



# Accuracy-Constrained Efficiency Optimization and GPU Profiling of CNN Inference for Detecting Drainage Crossing Locations

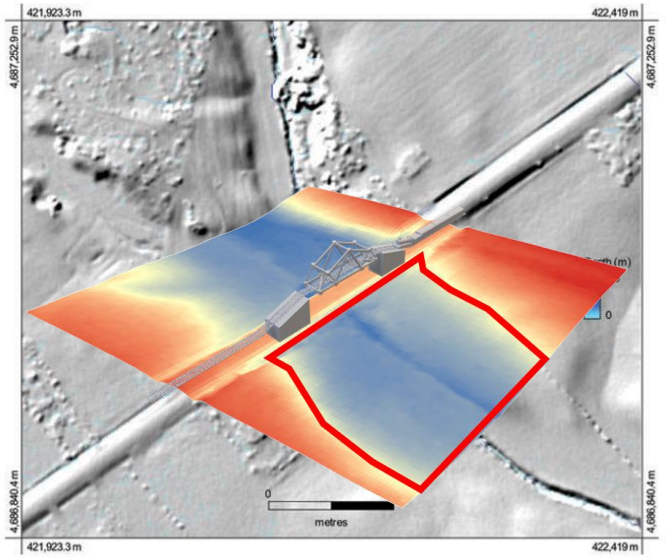
Yicheng Zhang[1], Dhroov Pandey[2], Di Wu[3], Turja Kundu[2],
Ruopu Li[3], Tong Shu[2]

[1]*University of California, Riverside*
[2]*University of North Texas*
[3]*Southern Illinois University*



# Detecting Drainage Crossing Locations

- Drainage crossing detection is **critical** for environmental management.
  - Nutrient transport.
  - Assessing sustainability of downstream.
  - Tracking watershed-scale water quality, etc.

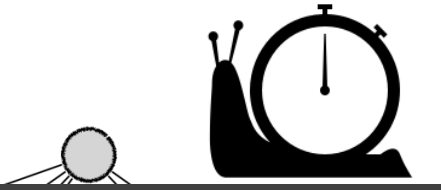



***Drainage crossing in digital view***

***Drainage crossing in real world***

# Challenges for Detecting Drainage Crossing

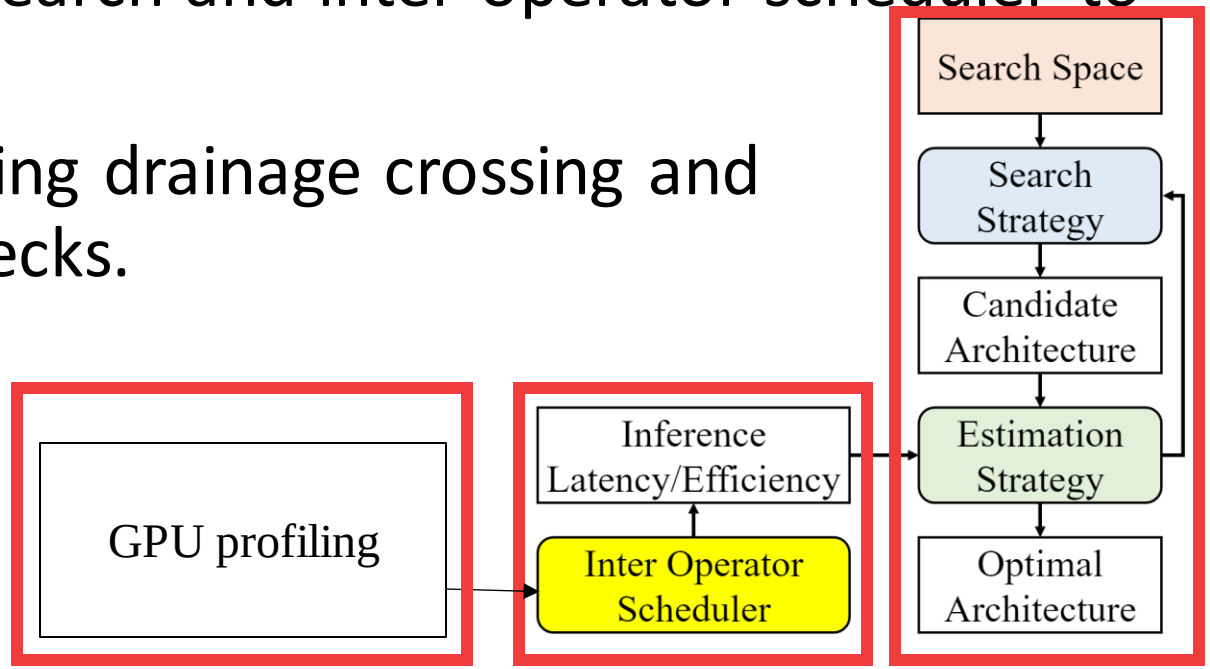
- Inaccuracies in detection via elevation-based delineation.
- Slow detection via traditional CNN-based detection.



**Research question**: can we detect drainage crossing *accurately* and *efficiently*?

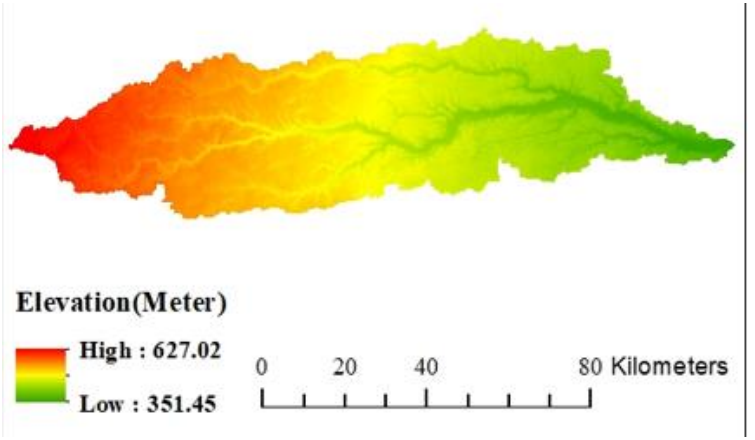
# Our Work Overview

- Developing a set of optimized CNN models for detecting drainage crossing accurately.
- Performing neural architecture search and inter-operator scheduler to make detection more efficiently.
- Profiling performance for detecting drainage crossing and identifying performance bottlenecks.



# Study Area and Datasets

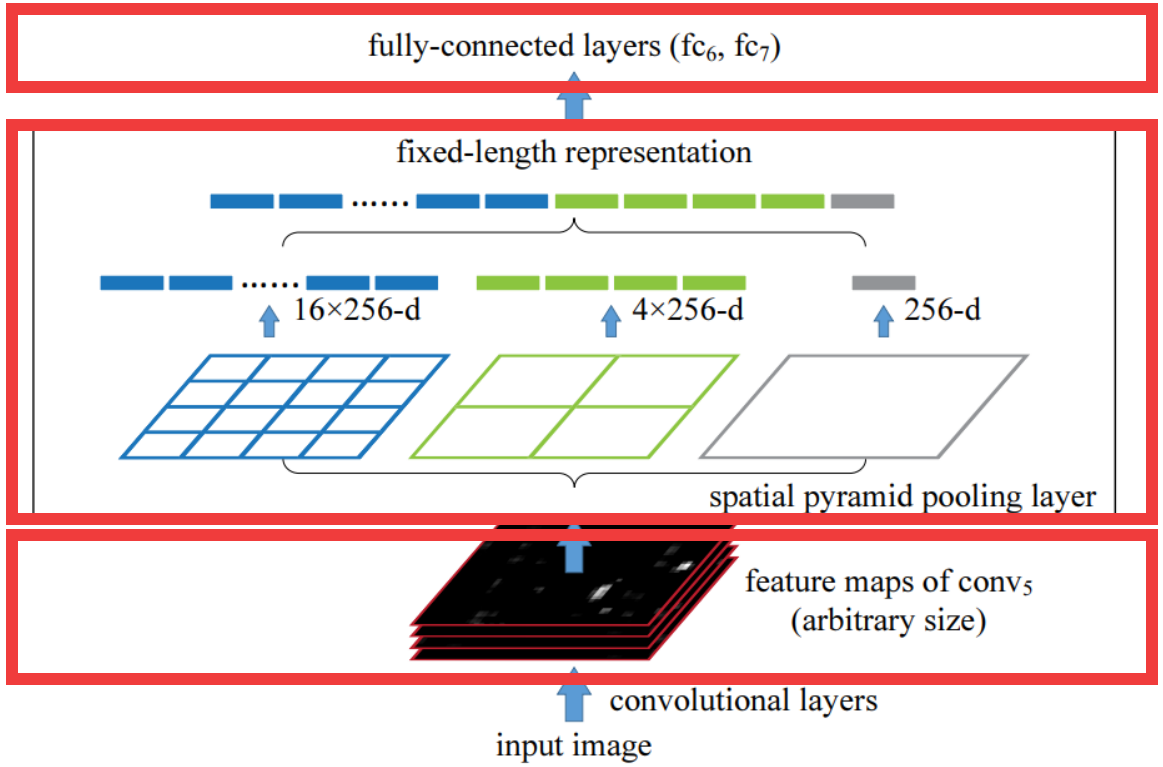
- Study Area.
  - West Fork Big Blue Watershed, Nebraska.
    - Dominated by intensive agriculture.
    - Relatively level topography.
    - Dense road networks.




- Datasets.
  - 2022 culverts were located manually.
  - Each sample contains one culvert which is with size of 100-meter by 100-meter.

# Accuray-Constrained CNN models

- Spatial Pyramid Pooling (SPP-Net).
  - Feature extraction.
  - SPP layer.
    - Sub-regions of various sizes.
    - Max-pooling within each sub-region.
    - Spatial information at different scales.
  - Fully connected layers.



# Neural Architecture Search for SPP-Net

- Neural Architecture Search (NAS).
  - Automatically tuning model structure to achieve optimal performance.
  - Neural Network Intelligence framework, Retiarii [1].
- Determining search space.
  - Feature engineering: Filter size of the first convolutional layer as ranging from 1 to 9 (1, 3, 5, 7, 9).
  - SPP layer: Filter sizes for the first SPP layer, spanning from 1 to 5 (1, 2, 3, 4, 5).
  - Fully-connected layers: Feature size for two fully-connected layers within the following ranges: 128, 256, 512, 1024, 2048, 4096, and 8192.

[1]. Zhang, Quanlu, et al. "Retiarii: A deep learning {Exploratory-Training} framework." *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*. 2020.

# Neural Architecture Search for SPP-Net

- Three candidate models.
  - C=Convolution and the subscripts of C stand for the size of the filter, numbers of filters, and stride.
  - F=Fully-connected and the number of neurons in F are shown in subscript.
  - P=Pooling, and its subscripts represent filter size and stride.
  - SPP=SPP layer, and its subscripts represent filter size.

| Model | Hyper-parameters Settings | Average Precision |
|---|---|---|
| Original SPP-Net | $C_{64,3,1} - P_{2,2} - C_{128,3,1} - P_{2,2} - C_{256,3,1} - P_{2,2} - SPP_{4,2,1} - F_{1024}$ | 95.00% |
| SPP-Net # 1 | $C_{64,5,1} - P_{2,2} - C_{128,3,1} - P_{2,2} - C_{256,3,1} - P_{2,2} - SPP_{4,2,1} - F_{1024}$ | 96.10% |
| SPP-Net # 2 | $C_{64,3,1} - P_{2,2} - C_{128,3,1} - P_{2,2} - C_{256,3,1} - P_{2,2} - SPP_{5,2,1} - F_{4096}$ | 96.70% |
| SPP-Net # 3 | $C_{64,3,1} - P_{2,2} - C_{128,3,1} - P_{2,2} - C_{256,3,1} - P_{2,2} - SPP_{5,2,1} - F_{2048}$ | 97.40% |

# Inter-operator scheduling for SPP-Net

- Inter-operator scheduling (IOS) [2].
  - Optimizing the tasks via increasing parallelism in the algorithm.
    - E.g. tiled matrix multiplication.
  - Increasing data parallelism.
    - E.g. Batching.
- Speed-up evaluation.

| Model | Sequential Inference Latency | Optimized Inference Latency |
|---|---|---|
| Original SPP-Net | 0.512 ms | 0.268 ms |
| SPP-Net # 1 | 0.419 ms | 0.379 ms |
| SPP-Net # 2 | 0.295 ms | 0.236 ms |
| SPP-Net # 3 | 0.562 ms | 0.427 ms |

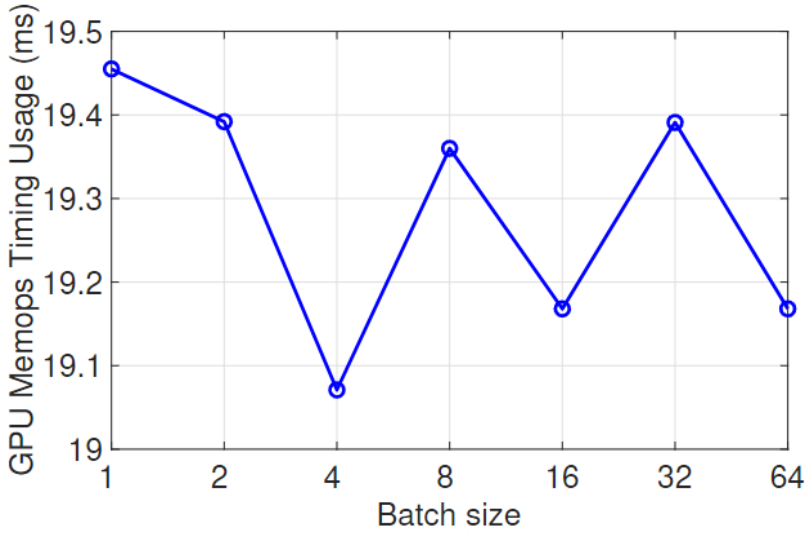
[2]. Ding, Yaoyao, et al. "IOS: Inter-operator scheduler for CNN acceleration." Proceedings of Machine Learning and Systems 3 (2021): 167-180.

# GPU Performance Profiling

- Profiling CNN inference and identify the performance bottleneck.
- Performance analysis on
  - GPU memory.
  - CUDA API utilization.
  - CUDA kernels.
- Profile GPU performance via Nvidia Nsight system.

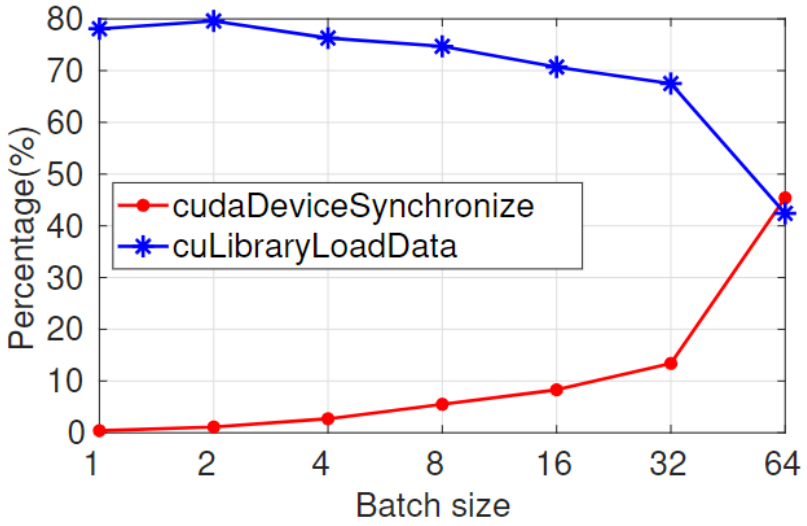
# GPU Performance Profiling – GPU memory

- GPU memory utilization when changing batch size.
  - Profile GPU performance via Nvidia Nsight.
  - Testing batch sizes ranging from 1 to 64 (1, 2, 4, 8, 16, 32, and 64).
- Observation: stable utilization when changing batch size.
  - GPU memory does not constrain the inference timing.



# GPU Performance Profiling – CUDA API

- Focusing on two primary API.
  - *cuLibraryLoadData* and *cudaDeviceSynchronize*.
  - Testing batch sizes ranging from 1 to 64 (1, 2, 4, 8, 16, 32, and 64).
- Bottleneck may due to limited PCIe bandwidth between CPU and GPU.
  - *CudaDeviceSynchronize* surpasses *cuLibraryLoadData* when batch size increases to 64.



# GPU Performance Profiling – CUDA Kernels

- Focusing on three CUDA kernels.
  - Testing batch sizes ranging from 1 to 64 (1, 2, 4, 8, 16, 32, and 64).
- Observation: When batch size increases, MatMul decrease and Conv increase.
- SPP-Net primarily composed of convolutional layers.

| Batch Size | Matrix Multiplication (%) | Pooling (%) | Conv (%) |
|---|---|---|---|
| 1 | 41.6 | 14.1 | 7.7 |
| 2 | 34.8 | 14.4 | 9.7 |
| 4 | 39.9 | 13.5 | 9.5 |
| 8 | 34.8 | 13.7 | 10 |
| 16 | 18.1 | 17.1 | 16.6 |
| 32 | 15.7 | 14.7 | 13.4 |
| 64 | 7.4 | 8.6 | 77.2 |

# Conclusion

- Accuracy-constrained efficiency optimization for detecting drainage crossing.
- Neural architecture search for accuracy-optimized CNN models.
- Inter-operator scheduler to accelerate speed of CNN inference.
- Detailed GPU performance analysis for CNN inference.

# Thank you!

## Any questions?

Yicheng Zhang

yzhan846@ucr.edu

https://yichez.site