

What's new in CoreML

UnderDog Developer Xcoders Meetup

By: Quin'darius Lyles-Woods

Overview of the Talk

- Introduction
- Overview of CoreML
 - What is a (ML)mode and how does it work?
 - CoreML Usage
- What's new in CoreML
- Conclusion

About Quin'darius

- Programmer
- Student
- Works at Rekor
- Tinker



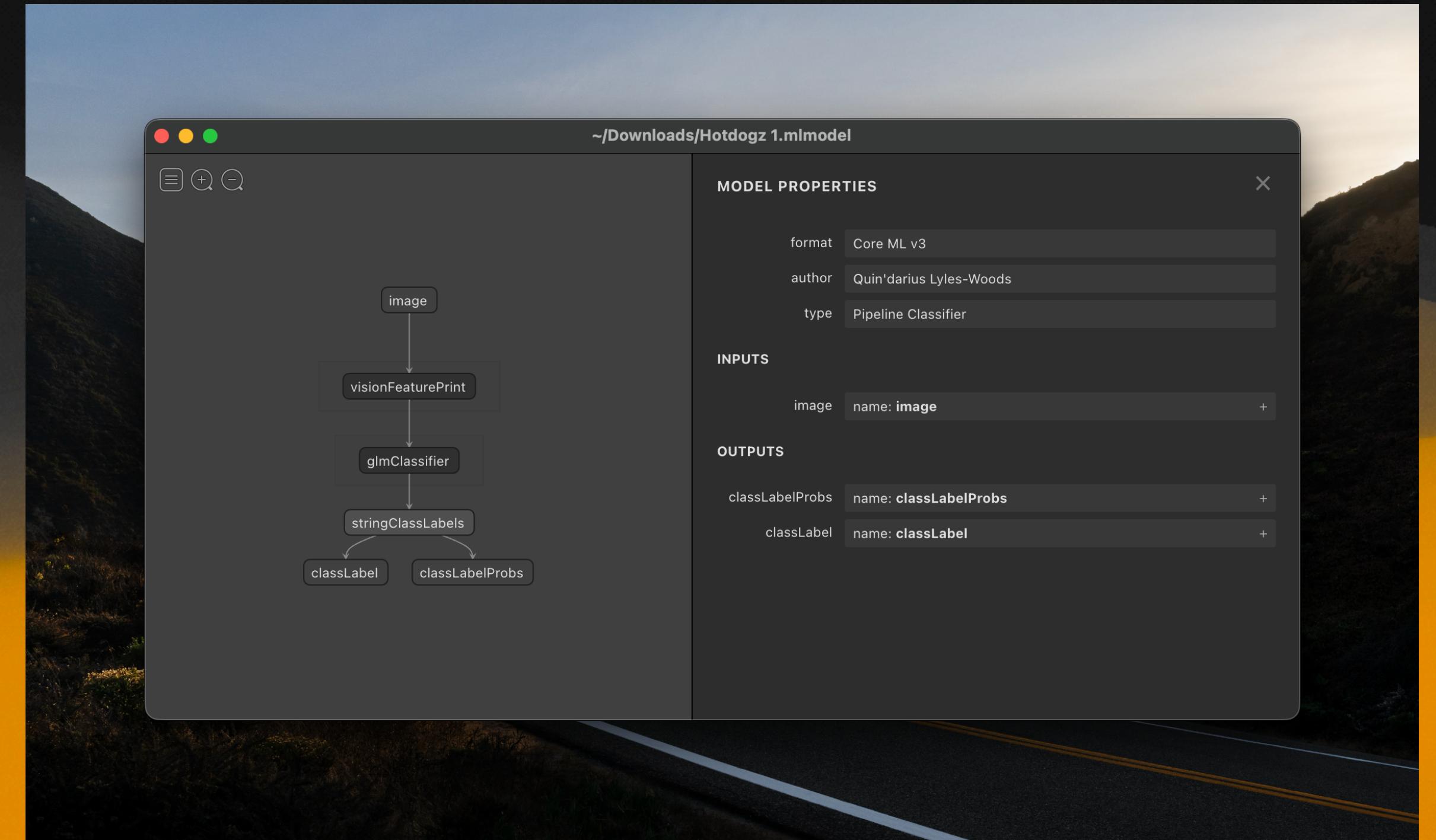
History with Machine Learning

- Building agents in the web
- Building Natural Language Models
- Now working on a Machine Vision Application

Overview of CoreML

What is a (ML)Model and how does it work?

- Framework that helps you make predictions.
- Predictions are based off of given models
- Models are Algorithms



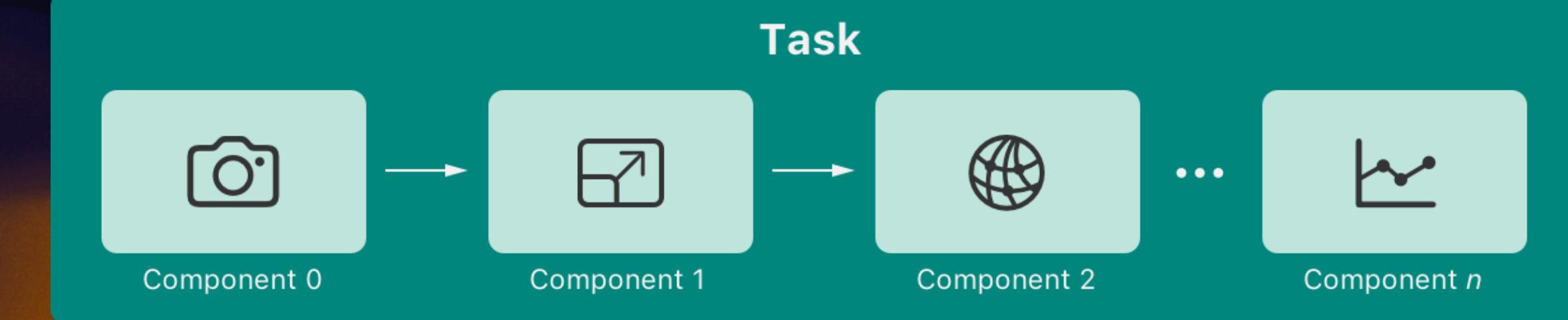
Overview of CoreML

CoreML Usage

- Gives you an API to make predictions
- Makes conforming to your models inputs easier
- Easier way of communicating to your models

What's new in CoreML

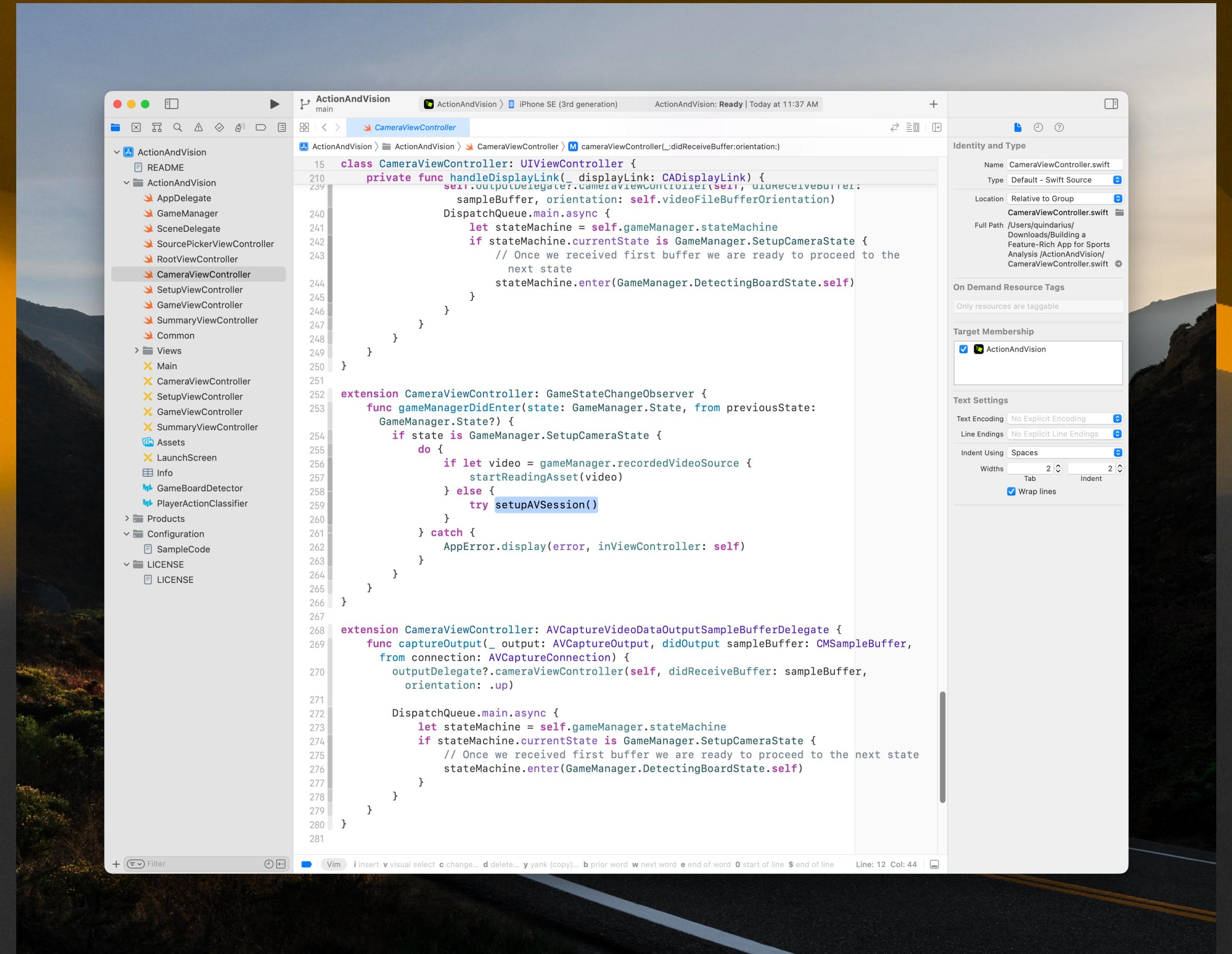
Actually not all too much.



What's new in CoreML

The old world order

- Using Vision to accomplish Machine Learning Task.
- 300 SLOC for Camera Configuration
- Easy to understand, not.



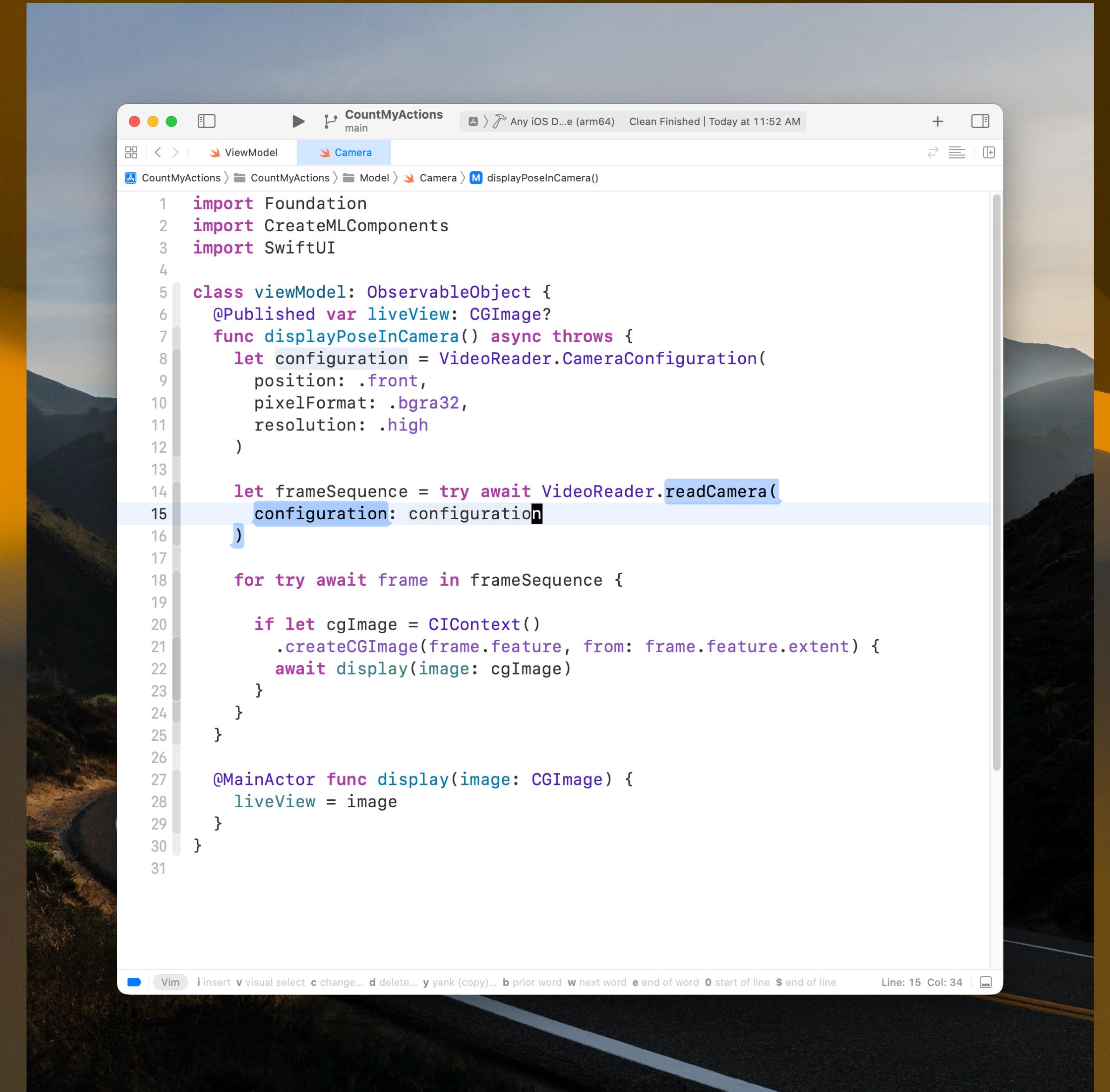
A screenshot of the Xcode IDE showing the `CameraViewController.swift` file. The code is written in Swift and handles camera configuration and state transitions. The interface includes a sidebar with project files like `README`, `ActionAndVision`, `Assets`, `Info`, `LICENSE`, and `SampleCode`. The main editor shows the following code:

```
15 class CameraViewController: UIViewController {
16     private func handleDisplayLink(_ displayLink: CADisplayLink) {
17         self.outputDelegate?.cameraViewController(self, didReceiveBuffer: sampleBuffer, orientation: self.videoFileBufferOrientation)
18         DispatchQueue.main.async {
19             let stateMachine = self.gameManager.stateMachine
20             if stateMachine.currentState is GameManager.SetupCameraState {
21                 // Once we received first buffer we are ready to proceed to the next state
22                 stateMachine.enter(GameManager.DetectingBoardState.self)
23             }
24         }
25     }
26
27     extension CameraViewController: GameStateChangeObserver {
28         func gameManagerDidEnter(state: GameManager.State, from previousState: GameManager.State?) {
29             if state is GameManager.SetupCameraState {
30                 do {
31                     if let video = gameManager.recordedVideoSource {
32                         startReadingAsset(video)
33                     } else {
34                         try setupAVSession()
35                     }
36                 } catch {
37                     AppError.display(error, inViewController: self)
38                 }
39             }
40         }
41
42         extension CameraViewController: AVCaptureVideoDataOutputSampleBufferDelegate {
43             func captureOutput(_ output: AVCaptureOutput, didOutput sampleBuffer: CMSampleBuffer,
44                               from connection: AVCaptureConnection) {
45                 outputDelegate?.cameraViewController(self, didReceiveBuffer: sampleBuffer,
46                                                   orientation: .up)
47             }
48         }
49     }
50
51     DispatchQueue.main.async {
52         let stateMachine = self.gameManager.stateMachine
53         if stateMachine.currentState is GameManager.SetupCameraState {
54             // Once we received first buffer we are ready to proceed to the next state
55             stateMachine.enter(GameManager.DetectingBoardState.self)
56         }
57     }
58 }
```

What's new in CoreML

The new world order

- VideoReader can produce an AsyncSequence of frames
- For await syntax for your pipeline
- 30 SLOC



```
CountMyActions main Any iOS Device (arm64) Clean Finished | Today at 11:52 AM + □
View Model Camera
CountMyActions CountMyActions Model Camera M displayPoseInCamera()
1 import Foundation
2 import CreateMLComponents
3 import SwiftUI
4
5 class viewModel: ObservableObject {
6     @Published var liveView: CGImage?
7     func displayPoseInCamera() async throws {
8         let configuration = VideoReader.CameraConfiguration(
9             position: .front,
10            pixelFormat: .bgra32,
11            resolution: .high
12        )
13
14        let frameSequence = try await VideoReader.readCamera(
15            configuration: configuration
16        )
17
18        for try await frame in frameSequence {
19
20            if let cgImage = CIContext()
21                .createCGImage(frame.feature, from: frame.feature.extent) {
22                await display(image: cgImage)
23            }
24        }
25    }
26
27    @MainActor func display(image: CGImage) {
28        liveView = image
29    }
30}
31
```

Vim i insert v visual select c change... d delete... y yank (copy)... b prior word w next word e end of word o start of line \$ end of line Line: 15 Col: 34

What's new in CoreML

An Overview of the Machine
Learning Ecosystem
currently at Apple.

Framework usage

Create Machine Learning Components

Accelerate using BNNS *Wizard level
100,000,000

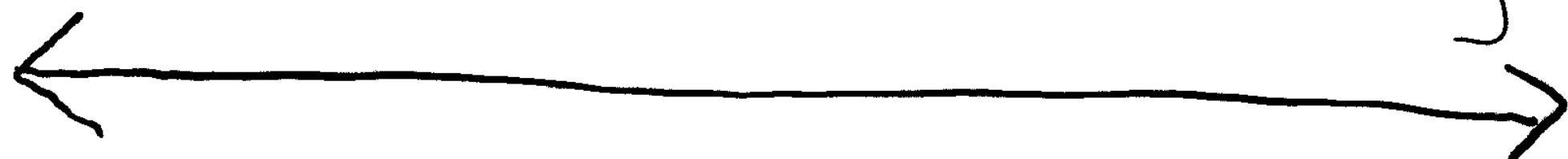
Vision

CoreML

Inference

Create ML

Training



Conclusion

Questions?

- Create Machine Learning Components Does more than just Training.
- Apple is challenging the way we think about machine learning models.
- Getting started with machine learning in an application is easier than ever.

The entire talk is available on my website
<https://quindarius.com>