



Computer Vision

Not Just For Breakfast Anymore

Yono Mittlefehldt – @yonomitt – 2021

Who is Yono?

Co-creator of [Gus on the Go](#)

Who is Yono?

Co-creator of [Gus on the Go](#)

Freelance software developer
specializing in Computer Vision and iOS



Who is Yono?

Co-creator of [Gus on the Go](#)

Freelance software developer
specializing in Computer Vision and iOS

Tutorial writer for [raywenderlich.com](#)



Who is Yono?

Co-creator of [Gus on the Go](#)

Freelance software developer
specializing in Computer Vision and iOS

Tutorial writer for [raywenderlich.com](#)

One-time chauffeur to Bruce Campbell



Will this be on the test?

Not at all

The Kids in the Hall

Canada's Monty Python





What does this have to do with Computer Vision?

— You, right about now

Crushing Your Head

The App

Crushing Your Head

The App*

* The kids love this one

Crushing Your Head

The App

Crushing Your Head

The App

Hand Pose Detector

Crushing Your Head

The App

Hand Pose Detector

Face Detector

Crushing Your Head

The App

Hand Pose Detector

Face Detector

Comparing Feature Vectors

Crushing Your Head

The App

Hand Pose Detector

Face Detector

Comparing Feature Vectors

Object Tracking

Crushing Your Head

The App

Hand Pose Detector

Face Detector

Comparing Feature Vectors

Object Tracking

BONUS:

Custom Classifier

Apple's Vision Framework

Vision Framework

Vision Framework

Common computer vision tasks

Vision Framework

Common computer vision tasks

Optimized to run on iPhones and iPads

Vision Framework

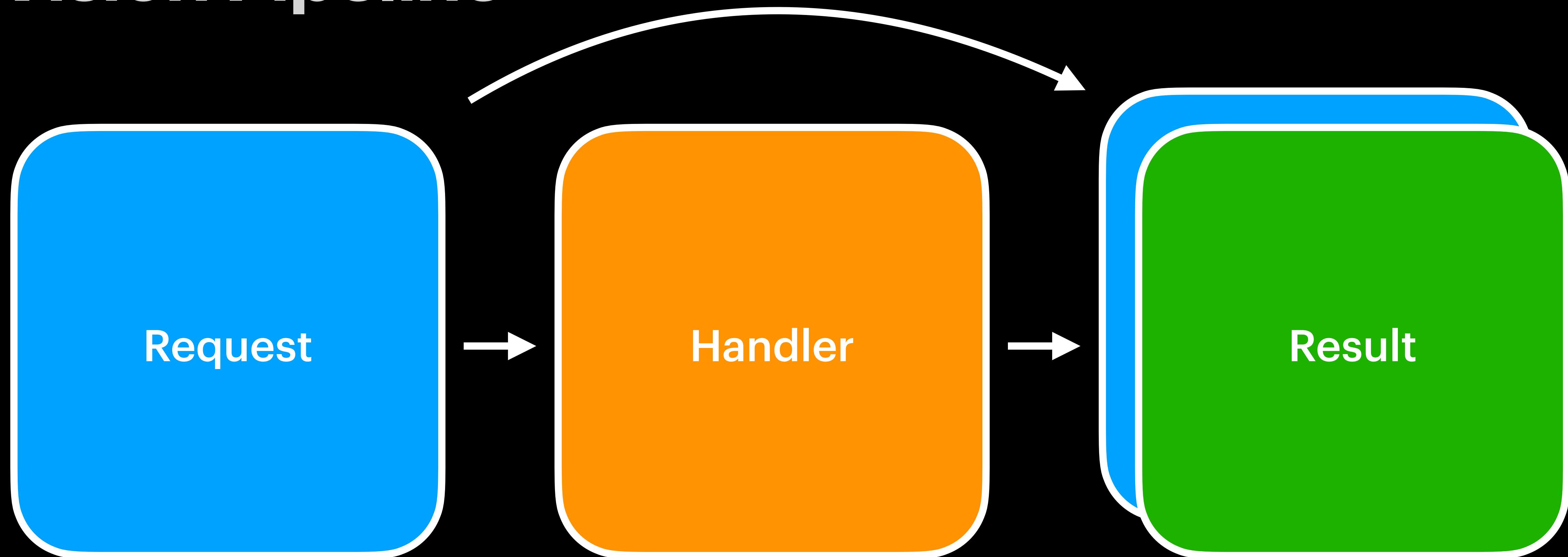
Common computer vision tasks

Optimized to run on iPhones and iPads

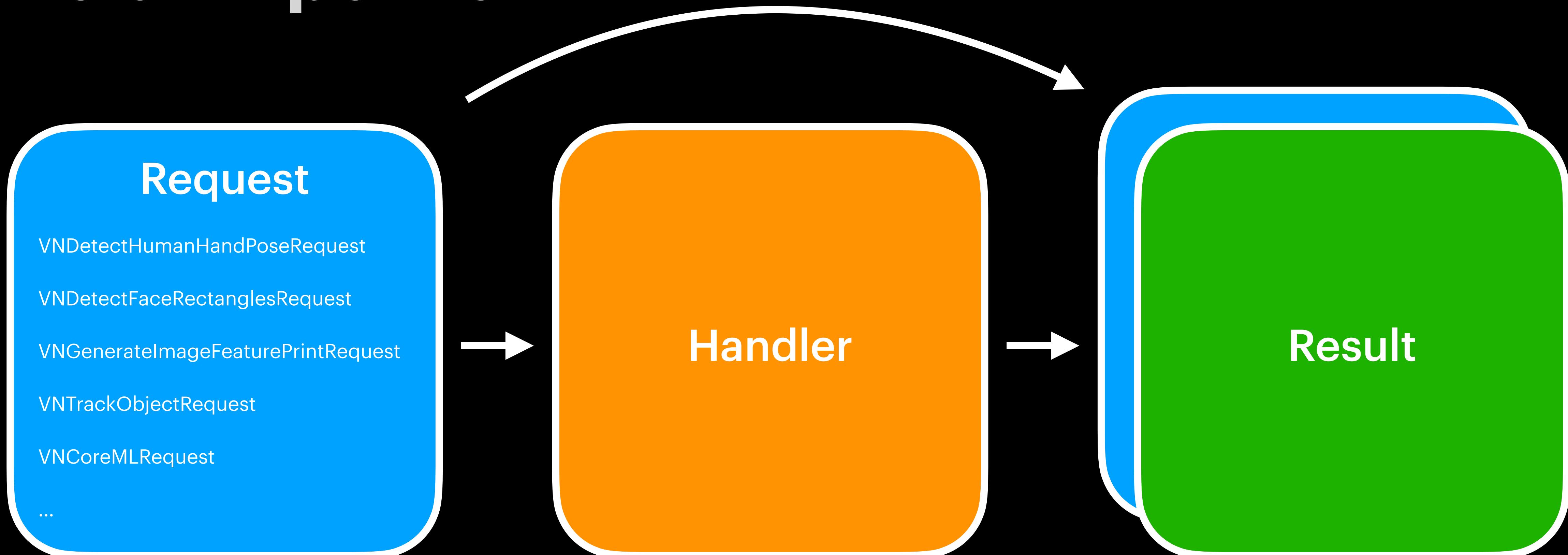
Updated yearly

Vision Pipeline

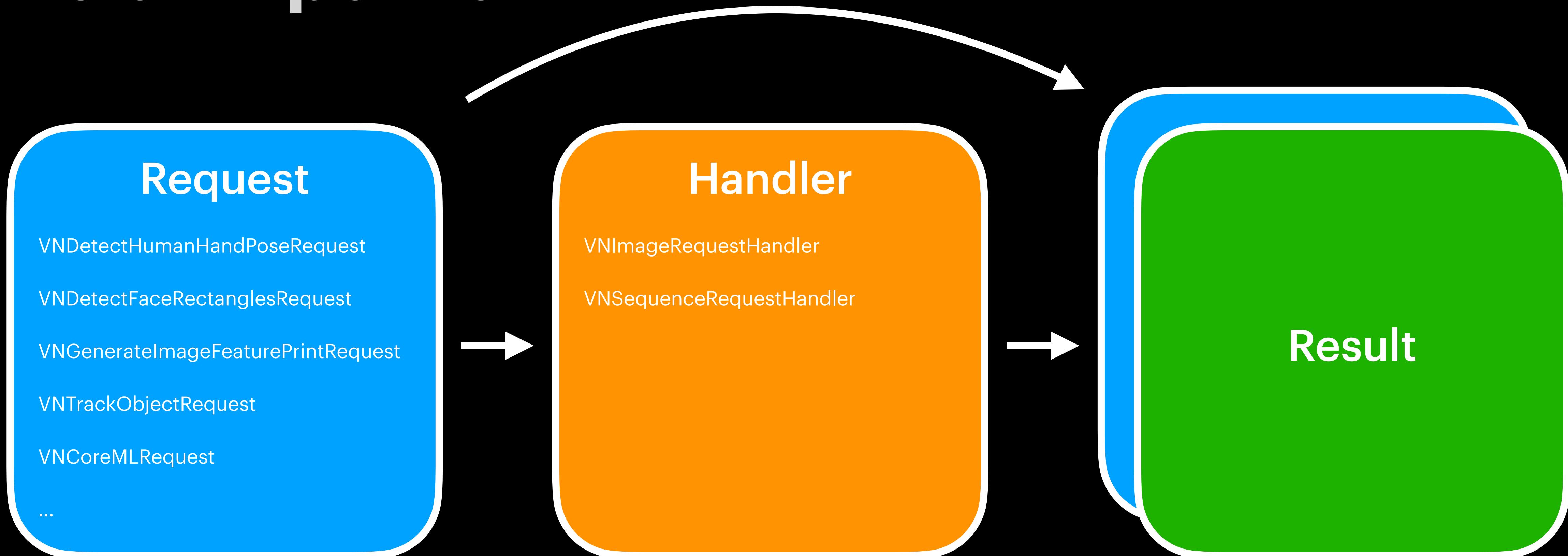
Vision Pipeline



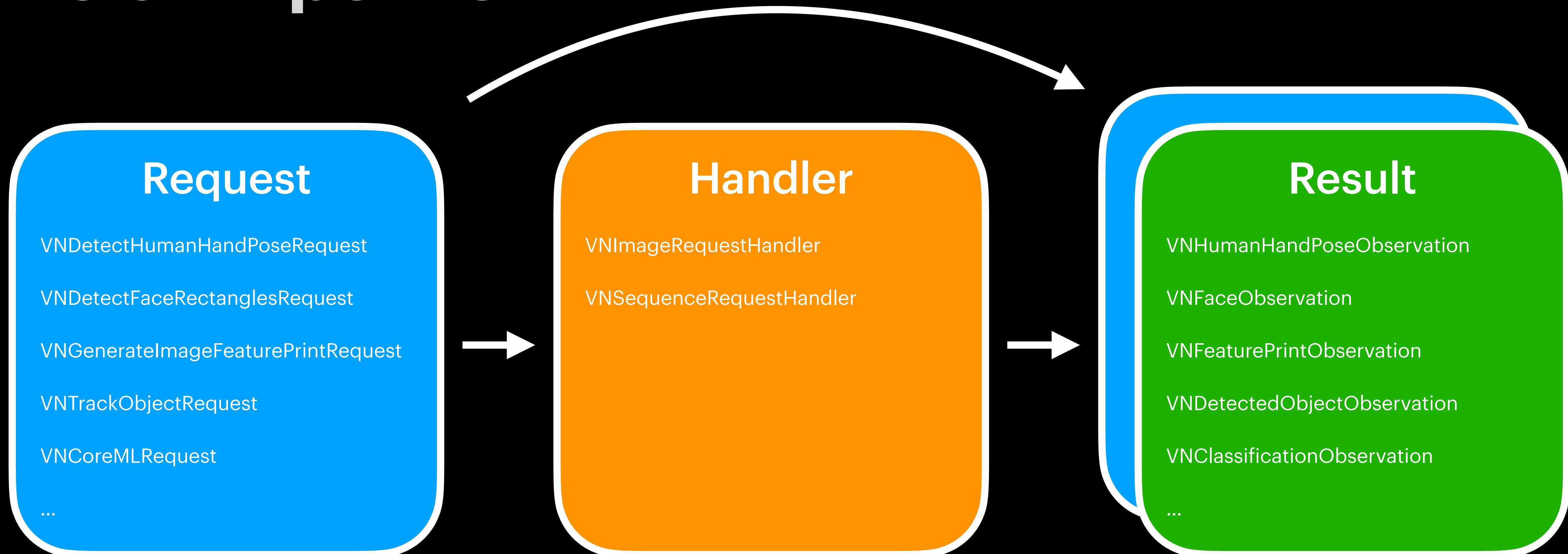
Vision Pipeline



Vision Pipeline



Vision Pipeline



Vision Tutorials

Vision Tutorials

Photo Stacking

www.raywenderlich.com/3733151-photo-stacking-in-ios-with-vision-and-metal



Vision Tutorials

Photo Stacking

Saliency Analysis

www.raywenderlich.com/5807038-saliency-analysis-in-ios-using-vision



Vision Tutorials

Photo Stacking

Saliency Analysis

Face Landmark Detection

www.raywenderlich.com/1163620-face-detection-tutorial-using-the-vision-framework-for-ios



Crushing Your Head

The App

Hand Pose Detector

Face Detector

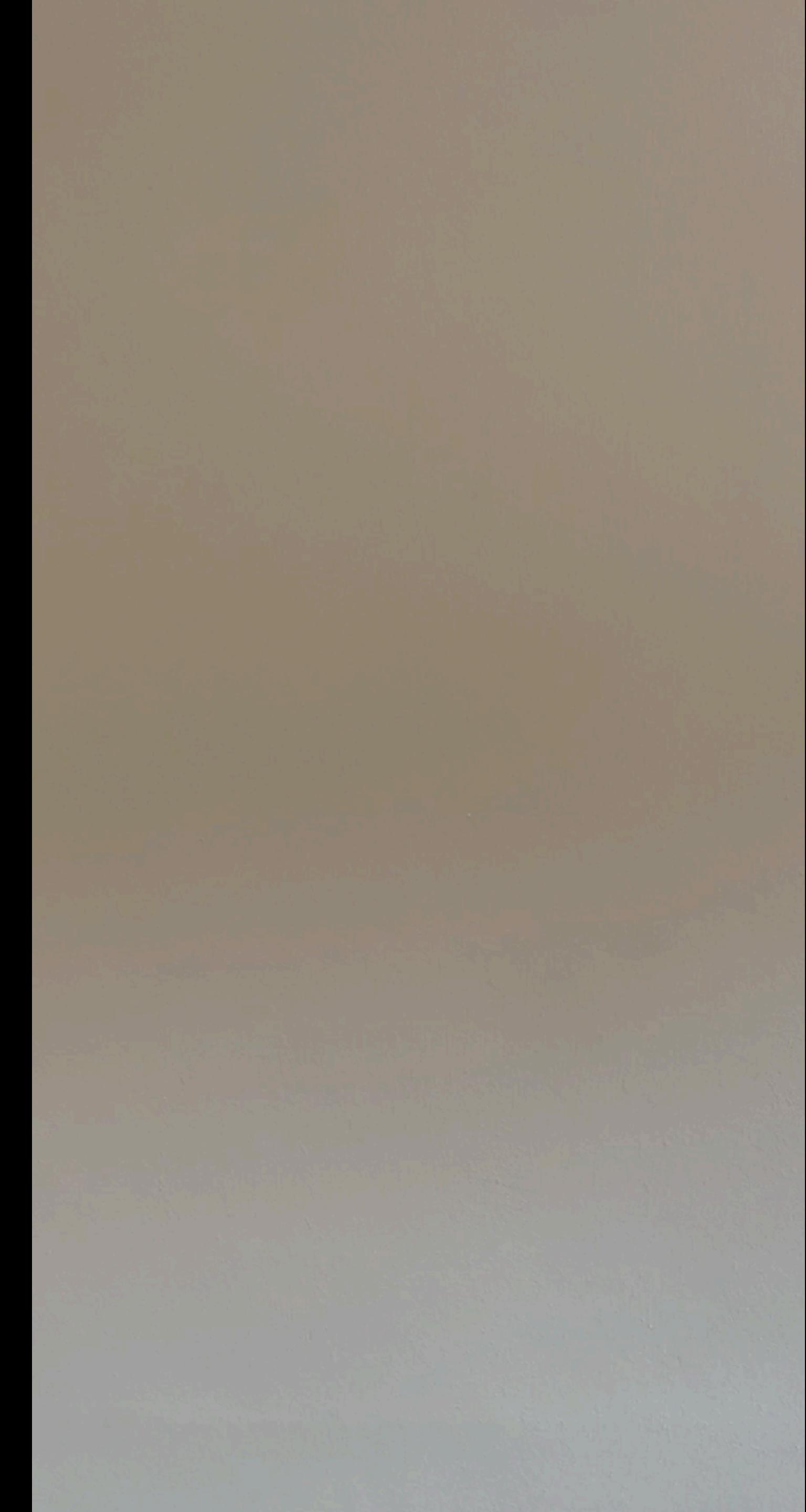
Comparing Feature Vectors

Object Tracking

BONUS:

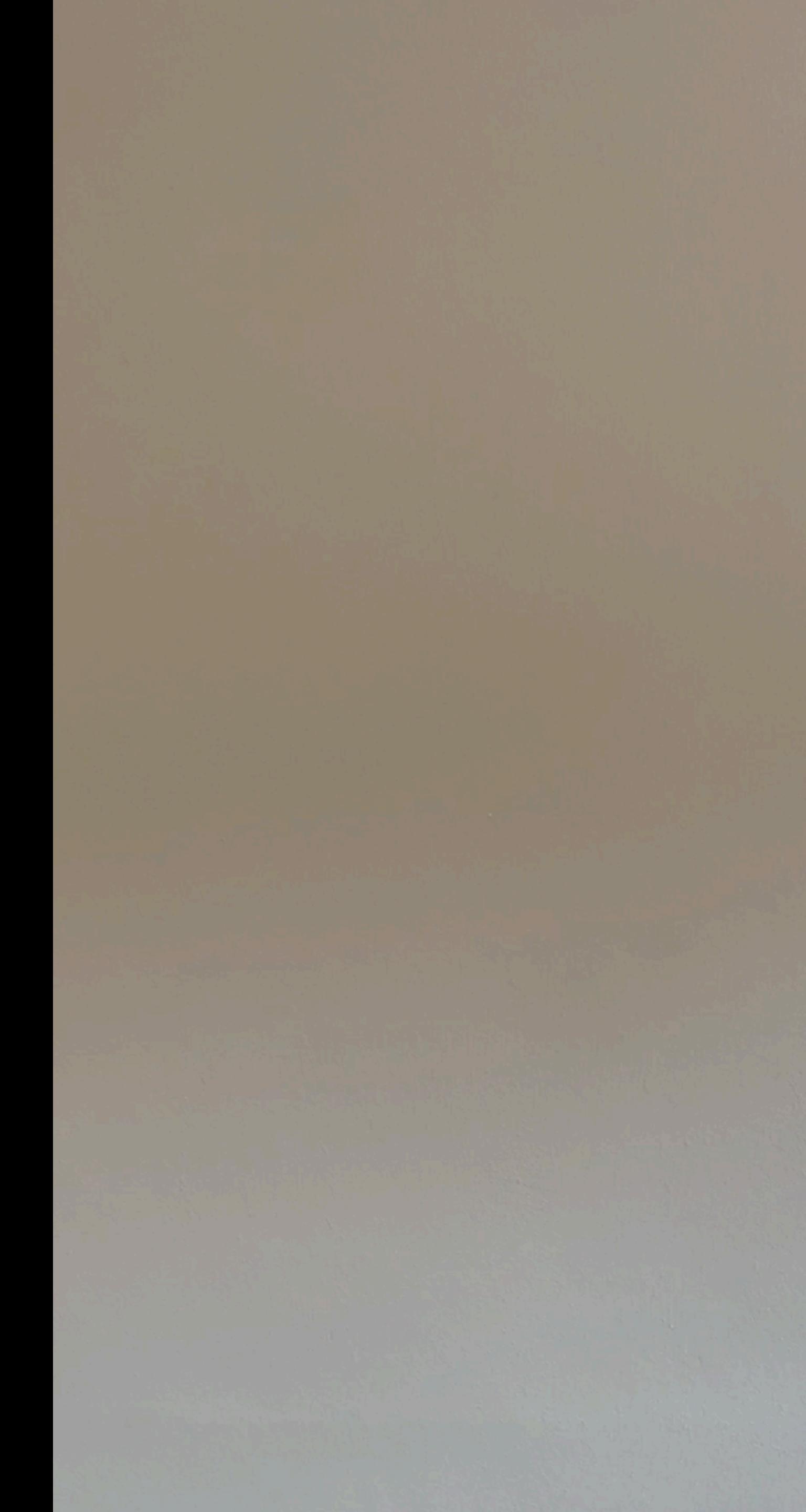
Custom Classifier

Hand Pose in the App



Hand Pose in the App

Unique game mechanic



Hand Pose in the App

Unique game mechanic

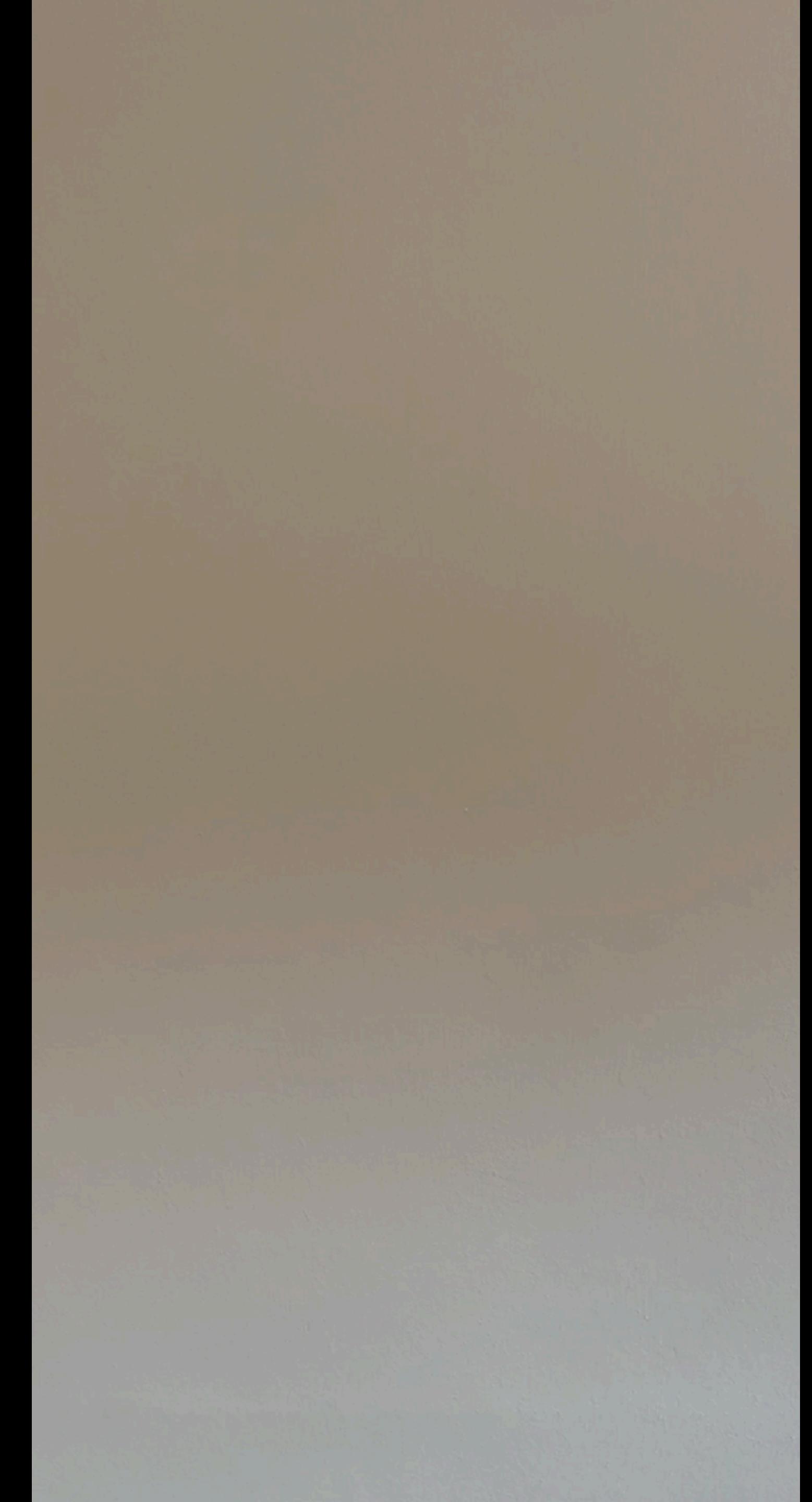
Simplify by just looking at the tip
of the thumb and index finger

Hand Pose in the App

Unique game mechanic

Simplify by just looking at the tip
of the thumb and index finger

If the tips are close together, the
user is pinching

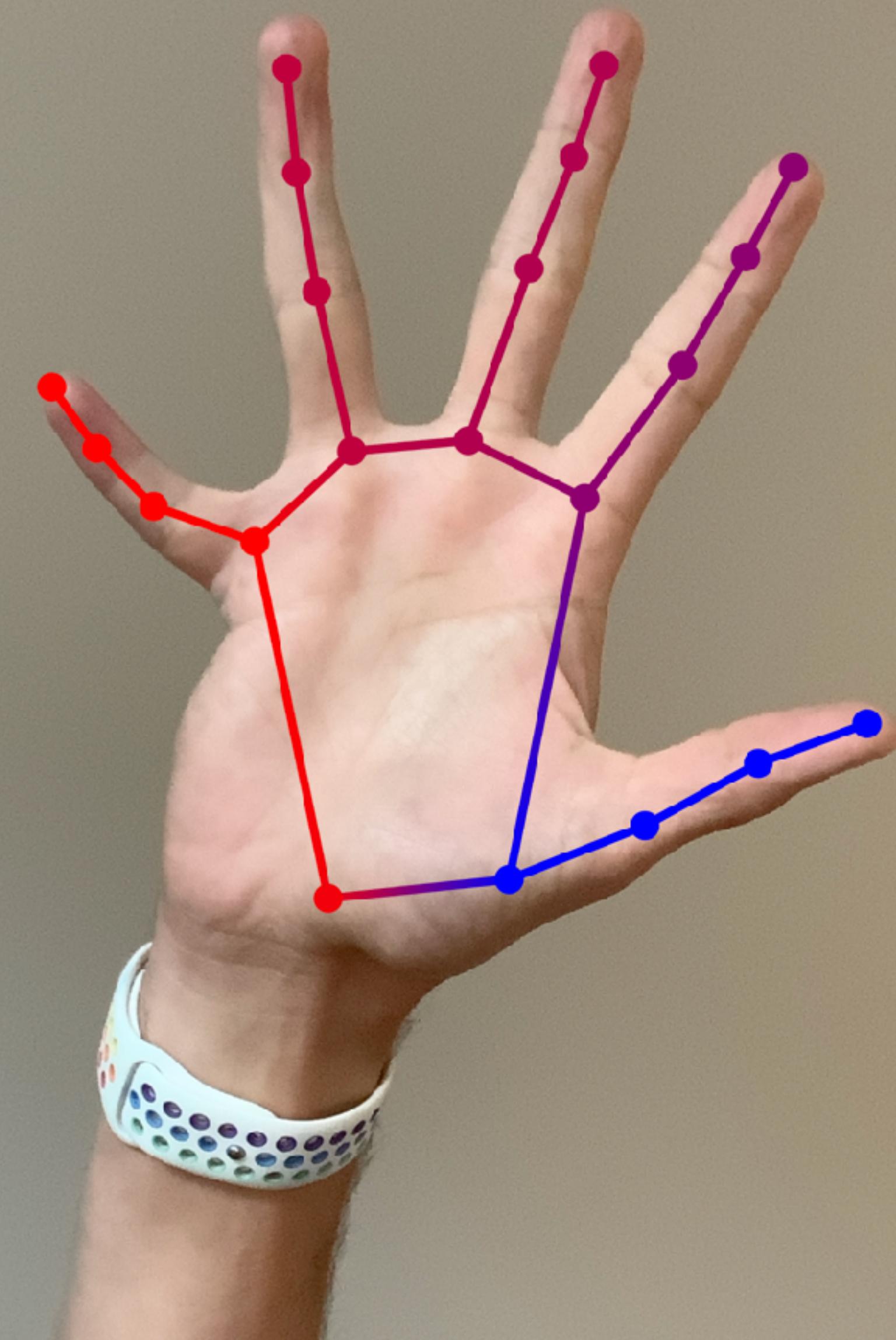


Hand Pose Request

Detects up to 21 points on a hand

4 points per finger and thumb

1 point for the wrist



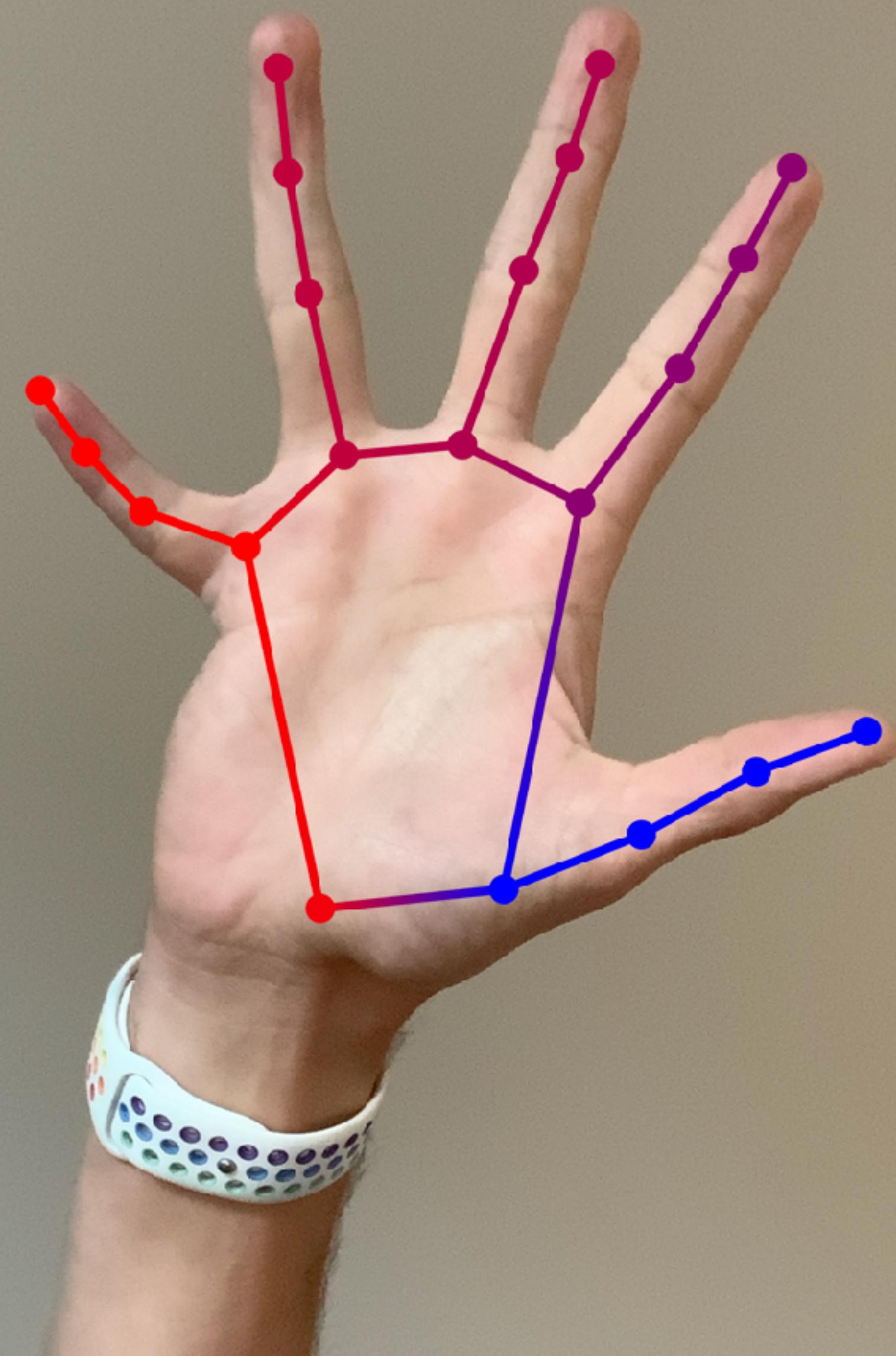
Hand Pose Request*

Detects up to 21 points on a hand

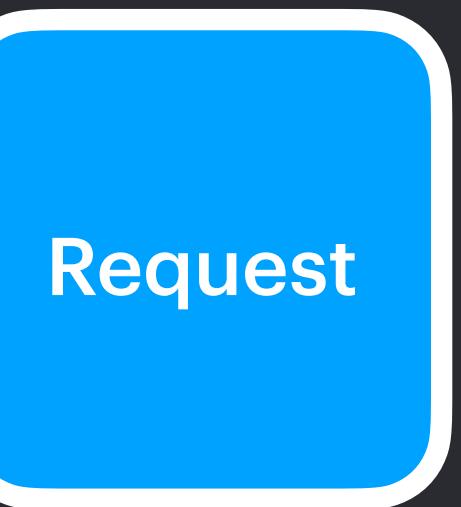
4 points per finger and thumb

1 point for the wrist

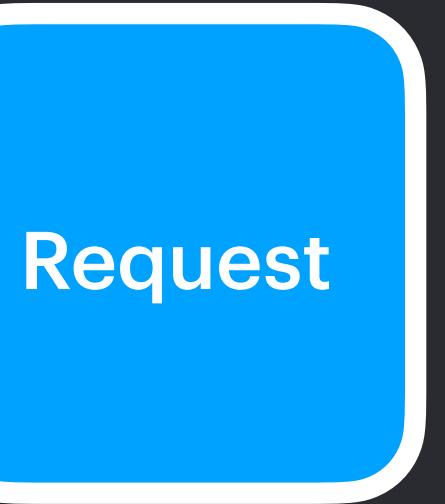
* Human hands



```
// Create the Vision request  
let request = VNRequest()
```

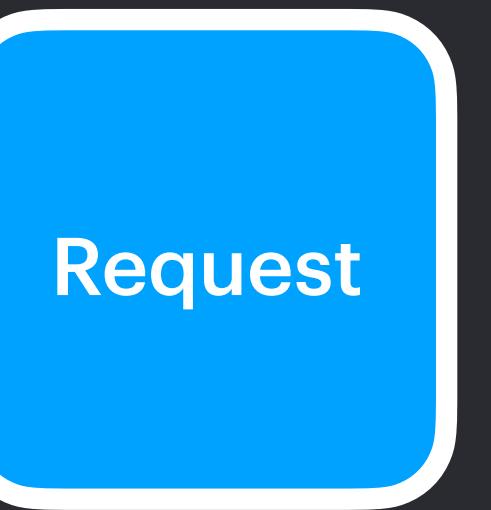


```
// Create the Vision request  
let request = VNRequest<VNHumanHandObservation>(  
    type: .hand,  
    features: [.pose])  
  
// Configure the request  
request.maximumHandCount = 1
```



```
// Create the Vision request
let request = VNRequest<VNHumanHandObservation>(VNRequestType.detectHumanHand)
request.maximumHandCount = 1

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])
```

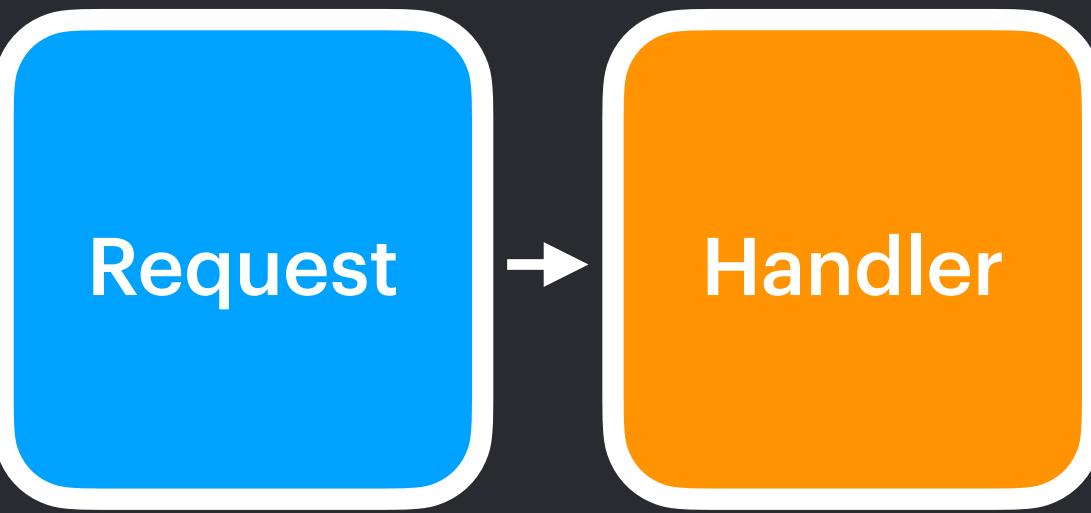


```
// Create the Vision request
let request = VNRequest()
request.name = "Human Hand Pose"
request.maximumHandCount = 1

// Configure the request
request.minimumDetectionConfidence = 0.5
request.minNumKeyPoints = 2
request.maxNumKeyPoints = 20

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])

// Perform the request
try? requestHandler.perform([request])
```



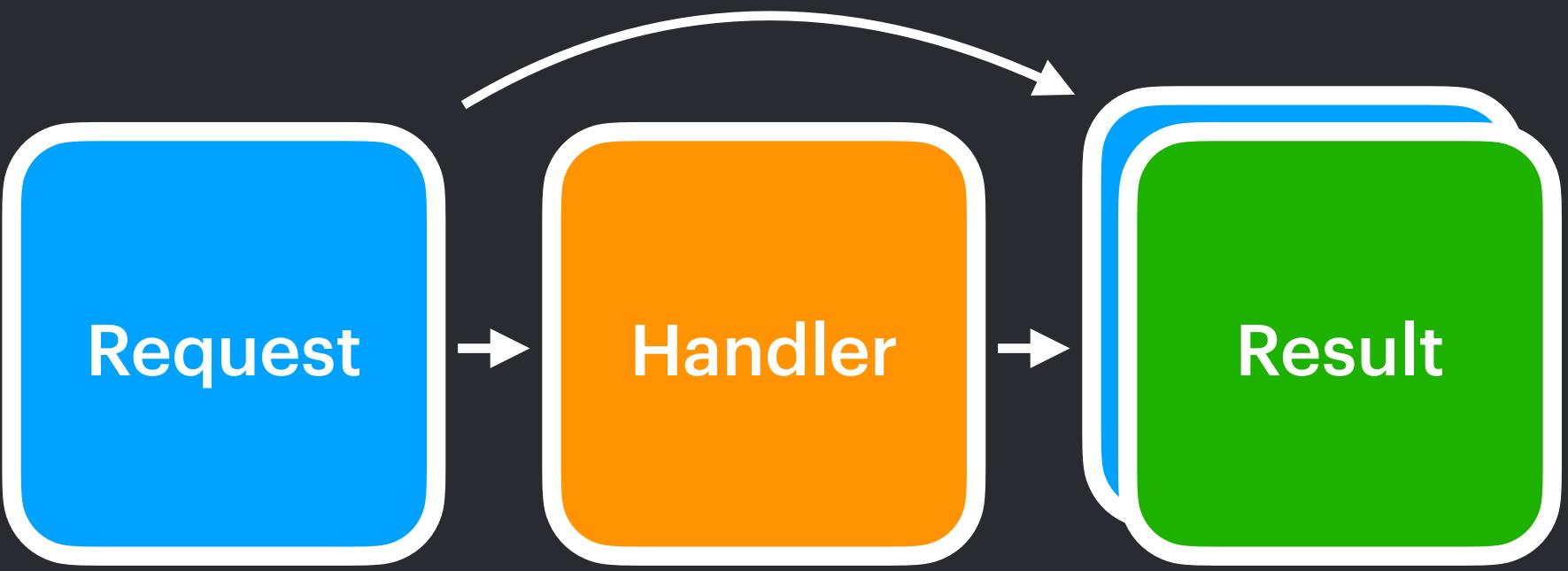
```
// Create the Vision request
let request = VNRequest()
request.name = "Human Hand Pose"
request.maximumHandCount = 1

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])

// Perform the request
try? requestHandler.perform([request])

// Check results
if let results = request.results as? [VNHumanHandPoseObservation],
    let result = results.first {

}
```

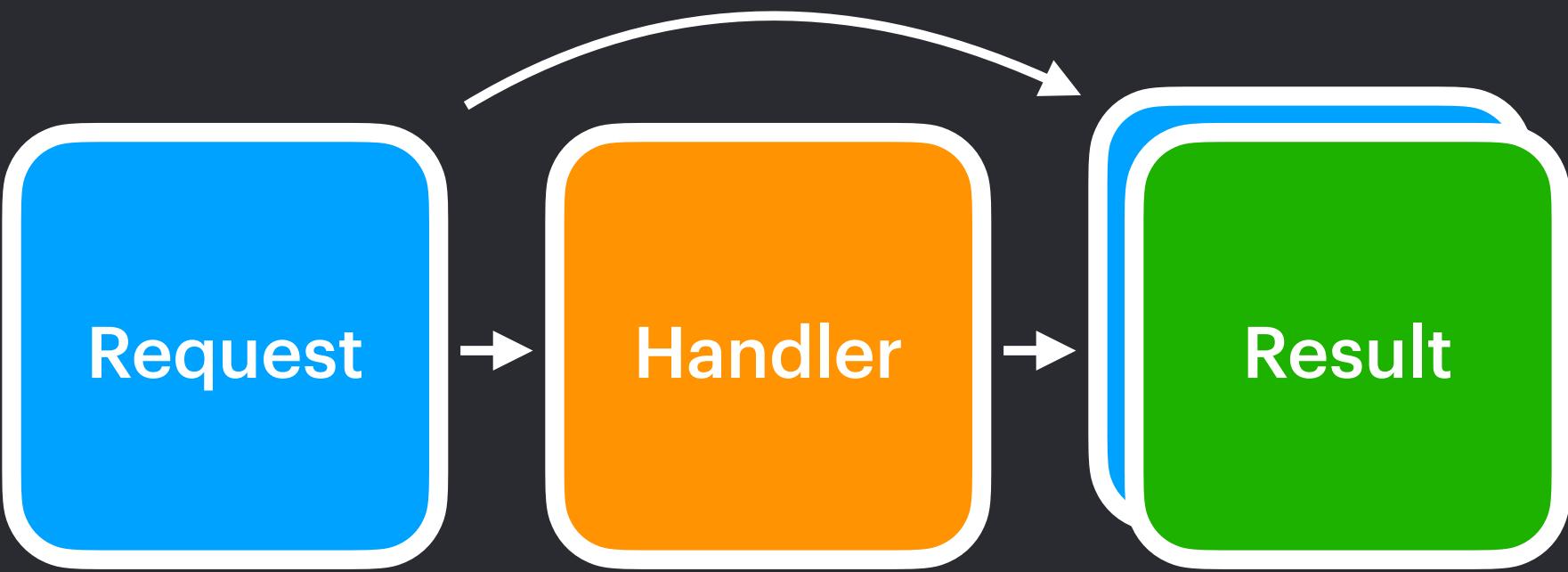


```
// Create the Vision request
let request = VNRequest()
request.name = "Human Hand Pose"
request.maximumHandCount = 1

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])

// Perform the request
try? requestHandler.perform([request])

// Check results
if let results = request.results as? [VNHumanHandPoseObservation],
    let result = results.first {
    // Do awesome things with the result
    return Pinch(from: result)
}
```



Crushing Your Head

The App

Hand Pose Detector

Face Detector

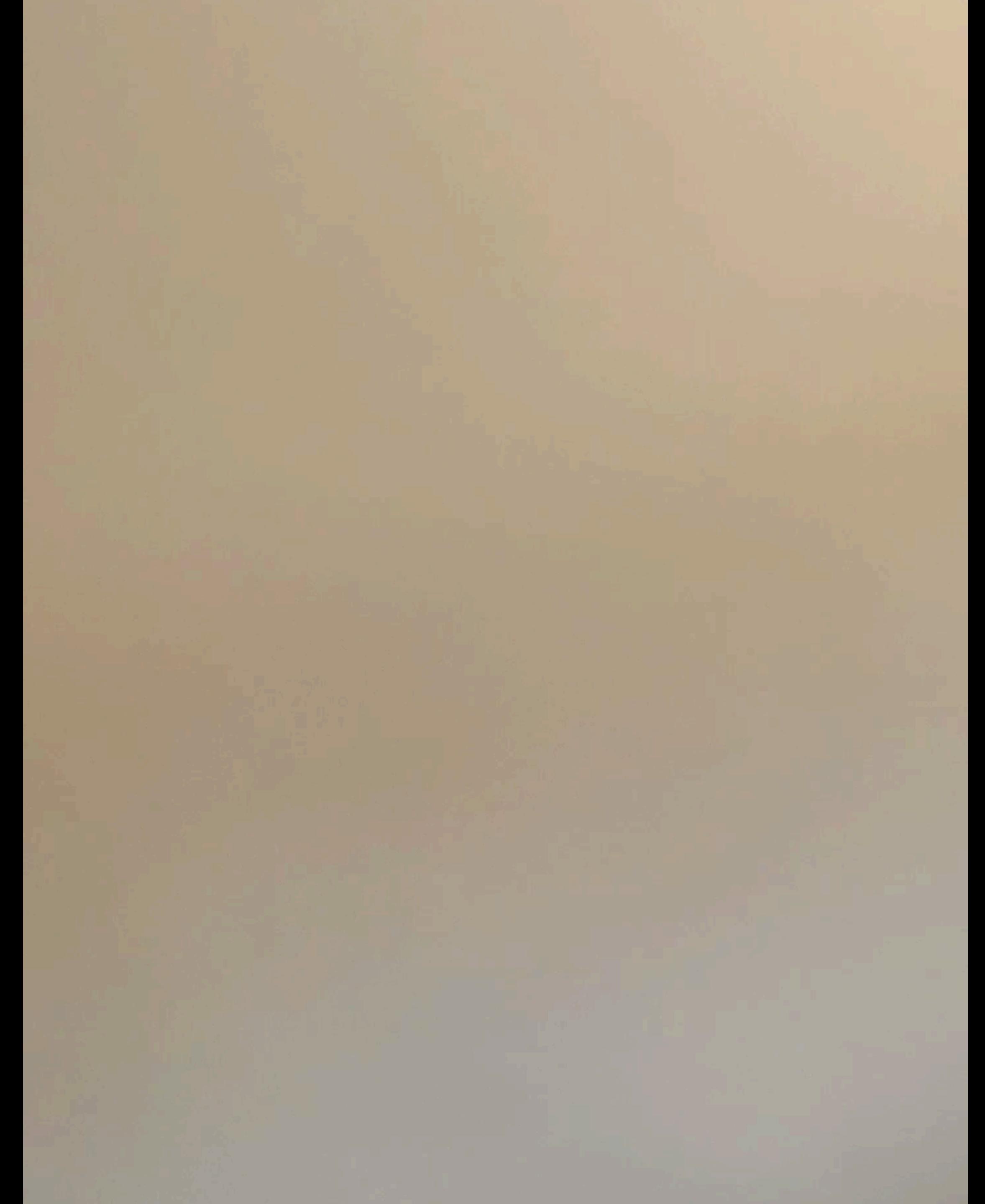
Comparing Feature Vectors

Object Tracking

BONUS:

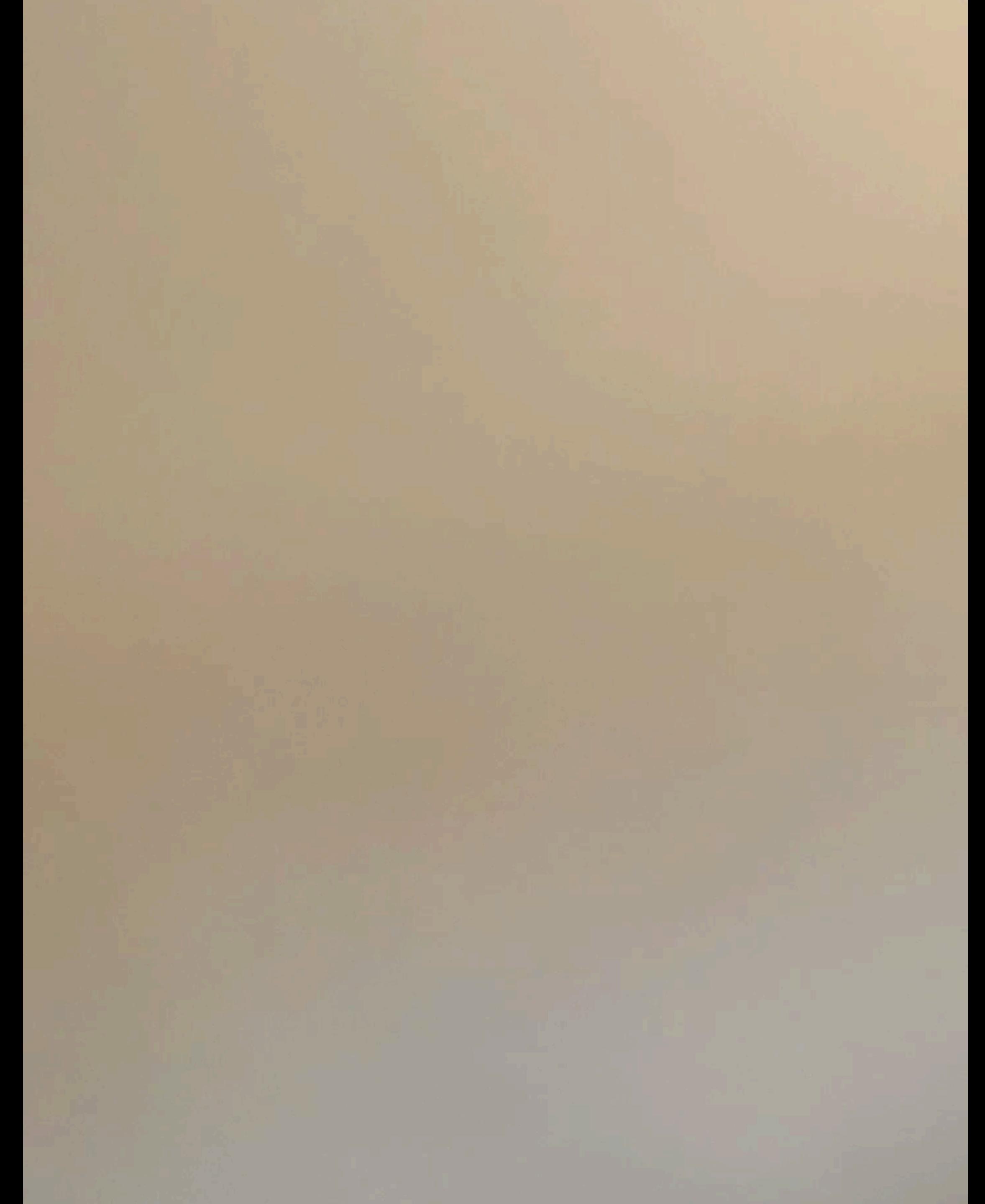
Custom Classifier

Face Detection in the App



Face Detection in the App

Goal of the app is to crush heads



Face Detection in the App

Goal of the app is to crush heads

Faces are a good proxy for heads

Face Detection in the App

Goal of the app is to crush heads

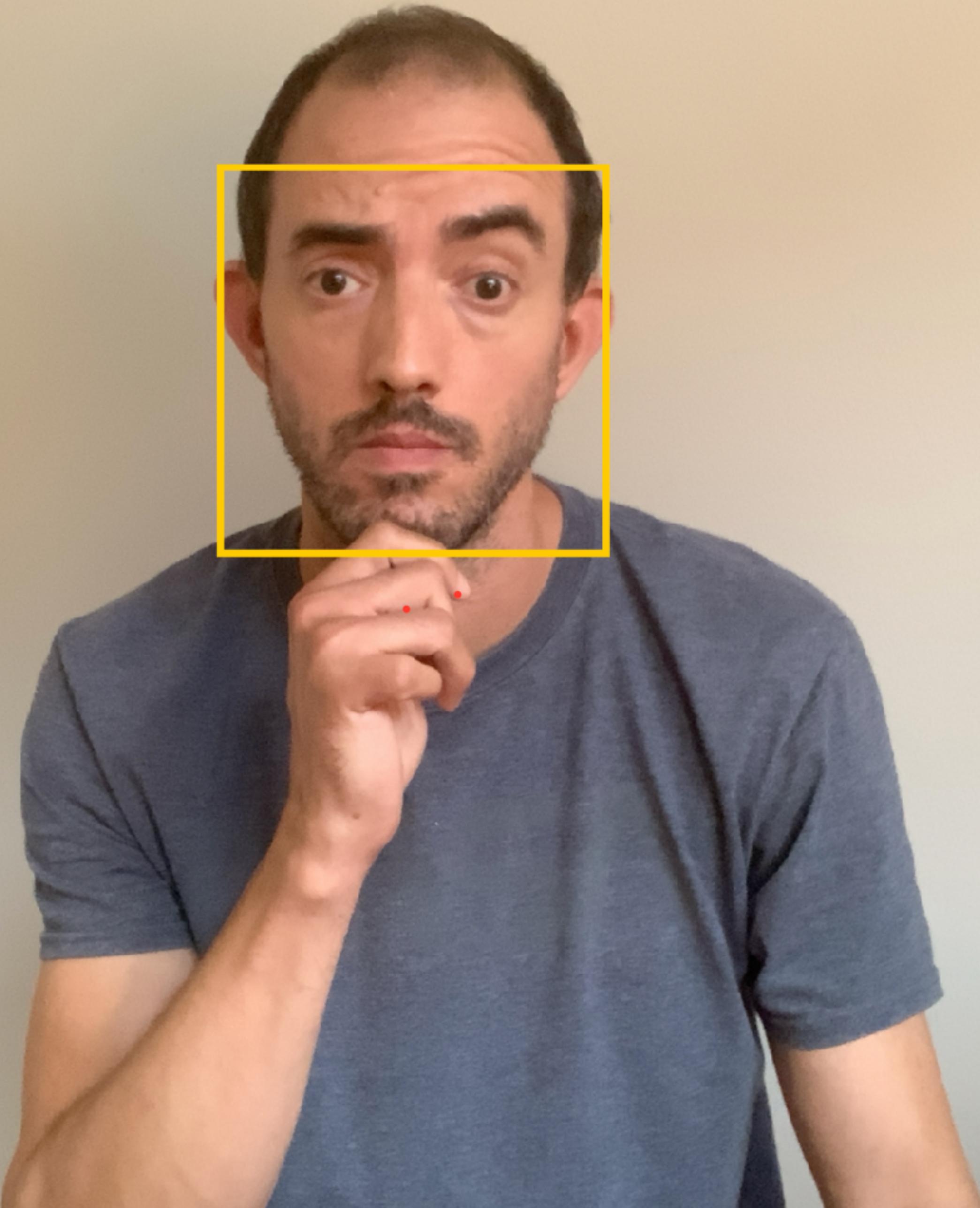
Faces are a good proxy for heads

If a pinch occurs over a detected face, we can be reasonably sure the head was crushed

Face Rectangles Request

Face Rectangles Request

Normal faces



Face Rectangles Request

Normal faces

Profile faces

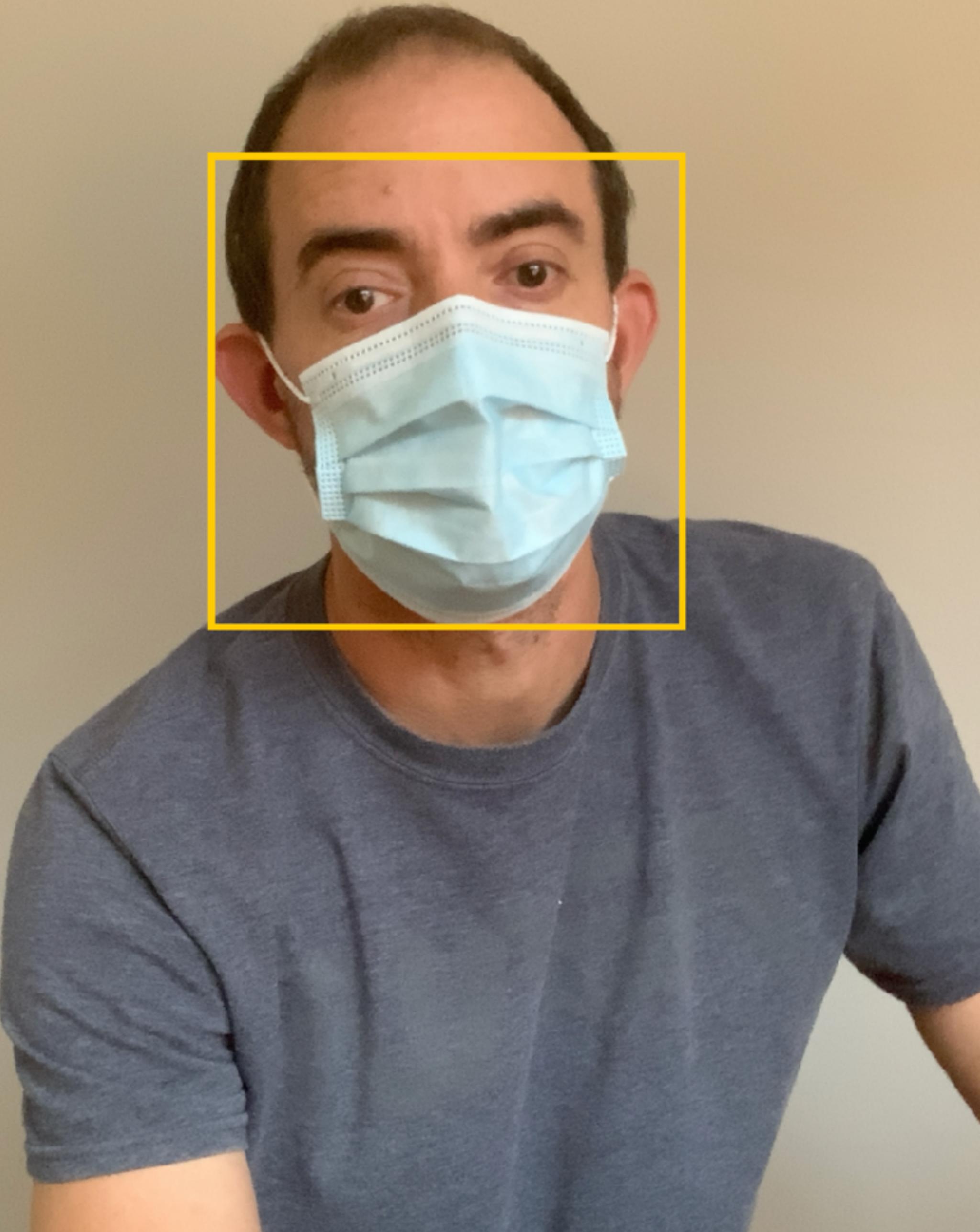


Face Rectangles Request

Normal faces

Profile faces

Obscured faces



Face Rectangles Request

Normal faces

Profile faces

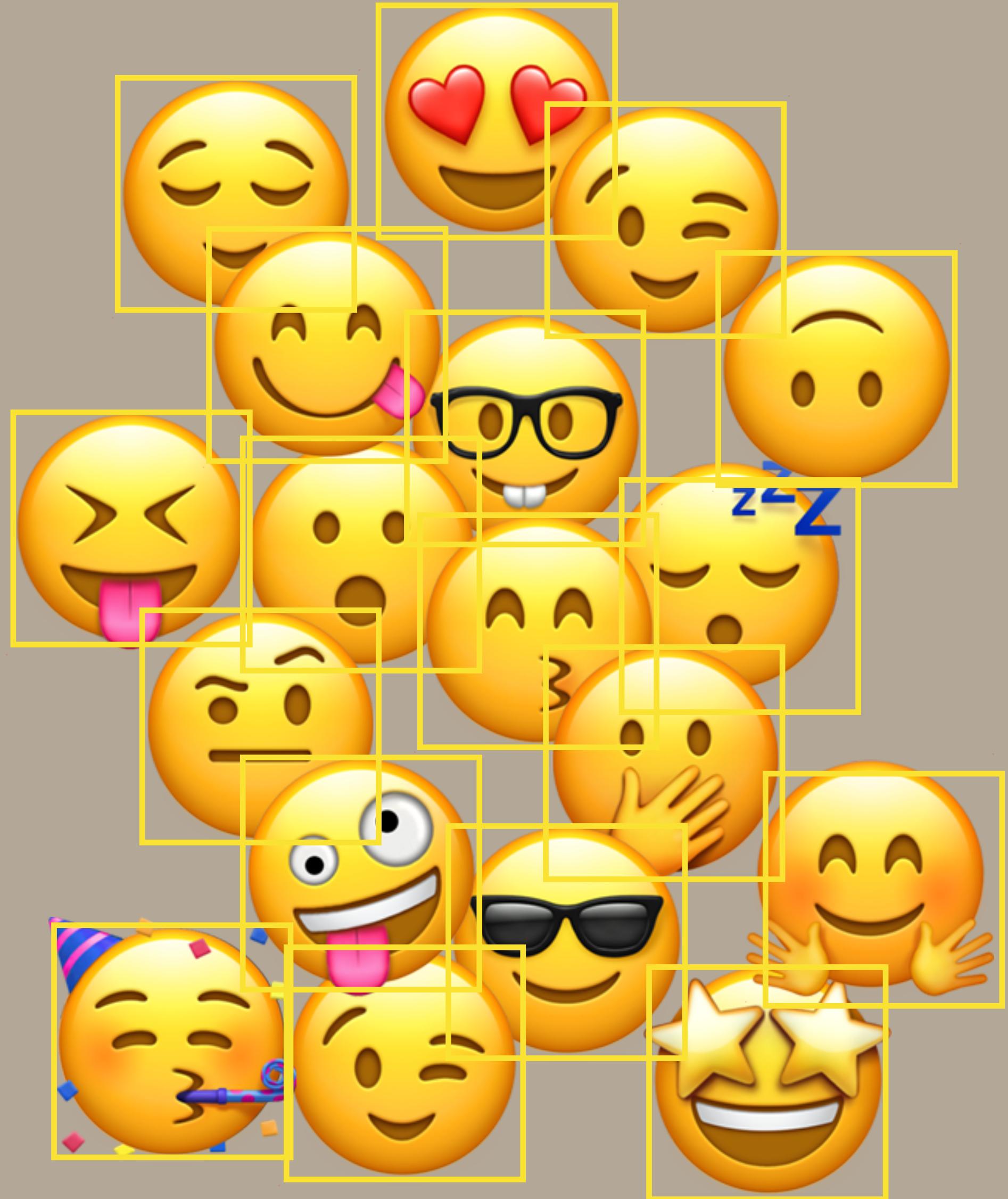
Obscured faces

Small faces

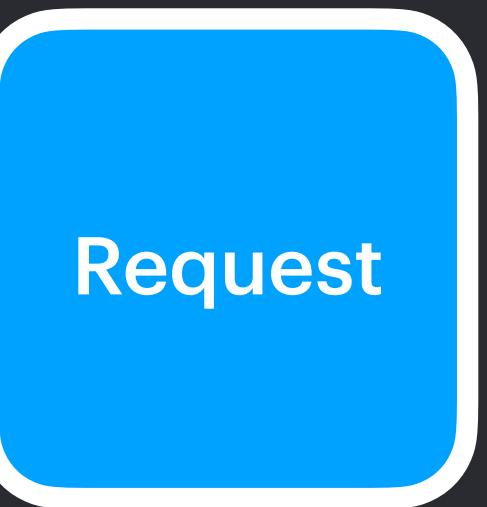


Face Rectangles Request

- Normal faces
- Profile faces
- Obscured faces
- Small faces
- Multiple faces



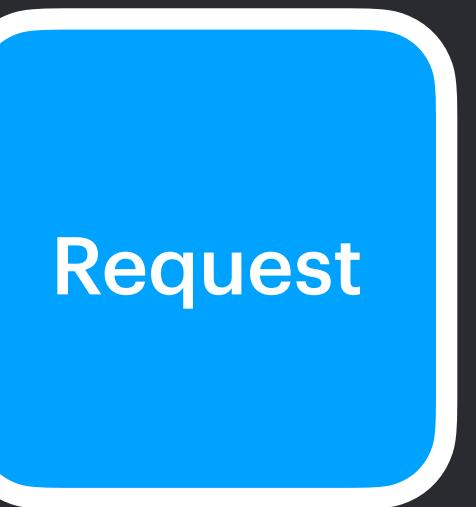
```
// Create the Vision request  
let request = VNRequest(rectangle)
```



```
// Create the Vision request  
let request = VNRequest(rectangle)
```



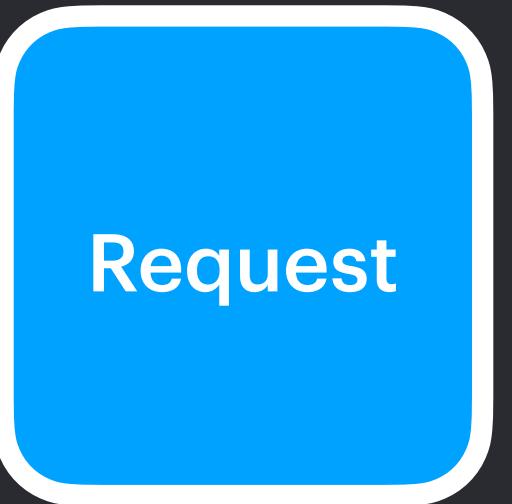
```
// Configure the request  
request.regionOfInterest = CGRect(x: 0.2, y: 0.5, width: 0.6, height: 0.3)
```



```
// Create the Vision request
let request = VNRequest(rectangle)

// Configure the request
request.regionOfInterest = CGRect(x: 0.2, y: 0.5, width: 0.6, height: 0.3)

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])
```

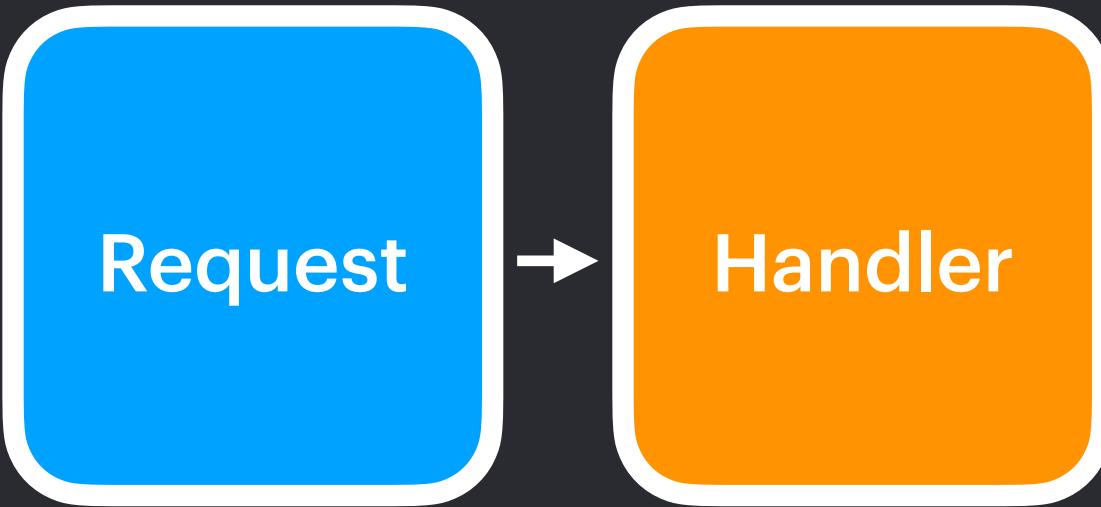


```
// Create the Vision request
let request = VNRequest(rectangle)

// Configure the request
request.regionOfInterest = CGRect(x: 0.2, y: 0.5, width: 0.6, height: 0.3)

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])

// Perform the request
try? requestHandler.perform([request])
```



```
// Create the Vision request
let request = VNRequest(rectangleFromImage: image)

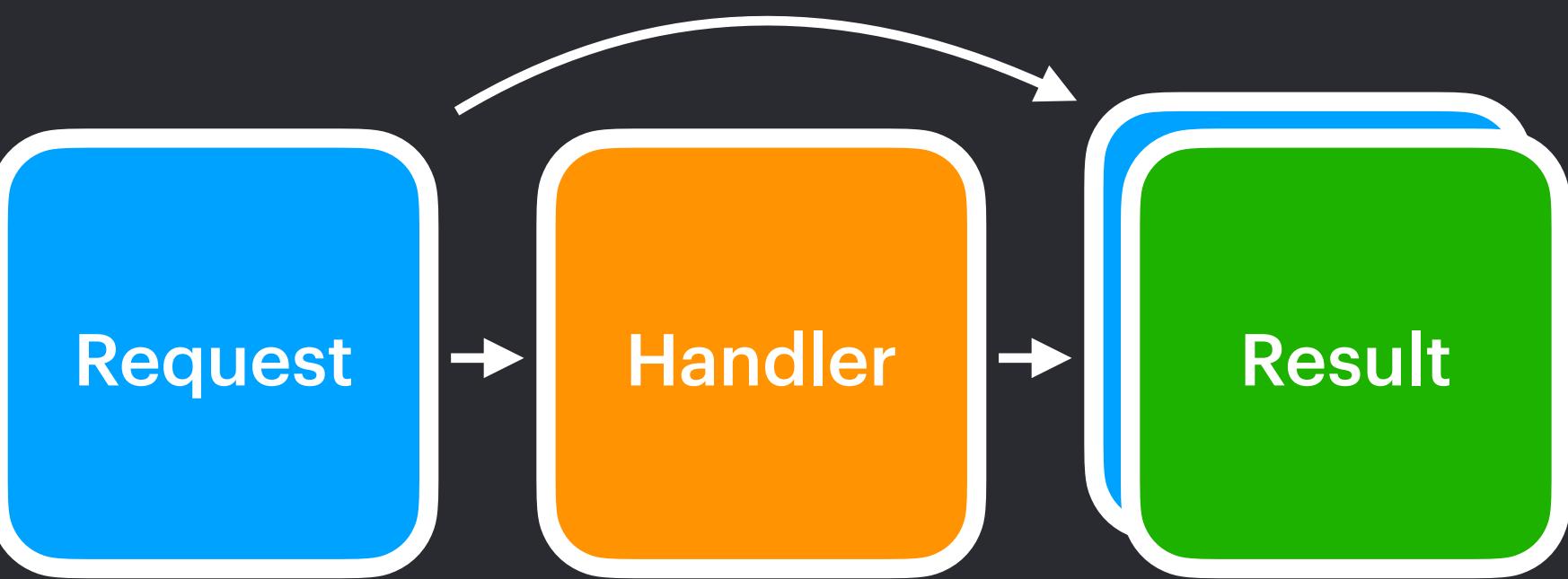
// Configure the request
request.regionOfInterest = CGRect(x: 0.2, y: 0.5, width: 0.6, height: 0.3)

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])

// Perform the request
try? requestHandler.perform([request])

// Check results
if let results = request.results as? [VNFaceObservation] {

}
```



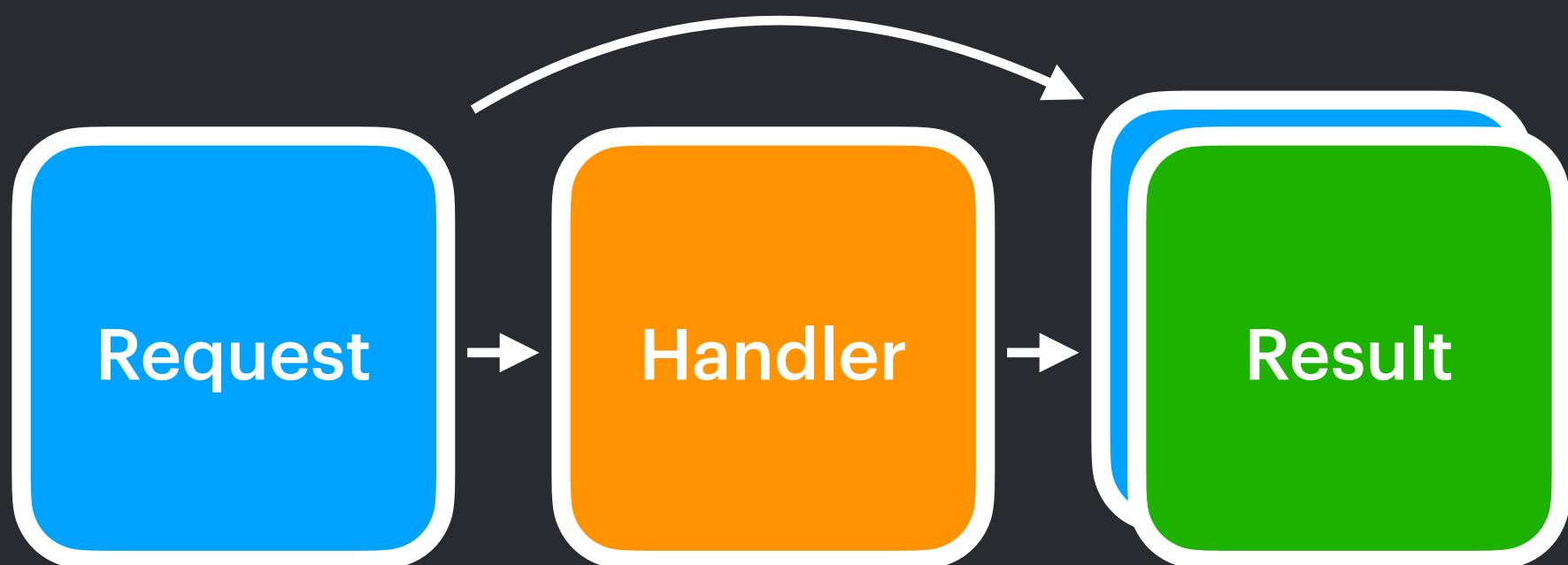
```
// Create the Vision request
let request = VNRequest(rectangleFromImage: image)

// Configure the request
request.regionOfInterest = CGRect(x: 0.2, y: 0.5, width: 0.6, height: 0.3)

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: cvImage, options: [:])

// Perform the request
try? requestHandler.perform([request])

// Check results
if let results = request.results as? [VNFaceObservation] {
    // Do awesome things with the results
    return results.map { Head(id: $0.uuid, bbox: $0.boundingBox) }
}
```



One Problem

Players can game the system

SCORE:

0

Crushing Your Head

The App

Hand Pose Detector

Face Detector

Comparing Feature Vectors

Object Tracking

BONUS:

Custom Classifier

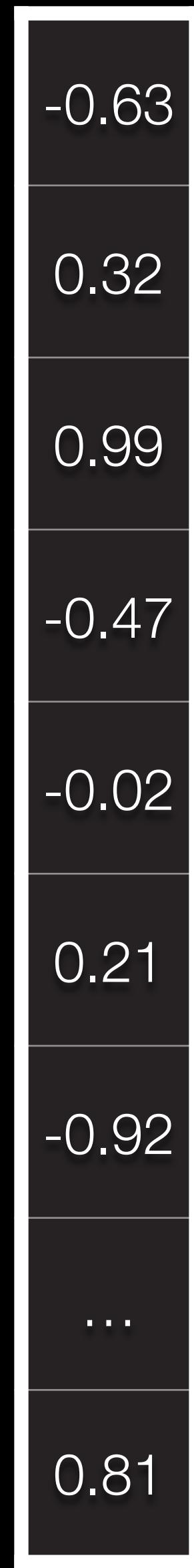
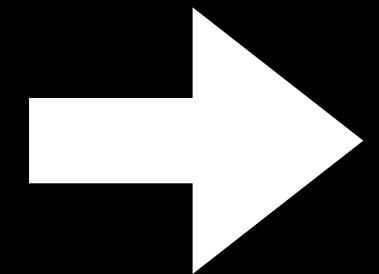
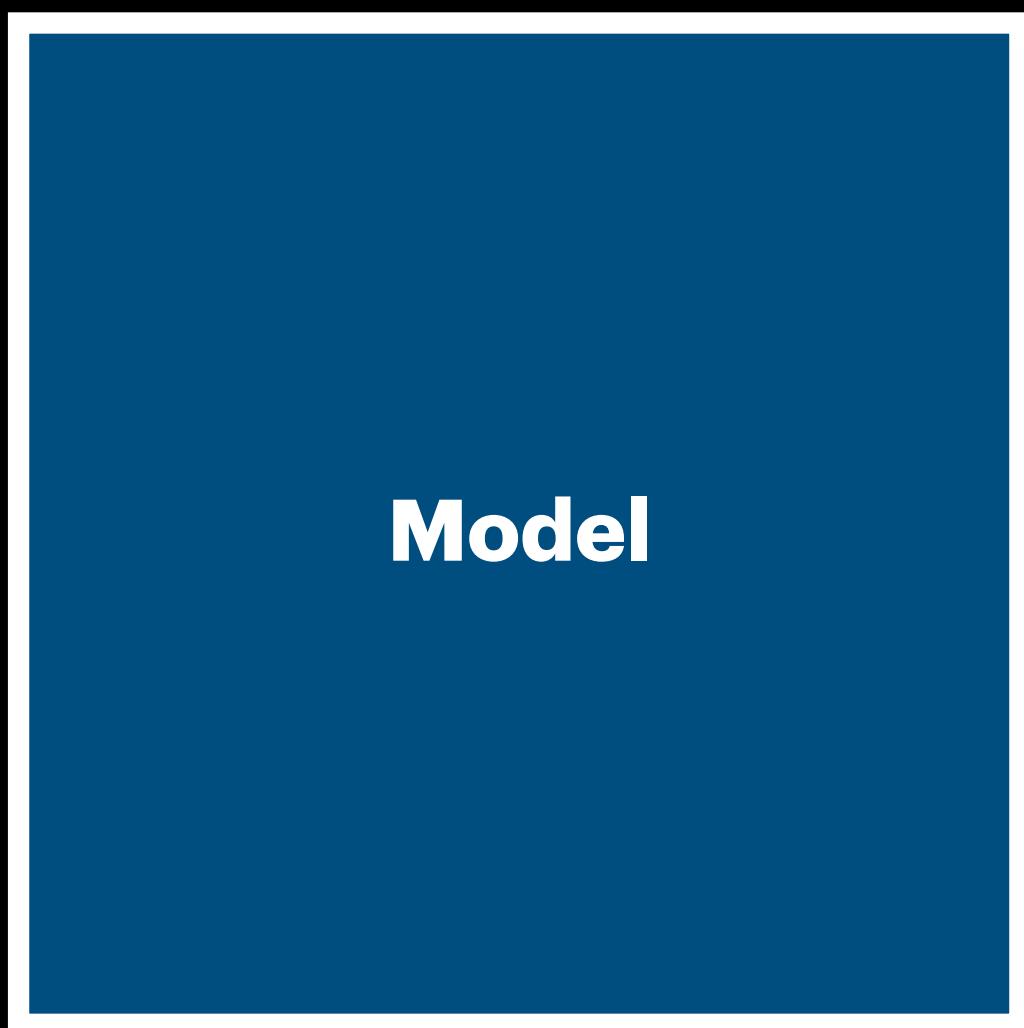
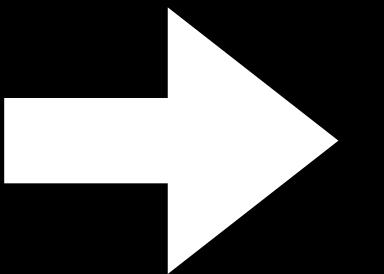
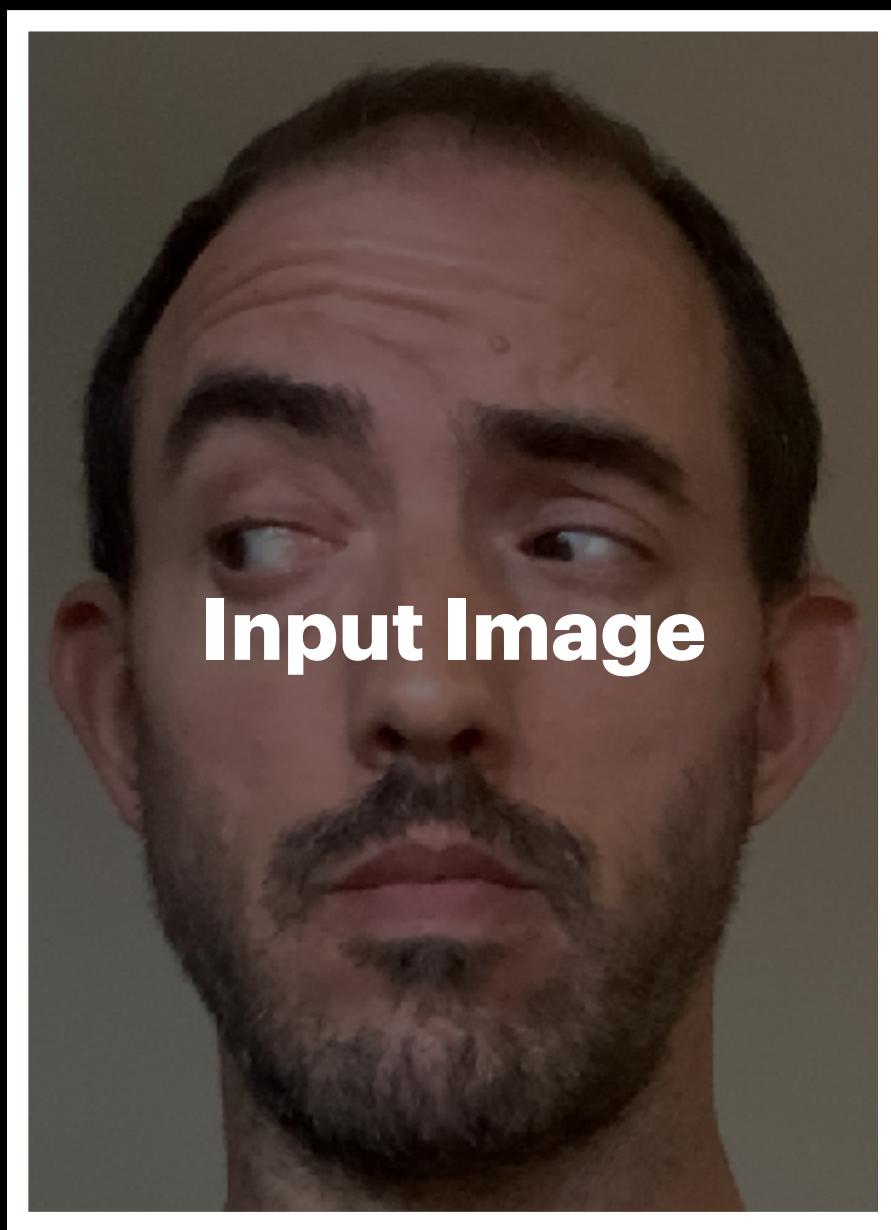
Feature Vector

a.k.a. Feature Print

a.k.a. Feature Embedding

An array of numbers representing important characteristics of an image

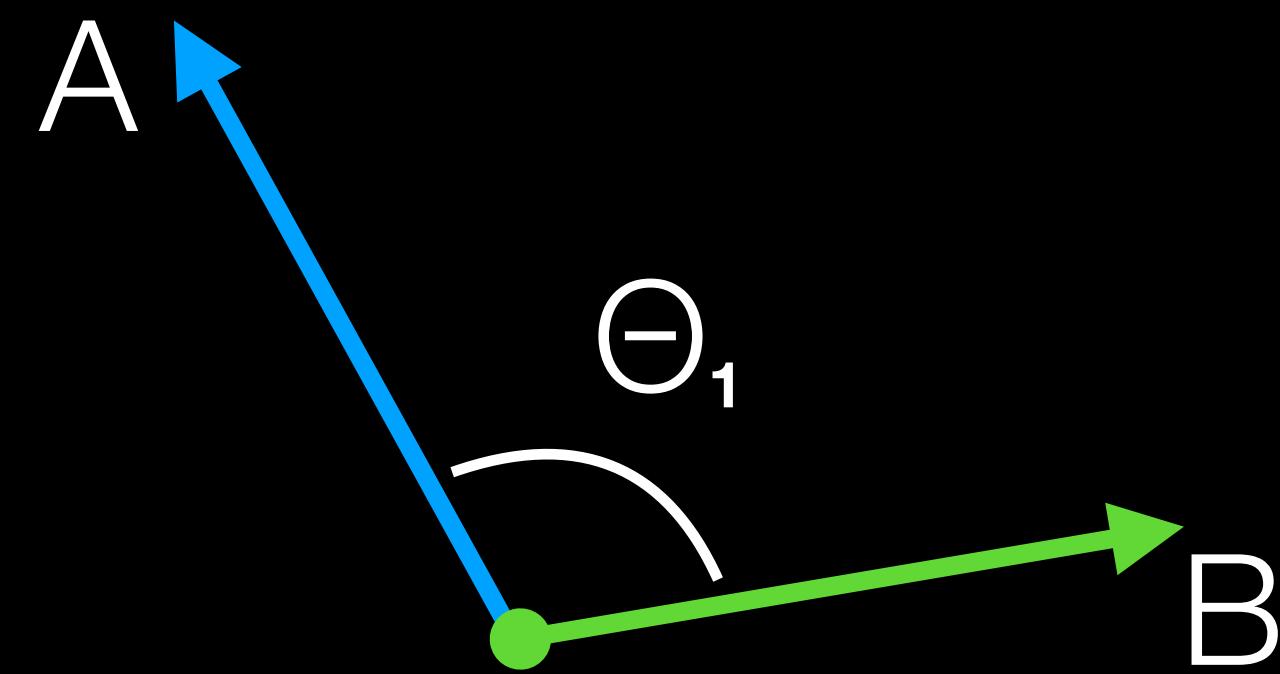
Feature Vector Generation



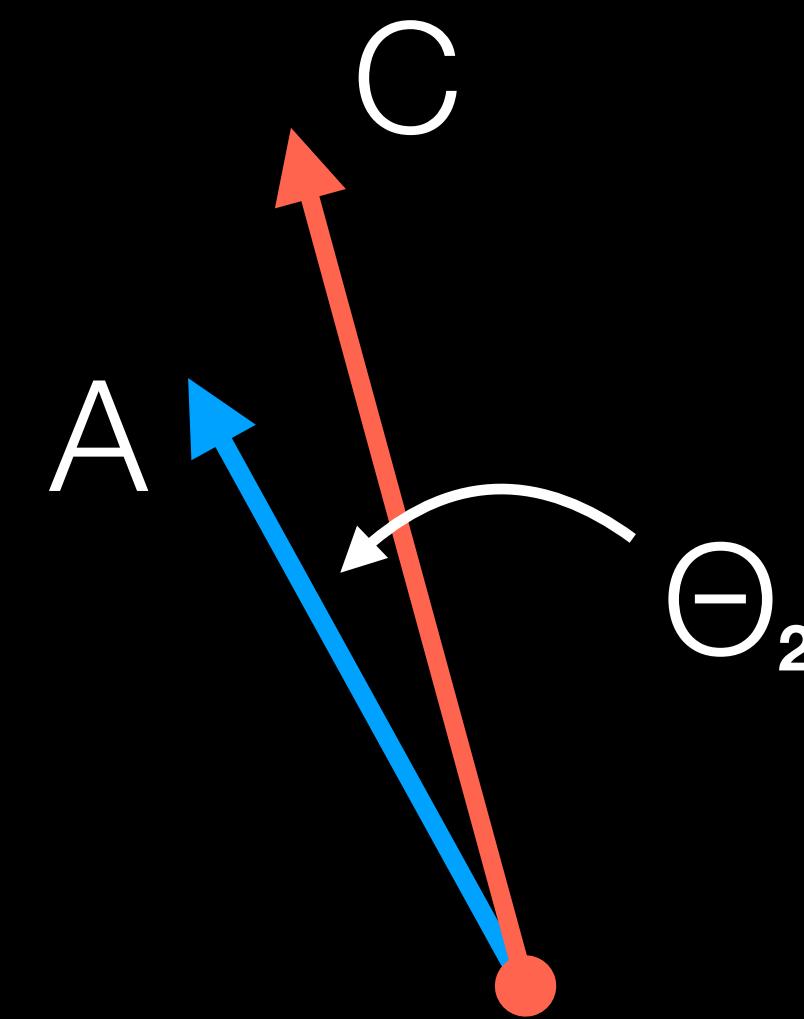
WARNING: Math Ahead

Comparing 2D Vectors

The Dot Product



Less similar

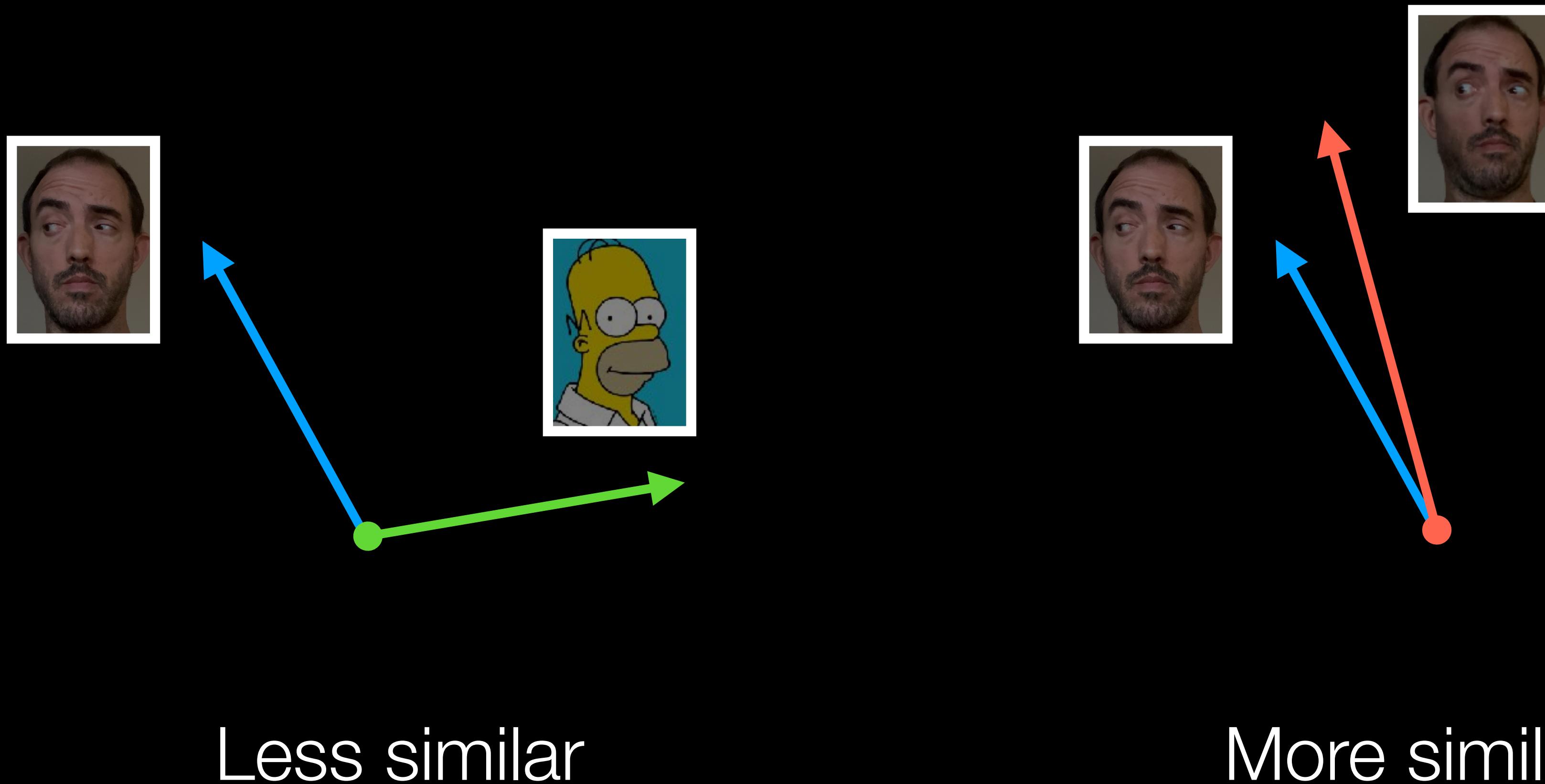


More similar

Even works on 1000s of dimensions
i.e. Feature Vectors



Comparing Feature Vectors



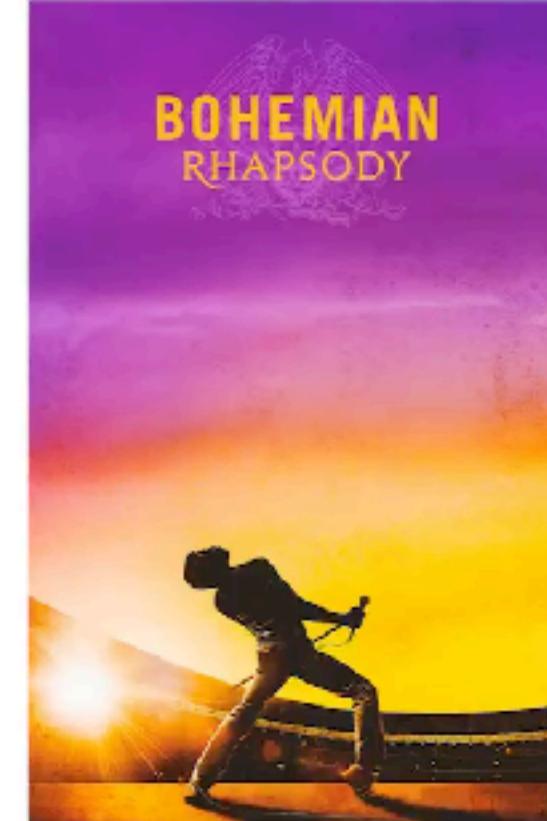
Feature Vector Demo

Vuu
Ålesund, NO

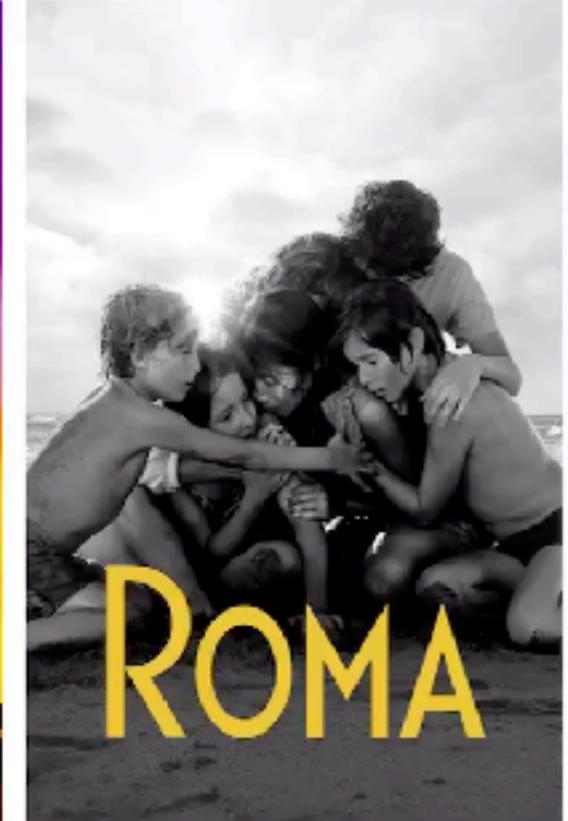
Search

Movies, TV-Series, Cast or Crew

Featured Movies & TV Series
Mrs. Robinson, you're trying to seduce me



Bohemian Rhapsody
2018



Roma
2018



Activity

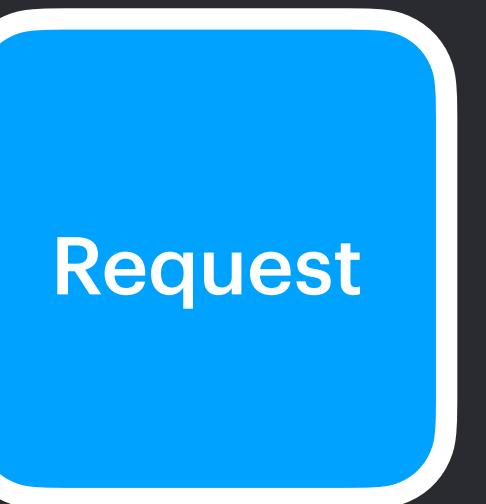
Search

Profile

```
// Create the Vision request  
let request = VNGenerateImageFeaturePrintRequest()
```



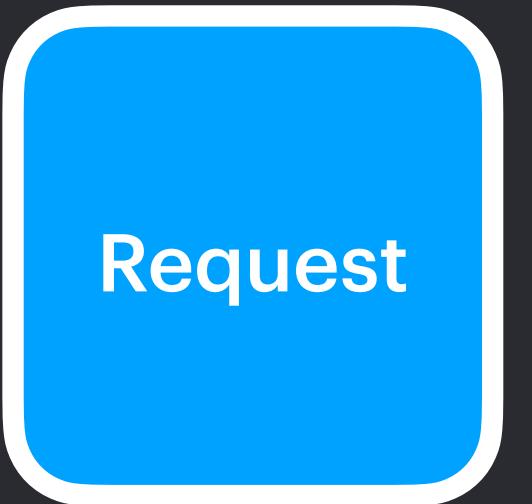
```
// Create the Vision request  
let request = VNGenerateImageFeaturePrintRequest()  
  
// Configure the request  
request.imageCropAndScaleOption = .centerCrop
```



```
// Create the Vision request
let request = VNGenerateImageFeaturePrintRequest()

// Configure the request
request.imageCropAndScaleOption = .centerCrop

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])
```

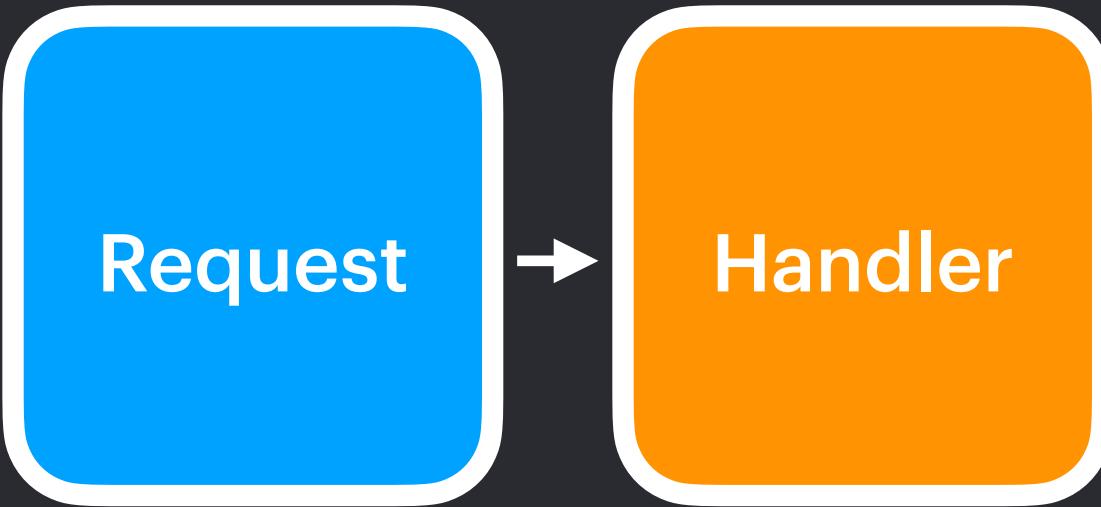


```
// Create the Vision request
let request = VNGenerateImageFeaturePrintRequest()

// Configure the request
request.imageCropAndScaleOption = .centerCrop

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])

// Perform the request
try? requestHandler.perform([request])
```



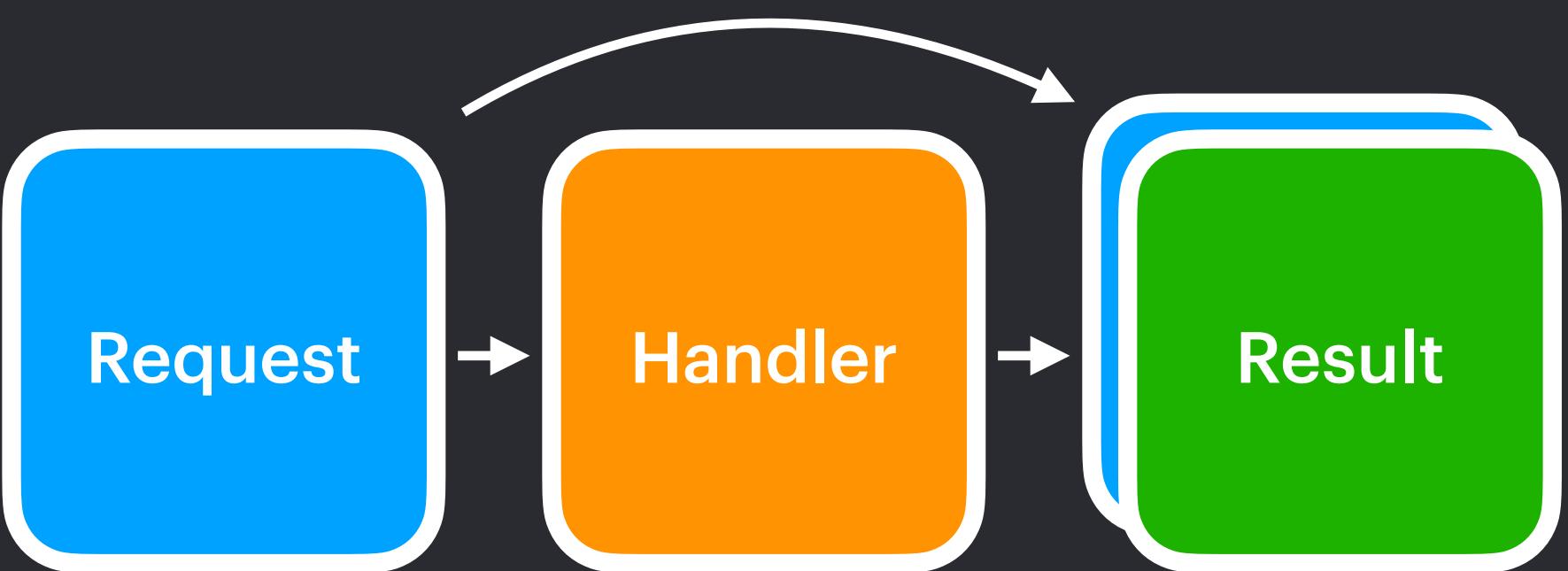
```
// Create the Vision request
let request = VNGenerateImageFeaturePrintRequest()

// Configure the request
request.imageCropAndScaleOption = .centerCrop

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])

// Perform the request
try? requestHandler.perform([request])

// Check results
if let results = request.results as? [VNFeaturePrintObservation],
    let result = results.first {
    // result.type will be set to .float or .double
    // result.data will be the feature vector represented as raw Data bytes
}
```



Feature Vectors Comparison in the App

Feature Vectors Comparison in the App

It's not in the app

Feature Vectors Comparison in the App

It's not in the app because:

1. Requires determining a similarity threshold

Feature Vectors Comparison in the App

It's not in the app because:

1. Requires determining a similarity threshold
2. Depends on whether Apple's model can discriminate well between faces

Feature Vectors Comparison in the App

It's not in the app because:

1. Requires determining a similarity threshold
2. Depends on whether Apple's model can discriminate well between faces
3. There's a simpler solution

Crushing Your Head

The App

Hand Pose Detector

Face Detector

Comparing Feature Vectors

Object Tracking

BONUS:

Custom Classifier

SCORE:

0

Object Tracking in the App

SCORE:

0

Object Tracking in the App

Each detected object has a UUID

SCORE:

0

Object Tracking in the App

Each detected object has a UUID

Track heads across frames

SCORE:

0

Object Tracking in the App

Each detected object has a UUID

Track heads across frames

Useful for preventing MHC

SCORE:

0

Object Tracking in the App

Each detected object has a UUID

Track heads across frames

Useful for preventing MHC*

* Multiple Head Crushings

SCORE:

0

Object Tracking in the App

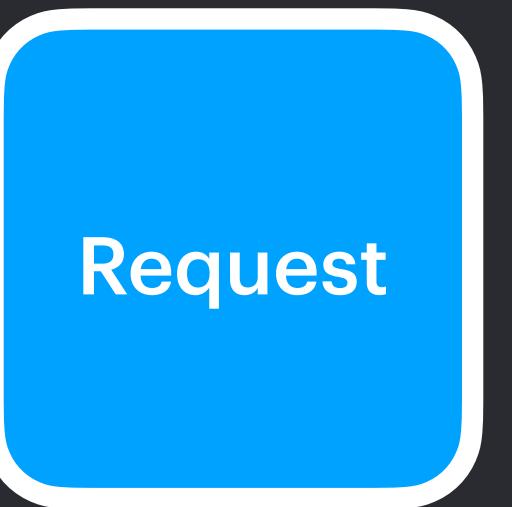
Each detected object has a UUID

Track heads across frames

Useful for preventing MHC* 

* Multiple Head Crushings

```
// Create array of Vision requests  
var requests = [VNRequest]()
```



```
// Create array of Vision requests
var requests = [VNRequest]()

// create a request for every head to be tracked
for head in self.trackedHeads {
    requests.append(VNTrackObjectRequest(detectedObjectObservation: head.value))
}
```



```
// Create array of Vision requests
var requests = [VNRequest]()

// create a request for every head to be tracked
for head in self.trackedHeads {
    requests.append(VNTrackObjectRequest(detectedObjectObservation: head.value))
}

if requests.isEmpty() {
    // detect fresh faces
    requests.append(VNDetectFaceRectanglesRequest())
}

}
```



```
// Create array of Vision requests
var requests = [VNRequest]()

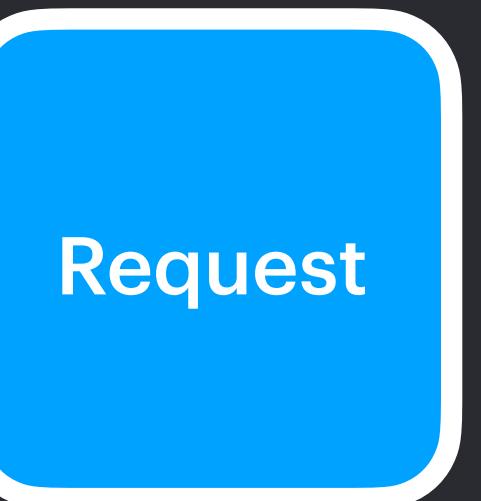
// create a request for every head to be tracked
for head in self.trackedHeads {
    requests.append(VNTrackObjectRequest(detectedObjectObservation: head.value))
}

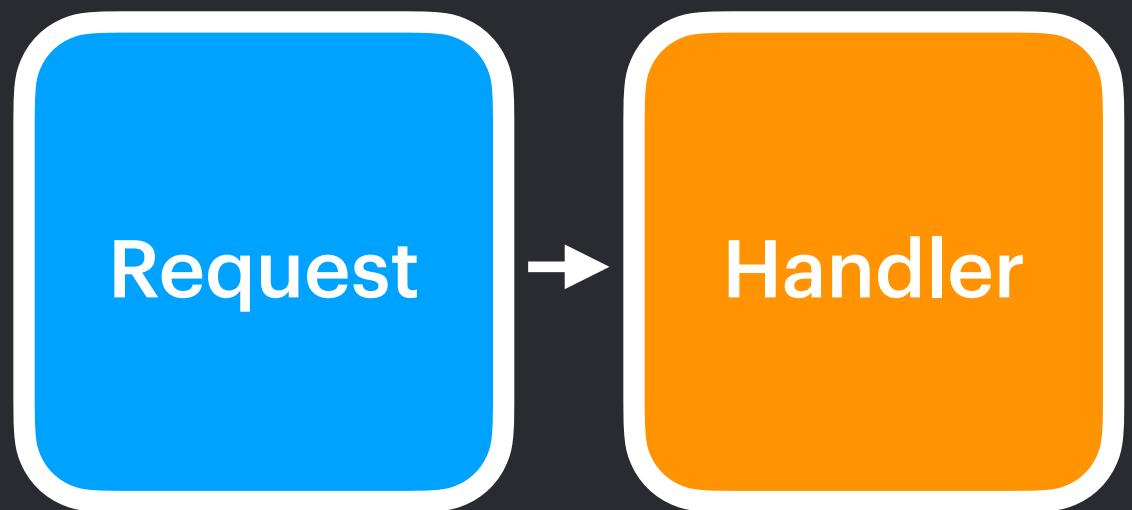
if requests.isEmpty() {
    // detect fresh faces
    requests.append(VNDetectFaceRectanglesRequest())

    // reset the sequence handler
    self.handler = VNSequenceRequestHandler()
}
```



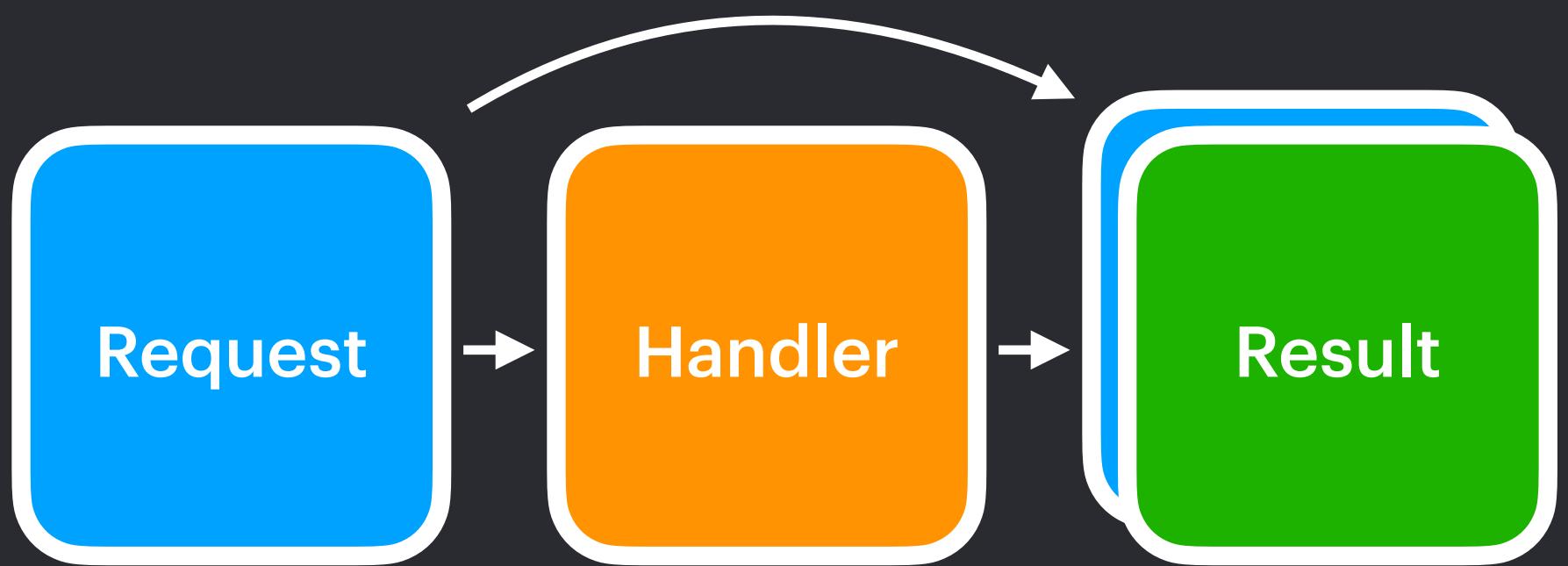
...
...





...

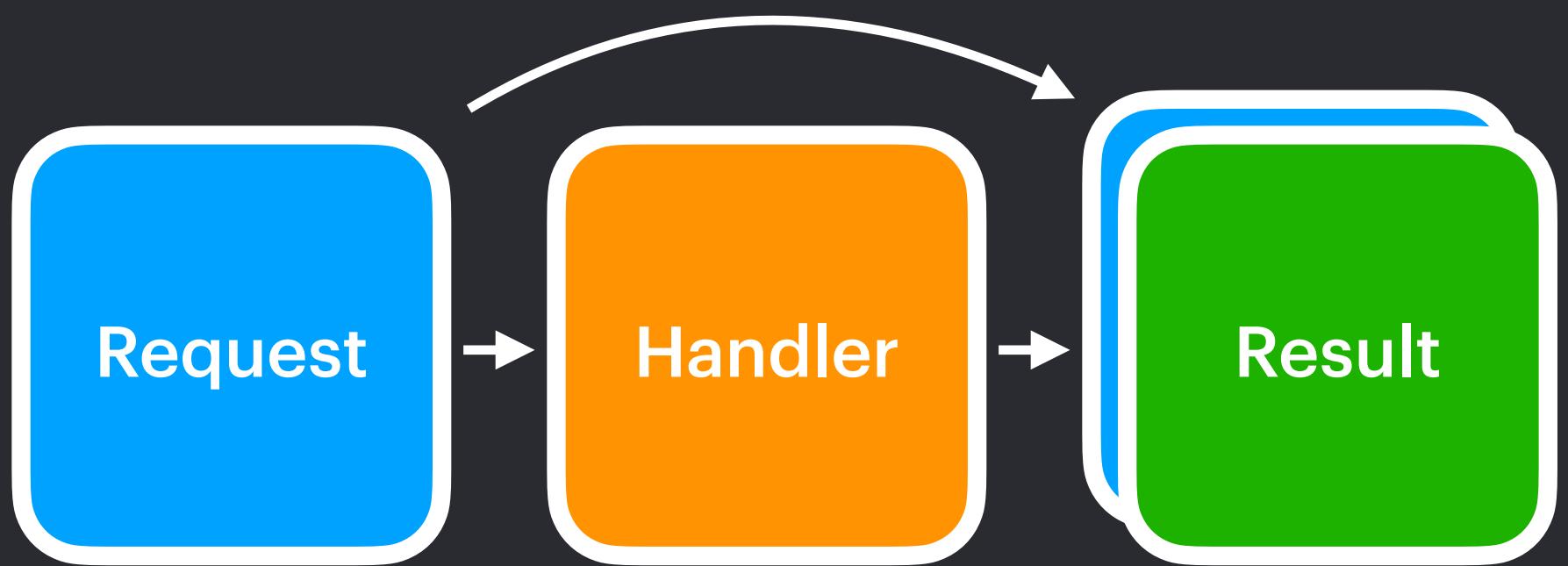
```
// Perform the request  
try? handler.perform(requests, on: image)
```



```
...
// Perform the request
try? handler.perform(requests, on: image, orientation: .up)

// iterate through the completed requests
for request in requests {
    if let results = request.results as? [VNRecognizedObjectObservation] {

    }
}
```



```
...  
  
// Perform the request  
try? handler.perform(requests, on: image, orientation: .up)  
  
  
// iterate through the completed requests  
for request in requests {  
    if let results = request.results as? [VNRecognizedObjectObservation] {  
        // handle a detected/tracked object  
        self.updateTrackedHeads(with: results)  
    }  
}
```

**What if Crushing Heads is too
gruesome for you?**

Smashing Pumpkins?

Crushing Your Head

The App

Hand Pose Detector

Face Detector

Comparing Feature Vectors

Object Tracking

BONUS:

Custom Classifier

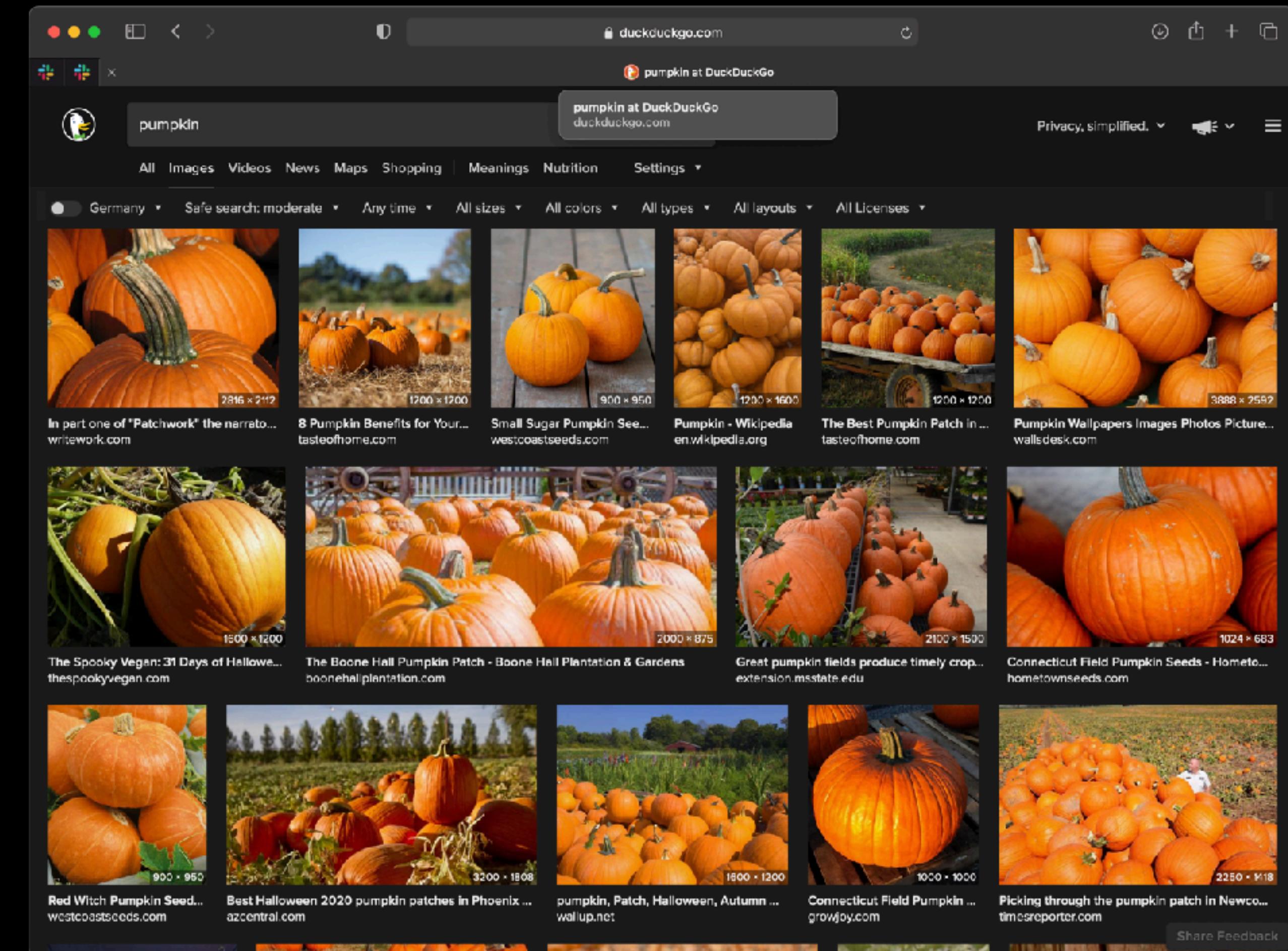
Smashing Pumpkins

Train a custom CoreML classifier

Smashing Pumpkins

Train a custom CoreML classifier

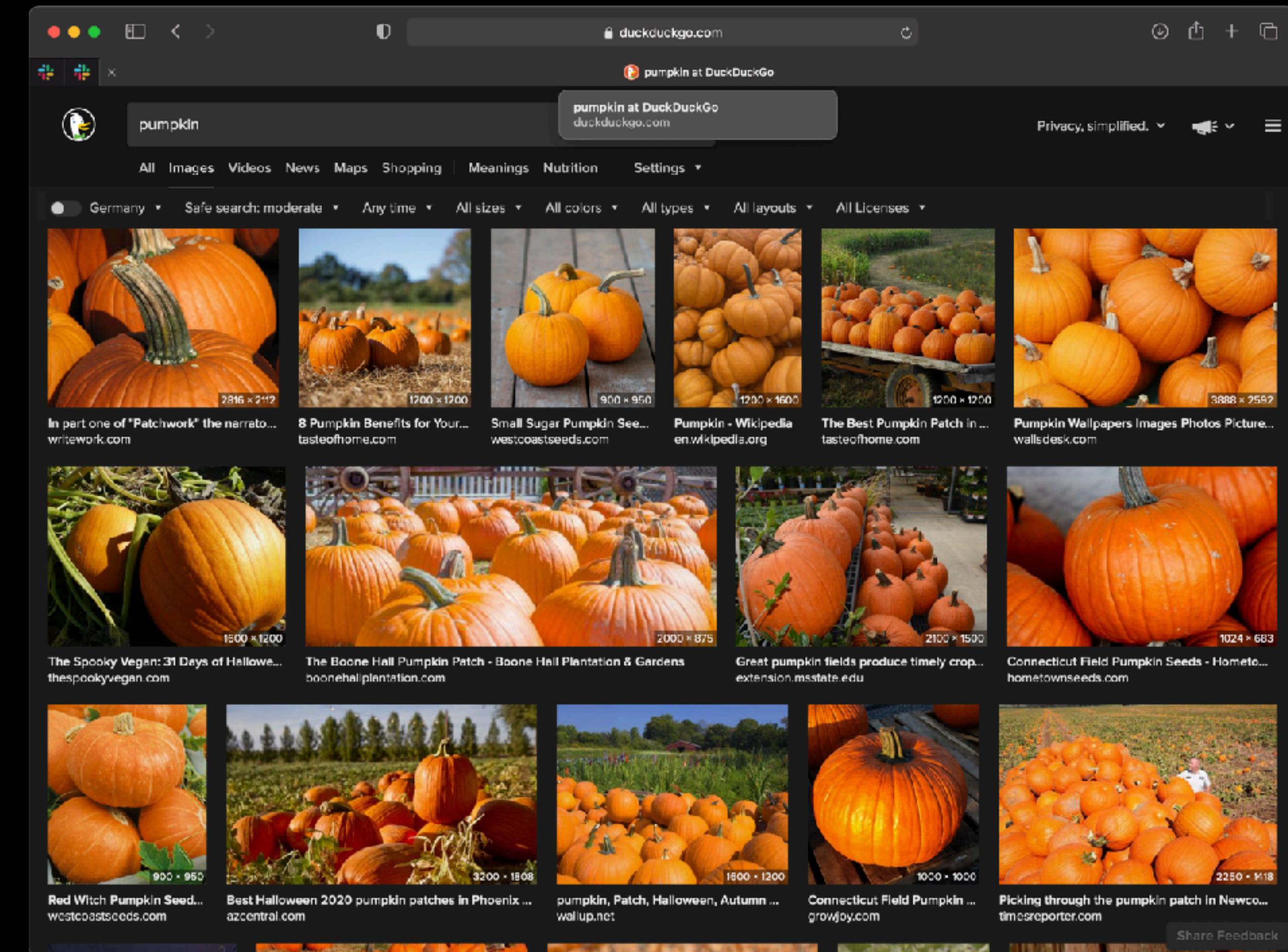
Collect images of pumpkins and not pumpkins



Smashing Pumpkins

Train a custom CoreML classifier

Collect images of pumpkins and
not pumpkins*



* Be mindful of image licenses

Smashing Pumpkins

Train a custom CoreML classifier

Collect images of pumpkins and
not pumpkins*

Organize the folder structure

Name	Date Modified	Size	Kind
> test	Today at 14:39	--	Folder
└ train	Today at 14:58	--	Folder
└ NotAPumpkin	Today at 14:58	--	Folder
1.new-suburban-neig...-happy-113440836.jpg	Today at 14:57	54 KB	JPEG image
2.neighborhood_street_1000x1000.png	Today at 14:57	1.8 MB	PNG image
3.nice-comfortable-n...260nw-468619256.jpg	Today at 14:57	27 KB	JPEG image
4.636995604771030000.jpg	Today at 14:57	423 KB	JPEG image
5.nicely-trimmed-man...60nw-523865659.jpg	Today at 14:57	36 KB	JPEG image
6.upscale-houses-on-...ture-id179060435.jpg	Today at 14:57	49 KB	JPEG image
7.cleveland-houses-re...ture-id593339720.jpg	Today at 14:57	64 KB	JPEG image
8.lizkuball_180506_0...rbank_lima_street.0.jpg	Today at 14:57	201 KB	JPEG image
9.cambridge.jpg	Today at 14:57	524 KB	JPEG image
10.neighborhood-stre...e-seen-153003104.jpg	Today at 14:57	49 KB	JPEG image
11.perfect-neighbour...ture-id532087451.jpg	Today at 14:57	50 KB	JPEG image
12.image-asset.png	Today at 14:57	460 KB	PNG image
13.neighborhood123rf.jpg	Today at 14:57	2.2 MB	JPEG Image
14.neighborhood_residential.jpg	Today at 14:57	146 KB	JPEG image
15.dscf0257mn_edit_gzz8oj.jpg	Today at 14:57	328 KB	JPEG image
16.6c792fc8-02e4-45...eb-14550589_bg1.jpg	Today at 14:57	428 KB	JPEG image
17.1000_f_306418516...zcy6qqa4kuzzw93.jpg	Today at 14:57	331 KB	JPEG image
18.op-res-streets1.jpg	Today at 14:57	414 KB	JPEG image
19.76c982a2e9d0b0adc182faadedb31440.jpg	Today at 14:57	86 KB	JPEG image
20.oldbrooklyn_blog-size.jpg	Today at 14:57	254 KB	JPEG image
21.street.jpg	Today at 14:57	134 KB	JPEG image
22.7805877.jpg	Today at 14:57	283 KB	JPEG image
23.traverse_city_neighborhood_street.jpg	Today at 14:57	982 KB	JPEG image
24.sidewalk-and-stre...ghborhood-c2c152.jpg	Today at 14:57	295 KB	JPEG image
25.inline_image_preview.jpg	Today at 14:57	26 KB	JPEG image
└ Pumpkin	Today at 14:58	--	Folder
1.frenchmarketpumpkinsb.jpg	Today at 14:55	1.5 MB	JPEG image
2.il_570xn.842821368_e1h6.jpg.webp	Today at 14:58	113 KB	JPEG image
2.pumpkin-3f3d894.jpg	Today at 14:55	32 KB	WebP Image
3.jack-o-27-lantern-300x2661.jpg	Today at 14:58	53 KB	JPEG image
3.pumpkin-for-babies-scaled.jpg	Today at 14:55	104 KB	JPEG image
4.a-bunch-of-pumpkins-in-a-box.jpg	Today at 14:55	66 KB	JPEG image
4.jack-o-lantern-mask...pkin-mask--28787.jpg	Today at 14:55	125 KB	JPEG image
5.192085-630x630-auto.jpg	Today at 14:58	442 KB	JPEG image
5.pumpkin_113989717.jpg	Today at 14:55	71 KB	JPEG image
6.220px-balle-%c3%a0-leunettes_11.jpg	Today at 14:58	610 KB	JPEG image
7.1000_f_306418516...zcy6qqa4kuzzw93.jpg	Today at 14:58	9 KB	JPEG image

* Be mindful of image licenses

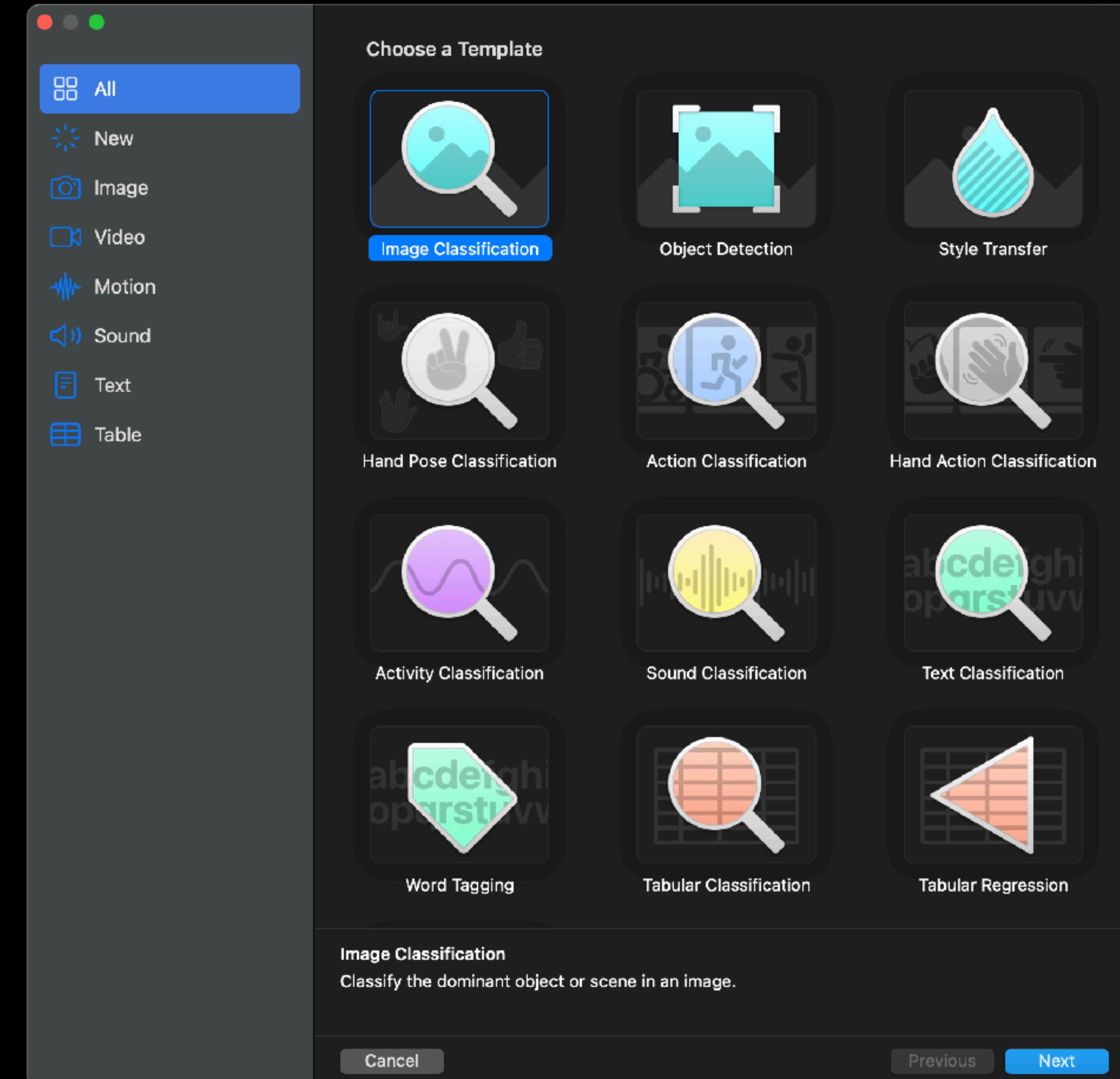
Smashing Pumpkins

Train a custom CoreML classifier

Collect images of pumpkins and
not pumpkins*

Organize the folder structure

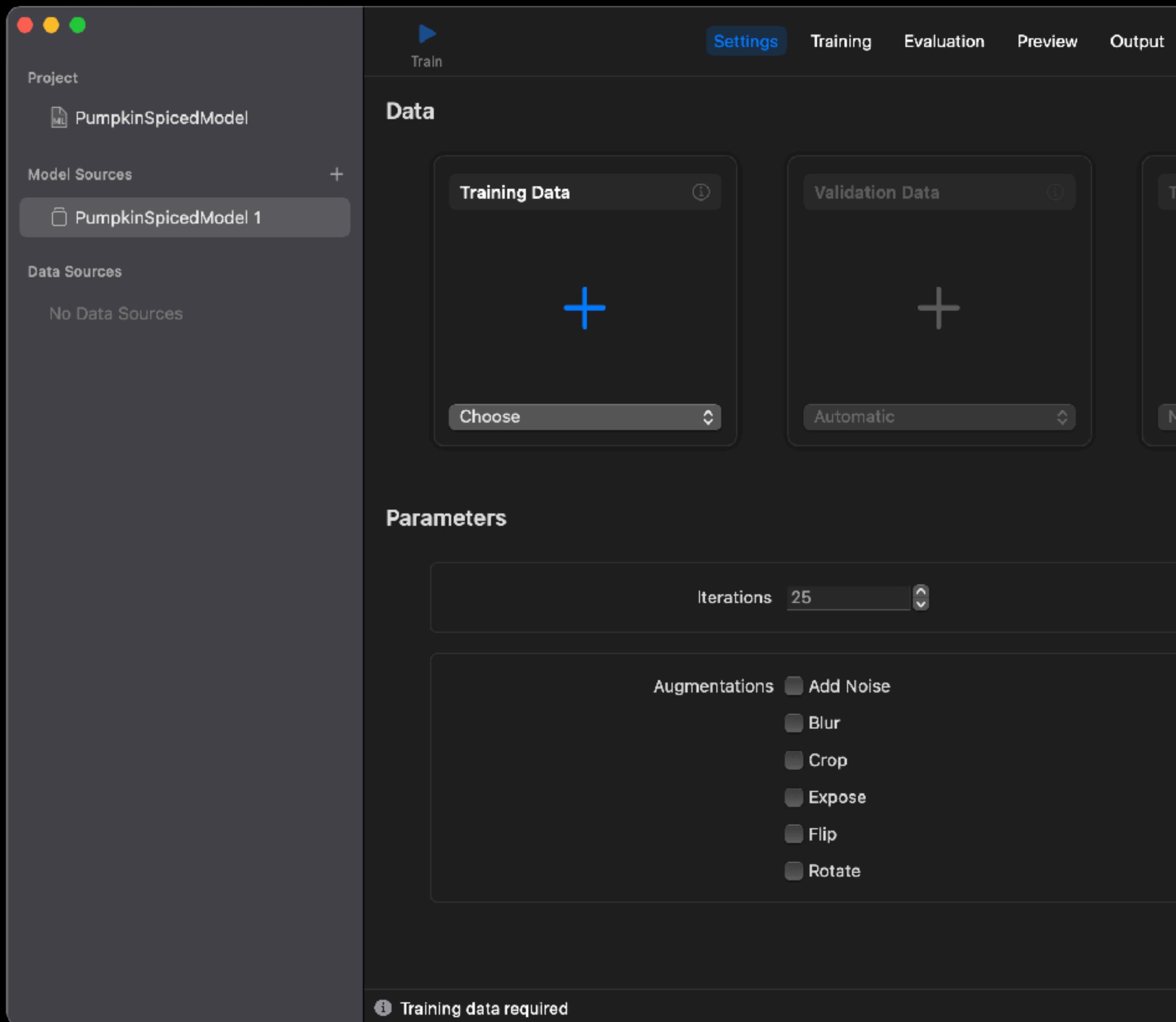
Create an image classification
project in CreateML



* Be mindful of image licenses

Smashing Pumpkins

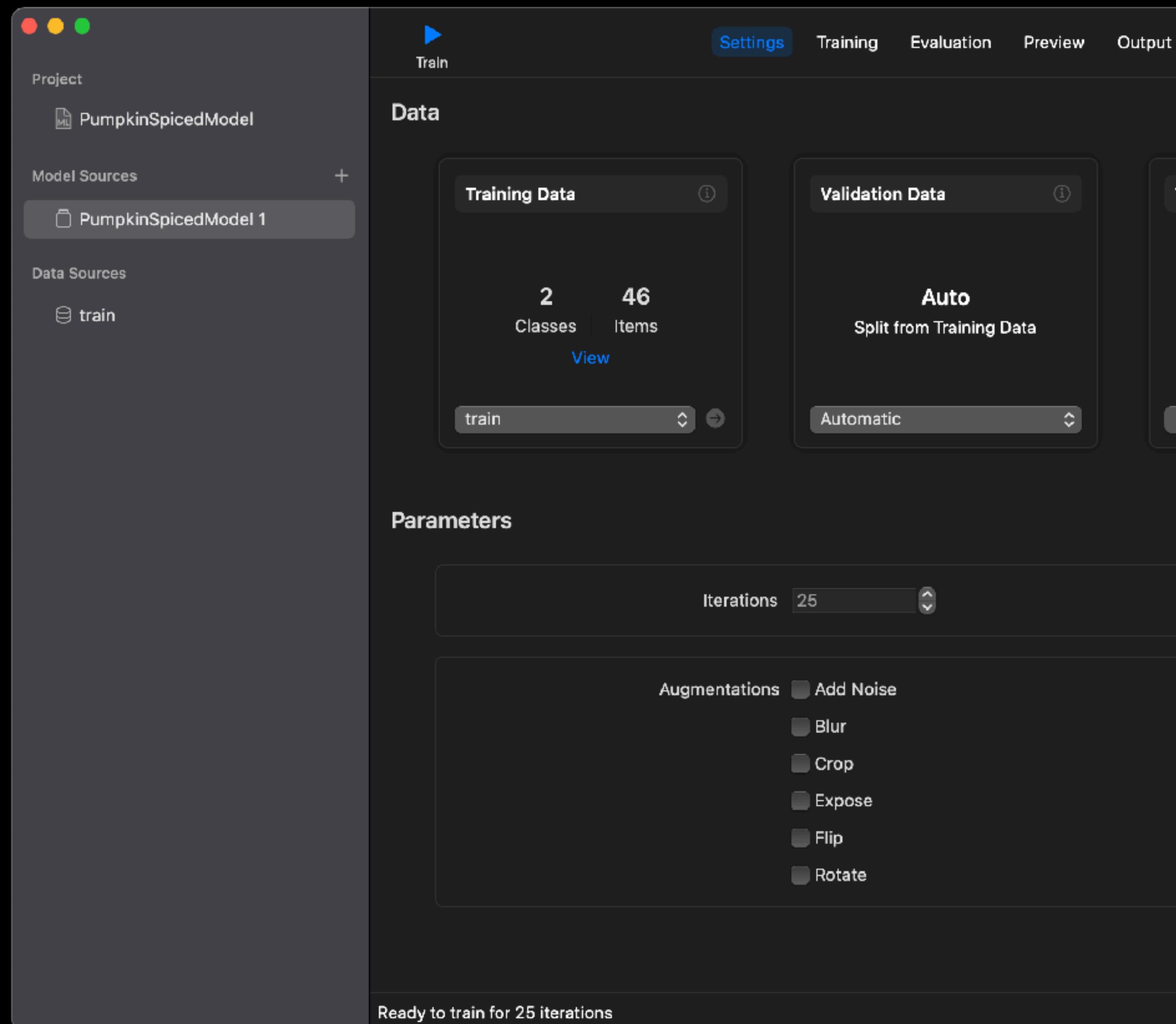
Train a custom CoreML classifier



Smashing Pumpkins

Train a custom CoreML classifier

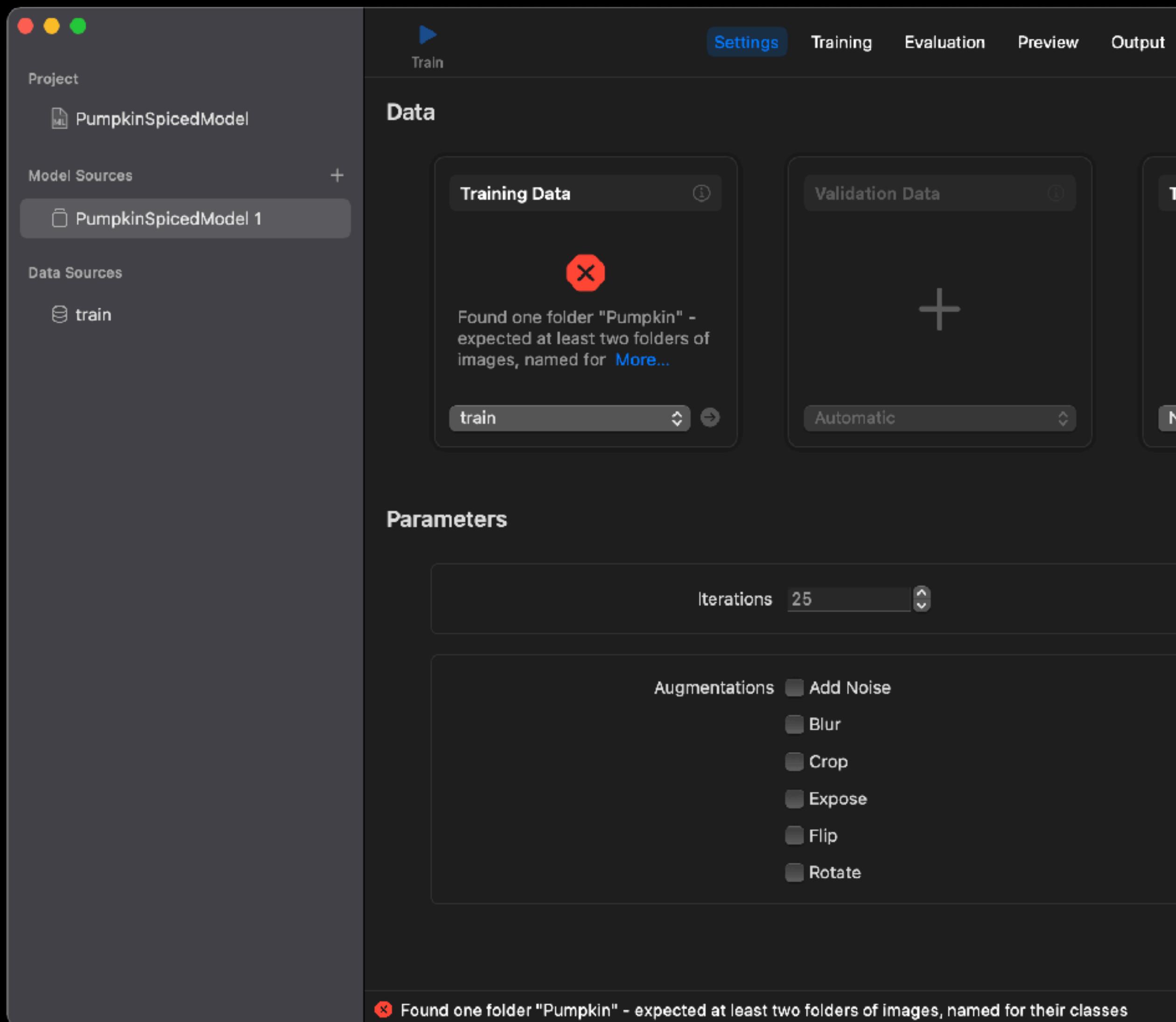
Add the training folder to your project and train



Smashing Pumpkins

Train a custom CoreML classifier

Add the training folder to your project and train

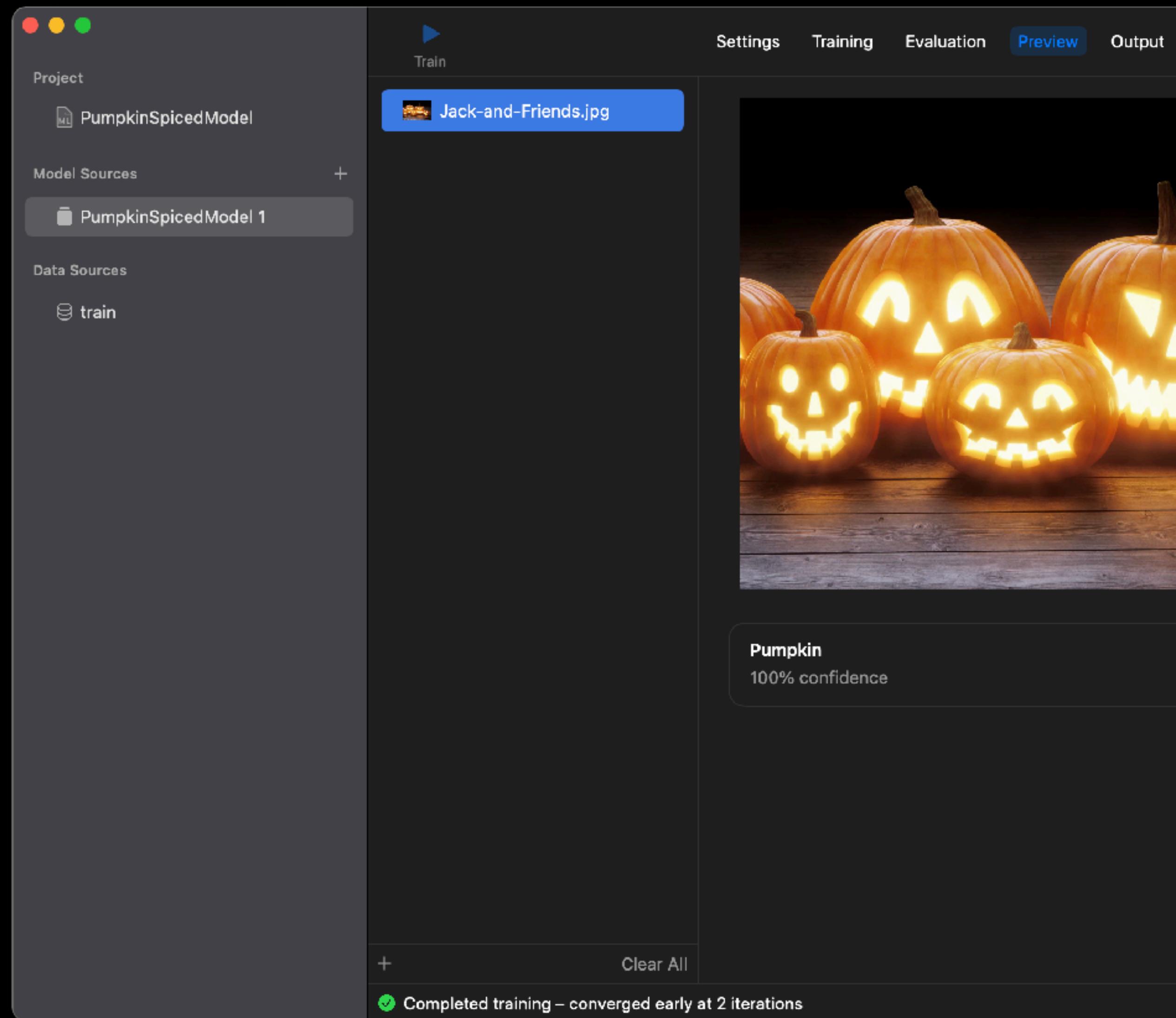


Smashing Pumpkins

Train a custom CoreML classifier

Add the training folder to your project and train

Evaluate or spot check the model

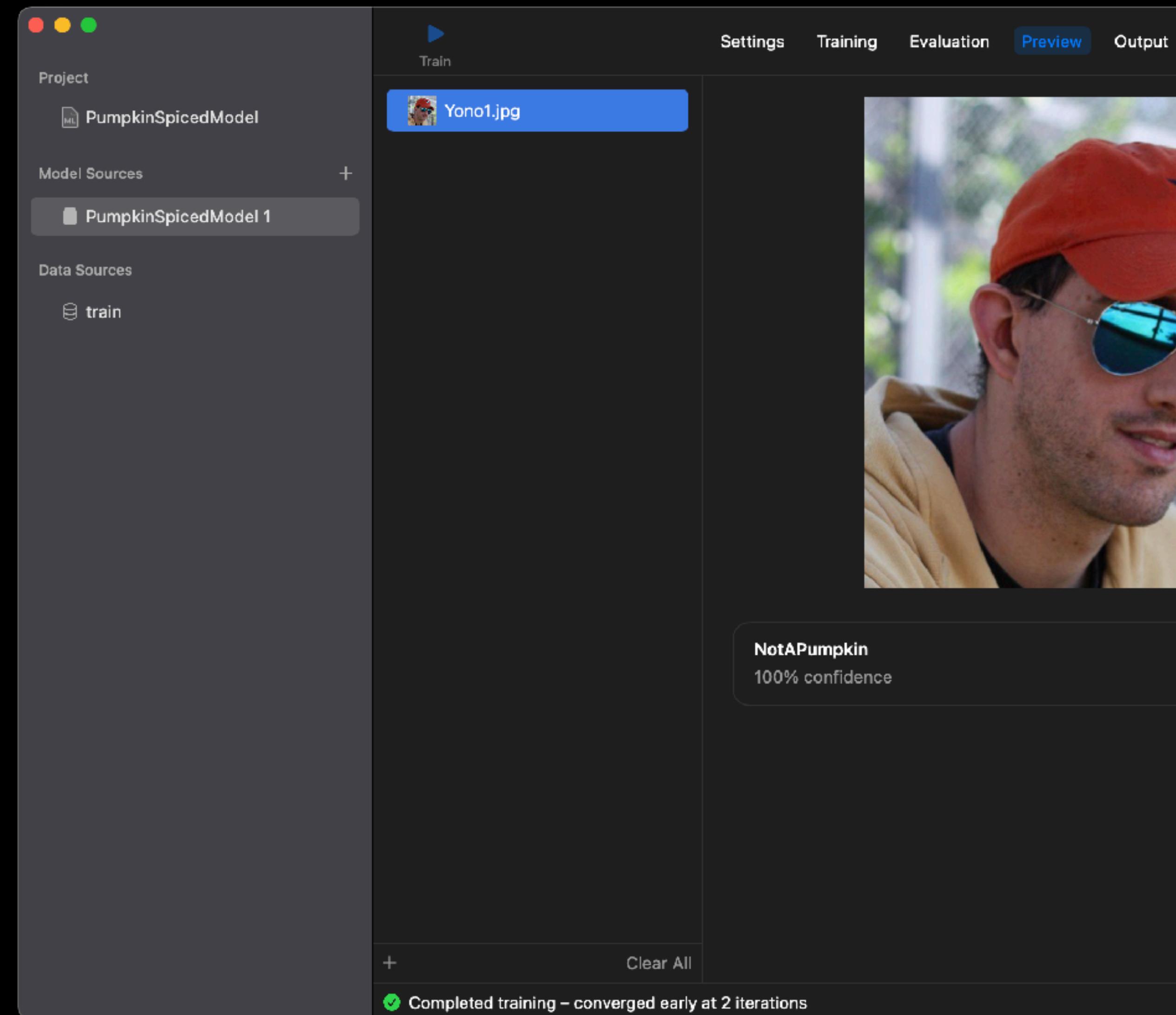


Smashing Pumpkins

Train a custom CoreML classifier

Add the training folder to your project and train

Evaluate or spot check the model



Smashing Pumpkins

Train a custom CoreML classifier

Add the training folder to your project and train

Evaluate or spot check the model

Export the CoreML model

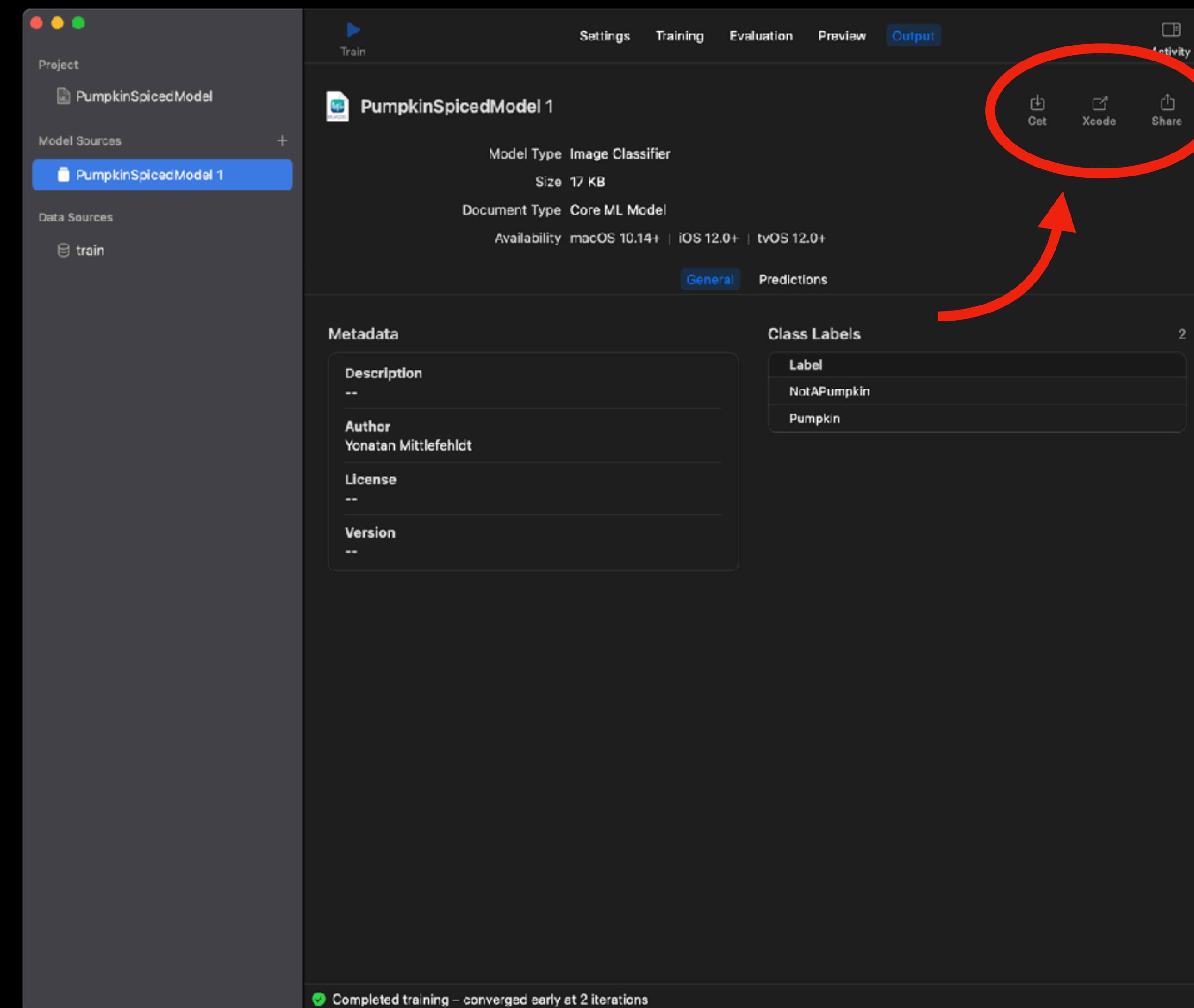


Image Classifiers

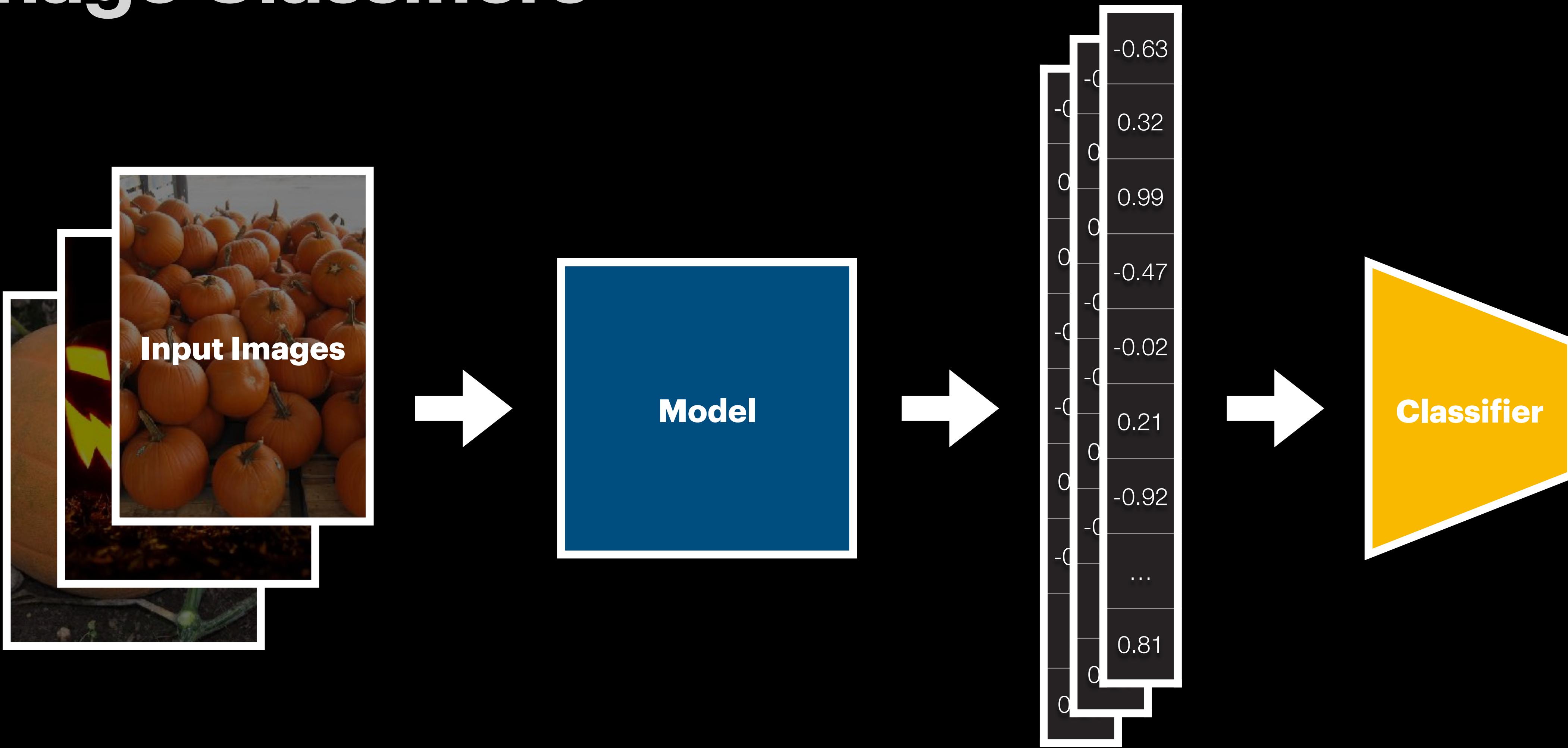
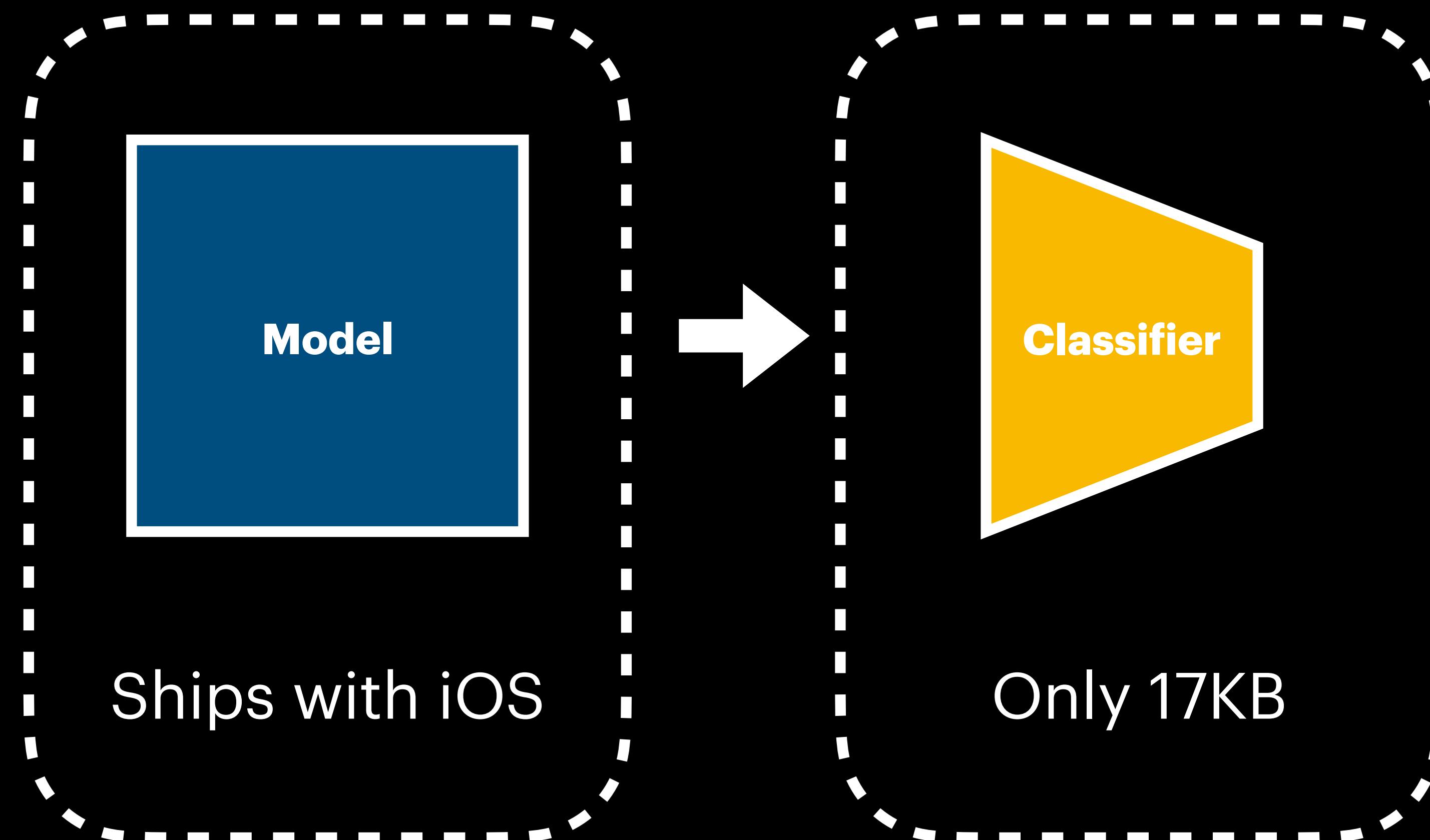


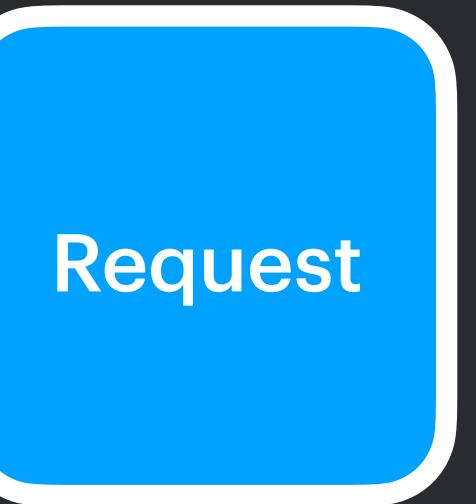
Image Classifiers



```
// Create the Vision request  
let request = VNCoreMLRequest(model: model)
```



```
// Create the Vision request  
let request = VNCoreMLRequest(model: model)  
  
// Configure the request  
request.regionOfInterest = potentialPumpkinRect
```



```
// Create the Vision request
let request = VNCoreMLRequest(model: model)

// Configure the request
request.regionOfInterest = potentialPumpkinRect

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])
```

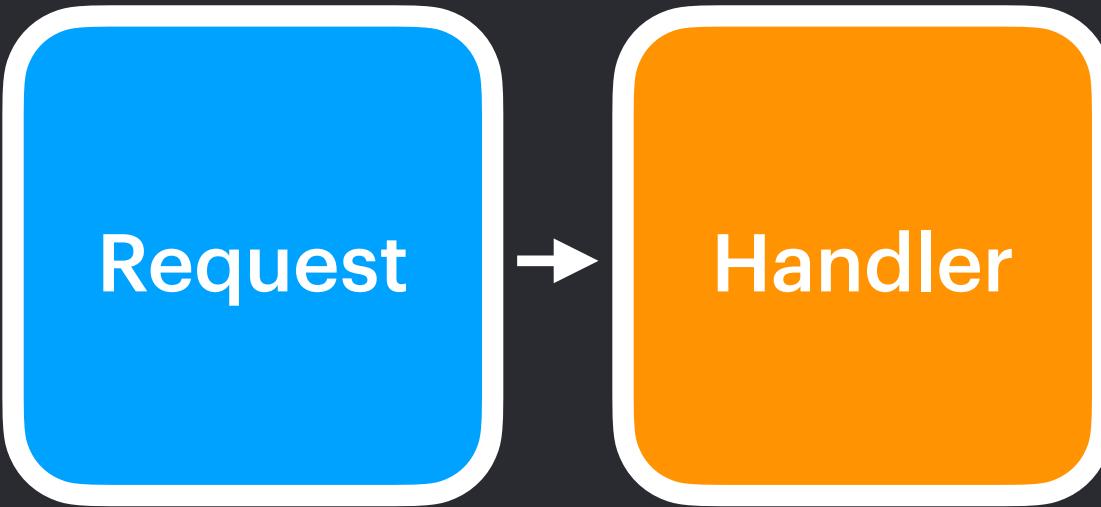


```
// Create the Vision request
let request = VNCoreMLRequest(model: model)

// Configure the request
request.regionOfInterest = potentialPumpkinRect

// Create the request handler
let requestHandler = VNIImageRequestHandler(cvPixelBuffer: image, options: [:])

// Perform the request
try? requestHandler.perform([request])
```



```
// Create the Vision request
let request = VNCoreMLRequest(model: model)

// Configure the request
request.regionOfInterest = potentialPumpkinRect

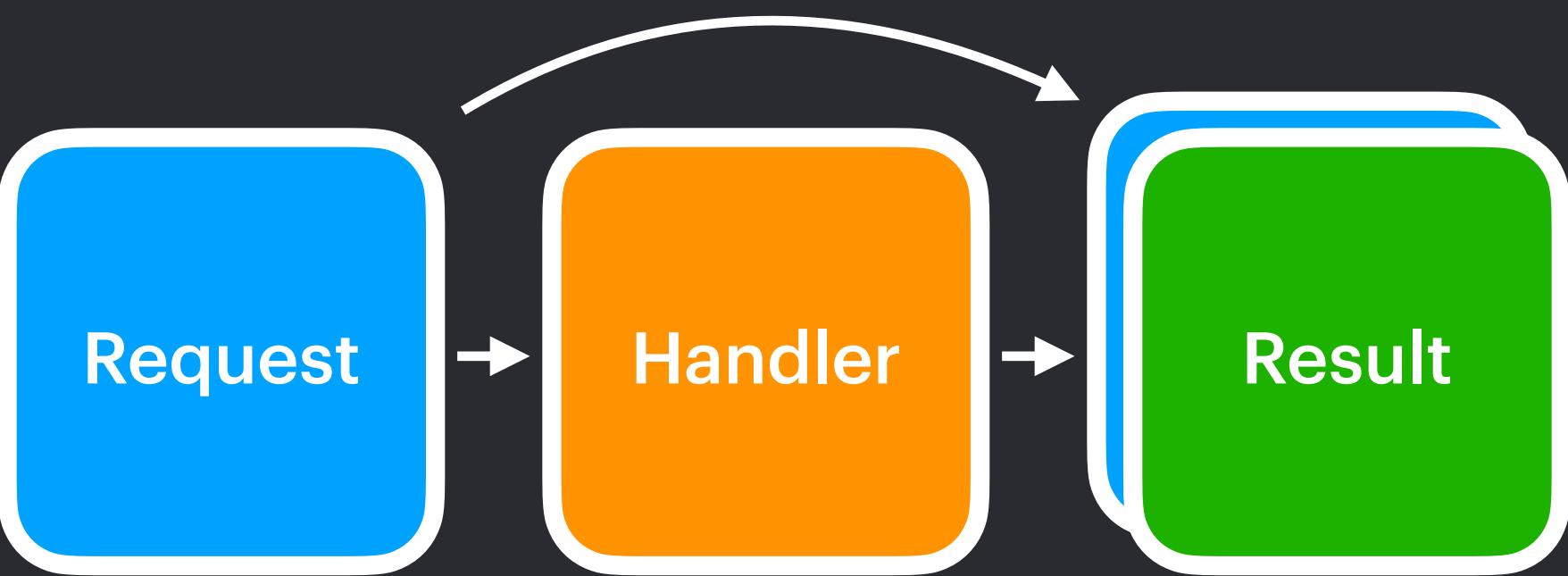
// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])

// Perform the request
try? requestHandler.perform([request])

// Check results
if let results = request.results as? [VNClassificationObservation],
    let result = results.first {

}

}
```



```
// Create the Vision request
let request = VNCoreMLRequest(model: model)

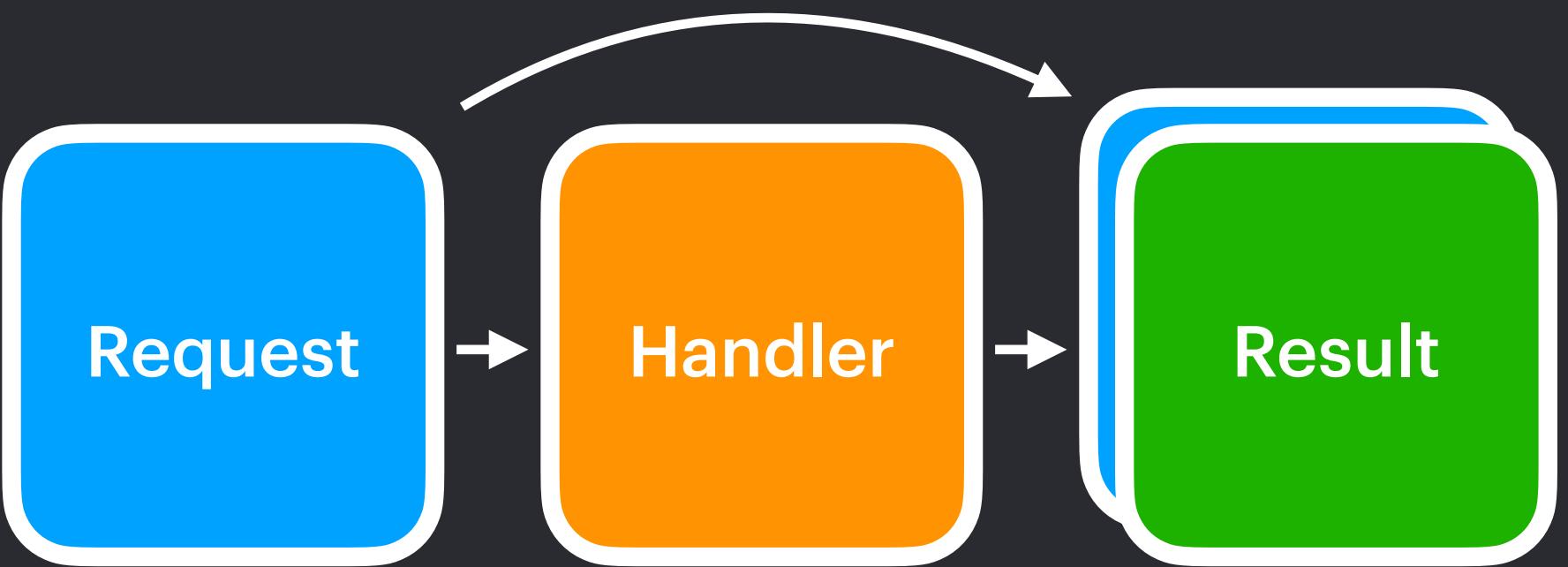
// Configure the request
request.regionOfInterest = potentialPumpkinRect

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])

// Perform the request
try? requestHandler.perform([request])

// Check results
if let results = request.results as? [VNClassificationObservation],
    let result = results.first,
    result.identifier == "Pumpkin" && result.confidence > minConfidence {

}
```



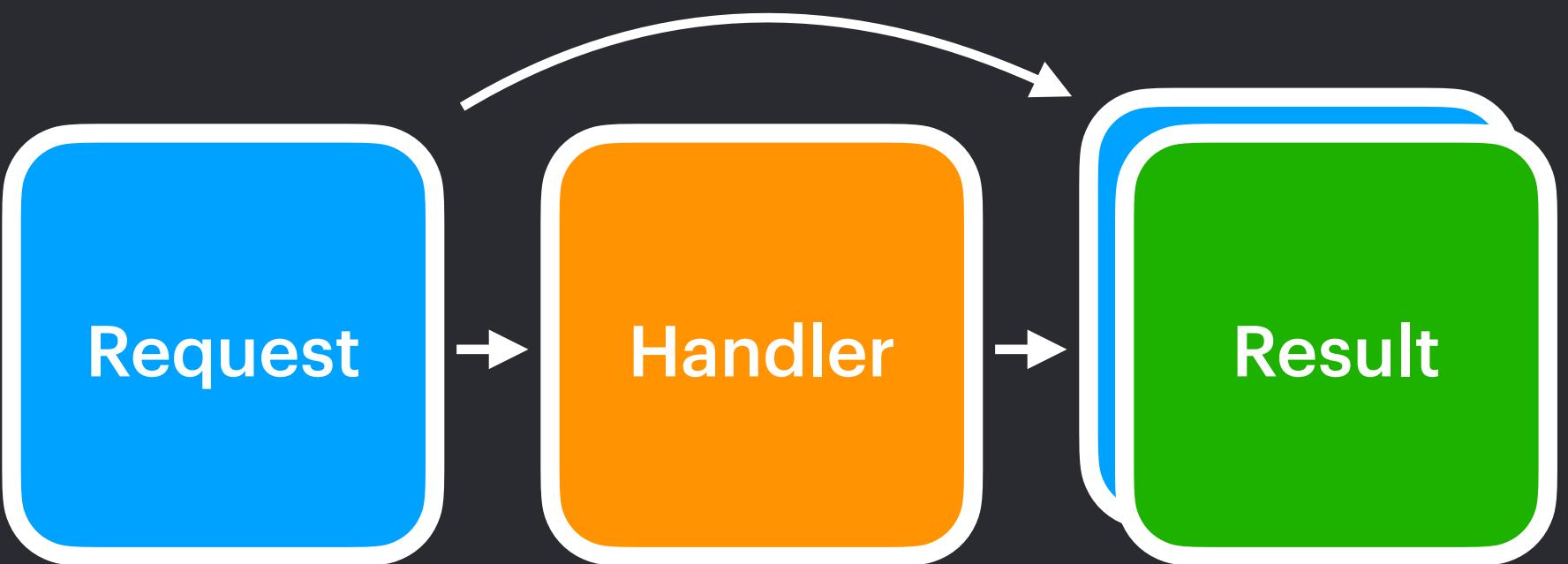
```
// Create the Vision request
let request = VNCoreMLRequest(model: model)

// Configure the request
request.regionOfInterest = potentialPumpkinRect

// Create the request handler
let requestHandler = VNImageRequestHandler(cvPixelBuffer: image, options: [:])

// Perform the request
try? requestHandler.perform([request])

// Check results
if let results = request.results as? [VNClassificationObservation],
    let result = results.first,
    result.identifier == "Pumpkin" && result.confidence > minConfidence {
    return Pumpkin(id: result.uuid, rect: potentialPumpkinRect)
}
```



Crushing Your Head

The App

Hand Pose Detector

Face Detector

Comparing Feature Vectors

Object Tracking

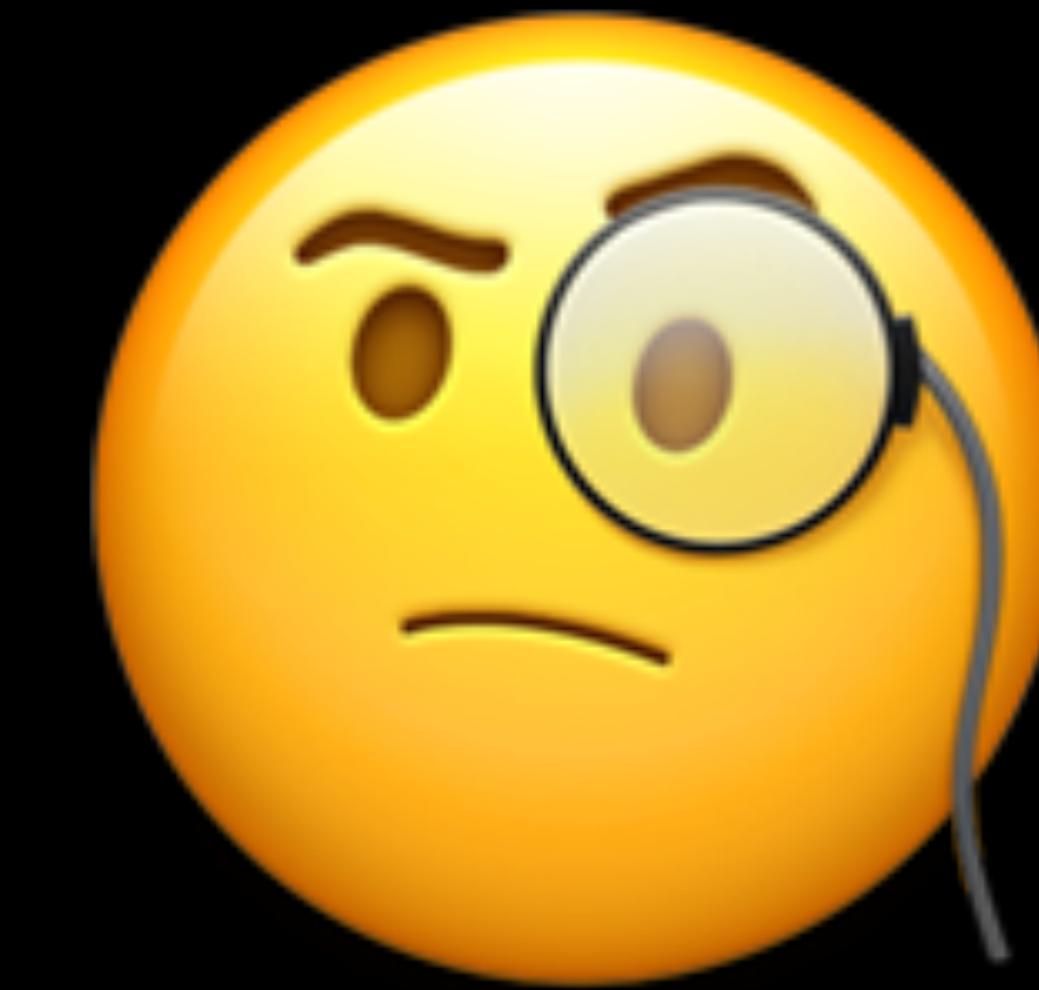
BONUS:

Custom Classifier



**Wouldn't it be great if this could be
run on a hands free device...**

...some sort of glasses maybe?



<https://github.com/yonomitt/Crushing-Your-Head>

Thank You

@yonomitt

<https://github.com/yonomitt/Crushing-Your-Head>