Artificial Intelligence Methods for Social Good

M3-4 [Machine Learning]:

Deep Learning and Its Applications

08-537 (9-unit) and 08-737 (12-unit)

Instructor: Fei Fang
feifang@cmu.edu
Wean Hall 4126

Outline

- Basics of Neural Network
- Convolutional Neural Network
- Faster RCNN
- Applications
  - Detecting wildlife and poachers from UAV videos

Learning Objectives

- Understand the concept of
  - Neural Networks
  - Convolutional Neural Networks
- Describe key ideas of Faster RCNN
- List a few applications of deep learning models for social good
- Know how to find the algorithm/solver/package

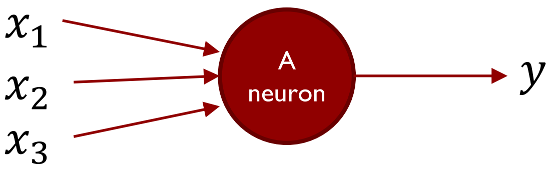
Basics of Neural Networks

- Broad applications
  - Image classification
  - Question answering
  - Machine translation
  - Spam filter
  - AlphaGo
- Long history
  - Inspired by how human brain works: electrical signals travel along axons triggering a chemical connection at another neuron's dendrite
  - Algorithms developed in the 80s and 90s
  - Thrive in the last few years
    - Due to (1) massive data; (2) significantly improved computing power; (3) advanced optimization technique

Basics of Neural Networks

- A neuron completes a simple operation of input variables: apply a (simple) non-linear function on a linear transformation of input

$$y = f(h^T x + b)$$

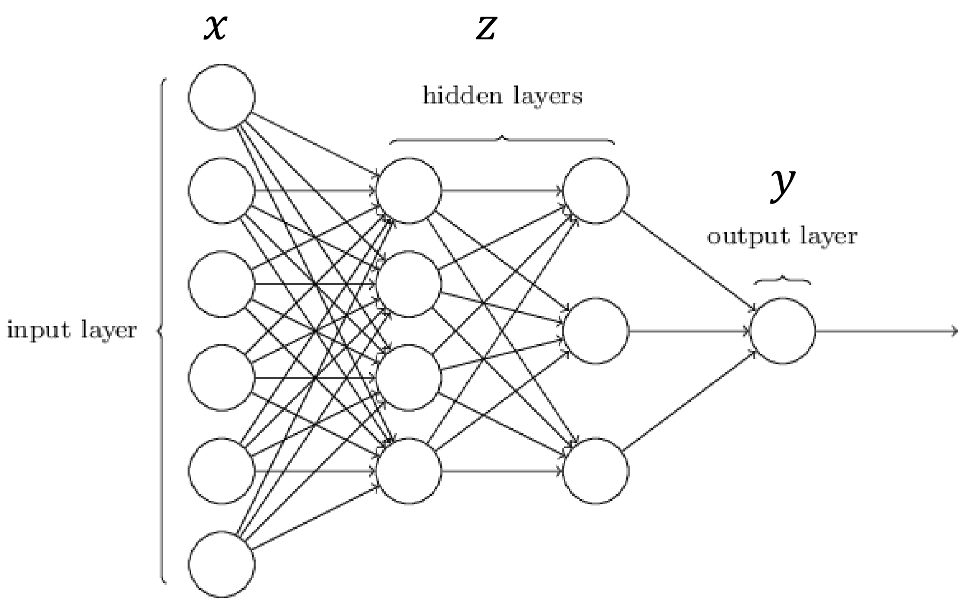


- Common choice of f
  - $f(x) = x$
  - tanh: $f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
  - sigmoid: $f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$
  - ReLU (rectified linear unit): $f(x) = \max\{0, x\}$

Basics of Neural Networks

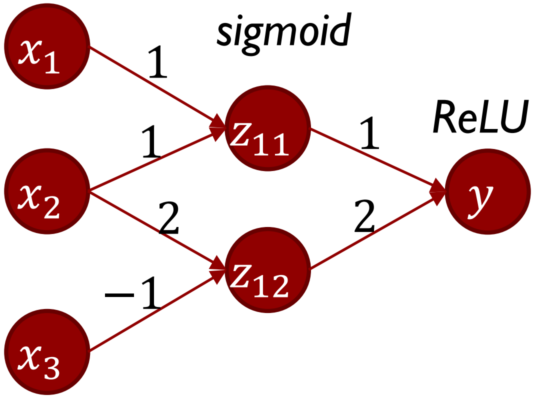
- Logistic regression: assume a simple relationship between input and output:

  $x \rightarrow$ a single neuron $\rightarrow y = f(x) = \sigma(h^T x + b)$
- Neural network: $x \rightarrow$ layers of neurons $\rightarrow y = f_\theta(x)$



Quiz 1

- For the provided neural network, what is the value of y when $x = (1,0,1)$? Hint: $\sigma(1) \approx 0.731, \sigma(-1) \approx 0.269$
  - A: 1.269
  - B: 1
  - C: 0.731



Basics of Neural Networks

- Logistic regression: find the value of h and b so as to minimize the loss function, e.g., the residual sum of squares given observed value $\hat{y}$ and predicted value $f(x) = \sigma(h^T x + b)$
- Train a network: find the value of the parameters of all neurons in the network (denoted as θ) so as to minimize the loss function defined on observed value $\hat{y}$ and output of network $f_\theta(x)$

$$\min_\theta \sum_{i=1}^{m} l(f_\theta(x^i), \widehat{y^i})$$

(Note: $f_\theta(x)$ is not necessarily a scalar value, it can be a vector)

- Solve the optimization problem: Stochastic gradient descent (SGD)
  - Initialize θ
  - Repeat until convergence
    - Randomly shuffle examples in the training set
    - For $i = 1 \dots m$
      - □ $\theta \leftarrow \theta - \alpha \nabla_\theta l(f_\theta(x^i), \widehat{y^i})$

Basics of Neural Networks

- Compute $\nabla_\theta l(f_\theta(x^i), \widehat{y^i})$: Back Propagation
  - Forward pass: given x^i and current value of θ, compute z^i and $f_\theta(x^i)$ starting from first layer to last layer
  - Backward pass: given $f_\theta(x^i)$ and y, compute $\nabla_{\theta_k} l\left(f_\theta(x^i), \widehat{y^i}\right)$ starting from last layer to first layer
    - θ_k =parameters in layer k
    - Using the chain rule $\frac{\partial f(g(x))}{\partial x} = \frac{\partial f(g(x))}{\partial g(x)} \frac{\partial g(x)}{\partial x}$
  - In practice: packages like Tensorflow, atugrad (Python) can automatically compute gradients once you specify the network structure

Basics of Neural Networks

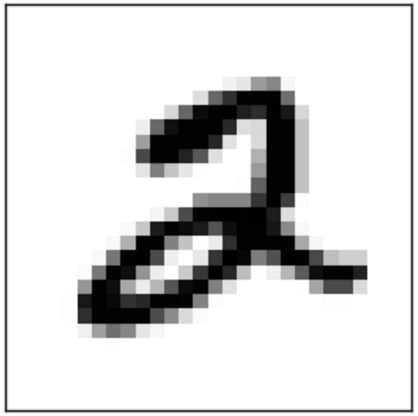
- Why neural networks are powerful: even with a single hidden layer, a neural network can approximate any function over x
- Why training neural networks is challenging: $\sum_i l(f_\theta(x^i), \widehat{y^i})$ is often highly non-convex w.r.t. θ for multi-layer NN
- Deep learning: approximately represent the relationship between output labels y and input feature values x using an NN with multiple hidden layers and train the NN using data

Basics of Neural Networks

- Connections and Interpretations
  - Embedding/representation/feature selection
    - View deep learning as finding a good embedding / finding an appropriate representation of the data / selecting non-linear features
  - Bayesian Network
    - Input nodes, hidden nodes, output nodes

Basics of Neural Networks

- Fully connected layer / dense layer: every node in layer k is connected with every node in layer $k+1$
- Given a set of grayscale images, each described by 28 by 28 pixels, if we treat the intensity of each pixel as an input feature, and the output is the probability that the image is representing a hand written digit 2, how many parameters are needed with one fully connected hidden layer with 100 nodes?



Basics of Neural Networks

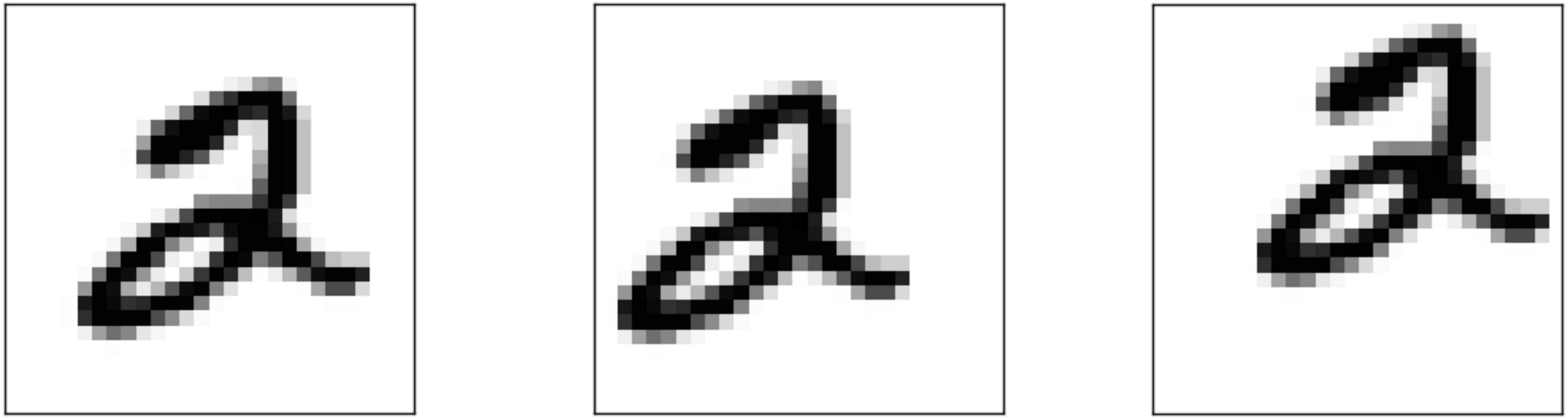
- Going beyond simple neurons and layers
  - A neuron can represent any function $y = f(x_1, x_2, \ldots)$
    - E.g., MaxPool: $y = \max_{i \in \mathrm{y}_{\rightarrow}} x_i$
    - No need to represent a too complicated function: can be approximated by stacking neurons
  - A layer may not consist of an array of neurons
    - A function which takes as input the vector $x \in \mathbb{R}^n$ and produces as output the vector $y \in \mathbb{R}^m$
    - E.g., Softmax Layer $(m = n)$: Often used for classification tasks or tasks that requires a probability distribution as output
      - □ $y_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$
      - □ (Recall Quantal Response $q_i = \frac{e^{\lambda * AttEU_i}}{\sum_j e^{\lambda * AttEU_j}}$)
  - Practical requirement: Can apply backpropagation (can easily infer output in forward pass, can easily compute gradient in backward pass)

Outline

- Basics of Neural Network
- Convolutional Neural Network
- Faster RCNN
- Applications
  - Detecting wildlife and poachers from UAV videos

Convolutional Neural Networks

- How to represent the relationship between an input image and an output label (is it a hand written digit 2?) efficiently?

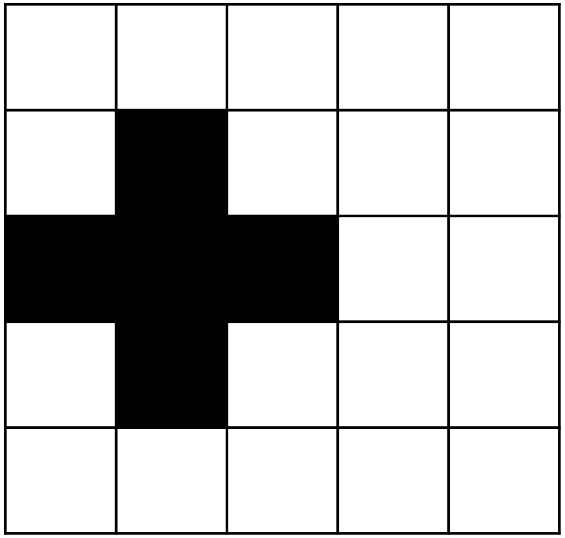


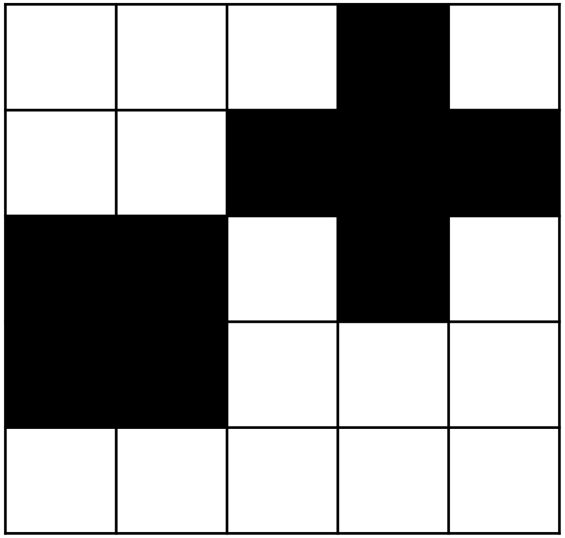
Convolutional Layer

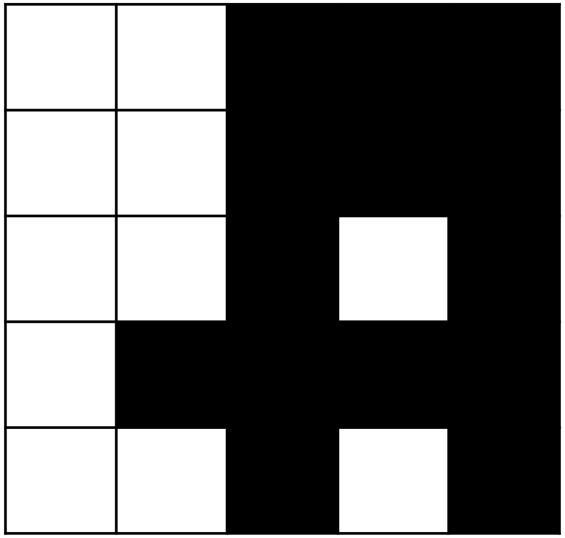
- Motivation
  - Reduce parameters
  - Enforce invariance to shift
- Key ideas
  - Construct a “filter”, apply the filter to every subregion of the image (equivalently, sliding the filter)

Convolutional Layer

- Finding "+"
  - 5 by 5 pixels, B/W image, allow for noise in irrelevant area
- Quiz 2: Construct an NN for the task using only basic neurons. How many hidden layers do you use?



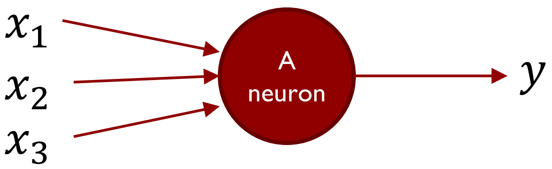




Recall

- A neuron completes a simple operation of input variables: apply a (simple) non-linear function on a linear transformation of input

$$y = f(h^T x + b)$$



- Common choice of f
  - $f(x) = x$
  - tanh: $f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
  - sigmoid: $f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$
  - ReLU (rectified linear unit): $f(x) = \max\{0, x\}$

Convolutional Layer

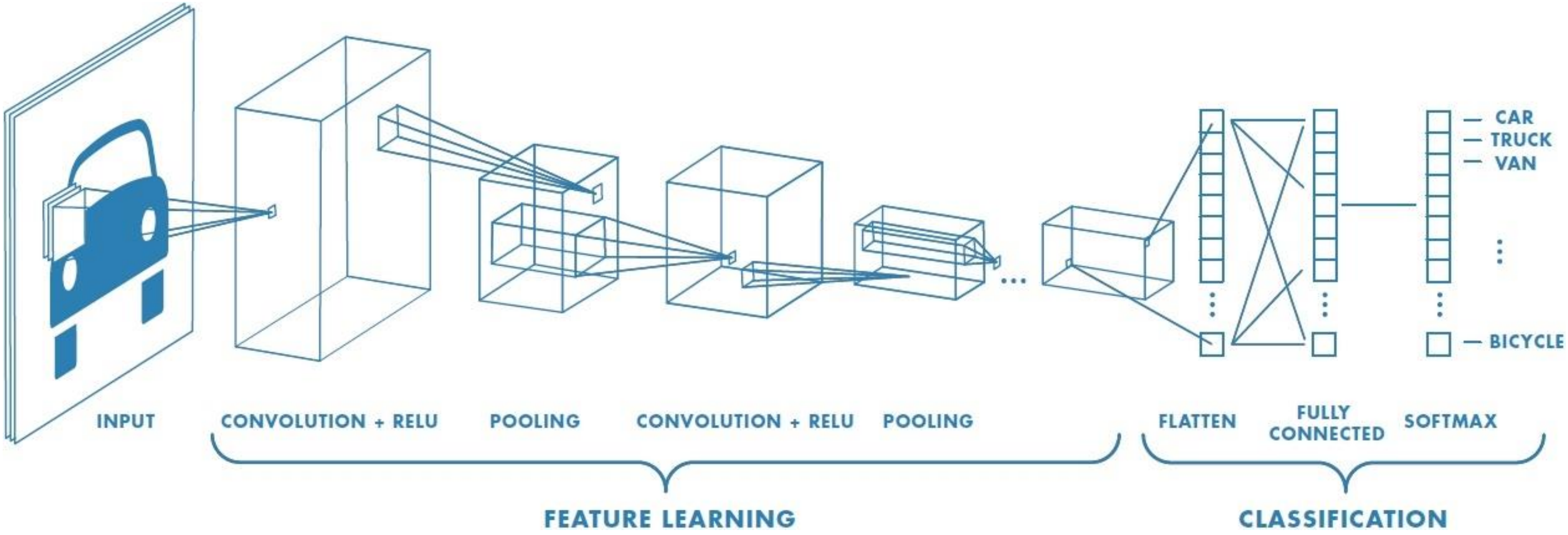
- Extend to general grayscale images
  - $z_{k+1} = z_k * w$ represent a set of nodes induced by one filter w
  - If w is a 3 by 3 matrix
    - $z_{k+1}^{11} = z_k^{11} w^{11} + z_k^{12} w^{12} + \cdots + z_k^{33} w^{33}$
    - $z_{k+1}^{12} = z_k^{12} w^{11} + z_k^{13} w^{12} + \cdots + z_k^{34} w^{33}$
    - $z_{k+1}^{ij} = z_k^{ij} w^{11} + z_k^{i,j+1} w^{12} + \cdots + z_k^{i+2,j+2} w^{33}$
    - …
  - Note: if we follow exact the standard convolution operation in image processing, it should be $z_{k+1}^{ij} = z_k^{ij} w^{33} + z_k^{i,j+1} w^{32} + \cdots + z_k^{i+2,j+2} w^{11}$, but it is equivalent to flipping the filter
- Extend to multi-band images (e.g., RGB images)
  - w is a 3 by 3 by #band matrix
- A single filter is not enough, often has many filters
  - Each filter leads to an output image

Convolutional Layer

- Sometimes zero-padding is used to get "images" of the same size in the next layer
- A convolutional layer is often followed by ReLU layer (element wise) and MaxPooling layer (region wise) in image related tasks

Convolutional Neural Network

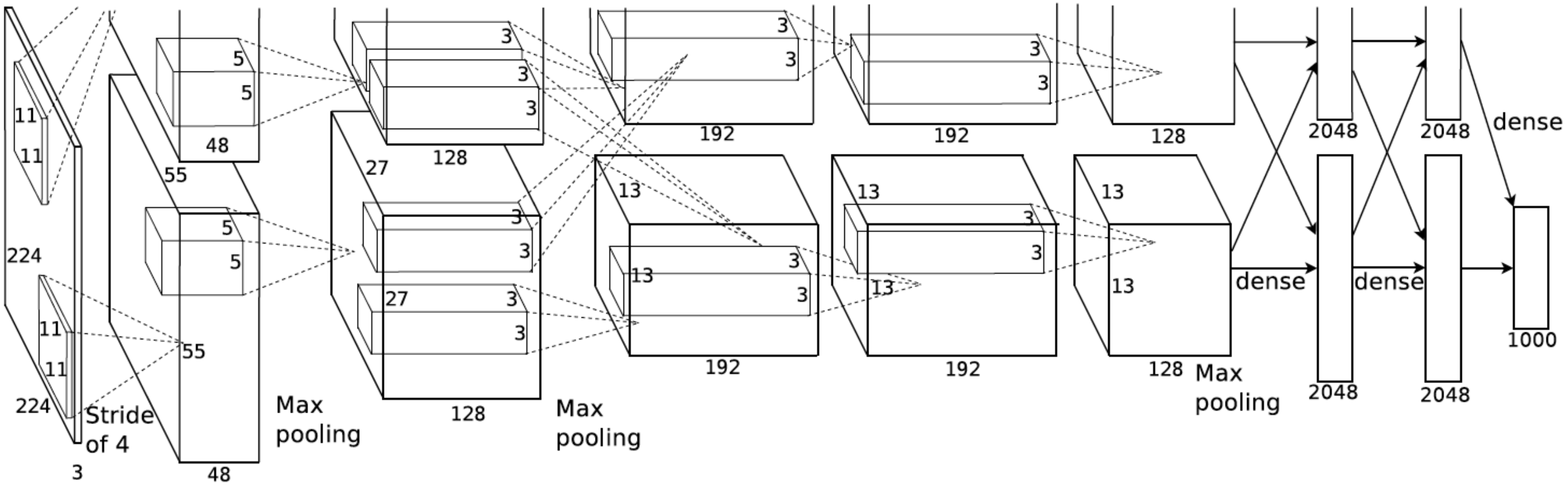
- CNN: An NN with convolutional layers



https://www.mathworks.com/discovery/convolutional-neural-network.html

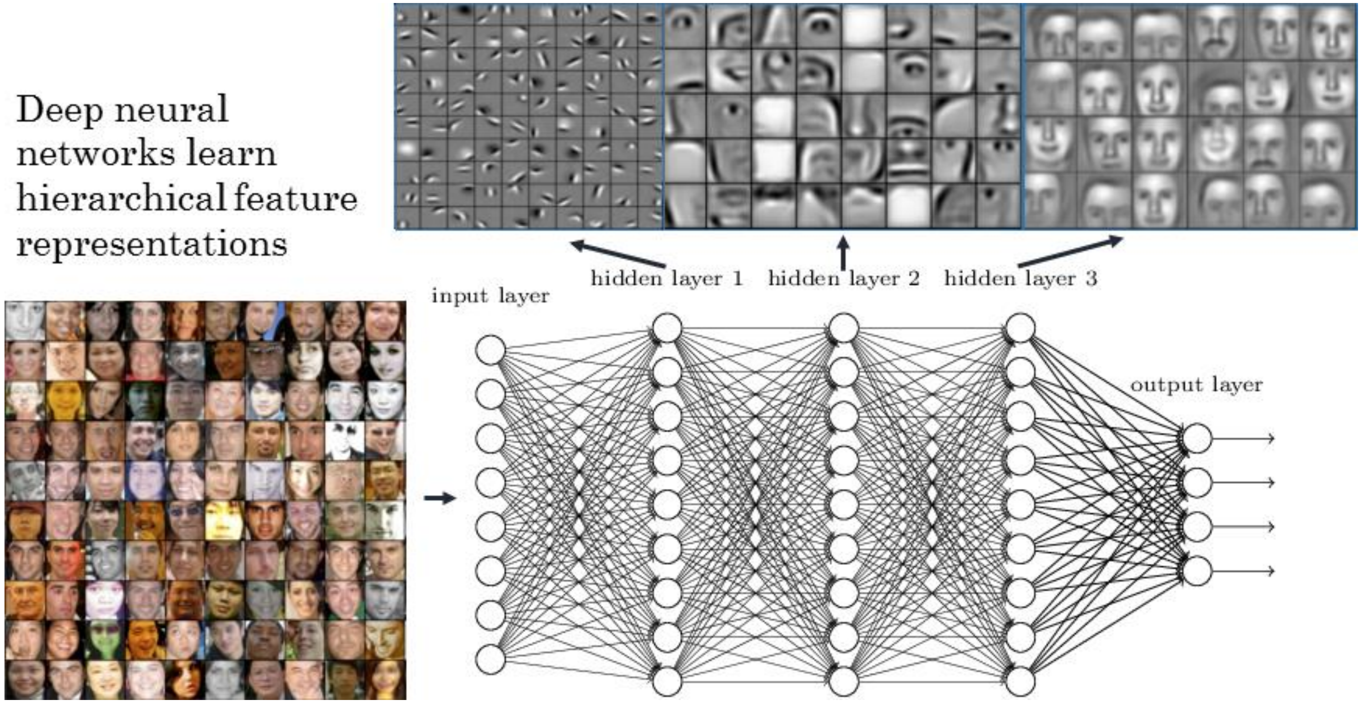
Convolutional Neural Network

- AlexNet: Won image classification competition on ImageNet by a large margin (Krizhevsky, Sutskever, Hinton, 2012)



Convolutional Neural Network

- Lower layers are reusable!



https://www.rsipvision.com/exploring-deep-learning/

Convolutional Neural Network

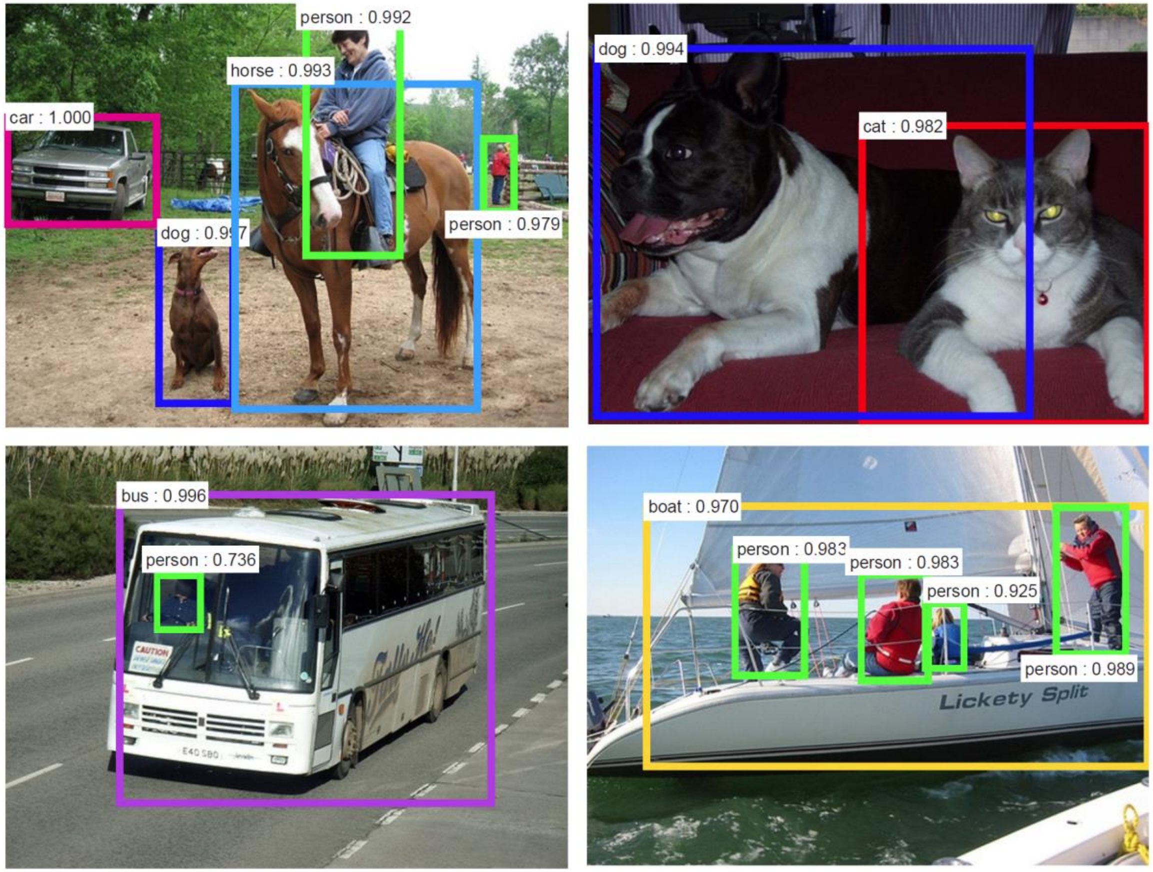
- Typical workflow
  - Train a network NN_A for Problem A with dataset D_A
  - Build a network NN_B for Problem B, with same lower layer architecture
  - Initialize the lower layers parameters of NN_B using NN_A
  - Refine NN_B with dataset D_B for Problem B

Outline

- Basics of Neural Network
- Convolutional Neural Network
- Faster RCNN
- Applications
  - Detecting wildlife and poachers from UAV videos

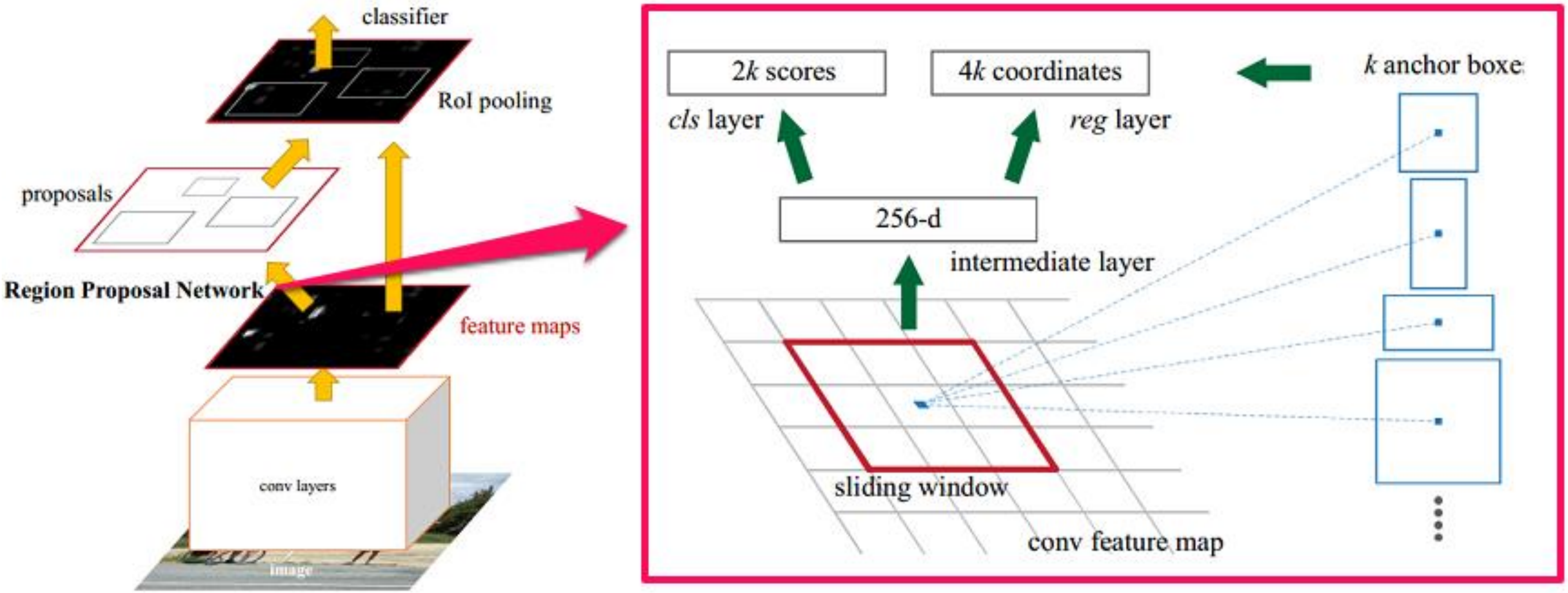
Faster R-CNN

- Detect and locate object of interest (Ren et al, 2016)



Faster R-CNN

- Detect and locate object of interest



https://andrewliao11.github.io/object_detection/faster_rcnn/

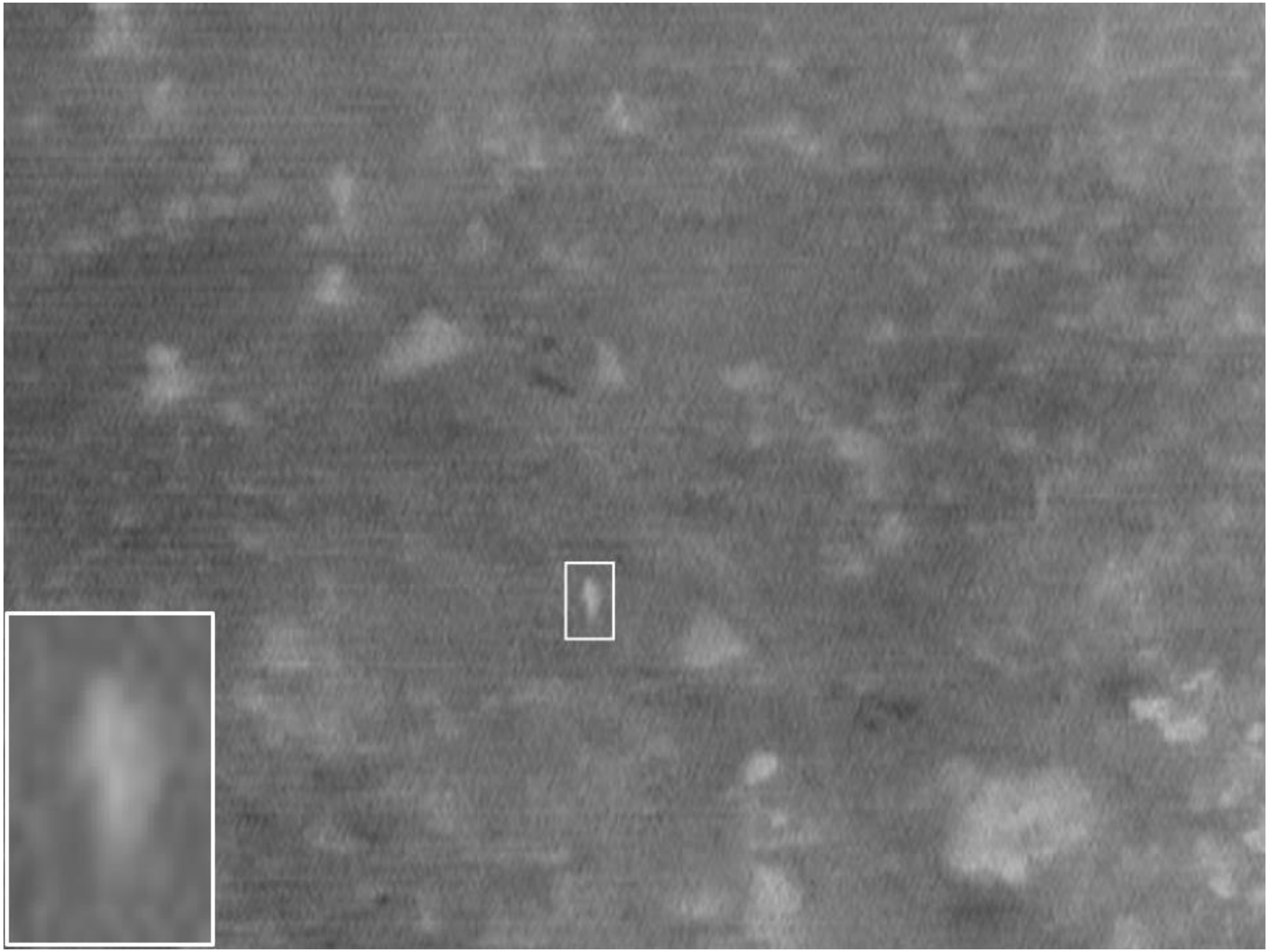
Outline

- Basics of Neural Network
- Convolutional Neural Network
- Faster RCNN
- Applications
  - Detecting wildlife and poachers from UAV videos

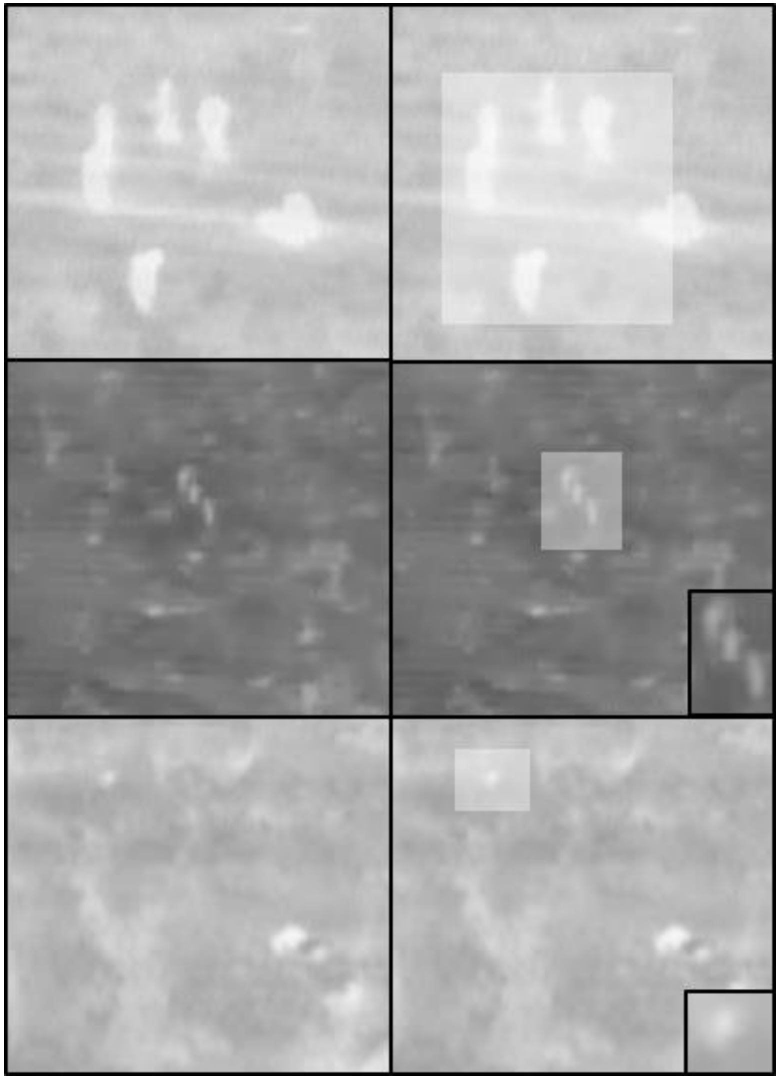
Detecting wildlife and poachers from UAV videos

- Need to complete detection in near real time

Challenges



Challenges



Traditional Computer Vision Approach

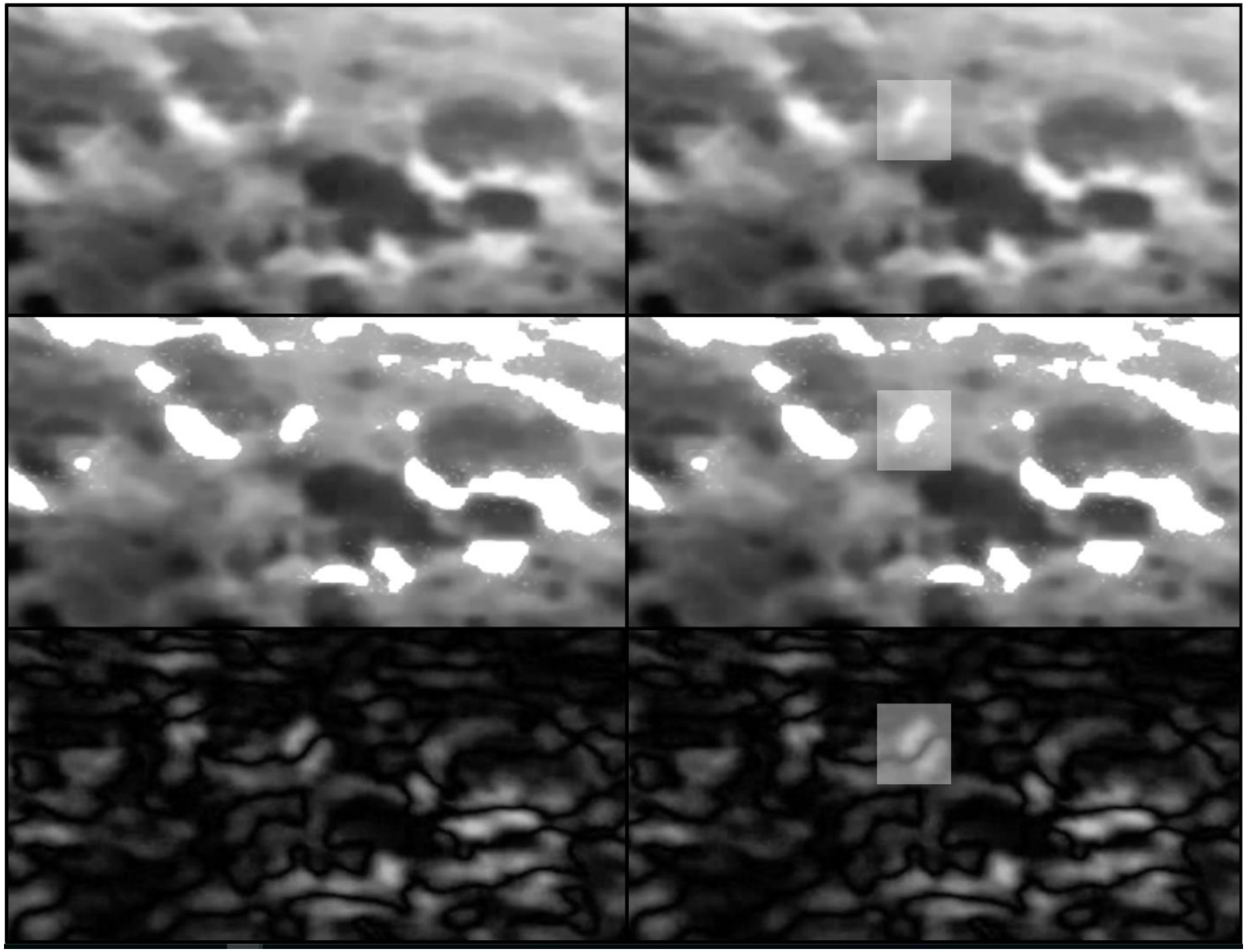
Processed Image

Poacher Highlighted

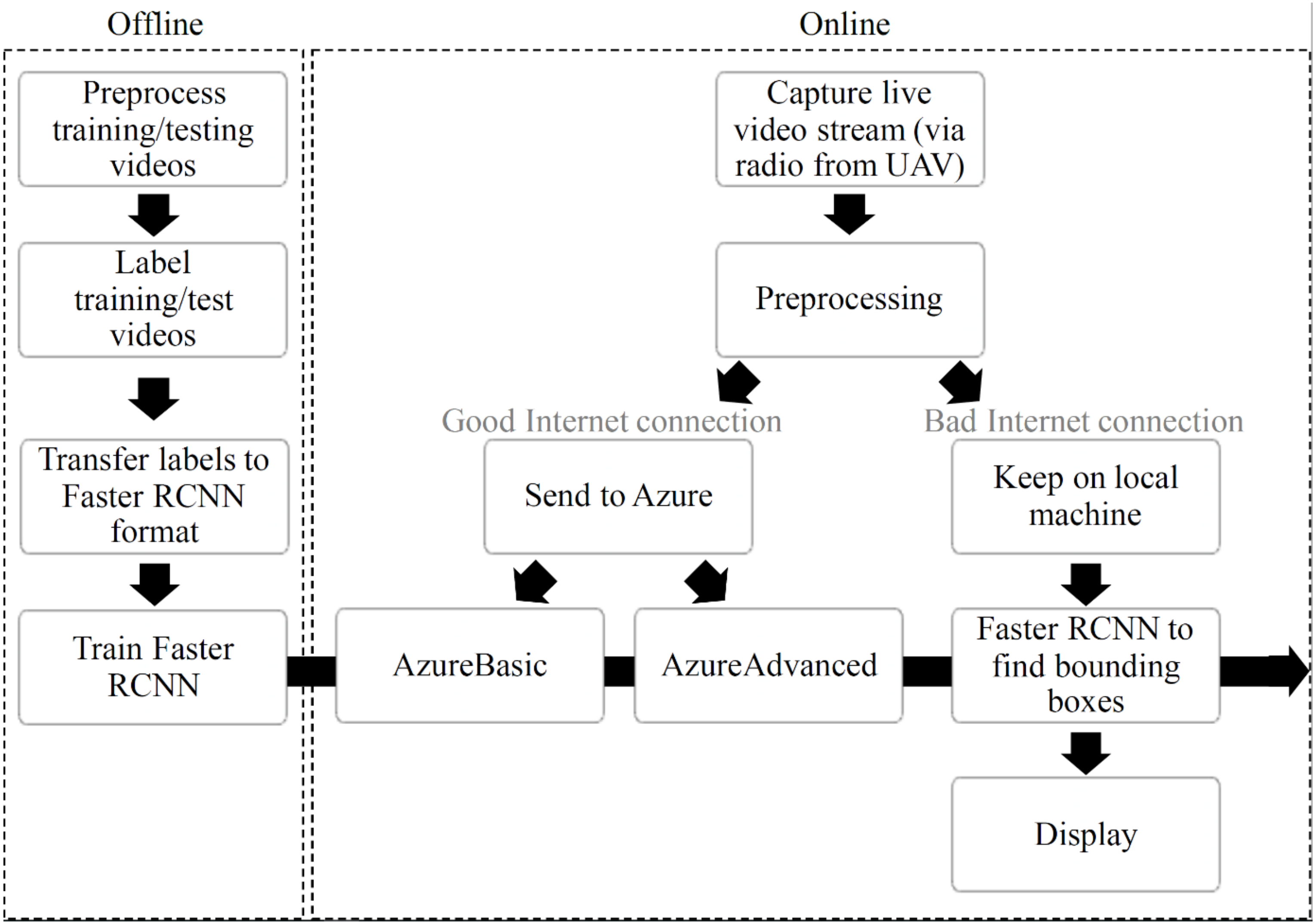
Original

Thresholded

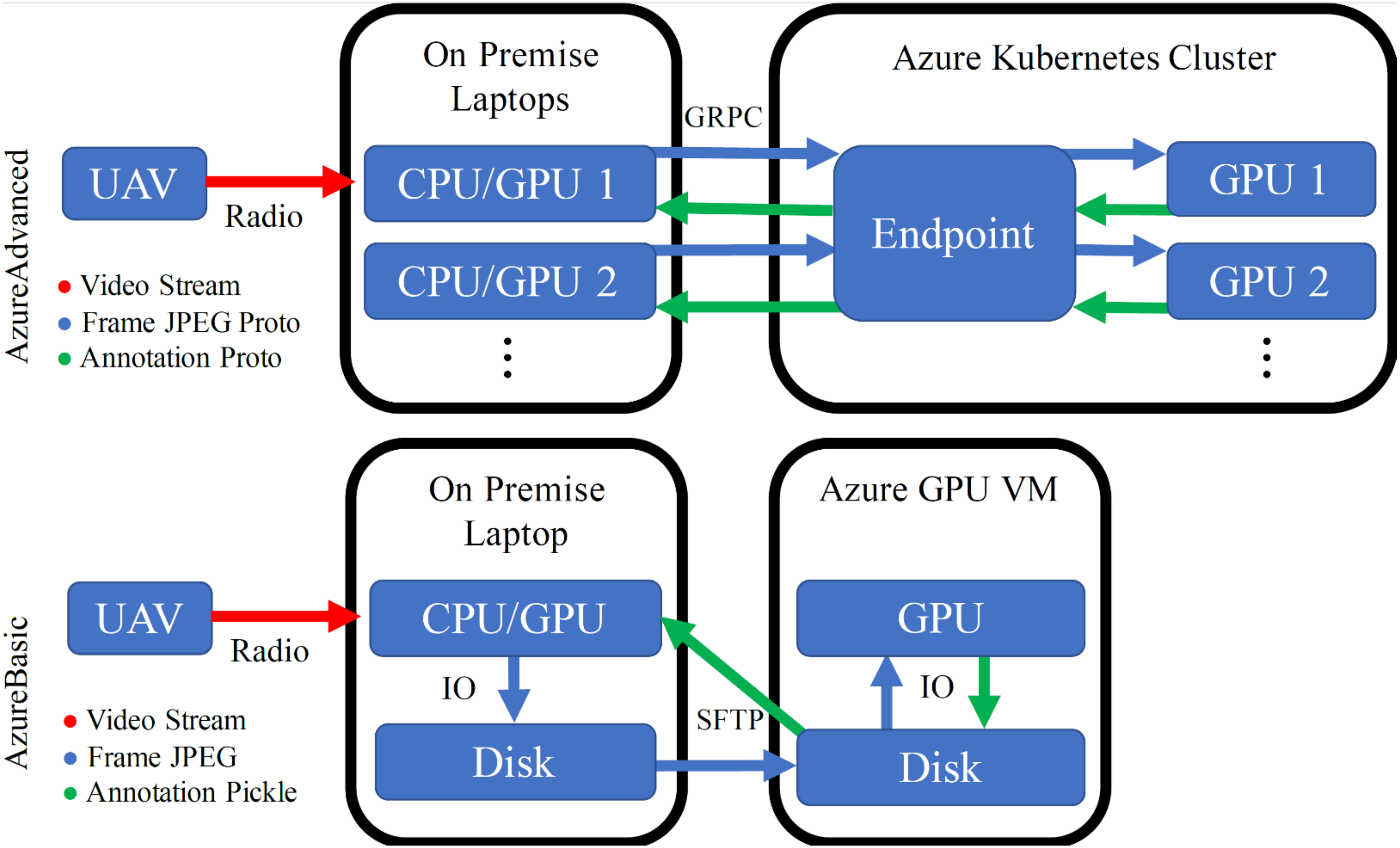
Stabilized frame difference



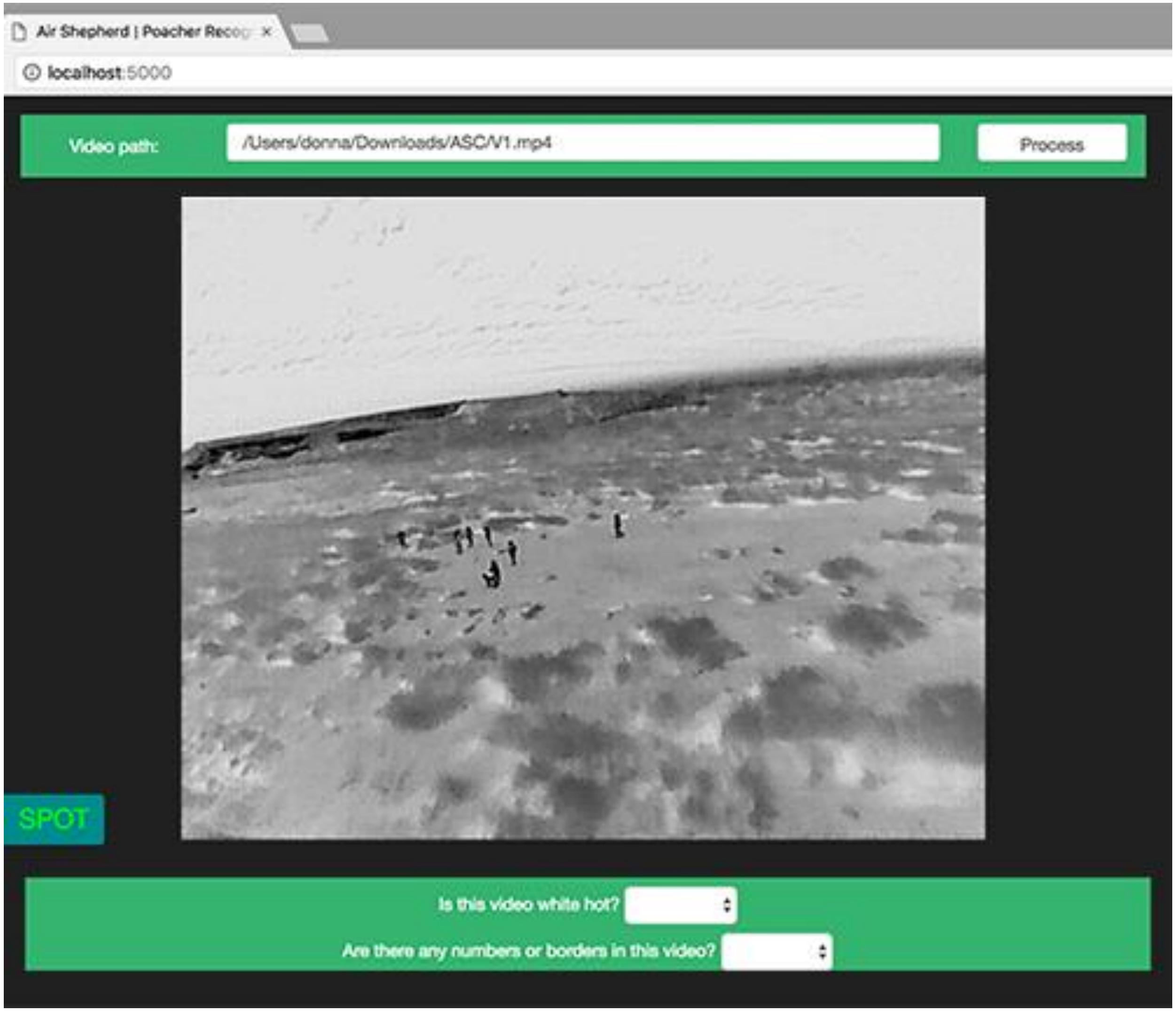
SPOT (Systematic POacher deTector)


Offline
Online
Preprocess training/testing videos
Label training/test videos
Transfer labels to Faster RCNN format
Train Faster RCNN
Capture live video stream (via radio from UAV)
Preprocessing
Good Internet connection
Bad Internet connection
Send to Azure
Keep on local machine
AzureBasic
AzureAdvanced
Faster RCNN to find bounding boxes
Display

Run on the Cloud



SPOT Interface

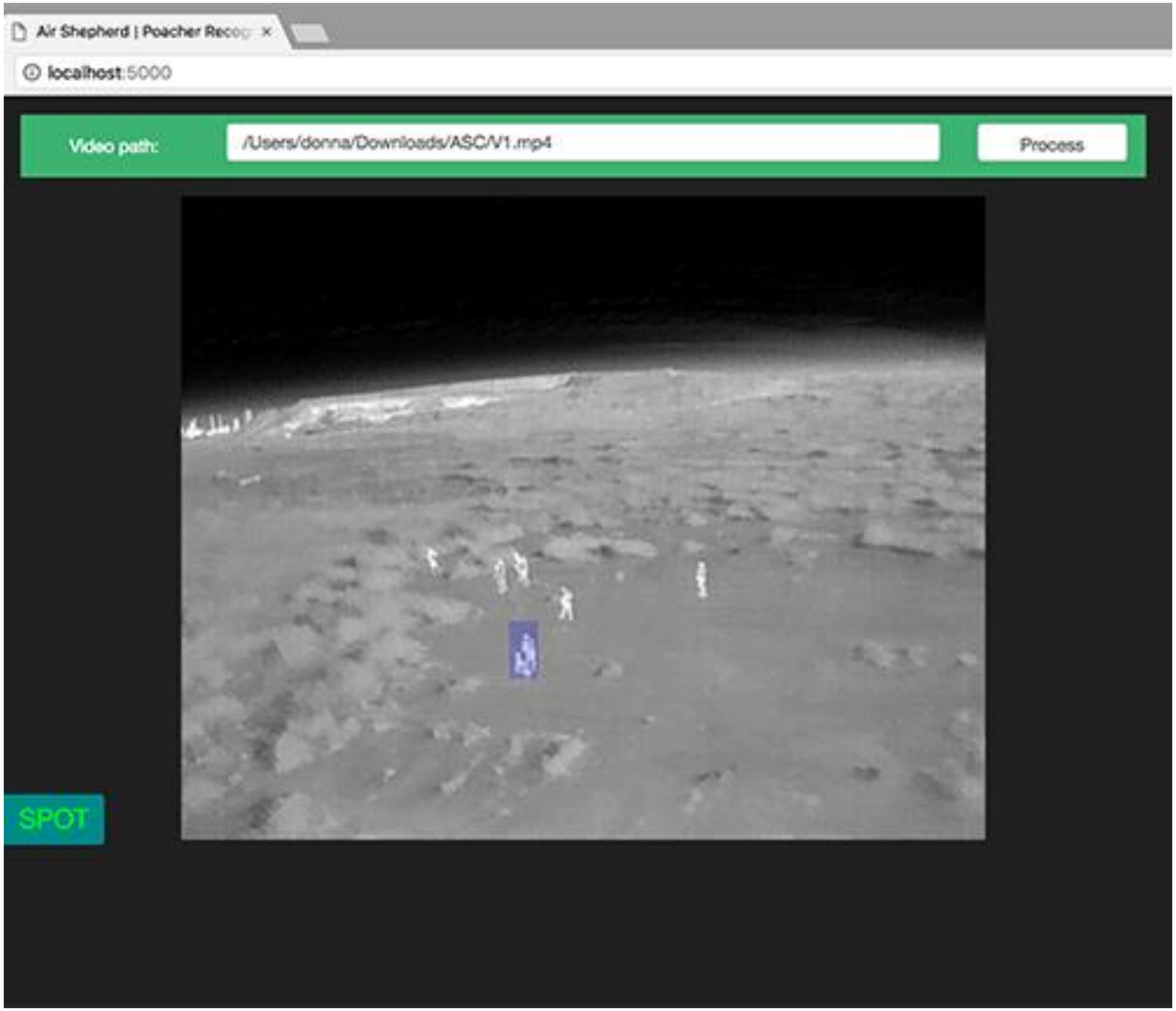


Video path: /Users/donna/Downloads/ASC/V1.mp4

Is this video white hot?

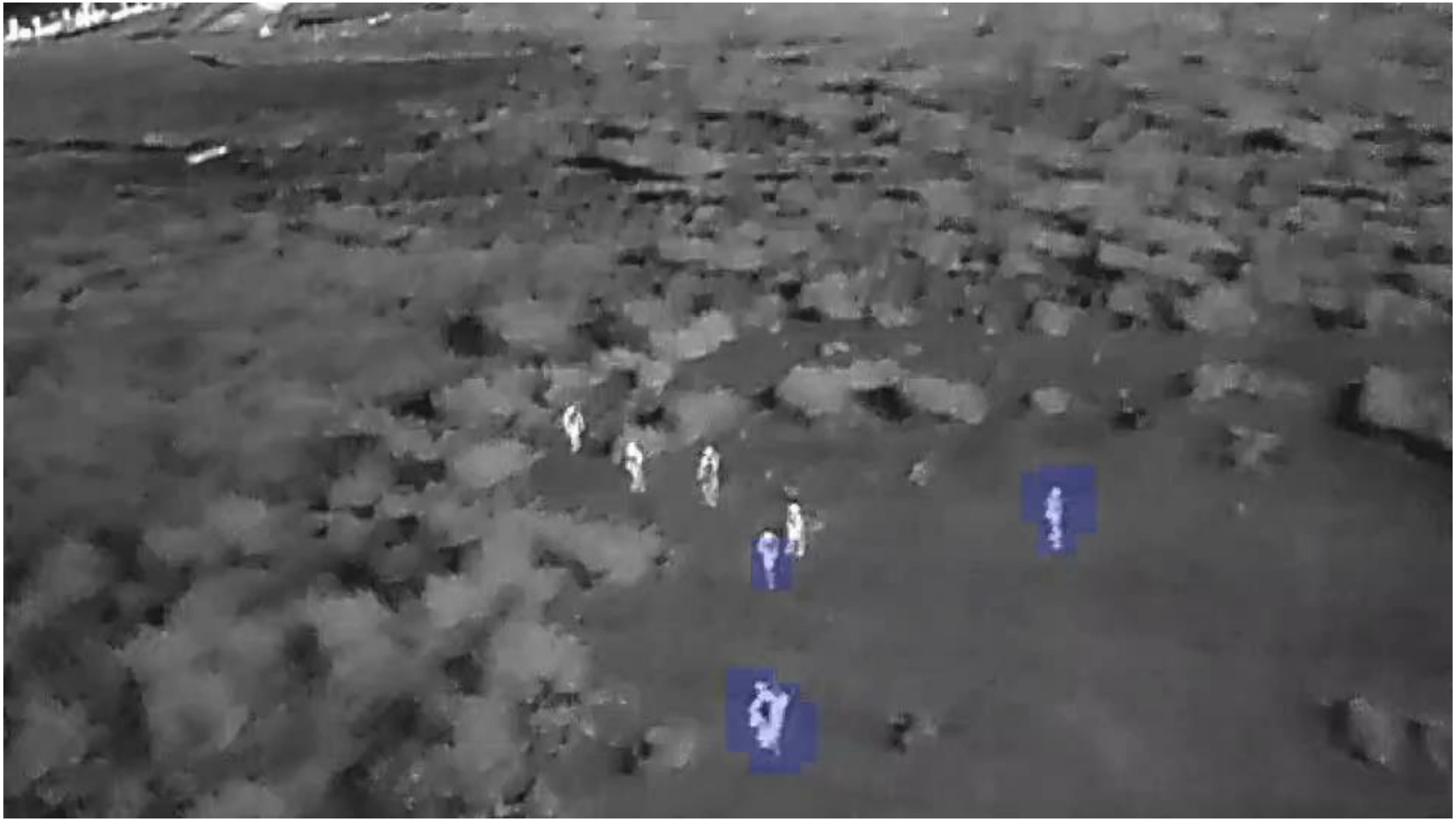
Are there any numbers or borders in this video?

SPOT Interface



Air Shepherd | Poacher Recog ×
localhost:5000
Video path:
/Users/donna/Downloads/ASC/V1.mp4
Process
SPOT

SPOT at Work (Field Test in Botswana)



Additional Resources

- Text book
  - *Deep Learning, Chapter 6, 9*
  - Ian Goodfellow and Yoshua Bengio and Aaron Courville
- Papers
  - *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*
  - Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun
  - *SPOT Poachers in Action: Augmenting Conservation Drones with Automatic Detection in Near Real Time*
  - Elizabeth Bondi, Fei Fang, Mark Hamilton, Debarun Kar, Donnabell Dmello, Jongmoo Choi, Robert Hannaford, Arvind Iyer, Lucas Joppa, Milind Tambe, Ram Nevatia
- Online course
  - https://www.coursera.org/learn/neural-networks-deep-learning
  - https://www.coursera.org/learn/convolutional-neural-networks