Structure

DetectRectanglesRequest

An image-analysis request that finds projected rectangular regions in an image.

iOS 18.0+ | iPadOS 18.0+ | macOS 15.0+ | tvOS 18.0+ | visionOS 2.0+

```
struct DetectRectanglesRequest
```

Overview

A rectangle-detection request locates regions of an image with rectangular shape, like credit cards, business cards, documents, and signs. The request returns its observations in the form of RectangleObservation objects, and contains normalized coordinates of bounding boxes containing a rectangle.

Use this type of request to find the bounding boxes of rectangles in an image. Vision returns observations for rectangles found in all orientations and sizes, along with a confidence level to indicate how likely it is that the observation contains an actual rectangle.

To further configure or restrict the types of rectangles found, set properties on the request specifying a range of aspect ratios, sizes, and quadrature tolerance.

Topics

Creating a request

init(DetectRectanglesRequest.Revision?)

Creates a rectangle-detection request.

Getting the revision

```
let revision: DetectRectanglesRequest.Revision
```
The algorithm or implementation the request uses.

```
static let supportedRevisions: [DetectRectanglesRequest.Revision]
```
The collection of revisions the request supports.

```
enum Revision
```
A type that describes the algorithm or implementation that the request performs.

Inspecting a request

```
var maximumAspectRatio: Float
```
The largest aspect ratio the rectangle request detects.

```
var maximumObservations: Int
```
The maximum number of rectangles Vision returns.

```
var minimumAspectRatio: Float
```
The smallest aspect ratio the rectangle request detects.

```
var minimumConfidence: Float
```
The minimum acceptable confidence level for detected rectangles.

```
var minimumSize: Float
```
The minimum size of the rectangle to be detected, as a proportion of the smallest dimension.

```
var quadratureToleranceDegrees: Float
```
The maximum number of degrees a rectangle corner angle can deviate from 90°.

Performing a request

```
func perform(on: URL, orientation: CGImagePropertyOrientation?) async
throws -> Self.Result
```
Performs the request on an image URL and produces observations.
Required Default implementations provided.

```
func perform(on: Data, orientation: CGImagePropertyOrientation?) async
throws -> Self.Result
```

Performs the request on image data and produces observations.

Required Default implementations provided.

```
func perform(on: CGImage, orientation: CGImagePropertyOrientation?)
async throws -> Self.Result
```

Performs the request on a Core Graphics image and produces observations.

Required Default implementations provided.

```
func perform(on: CVPixelBuffer, orientation: CGImagePropertyOrientation
?) async throws -> Self.Result
```

Performs the request on a pixel buffer and produces observations.

Required Default implementations provided.

```
func perform(on: CMSampleBuffer, orientation: CGImageProperty
Orientation?) async throws -> Self.Result
```

Performs the request on a Core Media buffer and produces observations.

Required Default implementations provided.

```
func perform(on: CIImage, orientation: CGImagePropertyOrientation?)
async throws -> Self.Result
```

Performs the request on a Core Image image and produces observations.

Required Default implementations provided.

```
struct RectangleObservation
```

An object that represents the four vertices of a detected rectangle.

Relationships

Conforms To

```
CustomStringConvertible
Equatable
Hashable
ImageProcessingRequest
Sendable
SendableMetatype
VisionRequest
```

See Also

Optical flow and rectangle detection

`class TrackOpticalFlowRequest`

A request that determines the direction change of vectors for each pixel from a previous to current image.