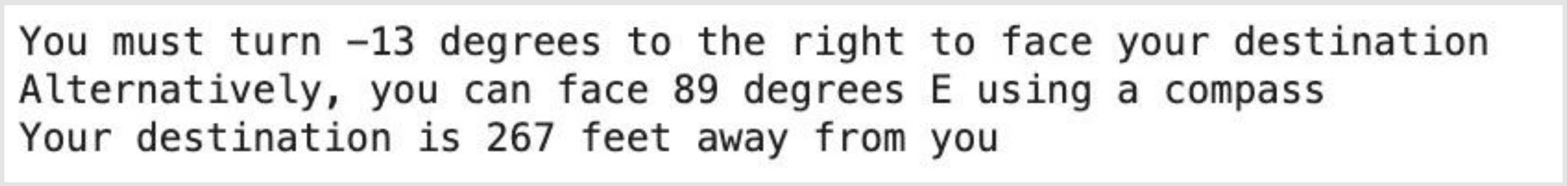
Foundation Models

- ResNet18
- AlexNet
- VGG16
- Simple classification head
  - Two hidden layers
  - Two dropout layers
- Deeper classification head
  - Three fully connected
  - Three batch normalization
  - Three dropout layers



Route Optimization

1. Extract coordinates from location labels
2. Calculate angle between start and end coordinates using trigonometry
3. Compute distance using Euclidean distance formula
4. Convert the distance in pixels to feet
5. Determine the angle the user must turn to face their destination, as well as the absolute heading on the compass
6. Relay instructions



```
You must turn -13 degrees to the right to face your destination
Alternatively, you can face 89 degrees E using a compass
Your destination is 267 feet away from you
```

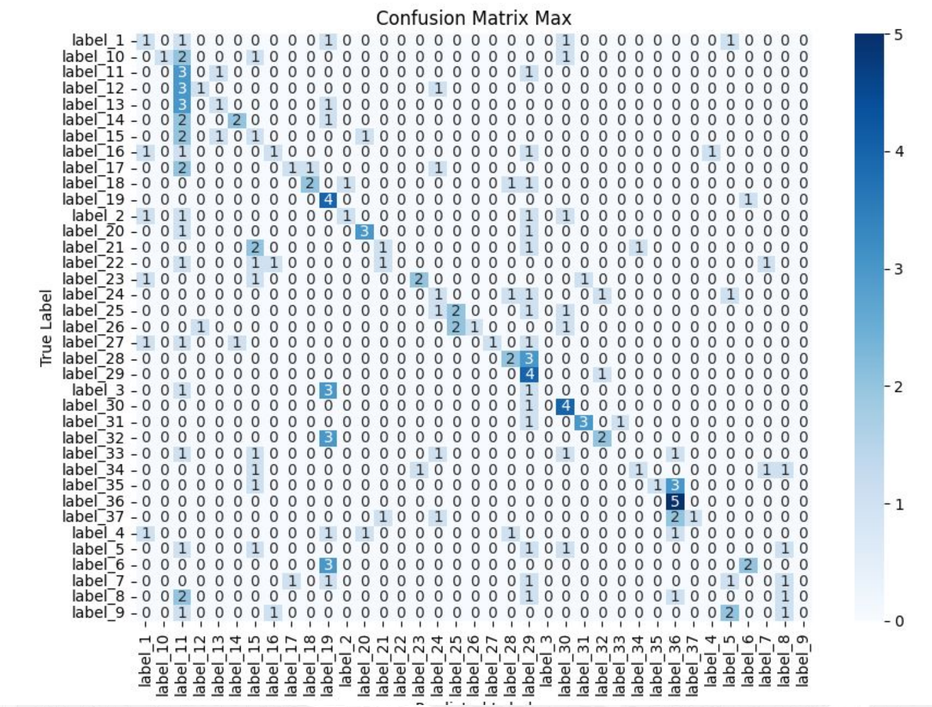
04

Results and Discussion

Model Performance (CNN and CLIP)

Model	Top-1 Accuracy (%)	Mean Error (ft)
Best Performing CNN	14.05	238.43
Embeddings (Max Score per Label, Train Data as Vocab)	30.27	194.88
Embeddings (Max Score per Label, Train Data as Vocab)	30.27	194.88
Embeddings (Average Score per Label, Train Data as Vocab)	12.97	326.46
Embeddings (Average Score per Label, Train Data as Vocab)	12.97	326.46

Model Performance (CNN and CLIP)



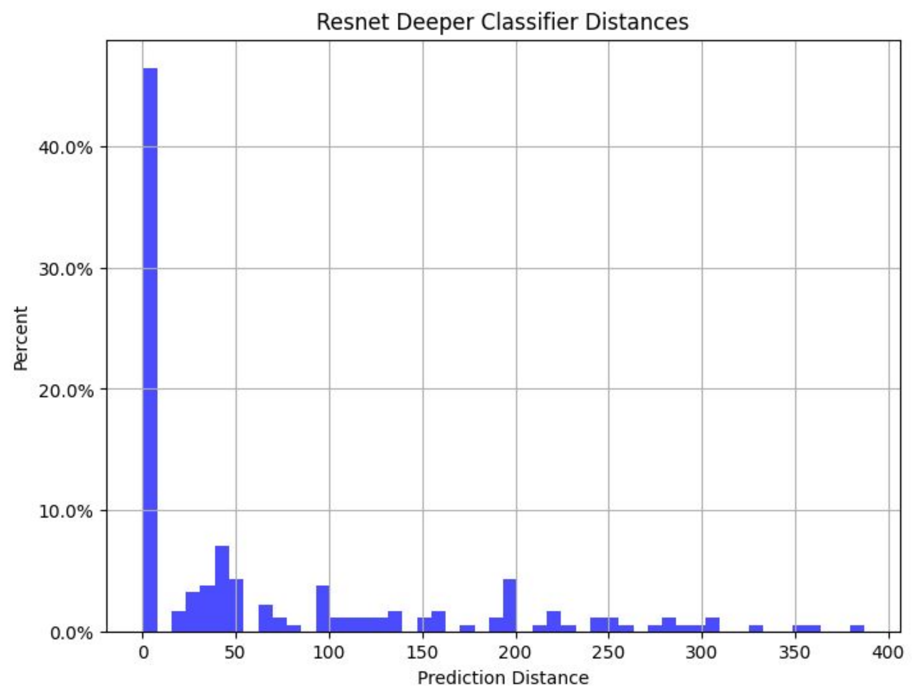
Confusion Matrix Max

	label_1	label_10	label_11	label_12	label_13	label_14	label_15	label_16	label_17	label_18	label_19	label_2	label_20	label_21	label_22	label_23	label_24	label_25	label_26	label_27	label_28	label_29	label_3	label_30	label_31	label_32	label_33	label_34	label_35	label_36	label_37	label_4	label_5	label_6	label_7	label_8	label_9
label_1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
label_10	0	1	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
label_11	0	0	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
label_12	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
label_13	0	0	3	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
label_14	0	0	2	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
label_15	0	0	2	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
label_16	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
label_17	0	0	2	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
label_18	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
label_19	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
label_2	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
label_20	0	0	1	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
label_21	0	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
label_22	0	0	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
label_23	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
label_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
label_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
label_26	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
label_27	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
label_28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
label_29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
label_3	0	0	1	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
label_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
label_31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	3	0	1	0	0	0	0	0	0	0	0	0	0
label_32	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
label_33	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
label_34	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0
label_35	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	0	0	0
label_36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
label_37	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0
label_4	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
label_5	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
label_6	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
label_7	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
label_8	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
label_9	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0

Model Performance (Foundation)

Model	Top-1 Accuracy (%)	Mean Error (ft)
AlexNet Simple	34.05	88.64
AlexNet Deeper	35.68	89.72
VGG Simple	31.89	83.71
VGG Deeper	32.97	85.60
ResNet Simple	40.54	78.88
ResNet Deeper	46.49	68.49

Model Performance (Foundation)



Key Findings

- Best model was ResNet Deeper
  - 46.5% Top-1 Accuracy
  - 63.8% Top-5 Accuracy
  - 29 ft. median error
- Foundation models greatly improved on embedding similarity models

Thank You!