

Private

developer.apple.com

Developer

NewsDiscoverDesignDevelopDistributeSupportAccount

Documentation

Language: **Swift** API changes: None

< All Technologies

Vision

Still-image analysis

{ } Classifying images for categorization and s...

> [S] ClassifyImageRequest

> [Pr] ImageProcessingRequest

> [C] ImageRequestHandler

> [Pr] VisionRequest

> [Pr] VisionObservation

Image sequence analysis

> [C] GeneratePersonSegmentationRequest

> [S] GeneratePersonInstanceMaskRequest

> [S] DetectDocumentSegmentationRequest

> [Pr] StatefulRequest

Image aesthetics analysis

{ } Generating high-quality thumbnails from vi...

> [S] CalculateImageAestheticsScoresRequest

Saliency analysis

> [S] GenerateAttentionBasedSaliencyImageReq...

> [S] GenerateObjectnessBasedSaliencyImageR...

Object tracking

Filter

/

Vision / Original Objective-C and Swift API / Recognizing Objects in Live Capture

Sample Code

Recognizing Objects in Live Capture

Apply Vision algorithms to identify objects in real-time video.

Download

iOS 12.0+ | iPadOS 12.0+ | Xcode 11.3+


Overview

With the [Vision](#) framework, you can recognize objects in live capture. Starting in iOS 12, macOS 10.14, and tvOS 12, Vision requests made with a Core ML model return results as [VNRecognizedObjectObservation](#) objects, which identify objects found in the captured scene.

This sample app shows you how to set up your camera for live capture, incorporate a Core ML model into Vision, and parse results as classified objects.


Croissant

Confidence: 0.95



Banana

Confidence: 1.00



Link: <https://developer.apple.com/documentation/vision/recognizing-objects-in-live-capture>

BreakfastFinder

main

BreakfastFinder › João Bruno's iPhone

BreakfastFinder: Ready | Today at 16:14

+

☰

☰

☒

🔍

⚠️

⬡

🔗

📄

☰

▼ BreakfastFinder

📖 README

▼ BreakfastFinder

🔥 AppDelegate

🔥 ViewController

🔥 VisionObject...ViewController

✂ Main

📁 Assets

✂ LaunchScreen

🔗 ObjectDetector

📁 Breakfast Finder Logo

📄 Info

➤ Products

▼ Configuration

⚙ SampleCode

▼ LICENSE

📄 LICENSE

BreakfastFinder › BreakfastFinder › 🔥 VisionObjectRecognitionViewController

M setupVision()

```
6 */
7
8 import UIKit
9 import AVFoundation
10 import Vision
11
12 class VisionObjectRecognitionViewController: UIViewController {
13
14     private var detectionOverlay: CALayer! = nil
15
16     // Vision parts
17     private var requests = [VNRequest]()
18
19     @discardableResult
20     func setupVision() -> NSError? {
21         // Setup Vision parts
22         let error: NSError! = nil
23
24         guard let modelURL = Bundle.main.url(forResource: "ObjectDetector", withExtension: "mlmodelc") else {
25             return NSError(domain: "VisionObjectRecognitionViewController", code: -1, userInfo: [NSLocalizedStringDescriptionKey: "Model file is missing"])
26         }
27         do {
28             let visionModel = try VNCoreMLModel(for: MLModel(contentsOf: modelURL))
29             let objectRecognition = VNCoreMLRequest(model: visionModel, completionHandler: { (request, error) in
30                 DispatchQueue.main.async(execute: {
31                     // perform all the UI updates on the main queue
32                     if let results = request.results {
33                         self.drawVisionRequestResults(results)
34                     }
35                 })
36             })
37             self.requests = [objectRecognition]
38         } catch let error as NSError {
39             print("Model loading went wrong: \(error)")
40         }
41
42         return error
43     }
44
45     func drawVisionRequestResults(_ results: [Any]) {
46         CATransaction.begin()
47         CATransaction.setValue(kCFBooleanTrue, forKey: kCATransactionDisableActions)
48         detectionOverlay.sublayers = nil // remove all the old recognized objects
49         for observation in results where observation is VNRecognizedObjectObservation {
```

+

Filter

⌚

☰

14 characters | 📄