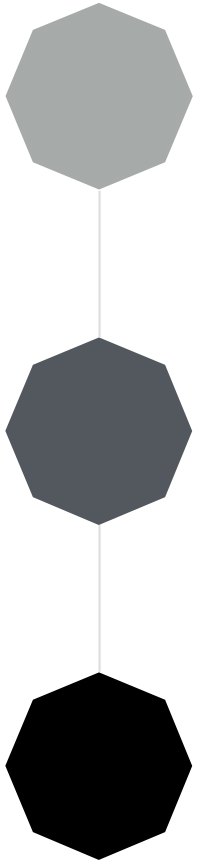
Teaching an iPhone to See

Michael Schneider
Hivebrain Software

360|iDev 2016

What Are We Going To Talk About?



Computer Vision

Object Detection

Machine Learning

The Haar wavelet's mother wavelet function $\psi(t)$ can be described as

$$\psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2}, \\ -1 & \frac{1}{2} \leq t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Its scaling function $\phi(t)$ can be described as

$$\phi(t) = \begin{cases} 1 & 0 \leq t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Who Am I

Indie Developer

Entrepreneur

Attorney

Author

≠

Computer Scientist

Some Background

Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and in general, deal with the extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions.

Some Background

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. **Machine learning** focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data.

Applications

What is ML good for?

Why?

Crazy Powerful

Challenging

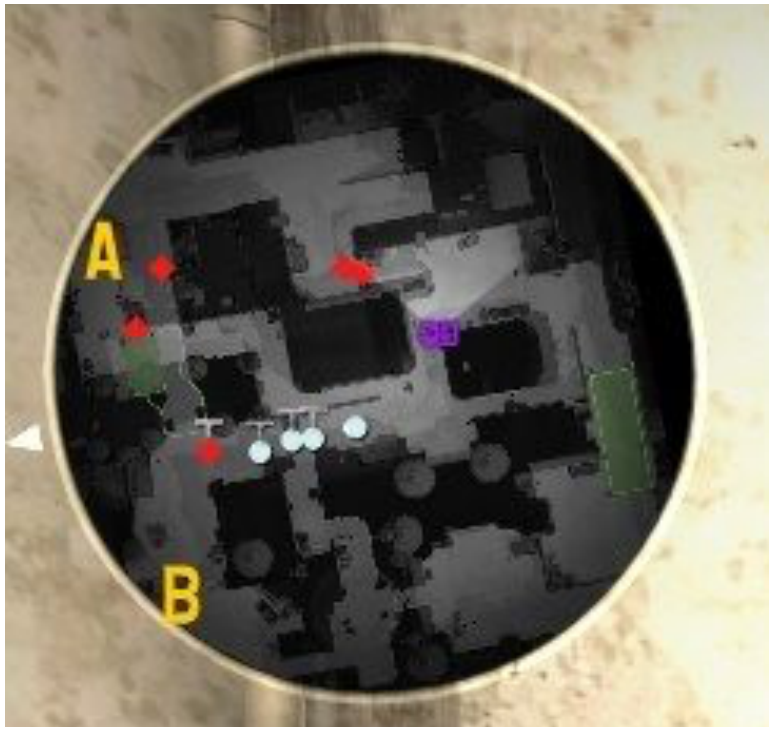
Untapped

Under Represented


Like a Golden
Hammer


HIT ALL THINGS
LIKE AN ANGRY GOD

My Nail



First Steps

Choices:

iOS SDK

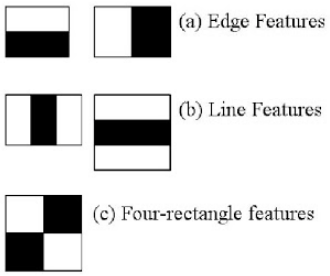
Tensor Flow

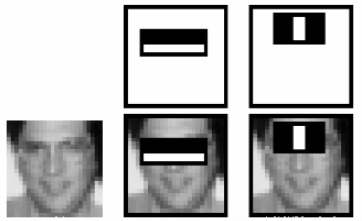
Caffe

OpenCV

OpenCV

Http://Opencv.Org





Like a series of waterfalls, the OpenCV cascade breaks the problem of detecting faces into multiple stages. For each block, it does a very rough and quick test. If that passes, it does a slightly more detailed test, and so on. The algorithm may have 30-50 of these stages or cascades, and it will only detect a face if all stages pass. The advantage is that the majority of the pictures will return negative during the first few stages, which means the algorithm won't waste time testing all 6,000 features on it. Instead of taking hours, face detection can now be done in real time.

Cascade Classifier

Code Break

https://github.com/hivebrain/CascadeClassifierDemo

Code Break

Show the App

Code Break

```
- (void)viewDidAppear:(BOOL)animated{

    [self loadClassifier];
    [self startCamera];
}
```

```
-(void)loadClassifier{

    NSString* pathToModel = [[NSBundle mainBundle] pathForResource:@"haarcascade_frontalface_default" ofType:@"xml"];
    const CFIndex CASCADE_NAME_LEN = 2048;
    char *CASCADE_NAME = (char *) malloc(CASCADE_NAME_LEN);
    CFStringGetFileSystemRepresentation( (CFStringRef)pathToModel, CASCADE_NAME, CASCADE_NAME_LEN);
    theClassifier.load(CASCADE_NAME);
    free(CASCADE_NAME);
}
```

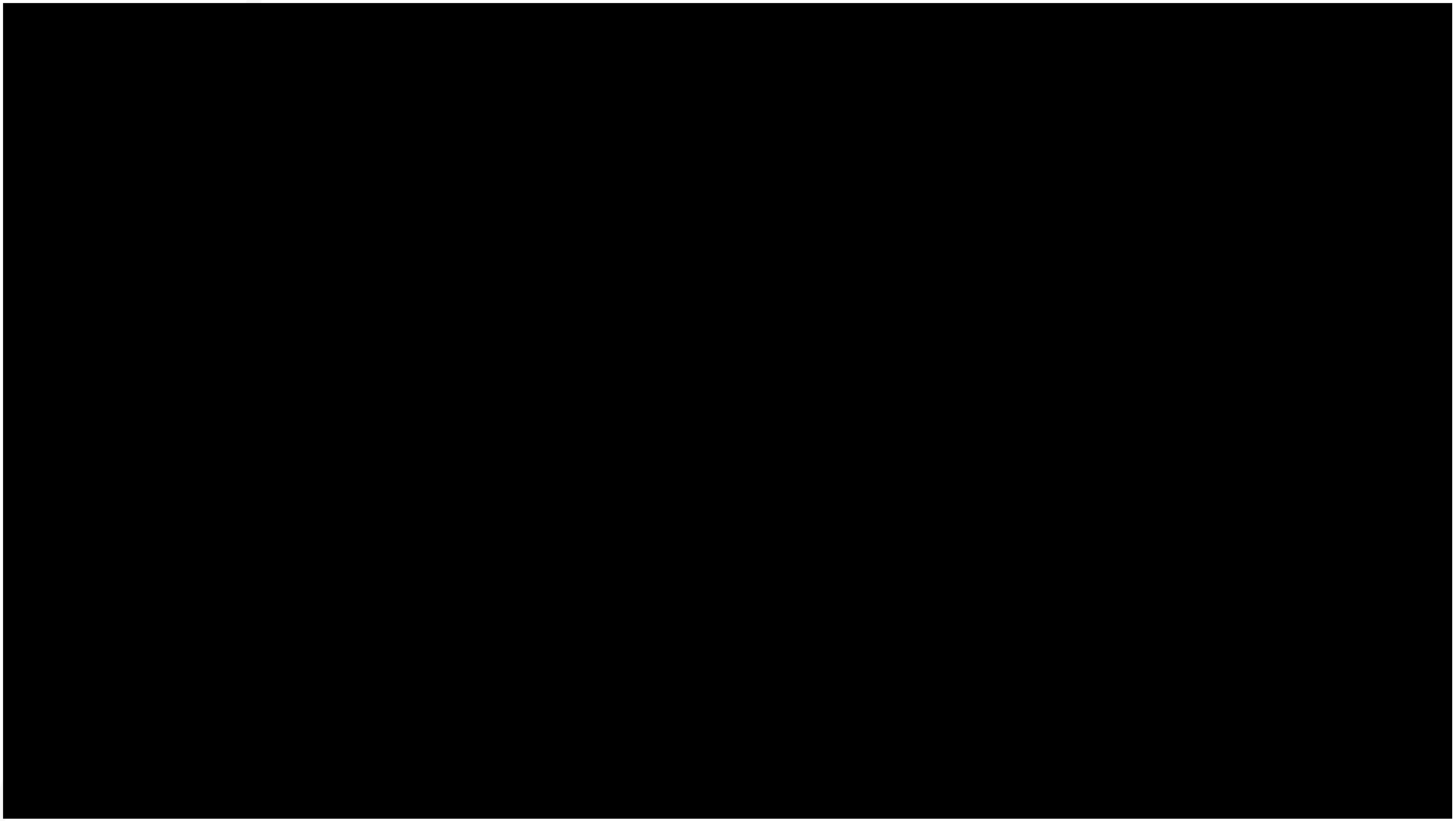
```
-(void)startCamera {

    self.videoCamera = [[CvVideoCamera alloc] initWithParentView:self.theImageView];

    self.videoCamera.defaultAVCaptureDevicePosition = AVCaptureDevicePositionFront;
    self.videoCamera.defaultAVCaptureSessionPreset = AVCaptureSessionPreset640x480;
    self.videoCamera.defaultAVCaptureVideoOrientation = AVCaptureVideoOrientationPortrait;
    self.videoCamera.defaultFPS = 30;
    self.videoCamera.grayscaleMode = NO;
    self.videoCamera.delegate = self;
    [self.videoCamera start];
}
```

Code Break

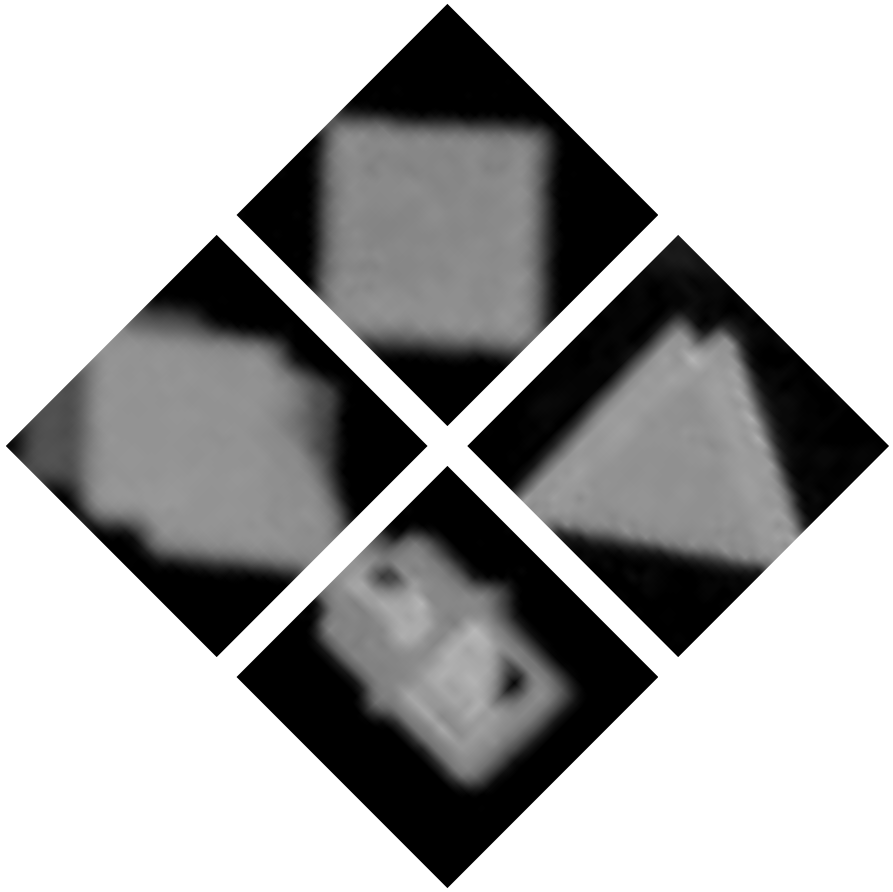
```
//This is the callback method that OpenCV calls with each frame from the camera as it captures video
- (void)processImage:(cv::Mat&)image {

    [self cascadeTest:image];
}
```

Code Break

```
// Classify.
- (void)cascadeTest:(cv::Mat&)image {

    std::vector<cv::Rect> faceRects;

    double scalingFactor = 1.1;
    int minNeighbors = 20;
    int flags = 0;
    int theMinSize = 64;
    int theMaxSize = 480;

    cv::Size minimumSize(theMinSize,theMinSize);
    cv::Size maximumSize(theMaxSize,theMaxSize);

    theClassifier.detectMultiScale(image, faceRects, scalingFactor, minNeighbors, flags, minimumSize, maximumSize );

    for( std::vector<cv::Rect>::const_iterator r = faceRects.begin(); r != faceRects.end(); r++)
    {
        //This is one of the rectangles returned as a hit by the classifier.
        cv::Rect theHit(r->x,r->y,r->width,r->height);

        bool saveHits = false;  //Set to true to capture hits as files to sort and use for samples in training.

        if (saveHits)
        {
        cv::Mat HitMat = image(theHit);
        [self writeMatToFile:HitMat withFolderName:@"theHits"];
        }

        //Draw a rectangle around the hit on the image before sending it on to be displaed by the image view.
        cv::rectangle( image, cvPoint( r->x , r->y), cvPoint( r->x + r->width, r->y + r->height), cv::Scalar(0,255,0)
            ,3);

    }

}
```



Getting Started

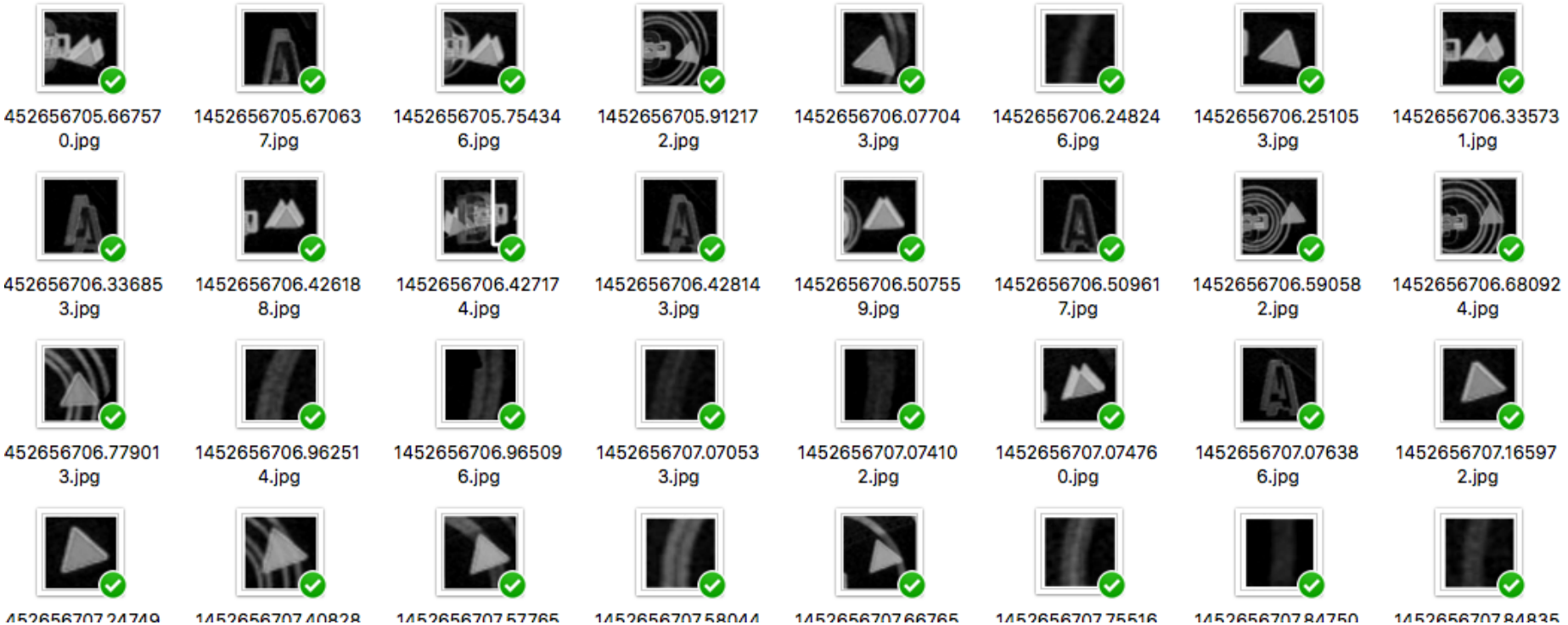


Baby Steps

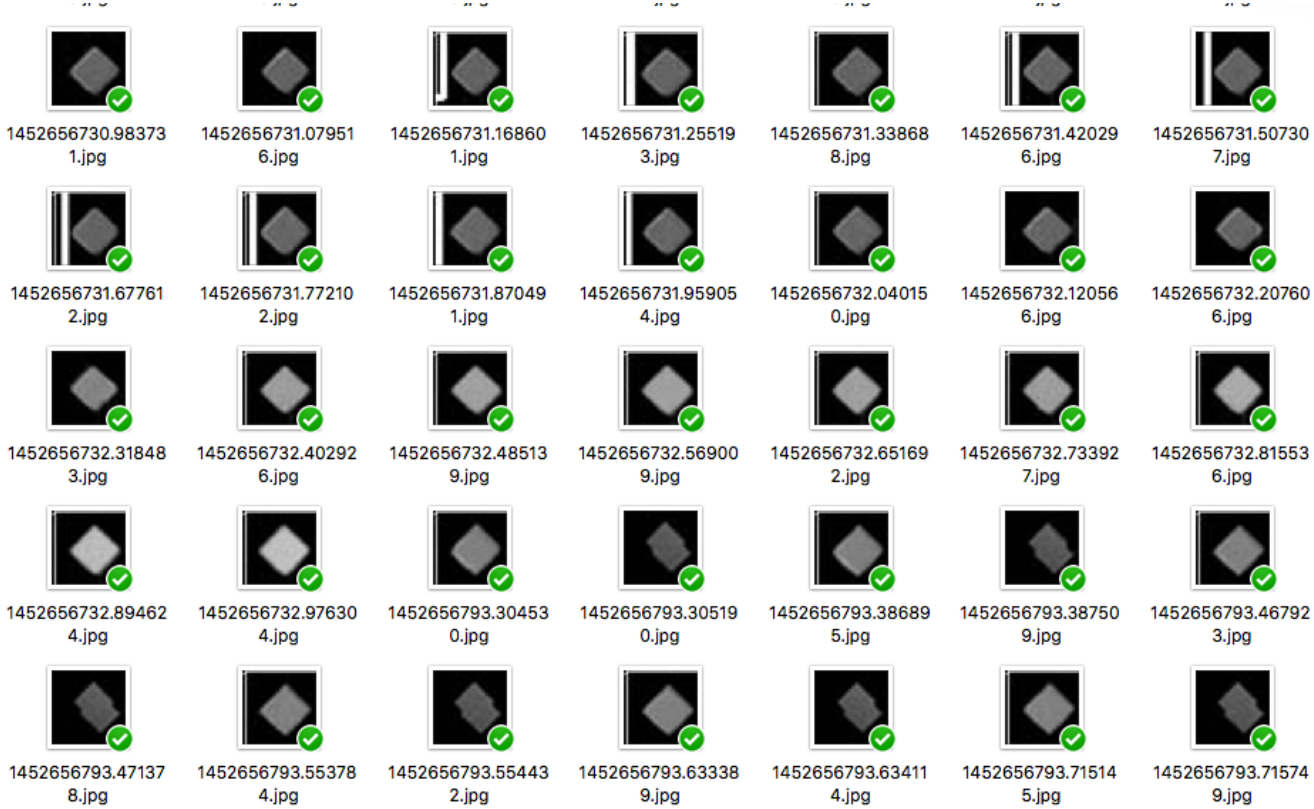
Capturing Samples

```
-(void)writeMatToFile:(cv::Mat&)image withFolderName:(NSString*)theFolderName {

    NSArray *paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES);
    NSString *docs = [paths objectAtIndex:0];
    NSString *unclassFolderPath = [docs stringByAppendingPathComponent:theFolderName];
    [[[NSFileManager alloc] init] createDirectoryAtPath:unclassFolderPath withIntermediateDirectories: YES
        attributes:nil error:nil];
    NSTimeInterval theMark = [[NSDate date] timeIntervalSince1970];
    NSString *theFileName = [NSString stringWithFormat:@"%f.jpg",theMark];
    NSString *vocabPath = [unclassFolderPath stringByAppendingPathComponent:theFileName];
    cv::String FullPath = [vocabPath UTF8String];
    cv::imwrite(FullPath, image);

}
```

Capturing Samples



Capturing Samples



Filtering Input

```
- (void)filter:(cv::Mat&)image {

    cv::Mat channel[4];

    // The actual splitting.
    split(image, channel);

    cv::Mat blue = cv::Mat::zeros(cv::Size(image.cols, image.rows), CV_8UC1);
    blue = channel[0];

    cv::Mat green = cv::Mat::zeros(cv::Size(image.cols, image.rows), CV_8UC1);
    green = channel[1];

    cv::Mat red = cv::Mat::zeros(cv::Size(image.cols, image.rows), CV_8UC1);
    red = channel[2];

    cv::Mat newThing = cv::Mat::zeros(cv::Size(image.cols, image.rows), CV_8UC1);

    newThing = (red * .8) - (blue * 0.6) - (green * 0.6);

    image = newThing;

}
```

Filtering Input

```
- (void)showEdges:(cv::Mat&)image {

    cv::Mat edge, draw;
    //cvtColor(image, gray, CV_BGR2GRAY);

    //Canny( image, edge, 50, 150, 3);

    Canny( image, edge, 200, 400, 3);

    edge.convertTo(draw, CV_8U);

    edge.copyTo(image);

}
```

Big Kid Steps

What We Learned

Second Baby Is Easier

Add Best Example

Inspiration

Take Aways

- This stuff is exciting
- There are tons of opportunities
- This stuff is accessible
- Start Looking for Nails
- Grab: https://github.com/hivebrain/CascadeClassifierDemo

Thanks

Michael Schneider,

Hivebrain Software / Bitwise Legal

Twitter: @hivebrain

Podcast: www.thelawofstartups.com