#WWDC18

Object Tracking in Vision

Session 716

Sergey Kamensky, Vision Team

Why Vision?

New in Vision

Vision in depth

Tracking in Vision

Why Vision?

New in Vision

Vision in depth

Tracking in Vision

Vision in a Nutshell

Vision in a Nutshell

One stop for solving computer vision problems

Vision in a Nutshell

One stop for solving computer vision problems

Simple, consistent interface

Vision in a Nutshell

One stop for solving computer vision problems

Simple, consistent interface

Runs on iOS, macOS, and tvOS

Vision in a Nutshell

One stop for solving computer vision problems

Simple, consistent interface

Runs on iOS, macOS, and tvOS

Privacy-oriented

Vision in a Nutshell

One stop for solving computer vision problems

Simple, consistent interface

Runs on iOS, macOS, and tvOS

Privacy-oriented

Continuously evolving

Vision Basics

What?

How?

Results

Vision Basics

What?	How?	Results
Request		
`VNRequest` family		

Vision Basics

What?	How?	Results
Request	Request Handler	
`VNRequest` family	`VNImageRequestHandler`, `VNSequenceRequestHandler`	

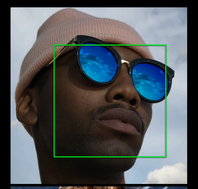
Vision Basics

What?	How?	Results
Request	Request Handler	Observations
`VNRequest` family	`VNImageRequestHandler`, `VNSequenceRequestHandler`	`VNObservation` family

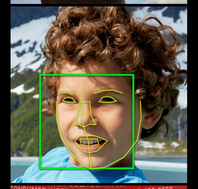
Requests

What?

VNDetectFaceRectanglesRequest



VNDetectFaceLandmarksRequest



VNDetectBarcodesRequest

VNDetectTextRectanglesRequest

VNDetectHorizonRequest

VNDetectRectanglesRequest

VNImageRegistrationRequest

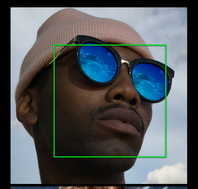
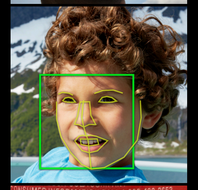
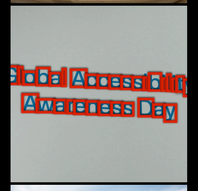
VNTrackObjectRequest

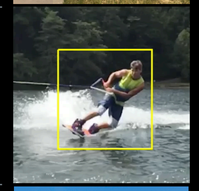
VNTrackRectangleRequest

VNCoreMLRequest

Requests

What?

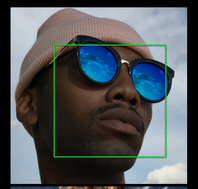
VNDetectFaceRectanglesRequest	
VNDetectFaceLandmarksRequest	
VNDetectBarcodesRequest	
VNDetectTextRectanglesRequest	
VNDetectHorizonRequest	

VNDetectRectanglesRequest	
VNImageRegistrationRequest	
VNTrackObjectRequest	
VNTrackRectangleRequest	
VNCoreMLRequest	

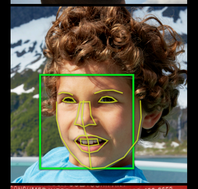
Requests

What?

VNDetectFaceRectanglesRequest



VNDetectFaceLandmarksRequest



VNDetectBarcodesRequest

VNDetectTextRectanglesRequest

VNDetectHorizonRequest

VNDetectRectanglesRequest

VNImageRegistrationRequest

VNTrackObjectRequest

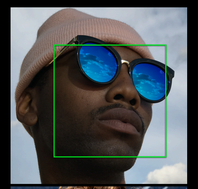
VNTrackRectangleRequest

VNCoreMLRequest

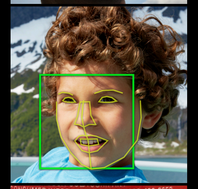
Requests

What?

VNDetectFaceRectanglesRequest



VNDetectFaceLandmarksRequest



VNDetectBarcodesRequest

VNDetectTextRectanglesRequest

VNDetectHorizonRequest

VNDetectRectanglesRequest

VNImageRegistrationRequest

VNTrackObjectRequest

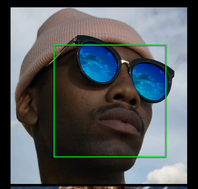
VNTrackRectangleRequest

VNCoreMLRequest

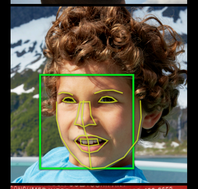
Requests

What?

VNDetectFaceRectanglesRequest



VNDetectFaceLandmarksRequest



VNDetectBarcodesRequest

VNDetectTextRectanglesRequest

VNDetectHorizonRequest

VNDetectRectanglesRequest

VNImageRegistrationRequest

VNTrackObjectRequest

VNTrackRectangleRequest

VNCoreMLRequest

Request Handlers

How?

VNImageRequestHandler

VNSequenceRequestHandler

Request Handlers

How?

VNImageRequestHandler

VNSequenceRequestHandler

Use Case

Benefits

Examples

Request Handlers

How?

VNImageRequestHandler		VNSequenceRequestHandler
Perform multiple requests on the same image	**Use Case**	
Optimized usage of image and its derivatives	**Benefits**	
Detection requests, request pipelines: Face Landmarks => Face Detection	**Examples**	

Request Handlers

How?

VNImageRequestHandler		VNSequenceRequestHandler
Perform multiple requests on the same image	**Use Case**	Process request(s) on a sequence of images
Optimized usage of image and its derivatives	**Benefits**	Caches frame-to-frame state
Detection requests, request pipelines: Face Landmarks => Face Detection	**Examples**	Tracking, Image Registration

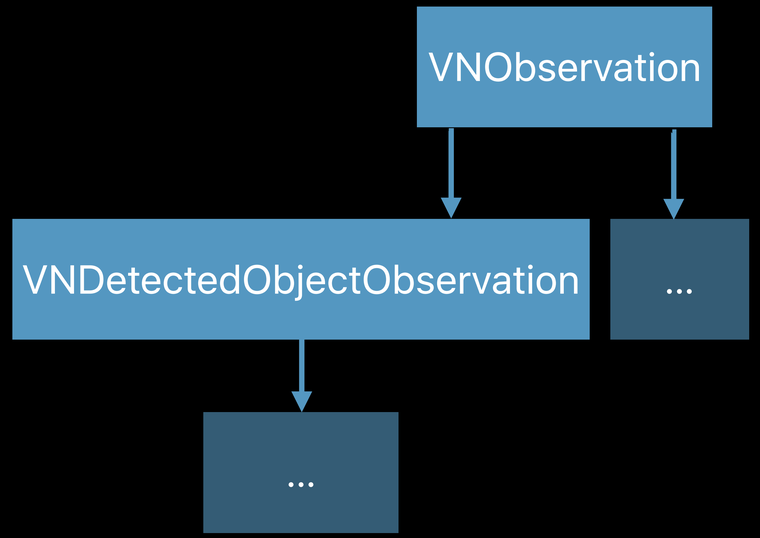
Observations

Results

Observations

Results

Family of classes derived from `VNObservation`

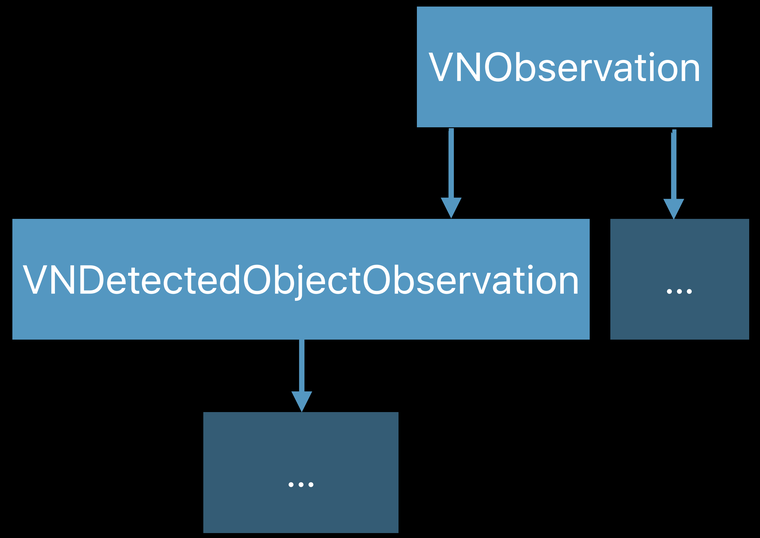


Observations

Results

Family of classes derived from `VNObservation`

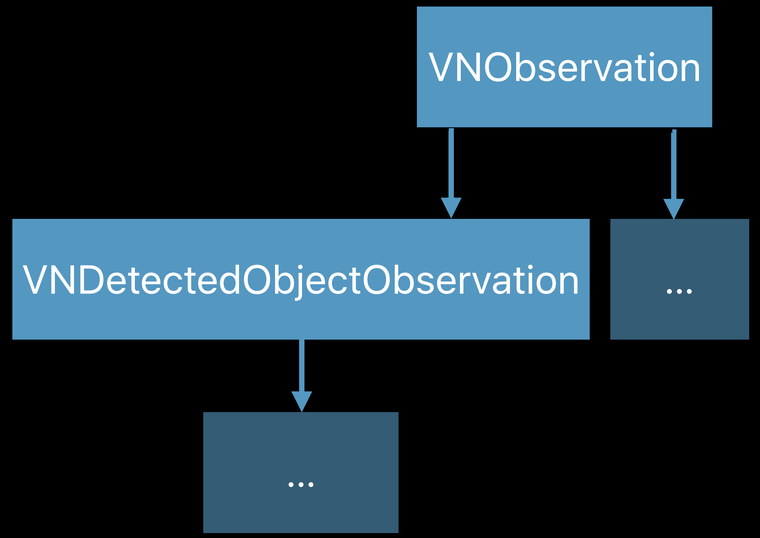
How to obtain a `VNObservation?`



Observations

Results

Family of classes derived from `VNObservation`



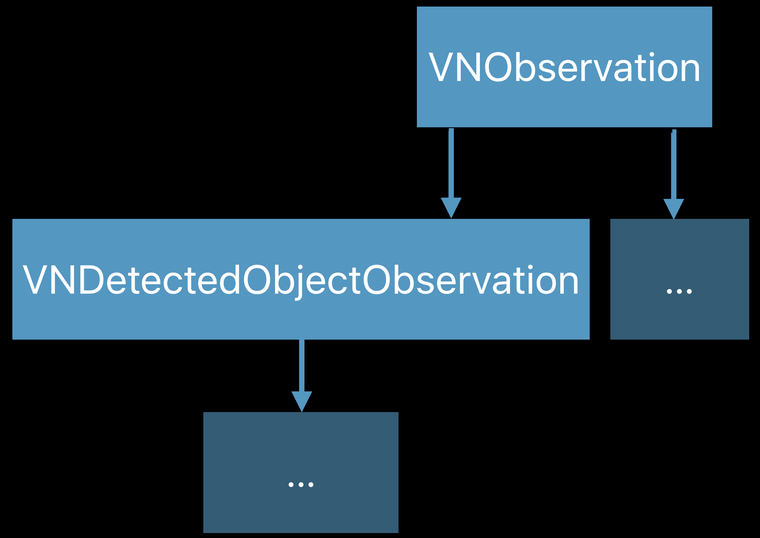
How to obtain a `VNObservation?`

- Returned in `VNRequest results` property

Observations

Results

Family of classes derived from `VNObservation`



How to obtain a `VNObservation?`

- Returned in `VNRequest results` property
- Can be manually created

Why Vision?

New in Vision

Vision in depth

Tracking in Vision

New Face Detector

NEW

New Face Detector

NEW

Finds more faces

New Face Detector

NEW

Finds more faces

Now orientation-agnostic

New Face Detector

NEW

Finds more faces

Now orientation-agnostic

Rev 1

New Face Detector

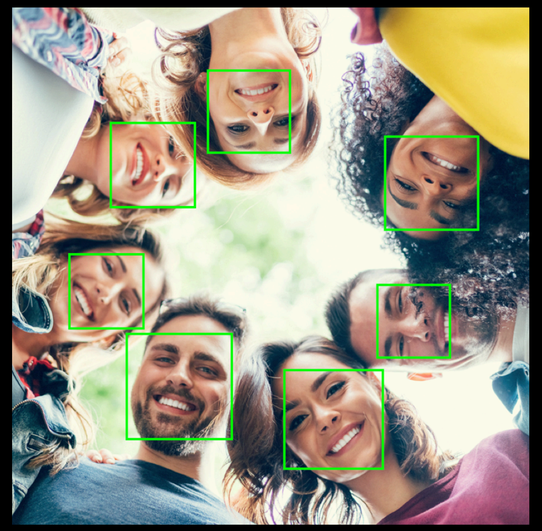
NEW

Finds more faces

Now orientation-agnostic

Rev 1

Rev 2



New Face Detector

NEW

New Face Detector

NEW

Same API: `VNDetectFaceRectanglesRequest`

New Face Detector

NEW

Same API: `VNDetectFaceRectanglesRequest`

`VNDetectFaceRectanglesRequestRevision2`

New Face Detector

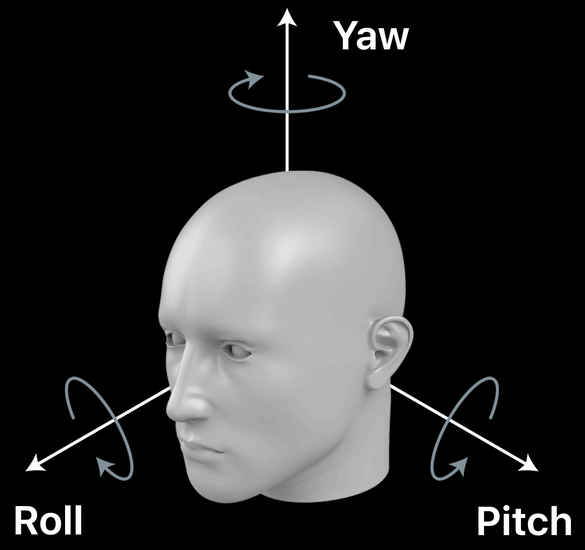
NEW

Same API: `VNDetectFaceRectanglesRequest`

`VNDetectFaceRectanglesRequestRevision2`

`VNFaceObservation` has two new properties:

```
open class VNFaceObservation : VNDetectedObjectObservation {
…
open var roll: NSNumber? { get }
open var yaw: NSNumber? { get }
…
}
```



Request Revisioning

NEW

Request Revisioning

NEW

Vision Requests now support revisioning

Request Revisioning

NEW

Vision Requests now support revisioning

Default	vs.	Explicit
Use latest revision from SDK your app is linked against		Programmatically specified

Request Revisioning

NEW

Vision Requests now support revisioning

Default	vs.	**Explicit**
Use latest revision from SDK your app is linked against		Programmatically specified

Future-proof your app—Error for unavailable functionality

Why Vision?

New in Vision

Vision in depth

Tracking in Vision

Image Request Handler

Image Request Handler

Used to process one or more requests on the same image

Optimizes performance by caching image derivatives and request results

Image Request Handler

Used to process one or more requests on the same image

Optimizes performance by caching image derivatives and request results

```
let detectFacesRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try requestHandler.perform([detectFacesRequest])
let fo = detectFacesRequest.results!.first! as! VNFaceObservation
```

Image Request Handler

Used to process one or more requests on the same image

Optimizes performance by caching image derivatives and request results

```
let detectFacesRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try requestHandler.perform([detectFacesRequest])
let fo = detectFacesRequest.results!.first! as! VNFaceObservation
```

Image Request Handler

Used to process one or more requests on the same image

Optimizes performance by caching image derivatives and request results

```
let detectFacesRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try requestHandler.perform([detectFacesRequest])
let fo = detectFacesRequest.results!.first! as! VNFaceObservation
```

Image Request Handler

Used to process one or more requests on the same image

Optimizes performance by caching image derivatives and request results

```
let detectFacesRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try requestHandler.perform([detectFacesRequest])
let fo = detectFacesRequest.results!.first! as! VNFaceObservation
```

Image Request Handler

Used to process one or more requests on the same image

Optimizes performance by caching image derivatives and request results

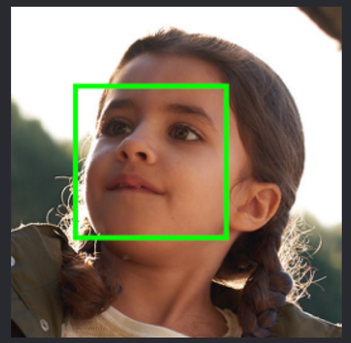
```
let detectFacesRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try requestHandler.perform([detectFacesRequest])
let fo = detectFacesRequest.results!.first! as! VNFaceObservation
```

Image Request Handler

Used to process one or more requests on the same image

Optimizes performance by caching image derivatives and request results

```
let detectFacesRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try requestHandler.perform([detectFacesRequest])
let fo = detectFacesRequest.results!.first! as! VNFaceObservation
```



Sequence Request Handler

Sequence Request Handler

Processes request(s) on the sequence of images

Used to process two types of requests—Tracking and Image Registration

Sequence Request Handler

Processes request(s) on the sequence of images

Used to process two types of requests—Tracking and Image Registration

```
let requestHandler = VNSequenceRequestHandler()
var inputObservation = VNDetectedObjectObservation(boundingBox: objectBoundingBox)

for _ in 1...5 {
   let frame = frameFeeder.nextFrame()
   let request = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
   try requestHandler.perform([request], on: frame)
   let observation = request.results!.first! as! VNDetectedObjectObservation
   inputObservation = observation
}
```

Sequence Request Handler

Processes request(s) on the sequence of images

Used to process two types of requests—Tracking and Image Registration

```
let requestHandler = VNSequenceRequestHandler()
var inputObservation = VNDetectedObjectObservation(boundingBox: objectBoundingBox)

for _ in 1...5 {
    let frame = frameFeeder.nextFrame()
    let request = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
    try requestHandler.perform([request], on: frame)
    let observation = request.results!.first! as! VNDetectedObjectObservation
    inputObservation = observation
}
```

Sequence Request Handler

Processes request(s) on the sequence of images

Used to process two types of requests—Tracking and Image Registration

```
let requestHandler = VNSequenceRequestHandler()
var inputObservation = VNDetectedObjectObservation(boundingBox: objectBoundingBox)

for _ in 1...5 {
    let frame = frameFeeder.nextFrame()
    let request = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
    try requestHandler.perform([request], on: frame)
    let observation = request.results!.first! as! VNDetectedObjectObservation
    inputObservation = observation
}
```

Sequence Request Handler

Processes request(s) on the sequence of images

Used to process two types of requests—Tracking and Image Registration

```
let requestHandler = VNSequenceRequestHandler()
var inputObservation = VNDetectedObjectObservation(boundingBox: objectBoundingBox)

for _ in 1...5 {
    let frame = frameFeeder.nextFrame()
    let request = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
    try requestHandler.perform([request], on: frame)
    let observation = request.results!.first! as! VNDetectedObjectObservation
    inputObservation = observation
}
```

Sequence Request Handler

Processes request(s) on the sequence of images

Used to process two types of requests—Tracking and Image Registration

```
let requestHandler = VNSequenceRequestHandler()
var inputObservation = VNDetectedObjectObservation(boundingBox: objectBoundingBox)

for _ in 1...5 {
    let frame = frameFeeder.nextFrame()
    let request = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
    try requestHandler.perform([request], on: frame)
    let observation = request.results!.first! as! VNDetectedObjectObservation
    inputObservation = observation
}
```

Sequence Request Handler

Processes request(s) on the sequence of images

Used to process two types of requests—Tracking and Image Registration

```
let requestHandler = VNSequenceRequestHandler()
var inputObservation = VNDetectedObjectObservation(boundingBox: objectBoundingBox)

for _ in 1...5 {
    let frame = frameFeeder.nextFrame()
    let request = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
    try requestHandler.perform([request], on: frame)
    let observation = request.results!.first! as! VNDetectedObjectObservation
    inputObservation = observation
}
```

Sequence Request Handler

Processes request(s) on the sequence of images

Used to process two types of requests—Tracking and Image Registration

```
let requestHandler = VNSequenceRequestHandler()
var inputObservation = VNDetectedObjectObservation(boundingBox: objectBoundingBox)

for _ in 1...5 {
    let frame = frameFeeder.nextFrame()
    let request = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
    try requestHandler.perform([request], on: frame)
    let observation = request.results!.first! as! VNDetectedObjectObservation
    inputObservation = observation
}
```

Sequence Request Handler

Processes request(s) on the sequence of images

Used to process two types of requests—Tracking and Image Registration

```
let requestHandler = VNSequenceRequestHandler()
var inputObservation = VNDetectedObjectObservation(boundingBox: objectBoundingBox)

for _ in 1...5 {
    let frame = frameFeeder.nextFrame()
    let request = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
    try requestHandler.perform([request], on: frame)
    let observation = request.results!.first! as! VNDetectedObjectObservation
    inputObservation = observation
}
```

Sequence Request Handler

Processes request(s) on the sequence of images

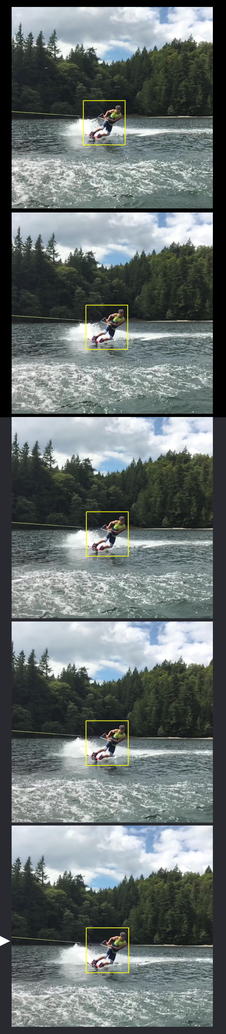
Used to process two types of requests—Tracking and Image Registration

```
let requestHandler = VNSequenceRequestHandler()
var inputObservation = VNDetectedObjectObservation(boundingBox: objectBoundingBox)

for _ in 1...5 {
    let frame = frameFeeder.nextFrame()
    let request = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
    try requestHandler.perform([request], on: frame)
    let observation = request.results!.first! as! VNDetectedObjectObservation
    inputObservation = observation
}
```

Sequence Request Handler

Processes request(s) on the sequence of images

Used to process two types of requests—Tracking and Image Registration



```
let requestHandler = VNSequenceRequestHandler()
var inputObservation = VNDetectedObjectObservation(boundingBox: objectBoundingBox)

for _ in 1...5 {
    let frame = frameFeeder.nextFrame()
    let request = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
    try requestHandler.perform([request], on: frame)
    let observation = request.results!.first! as! VNDetectedObjectObservation
    inputObservation = observation
}
```

VNRequest Initialization

Mandatory versus optional properties

VNRequest Initialization

Mandatory versus optional properties

Mandatory—Must be provided via initializer, overriding is OK

VNRequest Initialization

Mandatory versus optional properties

Mandatory—Must be provided via initializer, overriding is OK

```
// Create request with a mandatory property
let boundingBox = CGRect(x: 0.2, y: 0.3, width: 0.15, height: 0.15)
let inputObservation = VNDetectedObjectObservation(boundingBox: boundingBox)
let trackingRequest = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
```

VNRequest Initialization

Mandatory versus optional properties

Mandatory—Must be provided via initializer, overriding is OK

```
// Create request with a mandatory property
let boundingBox = CGRect(x: 0.2, y: 0.3, width: 0.15, height: 0.15)
let inputObservation = VNDetectedObjectObservation(boundingBox: boundingBox)
let trackingRequest = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
```

Optional—Initialized to default value, overriding is OK

VNRequest Initialization

Mandatory versus optional properties

Mandatory—Must be provided via initializer, overriding is OK

```
// Create request with a mandatory property
let boundingBox = CGRect(x: 0.2, y: 0.3, width: 0.15, height: 0.15)
let inputObservation = VNDetectedObjectObservation(boundingBox: boundingBox)
let trackingRequest = VNTrackObjectRequest(detectedObjectObservation: inputObservation)
```

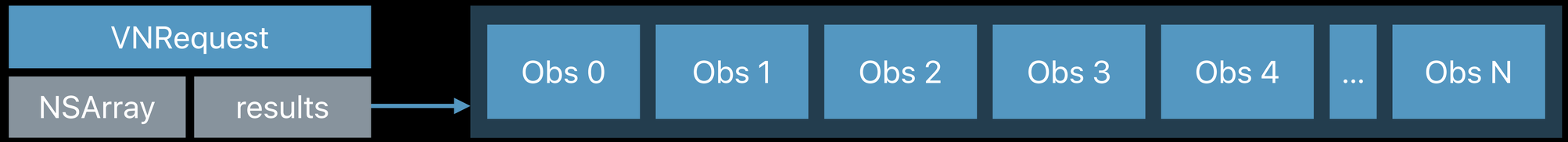
Optional—Initialized to default value, overriding is OK

```
// Create request and override an optional property
let request = VNDetectBarcodesRequest()
let roi = CGRect(x: 0.35, y: 0.35, width: 0.3, height: 0.3)
request.regionOfInterest = roi // default - entire image
```

Understanding Results

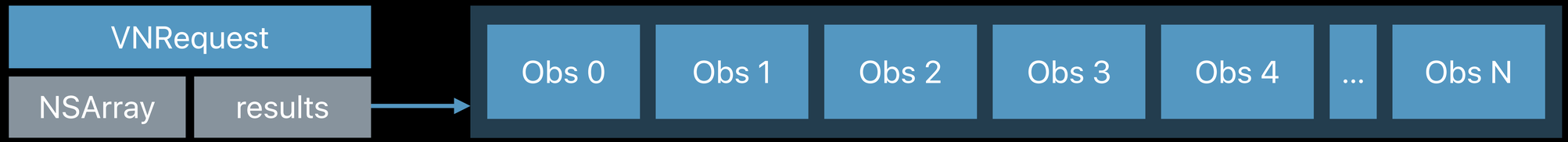
Understanding Results

Collection of `VNObservation` objects in `VNRequest results` property



Understanding Results

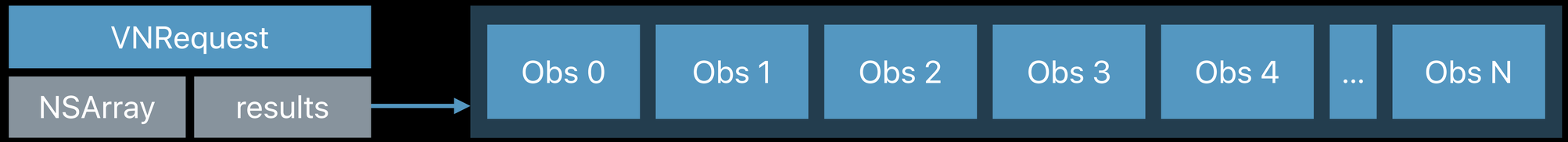
Collection of `VNObservation` objects in `VNRequest results` property



The number of observations is from 0 to N

Understanding Results

Collection of `VNObservation` objects in `VNRequest results` property

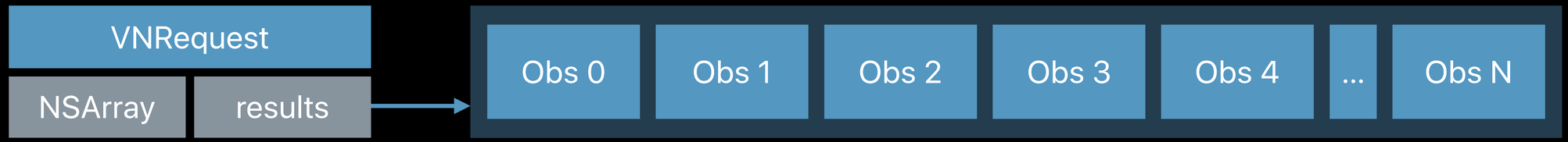


The number of observations is from 0 to N

`VNObservation` is immutable

Understanding Results

Collection of `VNObservation` objects in `VNRequest results` property



The number of observations is from 0 to N

`VNObservation` is immutable

Important common observation properties:

- `uuid`—Is used to match related results
- `confidence`—Shows quality of returned results

Request Pipelines

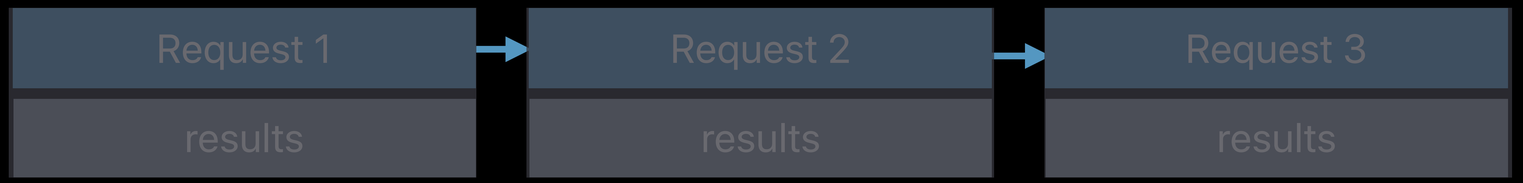
Request Pipelines

Pipeline—requests are executed to fulfill dependencies

Request Pipelines

Pipeline—requests are executed to fulfill dependencies

How is the pipeline executed?



Request Pipelines

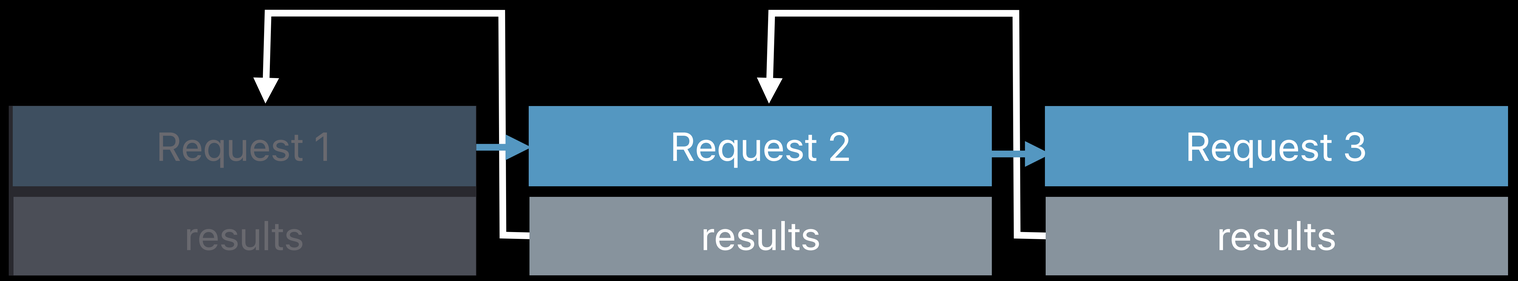
Pipeline—requests are executed to fulfill dependencies

How is the pipeline executed?

Request Pipelines

Pipeline—requests are executed to fulfill dependencies

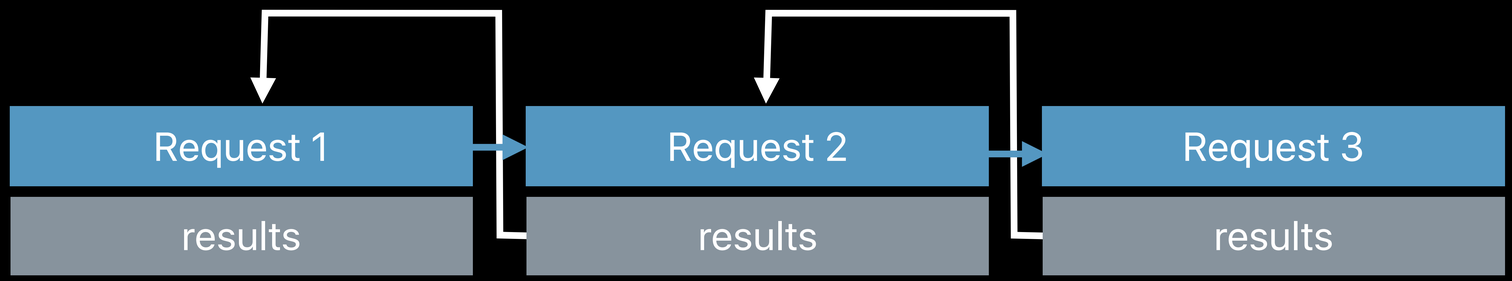
How is the pipeline executed?



Request Pipelines

Pipeline—requests are executed to fulfill dependencies

How is the pipeline executed?



```
//Request Pipelines, Implicit execution
```

```
let landmarksRequest = VNDetectFaceLandmarksRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([request])
let fo = landmarksRequest.results!.first as! VNFaceObservation
```

```
//Request Pipelines, Implicit execution
```

```
let landmarksRequest = VNDetectFaceLandmarksRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([request])
let fo = landmarksRequest.results!.first as! VNFaceObservation
```

```
//Request Pipelines, Implicit execution
```

```
let landmarksRequest = VNDetectFaceLandmarksRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([request])
let fo = landmarksRequest.results!.first as! VNFaceObservation
```

```
//Request Pipelines, Implicit execution
```

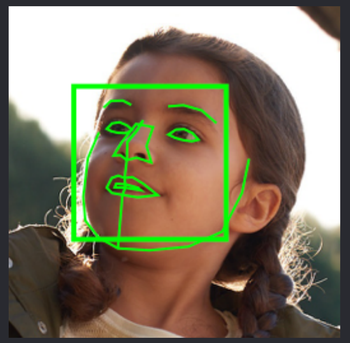
```
let landmarksRequest = VNDetectFaceLandmarksRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([request])
let fo = landmarksRequest.results!.first as! VNFaceObservation
```

```
//Request Pipelines, Implicit execution
```

```
let landmarksRequest = VNDetectFaceLandmarksRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([request])
let fo = landmarksRequest.results!.first as! VNFaceObservation
```

```
//Request Pipelines, Implicit execution
```

```
let landmarksRequest = VNDetectFaceLandmarksRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([request])
let fo = landmarksRequest.results!.first as! VNFaceObservation ↦
```

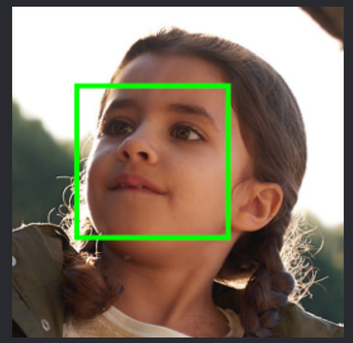


uuid	1234
boundingBox	[0.2, 0.3, 0.4, 0.4]
landmarks	0x97532468

```
//Request Pipelines, Explicit execution



let faceRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([faceRequest])
let fo = faceRequest.results!.first as! VNFaceObservation

let landmarksRequest = VNDetectFaceLandmarksRequest()
landmarksRequest.inputFaceObservations = [fo]
try imageRequestHandler.perform([landmarksRequest])
let lmo = landmarksRequest.results!.first as! VNFaceObservation
```

```
//Request Pipelines, Explicit execution
```

```
let faceRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([faceRequest])
let fo = faceRequest.results!.first as! VNFaceObservation
```

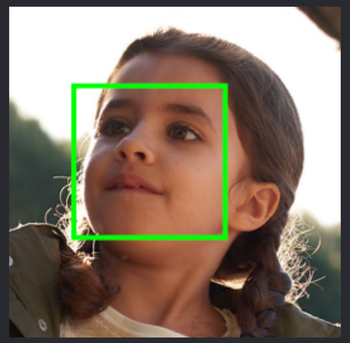


uuid	5678
boundingBox	[0.2, 0.3, 0.4, 0.4]
landmarks	nil

```
let landmarksRequest = VNDetectFaceLandmarksRequest()
landmarksRequest.inputFaceObservations = [fo]
try imageRequestHandler.perform([landmarksRequest])
let lmo = landmarksRequest.results!.first as! VNFaceObservation
```

```
//Request Pipelines, Explicit execution
```

```
let faceRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([faceRequest])
let fo = faceRequest.results!.first as! VNFaceObservation
```

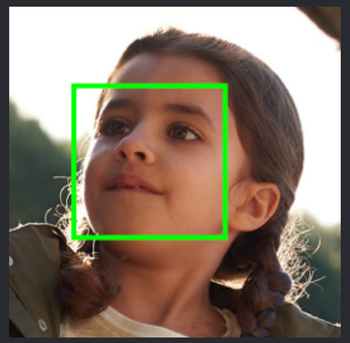


uuid	5678
boundingBox	[0.2, 0.3, 0.4, 0.4]
landmarks	nil

```
let landmarksRequest = VNDetectFaceLandmarksRequest()
landmarksRequest.inputFaceObservations = [fo]
try imageRequestHandler.perform([landmarksRequest])
let lmo = landmarksRequest.results!.first as! VNFaceObservation
```

```
//Request Pipelines, Explicit execution
```

```
let faceRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([faceRequest])
let fo = faceRequest.results!.first as! VNFaceObservation
```

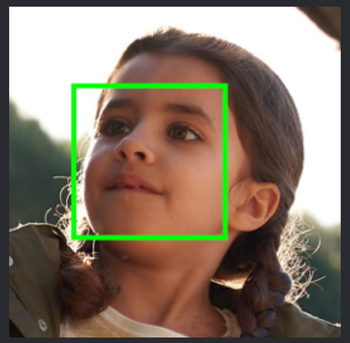


uuid	5678
boundingBox	[0.2, 0.3, 0.4, 0.4]
landmarks	nil

```
let landmarksRequest = VNDetectFaceLandmarksRequest()
landmarksRequest.inputFaceObservations = [fo]
try imageRequestHandler.perform([landmarksRequest])
let lmo = landmarksRequest.results!.first as! VNFaceObservation
```

```
//Request Pipelines, Explicit execution
```

```
let faceRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([faceRequest])
let fo = faceRequest.results!.first as! VNFaceObservation
```

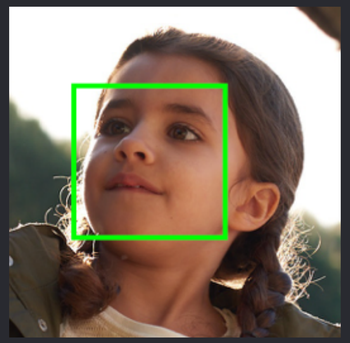


uuid	5678
boundingBox	[0.2, 0.3, 0.4, 0.4]
landmarks	nil

```
let landmarksRequest = VNDetectFaceLandmarksRequest()
landmarksRequest.inputFaceObservations = [fo]
try imageRequestHandler.perform([landmarksRequest])
let lmo = landmarksRequest.results!.first as! VNFaceObservation
```

```
//Request Pipelines, Explicit execution
```

```
let faceRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([faceRequest])
let fo = faceRequest.results!.first as! VNFaceObservation
```

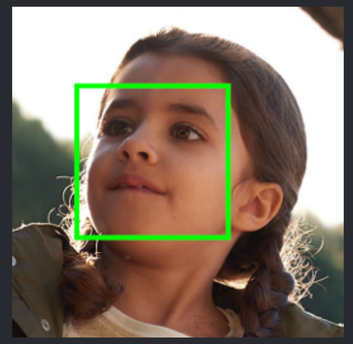


uuid	5678
boundingBox	[0.2, 0.3, 0.4, 0.4]
landmarks	nil

```
let landmarksRequest = VNDetectFaceLandmarksRequest()
landmarksRequest.inputFaceObservations = [fo]
try imageRequestHandler.perform([landmarksRequest])
let lmo = landmarksRequest.results!.first as! VNFaceObservation
```

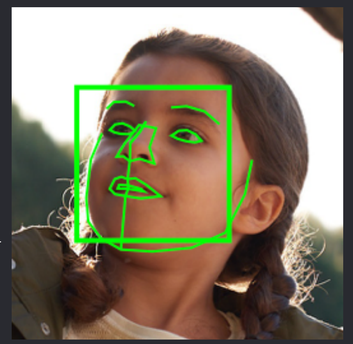
```
//Request Pipelines, Explicit execution
```

```
let faceRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([faceRequest])
let fo = faceRequest.results!.first as! VNFaceObservation
```



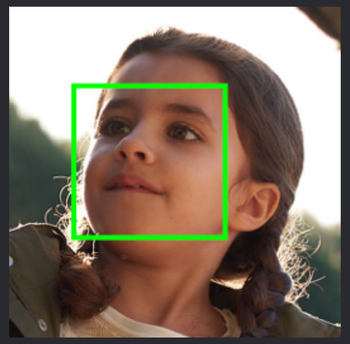
uuid	5678
boundingBox	[0.2, 0.3, 0.4, 0.4]
landmarks	nil

```
let landmarksRequest = VNDetectFaceLandmarksRequest()
landmarksRequest.inputFaceObservations = [fo]
try imageRequestHandler.perform([landmarksRequest])
let lmo = landmarksRequest.results!.first as! VNFaceObservation
```



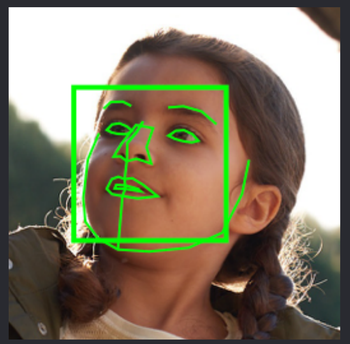
```
//Request Pipelines, Explicit execution
```

```
let faceRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([faceRequest])
let fo = faceRequest.results!.first as! VNFaceObservation
```



uuid	5678
boundingBox	[0.2, 0.3, 0.4, 0.4]
landmarks	nil

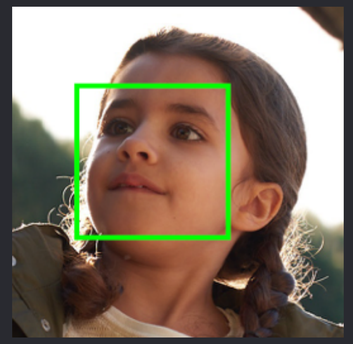
```
let landmarksRequest = VNDetectFaceLandmarksRequest()
landmarksRequest.inputFaceObservations = [fo]
try imageRequestHandler.perform([landmarksRequest])
let lmo = landmarksRequest.results!.first as! VNFaceObservation
```



uuid	5678
boundingBox	[0.2, 0.3, 0.4, 0.4]

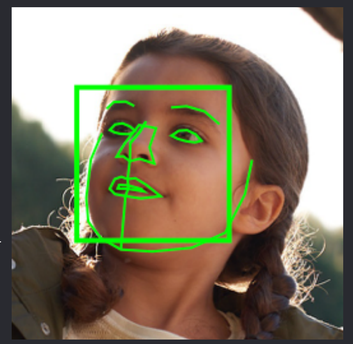
```
//Request Pipelines, Explicit execution
```

```
let faceRequest = VNDetectFaceRectanglesRequest()
let requestHandler = VNImageRequestHandler(url: imageURL)
try imageRequestHandler.perform([faceRequest])
let fo = faceRequest.results!.first as! VNFaceObservation
```



uuid	5678
boundingBox	[0.2, 0.3, 0.4, 0.4]
landmarks	nil

```
let landmarksRequest = VNDetectFaceLandmarksRequest()
landmarksRequest.inputFaceObservations = [fo]
try imageRequestHandler.perform([landmarksRequest])
let lmo = landmarksRequest.results!.first as! VNFaceObservation
```



uuid	5678
boundingBox	[0.2, 0.3, 0.4, 0.4]
landmarks	0x86461357

Lifecycle Management

How long to keep objects in memory?

Lifecycle Management

How long to keep objects in memory?

Image Request Handler—While the image needs processing

Lifecycle Management

How long to keep objects in memory?

Image Request Handler—While the image needs processing

Sequence Request Handler—While the sequence needs processing

Lifecycle Management

How long to keep objects in memory?

Image Request Handler—While the image needs processing

Sequence Request Handler—While the sequence needs processing

Requests/Observations—Lightweight objects, create/release as needed

Where to Process Your Requests?

Where to Process Your Requests?

Many requests in Vision rely on Neural Networks

Where to Process Your Requests?

Many requests in Vision rely on Neural Networks

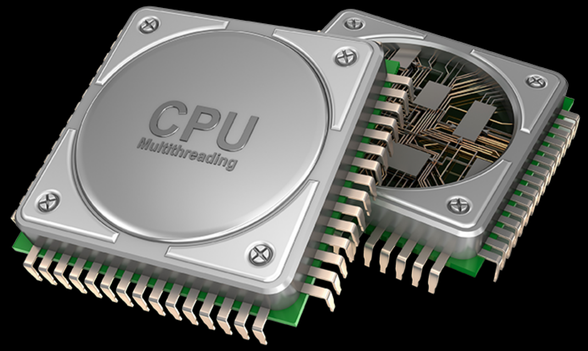
Neural Networks usually run faster on GPUs

Where to Process Your Requests?

Many requests in Vision rely on Neural Networks

Neural Networks usually run faster on GPUs

Natural question is CPU or GPU?



vs.

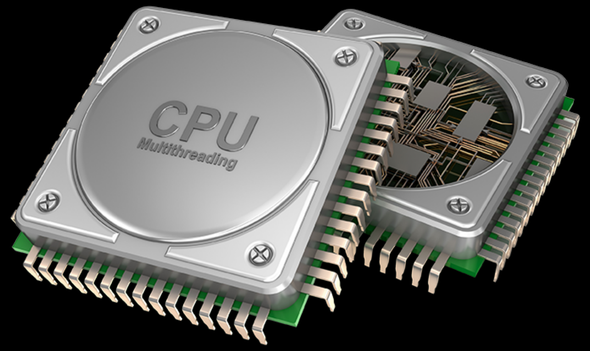
Where to Process Your Requests?

Many requests in Vision rely on Neural Networks

Neural Networks usually run faster on GPUs

Natural question is CPU or GPU?

Vision can run requests on both CPU and GPU



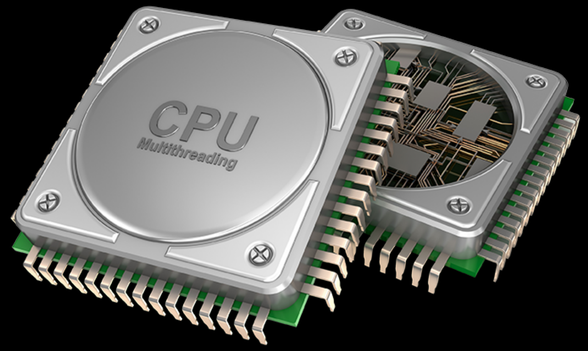
vs.

Where to Process Your Requests?

Many requests in Vision rely on Neural Networks

Neural Networks usually run faster on GPUs

Natural question is CPU or GPU?



vs.

Vision can run requests on both CPU and GPU

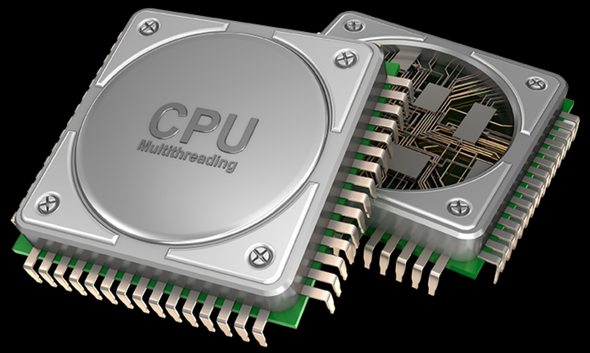
- Default—Use GPU, switch to CPU if GPU is busy

Where to Process Your Requests?

Many requests in Vision rely on Neural Networks

Neural Networks usually run faster on GPUs

Natural question is CPU or GPU?



vs.

Vision can run requests on both CPU and GPU

- Default—Use GPU, switch to CPU if GPU is busy
- Explicit—Set `VNRequest usesCPUOnly` to true

Why Vision?

New in Vision

Vision in depth

Tracking in Vision

Tracking in General

Object of interest—Auto-detected or manually selected

Sequence of frames—Camera feed, ...

Tracking—Look for the object of interest

Applications—Focus tracking with camera, ...

Why Tracking and Not Detection?

Why Tracking and Not Detection?

No specific detectors for all objects

Why Tracking and Not Detection?

No specific detectors for all objects

Need to match detected objects

Why Tracking and Not Detection?

No specific detectors for all objects

Need to match detected objects

Trackers use temporal information

Why Tracking and Not Detection?

No specific detectors for all objects

Need to match detected objects

Trackers use temporal information

Speed—Trackers are faster

Why Tracking and Not Detection?

No specific detectors for all objects

Need to match detected objects

Trackers use temporal information

Speed—Trackers are faster

Trackers are smoother, not as jittery

Tracking Types in Vision

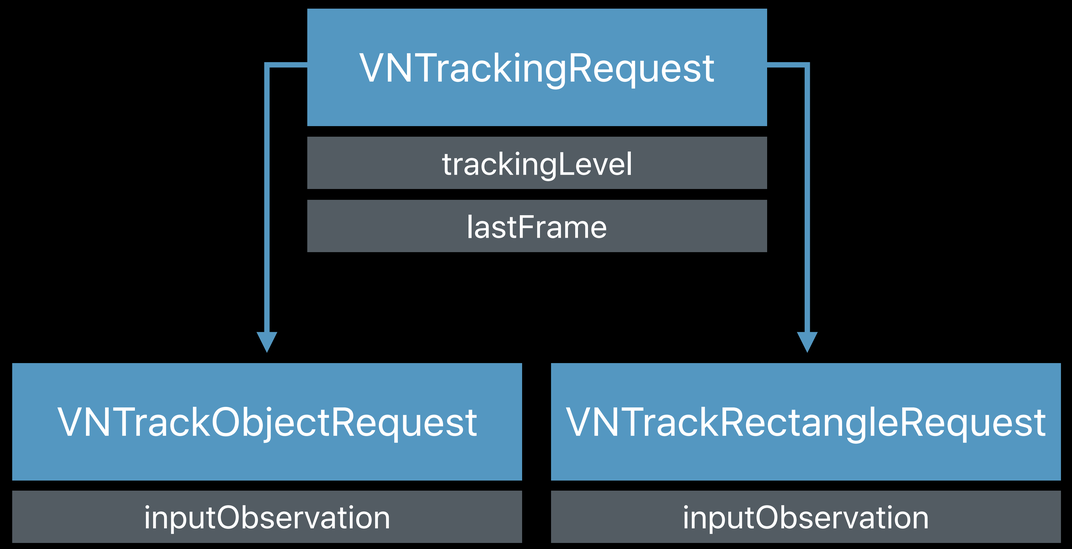
What?

How?

Results

Tracking Types in Vision

What?

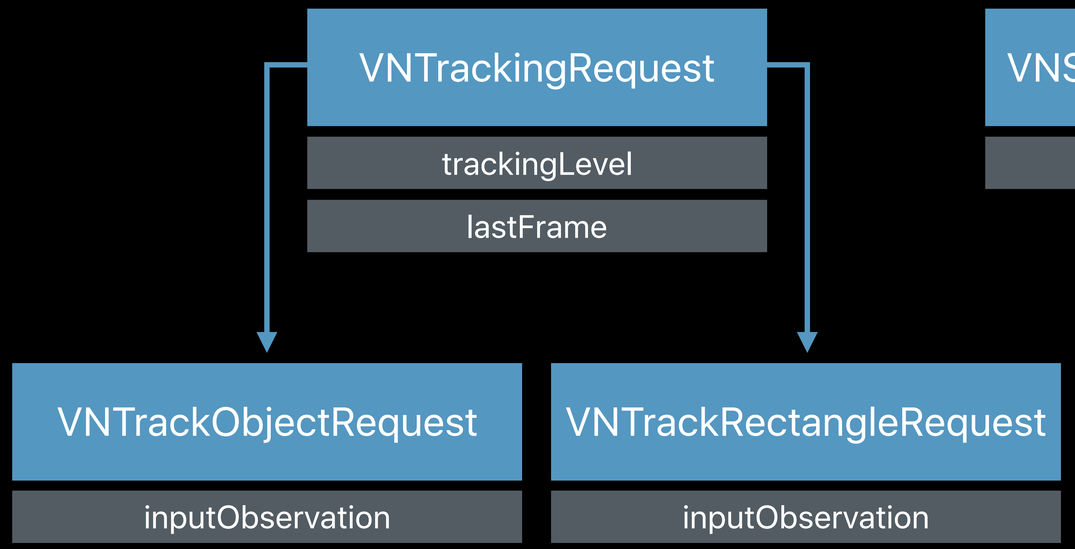


How?

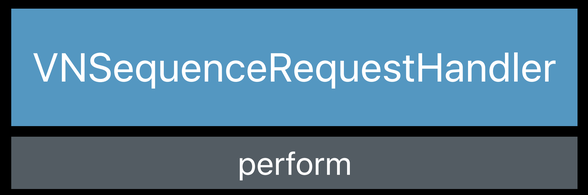
Results

Tracking Types in Vision

What?



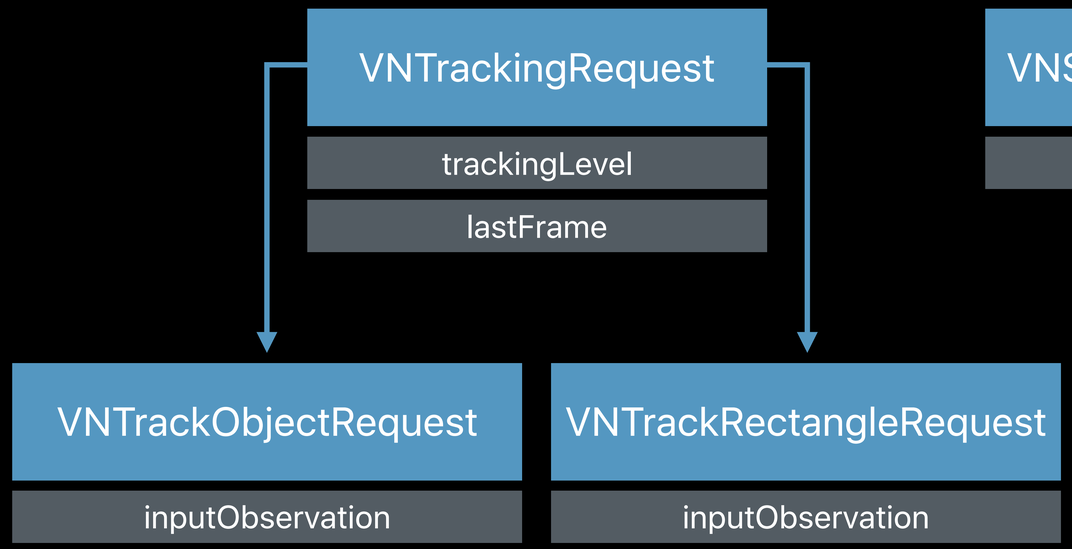
How?



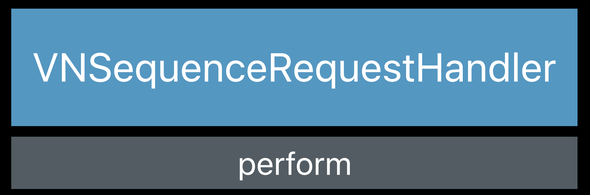
Results

Tracking Types in Vision

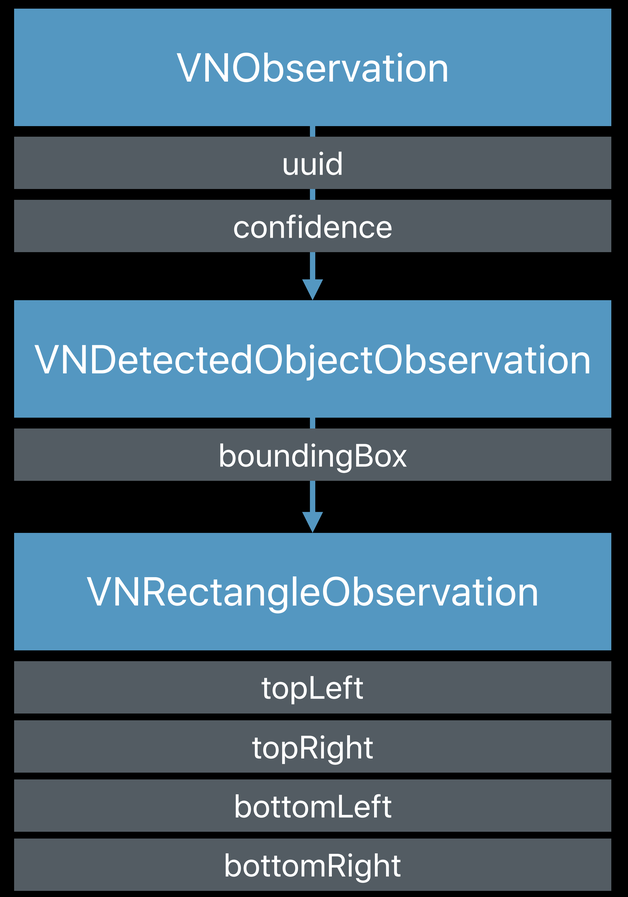
What?



How?



Results



Demo

Tracking in Vision

Tracking in Vision

Things to remember

Tracking in Vision

Things to remember

Initial object of interest selection:

- Automatic—By running an appropriate detector
- Manual—User input

Tracking in Vision

Things to remember

Initial object of interest selection:

- Automatic—By running an appropriate detector
- Manual—User input

One tracking request per tracked object (1:1)

Tracking in Vision

Things to remember

Initial object of interest selection:

- Automatic—By running an appropriate detector
- Manual—User input

One tracking request per tracked object (1:1)

Two types—`VNTrackObjectRequest` and `VNTrackRectangleRequest`

Tracking in Vision

Things to remember

Initial object of interest selection:

- Automatic—By running an appropriate detector
- Manual—User input

One tracking request per tracked object (1:1)

Two types—`VNTrackObjectRequest` and `VNTrackRectangleRequest`

Tracking algorithm—`trackingLevel = .fast or .accurate`

Tracking in Vision

Things to remember

Initial object of interest selection:

- Automatic—By running an appropriate detector
- Manual—User input

One tracking request per tracked object (1:1)

Two types—`VNTrackObjectRequest` and `VNTrackRectangleRequest`

Tracking algorithm—`trackingLevel = .fast or .accurate`

Tracking quality—Use observation `confidence` property

Tracking in Vision

Number of trackers limit

Tracking in Vision

Number of trackers limit

How many objects can we track simultaneously?

Tracking in Vision

Number of trackers limit

How many objects can we track simultaneously?

- Limit—16 trackers of each type at a time

Tracking in Vision

Number of trackers limit

How many objects can we track simultaneously?

- Limit—16 trackers of each type at a time
- Error is returned if over limit

Tracking in Vision

Number of trackers limit

How many objects can we track simultaneously?

- Limit—16 trackers of each type at a time
- Error is returned if over limit

How to release a tracker?

Tracking in Vision

Number of trackers limit

How many objects can we track simultaneously?

- Limit—16 trackers of each type at a time
- Error is returned if over limit

How to release a tracker?

- Request’s property `lastFrame = true`

Tracking in Vision

Number of trackers limit

How many objects can we track simultaneously?

- Limit—16 trackers of each type at a time
- Error is returned if over limit

How to release a tracker?

- Request’s property `lastFrame = true`
- Release `VNSequenceRequestHandler`

Tracking Challenges

Objects change their shape, size, appearance, color, ...

Tracking Challenges

Objects change their shape, size, appearance, color, ...

Fast or accurate?

Tracking Challenges

Objects change their shape, size, appearance, color, ...

Fast or accurate?

Initial bounding box location, use salient objects

Tracking Challenges

Objects change their shape, size, appearance, color, ...

Fast or accurate?

Initial bounding box location, use salient objects

Which confidence level threshold to use?

Tracking Challenges

Objects change their shape, size, appearance, color, ...

Fast or accurate?

Initial bounding box location, use salient objects

Which confidence level threshold to use?

Consider rerunning detectors every N frames

Summary

Summary

Why Vision?

- Multi-platform, privacy-oriented framework with simple, consistent interface

Summary

Why Vision?

- Multi-platform, privacy-oriented framework with simple, consistent interface

New in Vision

- Orientation-agnostic Face Detector, request revisioning

Summary

Why Vision?

- Multi-platform, privacy-oriented framework with simple, consistent interface

New in Vision

- Orientation-agnostic Face Detector, request revisioning

Vision in depth

- “What? = Requests” “How? = Request Handlers” “Results = Observations”

Summary

Why Vision?

- Multi-platform, privacy-oriented framework with simple, consistent interface

New in Vision

- Orientation-agnostic Face Detector, request revisioning

Vision in depth

- “What? = Requests” “How? = Request Handlers” “Results = Observations”

Tracking in Vision

- Two trackers, two algorithms—each to track multiple objects simultaneously

More Information

https://developer.apple.com/wwdc18/716

Vision with Core ML	Hall 1	Thursday 3:00PM
Vision Lab	Technology Lab 11	Friday 3:00PM


WWDC18