

JeVois Smart Machine Vision Camera

Open-source quad-core camera effortlessly adds powerful machine vision to all your PC, Arduino, and Raspberry Pi projects.

JeVois Guided Tour

v1.1



JeVois Smart Machine Vision Camera

Open-source quad-core camera effortlessly adds powerful machine vision to all your PC, Arduino, and Raspberry Pi projects.

Copyright © 2017-2019 by JeVois Inc.
All rights reserved.

JeVois® is a registered trademark of JeVois Inc, a California Corporation.



JeVois Smart Machine Vision Camera

Open-source quad-core camera effortlessly adds powerful machine vision to all your PC, Arduino, and Raspberry Pi projects.

Welcome to JeVois!

Please visit http://jevois.org/start for information about:

- Connecting JeVois to your computer
- Video capture software and setting different video resolutions
- Connecting to the command-line interface of JeVois
- Updating microSD card to latest JeVois software
- This guide was written for JeVois software **1.12 and later**
- Note: with older JeVois software, some modules described here will be missing, just skip over these pages.



JeVois Smart Machine Vision Camera

Open-source quad-core camera effortlessly adds powerful machine vision to all your PC, Arduino, and Raspberry Pi projects.

How to use this guide:

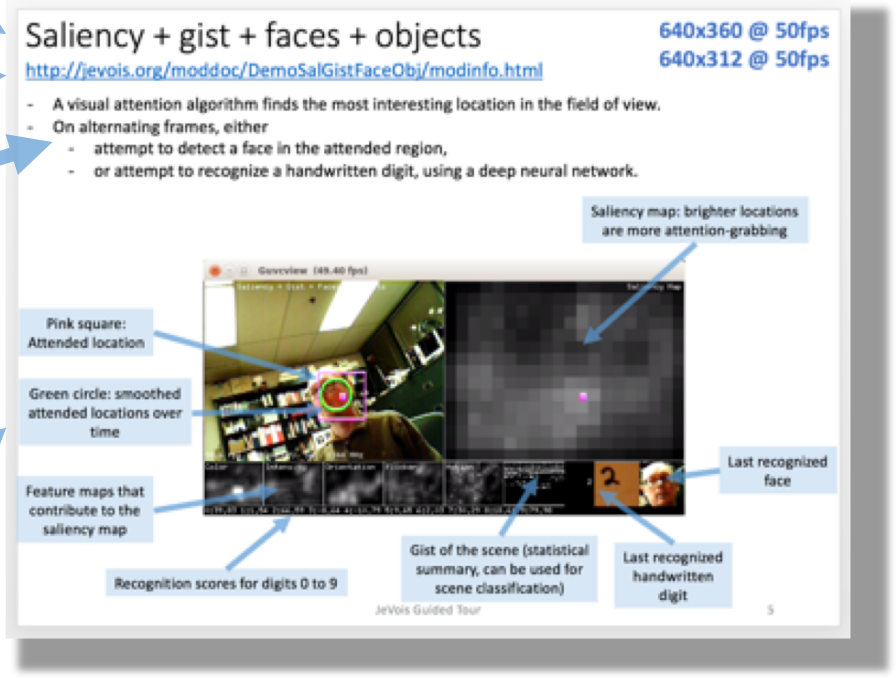
JeVois machine vision module name

Link to documentation page

Introduction

Explanation of the display organization

Video resolutions which will launch this module



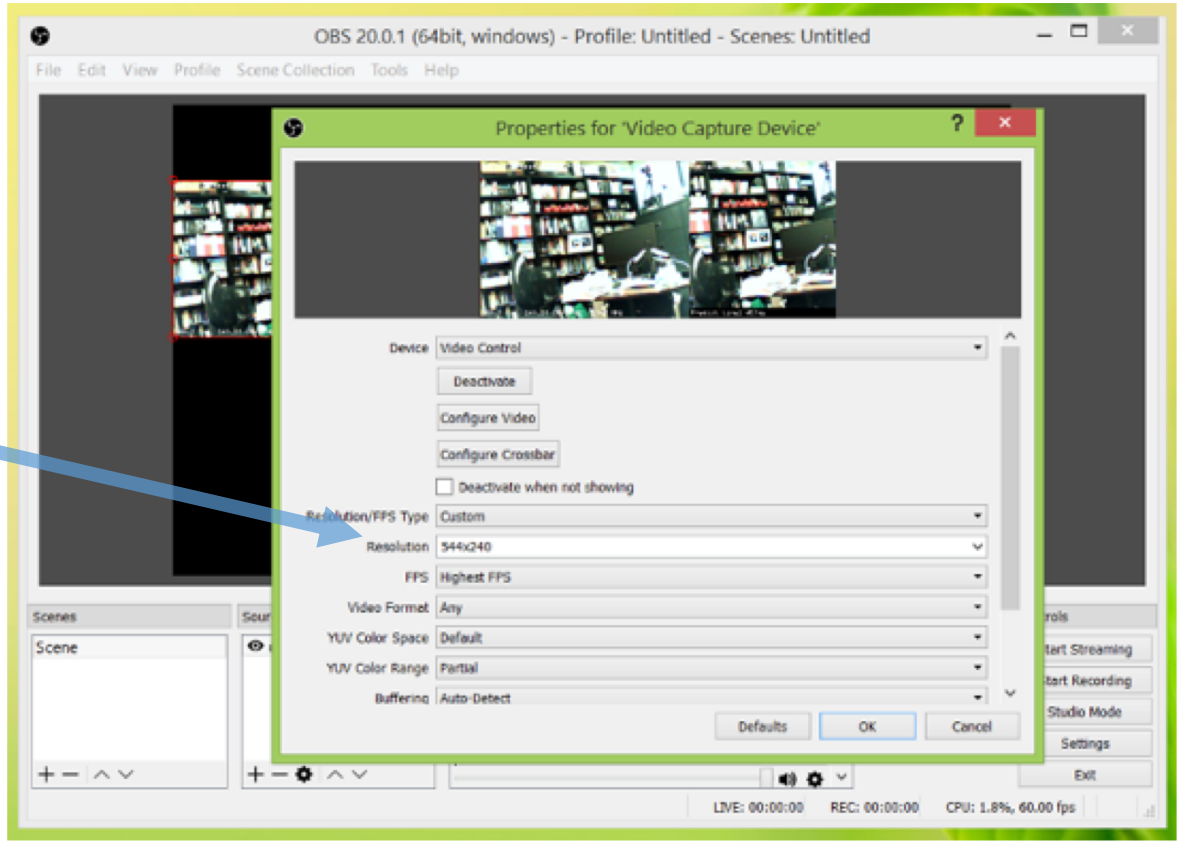


JeVois Smart Machine Vision Camera

Open-source quad-core camera effortlessly adds powerful machine vision to all your PC, Arduino, and Raspberry Pi projects.

To launch a module, select its video resolution (shown at top right of each page) in your video capture software or in JeVois Inventor.

Example: setting resolution in Open Broadcaster Studio



See http://jevois.org/start for detailed instructions.

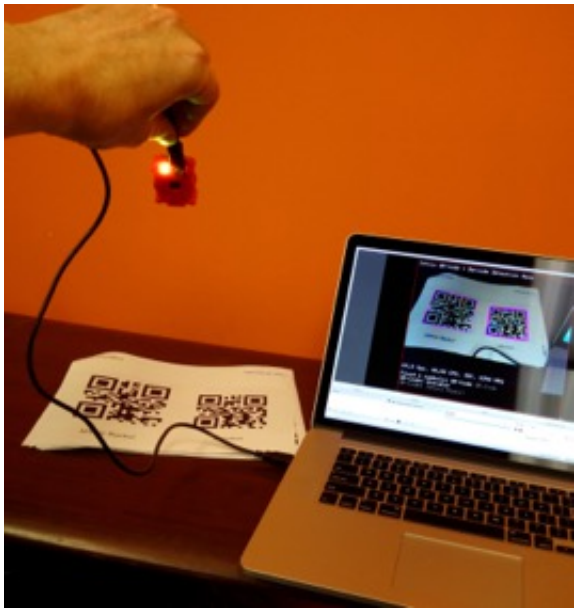


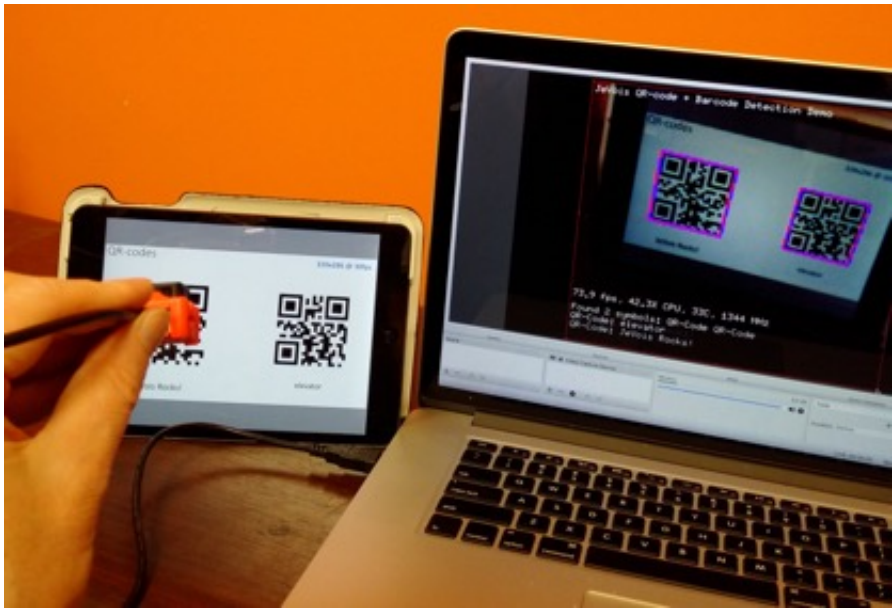
JeVois Smart Machine Vision Camera

Open-source quad-core camera effortlessly adds powerful machine vision to all your PC, Arduino, and Raspberry Pi projects.

For some modules, this guide provides sample images: point JeVois towards them while running that module. You may want to either:

- print this guide (color required for some examples to work well)
- display it on a tablet or computer screen (beware of reflections of lights in your room onto those screens, as they could interfere with JeVois processing; also some older screens may have too much flicker and may not work well)



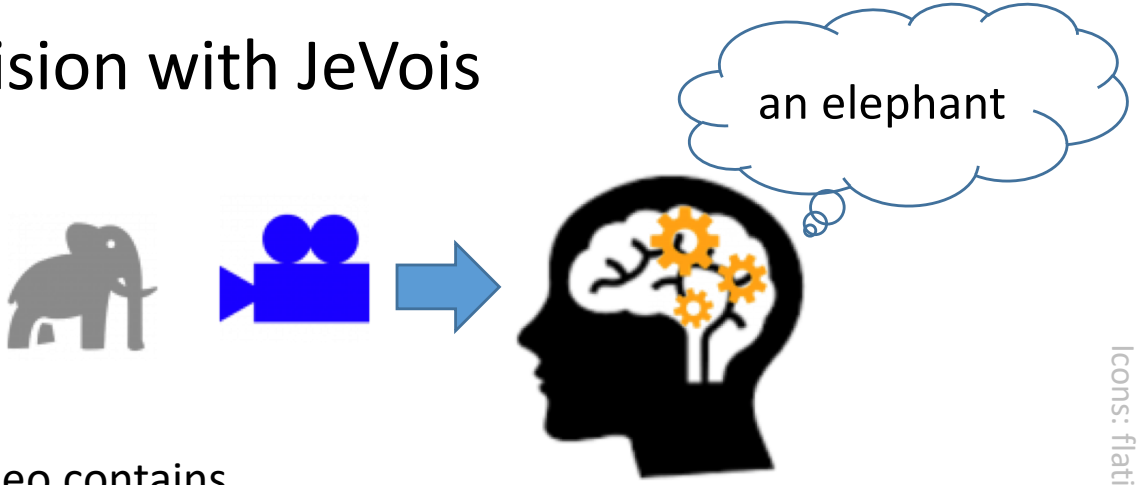




JeVois Smart Machine Vision Camera

Open-source quad-core camera effortlessly adds powerful machine vision to all your PC, Arduino, and Raspberry Pi projects.

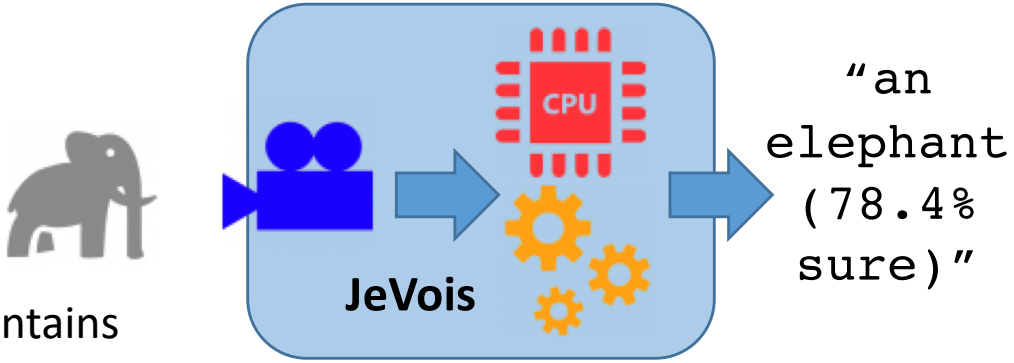
Computer (machine) vision with JeVois



Standard camera:

- captures video
- sends video to human user
- human brain interprets what video contains

Icons: flaticon



JeVois smart camera:

- captures video
- processes video to interpret what it contains
- sends interpretation to user, robot, or another computer



JeVois Smart Machine Vision Camera

Open-source quad-core camera effortlessly adds powerful machine vision to all your PC, Arduino, and Raspberry Pi projects.

Now get ready to be amazed and try these out:

1. Visual attention – finding interesting things
2. Face and handwritten digit recognition
3. QR-codes and other tags
4. Road detection
5. Object matching
6. Object recognition with deep neural networks
7. Scene analysis (object detection + recognition) with deep nets
8. Face detection and facial emotion analysis
9. Color-based object tracking
10. Moving object detection
11. Record video to the microSD card inside JeVois
12. Motion flow detection
13. Eye tracking
14. and more!



JeVois Smart Machine Vision Camera

Open-source quad-core camera effortlessly adds powerful machine vision to all your PC, Arduino, and Raspberry Pi projects.

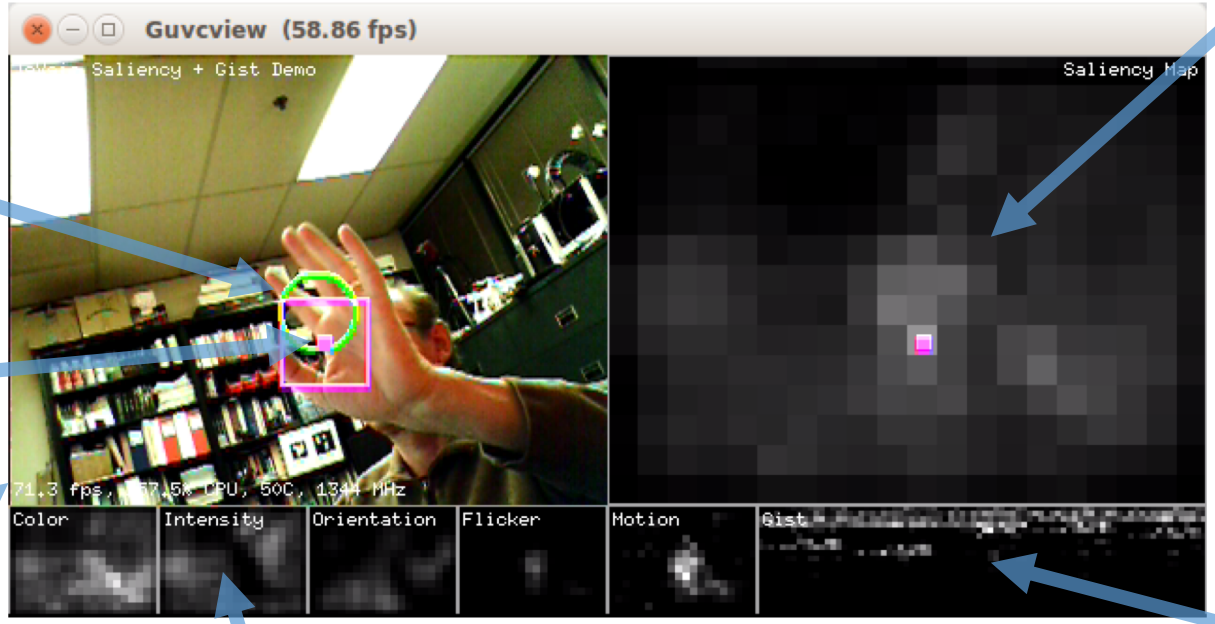
Turn on your JeVois camera and move to the next page.

Visual attention: Saliency

640x300 @ 60fps

http://jevois.org/moddoc/DemoSaliency/modinfo.html

- Finds the location in the camera's view that is the most attention-grabbing, conspicuous, or so-called **salient**. Based on a neuroscience model that mimics attention processing in the human brain.
- This module only finds interesting things. It does not recognize what they are.



Saliency map: brighter locations are more attention-grabbing

Pink square: Attended location

Green circle: smoothed attended location over time

Algorithm speed (here, attention algorithm runs at 71.3 frames/s)

Feature maps that contribute to the saliency map

Gist of the scene (statistical summary, can be used for scene classification)

Visual attention: Saliency

640x300 @ 60fps

http://jevois.org/moddoc/DemoSaliency/modinfo.html

- Try it: wave at JeVois, point JeVois towards complex scenes and see what it thinks is most interesting.



Image credit: Esmée Winnubst / flickr

Visual attention: Saliency

640x300 @ 60fps

http://jevois.org/moddoc/DemoSaliency/modinfo.html

- Search the web for images where something “catches your eye”, and show them to JeVois.



Image credit: Chris Fort / flickr

Visual attention: Saliency

640x300 @ 60fps

http://jevois.org/moddoc/DemoSaliency/modinfo.html

- Search the web for images where something “catches your eye”, and show them to JeVois.

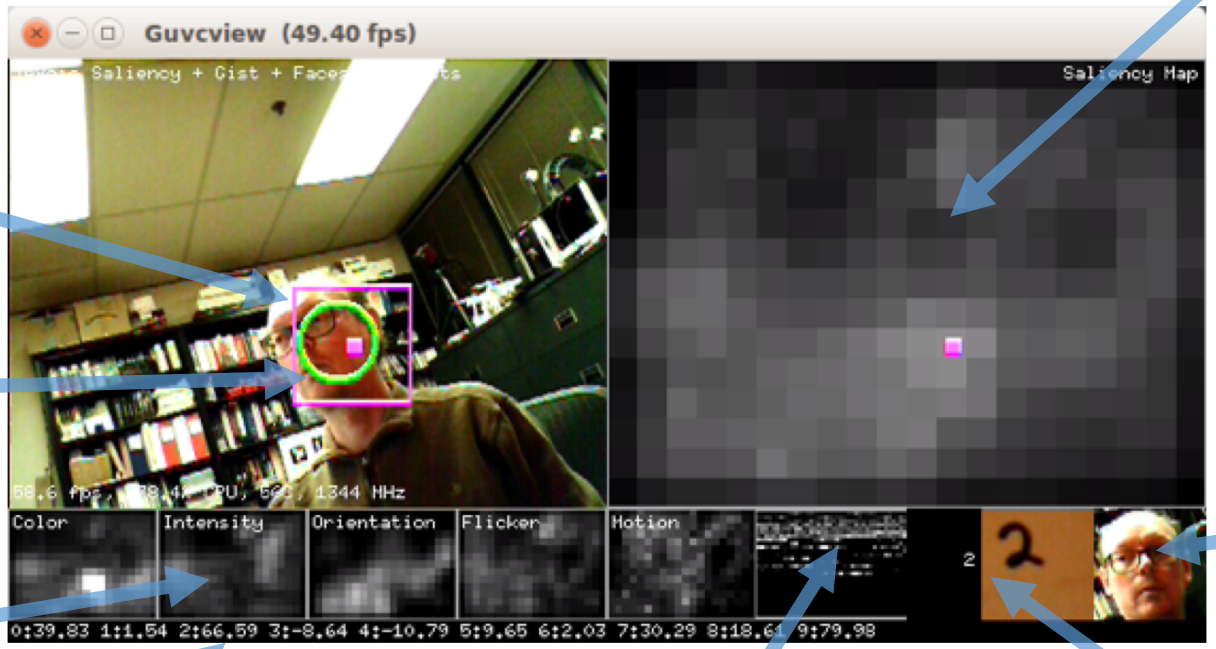


Saliency + gist + faces + objects

640x360 @ 50fps
640x312 @ 50fps

http://jevois.org/moddoc/DemoSalGistFaceObj/modinfo.html

- A visual attention algorithm finds the most interesting location in the field of view.
- Then, interpret what is in the attended region, in two ways:
  - attempt to detect a face, and,
  - attempt to recognize a handwritten digit, using a deep neural network.



Saliency map: brighter locations are more attention-grabbing

Pink square: attended location

Green circle: smoothed attended locations over time

Feature maps that contribute to the saliency map

Recognition scores for digits 0 to 9 (higher is better)

Gist of the scene (statistical summary, can be used for scene classification)

Last recognized handwritten digit

Last recognized face

Saliency + gist + faces + objects

640x360 @ 50fps
640x312 @ 50fps

http://jevois.org/moddoc/DemoSalGistFaceObj/modinfo.html

- Point JeVois to a digit, check bottom-right corner of video.
- Adjust distance so digit fits inside pink box (this algorithm is not very invariant to size).
- Hold JeVois straight (this algorithm does not handle rotation very well).



- Sometimes this algorithm makes mistakes.
- It was trained on 60,000 images. It was 99.2% correct on an independent set of 10,000 test images. But live video is always much harder.

Saliency + gist + faces + objects

640x360 @ 50fps
640x312 @ 50fps

http://jevois.org/moddoc/DemoSalGistFaceObj/modinfo.html

- Point JeVois to a face, check bottom-right corner of JeVois video.
- Adjust distance so face fits inside pink box (this algorithm is not very invariant to size).
- Hold JeVois straight (this algorithm does not handle rotation very well).



Image credit: Tyrone Daryl / flickr

Saliency + gist + faces + objects

640x360 @ 50fps
640x312 @ 50fps

http://jevois.org/moddoc/DemoSalGistFaceObj/modinfo.html

- Try with your own face too! You may need to move it and shake it to make it attention-grabbing!
- Adjust distance from camera to make sure your face fits in the pink square.







Image credit: Julius Cruickshank, The Next Web, Sadettin Canbay / flickr

QR-codes and barcodes

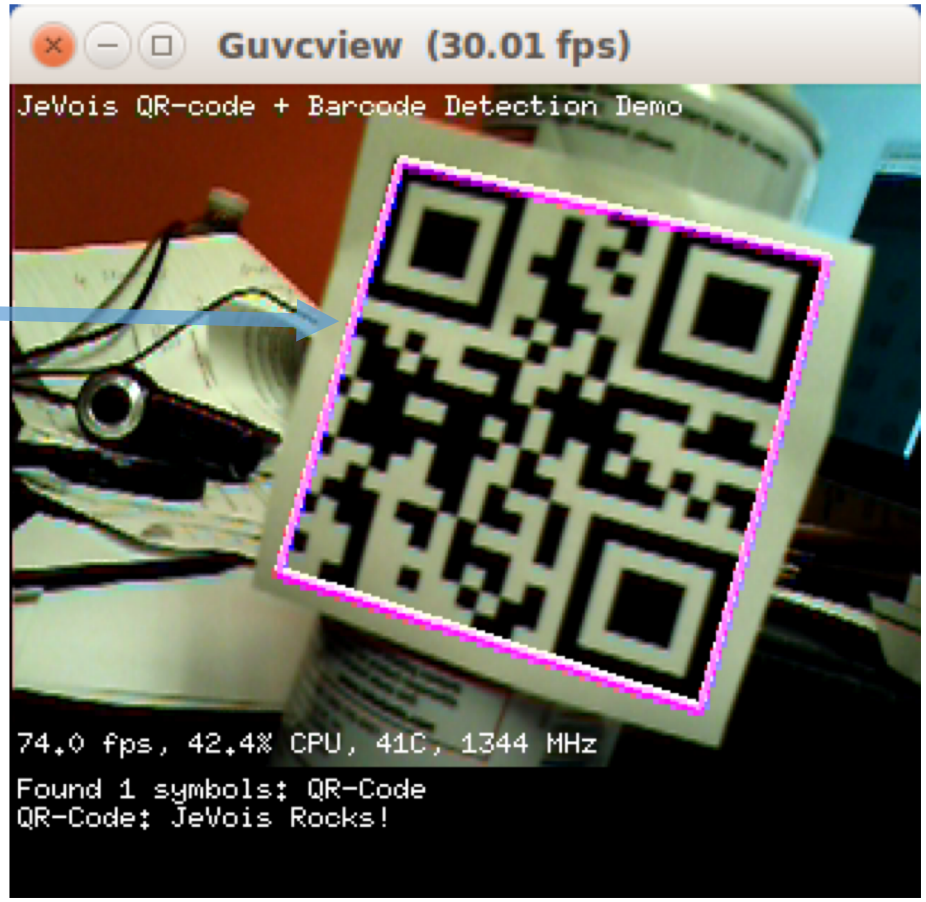
320x286 @ 30fps

http://jevois.org/moddoc/DemoQRcode/modinfo.html

- Detect and decode QR-codes (2D barcodes) and standard barcodes.
- Useful for robots, visual aids for the blind, automated processing of packaged goods, etc.
- Note: only QR-codes enabled by default (no barcodes); see documentation for how to enable barcodes.

Purple outline: detected code

Bottom text: decoded code contents



QR-codes and barcodes

320x286 @ 30fps

http://jevois.org/moddoc/DemoQRcode/modinfo.html



QR-codes and barcodes

320x286 @ 30fps

http://jevois.org/moddoc/DemoQRcode/modinfo.html



QR-codes and barcodes

320x286 @ 30fps

http://jevois.org/moddoc/DemoQRcode/modinfo.html

- Try it by searching for QR-codes on the web.
- Make your own QR-codes, for example at http://www.qr-code-generator.com





ArUco markers

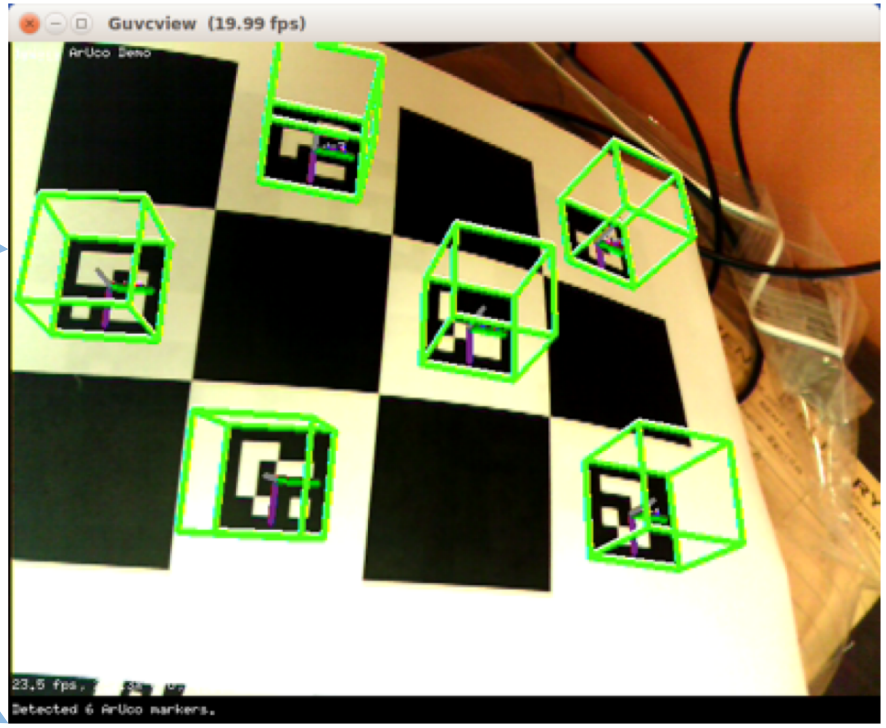
320x260 @ 30fps
640x500 @ 20fps

http://jevois.org/moddoc/DemoArUco/modinfo.html

- ArUco markers are small 2D barcodes often used in augmented reality and robotics.
- JeVois can find them, decode their identity, and recover their full 6D pose (3D location + 3D orientation).

Green outlines: detected markers

Bottom text: number of detected markers



- Note: default behavior is to only show 2D outlines of markers.
- To enable 6D pose computation and cube displays, connect to JeVois serial port and issue:

```
setpar dopose true
setpar showcube true
```

ArUco markers

http://jevois.org/moddoc/DemoArUco/modinfo.html

320x260 @ 30fps
640x500 @ 20fps



ArUco 42



ArUco 18



ArUco 12



ArUco 27



ArUco 43



ArUco 5

ARtoolkit markers

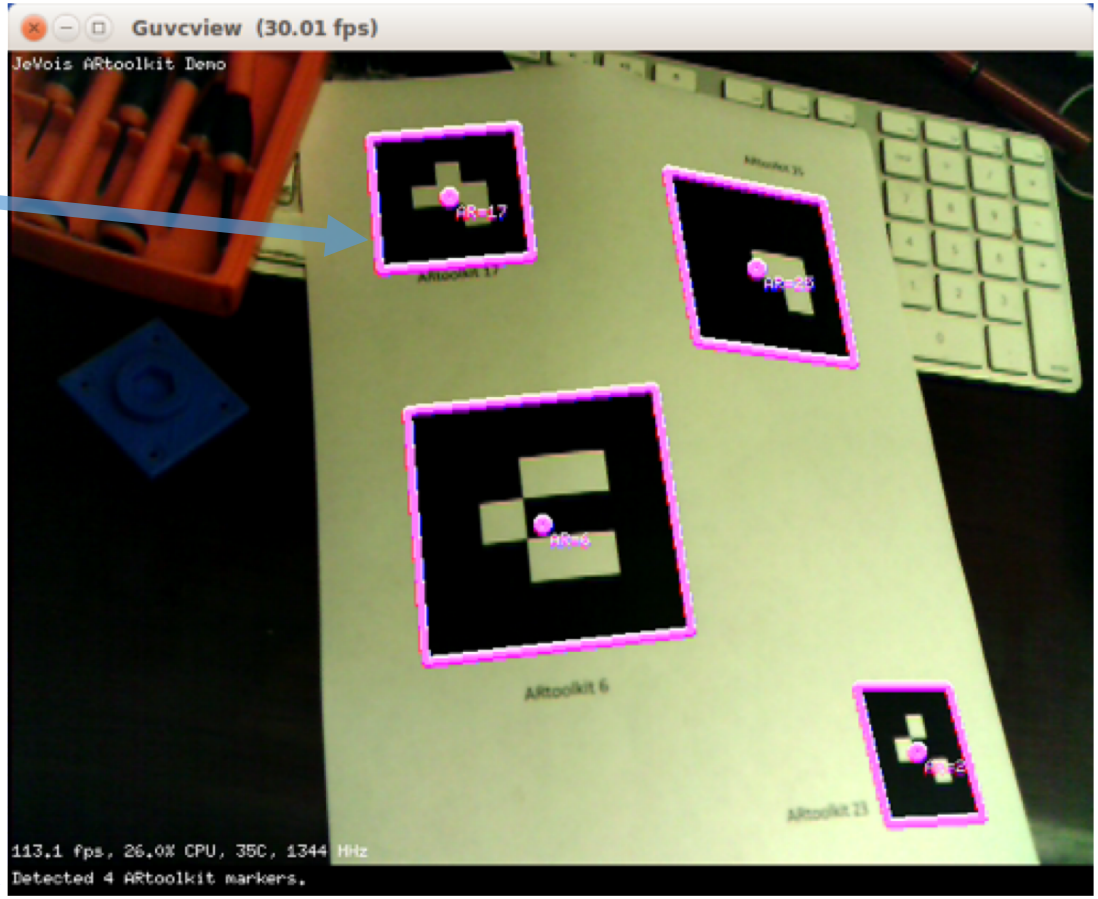
320x258 @ 60fps

http://jevois.org/moddoc/DemoARtoolkit/modinfo.html

- ARtoolkit markers are small 2D barcodes often used in augmented reality and robotics.
- JeVois can find them, decode their identity, and recover their full 6D pose (3D location + 3D orientation).
- This algorithm is ultra fast: 400+ fps @ at 320x240, 100+ fps @ 640x480, 20+ fps @ 1280x1024.

Pink outlines: detected markers

Bottom text: number of detected markers

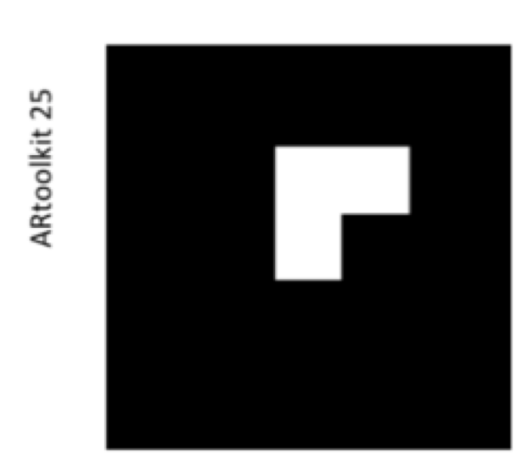


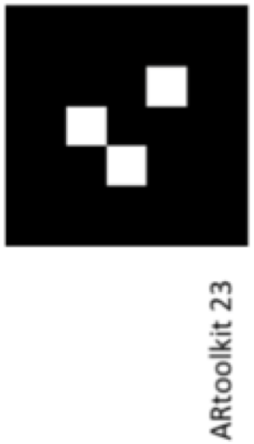
ARtoolkit markers

320x258 @ 60fps

http://jevois.org/moddoc/DemoARtoolkit/modinfo.html

- You can assign roles to markers to help a robot find its way. For example: you can decide that marker 25 means “turn right”, marker 23 means “charging station”, etc.









Markers Combination

640x546 @ 20fps

http://jevois.org/moddoc/MarkersCombo/modinfo.html

- Detect QR-codes, ArUco markers, and Artoolkit markers at the same time!
- The 3 algorithms run in parallel on the quad-core processor of JeVois

Purple outlines: detected QR-codes

Pink outlines: detected Artoolkit markers

Green outlines: detected ArUco markers

Bottom text: number of detected markers, decoded QR-codes



Markers Combination

640x546 @ 20fps

http://jevois.org/moddoc/MarkersCombo/modinfo.html



ARtoolkit 6



ArUco 18



office

Road detection

320x256 @ 30fps

http://jevois.org/moddoc/RoadNavigation/modinfo.html

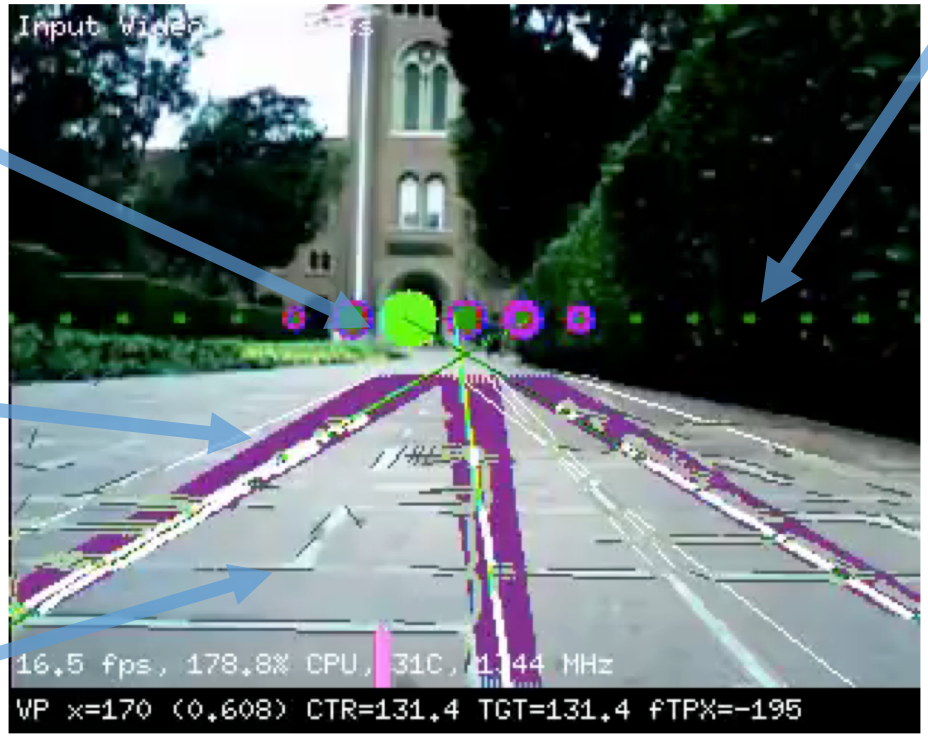
- Combines detection and tracking of line segments at the edges of the road or on the road (e.g., lane dividers), and texture analysis to distinguish the road region from its surroundings.
- Be sure to position JeVois so that the row of green dots is on the horizon line of your scene.
- Note: this algorithm runs much faster when there is no display (drawing all these lines slows it down).

Row of small green dots: vanishing point candidates

Largest green dot: vanishing point (heading of this road)

Purple lines: road lines that were reliably tracked over several frames

Black, white and grey lines: line segments detected on current frame



Road detection

320x256 @ 30fps

http://jevois.org/moddoc/RoadNavigation/modinfo.html

- Be sure to position JeVois so that the row of green dots is on the horizon line of your scene.
- Move JeVois left and right: large green dot should move too, as it tracks the road's vanishing point.



Image credit: Kārlis Dambrāns / flickr

Road detection

320x256 @ 30fps

http://jevois.org/moddoc/RoadNavigation/modinfo.html

- Be sure to position JeVois so that the row of green dots is on the horizon line of your scene.
- Move JeVois left and right: large green dot should move too, as it tracks the road's vanishing point.



Image credit: Nico Kaiser / flickr

Road detection

320x256 @ 30fps

http://jevois.org/moddoc/RoadNavigation/modinfo.html

- Be sure to position JeVois so that the row of green dots is on the horizon line of your scene.
- Move JeVois left and right: large green dot should move too, as it tracks the road's vanishing point.



Image credit: Ian Sane / flickr

Road detection

320x256 @ 30fps

http://jevois.org/moddoc/RoadNavigation/modinfo.html

- Be sure to position JeVois so that the row of green dots is on the horizon line of your scene.
- Move JeVois left and right: large green dot should move too, as it tracks the road's vanishing point.



Image credit: Cliffano Subagio / flickr

Road detection

320x256 @ 30fps

http://jevois.org/moddoc/RoadNavigation/modinfo.html

- Be sure to position JeVois so that the row of green dots is on the horizon line of your scene.
- Move JeVois left and right: large green dot should move too, as it tracks the road's vanishing point.
- Algorithm is quite robust to perspective and appearance/lighting changes.
- Works even when there is no well defined road "edge."



Image credit: Cliffano Subagio / flickr

Object detection and matching

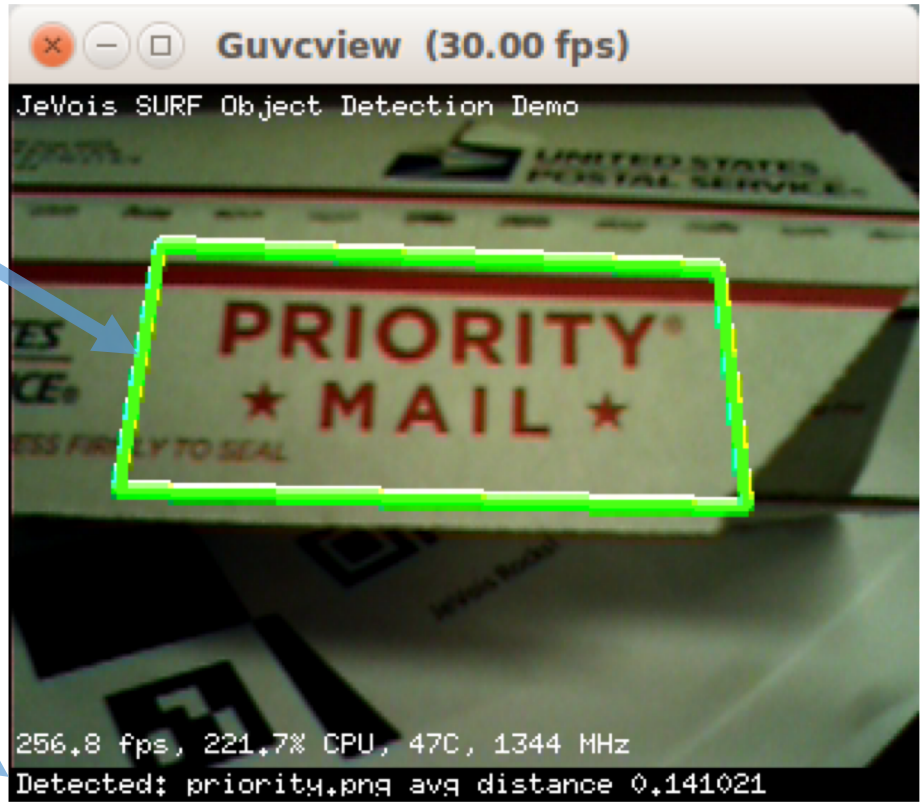
320x252 @ 30fps

http://jevois.org/moddoc/ObjectDetect/modinfo.html

- Finds objects by matching keypoint descriptors between the current image and a set of training images.
- 4 phases:
  - detect keypoint locations, typically corners or other distinctive texture elements or markings;
  - compute keypoint descriptors: summary representations of image patch around each keypoint;
  - match descriptors from current image to descriptors previously extracted from training images;
  - if enough matches found between current image and a given training image, draw green box.

Green box:
identified object

Text at bottom:
name of identified object



Object detection and matching

320x252 @ 30fps

http://jevois.org/moddoc/ObjectDetect/modinfo.html

- By default, this module is only trained for one object: Priority Mail logo (only the red part).
- See above documentation page and JeVois tutorials for how to train new objects live.



Salient object detection and matching

320x288 @ 30fps

http://jevois.org/moddoc/SaliencySURF/modinfo.html

- Combines saliency detection to locate objects, and keypoint matching.
- Recognizes salient objects by matching keypoint descriptors between a current set of salient regions and a set of training images.

Green box:
identified salient object

White box:
detected salient object
that was not recognized

Text at bottom:
names of identified objects

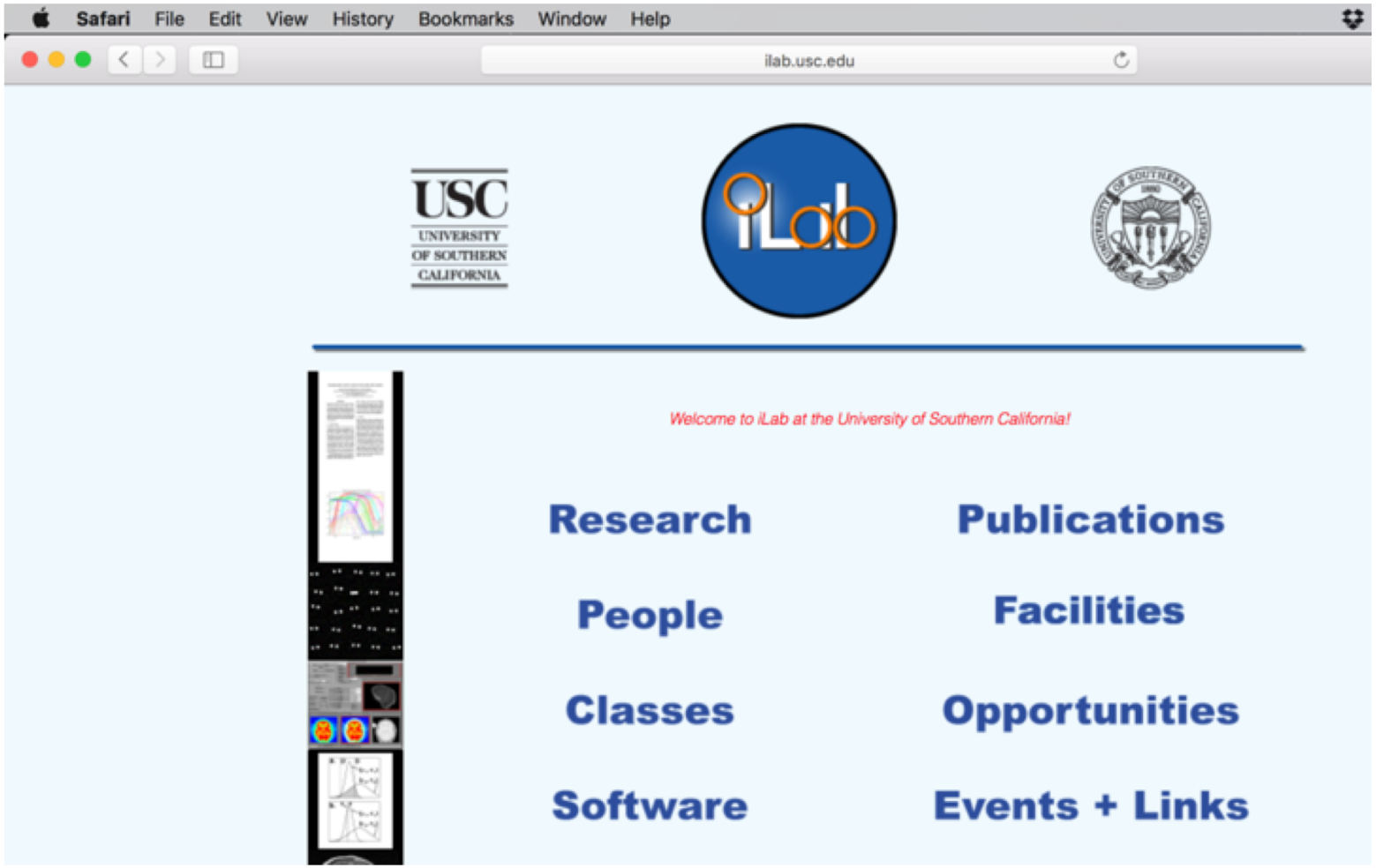


Salient object detection and matching

320x288 @ 30fps

http://jevois.org/moddoc/SaliencySURF/modinfo.html

- Trained by default on blue iLab logo, point JeVois to it and adjust distance so it fits in an attention box.
- Can easily add training images by just copying them to microSD card.
- Can tune number and size of salient regions, can save regions to microSD to create a training set.



Color-based object tracking

320x254 @ 60fps

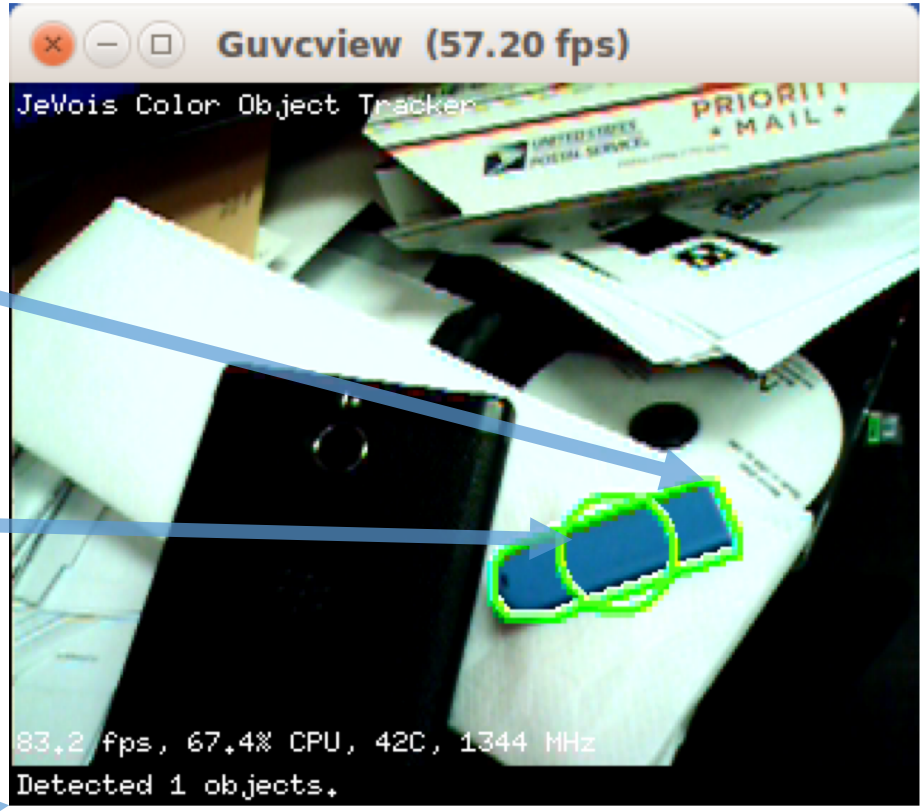
http://jevois.org/moddoc/ObjectTracker/modinfo.html

- Isolates pixels with a given hue, saturation, and brightness of color pixels, and extracts object contours.
- When image is clean (not too many blobs), identifies large enough blobs and mark them with a circle.

Green outlines:
all detected color blobs

Green circles:
centers of good
detected objects

Text at bottom:
number of good detected objects

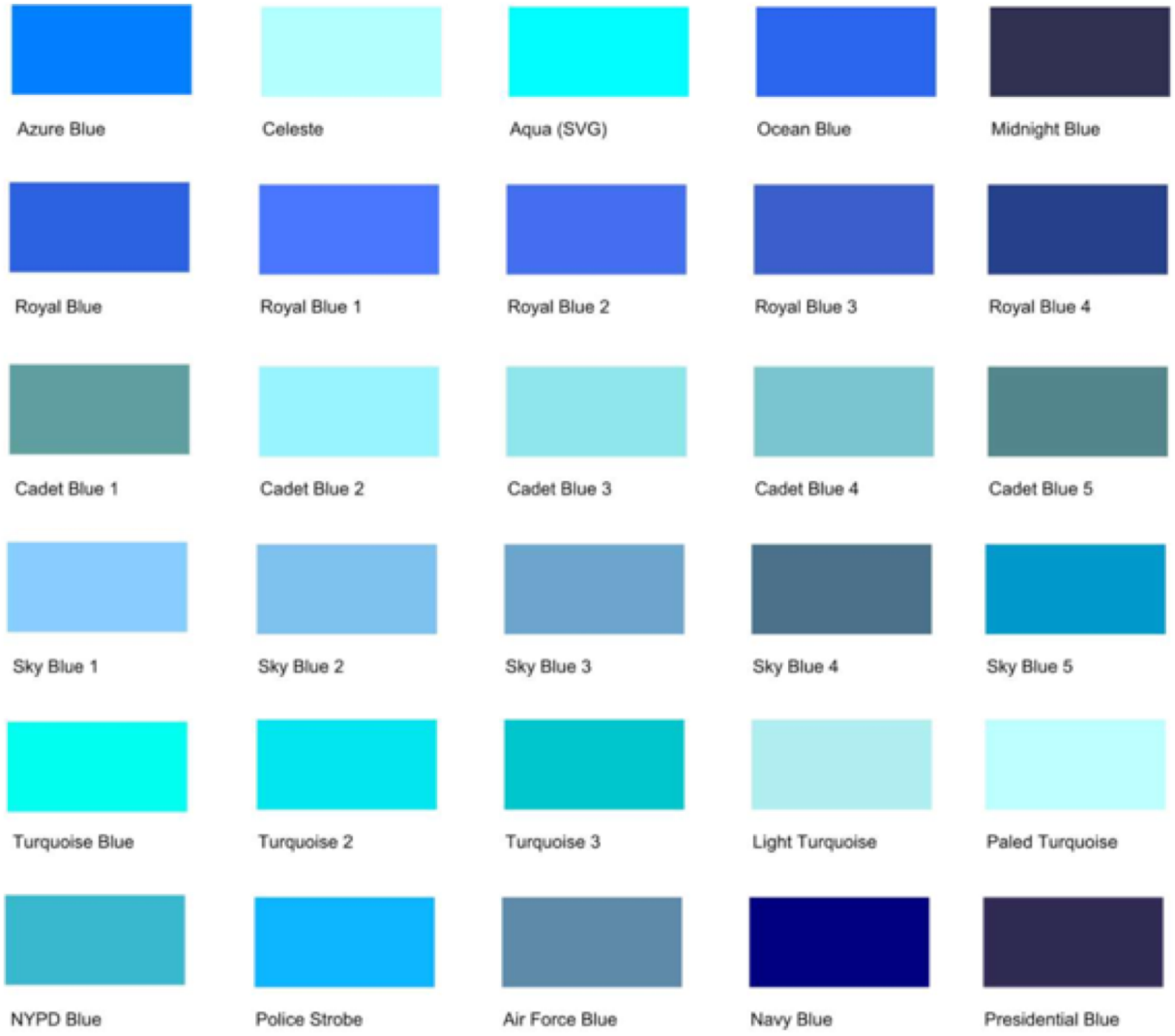


Color-based object tracking

320x254 @ 60fps

http://jevois.org/moddoc/ObjectTracker/modinfo.html

- Tuned by default for light blue objects. Can easily be changed using runtime parameters.
- Depending on your printer or screen, hopefully some of these objects should be detected.
- Try it with real blue objects!



source: drawingblog.mycoloringland.com

Object recognition with deep learning

544x240 @ 15fps
320x308 @ 30fps

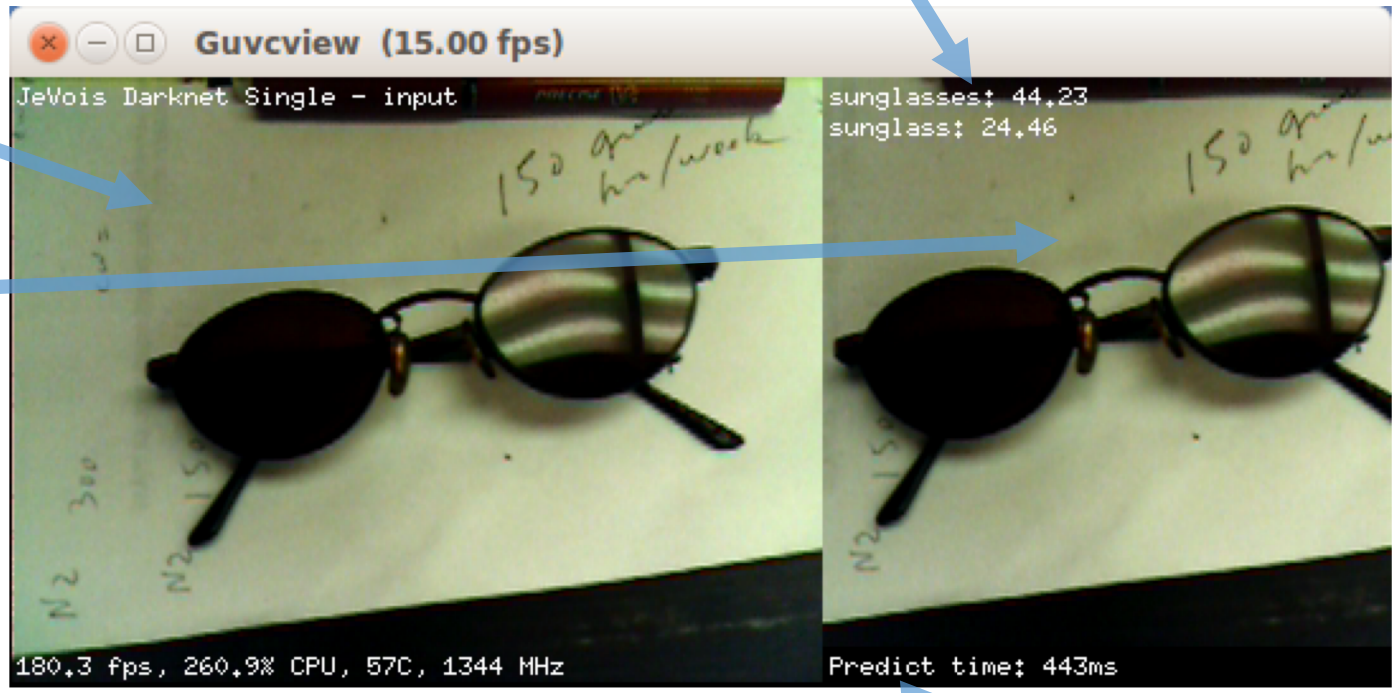
http://jevois.org/moddoc/DarknetSingle/modinfo.html
http://jevois.org/moddoc/TensorFlowEasy/modinfo.html

- Identifies the object in a square region in the center of the camera field of view using a deep convolutional neural network, trained to recognize 1000 different object types (object classes).
- Network size can be adjusted by changing video size. It directly affects both speed and accuracy. Larger networks run slower but are more accurate. See module documentation.

Top scoring object classes recognized by the deep network, and confidence scores (0 .. 100)

Live video at 15 frames/s

Central region that was last sent to the deep network



Time taken for neural network predictions on this image

Object recognition with deep learning

544x240 @ 15fps
320x308 @ 30fps

http://jevois.org/moddoc/DarknetSingle/modinfo.html
http://jevois.org/moddoc/TensorFlowEasy/modinfo.html

- Try it with these objects or any other object you like.
- Sometimes this module will make mistakes! This module was trained using 1.2 million images. Performance of Darknet is 58.7% correct (mean average precision) on a test set of 50,000 images.





Image credit: Virginia State Parks / flickr



Image credit: FotoSleuth / flickr



Image credit: Justin Seabrook-Rocha / flickr



Image credit: Pascal VOC



Image credit: Pascal VOC



Image credit: Pascal VOC



Image credit: Images Money / flickr



Image credit: Pascal VOC



Image credit: Pascal VOC



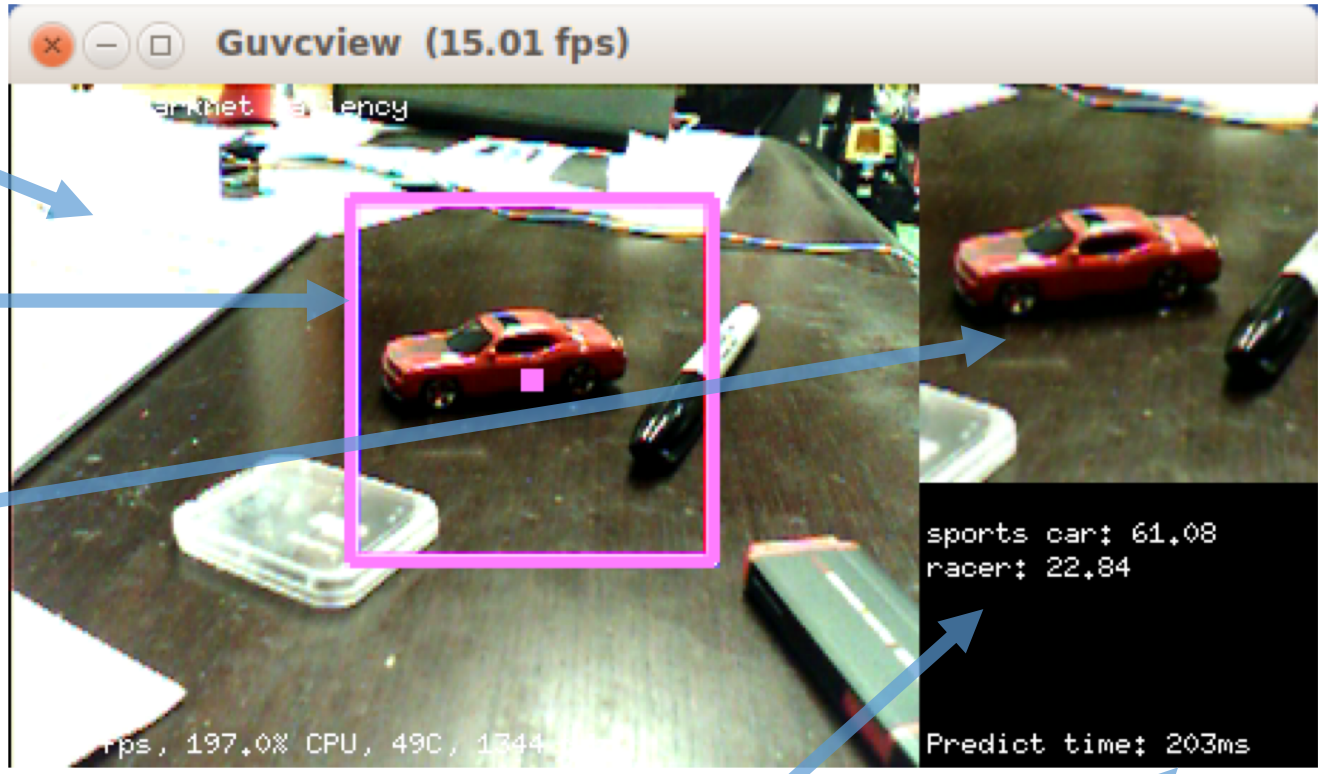
Image credit: fotosleuth / flickr

Attention + deep object recognition

460x240 @ 15fps
512x240 @ 30fps

http://jevois.org/moddoc/DarknetSaliency/modinfo.html
http://jevois.org/moddoc/TensorFlowSaliency/modinfo.html

- Find the most interesting object using visual attention algorithm, then identify it using a deep convolutional neural network, trained to recognize 1000 different object types (object classes).
- Network size can be adjusted by changing video size. It directly affects both speed and accuracy. Larger networks run slower but are more accurate. See module documentation.



Live video

Pink box: attended (most salient) region

Attended region that was last sent to the deep network

Top scoring object classes recognized by the deep network, and confidence scores (0 .. 100)

Time taken for neural network predictions on this image

Attention + deep object recognition

460x240 @ 15fps
512x240 @ 30fps

http://jevois.org/moddoc/DarknetSaliency/modinfo.html
http://jevois.org/moddoc/TensorFlowSaliency/modinfo.html

- Try it with the same images as the previous module!



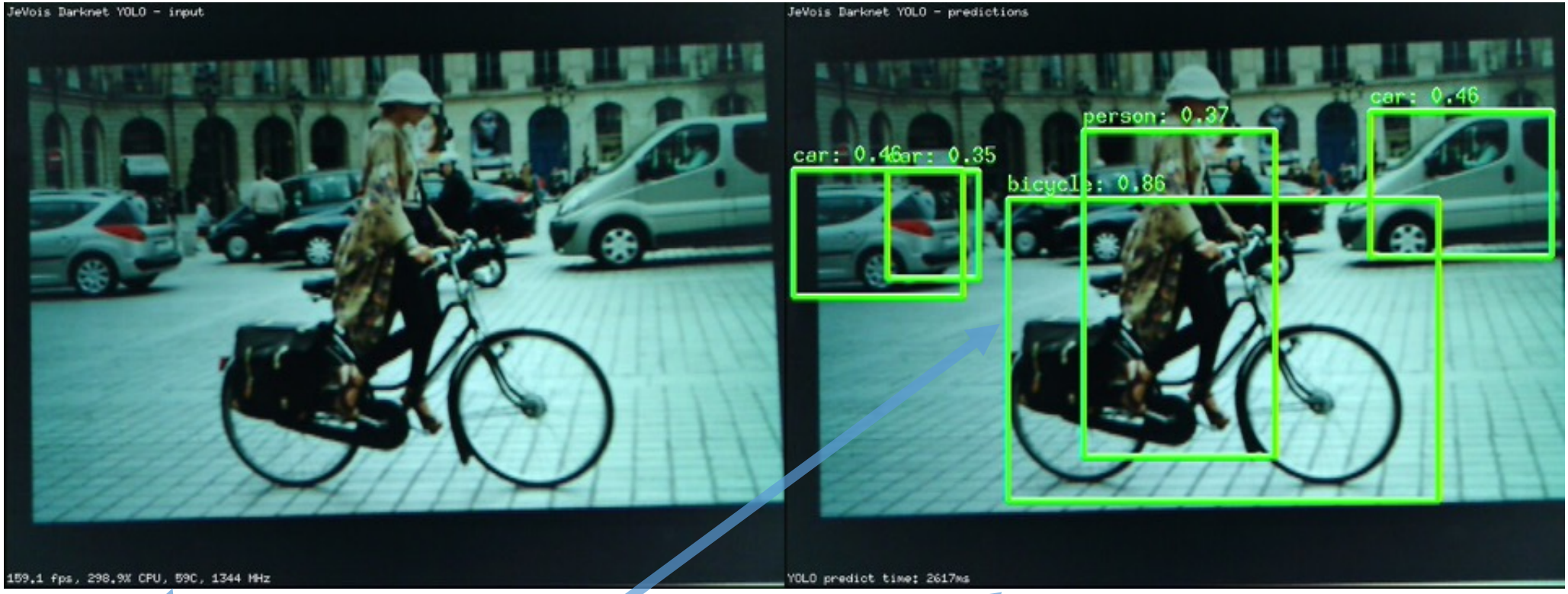
Image credit: Pascal VOC

Deep neural scene analysis

1280x480 @ 15fps

http://jevois.org/moddoc/DarknetYOLO/modinfo.html

- Detect and recognize all objects in the camera's view, using a state-of-the-art deep neural network for combined object localization and recognition.
- Quite slow but very powerful. Trained to recognize 20 object types: Aeroplanes, Bicycles, Birds, Boats, Bottles, Buses, Cars, Cats, Chairs, Cows, Dining tables, Dogs, Horses, Motorbikes, People, Potted plants, Sheep, Sofas, Trains, and TV/Monitors.



Live video at 15 frames/s

All objects detected and recognized, and confidence scores (0 .. 100)

Time taken for neural network predictions on this image

Deep neural scene analysis

1280x480 @ 15fps

http://jevois.org/moddoc/DarknetYOLO/modinfo.html



Image credit: Pascal VOC

Deep neural scene analysis

1280x480 @ 15fps

http://jevois.org/moddoc/DarknetYOLO/modinfo.html



Image credit: Pascal VOC

Deep neural scene analysis

1280x480 @ 15fps

http://jevois.org/moddoc/DarknetYOLO/modinfo.html



Image credit: Pascal VOC

Deep neural scene analysis

1280x480 @ 15fps

http://jevois.org/moddoc/DarknetYOLO/modinfo.html



Image credit: Pascal VOC

Deep neural scene analysis

1280x480 @ 15fps

http://jevois.org/moddoc/DarknetYOLO/modinfo.html



Image credit: Pascal VOC

Deep neural scene analysis

1280x480 @ 15fps

http://jevois.org/moddoc/DarknetYOLO/modinfo.html



Image credit: Pascal VOC

Deep neural scene analysis

1280x480 @ 15fps

http://jevois.org/moddoc/DarknetYOLO/modinfo.html



Image credit: Pascal VOC

Deep neural scene analysis

1280x480 @ 15fps

http://jevois.org/moddoc/DarknetYOLO/modinfo.html



Image credit: Pascal VOC

Deep neural scene analysis

1280x480 @ 15fps

http://jevois.org/moddoc/DarknetYOLO/modinfo.html







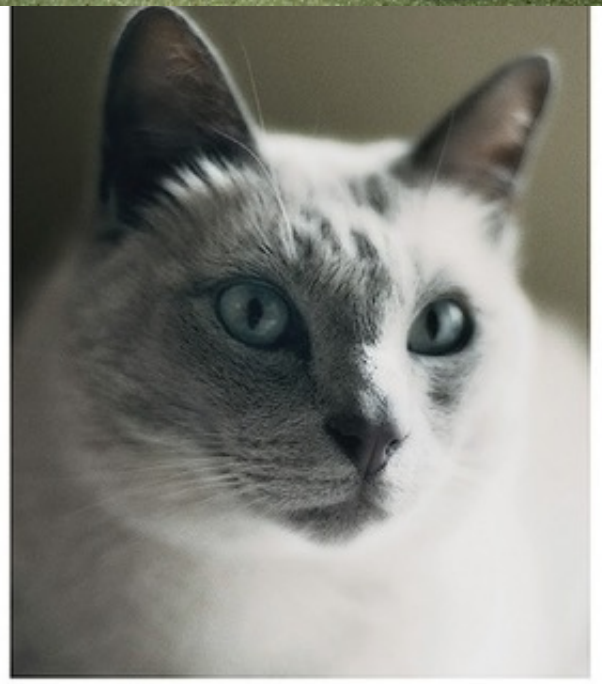


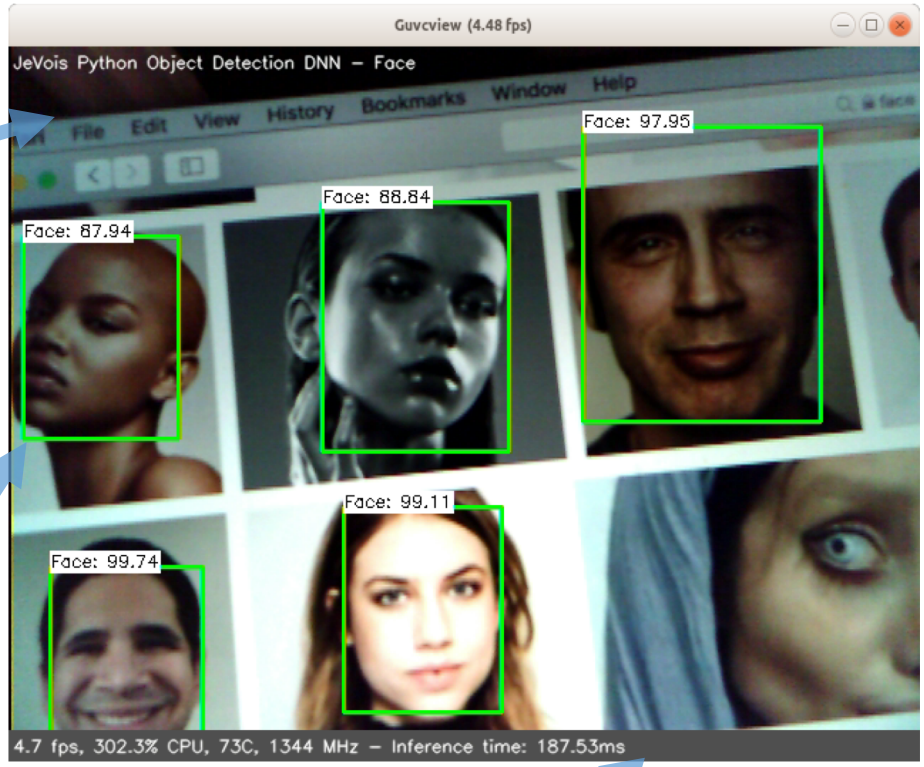
Image credit: Pascal VOC

Face detection with deep networks

640x498 @ 15fps
640x502 @ 20fps

http://jevois.org/moddoc/DetectionDNN/modinfo.html
http://jevois.org/moddoc/DetectionDNN/modinfo.html

- Detect and recognize all objects in the camera's view, using a state-of-the-art deep neural network for combined object localization and recognition.
- Can be configured to run a variety of object detection networks. Runs the OpenCV Face Detection network by default.



Live video

All objects detected and recognized, and confidence scores (0 .. 100)

Time taken for neural network predictions on this image

Facial emotion classification

320x336 @ 15fps

http://jevois.org/moddoc/PyEmotion/modinfo.html

- Analyze facial emotions along 8 dimensions.
- This module does not do face detection. It assumes that the camera is pointed at a face that is well centered in the field of view and occupies most of the image.



Live video

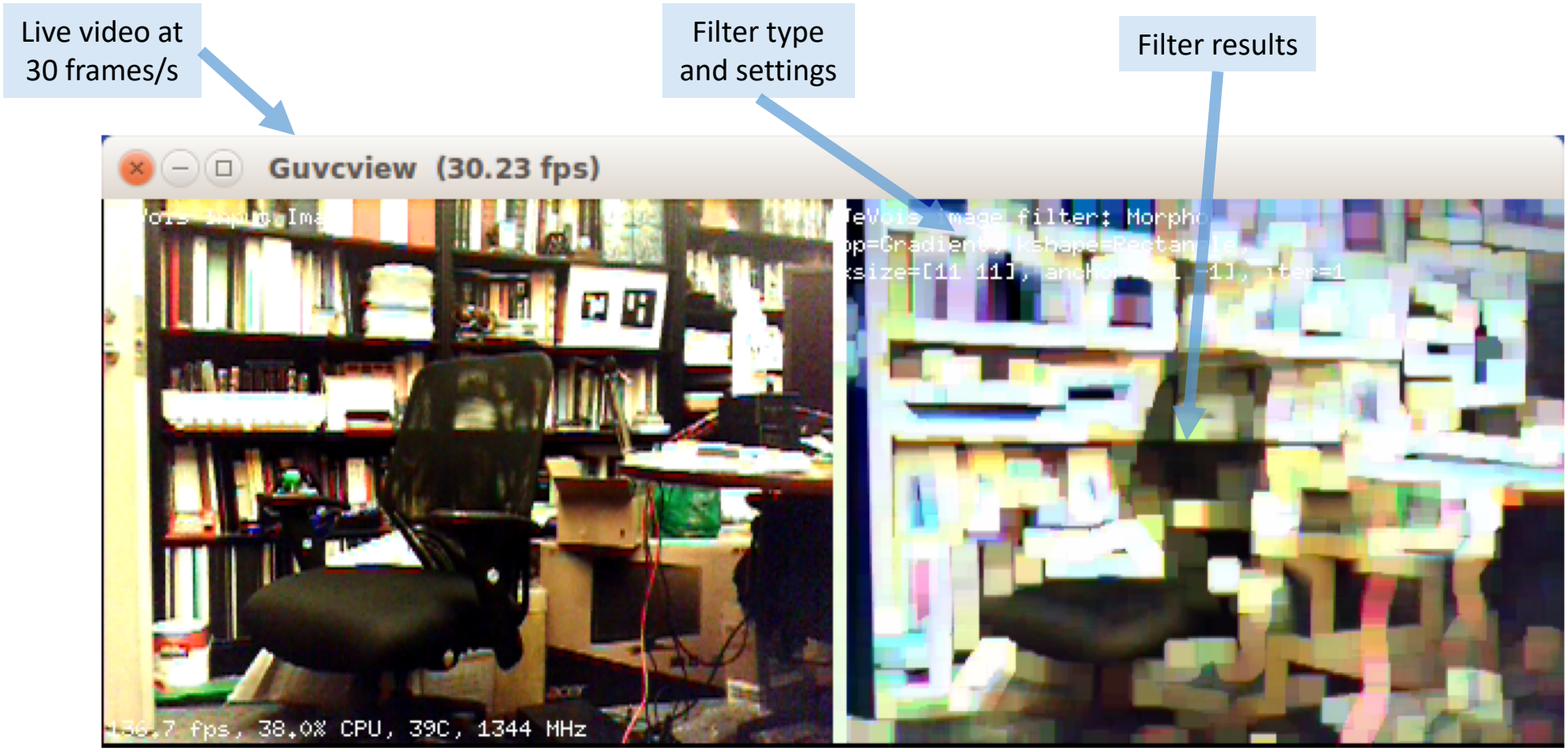
Facial emotion analysis along 8 criteria (scores in -1000 to 1000)

Color filtering

640x240 @ 30fps

http://jevois.org/moddoc/ColorFiltering/modinfo.html

- This module is to learn about basic image filtering. It was developed to allow students to instantly observe the effects of different filters and their parameters on live video. The module implements a variety of filters using OpenCV. Each filter exposes some paremeters (e.g., kernel size) that can be set interactively to understand their effects onto the filter behavior.
- To trigger different filters, you need to send commands to JeVois. See module documentation.



Save video to microSD

320x240 @ 60fps
176x144 @ 120fps

http://jevois.org/moddoc/SaveVideo/modinfo.html

- Records video and saves it to the MicroSD card inside the JeVois smart camera.
- Useful, for example, to record footage that will be used to train some machine vision algorithm.
- To start saving: connect to JeVois and send "`start`" command; to stop, send "`stop`" command.
- Successive start/stop cycles save to different video files.
- Access the files either live by exporting microSD as virtual flash drive, or offline by taking microSD out of JeVois and inserting it into your desktop or laptop.
- Supports high frame rates, e.g., 120fps at 176x144 or 88x72.

Recording indicator.
Turns to RECORDING after start
Turns to NOT RECORDING after stop

Recorded file name and directory on microSD card



Moving object detection

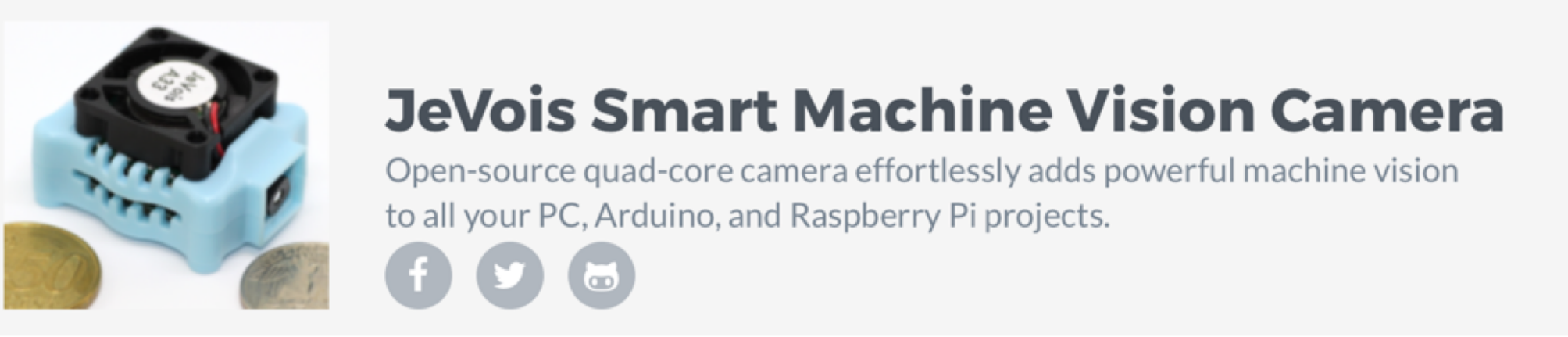
320x120 @ 30fps

http://jevois.org/moddoc/DemoBackgroundSubtract/modinfo.html

- Learns a statistical model of the scene over time, that is, learns the color and brightness of each pixel, including small variations due to noise, small changes in lighting, etc.
- When an image region becomes very different than the learned model, show it in the right panel as detected foreground region.
- Keep JeVois stable and wait for a few seconds. Right panel should turn mostly black. Now move you hand or some object in front of JeVois. It should appear as a white region in the right panel.



- Note: sometimes, moving a large object in front of JeVois will trigger a change in auto exposure setting, which may make the whole scene brighter, and hence the whole scene will be detected as changing on the right side. This module is hence best used with manual exposure and gain settings (see module documentation for details).



JeVois vision modes with greyscale outputs

- These may not work with all video capture software.

- Known issues with Windows and AMCap when trying to capture greyscale video (not only from JeVois).

- See http://jevois.org/start for more information.

Optical flow

GRAY 176x288 @ 100fps

http://jevois.org/moddoc/OpticalFlow/modinfo.html

- Computes horizontal and vertical optical flow into two separate optical flow maps.
- Very fast: 100 frames/s with 176x144 input video.
- Intended for use as a pre-processor: images sent out may be further processed by the host computer.
- Try it: move your hand or other objects in front of JeVois.

Horizontal motion map:
black: rightward motion
white: leftward motion
grey: no horizontal motion

Vertical motion map:
black: downward motion
white: upward motion
grey: no vertical motion



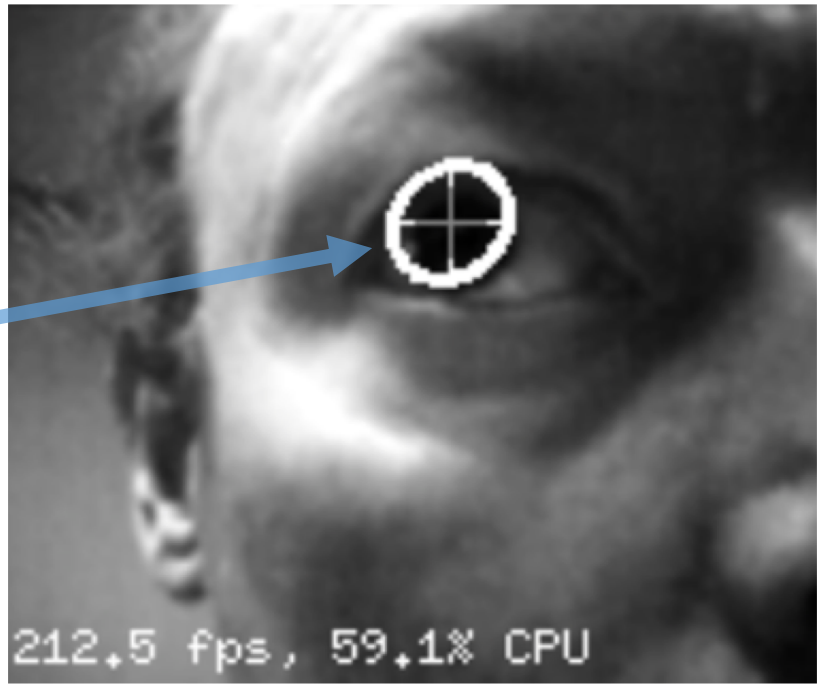
Eye tracking

GRAY 176x144 @ 120fps

http://jevois.org/moddoc/DemoEyeTracker/modinfo.html

- Implements an eye tracker, which is based on detecting the outline of the pupil.
- Used in research on attention, eye movements, psychophysics, human-computer interfaces, user interface design, gaming, advertising, driving safety, and neurology.
- Note: need to switch video capture software to grayscale mode (not supported by all software).
- Supports high frame rates, e.g., 30fps at 640x480, 60fps at 320x240, 120fps at 176x144.
- Note that the camera has to be very close to the eye for this to work well with stock optics. To be useful in practice, some prism or tele-lens should be used so that the camera can be out of the field of view of the human participant.

Crosshair: center of the pupil
Ellipse: edges of pupil

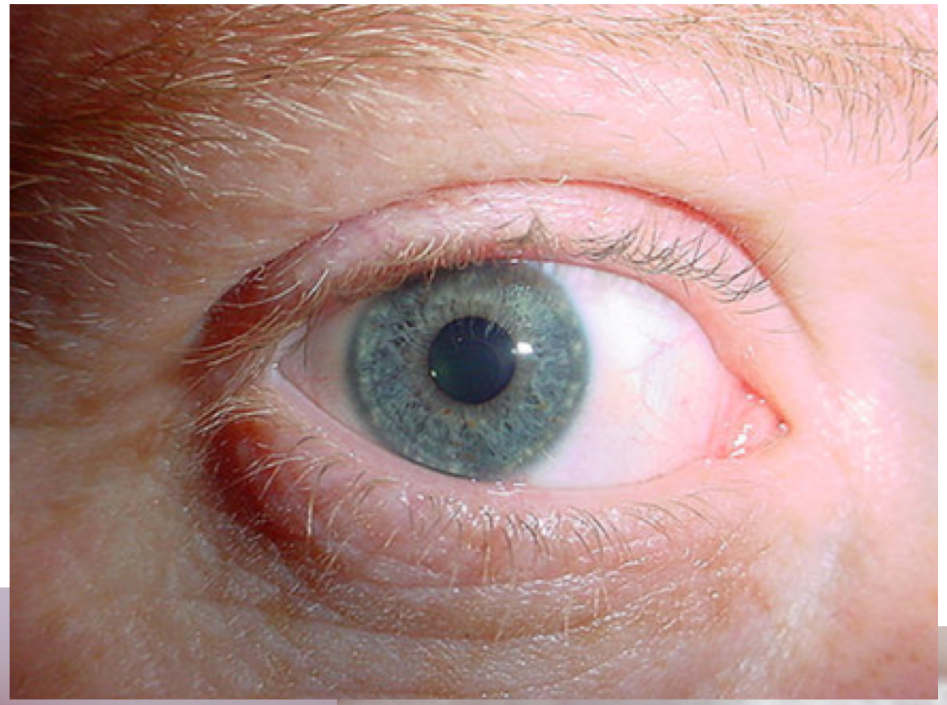


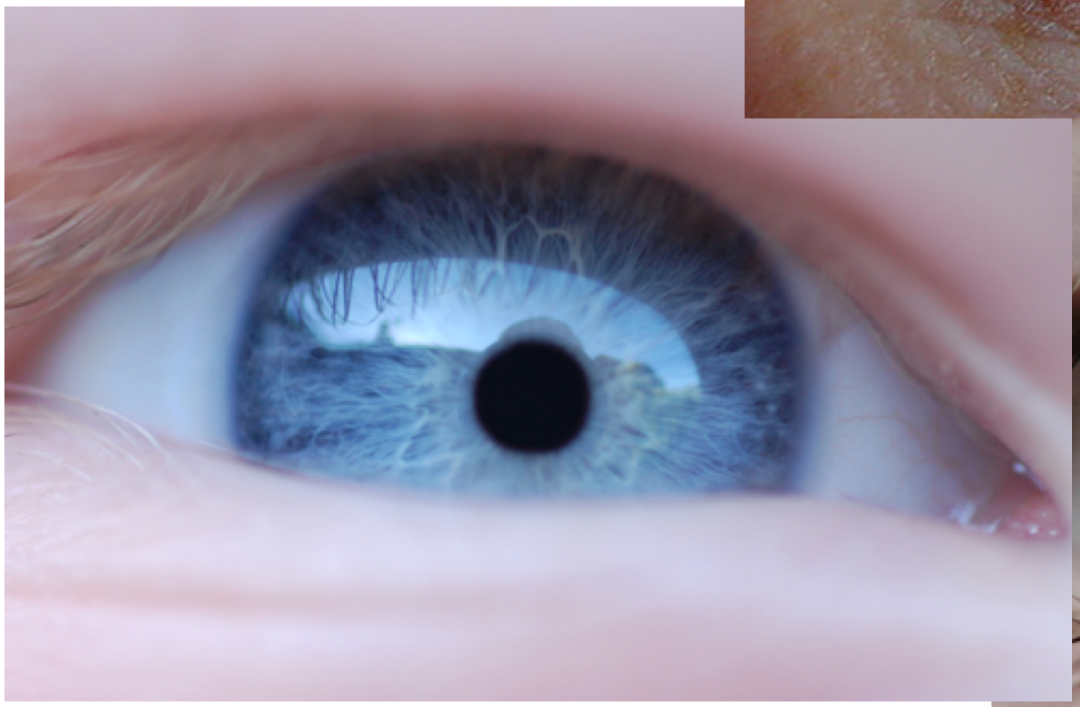
Eye tracking

GRAY 176x144 @ 120fps

http://jevois.org/moddoc/DemoEyeTracker/modinfo.html

- Try it: can it detect these pupils?





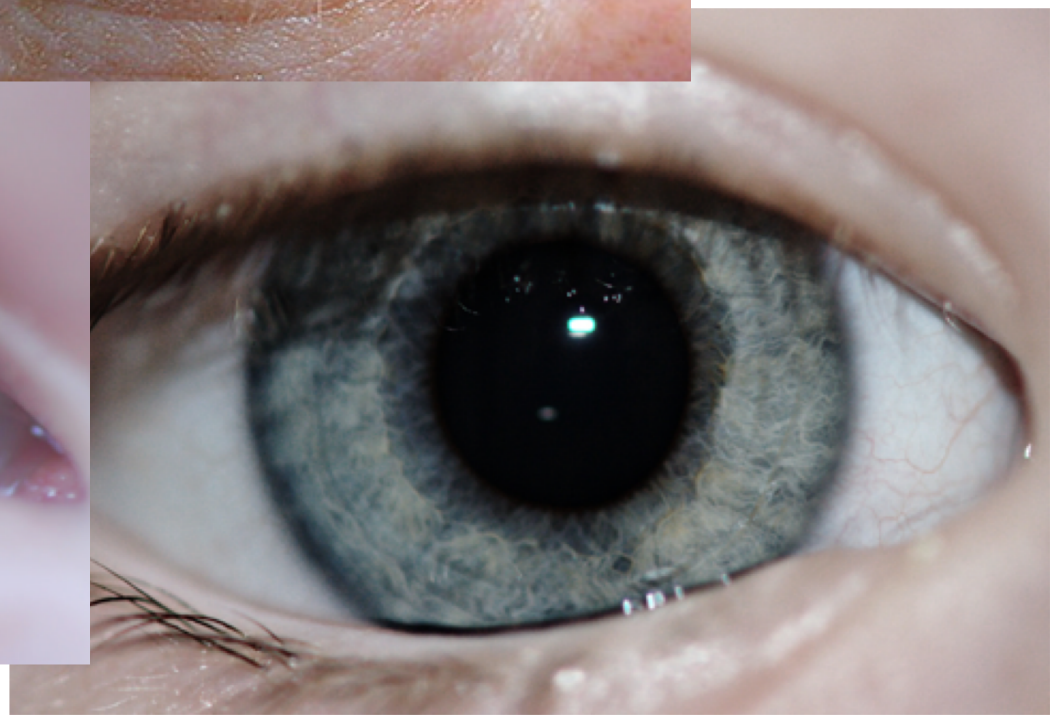


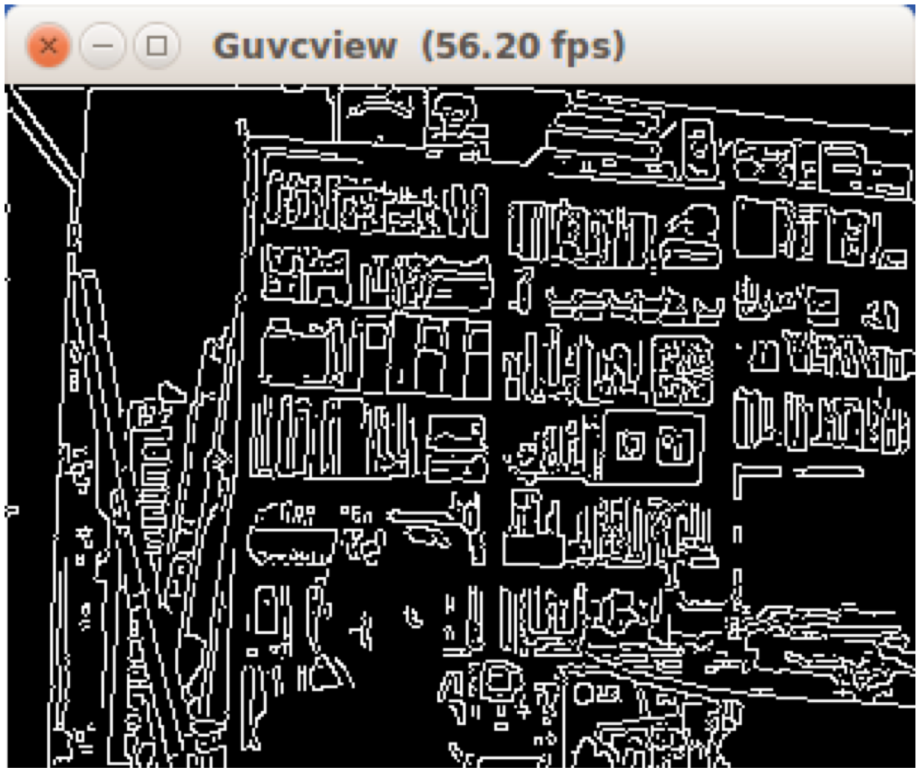
Image credit: steve lodefink, Glenn Reilly, Paul Scott / flickr

Edge detection

GRAY 640x480 @ 29fps

http://jevois.org/moddoc/EdgeDetection/modinfo.html

- Compute edges in an image using the Canny edge detection algorithm.
- Intended as a pre-processor, delivering edge maps to a host computer, which may then be in charge of further processing them, for example to detect objects of various shapes.
- Runs at 60+ frames/s with resolution 320x240 on the JeVois camera.
- Try it: point JeVois to anything!

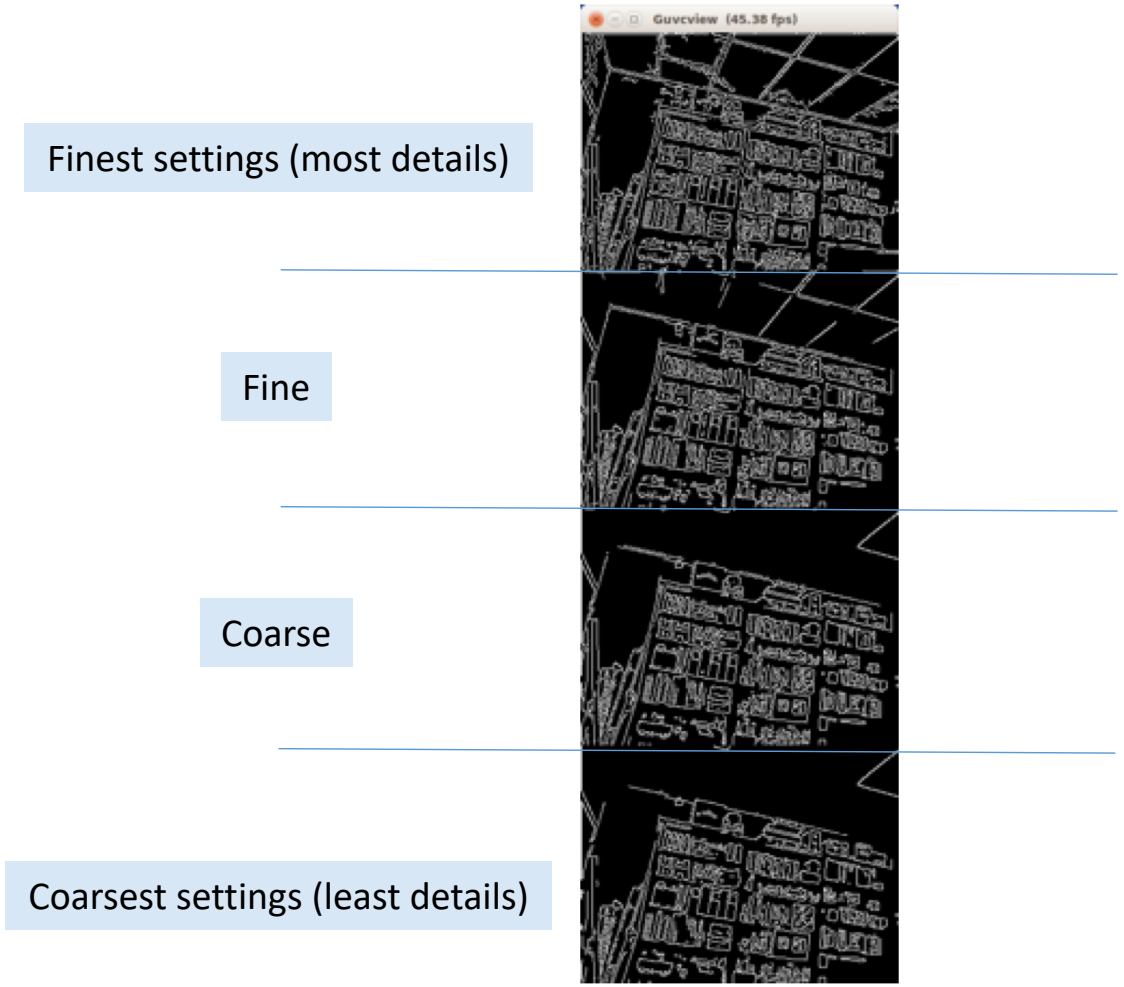


Edge detection X4

GRAY 320x960 @ 45fps

http://jevois.org/moddoc/EdgeDetectionX4/modinfo.html

- Compute 4 Canny edge detection filters with 4 different settings, in parallel.
- Intended as a pre-processor, delivering edge maps to a host computer, which may then be in charge of further processing them, for example to detect objects of various shapes and sizes.
- Runs at 45+ frames/s with resolution 320x240 on the JeVois camera.
- Try it: point JeVois to anything!



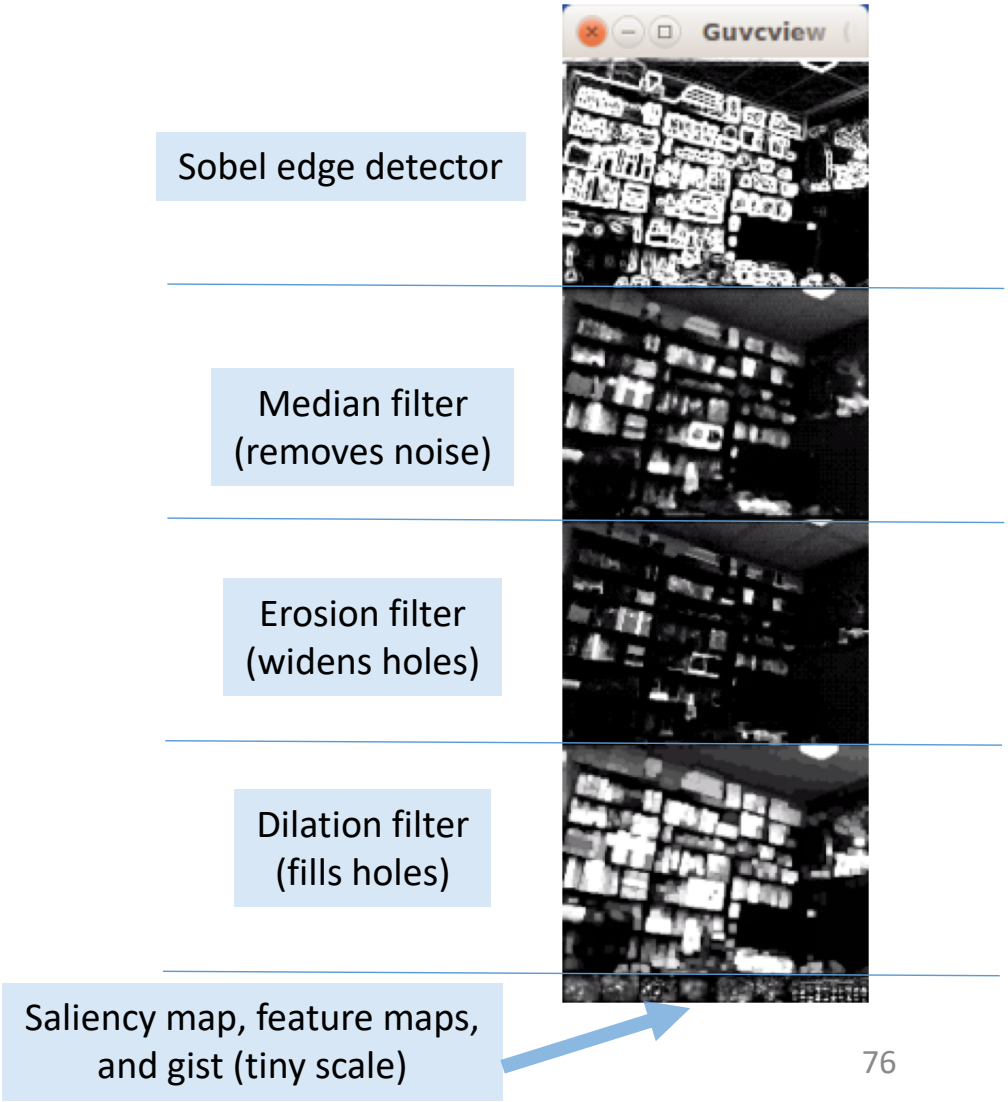
GPU image processing

GRAY 160x495 @ 60fps

http://jevois.org/moddoc/DemoCPUGPU/modinfo.html

- GPU (graphics processing unit) is a second processor element inside JeVois. Normally it is used to create smooth 3D displays. But, since JeVois has no screen, we can use it instead for machine vision!
- This module computes saliency and gist over our 4 CPU cores while also computing 4 different image filters over the dual-core GPU, finally combining all results into a single grayscale image:

  - saliency: multicore CPU-based detection of the most attention-grabbing location.
  - GPU filter 1: Sobel edge detector.
  - GPU filter 2: Median filter.
  - GPU filter 3: Morphological erosion filter.
  - GPU filter 4: Morphological dilation filter.





JeVois Smart Machine Vision Camera

Open-source quad-core camera effortlessly adds powerful machine vision to all your PC, Arduino, and Raspberry Pi projects.

There is more!

- Please visit http://jevois.org for more information.
- All available machine vision modules are listed at: http://jevois.org/doc/UserDemos.html

- You can program your own modules using Python + OpenCV 3.3
- You can also program your own modules in C++

- With JeVois, the future of machine vision is in your hands!



JeVois Smart Machine Vision Camera

Open-source quad-core camera effortlessly adds powerful machine vision to all your PC, Arduino, and Raspberry Pi projects.

Copyright © 2017-2019 by JeVois Inc.
All rights reserved.

JeVois® is a registered trademark of JeVois Inc, a California Corporation.

http://jevois.org