```
import Foundation

extension MLFeatureValue {

    /// Options keys passed into the
MLFeatureValue construction for image
types
    public struct ImageOption : Hashable,
Equatable, RawRepresentable, @unchecked
Sendable {

        public init(_ rawValue: String)

        public init(rawValue: String)
    }
}

extension MLFeatureValue.ImageOption {

    /// Key for CGRect describing a crop
region of interest of image source in
normalized coordinates
    @available(macOS 10.15, *)
    public static let cropRect:
MLFeatureValue.ImageOption

    /// Key for VNImageCropAndScaleOption
describing how to crop and scale the
image (or region of interest) to the
desired size
    @available(macOS 10.15, *)
    public static let cropAndScale:
MLFeatureValue.ImageOption
```

```
}

@available(macOS 10.15, *)
extension MLFeatureValue {

    /// Construct image feature value
from an image on disk. Orientation is
read from Exif if avaiable
    public convenience init(imageAt url:
URL, pixelsWide: Int, pixelsHigh: Int,
pixelFormatType: OSType, options:
[MLFeatureValue.ImageOption : Any]? =
nil) throws

    /// Construct image feature value
from an image on disk, using a model
specified image constraint. Orientation
is read from Exif if avaiable
    public convenience init(imageAt url:
URL, constraint: MLImageConstraint,
options: [MLFeatureValue.ImageOption :
Any]? = nil) throws

    /// Construct image feature value
from CGImage (orientation is assumed to
be kCGImagePropertyOrientationUp)
    public convenience init(cgImage:
CGImage, pixelsWide: Int, pixelsHigh:
Int, pixelFormatType: OSType, options:
[MLFeatureValue.ImageOption : Any]? =
nil) throws

    /// Construct image feature value
```

```
from CGImage, using the size and type
information required by feature
description (orientation is assumed to be
kCGImagePropertyOrientationUp)
    public convenience init(cgImage:
CGImage, constraint: MLImageConstraint,
options: [MLFeatureValue.ImageOption :
Any]? = nil) throws

    /// Construct image feature value
from an image on disk. The passed in
orientation supersedes any in the file
    public convenience init(imageAt url:
URL, orientation:
CGImagePropertyOrientation, pixelsWide:
Int, pixelsHigh: Int, pixelFormatType:
OSType, options:
[MLFeatureValue.ImageOption : Any]? =
nil) throws

    /// Construct image feature value
from an image on disk using a model
specified image constraint. The passed in
orientation supersedes any in the file
    public convenience init(imageAt url:
URL, orientation:
CGImagePropertyOrientation, constraint:
MLImageConstraint, options:
[MLFeatureValue.ImageOption : Any]? =
nil) throws

    /// Construct image feature value
from CGImage w/ specified orientation
```

```
    public convenience init(cgImage:
CGImage, orientation:
CGImagePropertyOrientation, pixelsWide:
Int, pixelsHigh: Int, pixelFormatType:
OSType, options:
[MLFeatureValue.ImageOption : Any]? =
nil) throws

    /// Construct image feature value
from CGImage w/ specified orientation,
using the size and type information
required by feature description
    public convenience init(cgImage:
CGImage, orientation:
CGImagePropertyOrientation, constraint:
MLImageConstraint, options:
[MLFeatureValue.ImageOption : Any]? =
nil) throws
}
```