

[Vision](#) / [Original Objective-C and Swift API](#) / Extracting phone numbers from text in images

## Sample Code

# Extracting phone numbers from text in images

Analyze and filter phone numbers from text in live capture by using Vision.

[Download](#)

iOS 13.0+ | iPadOS 13.0+ | Xcode 14.2+

## Overview

New in iOS 13, the Vision framework adds improved text-recognition capabilities that enable you to more easily extract text data from images. The sample app shows you how to use these capabilities to detect and display phone numbers scanned by the camera. It highlights the use of the fast recognition level and a reduced scope to provide responsive real-time text recognition.

The `VNRecognizeTextRequest` encapsulates all the necessary logic to recognize text in an image. There are two recognition levels you can choose from, to prioritize accuracy or speed of recognition. Additional configuration properties enable you to specify the languages to recognize, use language correction during recognition, specify the area of the live video to scan, and more. With all of these capabilities, there are tradeoffs between convenience, accuracy, and performance.

First the app creates a request to recognize text:

```
// Set up the Vision request before letting ViewController set up the camera
// so it exists when the first buffer is received.
request = VNRecognizeTextRequest(completionHandler: recognizeTextHandler)
```

Next, in the delegate method that Vision calls when it writes a sample buffer, the app configures the request to use fast recognition, disable language correction, and set a small rectangular region

of interest. Then, the app performs the request:

```
// Configure for running in real time.
request.recognitionLevel = .fast
// Language correction doesn't help in recognizing phone numbers and also
// slows recognition.
request.usesLanguageCorrection = false
// Only run on the region of interest for maximum speed.
request.regionOfInterest = regionOfInterest

let requestHandler = VNImageRequestHandler(cvPixelBuffer: pixelBuffer, orientation:
do {
    try requestHandler.perform([request])
} catch {
    print(error)
}
```

Disabling language correction improves performance but produces less accurate results. The app includes extensions on `Character` and `String` to perform correction optimized for phone-number recognition. See `StringUtils.swift` to review this code.

## Configure the sample code project

To run this sample app, you need the following:

- Xcode 11 or later
- iPhone with iOS 13 or later

Connect the iPhone to the Mac over USB. The first time you run this sample app, the system prompts you to grant the app access to the camera. You must allow the sample app to access the camera for it to function correctly.

### Note

This sample code project is associated with WWDC19 session 234: [Text Recognition in Vision Framework](#).

## See Also

# Text recognition



## Recognizing Text in Images

Add text-recognition features to your app using the Vision framework.



## Structuring Recognized Text on a Document

Detect, recognize, and structure text on a business card or receipt using Vision and VisionKit.



## Locating and displaying recognized text

Perform text recognition on a photo using the Vision framework's text-recognition request.

```
class VNRecognizeTextRequest
```

An image-analysis request that finds and recognizes text in an image.

```
class VNRecognizedTextObservation
```

A request that detects and recognizes regions of text in an image.