Intro To Computer Vision Using OpenCV

Evan Stoddart

03/24/2018

THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

About Me

Evan Stoddart

ADAS Subteam Leader

OSU EcoCAR 3

Graduate Research Assistant

OSU Center For Automotive Research

<u>Hometown</u>:

Pittsburgh, PA

<u>Major</u>:

Electrical and

Computer Engineering

<u>Year</u>:

4th Year BS, 1st Year MS

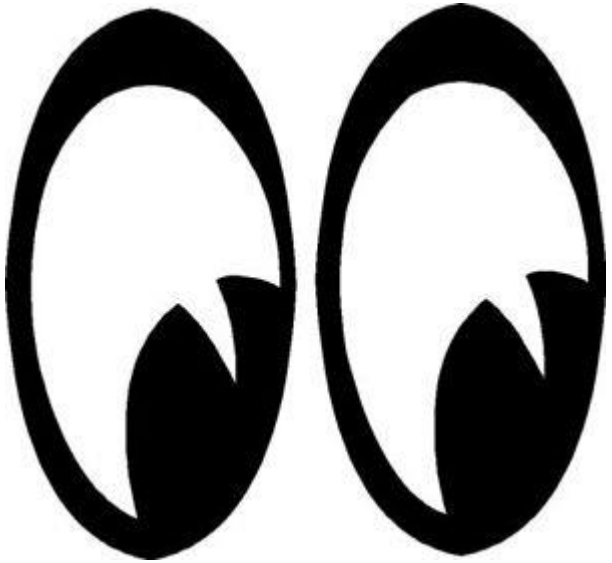
<u>Internships</u>:

Hewlett Packard, STERIS Corporation, Bosch

<u>Interests:</u>

robotics, cars, and medical devices

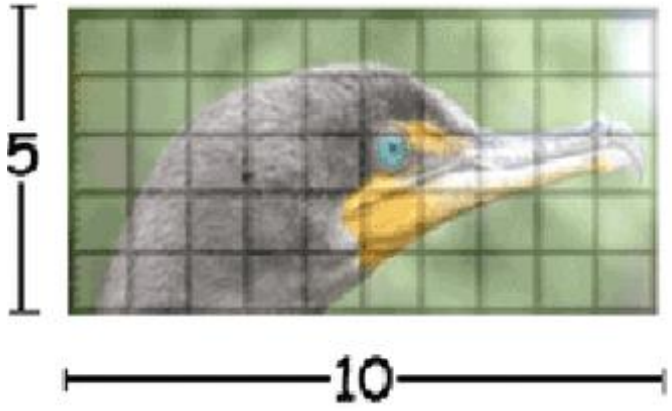
What is Computer Vision?



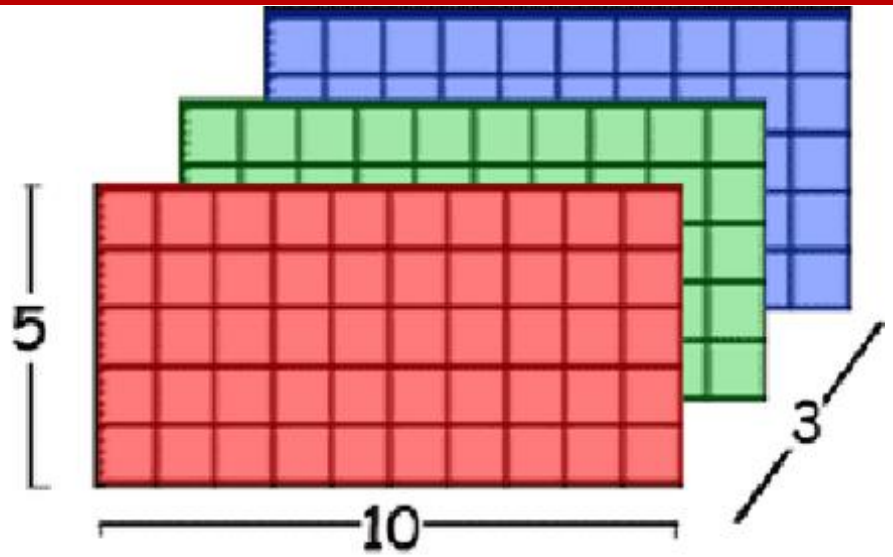
“Computer Vision (CV) is an interdisciplinary field that deals with how computers can be made for gaining high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do”

- Wikipedia

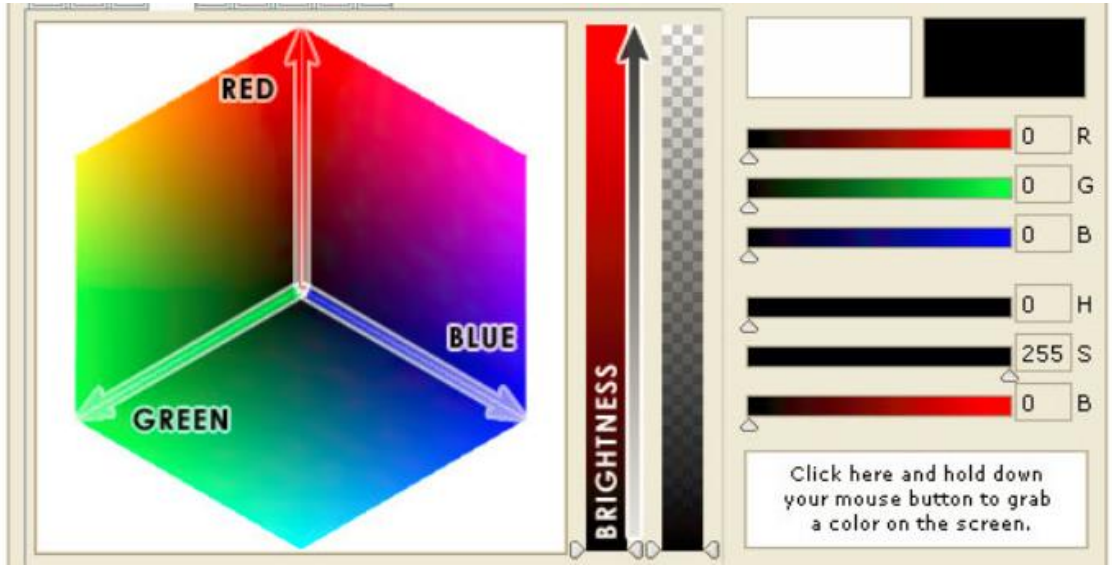
How do We Automate Tasks using CV?



Original Color Image



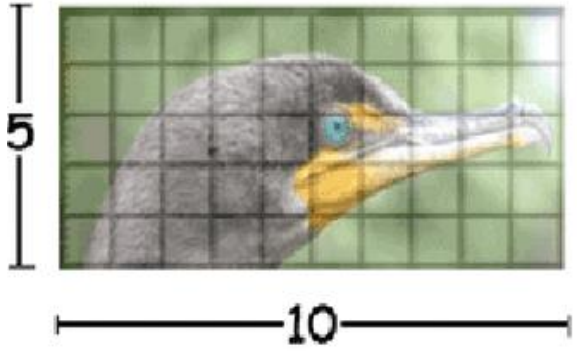
Matlab RGB Matrix



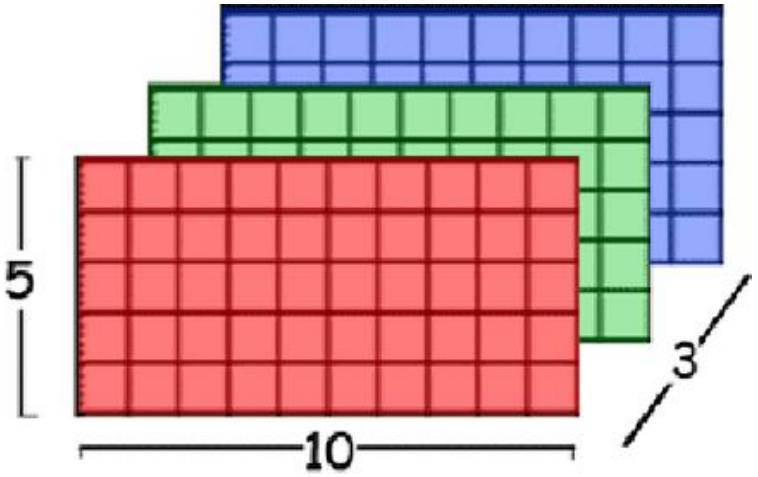
How do We Automate Tasks using CV?

The automation strategy depends on what data we want to find in the image.

- Kernels
  - Math used behind the scenes
- Image Processing
  - Basic operations on images
  - Crop images to gather a Region of Interest (ROI)
  - Adjust image for contrast, color space
- Object Detection/Classification
  - Search for an object in the image
  - Pattern Recognition
  - Machine Learning
  - Deep Learning
- Post Processing
  - Collect information about the object
  - Filter Noise from Data
  - Area of Object
  - Position of Object
  - Color of Object



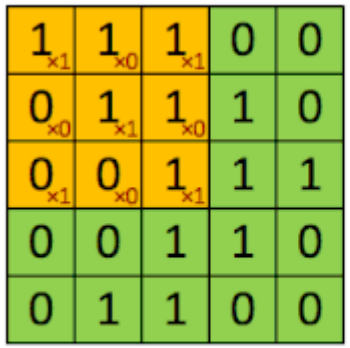
Original Color Image



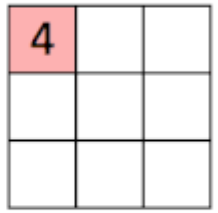
Matlab RGB Matrix

Kernels: The Basic Math used in CV

- Kernels
  - Smaller matrix that mathematically describes the desired operation
- Convolution (Sliding Window)
- When you use Photoshop, this is the math that is happening behind the scenes!
- Used in many CV algorithms for pre processing, object detection, filters and more!



Image

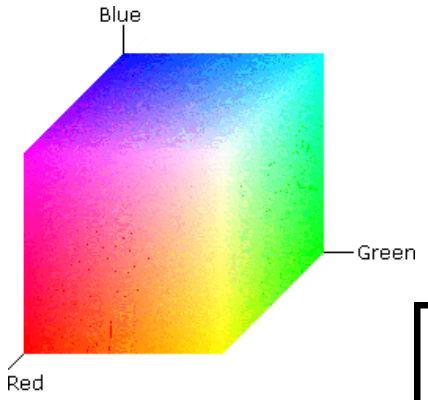


Convolved Feature

Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	

Image Processing

- Crop
- Change color space
  - Grayscale
  - HSV
  - RGB
- Filters
  - Gaussian Blur
- Morphological Operations
  - Dilation - grow image regions
  - Erosion - shrink image regions
  - Opening - structured removal of image region boundary pixels
  - Closing - structured filling in of image region boundary pixels



1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

Object Detection/Classification

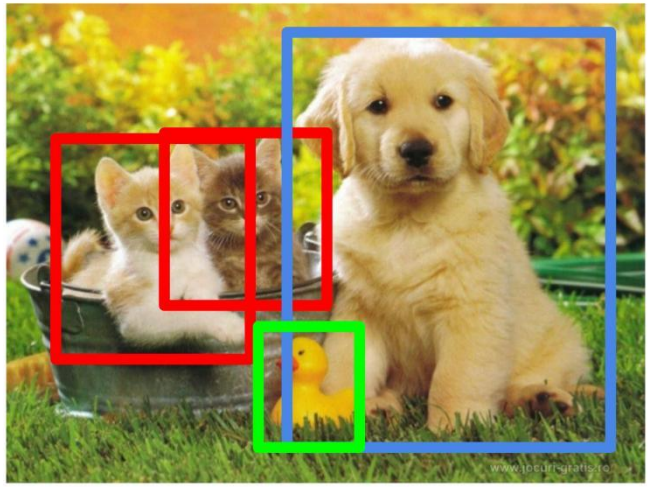
Classification

CAT

Classification + Localization

CAT

Object Detection



CAT, DOG, DUCK

Instance Segmentation

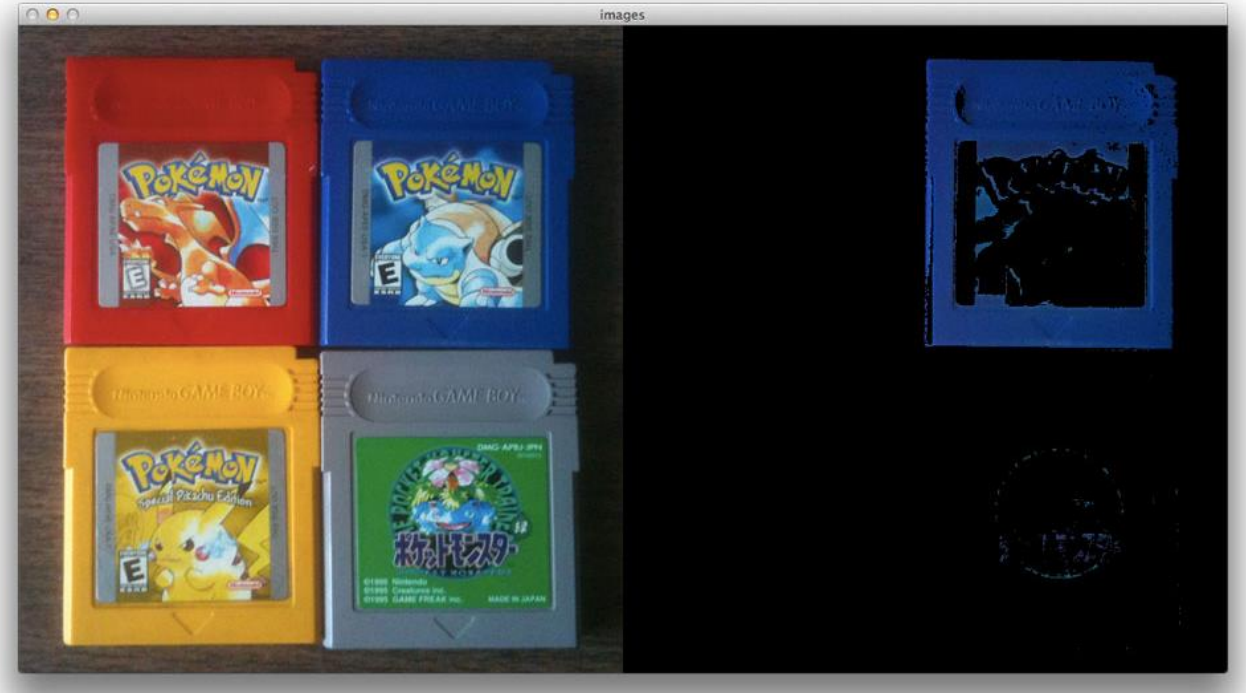
CAT, DOG, DUCK

Single object

Multiple objects

Pattern Recognition

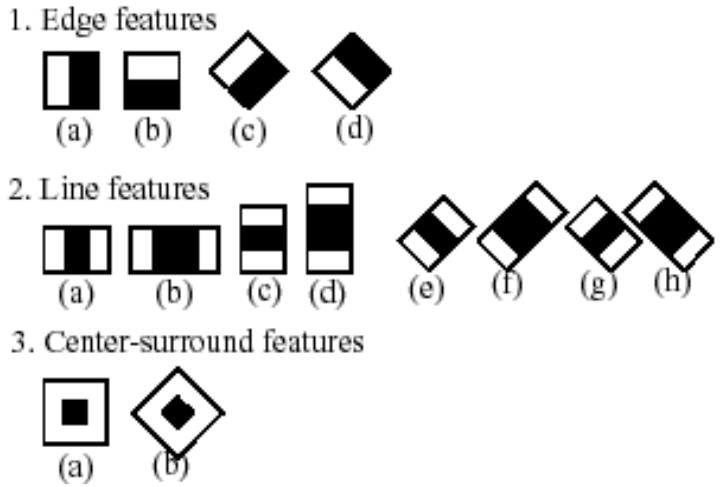
- Pattern recognition was a term popular in the 70s and 80s.
- Getting a computer program to do something “smart” like recognize the character "3“ in an image
  - Algorithms are manually coded by programmer
    - Tedious
    - Not robust
  - “Learning” is not performed by the computer
- Examples:
  - Color Thresholding/Masking
  - Edge Detection



Machine Learning

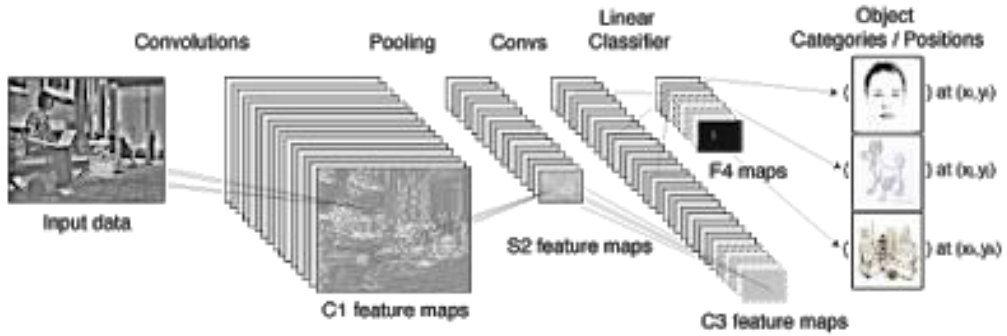
- Sometime in the early 90s people started realizing that a more powerful way to build pattern recognition algorithms is to replace an expert with data.
- Collect data (ex. a bunch of face images and non-face images), choose an algorithm, and wait for the computations “training” to finish.
- "Machine Learning" emphasizes that the computer program (or machine) must do some work to “learn” after it is given data.
- Example Algorithms:
  - Haar Features
  - HOG (Histogram of Gradients)
  - Easy to use in OpenCV,

    all you need is data! More on this later...



Deep Learning

- Similar to Machine Learning in that the computer “learns” from exposure to training data
- Except, Deep Learning programs learn not only the data, but developed its own algorithm to optimize finding the data within an input image
- Deep Learning is built on Convolutional Neural Networks
  - AlexNet
    - Used in the ImageNet Large Scale Visual Recognition Challenge in 2012.
    - The network achieved a top-5 error of 15.3%, more than 10.8 percentage points ahead of the runner up.
  - Inception
    - Modern network
    - trained on more than a million images and can classify images into 1000 object categories
- Mathematically, Deep Learning is hard, use a Deep Learning Frameworks to make it easier
  - Caffe from University of California at Berkeley
  - TensorFlow from Google
  - PyTorch/Torch
- Deep Learning is computationally expensive
  - Use GPU to run the algorithm in parallel



Post Processing

- Filters
  - Object detection isn't always perfect → noise
  - Filters are used to remove unwanted data
  - Examples: Temporal (Heat Map)
- Tracking
  - Persistently track an object as it moved about the screen, image by image, in a video
- Physical Traits
  - Obtain information from image objects
  - Size of object
  - Position of object in space
    - Depth estimation

		Prediction outcome		
		p	n	total
actual value	p′	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

How do We Automate Tasks using CV?

- In theory, implementing CV algorithms into a program can be mathematical and convoluted (no pun intended)

- In practice, libraries exist which abstract the math to provide the desired functionality.
  - More often than not, the function you need is built into a library
  - All you need is a basic understanding of the function, enough to enter parameters needed for your application

What is OpenCV?

What is OpenCV?

- OpenCV is a widely used open-source computer vision framework
  - Over 14 million downloads
  - Library available in C++ and Python
  - Compatible with Tensorflow, Torch/PyTorch, and Caffe Deep Learning Frameworks
  - Open source – anyone can use, edit, contribute to the library
  - First released in 1999 by Intel Corporation
  - Website: Opencv.org

Example Functions

- Using Python
- Image Preprocessing
  - Crop
  - Change colorspace (grayscale)
- Pattern Recognition
  - Color thresholding
- Machine Learning
  - Haar Classifier
- Post Processing
  - Temporal Filter
  - Overlaying Image

What to do if you get stuck

- Forums & Online Resources
  - Stackoverflow
  - pyimagesearch.com
  - OpenCV.org
  - Other application specific forums and websites

- Useful tips for Google searching
  - OpenCV syntax varies by version(popular versions are 2.4, 3.0, 3.3)
  - When searching, specify the version you are using as well as the language (Python/C++)
    - Include the function you are interested in
    - Include an error code if one was printed by your program
    - Example:
      - OpenCV 2.14 Python 2.7 ImportError: DLL load failed: %1 is not a valid Win32 application

Computer Vision Applications

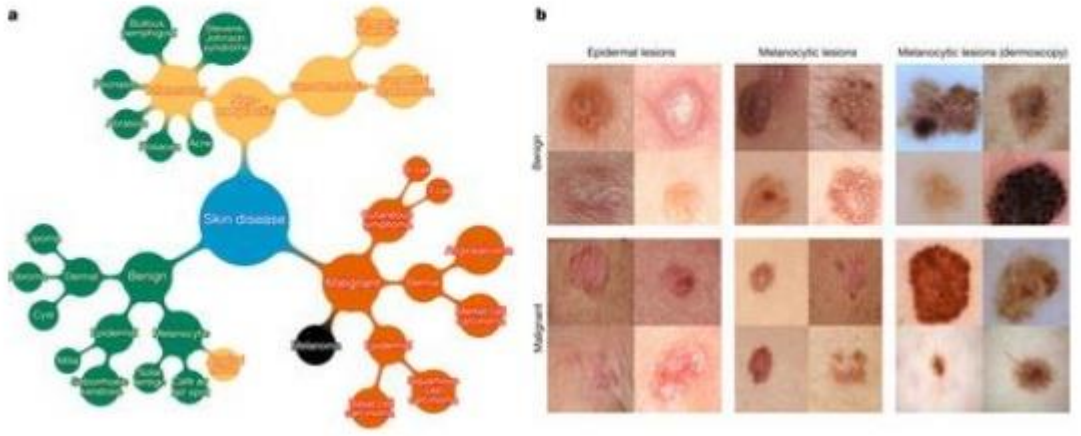
CV Applications

- Robotics
- Medical Devices
- Automotive
- ...the list goes on!

Classification: skin cancer detection

Images organized in a tree taxonomy of 2032 diseases (by medical experts)

CNN trained **757** disease classes: a disease partitioning algorithm to generate classes clinically and visually similar



THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

EcoCAR ADAS Applications

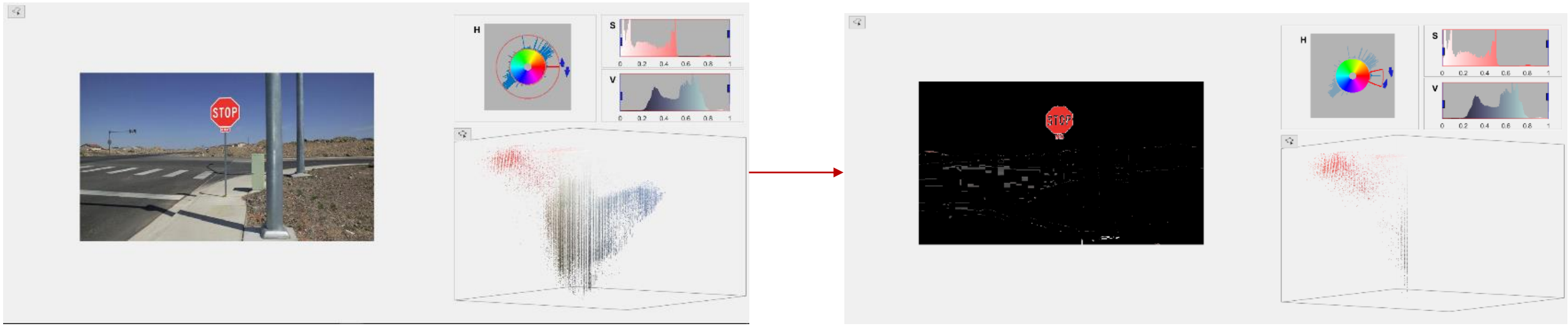
ECOCAR
THE OHIO STATE UNIVERSITY

EcoCAR ADAS Applications

- Advanced Driver Assistance Systems (ADAS)
  - Forward Collision Warning
  - Lane Departure Warning
  - Park assist
  - Adaptive cruise control
- We can use cameras with Computer Vision as well as other sensors like Radar and LiDAR to develop this functionality

Pattern Recognition

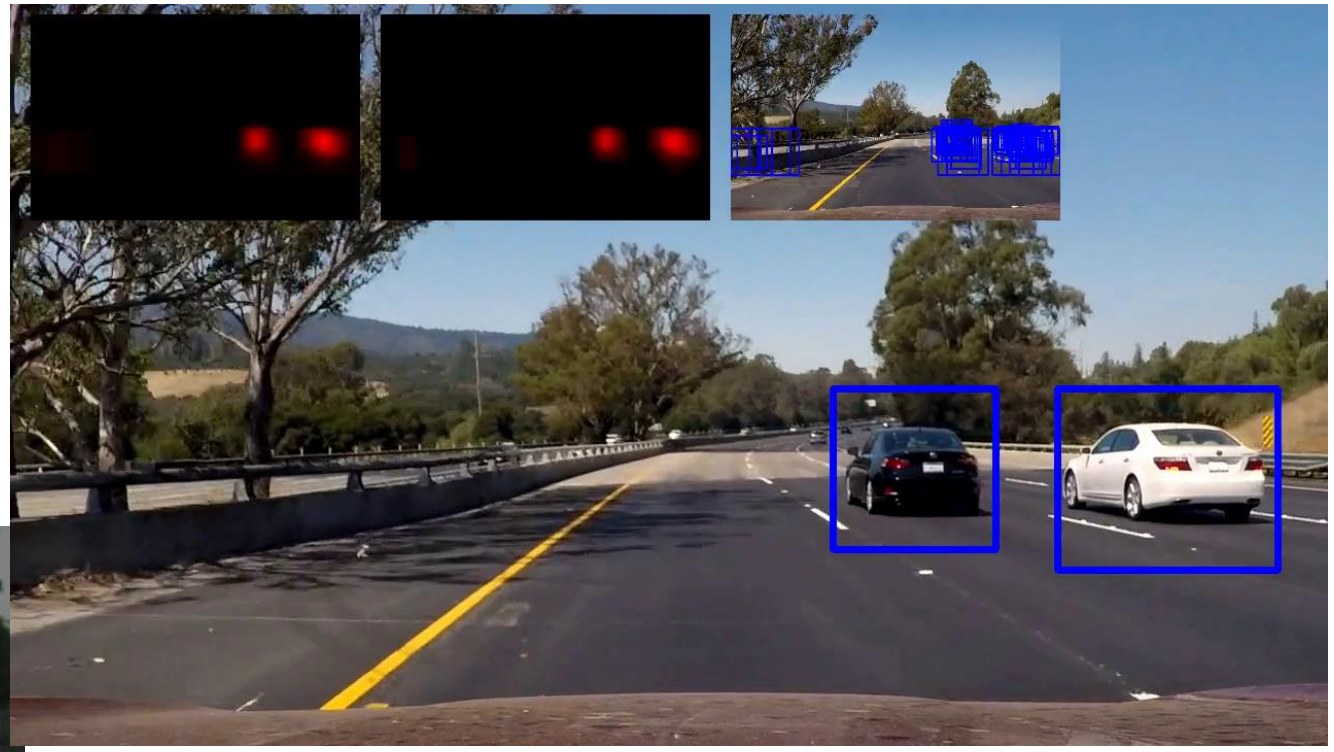
- Color thresholding:
  - An image is just a matrix of values where each value represents a color.
  - We can use computer algorithms to show the colors we want to see and “mask” those that we don’t by selecting corresponding ranges of these values.
  - ADAS application: detect the color red to produce a stop sign detector.



Pattern Recognition

- Canny edge detector + Hough transforms:
  - We can use computer algorithms to highlight edges (i.e. areas in the image with large changes in color/pixel value) – Canny edge detector.
  - After detecting edges, we can determine which edges belong to lines using Hough transforms.
  - ADAS application: detect lines on the road to produce a lane detector/lane departure warning system.

Object Detection



THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

QUESTIONS?