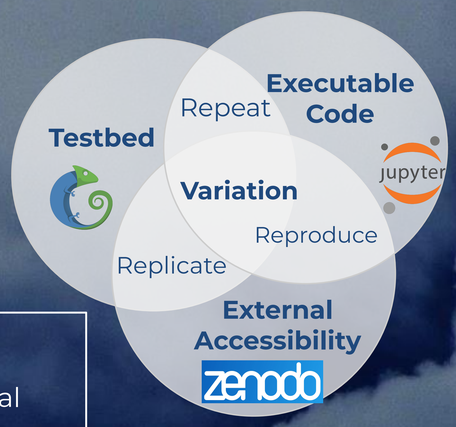
Future-Proof Your Research: Designing for Replicability and Reproducibility

Isabel Brunkan, Zhuo Zhen (Advisor), Kate Keahey (Advisor), Levent Tokosz (Advisor)

Computer Science is Agile. Iteration after iteration, moving quickly to solve new problems, uncover new questions, find the next big thing. Hardware, software, libraries, datasets, experiments - technology becomes outdated almost as soon as it's released. So why save code? Why share code? For replication, to verify results. For education, to train the next generation. For variation, to discover new insights. We seek to understand how to package experiments to encourage experiment exploration and longevity by replicating an AlexNet reproduction.

The Challenge

Research replicability is crucial but computer science research has a unique challenge due to **resource variety, availability, upgrading** and **data** and **code accessibility**.



How do you achieve...?

Repetition:
Same Experiment + Original Artifacts

Replication:
Same Experiment + Recreate Artifacts

Reproduction:
Same Experiment Idea + New Artifacts

Variation:
Repeat/Replicate + Modification

AlexNet: convolutional neural network model that motivated research into using GPUs and CPUs for deep learning

Goal: Replicate the original AlexNet model

In Practice: Package an experiment reproducing AlexNet on the Stanford Dogs dataset.

Repetition v Replication

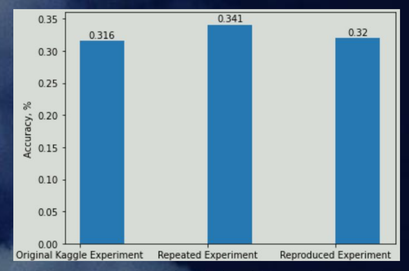
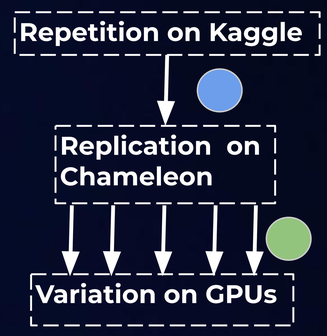


Figure 1: Results of Original Kaggle Experiment, the Repetition on Kaggle, and the Replication on Chameleon



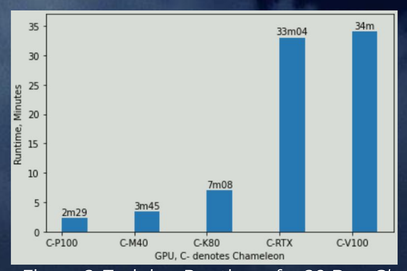


Figure 2: Training Runtimes for 20 Dog Classes on Chameleon GPUs

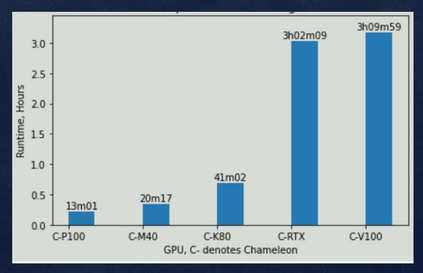


Figure 3: Training Runtimes on Multiple Chameleon GPUs for 120 Dog Classes

Repetition of the original AlexNet model was not possible, but the Kaggle-hosted reproduction was repeatable.

Replication occurred on Chameleon. Results were within 0.4% accuracy.

Variation, was introduced, running the experiment on different GPUs and with different dataset lengths and additional images.

Hardware Implementation

Table 1: Differences in AlexNet, Chameleon and Kaggle Experiments

	Original AlexNet + ImageNet	Chameleon + Stanford Dogs	Kaggle + Stanford Dogs
Images	1.2 million (Training)	20,000	20,000
Classes	1,000	120	120
GPU	2 (NVIDIA GTX 580)	P100, M40, K80, RTX-6000, V100	1 Tesla P100

Evaluation and Extensions

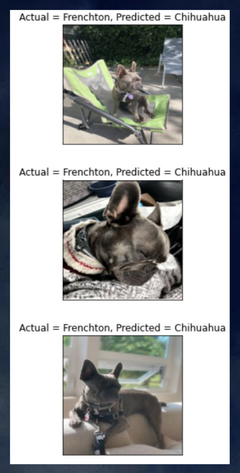
→ Choose stable, accessible datasets for replicability

Tensorflow's Datasets API is faster and easier to use than Kaggle's API, which requires credentials

→Experiment with your own data + transfer learning

Use additional data and pre-trained models to increase accuracy.

My frenchton puppy was classified as a chihuahua



Conclusions


Title of Your Experiment/Paper

Some description

1. Create Experiment Container

```
export OS_PROJECT_NAME="Chameleon Reproducibility Research"
export OS_REGION_NAME="CHI@TACC"
export AVAILABLE_DEVICE='GPU'
```

Create a reservation with a shell script

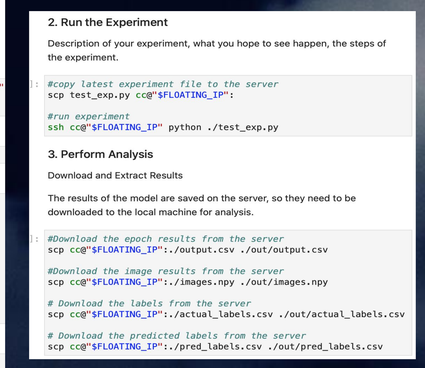
```
source ./reservation.sh
```

Provision resources

```
# Ensure your Jupyter keypair is present
key_pair_upload

stack_name="$RESOURCE_NAME"

openstack stack create "$stack_name" --wait \
  --template stack.yaml \
  --parameter floating_ip="$FLOATING_IP_ID" \
  --parameter reservation_id="$RESERVATION_ID" \
  --parameter key_name="$USER-jupyter"

wait_ssh "$FLOATING_IP"
```

Configure the instance

```
ssh cc@"$FLOATING_IP" <setup-gpu.sh
```


2. Run the Experiment

Description of your experiment, what you hope to see happen, the steps of the experiment.

```
#copy latest experiment file to the server
scp test_exp.py cc@"$FLOATING_IP":

#run experiment
ssh cc@"$FLOATING_IP" python ./test_exp.py
```

3. Perform Analysis

Download and Extract Results

The results of the model are saved on the server, so they need to be downloaded to the local machine for analysis.

```
#Download the epoch results from the server
scp cc@"$FLOATING_IP":./output.csv ./out/output.csv

#Download the image results from the server
scp cc@"$FLOATING_IP":./images.npy ./out/images.npy

# Download the labels from the server
scp cc@"$FLOATING_IP":./actual_labels.csv ./out/actual_labels.csv

# Download the predicted labels from the server
scp cc@"$FLOATING_IP":./pred_labels.csv ./out/pred_labels.csv
```

Container Scripts **Experiment Scripts**

Package for the Future by Separating Container Setup and Experiment Scripts

1. **Enable Experiment Transfer**
   a. **Container Scripts:** Can be reused for different experiments and easily adjusted for different hardware, especially those that haven't been invented yet.
   b. **Experiment Scripts:** Easily adjust or add variation to the experiment without affecting your container set up.

2. **Increase Readability**
   Code is hidden in scripts which can be expanded to direct focus as needed.

Citations

Aditya K., Nityananda J., Bangpeng Y.and Li, F. Novel dataset for Fine-Grained Image Categorization. *First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

Krizhevsky A., Sutskever I., and Hinton G. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Feitelson, D. G. (2015). From Repeatability to Reproducibility and Corroboration. *ACM SIGOPS Operating Systems Review, 49*(1), 3-11. doi:10.1145/2723872.2723875

Acknowledgements