New York Water Science Center AI

Daniel Beckman
Martyn Smith
Joint venture between NYWSC and Web Informatics Mapping-AI
BIO

• Graduate Student and the University of Colorado
• AI Engineer
  • Focused on applying AI in the real world
• Symbolic Artificial Intelligence/Data Mining/Machine Learning /Deep Learning
• Convolutes and Reinforcement Learning
• Mechatronic Engineering
Our Approach:

• Select a real world problem with broad applications
• Look for space that hasn’t been explored
• Work the problem using both extant technology and novel designed solutions to find the most cost effective solution
Camera based gaging

In Theory:
• Cheap
• Fully remote
• Scalability
• Durable
• Accurate
• Low power
• Effective
Camera based gaging

Gaps:

- Sustainable power scheme
- Safe power cycling
- Is it Edge capable? (Full capability at the Sensor)
- Do we have command and control? (Can we take control of the sensor in real-time)
- Chip wear
- AI reproducibility
- Supply lines
- Data glut
- Portability (Not a one site solution)
- Telemetry
- Light conditions
- Weather conditions
- Water Hazards
- Black box / Verification
Words of Wisdom and some books

• A convincing demonstration of correctness being impossible as long as the mechanism is regarded as a black box, our only hope lies in not regarding the mechanism as a black box.

• Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better.

• A picture may be worth a thousand words, a formula is worth a thousand pictures.

- Edsger Dijkstra
AI reproducibility

• Simple is not so simple.
• AI should only do a single task.
• Even so, they miss all the time.
• The trick, if it can be achieved, is to make it miss small compared to a margin of error.
• This has to be a maxim with AI and so, it informs the entire sensor design.
• Excellent AI is an illusion. There is always a trick you the user is not seeing.
• AI Engineers refer to this a “creative manipulations”*
Gage-cam
Gage-cam
Gage-cam
Gage-cam
Gage-cam’s components

Brain – Ras pi A+

• Tested with pretrained, tensor based AI networks on:
  • Raspi Zero  Failed
  • Raspi A+   Passed
  • Raspi B+   Passed
  • Raspi 4    Passed

• Robust testing of Light weight Operating Systems
• Power consumption
• Heat production
• Speed
Gage-cam’s components

Camera – Ras pi cam

• Tested on the Ras pi Zero and Ras pi A+
• Tested in a multitude of light conditions
• Tested power cycling
Gage-cam’s components

Cycling – Witty-Pi

• Ras pi SBC have no built in safe startup/shutdown
• Tested the witty pi and witty pi mini.
• Requires some firmware hacks
Gage-cam’s components

Power – Voltaic systems 9w solar panel and 12,800 MaH through charging battery

• Tested on the Ras pi Zero and Ras pi A+
• Weeks of off solar life achieved by effective power cycling
• Looked at voltage monitoring
Gage-cam’s components

Data – Adam Kelly’s COCO Synth data method
• Tested in both multi class forms and discreet object forms
• High degree of training accuracy
• Fast
• Portable
• Small investment in time
Gage-cam’s components

AI – U-net based segmented Convolutional Neural Network (CNN)
• Tested on everything
• Winner out of several AI schemes
• Low power
• High accuracy
• Simple
The Failures

Tech that either worked but didn’t serve the project or had issues that disqualified them

- Arduino Boards
- Coral Dev Boards
- Ras pi Zero
- Ras pi 4
- Ras pi B+
- Sparkfun Artemis boards
- Products made by Seeed
- EAST Algorithm paired with Tesseract
- YOLO Algorithm
- Mask R-CNN Algorithm
- Eigen vector based computer vision
- Witty Pi
Other

Tech that was researched adjacent to the project
• 3D printing Form factors / Material science of 3d Printing
• Small part fabrication
• Weather Proofing
• Thermal isolation, heat sinking, and heat mitigation
• AI automated validation
• Polarization
• AWS containers
• AWS EC2 ML builds
• Jupyter Notebook and AI Prototyping
Topic List

• Gage Cam
• Synthetic Data
• U-Nets
• Mask R-CNN
• East Algo and Teserract
• YOLO
• PCA (Eigen Vector based AI)
• Raspberry Pi
• SparkFun Boards
• Coral Boards
• Seeed Boards
• Jupyter Notebook and AI Prototyping
• 3D Printing
• Sensor Fabrication
• Cameras
• Witty Pi
• Voltaic Systems
U-Net
U-Net
def down_block(x, filters, kernal_size = (5, 5), padding = 'same', strides=1):
    c = keras.layers.Conv2D(filters, kernal_size, padding=padding, strides=strides, activation = 'relu')(x)
    c = keras.layers.Conv2D(filters, kernal_size, padding=padding, strides=strides, activation = 'relu')(c)
    p = keras.layers.MaxPool2D((2, 2), (2, 2))(c)
    return c, p

def up_block(x, skip, filters, kernal_size = (5, 5), padding = 'same', strides=1):
    us = keras.layers.UpSampling2D((2, 2))(x)
    concat = keras.layers.Concatenate()([us, skip])
    c = keras.layers.Conv2D(filters, kernal_size, padding=padding, strides=strides, activation = 'relu')(concat)
    c = keras.layers.Conv2D(filters, kernal_size, padding=padding, strides=strides, activation = 'relu')(c)

    return c

def bottleneck(x, filters, kernal_size = (5, 5), padding = 'same', strides=1):
    c = keras.layers.Conv2D(filters, kernal_size, padding=padding, strides=strides, activation = 'relu')(x)
    #c = LeakyReLU(alpha=0.05)(c)
    c = keras.layers.Conv2D(filters, kernal_size, padding=padding, strides=strides, activation = 'relu')(c)
    #c = LeakyReLU(alpha=0.05)(c)
    return c
```python
## UNET Architecture

def UNet():
    # Filters per layers
    # f = [16, 32, 64, 128, 256]
    f = [20, 40, 80, 160, 320]

    # Data shape entering the convolution
    inputs = keras.layers.Input((image_size, image_size, 1))

    # Input layer
    p0 = inputs

    # Down bloc encoding
    c1, p1 = down_block(p0, f[0])
    c2, p2 = down_block(p1, f[1])
    c3, p3 = down_block(p2, f[2])
    c4, p4 = down_block(p3, f[3])

    # Switch over layer to up bloc decoder
    bn = bottleneck(p4, f[4])

    # Up bloc decoding
    u1 = up_block(bn, c4, f[3])
    u2 = up_block(u1, c3, f[2])
    u3 = up_block(u2, c2, f[1])
    u4 = up_block(u3, c1, f[0])

    # Autoencoder egress layer. Flatten and any perceptron layers would succeed this layer
    outputs = keras.layers.Conv2D(1, (1, 1), padding='same', activation='sigmoid')(u4)

    # Keras model output
    model = keras.models.Model(inputs, outputs)

    return model
```
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
Matterport Mask Residual-CNN (R-CNN)
Matterport Mask R-CNN
Matterport Mask R-CNN
Matterport Mask R-CNN
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
EAST Algorithm
(Efficient and Accurate Scene Text Detector)
EAST Algorithm
Tesseract Algorithm
Optical Character Recognition
Tesseract Algorithm
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
YOLO Algorithm
You Only Look Once
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
Eigen Vector Based Computer Vision
Eigen Vector Based CV
Eigen Vector Based CV

```
Iteration 34, loss = 1.53210336
Validation score: 0.052632
Validation score did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
(105, 2)
(105, )
Random: 2.0556323398806323
True: 0.6666711504608415

--Reduced dimensions to 6--
100 450 2
(2, 326400)
y true = b'10.07' y hat = b'10.18'
b'10.07'
[(72, 10.2), (15, 10.0), (12, 10.1), (1, 9.9), (0, 20.0), (0, 19.9), (0, 19.8), (0, 19.7), (0, 19.6), (0, 19.5), (0, 19.4), (0, 19.3), (0, 19.2), (0, 19.1),
Process finished with exit code 0
```
Eigen Vector Based CV

Scree Plot

Eigenvalues from PCA

Principal Component

Eigenvalue
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
Synthetic data
Synthetic data
Synthetic data
Synthetic data
Synthetic data
Synthetic data
COCO Data method
Synthetic data
Synthetic data
Synthetic data
Synthetic data
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based Al)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
Ras Pi Single Board Computer
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
Coral Boards
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
Artemis Boards
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
3D Printing
Filament

Prototyping
PLA

Durable/Outdoor
PET-G

Machine parts
Poly Carbonate

Bamboo

ABS

Carbon Fiber
Filament

Prototyping
- PLA

Durable/Outdoor
- PET-G

Machine parts
- Poly Carbonate
- Bamboo
- ABS
- Carbon Fiber
Prusa Mk3
Ultimaker
3DP 400
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
What is a hardware accelerator?

It’s a share-space shop setup for rapid prototyping where engineers can kick the tires on a build with low overhead.

Who has hardware accelerators?
- Most Universities
- NASA N220 “Space Shop”
- USDA Brookings Area Makerspace
- HAX
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
Cameras
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
Witty Pi
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
Voltaic Systems
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems
How Does Gage Cam Work?
How Does Gage Cam Work?

- Site Selection
How Does Gage Cam Work?

- Training Imagery Capture
How Does Gage Cam Work?

- Synth Data Generation
How Does Gage Cam Work?

- Network Training
How Does Gage Cam Work?

- Deployment
How Does Gage Cam Work?

• Calibration

Train the AI to identify the precise dimensions of the exposed staff plate and then derive the height of the submerged portion to get a stage reading.
How Does Gage Cam Work?

- Calibration

Obtain the height and the width from the mask in pixels.

Height = 53 px
Width = 16 px
How Does Gage Cam Work?

- Calibration

Using the known width in inches, derive a px to inch ratio

Width = 16 px = 4 inches ⇒ 1 px = 4/16 inches
How Does Gage Cam Work?

- Calibration

Convert exposed height.

\[ \text{Height} = 53 \text{ px} = 53 \times \frac{4}{16} \text{ inches} \]

Or

\[ \text{Height} = 13.25 \text{ inches} \]
How Does Gage Cam Work?

- Calibration

The theory: Subtract total height by exposed height.

Height = 53 px = 53 \times 4/16 inches
Or
Height = 13.25 inches

Total Height - Exposed height = Stage Height
33 inches - 13.25 inches = 19.75 inches
How Does Gage Cam Work?

- Activation
Topic List

- Gage Cam
- Synthetic Data
- U-Nets
- Mask R-CNN
- East Algo and Teserract
- YOLO
- PCA (Eigen Vector based AI)
- Raspberry Pi
- SparkFun Boards
- Coral Boards
- Seeed Boards
- Jupyter Notebook and AI Prototyping
- 3D Printing
- Sensor Fabrication
- Cameras
- Witty Pi
- Voltaic Systems