

Create an App without Coding?

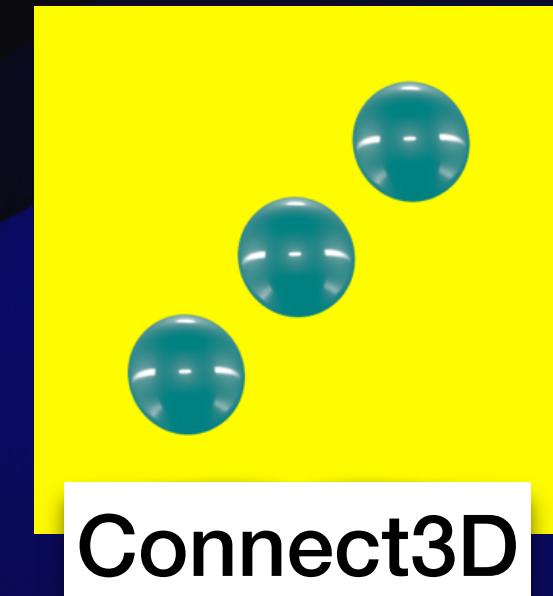
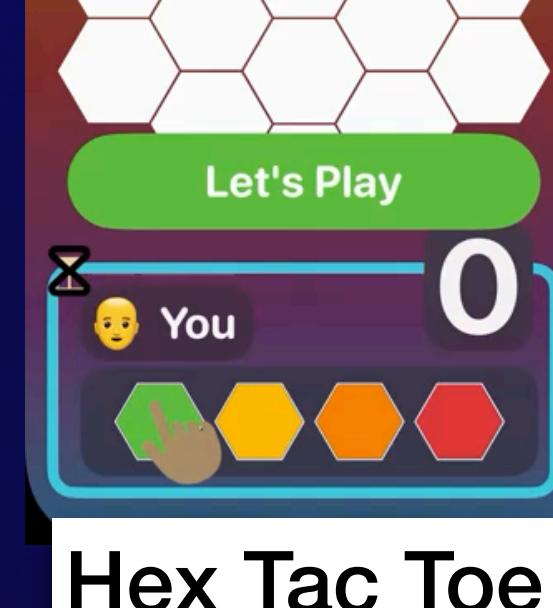
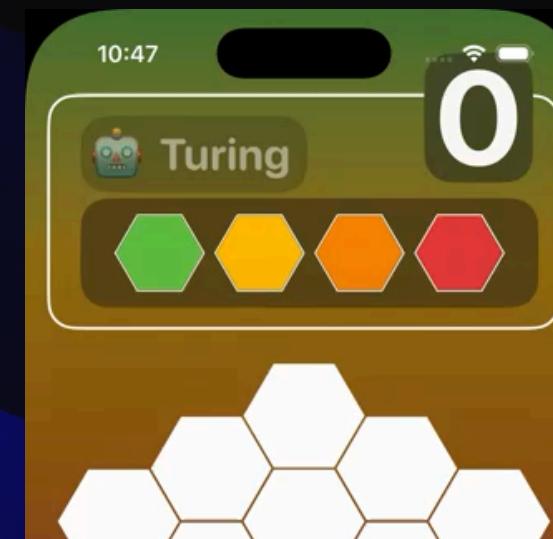
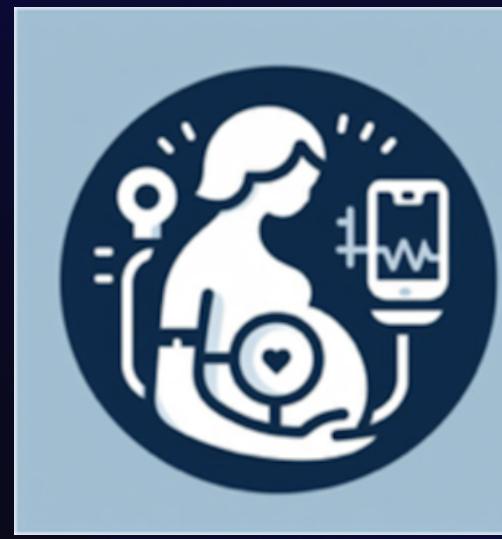
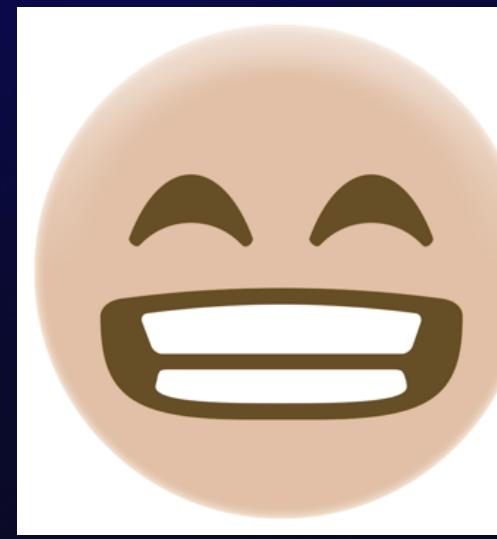
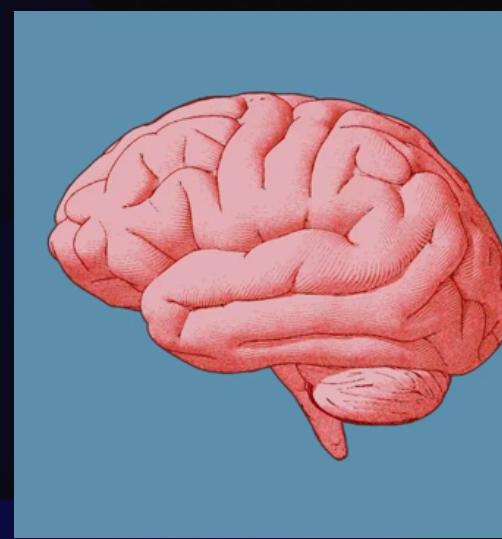
Can AI write an app for me?

Ed Arenberg

Mar 2025

Shameless Plug

Over 200 mobile apps since 2008



Contract Work

arenberg@e-page.com

Indie Apps

GitHub Copilot

- <https://github.com/features/copilot>
- Limited Free version, \$10 / mo for Pro
- Integrations - Xcode, VS Code, IntelliJ, more
- Chat interface
- Multiple models - 3.5 Sonnet, o3, 4o, etc.
- GitHub integration
- Company AI policy considerations

Xcode Integration

- <https://github.com/github/CopilotForXcode>
- 8 install steps - do them all
- Tab, option, option-tab
- Comments
- Hallucinations

ChatGPT App for Mac

Swift code to initialize an array with zeros

In Swift, you can initialize an array filled with zeros using different methods depending on the use case. Here are a few ways:

1. Using repeating and count

swift

Copy

```
let zerosArray = Array(repeating: 0, count: 10) // Creates an array of 10 zeros
print(zerosArray) // [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

2. Using map with a range

swift

Copy

```
let zerosArray = (0..<10).map { _ in 0 }
print(zerosArray) // [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

3. Using a loop (mutable array)

swift

Copy

```
var zerosArray = [Int](repeating: 0, count: 10)
print(zerosArray) // [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Would you like a function to generalize this? 🚀

VS Code Plugin

- Copy to code editor action

The screenshot shows a dark-themed interface for the GitHub Copilot VS Code plugin. At the top left is a user profile icon of a man with a purple background and the name "EPage-Ed". To the right is a close button "X". Below the profile, the text "Swift code to initialize an array with zeros" is displayed. A tooltip-like box contains the text "ⓘ README.md:105-134". On the left, there's a GitHub Copilot icon (two overlapping circles) and the text "GitHub Copilot". Underneath it, a link "Used 1 reference" is shown with a right-pointing arrow. The main content area contains the instruction "To initialize an array with zeros in Swift, you can use the following code:" followed by a code block:

```
let zerosArray = [Int](repeating: 0, count: 10)
```

A note below the code says "This creates an array of 10 integers, all initialized to zero. You can adjust the `count` parameter to the desired length of the array." At the bottom left are four small icons: a circular arrow, a thumbs up, a thumbs down, and a square with a plus sign.

Let's Built an App

- Image classifier - Pizza or Not
- CoreML Model
- SwiftUI
- Photo Library

Create ML

- Generate a CoreML .mlmodel file

Choose a Template

The screenshot shows the 'Choose a Template' section of the Create ML app. It displays a grid of 15 icons, each representing a different machine learning task. The first icon, 'Image Classification', is highlighted with a blue border. Other icons include 'Multi-Label Image Classification', 'Object Detection', 'Style Transfer', 'Hand Pose Classification', 'Action Classification', 'Hand Action Classification', 'Activity Classification', 'Sound Classification', 'Text Classification', 'Word Tagging', 'Tabular Classification', 'Tabular Regression', 'Recommendation', and 'Object Tracking'. Each icon has a small preview image and a descriptive label below it.

Data

Training Data	Validation Data	Testing Data
2 Classes 50 Items View ModellImages	Auto Split from Training Data Automatic	2 Classes 5 Items View TestImages

Parameters

Feature Extractor: Image Feature Print V1
The feature extractor model scales the input image to 299 x 299 and yields a feature embedding size of 2048.
Model Availability: macOS 10.14+ | iOS 12.0+ | tvOS 12.0+

Training Accuracy: 100.0% | Validation Accuracy: 100.0%
Iteration 12

TestImages (From Initial Setup)
3/8/2025, 10:06 PM
2 classes with 5 items

Metrics	Value	Details
Test Accuracy	100%	Top Confusion: --
Correct	5	Lowest Precision: --
Incorrect	0	Lowest Recall: --

Metrics Explore

Filter class

Class	Count	Correct	False Positives	False Negatives	Precision
Pizza	2	2	0	0	100%

Pizza or Not Pizza

- Prompt: **I want to build an iOS app using SwiftUI to select a picture from the photo library and classify it as a pizza or not a pizza. What steps do I need to take?**

To build an iOS app using SwiftUI that allows users to select a picture from the photo library and classify it as a pizza or not a pizza, you can follow these steps:

- Set Up the Project:** Create a new SwiftUI project in Xcode.
- Add Permissions:** Add the necessary permissions to access the photo library.
- Create the UI:** Build the SwiftUI interface to select and display the image.
- Implement Image Picker:** Use `UIImagePickerController` to select an image from the photo library.
- Classify the Image:** Use a machine learning model to classify the image as a pizza or not a pizza.

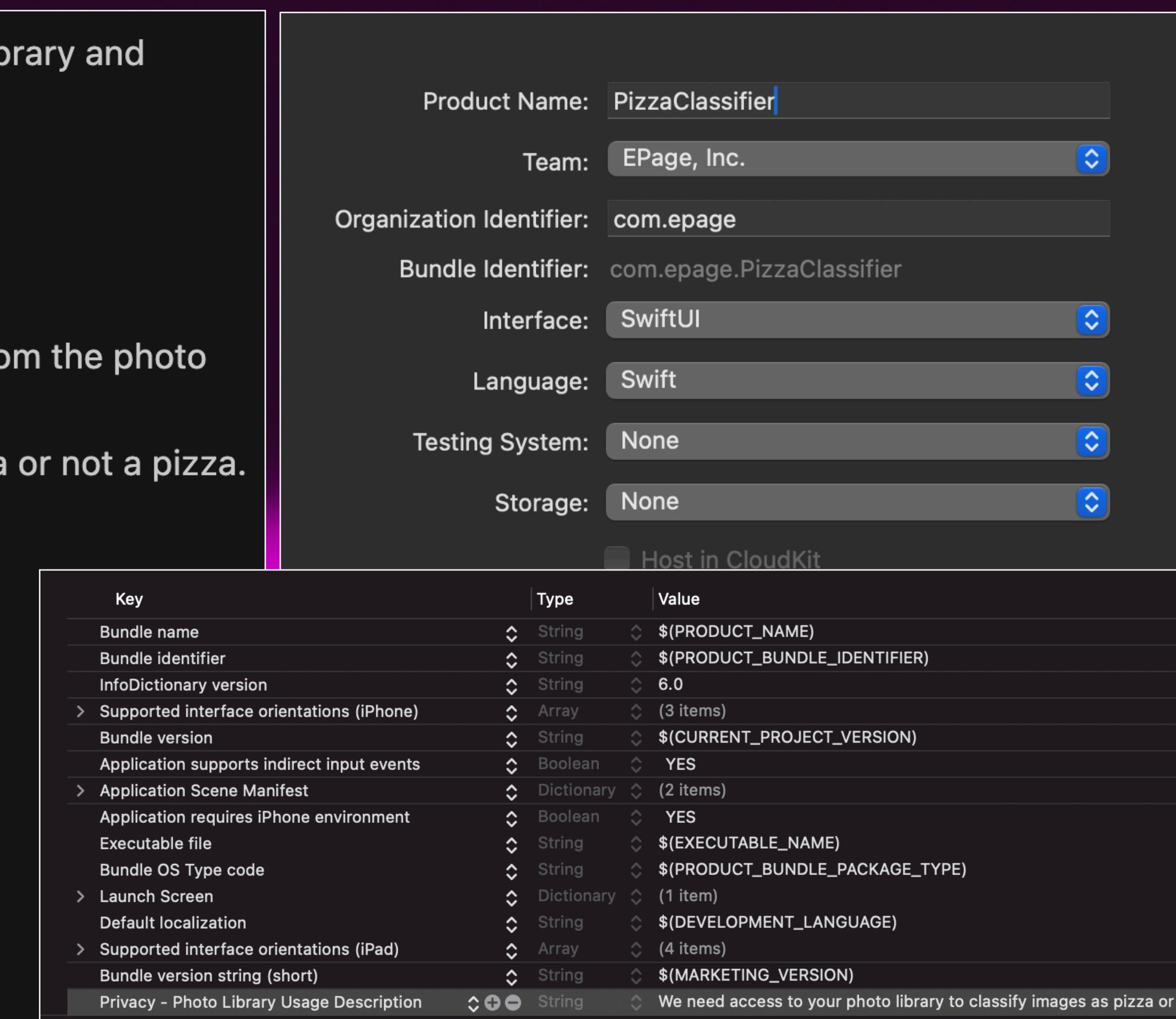
Step-by-Step Guide

Step 1: Set Up the Project

- Open Xcode and create a new SwiftUI project.
- Name the project (e.g., `PizzaClassifier`).

Step 2: Add Permissions

- Open the `Info.plist` file.
- Add a new key `Privacy – Photo Library Usage Description` with a value describing why you need access to the photo library (e.g., "We need access to your photo library to classify images as pizza or not pizza.").



Pizza or Not Pizza

Step 3: Create the UI

Create a SwiftUI view to display the selected image and a button to open the photo library.

ContentView.swift:

```
import SwiftUI
import PhotosUI

struct ContentView: View {
    @State private var selectedImage: UIImage?
    @State private var isImagePickerPresented = false
    @State private var classificationResult: String = ""

    var body: some View {
        VStack {
            if let selectedImage = selectedImage {
                Image(uiImage: selectedImage)
                    .resizable()
                    .aspectRatio(contentMode: .fit)
                    .frame(width: 300, height: 300)
                    .padding()
            } else {
                Rectangle()
                    .fill(Color.gray.opacity(0.5))
                    .frame(width: 300, height: 300)
                    .padding()
            }
        }
    }

    var body: some View {
        Button(action: {
            isImagePickerPresented = true
        }) {
            Text("Select Image")
                .padding()
                .background(Color.blue)
                .foregroundColor(.white)
                .cornerRadius(10)
        }
        .padding()

        Text(classificationResult)
            .font(.title)
            .padding()
        }
        .sheet(isPresented: $isImagePickerPresented) {
            ImagePicker(selectedImage: $selectedImage, classificationResult: $classificationResult)
        }
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

Pizza or Not Pizza

Step 4: Implement Image Picker

Create a SwiftUI wrapper for `UIImagePickerController` to select an image from the photo library.

ImagePicker.swift:

```
import SwiftUI
import UIKit
import CoreML
import Vision

struct ImagePicker: UIViewControllerRepresentable {
    @Binding var selectedImage: UIImage?
    @Binding var classificationResult: String

    class Coordinator: NSObject, UINavigationControllerDelegate, UIImagePickerControllerDelegate {
        var parent: ImagePicker

        init(parent: ImagePicker) {
            self.parent = parent
        }

        func imagePickerController(_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey : Any]) {
            if let uiImage = info[.originalImage] as? UIImage {
                parent.selectedImage = uiImage
                parent.classifyImage(uiImage)
            }
            picker.dismiss(animated: true)
        }
    }
}
```

```
func makeCoordinator() -> Coordinator {
    Coordinator(parent: self)
}

func makeUIViewController(context: Context) -> UIImagePickerController {
    let picker = UIImagePickerController()
    picker.delegate = context.coordinator
    return picker
}

func updateUIViewController(_ uiViewController: UIImagePickerController, context: Context) {
}

func classifyImage(_ image: UIImage) {
    guard let model = try? VNCoreMLModel(for: PizzaClassifier().model) else {
        classificationResult = "Failed to load model"
        return
    }

    let request = VNCoreMLRequest(model: model) { request, error in
        guard let results = request.results as? [VNClassificationObservation],
              let firstResult = results.first else {
            classificationResult = "Failed to classify image"
            return
        }

        if firstResult.identifier == "pizza" {
            classificationResult = "It's a pizza!"
        } else {
            classificationResult = "Not a pizza"
        }
    }

    guard let ciImage = CIImage(image: image) else {
        classificationResult = "Failed to convert image"
        return
    }

    let handler = VNImageRequestHandler(ciImage: ciImage)
    DispatchQueue.global(qos: .userInitiated).async {
        do {
            try handler.perform([request])
        } catch {
            classificationResult = "Failed to perform classification"
        }
    }
}
```

Pizza or Not Pizza

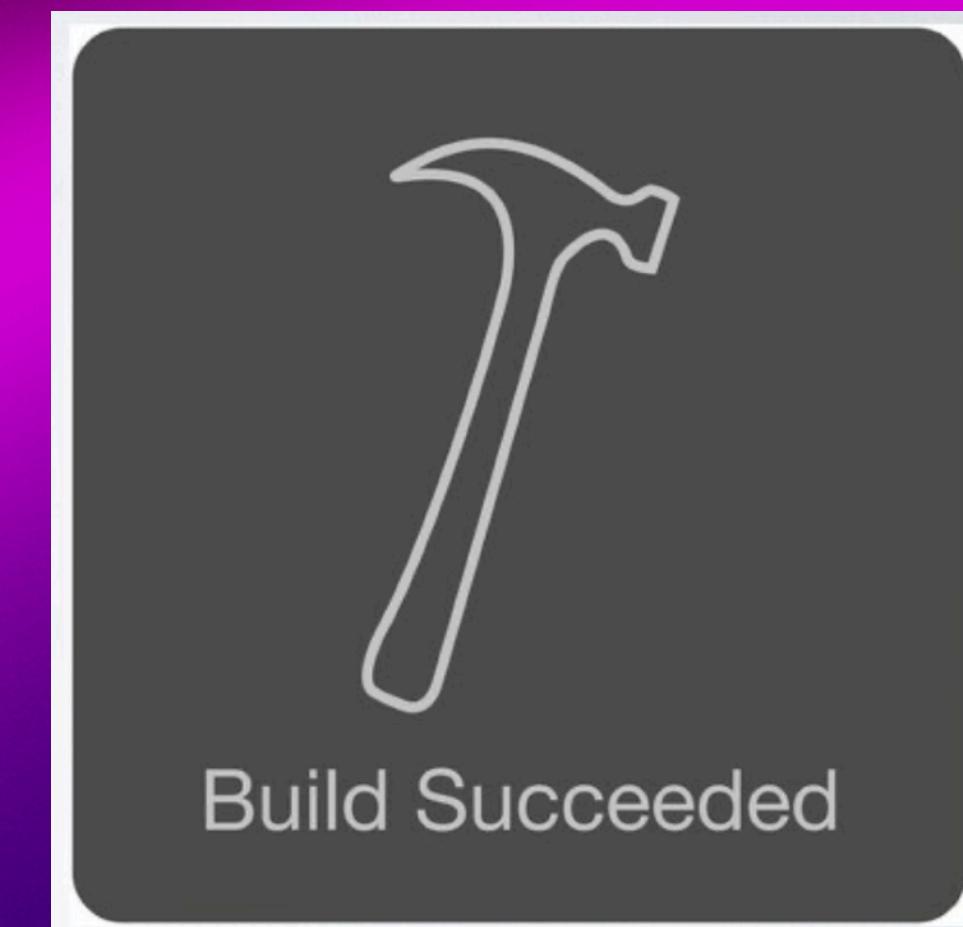
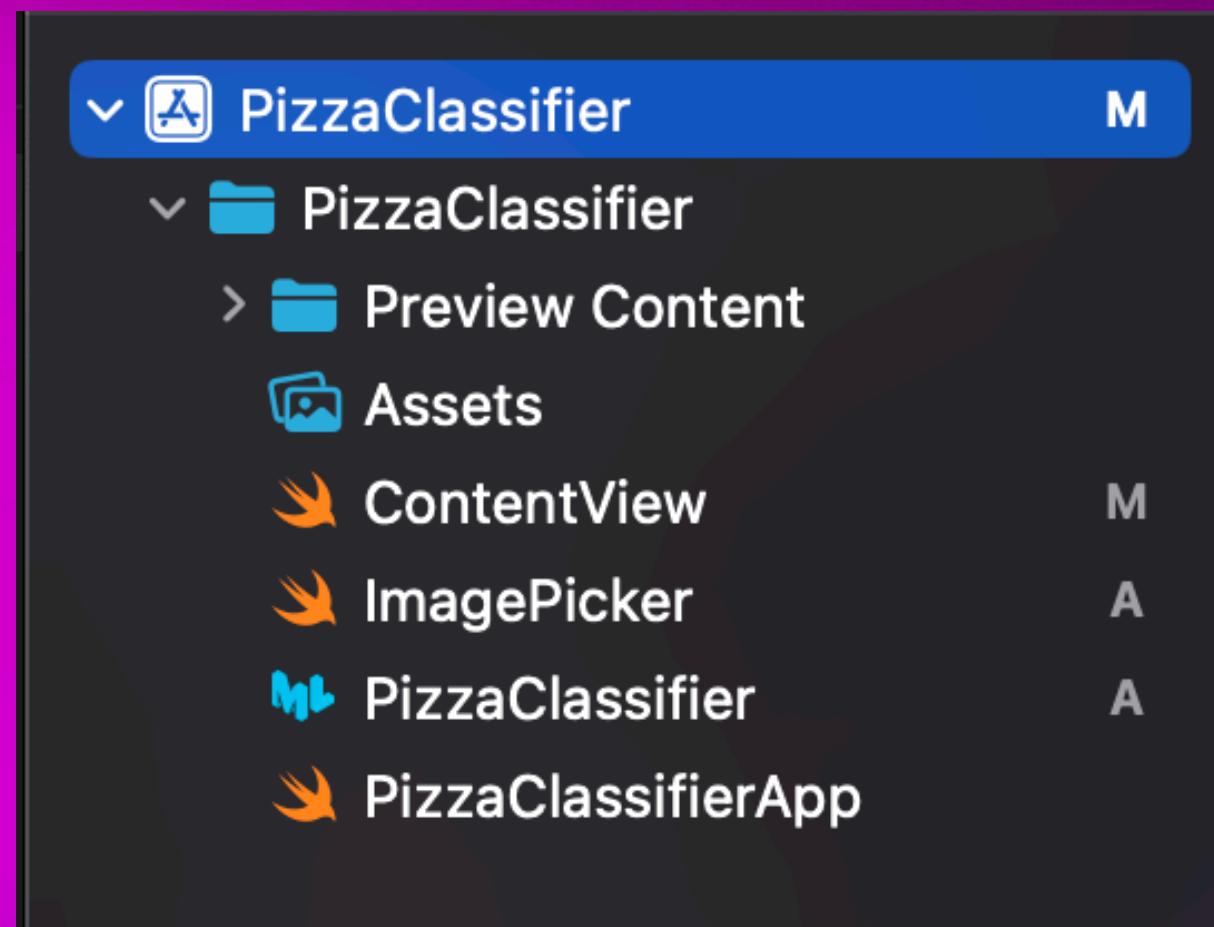
Step 5: Classify the Image

Use a Core ML model to classify the image. You can create a custom model using Create ML or use a pre-trained model. For this example, let's assume you have a Core ML model named `PizzaClassifier`.

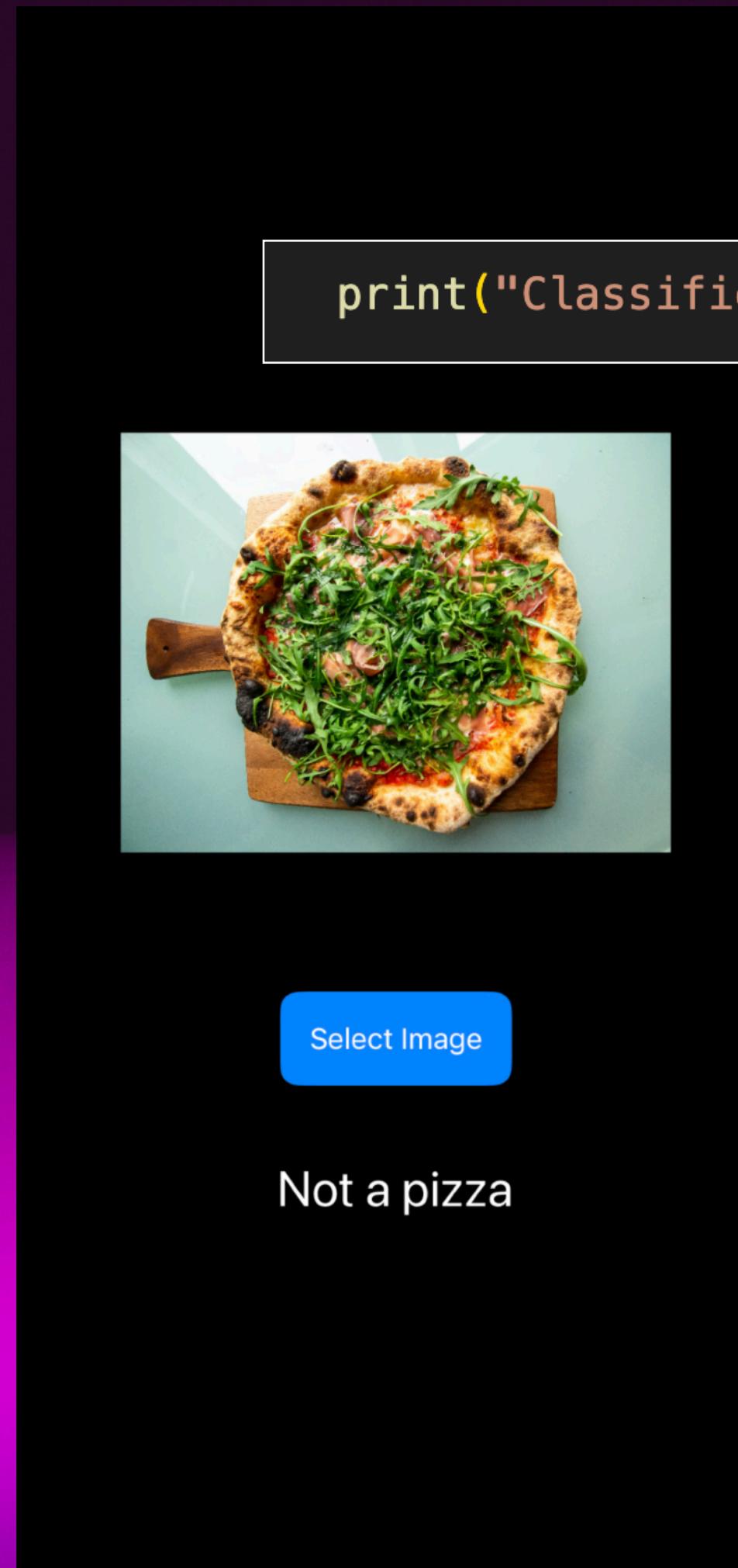
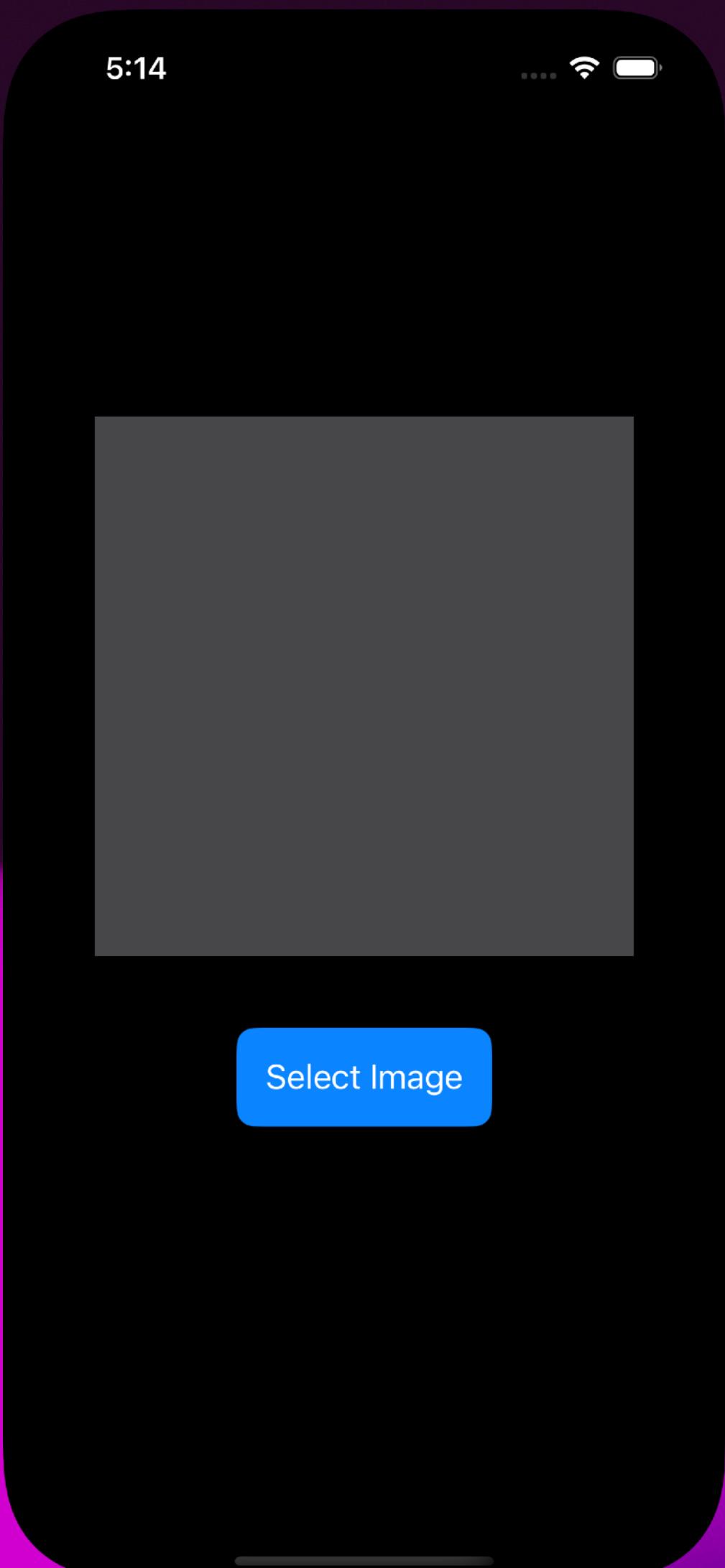
- 1. Add the Core ML model:** Drag and drop the `PizzaClassifier.mlmodel` file into your Xcode project.
- 2. Generate the model class:** Xcode will automatically generate a class for the model.

Conclusion

By following these steps, you can create an iOS app using SwiftUI that allows users to select a picture from the photo library and classify it as a pizza or not a pizza. This example uses Core ML and Vision frameworks to perform image classification. Make sure to test the app thoroughly and adjust the model and UI as needed.



Pizza or Not Pizza



Updated Code with Debugging

Let's add more debug statements to understand where the problem might be.

```
print("Classification result: \(firstResult.identifier) with confidence \(firstResult.confidence)")
```

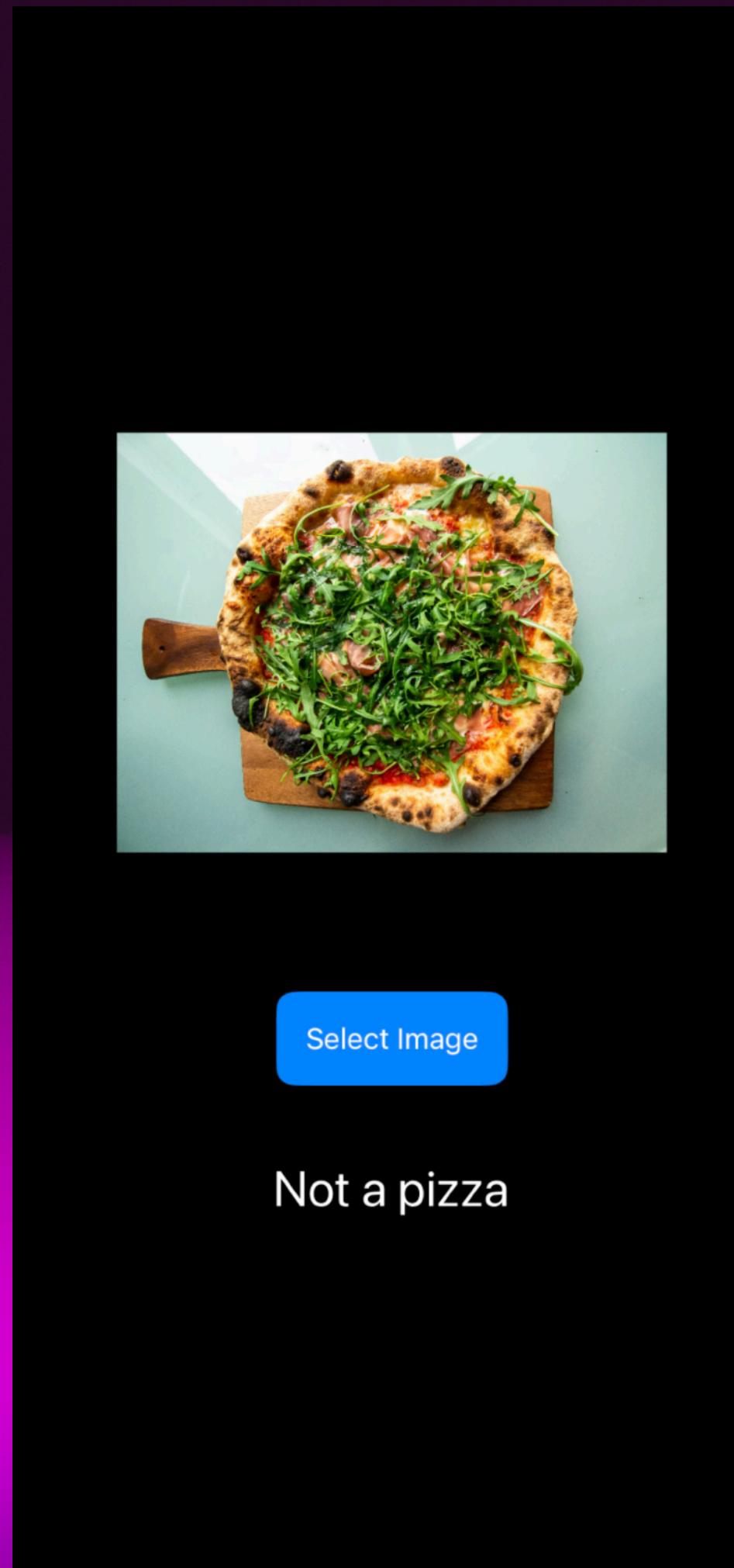
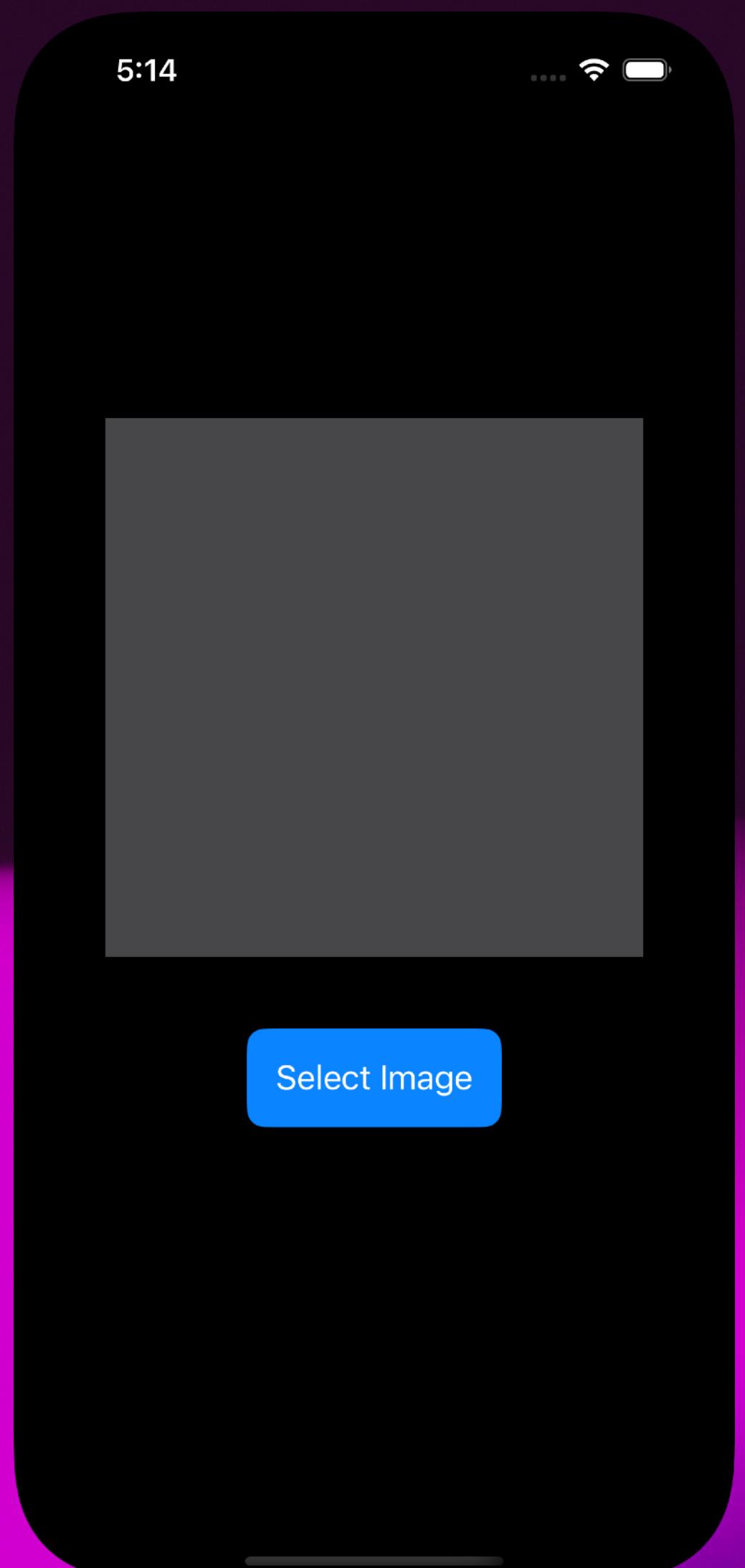


Pizza

≠

```
if firstResult.identifier == "pizza" {  
    classificationResult = "It's a pizza!"  
} else {  
    classificationResult = "Not a pizza"  
}
```

Pizza or Not Pizza



EPage-Ed

Maybe the result identifier comparison should be lowercased?

ⓘ README.md:105-125

GitHub Copilot

Used 1 reference

Yes, making the result identifier comparison case-insensitive can help avoid issues where the identifier might have different cases. Let's update the code to ensure the comparison is case-insensitive.

Explanation

- 1. Lowercase Comparison:** The identifier comparison is now case-insensitive by converting `firstResult.identifier` to lowercase using `lowercased()` before comparing it to `"pizza"`.

```
if firstResult.identifier.lowercased() == "pizza" {
    classificationResult = "It's a pizza!"
} else {
    classificationResult = "Not a pizza"
}
```

Pizza or Not Pizza



Select Image

It's a pizza!



Select Image

Not a pizza



Select Image

Not a pizza



Select Image

Not a pizza



Select Image

It's a pizza!



Create an App without Coding?

Can AI write an app for me?

<https://github.com/EPage-Ed/PizzaClassifier>

arenberg@epage.com

Ed Arenberg

Mar 2025