From Simulation to Reality with Random Noise

Steven Luo
University of California, Berkeley

About

- Second year undergraduate at UC Berkeley
  - Computer Science, Data Science
  - Minor in Public Policy
  - Interests in Machine Learning, Computer Vision, Computer Ethics, Technology Policy

- Let's connect!
  - stevenfluo.github.io
  - linkedin.com/in/stevenfluo

Outline

1. Background
2. Research Overview
3. Applications to ICANN

Simulation-to-Reality

Real data can be

- ❌ Hard to gather
- ❌ Resource-intensive
- ❌ Expensive
- ❌ Time-consuming
- ❌ Potentially unsafe

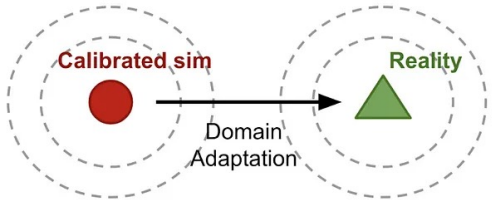
Simulated data is

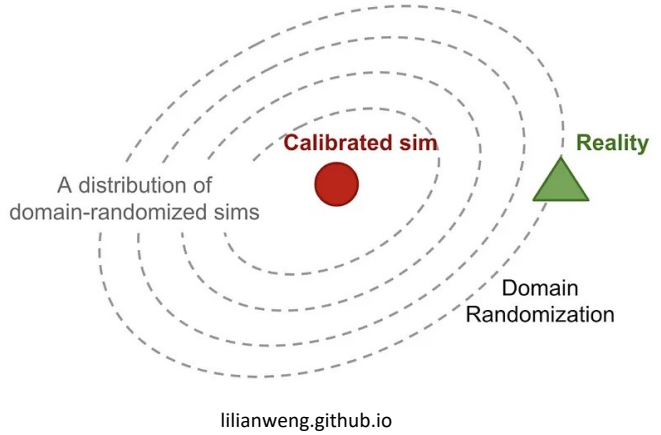
- ✅ Easy to gather large volumes
- ✅ Low-cost
- ✅ Fast
- ✅ No threats to physical health

Zhao, W., Queralta, J. P., & Westerlund, T. (2020, December). Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 737-744). IEEE.

Simulated Data Isn’t Perfect

- “Reality gap” — physical and visual differences
- Methods for sim2real transfer
  - Domain/dynamics randomization
  - Simulation environments
  - Domain adaptation



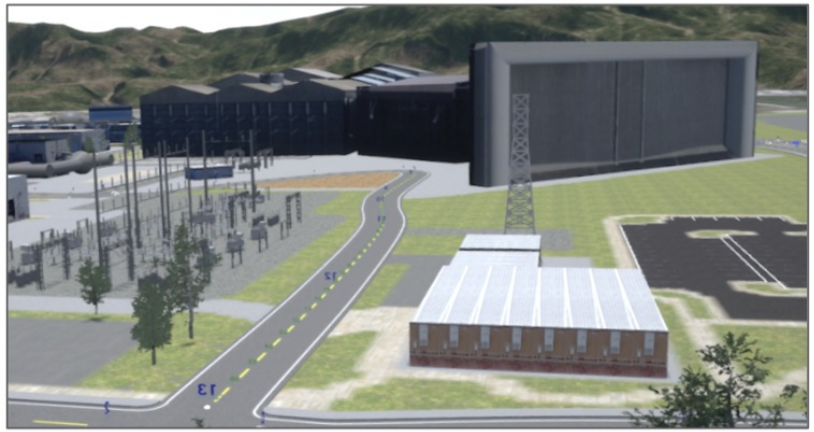


Goals & Questions

1. Use the RRAV testbed (rover) to conduct experiments
2. Validate AmesSim as a source of simulated road data
3. Quantify the effect of domain randomization on the real-world accuracy of classifiers trained in simulation

- How can we improve classifier robustness?
- Can data augmentation bridge the reality gap?
- What baseline can we establish for sim2real?

AmesSim and RRAV



Lopez-Francos, I. G., Mitchell, S. C., Lipkis, R., Vlastos, P. G., Mbaye, S., & Infeld, S. I. (2023). A Model-Based Systems Engineering Approach for Developing an Autonomous Rover Testbed. In AIAA SCITECH 2023 Forum (p. 1894).

Data

Figure 1: Images from AmesSim (left) and the `real_data` dataset (right).

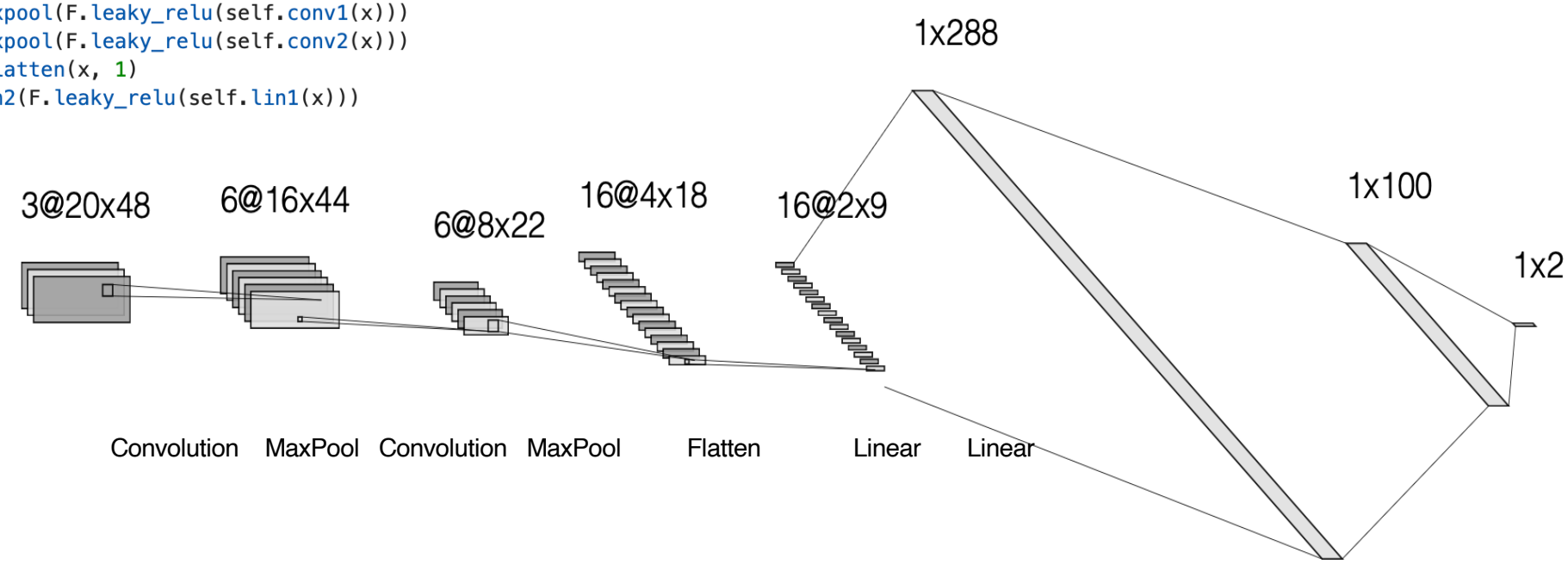
Figure 2: Sample images from each dataset. Clockwise from top left: `none_sim`, `sharpcj_sim`, `shot_sim`, and the `sharp_sim` dataset.

Model Information: Left/Right Classifier

```python
class ConvNet(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)   # 5x5, 6 channels
        self.maxpool = nn.MaxPool2d(2, 2) # 2x2 max pool
        self.conv2 = nn.Conv2d(6, 16, 5)  # 5x5, 16 channels
        self.lin1 = nn.Linear(288, 100)   # fully connected to hidden layer
        self.lin2 = nn.Linear(100, 2)     # fully connected to logit output

    def forward(self, x):
        x = self.maxpool(F.leaky_relu(self.conv1(x)))
        x = self.maxpool(F.leaky_relu(self.conv2(x)))
        x = torch.flatten(x, 1)
        x = self.lin2(F.leaky_relu(self.lin1(x)))
        return x
```

Hyperparameters
Batch size: 128
Learning rate: 1e-3
Epochs: 25
Adam optimizer



Results

Dataset	Augmentations	Self Accuracy	Sim. Accuracy	Real Accuracy
`none_sim`	• 48,000 images • No augmentations	98.03%	N/A	64.10%
`sharp_sim`	• 528,000 images • Sharpness	99.75%	99.78%	69.55%
`sharpcj_sim`	• 528,000 images • Sharpness, Jitter	99.92%	99.94%	71.80%
`shot_sim`	• 528,000 images • Sharpness, Jitter, Poisson	98.59%	99.51%	78.59%

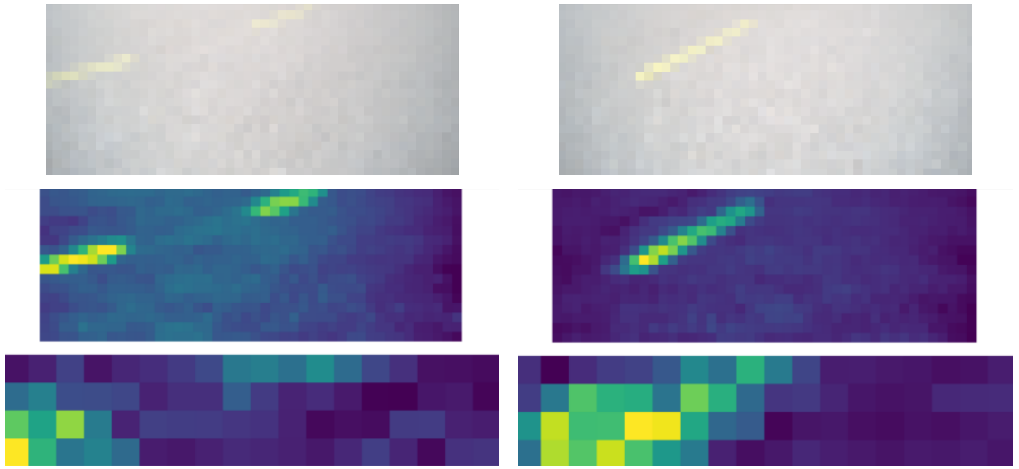
Real dataset: 4,800 real-world images captured by RRAV camera in different positions and times of day

Discussion

- Validated AmesSim and RRAV testbed for sim-to-real transfer learning
- Well-selected augmentations can bridge the reality gap, improve performance and classifier robustness
- Applications include use as low-fidelity runtime monitor for airplane taxiing/centerline following
- Baseline for evaluating AI-generated data augmentation

Other Contributions

- Real-time data collection script, saved many hours of labeling data
- Livestreaming of convolutional layer outputs
- Packaged helper functions and analysis methods in Python module, contributing to RRAV repository



Confidence: 0.67219561
(Predicted, Actual): (L, R)

Confidence: 0.99999988
(Predicted, Actual): (R, R)

Feature maps from a model trained on simulated data with Poisson noise applied. The top image is the preprocessed, raw image; the middle and bottom images are the channel-averaged outputs from the first and second convolutional layers.

ICANN Applications

- Simulate cybersecurity threats/attacks on DNS infrastructure
  - Broadly, machine learning for DNS security and anti-abuse
  - Support cyber range testing
- Create synthetic DNS usage data
  - Better DNS trend prediction, routing optimization
  - Improved data augmentation and cleaning in collection of real DNS data
  - Support ongoing research in root server system traffic analysis
- Combat abuse and fraud
  - Inferential Analysis of Maliciously Registered Domains (INFERMAL)
  - ICANN DNS Security Threat Mitigation Program
- DNS physical infrastructure object recognition

Acknowledgements

Rory Lipkis, Adrian Agogino, Guillaume Brat, the RRAV group, RSE & Intelligent Systems Division, N-269 interns, and everyone I was fortunate enough to meet during my internship.

Denise Hochbaum, Alfredo Calderon, Deborah Escalera, and all of the staff & mentors at ICANN. Thank you!

Contact me:
sfluo@berkeley.edu
linkedin.com/in/stevenfluo

Appendix

Datasets

Left column:

- Row 1: Raw real image
- Row 2: preprocessed raw
- Row 3: saturated version of row 2

Right column:

- Row 1: Raw AmesSim image
- Row 2: *none_sim* (preprocessed raw)
- Row 3: *sharp_sim* (blur)
- Row 4: *sharpcj_sim* (blur + colorjitter)
- Row 5: *shot_sim* (blur + colorjitter + Poisson noise)

Images from reality (left) and AmesSim (right).

Data Limitations