Exercise - 8

November 9, 2023

Objective

Create an iPhone and iWatch app that recognizes text/objects in the picture and indicates that on the watch.

Motivation

You will learn how to use: SwiftUI, WatchConnectivity, NaturalLanguage, CoreML, Resnet50, Vision .

Must Follow

You must follow the rules below. Otherwise, you will get 50% of your actual score.

1. Start your XCode project "**Exercise8_LastName_FirstName**" (replace **LastName** with your last name and **FirstName** with your first name).
a. **DON'T DO** the following:
    i. Start with any other project name and change the zip file name later.
    ii. This will not be accepted at all.

2. You must have to do the exercise compatible with **XCode version 15.0 or higher**.

Tips

- **Read the question carefully, then start coding!**
- **Build, Build, and Build**

    o If you add anything on storyboard -> **Build**
    o If you make a reference from storyboard -> **Build**
    o Do not wait until finishing all parts and build.
    o It is easier to debug after each single feature added.

UNIVERSITY of
HOUSTON

Details

Create a double view iPhone application and a paired single view iWatch application using Swift as a programming language. Start your XCode project "**Exercise8_LastName_FirstName**" (replace **LastName** with your last name and **FirstName** with your first name).

[5 pts] Design your interface to look like the screenshots [Figures 1-5]
• Design your UI for the Apple Watch and iPhone
• Single emoji, system font, size 80; Phone central image 350x350; tab icons – "photo", "doc.text"; use homework assets for proper images.
• Logo, system font, size 36, bold
• Colors: red and gray

[1 pt.] When user launches the app for the first time and no data (or unknown object) were sent to watch it displays question mark [Figure 1].

[1 pt.] After user launches phone app it shows car image and is set to "Object identification" tab. Objects are recognized on the image and probabilities together with the names of the objects are depicted in the first 7 lines of output. Data sent to watch. [Figure 3]

[1 pt.] When user taps "Text identification" tab, app switches user to another view [Figure 2] and same steps are performed but with text (text recognized, first 7 lines displayed, data sent to watch). [Figure 2]

[1 pt.] When user taps on the image, image is replaced with randomly selected image (excluding the current one), is analyzed and proper data sent to watch. When user switches between tabs, current image is not changing.

• The application should not crash permissions were not granted. Add an appropriate alert for the user in case of any error.

• Make sure to add the appropriate properties to the target properties list:
Privacy – Health Share Usage Description
Privacy – Health Update Usage Description
Privacy – Health Records Usage Description

Bonus
[1 pts.] New feature: show first three recognized objects on the watch together with the probabilities. Use whole emoji library to represent the object, use '?' when there are no fitting emojis.
[1 pts.] Add massive feature: support of the photo library and camera. Recognize all around you 😊
[1 pts.] Use concurrent tasks in the app, display proper 'loading'/'waiting' status at the text field. That would be a valuable skill for your future career! UIKit implementation in git.

Submission
Zip XCode project and submit to the blackboard. The name of your zip file will be automatically "**Exercise7_LastName_FirstName.zip**" (**Last Name** is your last name and **FirstName** is your first name). One submission per person.
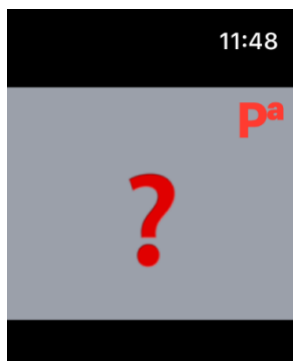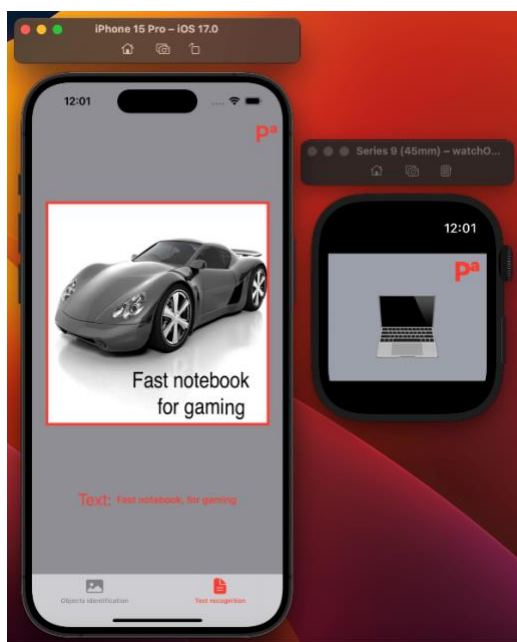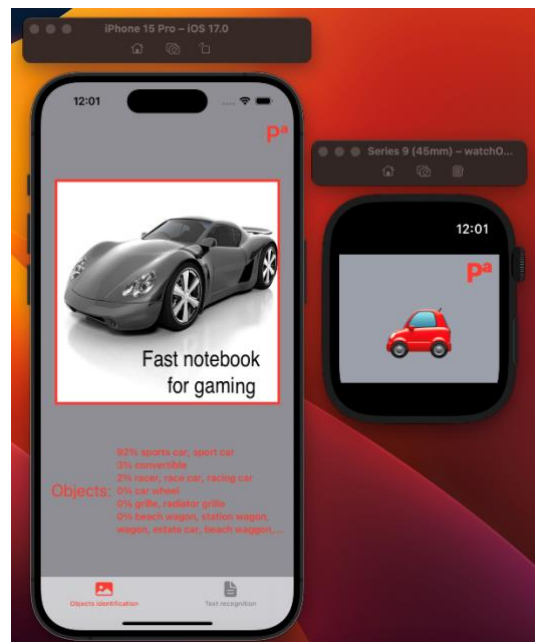
Figure 1

Figure 2

Figure 3

Figure 4

Figure 5