Computer Vision with Cognitive Services and Open CV

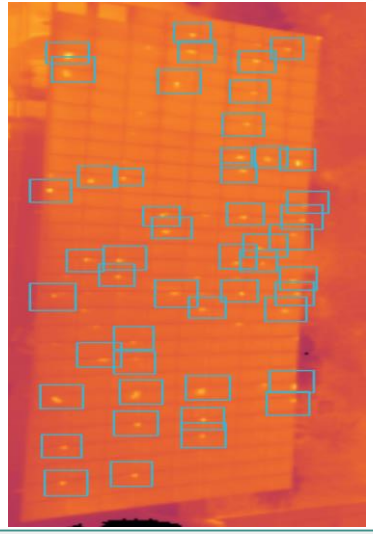
Daniel Egan, Microsoft

Adnan Masood, PhD. UST Global

Daniel Egan

Sr Software Engineer – Microsoft
Consumer Software Engineering

Three real projects



Adnan Masood

Adnan Masood, Ph.D. is an Artificial Intelligence and Machine Learning researcher, software architect, and Microsoft MVP (Most Valuable Professional) for Data Platform. As Chief Architect of AI and Machine Learning at UST Global, he collaborates with Stanford Artificial Intelligence Lab, and MIT AI Lab for building enterprise solutions.

Author of Amazon bestseller in programming languages, **"Functional Programming with F#",** Dr. Masood teaches Data Science at Park University, and has taught Windows Communication Foundation (WCF) courses at the University of California, San Diego. He is a regular speaker to various academic and technology conferences (WICT, DevIntersection, IEEE-HST, IASA, and DevConnections), local code camps, and user groups. He also volunteers as STEM (Science Technology, Engineering and Math) robotics coach for elementary and middle school students.

A strong believer in giving back to the community, Dr. Masood is a co-founder and president of the Pasadena .NET Developers group, co-organizer of Tampa Bay Data Science Group, and Irvine Programmer meetup. His recent talk at Women in Technology Conference (WICT) Denver highlighted the importance of diversity in STEM and technology areas, and was featured by variety of news outlets.

Microsoft® MVP Most Valuable Professional

Business Problem

The space between the top of the cart and the bottom of the cart is responsible for millions of dollars worth of merchandise that is unpaid. How do we stop this pragmatically?

Key Business Value:

Reducing Fraud is adding millions back to the bottom line without doing much. By having a camera that scans the bottom of the cart, cashiers can make sure that all products are scanned and accounted for.

Technologies Used:

Computer vision, Object Detection, and Machine learning.

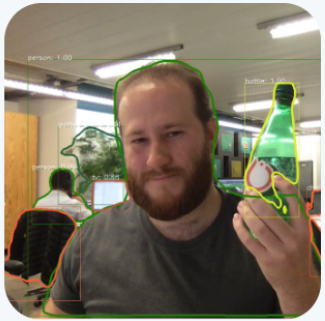
Industry Sector / Domain:

Retail industry

DEV intersection

Christopher Manning @chrmanning · Nov 1

Just a couple of years ago, I found it hard to believe that vision people were still all working with rectangular bounding boxes. I guess they've fixed that now. 🙂

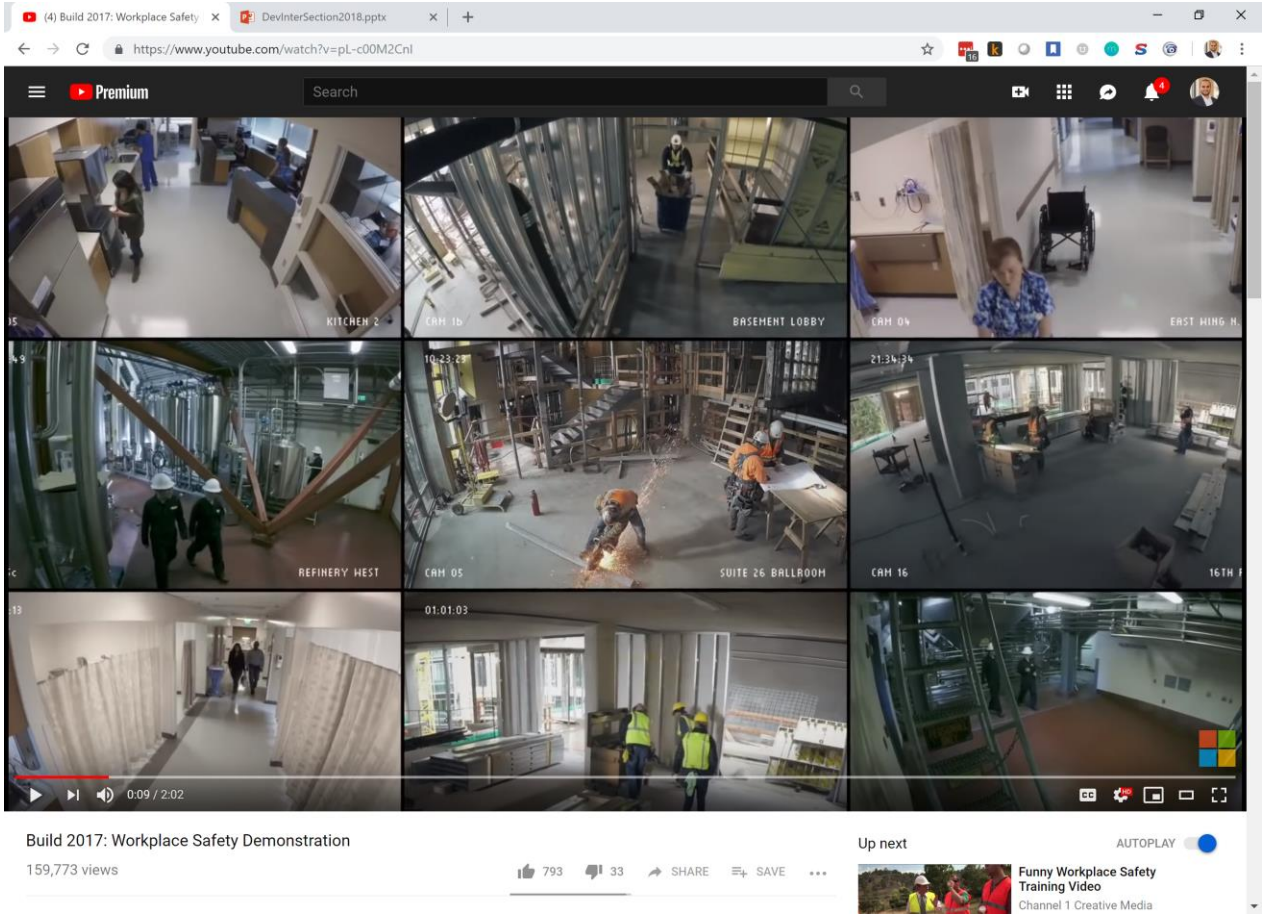


Francisco Massa @fvsmassa

Today we are releasing Mask R-CNN Benchmark: a fast and modular implementation for Faster R-CNN and Mask R-CNN written entirely in @PyTorch 1.0. It brings up to 30% speedup compared to mmdetection during training....

4 55 271

intersection



DEV intersection

Microsoft AI Services

With Azure AI services you can build next-gen of smart applications where your data lives, in the intelligent cloud, on-premises and on the intelligent edge. AI services include from pre-built APIs, such as Cognitive Services & Conversational AI with Bot tools, to building custom models with Azure Machine Learning for any scenario.

Microsoft Cognitive Toolkit / Deep Learning

A free, open-source, commercial-grade toolkit that trains deep learning algorithms to learn like human brain.

- Empowers you to harness the intelligence within massive datasets
- Provides uncompromised scale, speed, and accuracy with commercial-grade quality
- Compatibility with languages you use like C++ and Python

Intelligent Applications

Experience the intelligence built into Microsoft products and services you use every day – from Office to Skype to Calendar

- Intelligent Search – Makes it easier to find what you're looking for
- Microsoft Translator - translate real time via text, voice and video across experiences like Skype
- AI in Office 365 - Connect to relevant information, and surface new insights

Azure Bot Framework

Build, connect, deploy, and manage intelligent bots to interact naturally with your users on websites, and other apps.

- Integrated environment that is purpose-built for bot development with the Microsoft Bot Framework
- Give your bot intelligence with Cognitive Services
- Integration across multiple channels – Website, App, Email, Skype, Slack, Cortana, Facebook Messenger and more

Azure Machine Learning

A fully-managed cloud service that enables you to easily build, deploy and share predictive analytics solutions.

- Container based AI deployment from cloud to edge
- Python pip-installable extensions for Azure ML – Build & deploy highly accurate ML & deep learning models
- Package pre-built models like predictive maintenance, image classification & speech processing and deploy to IoT devices

Cognitive Services

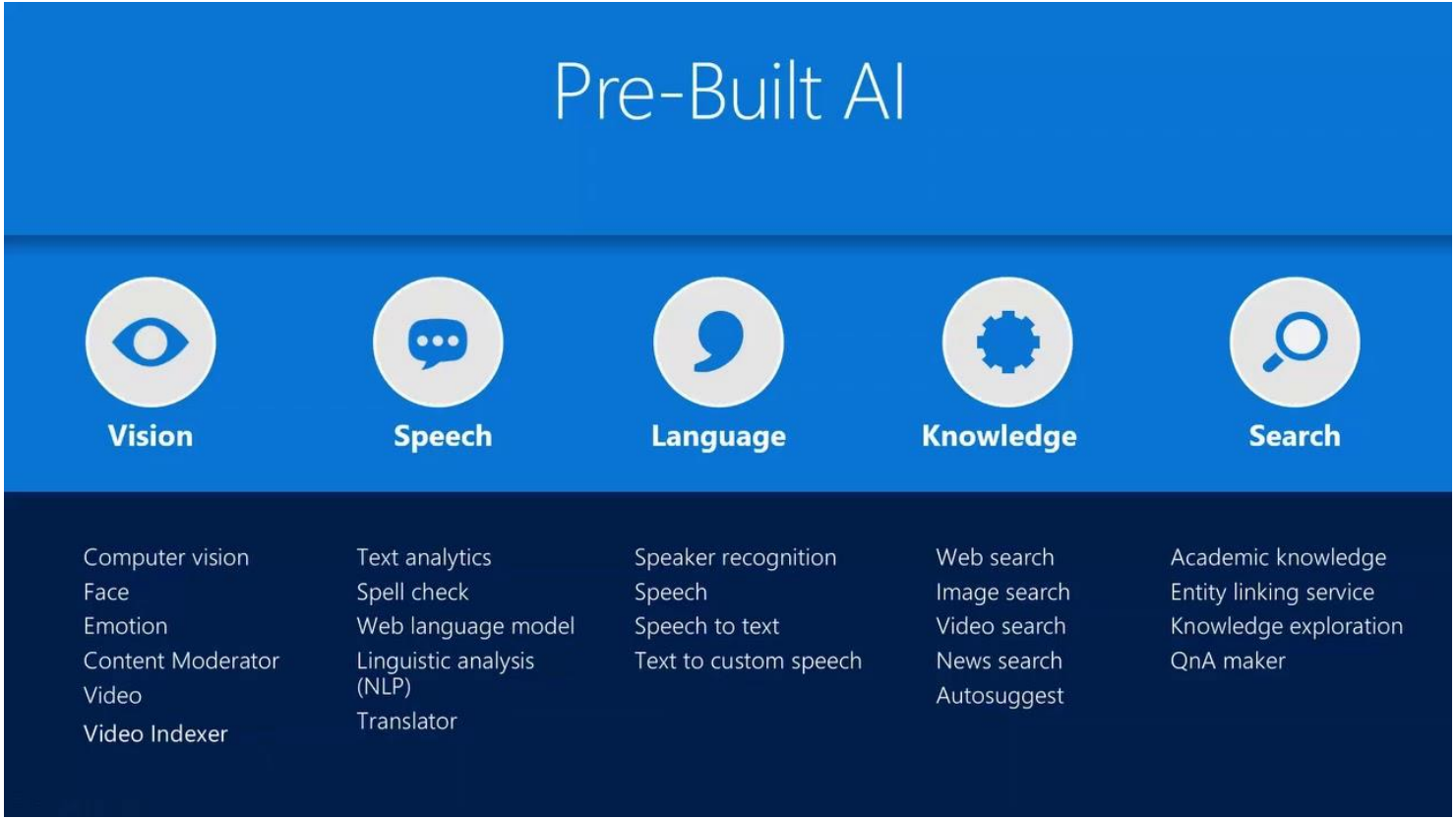
APIs and services that allow systems to speak, hear, see, understand and interpret human needs

- APIs for speech, vision, language, search, and knowledge
- Customised to your organisation's availability, security and compliance requirements
- Works across devices and platforms such as iOS, Android, and Windows

Ask us for an Envisioning Session on Microsoft AI Services.

Source **WinWire** via **@BrianJohnson_01**

DEV intersection

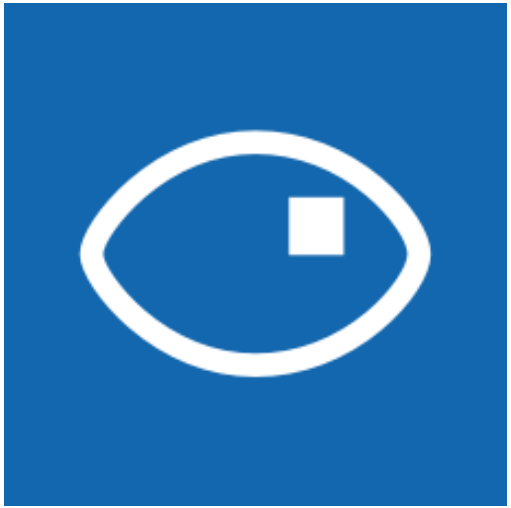


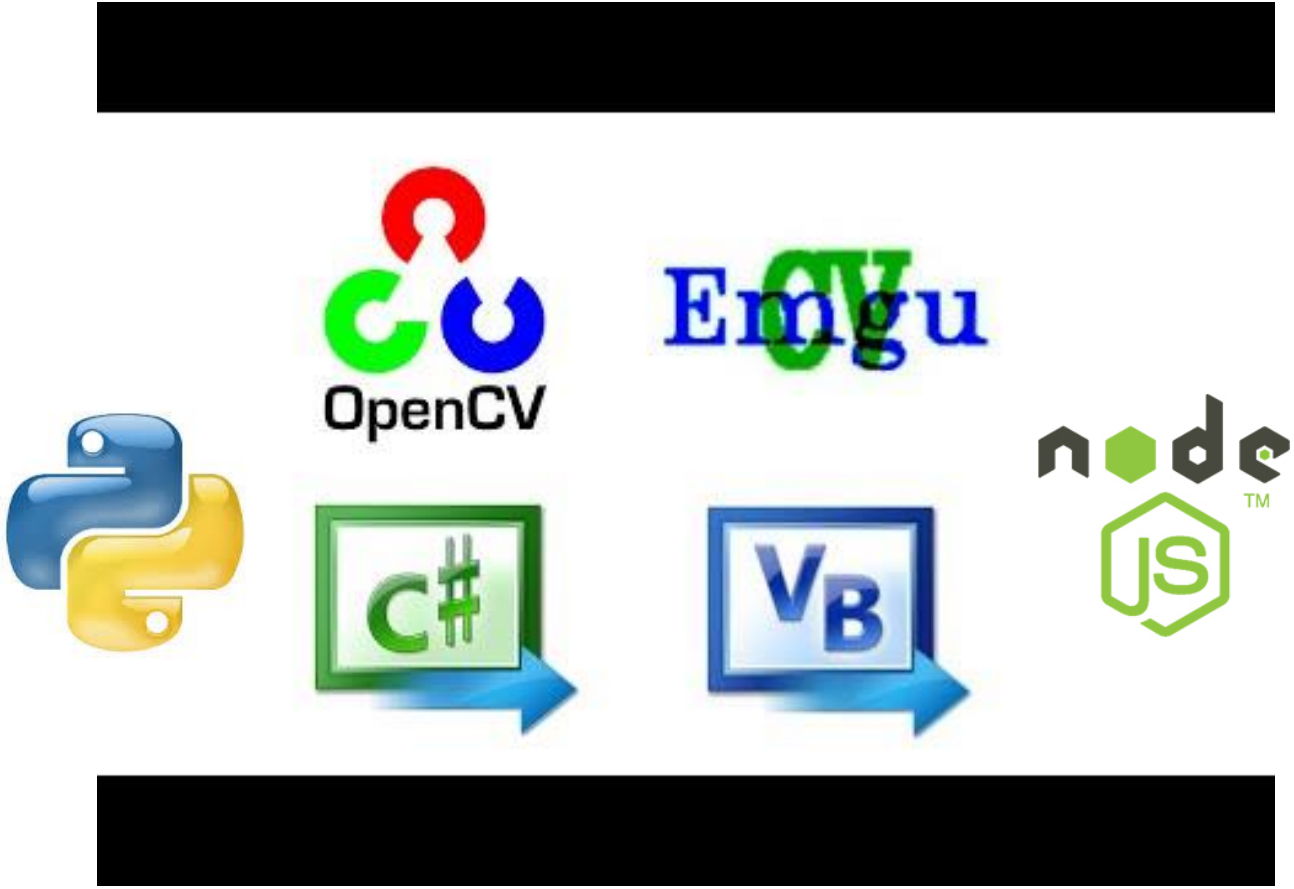
Cognitive Services

microsoft.com/cognitive

Vision	Speech	Language	Knowledge	Search
Computer Vision	Custom Recognition	Bing Spell Check	Academic Knowledge	Bing Web Search
Emotion	Speaker Recognition	Linguistic Analysis	Entity Linking	Bing Image Search
Face	Speech	Language Understanding	Knowledge Exploration	Bing Video Search
Video	Translator	Text Analytics	Recommendations	Bing News Search
		WebLM		Bing Autosuggest

Three stepping stones




OpenCV
Emgu CV
node
JS
C#
VB

Computer Vision

Analyze Image Service

Understand content and features within an image

Analyze Image – Example

Type of Image:

Clip Art Type	0 Non-clipart
Line Drawing Type	0 Non-Line Drawing
Black & White Image	False

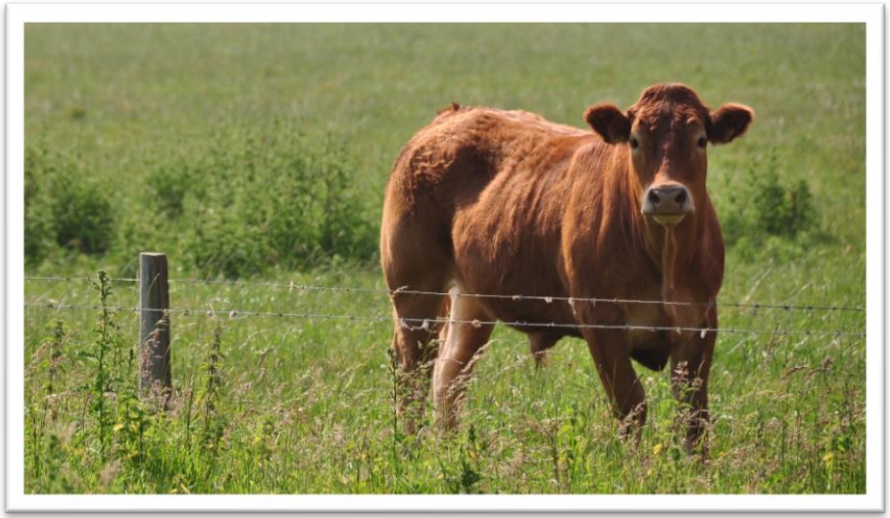
Content of Image:

Categories	[{ “name”: “people_swimming”, “score”: 0.099609375 }
Adult Content	False
Adult Score	0.18533889949321747
Faces	[{ “age”: 27, “gender”: “Male”, “faceRectangle”: {“left”: 472, “top”: 258, “width”: 199, “height”: 19

Image Colors:

Dominant Color Background	White
Dominant Color Foreground	Grey
Dominant Colors	White
Accent Color	

Analyze Image – Example



Content of Image:

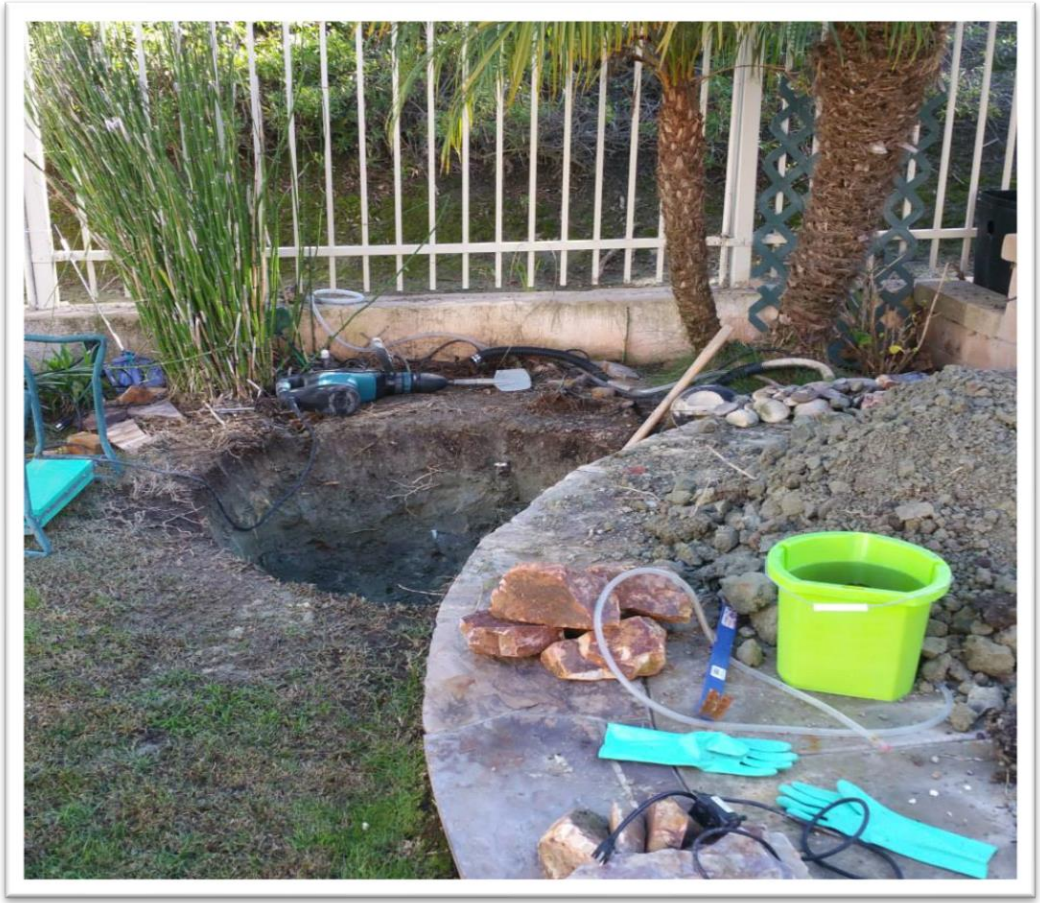
Categories


```
v0: [{ “name”: “animal”, “score”: 0.9765625 }]
V1: [{ "name": "grass", "confidence": 0.9999992847442627 },
{ "name": "outdoor", "confidence": 0.9999072551727295 },
{ "name": "cow", "confidence": 0.99954754114151 },
{ "name": "field", "confidence": 0.9976195693016052 },
{ "name": "brown", "confidence": 0.988935649394989 },
{ "name": "animal", "confidence": 0.97904372215271 },
{ "name": "standing", "confidence": 0.9632768630981445 },
{ "name": "mammal", "confidence": 0.9366017580032349,
"hint": "animal" },
{ "name": "wire", "confidence": 0.8946959376335144 },
{ "name": "green", "confidence": 0.8844101428985596 },
{ "name": "pasture", "confidence": 0.8332059383392334 },
{ "name": "bovine", "confidence": 0.5618471503257751,
"hint": "animal" },
{ "name": "grassy", "confidence": 0.48627158999443054 },
{ "name": "lush", "confidence": 0.1874018907546997 },
{ "name": "staring", "confidence": 0.165890634059906 }]
```


Describe


```
0.975 "a brown cow standing on top of a lush green field“
0.974  “a cow standing on top of a lush green field”
0.965  “a large brown cow standing on top of a lush green field”
```

Wackoon



Wackcoon

Wackcoon

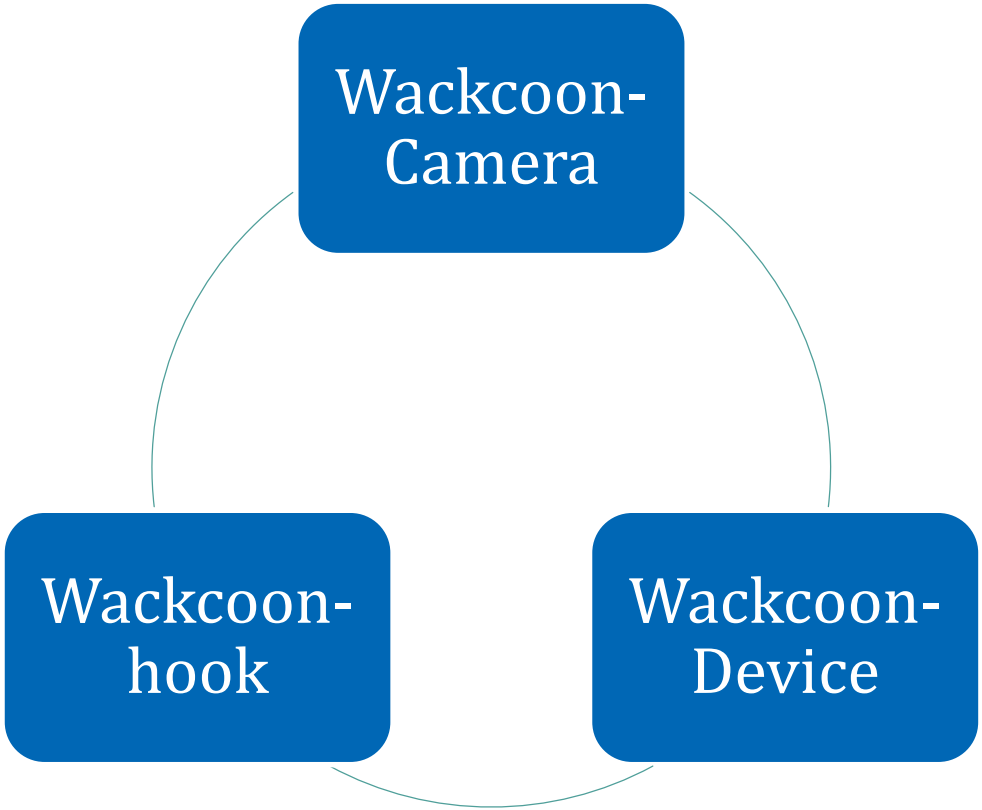
Wackcoon

Wackcoon

Content of Image:

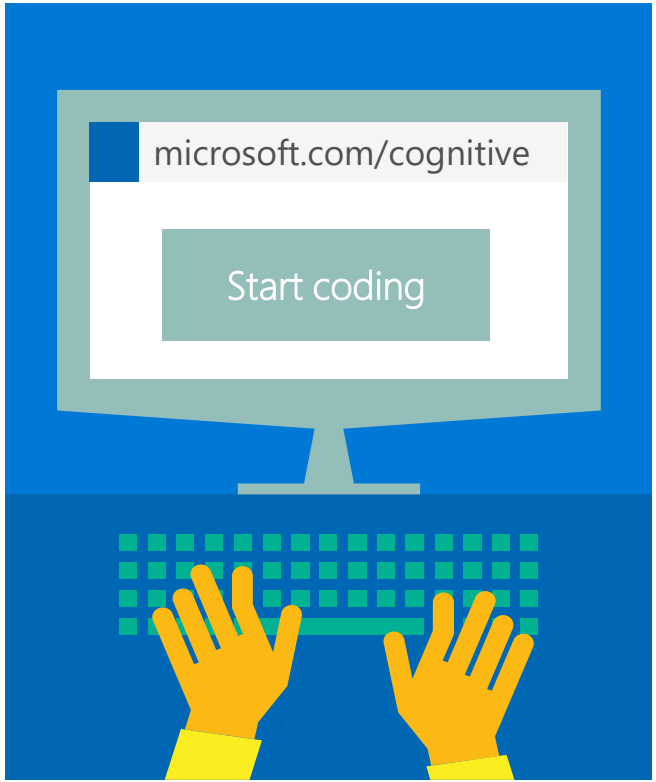
Categories


```
v0: [{ “name”: “animal”, “score”: 0.9765625 }]
V1: [{ "name": "grass", "confidence": 0.9999992847442627 },
{ "name": "outdoor", "confidence": 0.9999072551727295 },
{ "name": "cat", "confidence": 0.99954754114151 },
{ "name": "raccoon", "confidence": 0.9976195693016052 },
{ "name": "grey", "confidence": 0.988935649394989 },
{ "name": "animal", "confidence": 0.97904372215271 },
{ "name": "standing", "confidence": 0.9632768630981445 },
{ "name": "mammal", "confidence": 0.9366017580032349,
"hint": "animal" },
{ "name": "aquarium", "confidence": 0.8946959376335144 },
{ "name": "green", "confidence": 0.8844101428985596 },
{ "name": "grass", "confidence": 0.8332059383392334 },
{ "name": "water", "confidence": 0.5618471503257751 },
{ "name": "grassy", "confidence": 0.48627158999443054 },
{ "name": "lush", "confidence": 0.1874018907546997 },
{ "name": "staring", "confidence": 0.165890634059906 }]
```

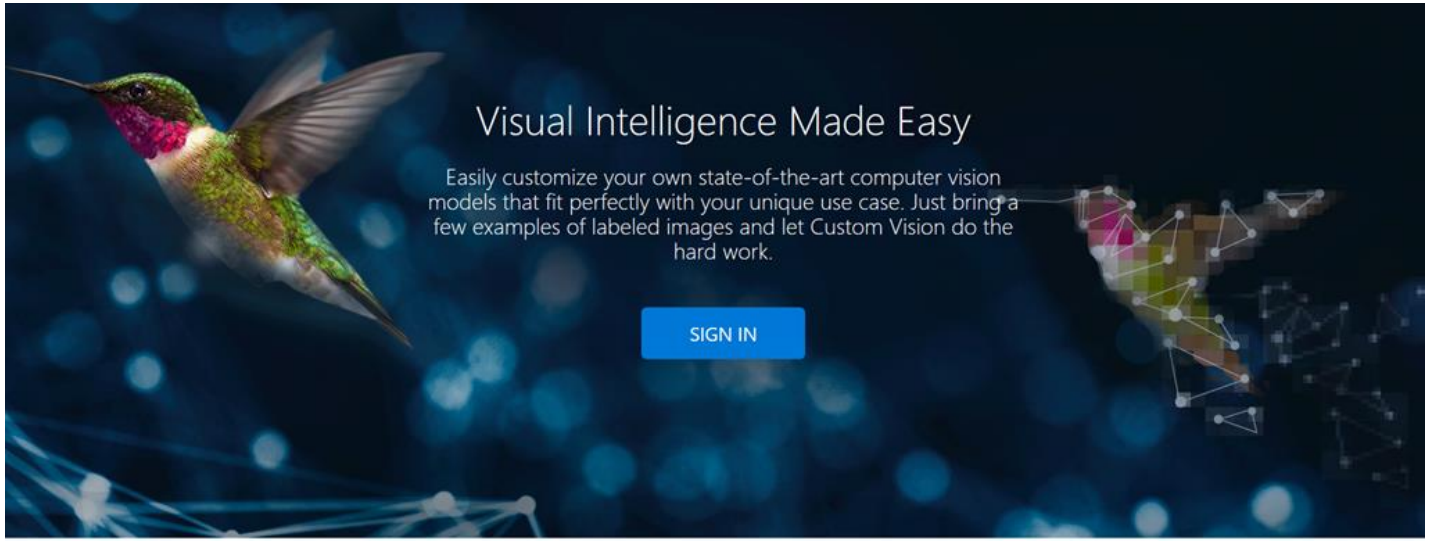

Wackcoon-Camera
Wackcoon-hook
Wackcoon-Device

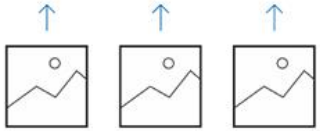
Demo

Lets see the online API



Custom Vision





Upload Images

Bring your own labeled images, or use Custom Vision to quickly add tags to any unlabeled images.

Train

Use your labeled images to teach Custom Vision the concepts you care about.

Evaluate

Use simple REST API calls to quickly tag images with your new custom computer vision model.

AI Problems

Dog?

Mmm…Blueberries

Bread?

Pastries anyone?

Ice Cream?

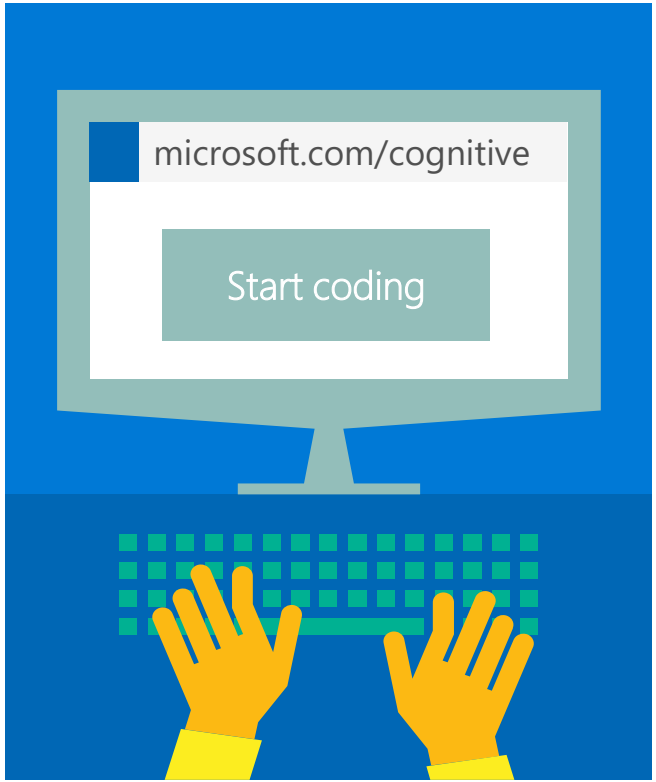
Results

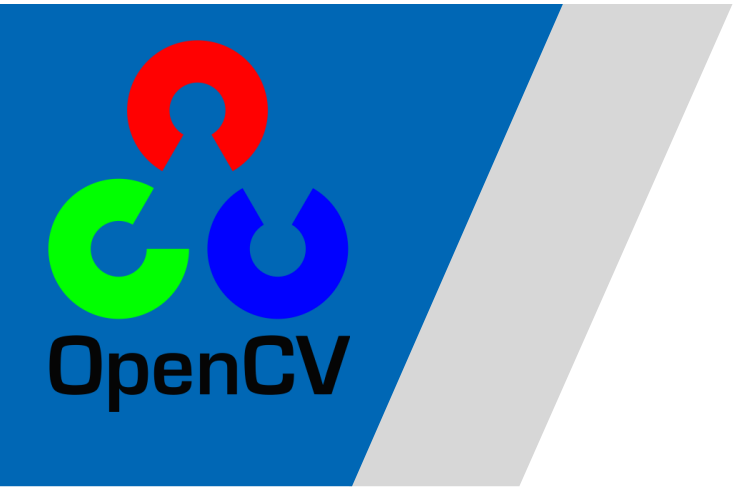
Tag	Probability
daisy	99.9%
trillium	3.1%
lily of the valley	0.1%
dogwood	0.0%

DEV intersection

Demo

Lets see the Custom Vision





Open CV

OpenCV – Introduction

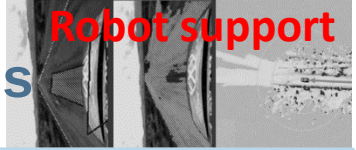
- OpenCV stands for the Open Source Computer Vision Library.
  - It was founded at Intel in 1999, went through some lean years after the .bust but is now under active development, now receiving ongoing support from Willow Garage.
- OpenCV is free for commercial and research use.
  - It has a BSD license. The library runs across many platforms and actively supports Linux, Windows and Mac OS.
- OpenCV was founded to advance the field of computer vision.
  - It gives everyone a reliable, real time infrastructure to build on. It collects and makes available the most useful algorithms.

OpenCV - Features

- Cross Platform
  - Windows, Linux, Mac OS
- Portable
  - iPhone
  - Android.
- Language Support
  - C/C++
  - Python

OpenCV Overview:

> 500 functions



General Image Processing Functions

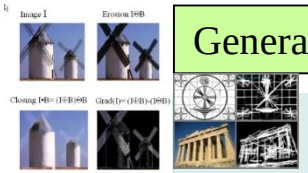
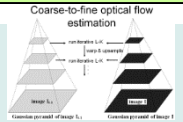
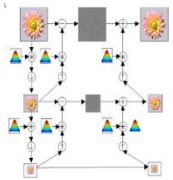
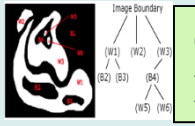


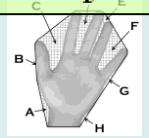
Image Pyramids



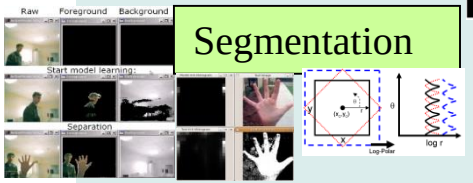


Geometric Descriptors

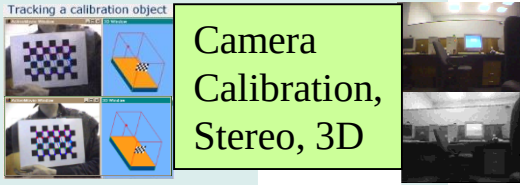




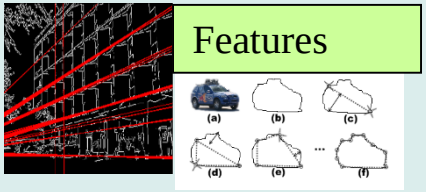
Segmentation



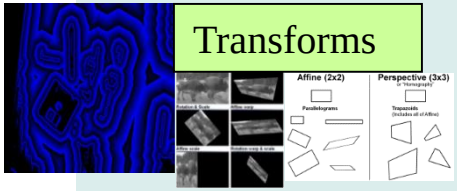
Camera Calibration, Stereo, 3D



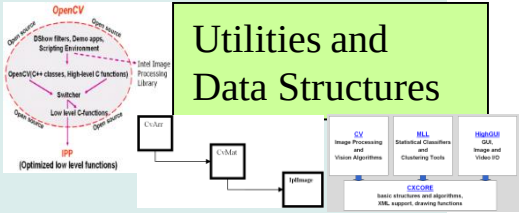
Features



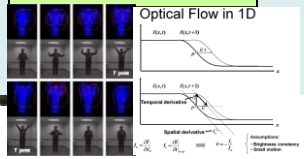
Transforms



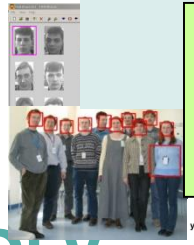
Utilities and Data Structures



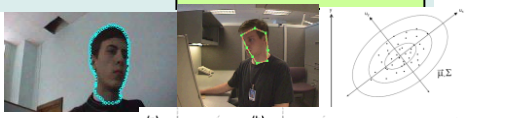
Tracking

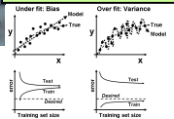


Machine Learning:
•Detection,
•Recognition

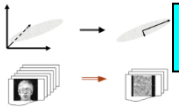


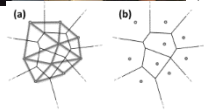
Fitting





Matrix Math





DEV intersection

History

- 1999 Jan : lanched by Intel, real time machine vision library for UI, optimized code for intel
- 2000 Jun : OpenCV alpha 3。
- 2000 Dec : OpenCV beta 1 for linux
- 2006 : the first 1.0 version supports Mac OS
- 2008 mid : obtain corporate support from Willow Garage
- 2009 Sep : OpenCV 1.2（beta2.0
- 2009 Oct : Version 2.0 released。
- 2010 Dec : OpenCV 2.2。
- 2011 Aug : OpenCV 2.3。
- 2012 Apr : OpenCV 2.4.
- 2015 – Jun: OpenCV 3.0

DEV intersection

OpenCV 4.0

Nov 20, 2018

After almost 3.5 years since groundbreaking 3.0 release, we are glad to present the first stable release in the 4.x line.

Release highlights:

- OpenCV is now C++11 library and requires C++11-compliant compiler. Minimum required CMake version has been raised to 3.5.1.
- A lot of C API from OpenCV 1.x has been removed.
- Persistence (storing and loading structured data to/from XML, YAML or JSON) in the *core* module has been completely reimplemented in C++ and lost the C API as well.
- New module G-API has been added, it acts as an engine for very efficient graph-based image procesing pipelines.
- *dnn* module was updated with Deep Learning Deployment Toolkit from the OpenVINO™ toolkit R4. See the guide how to build and use OpenCV with DLDT support.
- *dnn* module now includes experimental Vulkan backend and supports networks in ONNX format.
- The popular Kinect Fusion algorithm has been implemented and optimized for CPU and GPU (OpenCL)
- QR code detector and decoder have been added to the *objdetect* module
- Very efficient and yet high-quality DIS dense optical flow algorithm has been moved from opencv_contrib to the *video* module.
- More details can be found in previous announces: 4.0-alpha, 4.0-beta, 4.0-rc and in the changelog

Branch 3.4 will be switched to maintanence mode: only bugfixes and light features will be accepted. **BTW, release 3.4.4 is ready too!**

For those who have not took part in the OpenCV 2018 survey yet, feel free to share your thoughts.

Overview

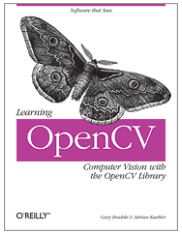
- **Goals**
  - Develop a universal toolbox for research and development in the field of Computer Vision
- **Algorithms**
  - More than 350 algorithms, 500 API
- **Programming language**
  - C/C++, C#, Ch , Python, Ruby, Matlab, and Java (using JavaCV)
- **OS support**
  - Windows, Android, Maemo, FreeBSD, OpenBSD, iOS, Linux and Mac OS.
- **Licence**
  - BSDlisence, free for commercial and non-commmercial

Overview - Applications

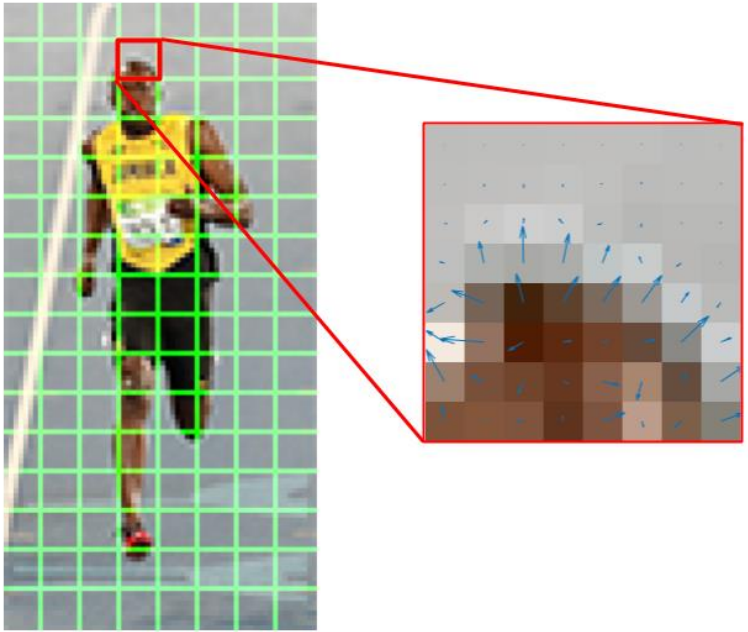
- 2D and 3D feature toolkits
- Egomotion estimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding
- Object identification
- Segmentation and Recognition
- Stereopsis Stereo vision: depth perception from 2 cameras
- Structure from motion (SFM)Motion tracking

OpenCV – Getting Started

- Download OpenCV
  - http://opencv.org
  - Install from macports/aptitude
- Setting up
  - Comprehensive guide on setting up OpenCV in various environments at the official wiki.
- Online Reference:
  - http://docs.opencv.org
- Two books



Histogram of Oriented Gradients



2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

DEV intersection

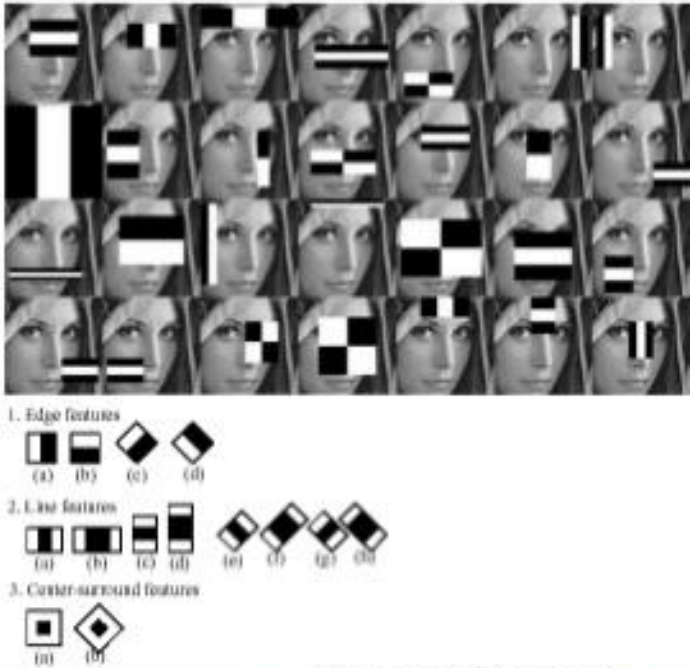
Haar Feature-based Cascade Classifier

OpenCV/EmguCV uses a type of face detector called a **Haar Cascade.**

The Haar Cascade is a classifier (detector) trained on thousands of human faces.

This training **data is stored in an XML file**, and is later used by the classifier during detection.

It's the easiest ready to use face detection method which is supported by OpenCV/EmguCV and has great results.



.NETcampus

DEV intersection

Training your own classifiers

http://juliaimages.github.io/ImageFeatures.jl/latest/tutorials/object_detection.html

Demo – Where's Wally w/ ObjDetect Module

The detection of a cat and some cars using Latent SVM

DEV intersection

Latent SVM detector

```
#include "opencv2/core/core.hpp"
#include "opencv2/objdetect/objdetect.hpp"
#include "opencv2/highgui/highgui.hpp"
#include <iostream>

using namespace std;
using namespace cv;

int main(int argc, char* argv[]){

   String model = argv[1];
   vector<String> models;
   models.push_back( model );
   vector<String> names;
   names.push_back( "category" );
   LatentSvmDetector detector( models , names);
   if( detector.empty() ) {
      cout << "Model cannot be loaded" << endl;
      return -1;
   }
```

```
    String img = argv[2];
    Mat image = imread( img );
    if( image.empty() ){
        cout << "Image cannot be loaded" << endl;
        return -1;
    }

    vector<LatentSvmDetector::ObjectDetection> detections;
    detector.detect( image, detections, 0.1, 1);
    for( size_t i = 0; i < detections.size(); i++ ) {
        Point center( detections[i].rect.x +
                      detections[i].rect.width*0.5,
                      detections[i].rect.y +
                      detections[i].rect.height*0.5 );
        ellipse( image, center, Size(
detections[i].rect.width*0.5,
                 detections[i].rect.height*0.5), 0, 0, 360,
                 Scalar( 255, 0, 255 ), 4, 8, 0 );
    }
    imshow( "result", image );
    waitKey(0);
    return 0;
}
```

DEV intersection

In addition to the car and cat detectors, OpenCV provides pretrained detectors for the rest of the classes defined in *The PASCAL Visual Object Classes Challenge 2007* (http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007). These detectors are as follows:

- `aeroplane.xml`
- `bicycle.xml`
- `bird.xml`
- `boat.xml`
- `bottle.xml`
- `bus.xml`
- `car.xml`
- `cat.xml`
- `chair.xml`
- `cow.xml`
- `diningtable.xml`
- `dog.xml`
- `horse.xml`
- `motorbike.xml`
- `person.xml`
- `pottedplant.xml`
- `sheep.xml`
- `sofa.xml`
- `train.xml`
- `tvmonitor.xml`

Tuesday (today) 4:00-5:00 Room 116

The Vision of Computer Vision: The bold promise of teaching computers to see

Tim Huckaby

4:00pm - 5:00pm	Keeping it DRYer with Templates **Layla Porter**	Bulletproof Transient Error Handling with Polly **Carl Franklin**	Build Your First Xamarin App in 60 Minutes **Robert Green**	Docker for Web Developers **Dan Wahlin**	React for Beginners **Elijah Manor**	The Vision of Computer Vision: The Bold Promise of Teaching Computers to See **Tim Huckaby**

DEV intersection

Conclusion

DEV intersection

Source Code

https://github.com/adnanmasood/opencv-code-samples

Questions?

Please use EventsXD to fill out a session evaluation.

Thank you!

DEV intersection