



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Automatizálási és Alkalmazott Informatika Tanszék

Dudás Tamás Alex

**Látássérültek támogatása alkalmazott
mesterséges intelligencia eszközökkel iOS
környezetben**

Konzulens

Dr. Ekler Péter

Budapest, 2023

Tartalomjegyzék

Összefoglaló	3
Abstract	4
1. Bevezetés	5
1.1 <i>Problémafelvetés és a tématerület áttekintése</i>	5
1.2 <i>Célkitűzés és motiváció</i>	5
1.3 <i>Dolgozat felépítése</i>	6
2. Felhasználói igények és követelmények	8
2.1 <i>Látássérült felhasználók mindennapi kihívásai</i>	8
2.2 <i>Universális tervezés iOS platformon általánosságban</i>	9
3. Tervezési folyamat és az alkalmazás átfogó prezentációja	10
3.1 <i>Univerzális tervezési alapelvek</i>	10
3.2 <i>Felhasználói élmény tervezése</i>	12
3.3 <i>Az alkalmazás funkcionalitása</i>	14
4. Mesterséges intelligencia és gépi látás keretrendszerek iOS környezetben	15
4.1 <i>Bevezetés Vision – Core ML</i>	15
4.2 <i>Vision / Core ML – Valós idejű objektum detektálás</i>	16
4.2.1 <i>Bevezetés</i>	16
4.3 <i>Objektum detektálás</i>	16
4.3.1 <i>Bevezetés</i>	17
4.3.3 <i>Objektumazonosságok felkutatása és hasonlóságok elemzése</i>	21
4.4 <i>Objektum szkennelés</i>	27
4.4.1 <i>Bevezetés</i>	27
4.4.2 <i>Megvalósítás</i>	28
4.4.3 <i>Konklúzió, eredmények összefoglalása</i>	28
5. Navigáció és tájékozódás	31
5.1 <i>Térkép</i>	31
5.1.1 <i>Bevezetés</i>	31
5.1.2 <i>Eredmények bemutatása és magyarázata</i>	32
5.2 <i>Összegzés</i>	37
6. Összegzés és továbbfejlesztési lehetőség	38
6.1 <i>Elért eredmények és elemzése</i>	38
6.2 <i>Kapcsolódó már létező megoldások</i>	39
6.3 <i>Jövőbeli fejlesztési irányok</i>	40
7. Elkészült alkalmazás bemutatása	41
8. Irodalomjegyzék	42

Összefoglaló

Az információs technológia rohamos fejlődése az okos telefonok elterjedésével új lehetőségeket teremt mindennapi életünkben. Az emberek széles körben használják az iOS rendszert futtató okos telefonokat, melyek számos funkciót kínálnak az egyszerű kommunikációtól a szórakoztatásig. Azonban azoknak, akik látássérüléssel élnek, továbbra is kihívást jelenthet az ilyen eszközök használata, mivel a jelenlegi megoldások nem elégségesek ahhoz, hogy teljes mértékben kihasználják az okos telefonok nyújtotta lehetőségeket. A látássérültek számára napi tevékenységeik során sokszor nehézkes vagy akár lehetetlen az önálló navigáció, információk elérése és kommunikáció másokkal. Az alkalmazott mesterséges intelligencia (AI) eszközök képesek forradalmi változásokat hozni ezen a területen. Ez a TDK dolgozat azt kutatja, hogy hogyan lehet az iOS környezetben alkalmazni a mesterséges intelligenciát a látássérült emberek mindennapi életének támogatására. A dolgozatomban bemutatom az általam fejlesztett alkalmazást, amely az iOS platformon elérhető natív mesterséges intelligencia technológiákat felhasználva nyújt többféle segítséget látásban korlátozott személyeknek, úgy mint objektumok detektálása és a detektált objektumon való szöveg keresés és felolvasás (alapértelmezetten magyar nyelven), valós idejű objektum keresés és összehasonlítás, továbbá egy térkép funkció is elérhető, amellyel önkéntes segítőknek nyílik lehetőség a látássérült emberek támogatására. Egy olyan alkalmazást igyekeztem fejleszteni, amely során kiemelt figyelmet kap az egyszerű kezelhetőség és a felhasználói élmény javítása, ezen túl az alkalmazás sok funkcióval és modullal egészült ki, amelyek megvalósításánál olyan architektúrális döntéseket hoztam meg, amellyel később könnyen továbbfejleszthető és kiegészíthető. Emellett a dolgozat elemzi az AI algoritmusokat, amelyek az érzékelés és szövegátvitel területén működnek, és ezeknek az algoritmusoknak az iOS platformon való implementálását. A kutatás során végzett kísérletek és felhasználói tesztek eredményei azt mutatják, hogy az alkalmazott mesterséges intelligencia eszközök jelentős javulást hoznak a látássérült emberek mindennapi életminőségében. Az általam vizsgált megoldások és a megvalósított funkciók eredményei alapján megállapítható, hogy az ilyen eszközök széles körű alkalmazása elősegítheti a társadalmi inklúziót és lehetővé teszi a látássérültek számára, hogy teljes értékűen részt vegyenek az információs társadalomban. A dolgozat összefoglalja az általam vizsgált és elkészített alkalmazott mesterséges intelligencia eszközök jelenlegi állapotát és jövőbeli kilátásait az iOS környezetben, továbbá megvizsgálja azok társadalmi és etikai vonatkozásait. A dolgozatom során kidolgoztam olyan innovatív megoldásokat, amik a technológiát felhasználva valóban segítséget nyújthatnak, mint például a gépi látás alapú környezet felismerés, szöveg felolvasás és saját tárgyak megkeresése. Az általam kidolgozott megoldásokat a tervezés és fejlesztés után valós környezetben is teszteltem. A dolgozatom során bemutatom az általam megvalósított újszerű megoldások tervezési és fejlesztési lépéseit és a tesztelés tapasztalatait is.

Abstract

The rapid development of information technology with the spread of smart phones is creating new opportunities in our daily lives. People make widespread use of smartphones running iOS, which offer a range of features from simple communication to entertainment. However, for those living with visual impairments, using these devices can remain a challenge as current solutions are not sufficient to take full advantage of the possibilities offered by smartphones. Visually impaired people often find it difficult or even impossible to navigate independently, access information and communicate with others in their daily activities. Artificial intelligence (AI) tools have the potential to revolutionise this area. This TDK thesis explores how AI can be applied in the iOS environment to support the daily lives of people with visual impairments. In my thesis, I will present an application I developed that uses native artificial intelligence technologies available on the iOS platform to provide multiple types of assistance to visually impaired people, such as object detection and text search and reading on the detected object (in Hungarian as a default language), real-time object search and comparison, and a map function to enable volunteer helpers to support visually impaired people. I have tried to develop an application with a focus on ease of use and improving the user experience, and I have added many features and modules to the application, which I have implemented with architectural choices that allow for easy future development and enhancement. In addition, the thesis analyses the AI algorithms that operate in the field of perception and text translation, and the implementation of these algorithms on the iOS platform. The results of the experiments and user tests carried out in this research show that the AI tools used bring significant improvements in the quality of daily life of people with visual impairments. I have tried to develop an application with a focus on ease of use and improving the user experience, and I have added many features and modules to the application, which I have implemented with architectural choices that allow for easy future development and enhancement. In addition, the thesis analyses the AI algorithms that operate in the field of perception and text translation, and the implementation of these algorithms on the iOS platform. The results of the experiments and user tests carried out in this research show that the AI tools used bring significant improvements in the quality of daily life of people with visual impairments. Based on the results of the solutions I have investigated and the functions implemented, it can be concluded that the widespread use of such tools can promote social inclusion and enable visually impaired people to participate fully in the information society. This paper summarizes the current state and future prospects of the applied AI tools I have studied and developed in the iOS environment, and examines their social and ethical implications. In the course of my thesis, I have developed innovative solutions that can really help using the technology, such as machine vision-based environment recognition, text reading and finding your own objects. After designing and developing the solutions, I tested them in a real environment. In my thesis, I will also present the design and development steps of the novel solutions I implemented and the testing experiences.

1. Bevezetés

Napjainkban a látássérültek számára a mobil alkalmazások hozzáférhetősége kritikus fontosságú. Azoknak a felhasználóknak, akik látásukban korlátozottak, ezen alkalmazások hozzáférhetősége lehetővé teszi számukra a digitális tartalmakhoz és szolgáltatásokhoz való egyenlő és hatékony hozzáférést, továbbá a külvilággal való kommunikációban is hasznos szerepet tölthetnek be. Ennek során fontos szempont az alkalmazások hangutasítások, képernyőolvasók és egyéb segítségével való könnyű navigációja és használata.

1.1 Problémafelvetés és a tématerület áttekintése

A látássérültek emberek számára a környezeti tájékozódás és az információhoz való hozzáférés alapvető fontosságú mindennapi életükben. Azonban sok esetben a látássérült emberek számára a digitális térben való hozzáférés és navigáció korlátozott, ami jelentősen befolyásolja életminőségüket és a társadalmi részvételüket. Az információs és kommunikációs technológiák fejlődése és a mesterséges intelligencia térnyerése új lehetőségeket kínál a digitális térben való hozzáférés javítására. Ugyanakkor fontos megjegyezni, hogy az olyan alkalmazások, amelyeket a látássérült felhasználók támogatására terveztek, még mindig elmaradnak a fejlődésben. Ezért a kutatás középpontjában a látássérült felhasználók specifikus igényeinek és az iOS környezetben alkalmazott mesterséges intelligencia eszközök fejlesztésének a fontossága áll. Az Apple vállalat ugyan elkötelezett a hozzáférhetőség javítása iránt az iOS operációs rendszerében is, de megállapítható, hogy a látássérült felhasználók számára a digitális térben történő hozzáférés és navigáció terén számos specifikus probléma fennáll. Az iOS rendszer alapvetően vizuális interfészen alapul, ami korlátozott hozzáférést eredményez azok számára, akik látássérültek vagy korlátozott látással rendelkeznek. Ezek a problémák nem csupán technológiai jellegűek, hanem az eszközök és alkalmazások fejlesztői közreműködését is igénylik. Annak ellenére, hogy az Apple hangsúlyozza a hozzáférhetőség fontosságát, az alkalmazások nagy része továbbra sem felel meg a látássérült felhasználók igényeinek.

1.2 Célkitűzés és motiváció

A kutatás fő célja egy demonstrációs alkalmazás létrehozása [1], amelynek célja bemutatni, hogy hogyan lehet hatékonyan biztosítani a látássérült felhasználók számára a digitális térben történő hozzáférést és navigációt, ezzel hozzájárulva mindennapi életük könnyebben való éléséhez. Az alkalmazás kidolgozásával azt szeretném demonstrálni, hogy hogyan tervezhetünk és fejleszthetünk digitális tartalmakat, figyelembe véve a hozzáférhetőségi szempontokat és a

látássérült felhasználók specifikus igényeit. Emellett azt kutatom, hogy hogyan alkalmazhatjuk a legújabb technológiai megoldásokat a látássérült felhasználók számára, hogy segítsük őket hatékonyan navigálni és hozzáférni az információkhoz a digitális térben. Másodlagos cél a látássérült felhasználók általános igényeinek és elvárásainak feltárása, valamint azok pontos azonosítása, hogy hatékony technológiai megoldásokkal szolgálhassunk számukra a digitális térben való hatékony navigáció, információhoz való hozzáférés és hatékony kommunikáció terén. Ezen keresztül vizsgálom a legfrissebb technológiai alapokat, modelleket és algoritmusokat, hogy a látássérült felhasználók számára még hatékonyabb eszközöket és alkalmazásokat fejleszthessünk. A kutatás és fejlesztés központjában egy olyan mintaalkalmazás áll, amely célja a látássérült felhasználói elvárások teljesítése az iOS platformon. Az alkalmazás az elérhető legmodernebb technológiákat alkalmazza, hogy bemutassa, hogyan lehet a mesterséges intelligencia és natív könyvtárak segítségével létrehozni olyan hasznos eszközöket, amelyek segítik a látássérült felhasználókat mindennapi életük során. Az alkalmazás kiemelkedő jellemzője a natív mesterséges intelligencia könyvtárak integrálása, amelyek lehetővé teszik a virtuális asszisztensek és hangalapú kommunikációs megoldások használatát. Az alkalmazás nem csupán az információ és kommunikáció forrásait kínálja, hanem egyedi funkcionalitást is biztosít, amely lehetővé teszi a felhasználók számára a hangalapú kommunikációt a környezetükkel. Ezenkívül az alkalmazás olyan eszközöket és funkciókat kínál, amelyek megkönnyítik a látássérült felhasználók mindennapi tevékenységeit. Az alkalmazás egyaránt alkalmazkodik a már meglévő kommunikációs platformokhoz és saját fejlesztésekhez, hogy minél hatékonyabban és sokoldalúbban segítse a látássérült embereket.

1.3 Dolgozat felépítése

Ebben a bekezdésben kifejtem a következő fejezetek címét és egy rövid összefoglalót arról, hogy melyik fejezet miről fog szólni a dolgozatban. Ezzel előzetes betekintést nyújtva a dolgozat struktúrájába és tartalmába.

- A 2. fejezet a látássérült felhasználók felhasználói igényeivel és követelményeivel foglalkozik, és az iOS platformon történő hozzáférhetőségi szempontokat mutatja be. A fejezet továbbá kitér az univerzális tervezés alapjaira a látássérült felhasználók számára.
- A 3. fejezet részletesen tárgyalja a technikai döntéseket, amelyek az alkalmazás kialakításához vezettek, bemutatva az alkalmazás technikai felépítését. Emellett a felhasználói élmény tervezésére is fókuszál, és részletesen ismerteti az applikáció elérhető funkcionalitását, amely a kutatás során fejlesztésre került.

- A 4. fejezet részletesen tárgyalja a platformon elérhető és az alkalmazásban alkalmazott mesterséges intelligencia könyvtárak, mind elméleti, mind gyakorlati szempontból.
- Az 5. fejezetben belül, a navigációs és tájékozási kihívások körét mutatom be egy lehetséges általam fejlesztett lehetséges megoldással kiegészítve.
- A 6. fejezet az összegzés és az elért eredményeket részletezi, kiegészítve a kutatás során fejlesztett applikáció bemutatói videóival, amelyek az alábbi linkeken elérhetőek.
 - <https://youtu.be/wQh2VjNhVuA> - Applikáció bemutató és indítás, háttérrendszerek ismertetése [29]
 - <https://youtube.com/shorts/ITouEsw8mkQ?feature=share> – Regisztráció fizikai eszközön és autentikáció bemutatása [30]
 - <https://youtu.be/x1EKSoDwHgg> - *VoiceOver* és interaktív mód bemutatása [31]
 - <https://youtu.be/SHcgXmA4BQw> - Objektum szkennelés bemutatása [32]
 - <https://youtu.be/VeV8trq0PBM> - Térkép funkció bemutatása [33]
 - <https://youtu.be/1s1MvMK9JIU> - Objektum detektálás bemutatása [34]

2. Felhasználói igények és követelmények

A látássérült felhasználók speciális hozzáférési igényekkel rendelkeznek, amelyeket a mobilalkalmazások fejlesztőinek figyelembe kell venniük. A hangvezérlés, képernyőolvasók támogatása, nagyítás és kontrasztbeállítások lehetősége, valamint a valós idejű navigációs funkciók mind olyan elemek, amelyek segítik a látássérült felhasználók hozzáférését és használatát. A kutatásom célja, a már korábbi elérhető segítségek mellett, a natív mesterséges intelligencia könyvtárak alkalmazásával további hozzáférhetőségi megoldások fejlesztése. Az elkészítendő alkalmazás leendő fejlettségi szintjén a teljesen vak felhasználók számára az applikáció használata bizonyos kihívásokkal fog járni, és még nem lesz képes teljesen támogatni szükségleteiket. Fontos hangsúlyozni azonban, hogy a dolgozat későbbi fejezetében lévő technikai döntések és az alkalmazás alapstruktúrája miatt kiválóan továbbfejleszhető. Ennek eredményeként a jövőbeni fejlesztések során lehetőség nyílik majd az akadályok elhárítására, hogy a teljesen vak felhasználók számára könnyedén és hatékonyan használható legyen az alkalmazás. Emellett érdemes hangsúlyozni, hogy az alkalmazás nem csupán a teljesen vak felhasználók számára készül. A gyengén vagy részlegesen látó felhasználók számára is rendkívül hasznos, akár azoknak, akik látási nehézségekkel küzdenek betegségük vagy más tényezők miatt. Az alkalmazás leendő funkcionalitása is jelentős segítséget nyújthat számukra.

2.1 Látássérült felhasználók mindennapi kihívásai

A mobil alkalmazások használata a látássérült felhasználók számára komoly kihívásokat rejt. A rohamosan fejlődő mobilalkalmazások piacán a látássérült felhasználók gyakran találják magukat az alkalmazások hozzáférhetőségének hiányosságaival szembesülve. A legtöbb mobilalkalmazás *vizuális orientációval* tervezett, és nem mindig veszi figyelembe a látássérült felhasználók speciális szükségleteit. Ezeknek a felhasználóknak gyakran nehézséget okoz a vizuális elemek, mint például gombok és menük azonosítása, valamint az alkalmazásokban található információk és tartalmak elérése. A navigációs struktúra és a felhasználói felület tervezése létfontosságú, hogy könnyűvé tegye a látássérült felhasználók számára az alkalmazások hatékony használatát. Az operációs rendszerek és alkalmazások tervezői számára alapvető fontosságú a technológiai innováció támogatása és a hozzáférhetőségi szabványok szigorú betartása a látássérült felhasználók számára. A tervezők és a látássérült felhasználók közötti együttműködés előmozdíthatja az alkalmazások hozzáférhetőségének és funkcionalitásának folyamatos fejlesztését. Ezen keresztül lehetőséget teremthetünk a

látássérült felhasználók mindennapi életminőségének javítására, és elősegíthetjük, hogy teljeskörűen részt vegyenek a digitális világban. Az innováció és a közös munka révén megteremthetjük a látássérült felhasználók számára az esélyt a digitális térben való hatékonyabb részvételre, és ezáltal hozzájárulhatunk az esélyegyenlőséghez és az inkluzív digitális társadalom kialakításához.

2.2 Universális tervezés iOS platformon általánosságban

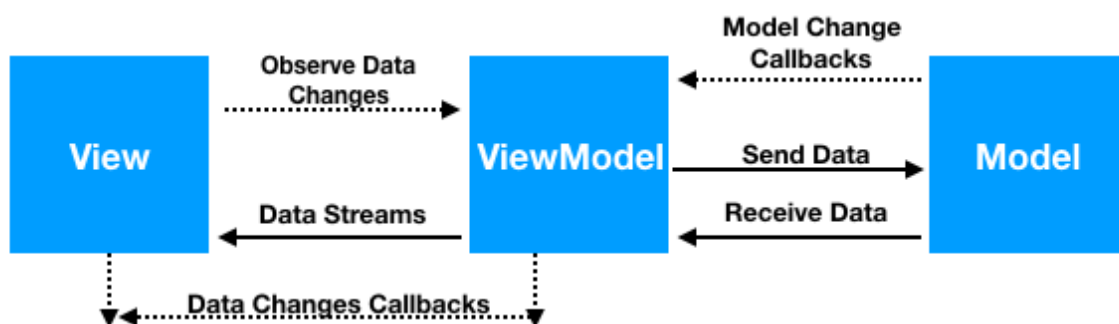
Az Apple tervezési útmutatója [2] kiemelkedő forrás a fejlesztők számára, amelynek segítségével részletes és szakmailag megalapozott iránymutatásokat és elveket kapunk az iOS alkalmazások tervezéséhez és fejlesztéséhez. Az útmutató kiemelt hangsúlyt fektet az univerzális tervezésre és hozzáférhetőségre, figyelembe véve az olyan felhasználókat is, akik látássérültek vagy egyéb speciális igényekkel rendelkeznek [3]. Ezen útmutató alapján igyekeztem egy olyan alkalmazást fejleszteni, amely nemcsak megfelel az általános felhasználói igények mellett a látássérültek igényeiknek is, hanem kifejezetten az ipari környezetben is hatékonyan használható. A fókusz azonban elsősorban a technikai működésre irányult, hogy az alkalmazás kiemelkedő teljesítményt és megbízhatóságot nyújtson azoknak, akiknek szükségük van rá.

3. Tervezési folyamat és az alkalmazás átfogó prezentációja

A kutatás technikai része az alkalmazás alapköveinek letételével indul. Az alkalmazás fejlesztésekor kiemelten fontosnak tartottam a stabil alapok létrehozását, melyek lehetővé teszik a strukturált és könnyen karbantartható szoftver kialakítását. Ezzel hosszú távú továbbfejlesztési lehetőségeket biztosítva az alkalmazás és az esetleges új ötletek számára.

3.1 Univerzális tervezési alapelvek

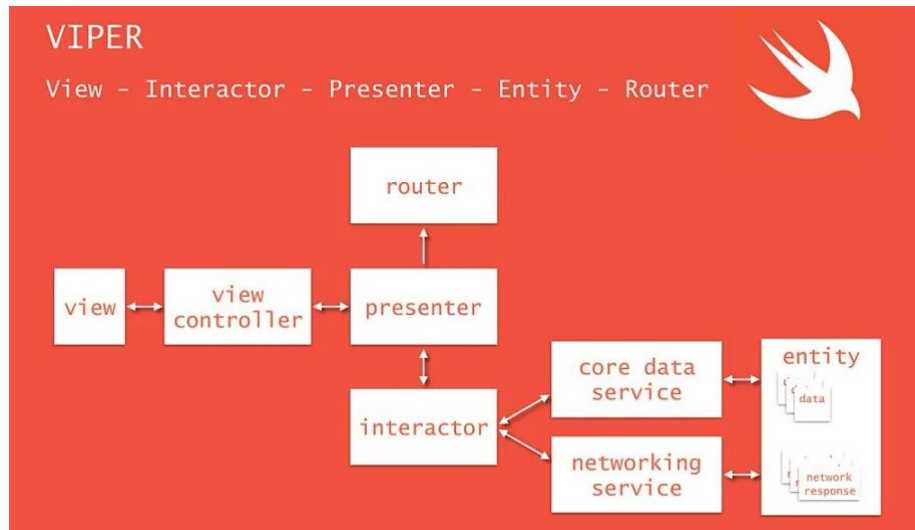
Az alkalmazás tervezése és fejlesztése során a legújabb és legkorszerűbb technológiákat alkalmaztam annak érdekében, hogy a területet a lehető leghatékonyabban és innovatívabban dolgozzam fel. Az új technológiai megoldások és eszközök bevezetése segített a fejlesztés során az alkalmazás teljesítményének és funkcionalitásának növelésében. Az alkalmazást a 2023. szeptember 18-án megjelent iOS 17 operációs rendszerhez fejlesztettem. A továbbfejleszhetőség miatt dokumentáció is készült a forráskódhoz *DocC* [4] formátumban illetve az egységes kódstílus miatt *SwiftLintet* [5] használtam. Az alkalmazás fő része *SwiftUI* keretrendszerrel készült, de néhány helyen *UIKit* komponensek használata is szükséges volt, hogy biztosítsam a funkciók zavartalan működését. Az alkalmazás tervezése során a *MVVM* (*Model-View-ViewModel*) architektúrát választottam kiindulási pontként. A *MVVM* egy olyan tervezési minta, amely segít elkülöníteni az alkalmazás logikáját és a felhasználói felületet. Ebben az architektúrában a *Model* reprezentálja az adatokat és az üzleti logikát, a *View* a felhasználói felületet, míg a *ViewModel* az összekötő elem a *Model* és a *View* között.



1. Ábra – MVVM (*Model-View-ViewModel*) architektúra [6]

Az alkalmazás fejlesztése során észrevettem, hogy az alkalmazás egyes részeihez a *VIPER* (*View-Interactor-Presenter-Entity-Router*) architektúrából eredeztethető *Interactor* komponens is jól illeszkedne, így az beintegráltam az architektúrába. Az *Interactor* funkciójába tartozik az üzleti logika kezelése és az adatok előkészítése, amelyeket a *ViewModel* használ a felhasználói

felületen való megjelenítéshez. Ennek eredményeképpen az alkalmazás műveletei (pld. hálózati kommunikáció) el vannak különítve és hatékonyan szervezettek, ami növeli a rendszer karbantarthatóságát és tovább fejleszthetőségét.



2. Ábra - VIPER architecture [7]

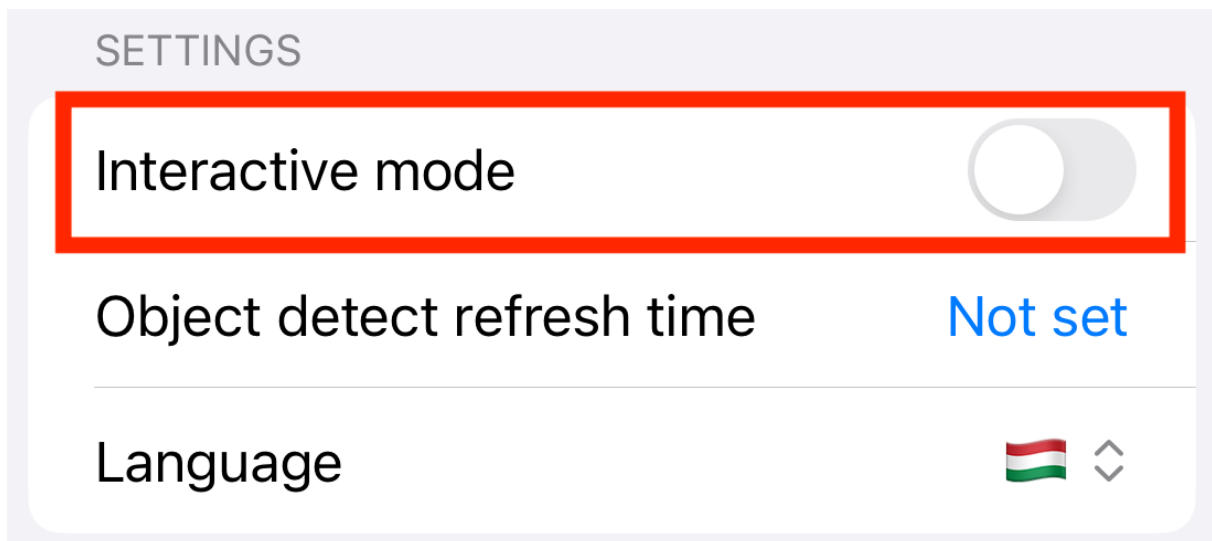
Az aszinkron műveletek hatékony kezelésére a *Combine* [8] keretrendszert alkalmaztam. A *Combine* egy erőteljes eseményvezérelt keretrendszer, amely lehetővé teszi az aszinkron események, adatfolyamok és műveletek kezelését és összekapcsolását. A *Combine* keretrendszer a *Publisher-Subscriber* mintát alkalmazza az aszinkron események és adatfolyamok kezelésére az iOS alkalmazásokban is. Ebben a mintában a *Publisher* felelős az adatok vagy események kiadásáért, míg a *Subscriber* (vagy *Observer*) feliratkozik a *Publisherre* és értesül az általa kiadott adatokról vagy eseményekről. A *Publisher-Subscriber* minta összhangban van a szoftvertervezési elvekkel, különösen az egyik legfontosabbal, azaz a szétválasztással (*separation of concerns* [9]). A *Publisherek* és *Subscriberek* közötti szigorú szétválasztás lehetővé teszi az alkalmazások könnyű karbantarthatóságát és bővíthetőségét. Ezenkívül lehetővé teszi az eseményvezérelt architektúra megvalósítását, amely sok alkalmazásban kritikus fontosságú. A függőségi injekció kezeléséhez pedig a *Resolver* keretrendszert alkalmaztam. A *Resolver* [10] egy könnyen használható eszköz a függőségek injektálásához az alkalmazás különböző részeiben. Ezzel a megközelítéssel lehetővé válik a komponensek független fejlesztése és újrafelhasználása, valamint az alkalmazás struktúrájának jobb karbantarthatósága és bővíthetősége. Az így létrejött architektúra lehetővé teszi az alkalmazás egyszerű bővíthetőségét és a komponensek könnyű cseréjét, amely hozzájárul az alkalmazás hosszú távú fenntarthatóságához és fejleszthetőségéhez. Az alkalmazás fejlesztése

során a hatékony adattárolás és adatkezelés is kulcsfontosságú szempont. Ennek érdekében beépítettem a *Firestore* felhőszolgáltatást az alkalmazásba, amely lehetővé teszi a felhasználói adatok és tartalmak biztonságos tárolását a felhőben. Ezáltal a felhasználók személyes adatai mindig elérhetők és szinkronizáltak az eszközeiken, függetlenül attól, hogy melyik eszközüket használják. A *Firestore* további előnye, hogy lehetőséget biztosít valós idejű adatkommunikációra, így a felhasználók azonnal friss információkat kaphatnak az alkalmazásban. Ezenkívül a *Firestore* által biztosított hitelesítési rendszerrel a felhasználói fiókok és az azokhoz tartozó jogosultságok is könnyen kezelhetők. Az alkalmazásban való lokális adattárolást a *Core Data* [11] keretrendszerrel oldottam meg. Ez egy erős keretrendszer az adatok kezelésére az alkalmazásban, amely lehetővé teszi az adatbázisba történő adatok könnyű be- és kinyerését (lényegében egy *ORM*, amely elfedi előlünk az *SQLite* réteget és annak kezelését). Ez különösen hasznos, amikor offline hozzáférést és gyors adatelérési lehetőséget szeretnénk biztosítani a felhasználók számára.

3.2 Felhasználói élmény tervezése

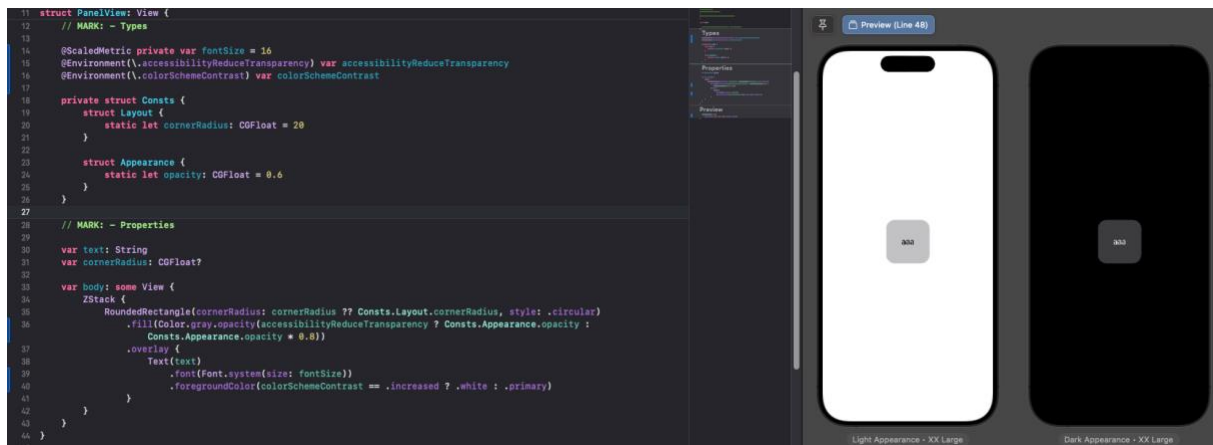
Az iOS környezetben történő látássérült támogatás tervezése és megvalósítása rendkívül komplex folyamatot jelent a fejlesztők és tervezők számára. A cél az, hogy olyan eszközöket és funkciókat vezessünk be, amelyek kiterjesztik a hozzáférést a vak és gyengén látó felhasználók számára, miközben továbbra is fenntartják az átlagos felhasználók számára kiváló élményt. Az egyik kulcsfontosságú aspektusa a *VoiceOver* [12] alkalmazása, amely egy gesztusalapú képernyőolvasó, és lehetővé teszi a látássérült felhasználók számára, hogy hallható visszajelzést kapjanak az iOS-eszközök használata közben. A *VoiceOver* képes értelmezni és navigálni az alkalmazás tartalmát hangutasítások vagy érintéses mozdulatok segítségével. Ez alapvető fontosságú a vak vagy gyengén látó felhasználók számára, akiknek nem áll rendelkezésre a vizuális információkra támaszkodás lehetősége. A *VoiceOver* alkalmazása nem csupán a látássérült felhasználóknak nyújt segítséget, hanem másoknak is, akiknek például rosszullet vagy más környezeti tényezők miatt segítségre van szükségük. Például azok, akik mozgó járműben utaznak, is bekapcsolhatják a *VoiceOver* módot a könnyebb navigálás érdekében. Ezáltal a *VoiceOver* rendszere sokféle felhasználótípus számára nyújt hasznos funkcionalitást, ami javítja az eszközök hozzáférhetőségét. Az eszközök hozzáférhetőségének növelése érdekében azonban nem csak a *VoiceOver* alkalmazása a kulcsfontosságú tényező. Fontos az is, hogy a dinamikus betűméret, a kontrasztos színek és az általánosan jól látható dizájn elveit is beépítsük az alkalmazás tervezésébe. Ezek a módosítások hozzájárulnak a

látássérült felhasználók kényelmes és hatékony használatához, miközben nem befolyásolják negatívan az átlagos felhasználók élményét. A *VoiceOver* rendszere rendkívül hasznos a látássérült felhasználók számára, de egyes esetekben a használata lehet, hogy túl bonyolult vagy felesleges. Ennek felismeréséből kiindulva bevezettem egy alternatív módot az alkalmazásban, amely hasonló funkciót lát el. Ennek a módosításnak köszönhetően a felhasználók számára lehetőség van arra, hogy az aktuális képernyőn elérhető vagy megjelenő információkat felolvassák, amennyiben ezt az opciót beállították az eszközükön. Az új funkció lehetővé teszi a felhasználók számára, hogy kényelmesen és hatékonyan használják az alkalmazást anélkül, hogy a *VoiceOver* használatát kellene elsajátítaniuk, a funkció egy limitáltabb funkcionalitást biztosít, nem támogatja a natív komponenseket csupán a képernyők közötti navigációt és az esetleges használati információ hozzáférhetőségét teszi kényelmesebbé. Ezáltal a hozzáférhetőség és a felhasználói élmény tovább javul, miközben megtartja az alkalmazás funkcionalitását és széleskörű használhatóságát. Az ilyen módosítások és alternatív megoldások hozzájárulnak ahhoz, hogy az alkalmazás minden típusú felhasználó számára elérhető és könnyen használható legyen.



3. Ábra – Interaktív mód

Az eszközök hozzáférhetőségének növelése érdekében azonban nem csak a *VoiceOver* alkalmazása a kulcsfontosságú tényező. Fontos az is, hogy a *dinamikus betűméret*, a *kontrasztos színek* és az általánosan jól látható dizájn elveit is beépítsük az alkalmazás tervezésébe. Amelyre alább látható egy egyszerűbb példa (az alkalmazás egyik nézet komponense):



4. Ábra – Hozzáférhetőségi változók bemutatásának használata *SwiftUI*-ban

Ahol a `@ScaledMetric` “*Property Wrapper*”-el [13] érhetjük el, hogy amennyiben a felhasználó a fizikai eszközén a rendszer beállításoknak változtatja az aktuális betűméretet, akkor a rendszer leköveti az applikáción belül. Az `@Environment(\.colorSchemeContrast)` környezeti változó pedig a rendszer aktuális kontrasztját adja vissza, ami alapján befolyásolhatjuk a különböző színeket.

3.3 Az alkalmazás funkcionalitása

Az alkalmazás lehetőséget biztosít autentikációra a *Firebase* segítségével, amely lehetővé teszi a felhasználók azonosítását és bejelentkezését. Két fő típusú felhasználót különböztetünk meg az applikáción belül: segítők és látássérülteket. Az alkalmazás főmenüjében lehetőség van különböző tárgyakvalós idejű keresésre és, azok detektálására a keresés funkcióban. A felhasználók képesek összehasonlítani ezeket a tárgyakat az adatbázisban meglévővel, mérve, mennyire egyeznek meg a képek. Az alkalmazás harmadik funkciója egy térképet kínál, ahol a segítők segítséget nyújthatnak a látássérülteknek, hogy eljussanak egy adott helyre. Emellett a segítők is kérvényezhetik ezt a szolgáltatást, és a rendszer valós időben értesítést küld, amikor a segítő elindul. Az alkalmazás lehetővé teszi objektumok szkennelését és a rajtuk lévő szöveg felolvasását magyarul is (a rendszer képes detektálni a fizikai eszköz aktuális használt nyelvét de akár konfigurálható is). Ezenkívül képes hasonlóságokat keresni azok az objektumok között, amelyek már meglévők az adatbázisban. Az alkalmazás további funkcionalitást biztosít a meglévő tárgyak és azok tulajdonságainak megtekintéséhez, így a felhasználók könnyen áttekinthetik a rendszerüket és azok adatait, valamint szükség esetén azokat fel is olvashatják.

4. Mesterséges intelligencia és gépi látás keretrendszerek iOS környezetben

A kutatás folytatásként érdemes elmélyedni a platformon rendelkezésre álló mesterséges intelligencia könyvtárak világában. Ezek a könyvtárak széles körű lehetőségeket és funkcionalitást nyújtanak, de egyben limitációkat is hordoznak. A megfelelő könyvtár kiválasztása és hatékony kihasználása kulcsfontosságú a sikeres alkalmazásfejlesztéshez. A dolgozat következő szakaszaiban bemutatom az általam fejlesztett funkciók lényegi részeit képező megoldásokat és kódrészleteket, hogy hogyan lehet a platformon elérhető mesterséges intelligencia könyvtárakat integrálni és használni.

4.1 Bevezetés Vision – Core ML

Az *Apple Vision* framework egy erőteljes és komplex eszköz az iOS és macOS fejlesztők számára, amely lehetővé teszi a gépi látás és képfeldolgozás számos aspektusának kivitelezését. Az alkalmazások számára számos funkcionalitást kínál, amelyek segítenek az objektumok és szövegek azonosításában, elemzésében és interpretálásában, megnyitva az utat a kiterjesztett valóság (*AR*), gépi tanulás és képfelismerés területein. A *Vision* [14] framework által kínált alapvető lehetőségek közé tartozik az arcfelismerés és a képfelismerés. Az arcfelismerés képes azonosítani és követni arcokat a fényképeken és videofelvételeken, és lehetővé teszi az alkalmazások számára az érzelmek, kor és egyedi azonosítás elemzését. Ezenkívül az objektumfelismerés lehetővé teszi a konkrét tárgyak és objektumok azonosítását a képekben, segítve az *AR* alkalmazásokat a valós idejű objektumok és modellek elhelyezésében. A szövegfelismerés egy másik kulcsfontosságú aspektusa a *Vision* frameworknek, amely lehetővé teszi az írott vagy nyomtatott szöveg olvasását dokumentumokból vagy képekből. Az *OCR* (*Optical Character Recognition*) technológia segítségével az alkalmazások szöveget olvasnak le és konvertálnak szöveges formátumokká. Ezen kívül a *Vision* framework a képek tartalmának analízisét is lehetővé teszi, például a színek, tárgyak és képek vizsgálatát. A *Vision* framework könnyen integrálható az iOS és macOS alkalmazásokba, és rendelkezik az Apple saját technológiáival, például a *Core ML* [15] és a *Metal* [16] API-kal való összekapcsolhatóságban, ami lehetővé teszi a gépi tanulás és az ellenőrzés fejlett szintű alkalmazásait. Az Apple által kifejlesztett eszközökkel, például a *Core Image*-el és a *Core ML* keretrendszerrel való integrálódás is lehetőséget kínál a *Vision* framework teljes kihasználására. Az egyik legfőbb előnye a *Core ML*-nek az, hogy egyszerű használni, és tökéletesen integrálódik az iOS és macOS ökoszisztémába, annak ellenére, hogy saját modellek betanítására és alkalmazására is

lehetőséget biztosít. A modellek olyan keretrendszerrel való együttműködést tesznek lehetővé, mint például a *Vision* framework vagy a *Natural Language* framework, amelyek lehetővé teszik a gépi látás és a nyelvi elemzési feladatok gyors végrehajtását. A *Core ML* támogatja a különböző gépi tanulási modellek formátumait, például a *TensorFlow*, az *ONNX* és a *scikit-learn* által használtakat. Ezáltal a fejlesztők széles körű modelleket hozhatnak létre és használhatnak az alkalmazásaikban. Az Apple által kifejlesztett konvertáló eszközök is rendelkezésre állnak, hogy könnyen alakíthassuk át a modelleket a *Core ML* kompatibilis formátumokba. A *Core ML* lehetővé teszi az alkalmazások számára a gépi tanulás előnyeinek kihasználását anélkül, hogy az adatokat egy távoli szerverre kellene küldeni. A gépi tanulás a készüléken történik, így biztonságos és gyors eredményeket nyújt. Ez különösen fontos az olyan alkalmazásokban, ahol az adatok védelme és a felhasználói magánélet kulcsfontosságú. Ezen kívül az on-device gépi tanulás gyorsabb eredményeket is eredményez, mivel a feldolgozás a helyi hardveren történik. Ez különösen fontos olyan alkalmazásokban, ahol a valós idejű vagy az alacsony késleltetésű válasz szükséges, például a kamerakép elemzése során. Az *Apple Core ML* és *Vision* keretrendszerei folyamatos fejlődésen mennek keresztül, és évente újabb és újabb funkciókkal gazdagodnak. Ez a folyamatos innováció teszi lehetővé, hogy az iOS és macOS fejlesztők még hatékonyabban használhassák a gépi tanulás és a képfeldolgozás előnyeit az alkalmazásaikban. [17]

4.2 Vision / Core ML – Valós idejű objektum detektálás

4.2.1 Bevezetés

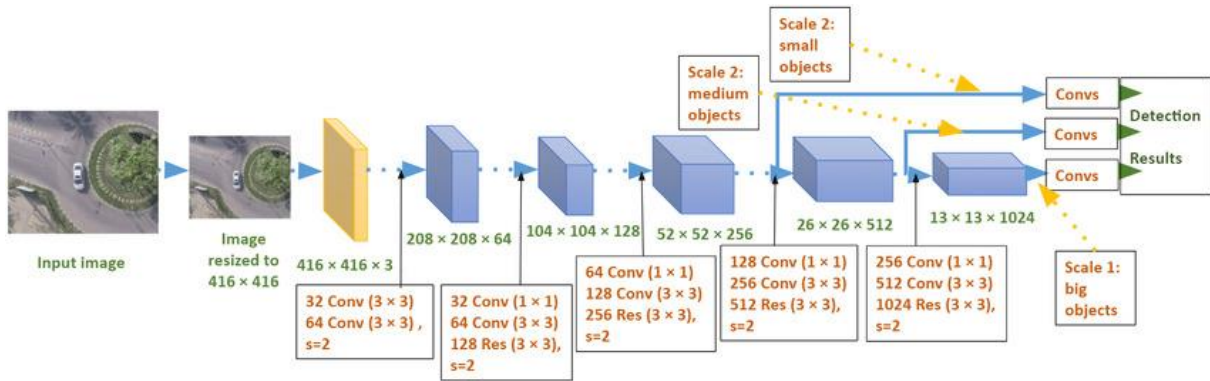
Az adott terület fejlesztése során az elsődleges célom az volt, hogy lehetővé tegyem a valós idejű képfelismerést és elemzést olyan környezetben, ahol számos zavaró tényező jelen van. Ezek a tényezők nagyjából reprezentálják a valós ipari helyzeteket vagy a mindennapi életben előforduló szituációkat, amelyekkel gyengén látó vagy teljesen látássérült emberek naponta találkozhatnak.

4.3 Objektum detektálás

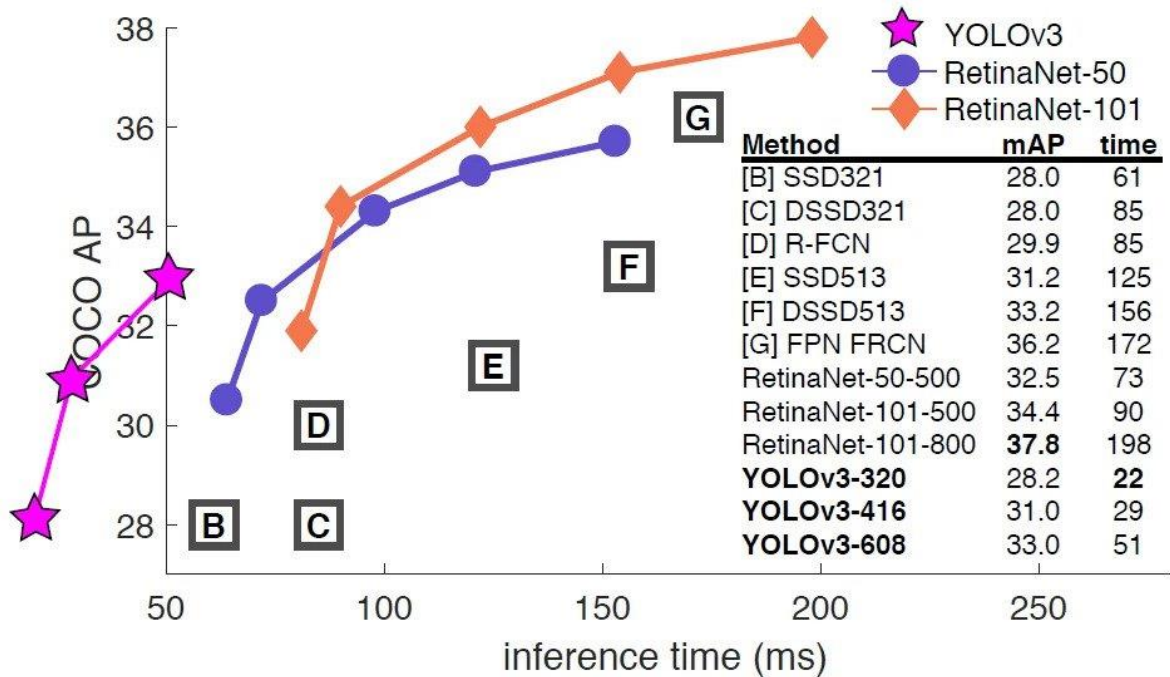
Ahhoz, hogy belátást nyerhessünk a platformon elérhető objektum detektálás lehetőségeihez, először az elméleti majd a gyakorlati megvalósítását szeretném megmutatni konkrét példákon keresztül.

4.3.1 Bevezetés

Az objektumdetektálás terén a *YOLOv3* modellt [18] választottam, mivel rendkívül hatékony és megbízható a feladat elvégzésében. A *YOLOv3* architektúrája és számítási kapacitása.



5. Ábra – YOLOv3 működésének elve [18]



6. Ábra – Modellek összehasonlítás YOLOv3-hoz képest [19]

A *YOLO* egy konvolúciós neurális hálózat (*CNN*), amely valós idejű tárgyfelismerést végez. A *CNN*-ek olyan osztályozó alapú rendszerek, amelyek a bemeneti képeket strukturált adattömegekként képesek feldolgozni és felismerni a köztük lévő mintákat (lásd az alábbi képet). A *YOLO* előnye, hogy sokkal gyorsabb, mint más hálózatok, és mégis megőrzi a pontosságot. Lehetővé teszi a modell számára, hogy a teljes képet vizsgálja a tesztelés idején,

így előrejelzéseikhez a kép globális kontextusát is figyelembe veszi. A *YOLO* és más konvolúciós neurális hálózati algoritmusok az előre meghatározott osztályokhoz való hasonlóságuk alapján "pontozzák" a régiókat. A magas pontszámot kapott régiók pozitív észlelésként jelennek meg abban az osztályban, amellyel a leginkább azonosulnak. Ez a mélytanulási modell lehetővé teszi a valós idejű objektumok felismerését és követését a képeken, és kiváló teljesítményt nyújt olyan kihívást jelentő környezetekben is, ahol számos zavaró tényező jelen van. A *YOLOv3* modell rendkívül rugalmas és alkalmazkodóképes, és lehetővé teszi a felhasználók számára, hogy saját igényeiknek megfelelően finomhangolják és testre szabják az objektumfelismerési feladatokat. Azáltal, hogy ezt a modellt alkalmaztam, sikerült egy hatékony és pontos megoldást kifejlesztenem az objektumdetektálás terén, amely számos alkalmazási területen hasznos lehet a látássérült emberek számára. Az általam választott modellek a *YOLOv3* és a *YOLOv3-Tiny*, illetve *YOLOv3Int8LUT* ezek a modellek a kameraképen vagy képen jelen lévő 80 különböző típusú objektum megtalálása és osztályozására képesek. Az ezek közötti különbségek. A *YOLOv3Tiny* egy kisebb és egyszerűbb verzió a *YOLOv3*-nak. Kevésbé paraméterrel rendelkezik, és gyorsabb, de kevésbé pontos az objektumfelismerésben. A *YOLOv3Int8LUT* egy 8-bites egész számot reprezentál. Így a *YOLOv3Int8LUT* egy olyan változat, amely az int8 típust és előre definiált táblázatokat használja a neurális hálózat értékeinek kódolására és/vagy értelmezésére. A *YOLOv3Int8LUT*-nek a neve alapján int8 típusú értékeket használ, ami az eredeti *YOLOv3* és *YOLOv3Tiny* 32-bites lebegőpontos számokkal dolgozó modelljeivel szemben hatékonyabb memóriahasználatot tesz lehetővé. Az int8 használata kisebb memóriaigényt és gyorsabb kiértékelést eredményezhet. Ugyanakkor a pontos pontosság az ilyen kvantálási eljárásoknál sérülhet. A *YOLOv3Int8LUT*, bár hatékony, általában nem ér el olyan magas pontosságot a detektálásban, mint a teljes *YOLOv3* modell. Az optimális választás tehát függ a konkrét felhasználási esettől. Ha a prioritás a gyors és hatékony feldolgozás, akkor a *YOLOv3Int8LUT* lehet a megfelelő választás. Ugyanakkor, ha a magas pontosság a kulcsfontosságú, akkor a teljes *YOLOv3* modell használata ajánlott. Az eredményeket a következő fejezetben lévő táblázatban láthatóak. Alapvetően nincs túl nagy különbség az eredeti *YOLOv3* és a *YOLOv3Int8LUT* között. Ugyanakkor fontos megjegyezni, hogy a *YOLOv3Int8LUT* gyakrabban próbál mintát venni, amely sajnos ronthatja a minőséget. Még ha a képeken nem is látszik jelentős különbség, azonban a gyakorlati felhasználásban jelentősen nehezebb a használata, sokkal gyorsabb így nincs idő bemérni a keresendő objektumot. Ezért a detektálás minősége és sebessége között egyfajta kompromisszumot kell találni. Fontos szempont lehet továbbá, hogy az általunk kiválasztott modell milyen elérhető információkat nyújt a detektált objektumokról.

4.3.2 Eredmények és gyakorlati megvalósítás

A *YOLOv3* objektumfelismerő modell, amely az észlelt objektumokat részletesen jellemezve teszi lehetővé az azonosításukat és kategorizálásukat. A modell a detektált objektumokat körülvevő négyzetek (*bounding box*) és az objektumok neveinek felismerésével dolgozik, és alapértelmezésben 80 különböző objektum típusra van betanítva. Ennek az integrált megközelítésnek számos előnye van, amelyek hatékony objektumazonosítást és kategorizálást tesznek lehetővé. Az általános képfeldolgozásban a bounding boxok használata segít a detektált objektumok vizuális elkülönítésében és azok pontos lokalizálásában a képeken. A *YOLOv3* ennek a megoldásnak a részeként a modell minden detektált objektumhoz hozzárendel egy bounding boxot, amely pontosan meghatározza az objektum pozícióját a képen. Ezáltal könnyen meghatározható, hol található az objektum a képen, és ennek a pozíciónak az ismeretében további elemzéseket és cselekvéseket végezhetünk. A különböző modellek felhasználásával kapott eredményeket az alábbi táblázatban láthatjuk.



A gyakorlati megvalósítása a következő módon zajlik. Definiálnunk kell egy kéréseket feldolgozó osztályt, amely elfogja végezni a felolgozást, ehhez *VNRequest* [20] kell használnunk, ami egy absztrakt őosztály a kérések analizálásához. Ezt az alábbi módon tehetjük meg. `@Published var requests= [VNRequest]()`, a *Published* egy olyan típus (*Property wrapper*), amely egy attribútummal jelölt tulajdonságot tesz közzé, ez a *Combine* framework egy része. Miután megvannak a kéréseket majd tárolandó tömb, felkonfigurálható a Vision rész.

@discardableResult

```
private func setupVision () -> NSError? {
    var error: NSError?

    guard let modelURL = Bundle.main.url (forResource: MODEL_NAME,
        withExtension: MODEL_EXTENSION) else {
        return NSError (domain: "\(#file)",
            code: -1, userInfo: [NSLocalizedStringKey:"Model file is missing!"])
    }
    do {
        let visionModel = try VNCoreMLModel (for: .init(contentsOf: modelURL))
        let objectRecognition = VNCoreMLRequest (model: visionModel) { request,
            capturedError in
            DispatchQueue.main. async {
                if let result = request. results {
                    if result. count != .zero {
                        self. handleResults(result)
                    }
                } else {
                    error = (capturedError as? NSError)
                }
            }
        }
        self. requests = [objectRecognition]
    } catch let error as NSError {
        print("ERROR: \(String(describing: error.localizedDescription))
            | AT: \(#file),\(#function), \(#line)")
    }

    return error
}
```

Ezek után csak a feldolgozó függvényt kell megírni, amely végzi a kiválasztást és a szelektálást a minimum konfidencia értéknek megfelelően.

```

private func handleResults(_ results: [Any]) {
    for observation in results where observation is VNRecognizedObjectObservation {
        guard let objectObservation = observation as? VNRecognizedObjectObservation
        else {
            continue
        }

        let topLabelObservation = objectObservation.labels[0]

        guard
        topLabelObservation.confidence > Float(MINIMUM_CONFIDENCE_VALUE)
        else { continue }

        ...
    }
}

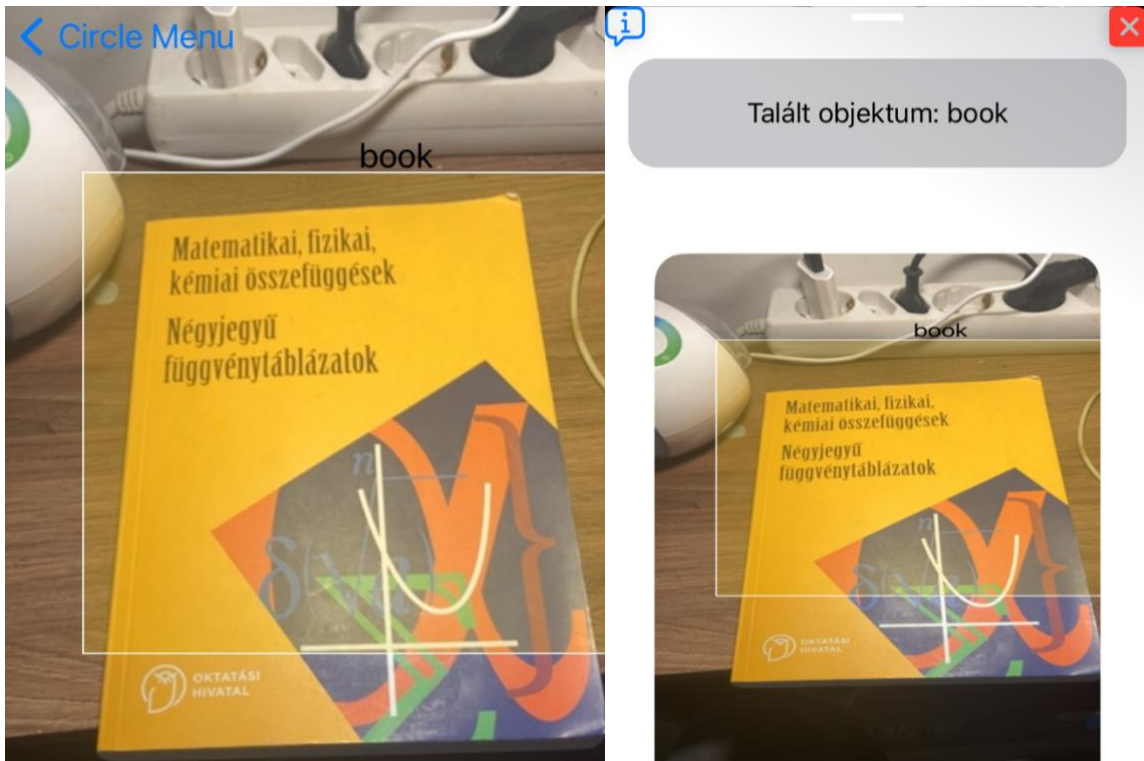
```

A fentebb látható kódrészlet bemutatja, hogyan érhető el objektum detektálás mesterséges intelligenciával a *Vision* keretrendszer segítségével.

4.3.3 Objektumazonosságok felkutatása és hasonlóságok elemzése.

Az objektumok megtalálását követően a kutatás fókuszába kerül a már meglévő képek közötti hasonlóság keresése. Ez azt jelenti, hogy a detektált objektumokat összevetjük a korábban rögzített képekkel, és megpróbáljuk azonosítani azokat a hasonlóságokat, amelyeket két objektum között találunk. Ennek során az algoritmusoknak nagy figyelmet kell fordítaniuk a különböző képek megfelelő sikerességű párosítására és azok pontosságára. A kutatás kettős célkitűzése a pontosság és a megbízható eredmény elérése érdekében az objektumok közötti hasonlóságokra is kiterjed. Az első irányvonal az objektumokon található szövegek közötti hasonlóság vizsgálata, amely elengedhetetlen a pontos azonosításhoz. A kutatás során a képeken megtalált szövegeket különböző algoritmusokkal azonosítom és elemzem. Az összehasonlítás során figyelembe veszem a karakterek számát, a szöveg elrendezését és a tartalmat, hogy maximalizáljuk a találatok pontosságát és megbízhatóságát. A második irányvonal az objektumokon talált szövegek összehasonlítása. Az elképzelt alkalmazási területeken a kutatás az emberi élet sok területén lehet hasznos és segítséget nyújthat. Az eredmények széles körű alkalmazása és az emberi élet számos területén való valószínű hasznosítás a kutatás hosszú távú célja. Az objektumok közötti hasonlósághoz szintén a *Vision* keretrendszerre támaszkodunk. A teljes feldolgozás való idejű információhoz való hozzáférést nyújt, így lehetővé téve az objektumok

azonosítását és követését. Ennek következtében a befoglaló négyzetek nem mindig azonosak, mivel folyamatosan frissülnek a detektálás során. Ez azért fontos, mert a valós idejű alkalmazásokban az objektumok pozíciója és mérete folyamatosan változhat, és a rendszernek dinamikusan kell kezelnie ezeket a változásokat a legpontosabb eredmények eléréséhez. (A hozzáférhetőség növelése érdekében kizárólag a hangerőszabályzó gombok használhatók a funkció elérésére.)



13., 14. Ábra Objektum detektálás folyamat

Az előkészítő lépések után, mint például a *Vision* kezdeti állapotának beállítása és a kérések deklarációja, a rendszer képes megtalálni a képek közötti hasonlóságokat. A *Vision* keretrendszer lehetővé teszi a képek részleteinek összehasonlítását, és azokat azonosítja, amelyek hasonlóságot mutatnak. Ezáltal javítja a rendszer teljesítményét és növeli a megbízhatóságát, amikor az objektumok közötti hasonlóságot keresi.

4.3.3.1 Képben való szöveg keresés

@discardableResult

```
func findIn(image: UIImage) -> [VNRecognizedText] {
    guard let ciImage = CIImage(image: image) else { return [] }
```

```
    var foundTexts = [VNRecognizedText]()
```

```

let requestHandler = VNImageRequestHandler(
    ciImage: ciImage,
    orientation: .upMirrored, options: [:]
)

let request = VNRecognizeTextRequest { request, error in
    if let error {
        print("Error: \(error)")

        return
    }

    guard let observations = request.results as? [VNRecognizedTextObservation] else {
        return
    }

    for observation in observations {
        if let topCandidate = observation.topCandidates(1).first {
            foundTexts.append(topCandidate)
        }
    }
}

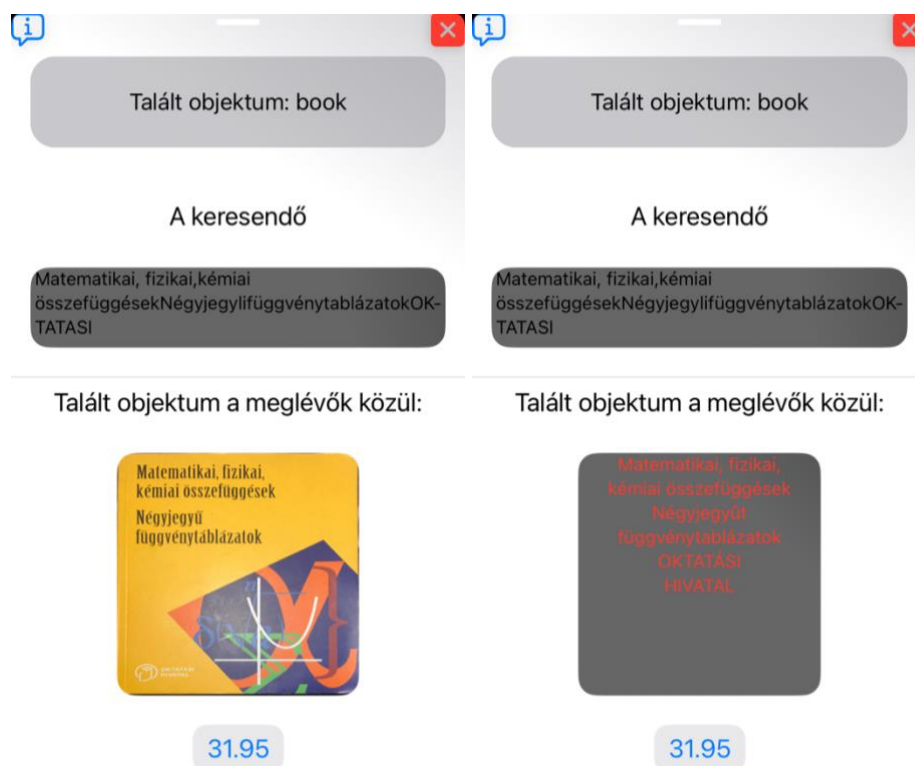
do {
    try requestHandler.perform([request])
} catch {
    print("Error: \(error)")
}

return foundTexts
}

```

A fenti kódrészlet kiemelt szerepet játszik a képfelismerés folyamatában, amelynek célja a képek által tartalmazott szövegek kinyerése és további elemzésre való felkészítése. A *Vision* keretrendszer felhasználásával ez a folyamat hatékonyan valósul meg, és számos előnyt kínál a szövegfeldolgozás területén. A kódrészlet segítségével a rendszer képes megtalálni és kinyerni a képeken található szövegeket. Ennek eredményeként a szöveg a képeken szövegezhető, és készen áll az további elemzésre és feldolgozásra. Ez a képfelismerés egyik alapvető lépése, amely lehetővé teszi a rendszer számára a képek tartalmának szöveges információkká alakítását. A *Vision* keretrendszer továbbá lehetőséget biztosít a felismert szöveg nyelvének meghatározására is. Ez azt jelenti, hogy a rendszer nemcsak a szöveg azonosítására, hanem annak nyelvének felismerésére is képes. Ennek a tulajdonságnak a kihasználása segíti a további elemzés és feldolgozás folyamatát, mivel a szöveg nyelve

befolyásolhatja annak értelmezését és kezelését. Az algoritmus működésének megértése és alkalmazása számos területen előnyös lehet. Azáltal, hogy a rendszer képes a képekben található szövegek felismerésére és nyelvi azonosítására, lehetővé válik az információk automatikus kinyerése és kategorizálása. Emellett segíti a gépi tanulási és mesterséges intelligencia alkalmazásokat a szövegek elemzésében és értelmezésében. Az alábbi képek bemutatják az algoritmus működését és a felismert szöveg gyakorlati alkalmazását.



15., 16. Ábra - Objektumon való szöveg és hasonlóság objektum keresés eredménye

A talált szövegek közötti hasonlóság megállapítására a *Levenshtein* [21] távolság algoritmus használatával végeztem el. A *Levenshtein* távolság (*Levenshtein distance*) egy olyan algoritmus, amelyet a karakterláncok közötti hasonlóság mérésére használnak fő célja annak meghatározása, hogy két karakterlánc között mennyi módosítást, beszúrást vagy törlést kell végrehajtani ahhoz, hogy az egyik karakterlánc megegyezzen a másikkal. Az algoritmus működésének lényege a dinamikus programozásban rejlik. A karakterláncok közötti hasonlóság mértékét egy táblázat segítségével számolja ki, ahol minden egyes cella tartalmazza a megfelelő pozíciók közötti távolságot. Az algoritmus sorban megy végig az összes karakteren, és a távolságértékeket iteratívan számolja ki. A fenti képen a szövegek közötti hasonlóság sajnos nem vizuálisan észlelhető, hanem kizárólag hangon keresztül érzékelhető. Ezt egy másik példán keresztül, amely megtalálható a [6.4] alcímben megfigyelhető.

	m e i l e n s t e i n											
	0	1	2	3	4	5	6	7	8	9	10	11
l	1	1	2	3	3	4	5	6	7	8	9	10
e	2	2	1	2	3	3	4	5	6	7	8	9
v	3	3	2	2	3	4	4	5	6	7	8	9
e	4	4	3	3	3	3	4	5	6	6	7	8
n	5	5	4	4	4	4	3	4	5	6	7	7
s	6	6	5	5	5	5	4	3	4	5	6	7
h	7	7	6	6	6	6	5	4	4	5	6	7
t	8	8	7	7	7	7	6	5	4	5	6	7
e	9	9	8	8	8	7	7	6	5	4	5	6
i	10	10	9	8	9	8	8	7	6	5	4	5
n	11	11	10	9	9	9	8	8	7	6	5	4

1. Táblázat - Levenshtein távolság [22]

4.3.3.2 Kép, képek közötti keresése és rendezése

A következő fejezet célja a képkeresés, rendezés és összehasonlítás módszereinek részletes tárgyalása, mind elméleti, mind gyakorlati szempontból, a kutatás mélyebb megértése érdekében. Ezen fejezetben az algoritmusok, technikák és megközelítések ismertetése kerül sor, amelyeket a kutatásom során alkalmaztam és vizsgáltam az objektumdetektálás és az összehasonlítás során. A fejezet célja, hogy részletesen bemutassa az alkalmazott elveket és módszereket, valamint azok gyakorlati alkalmazását a kutatási témában.

4.3.3.2.1 Ismertetés

Ezen szakasz célja az, hogy a meglévő képek között megtalálja azokat, amelyek tartalmilag vagy egyéb szempontból hasonlítanak a referencia képhez. Ez a folyamat komplex lépéseket foglal magában a képek előfeldolgozásától kezdve a hasonlóság metrikák alkalmazásán és a rangsoroláson keresztül a végleges eredmények bemutatásáig. A feldolgozáshoz most a *Vision*-re támaszkodtam. A fenti képen láthatjuk a végeredményt, amely alapján a képek 31.95%-ban egyeznek meg. Bár a képek közötti hasonlóság első pillantásra szabad szemmel megegyezőnek tűnhet, azonban az eltéréseket a fényviszonyok és más tényezők is befolyásolhatják, így a hasonlóság az algoritmus szempontjából nem mindig teljeskörű.

4.3.3.2.2 Gyakorlati megvalósítás és annak magyarázata

A függvény bejövő paraméterei között szerepel egy referencia kép és a már meglévő objektumok adatainak tömbje.

```
func processImages(
    original: CarouselModel,
    contestants: [CarouselModel],
    completion: @escaping (Result<[(model: CarouselModel, index: Int,
    featureprintDistance: Float)], URLError>) -> Void) {
```

```

var ranks = [(model: CarouselModel, index: Int, featureprintDistance: Float)]()

...

ranks.sort { lhs, rhs in
  lhs.featureprintDistance < rhs.featureprintDistance
}

...

switch ranks.count {
case 0:
  completion(.failure(URLError.init(.badURL)))
default:
  completion(.success(ranks))
}
}

```

Ezen paraméterek segítségével a függvény képes összehasonlítani a referencia képet a meglévő objektumokkal és rangsorolni azokat a hasonlóság alapján. Ezután a referencia képnek számítjuk ki az *n*-edik *VNFeaturePrintObservation* [23], ami egy olyan osztály a *Vision* keretrendszerben, amely a képfeldolgozás során kinyert jellemző pontokat és leírókat reprezentálja. Ezek a jellemzők lehetővé teszik a képek közötti hasonlóságok mérését és az objektumok azonosítását. Az *VNFeaturePrintObservation* segítségével két kép közötti hasonlóság mértékét lehet számítani, ami fontos a képfelismerés, az objektum detektálás és a tartalomelemzés során.

```

guard let originalFPO = featureprintObservationForImage(data: original.imageData)
else {
  print("TODO | ERROR | AT: \(#function)")

  return
}

```

Ezt követően a függvény paraméterként kapott objektumokat összehasonlítjuk egymással, a *VNFeaturePrintObservation* segítségével. Ennek során a képek közötti hasonlóságot mérjük meg, és különböző objektumokat rangsorolunk a hasonlóság alapján.

```

for indexAt in contestants.indices {
  let contestantData = contestants[indexAt].imageData
  if let indexAtContestantDataFPO = featureprintObservationForImage(
    data: contestantData) {

    do {
      var distance = Float(0)
    }
  }
}

```

```

try indexAtContestantDataFPO.computeDistance(&distance, to: originalFPO)
  ranks.append((
    model: contestants[indexAt],
    index: indexAt, featureprintDistance: distance))
  } catch {
    print("Error computing distance between featureprints.")
  }
} else {
  print("indexAtContestantDataFPO | IDX:\(indexAt)
    | AT: \(#file) ROW: \(#line)")
}
}
}

```

Az iteráció során minden érkező objektum esetén számítást hajtunk végre a *VNFeaturePrintObservation* és a referencia *VNFeaturePrintObservation* közötti különbség meghatározására. Ennek eredményeképpen megkapjuk az eltérést, ami segít megértetni, mennyire hasonló vagy különböző az adott objektum a referenciához képest, amely után már csak rendeznünk kell az eredményeket és megkapjuk a fentebb látható eredményt.

```

private func featureprintObservationForImage(data: Data) ->
  VNFeaturePrintObservation? {

  let requestHandler = VNImageRequestHandler(data: data, options: [:])

  let request = VNGenerateImageFeaturePrintRequest()
  do {
    try requestHandler.perform([request])
    return request.results?.first as? VNFeaturePrintObservation
  } catch {
    print("Vision error: \(error)")
    return nil
  }
}

```

4.4 Objektum szkennelés

Az objektum szkennelés funkcióhoz már rendelkezésre áll minden olyan technikai előkészület, amely a kutatásomban bemutatatható, így a következőekben ez kerül bemutatásra.

4.4.1 Bevezetés

Az objektum szkennelés (vagy dokumentum szkennelés) egy olyan technológiai folyamat, amelynek célja a fizikai dokumentumok, például papírok vagy szövegek digitalizálása és átalakítása elektronikus formátumba. Az iOS platform számára elérhető *Vision* keretrendszer

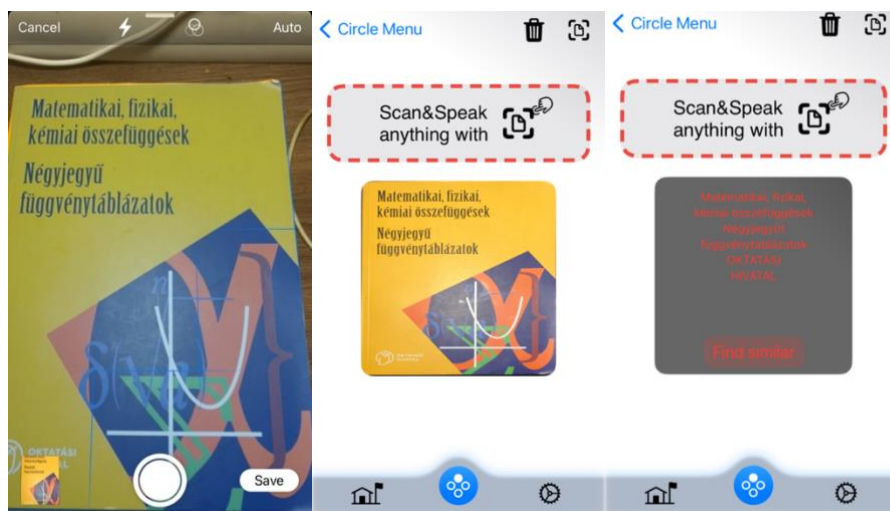
olyan eszközöket és funkciókat kínál, amelyek lehetővé teszik az objektumok hatékony és pontos szkennelését a *Vision Document (ViewController)* szkennelési eszközeinek használatával. Amelyet kibővítettem nem csak dokumentumokra, hanem bármilyen objektumra használható, amely akár tartalmaz szöveget akár nem, de a későbbiekben kereshető a tartalma, a fentebb taglalt funkciókkal.

4.4.2 Megvalósítás

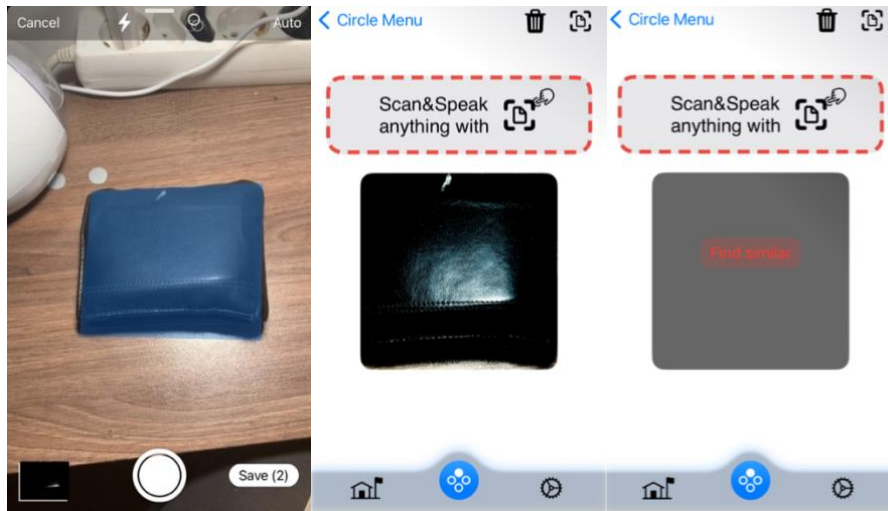
A fentebb említett algoritmusok és függvények itt ebben a szakaszban újrafelhasználásra kerülnek.

4.4.3 Konklúzió, eredmények összefoglalása

A következő ábrákon a szkennelés funkció működését mutatom be, mind szöveget mind szöveget nem tartalmazó esetekre.



17., 18., 19. Ábra - Szöveget tartalmazó objektum szkennelése

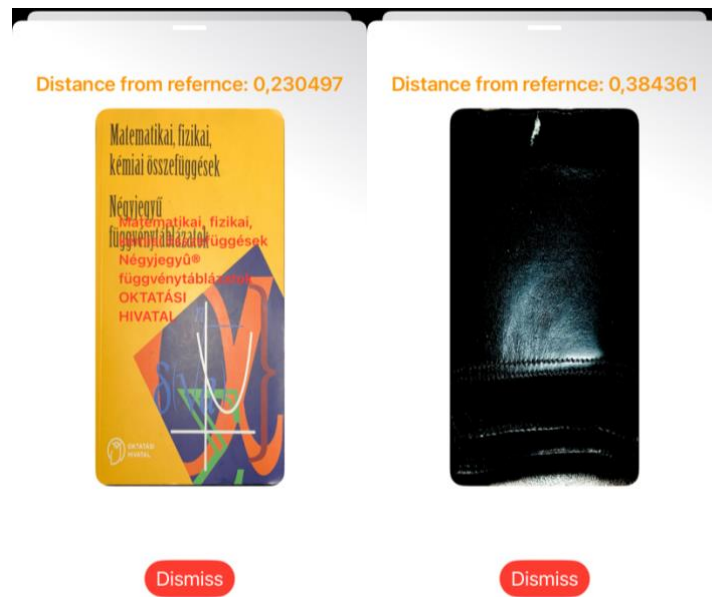


20., 21., 22. Ábra - Szöveget nem tartalmazó objektum szkenerelése

Amennyiben a lokális adatbázis már tartalmaz legalább egy objektumot, elkezdődik a legmegfelelőbb objektum keresése. (Egyébként az alkalmazásban definiált animált általános üres képernyő fog megjelenni, amelyen lévő szöveg *VoiceOver* használatával felolvastatható.) A következő ábrákon a kutatás ezen pillanatában képeket tartalmazó lokális adatbázis tartalmát megjelenítő nézetet és a fentebb lévő könyv és pénztárcához tartozó objektum azonosság találati eredményeket ismertetem.



23. Ábra - Lokális adatbázis tartalmát megjelenítő nézet



24., 25. Ábra - A keresési eredmények a lokális adatbázis elemei alapján

A keresési eredmények részletesen ábrázolják, hogy amikor a rendszer már talál hasonló képeket az adatbázisban, akkor az eredmények között megjelenik a megfelelő távolság értéke. A távolság kiszámításáért felelős rész a [4.3.3.2.2]-ben található, amely megállapítja az adott képhez tartozó távolságot, majd ezt az eredmények között szolgáltatja a modelhez fűzött elemmel.

```
var distance = Float(0)
try indexAtContestantDataFPO.computeDistance(&distance, to: originalFPO)
ranks.append((model: contestants[indexAt], index: indexAt, featureprintDistance: distance))
```

struktúrához majd ez került visszaadásra a completion blokkon keresztül.

A kapott eredményeken a számított *VNFeaturePrintObservation* által mért távolságnak nincs hagyományos mértékegysége, mint például méter vagy más hosszúság egység. Ennek az az oka, hogy ezek a távolságok nem közvetlenül fizikai távolságokat vagy mértékegységeket fejeznek ki. Az általános elv az, hogy minél kisebb a *VNFeaturePrintObservation* által mért távolság, annál hasonlóbba az adott objektumok vagy képek egymáshoz. Ezek az értékek inkább arányosak vagy normalizáltak, és a *Vision* keretrendszer használja őket az objektumok vagy képek hasonlóságának értékelésére. Továbbá a kapott eredményeken jól látható, a már korábban említett fényviszonyok fontossága is.

5. Navigáció és tájékozódás

A látássérült emberek számára a navigáció és a tájékozódás kihívást jelent mindennapi életük során. A vizuális információk hiányában vagy korlátozott jelenlétük mellett a helyes irány és útvonal megtalálása nehezített. A következő szakaszban erről és a megvalósított funkció által kinyitott új lehetőségekről és annak eredményeit osztom meg.

5.1 Térkép

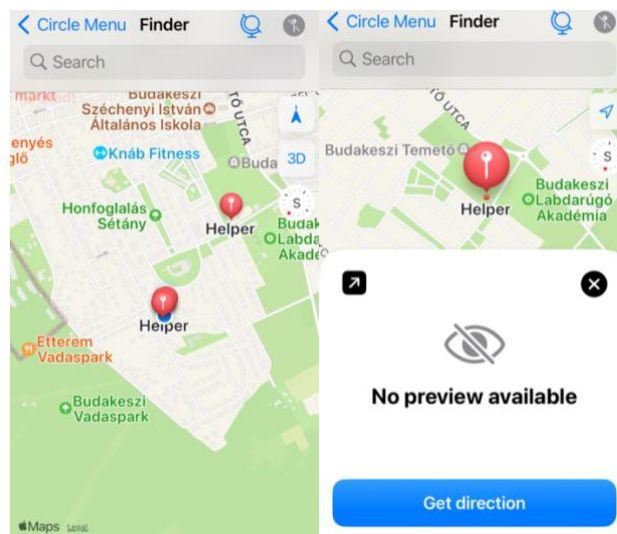
A térkép funkció létjogosultsága a látássérült felhasználók esetében vitathatatlan. Ez a funkció alapvető eszközként szolgál a mindennapi életben, lehetővé téve számukra, hogy felfedezzék és navigáljanak a környezetükben. A térkép funkció a látássérült felhasználóknak lehetőséget nyújt arra, hogy megtalálják a kívánt célpontokat, legyen szó közlekedésről, vásárlásról vagy szabadidős tevékenységekről. Az önkéntes segítők számára is kulcsfontosságú, mivel segítségükkel könnyedén iránytűként szolgálhatnak a látássérült felhasználók számára. A térkép funkcióval valós idejű útvonalak tervezhetők, segítve ezzel a hatékony navigációt és a célok elérését.

5.1.1 Bevezetés

A kutatásom során kiegészítettem a korábban kidolgozott funkciókat egy térkép funkcióval, amelynek célja a látássérült felhasználók támogatása és az együttműködés lehetőségének elősegítése. Ennek érdekében az alkalmazásban implementáltam egy térképes megjelenítést, amely lehetővé teszi a felhasználók számára, hogy térképi adatokat jelenítsenek meg és interaktívan használják. Ugyanakkor a térkép funkció lehetővé teszi az önkéntes segítők bevonását is, akik közvetlenül vagy virtuális úton tudnak segíteni a látássérült felhasználóknak. A térképen megjelenítik saját pozíciójukat és a látássérült felhasználó pozícióját, ami lehetővé teszi az egyszerű találkozást és kommunikációt. Az önkéntesek virtuális útvonaltervezést is használhatnak a látássérült felhasználók támogatására. Ez a térképes funkció további előnye, hogy tartalmazza az általános térkép funkciókat is, például helyek keresését, nevezetességek megjelenítését stb. Mindez egy komplex és felhasználóbarát megoldást nyújt, amely hozzájárul a látássérült felhasználók életminőségének javításához és az önállóságuk növeléséhez. A következőkben ennek a funkciónak a működését mutatom be.

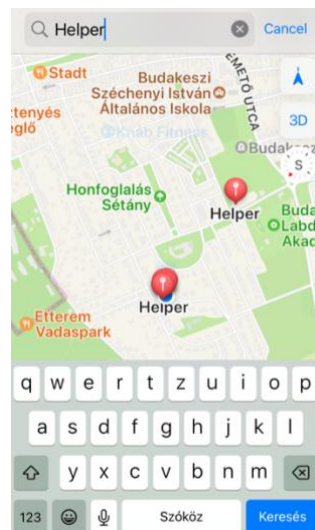
5.1.2 Eredmények bemutatása és magyarázata

A térkép funkció fejlesztése során kiemelten foglalkoztam a kutatási eredmények optimalizálásával és a felhasználói élmény fokozásával. Az iOS 17 új funkcióinak [24] kihasználásával olyan térképes elemeket integráltam az alkalmazásba, amelyek növelik a térkép funkcionalitását és interaktivitását. Az offline térképek használata révén a felhasználók akkor is hozzáférhetnek térképi információkhoz, amikor nincs internetkapcsolatuk. Ez különösen hasznos lehet olyan területeken, ahol a mobilhálózat lefedettsége korlátozott vagy ingadozó. A 3D-s nézet és körülnézés funkciók lehetővé teszik a felhasználók számára, hogy valósághűen felfedezzék a térképen látható területeket. Ez vizuálisan vonzóbb és intuitívabb módja a térképek használatának, és hozzájárul a felhasználói élmény fejlesztéséhez. Emellett az alkalmazás által használt térképi adatok folyamatosan frissülnek, hogy mindig naprakészek legyenek. A kutatási eredmények alapján a térkép funkció további fejlesztéseken esett át, hogy a felhasználók mindenkor megbízható és hasznos térképi információkat kapjanak. A térkép funkció további szempontja, hogy az elérhető kiegészítő területek / adatok (például térkép annotációk) állandóan frissítve vannak. Ezek a frissítések automatikusan történnek, vagy amikor a felhasználó belép a funkcióba, de lehetőség van valós idejű lekérdezésekre is. A fejlesztőnek lehetősége van testre szabni, hogy a térkép funkció mikor frissítse az adatokat, ezáltal az alkalmazás valóságos ipari felhasználhatóságát növelve. Ez a folyamatos frissítési mechanizmus biztosítja, hogy a térképi információk mindig naprakészek legyenek, így a felhasználók mindig a legaktuálisabb adatokat láthatják a térképen. Ezzel a megközelítéssel a kutatási eredmények optimalizálására és a valóságban való hatékony használatra törekedtem.



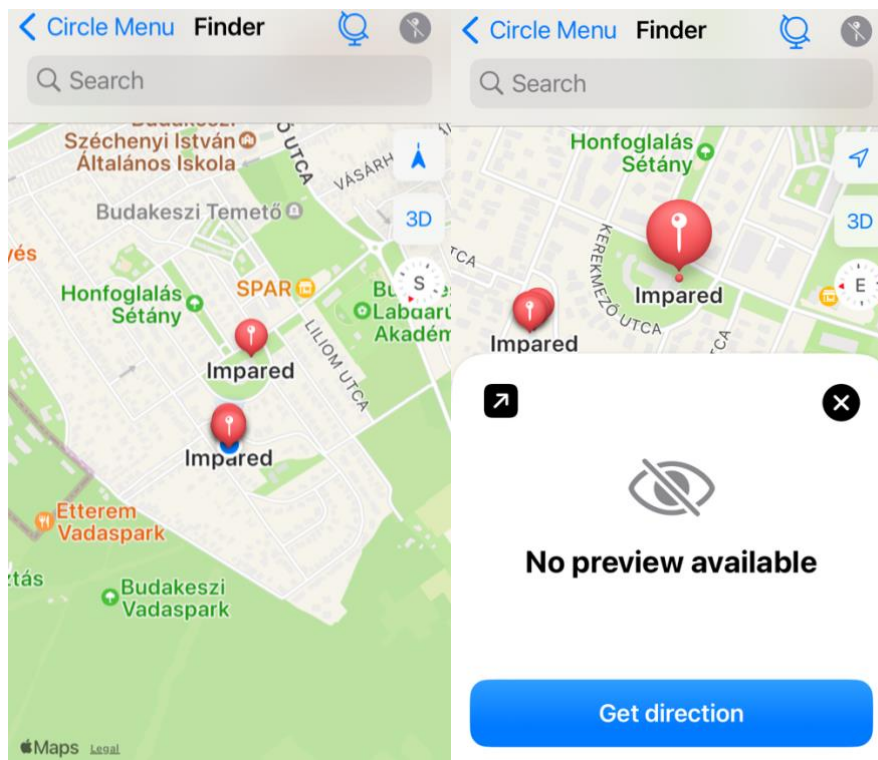
26., 27. Ábra - Látásban korlátozott személyek térkép funkció nézetet

A térkép funkció *VoiceOver*-el való támogatása alapvető fontosságú a látássérült felhasználók számára, mivel ez a technológia lehetővé teszi számukra, hogy információhoz jussanak és navigáljanak a térképi tartalmak között, mivel a térkép funkció megvalósítása natív komponensekből áll így ez alapértelmezett módon elérhető. Az aktuális lokáció mindig elérhető a térképen a követő mód használatával. Ez azt jelenti, hogy a térkép a felhasználó aktuális pozícióját folyamatosan követi, és biztosítja a helyszín folyamatos frissítését, így a látássérült felhasználók mindig tudják, hol vannak és merre tartanak. Így a térkép funkció biztosítja, hogy a használata intuitív és hozzáférhető legyen a látássérült felhasználók számára, és lehetővé teszi számukra, hogy hatékonyan navigáljanak és felfedezzenek térképi tartalmakat. A funkcionalitás kulcsfontosságú eleme a térkép funkcióval történő kommunikációnak, és hozzájárul a látássérült felhasználók önállóságának és mobilitásának növeléséhez. A térképen elérhető egy hatékony keresési funkció, amely lehetővé teszi a felhasználók számára, hogy könnyedén megtalálják az általuk keresett helyeket és tartalmakat. Az úgynevezett "kényelmes keresés" funkció lehetővé teszi a felhasználók számára, hogy egyszerűen és gyorsan keressenek térképi elemek között anélkül, hogy külön keresési gombot kellene megnyomni. Ez a funkció az adatbázis alapján képes a keresést elvégezni, és azonnal megjeleníti a releváns találatokat a térképen. Így a felhasználók egyszerűen rákattinthatnak a keresési eredményekre, és azokhoz a helyekhez vagy tartalmakhoz navigálhatnak. A kényelmes keresés funkció lehetővé teszi a gyors és hatékony információkeresést a térképen, és hozzájárul a felhasználók tapasztalatának javításához.



28. Ábra - Kényelmi keresés

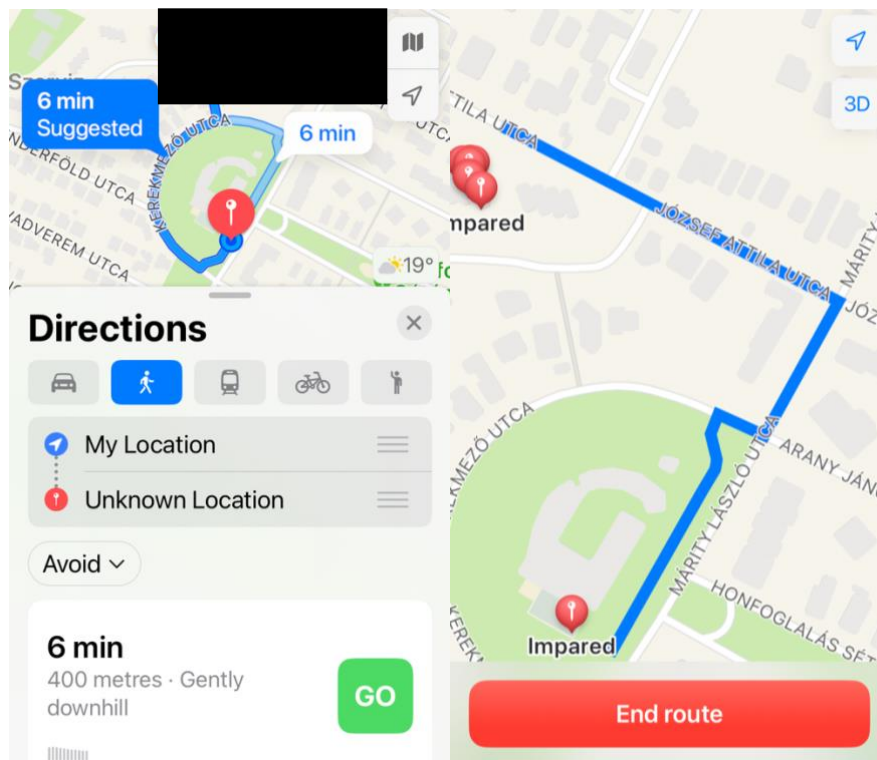
Az önkéntes segítők számára a térkép számos funkcionalitást biztosít, amelyek segítik őket a látássérült felhasználók támogatásában és a térkép használatában. Lehetőséget nyújt a látásban sérült emberekhez való eljuttatásban és ahhoz kapcsolódó útvonal tervezésben. Az útvonal tervezése során az alkalmazás részletesen kiszámolja az optimális útvonalat az aktuális célponthoz vagy helyszínhez, amelyben a “körbenézés” (*look around* [25] [26] nagy segítség).



29. 30. Ábra - Segítők számára elérhető térképes nézet és útvonal tervezési lehetősége

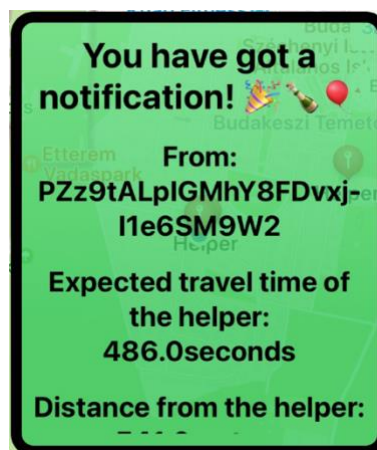
Az alkalmazásban az útvonaltervezés egy kulcsfontosságú elem, amely a látássérült felhasználók és segítők közötti hatékony kommunikációt és navigációt teszi lehetővé. Amikor egy segítő kezdeményezi az útvonaltervezést, az alkalmazás lehetőséget nyújt részletes útvonal tervezésére a látássérült felhasználó számára. Ebben a folyamatban a segítő figyelembe veszi az aktuális helyzetet és a célponthoz vezető legoptimálisabb útvonalat. Ez magába foglalja a különböző útvonaltervezési lehetőségek felkínálását és a döntéshozás támogatását. A térkép funkció teszi lehetővé az útvonaltervezést, ahol a felhasználók könnyedén meghatározhatják az optimális útvonalat céljuk eléréséhez. Komplexebb útvonaltervezési igények esetén a rendszer további lehetőségeket kínál a térkép integrálásával (bal felső sarokban lévő nyíl megnyomása után). Ennek eredményeként a felhasználók testreszabhatják az útvonaltervezést a saját igényeikhez igazítva, és ha egyedi vagy összetett útvonalra van szükség, a térkép integrálása is elérhetővé válik a rendszerben. Az útvonaltervezés és térkép használata együttműködést és

hatékonyságot hoz létre, így a látássérült felhasználók könnyedén és biztonságosan juthatnak el céljaikhoz. A rendszer nyújtotta lehetőségek révén a navigáció és kommunikáció sokkal gördülékenyebbé válik mind a segítők, mind a látássérült felhasználók számára. Az integrált térkép és útvonaltervezés funkciók segítik a felhasználókat a könnyebb és hatékonyabb navigációban, függetlenül attól, hogy milyen céljuk vagy igényük van az adott helyzetben



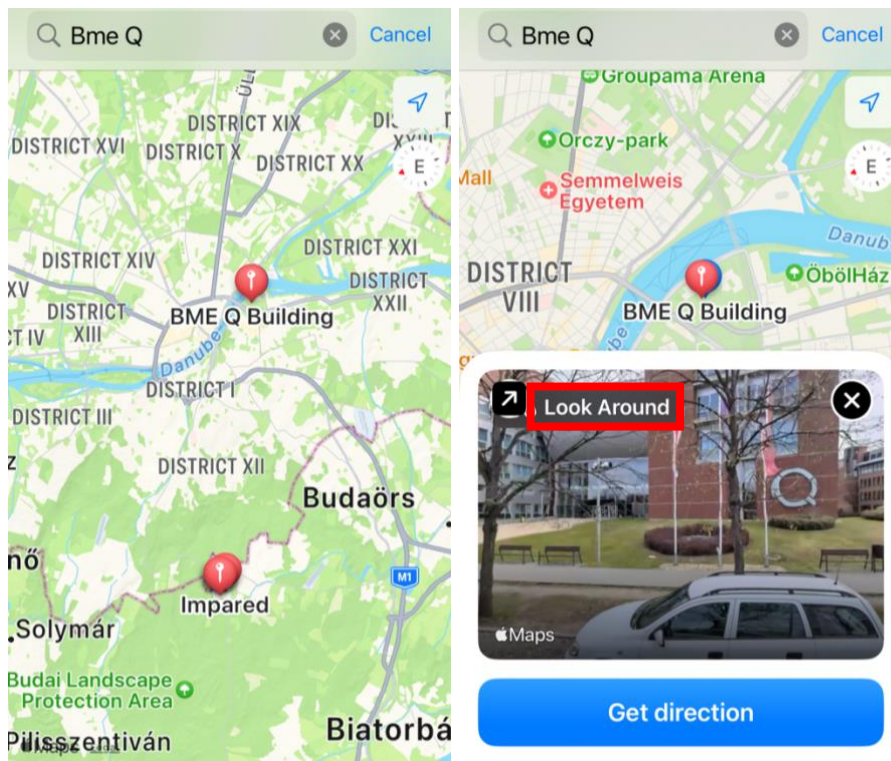
31., 32. Ábra - A baloldali képen a rendszer térkép integrációt, míg jobb oldali képen a térkép funkció útvonaltervezését figyelhetjük meg

Útvonaltervezés után a kiválasztott pozícióhoz tartozó látásban sérült személy értesítést kap a segítség megindulásáról és az ehhez kapcsolódó adatokról.

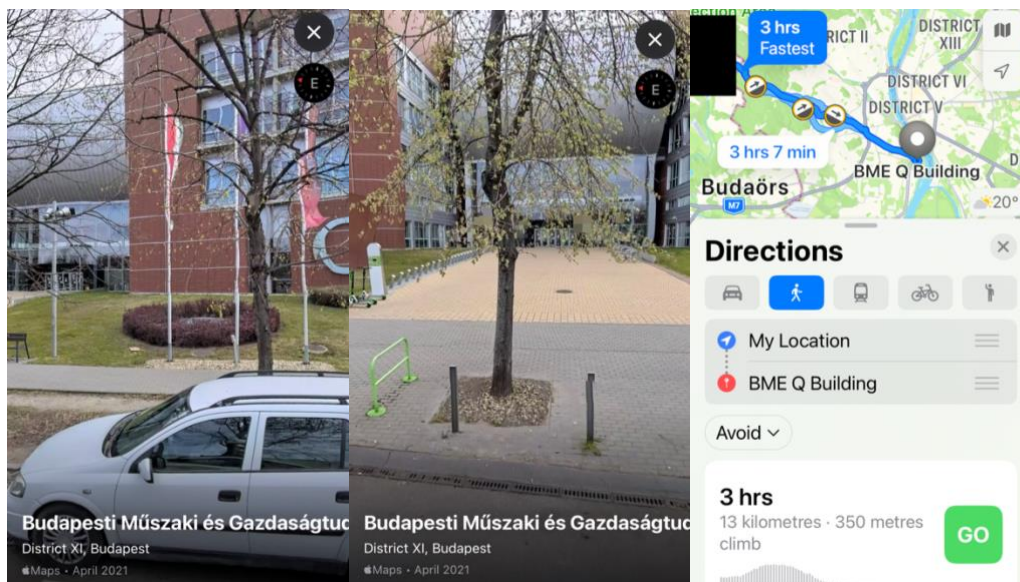


33. Ábra - Értesítés a segítő elidulásáról

Az értesítések az ipari felhasználhatóság miatt valós időben kérdeződnék le, úgy nevezett “pollingozással” vagyis ismétlődően vagy rendszeresen lekérdeződik az aktuális felhasználóhoz ez az attribútum az adatbázisból. Amennyiben a látásban korlátozott célszemély valamely ismertebb helyen tartózkodik, akkor a “körbenézés” (look around funkció is elérhető).



34., 35. Ábra - Ismertebb helyen való tartózkodás



36., 37. Ábra - Körbenézés és integrált térkép használata

Ezek a funkciók közvetlenül hozzájárulnak az emberi segítségnyújtáshoz és az együttműködéshez a látássérült felhasználók és az önkéntes segítők között.

5.2 Összegzés

Az elkészült térkép funkció nem csak a látássérült felhasználók támogatására kínál egy lehetséges megoldás alapját, hanem számos ipari alkalmazásban is hasznos lehet, amelyek hasonló területtel foglalkoznak. Ennek a funkciónak az integrálásával lehetőség nyílik olyan alkalmazások kifejlesztésére, amelyek segítik a felhasználókat a térképek alapján való navigációban és információgyűjtésben. Az ipari megoldások területén ez a funkció lehetővé teszi a térképekkel való hatékony és pontos munkavégzést. Az eredmény egy olyan funkció, amely széles körben alkalmazható, és támogatja a felhasználókat a térképekkel kapcsolatos különféle igényeikben. Emellett a funkció erős alapot biztosít olyan ipari alkalmazásokhoz, amelyek a térképek alapján történő tájékozódást és munkavégzést igényelnek.

6. Összegzés és továbbfejlesztési lehetőség

Az általam elkészített alkalmazás fejlesztése és kutatásának elért eredményei alapján egy kifejezetten látássérült felhasználók számára tervezett iOS alkalmazást sikerült létrehoznom. Az alkalmazás komplex funkcionalitással rendelkezik, lehetővé téve a felhasználóknak valós idejű objektumdetektálást, szövegfelismerést és keresést. Emellett az alkalmazás képes az objektumok közötti hasonlóságok felderítésére és a megtalált eredmények hangos jelzésére. Továbbá lehetőséget nyújt a felhasználóknak saját objektumok keresésére és a látássérült felhasználók és segítőik közötti térképes kapcsolat létrehozásában. Az általam fejlesztett mobil applikációnak köszönhetően a látássérült felhasználók könnyebben tudnak tájékozódni és információhoz jutni, javítva ezzel életminőségüket és hozzáférésüket a környezetükhöz. Az applikáció kiváló alapot nyújt a jövőbeli bővítésekhez és továbbfejlesztésekhez. Az elérhető fejlesztési lehetőségek széleskörűek, és lehetőséget adnak az alkalmazás funkcionalitásának további kibővítésére. Emellett az applikáció kialakítása és architektúrája olyan, hogy könnyen beilleszthetők benne új funkciók és modulok, így alkalmazkodva az igényekhez és a felhasználók visszajelzéseikhez. Ezeket túl rugalmas és jól skálázható, így a jövőbeni változtatások könnyedén megvalósíthatók. Az alkalmazás jelenlegi fejlettségi szintjén a teljesen vak felhasználók számára az applikáció használata bizonyos kihívásokkal jár, és még nem képes teljesen támogatni a szükségleteiket. Fontos hangsúlyozni azonban, hogy a korábbi technikai döntések és az alkalmazás alapstruktúrája miatt kiválóan továbbfejleszhető. Ennek eredményeként a jövőbeni fejlesztések során lehetőség nyílik az akadályok elhárítására, hogy a teljesen vak felhasználók számára könnyedén és hatékonyan használható legyen az alkalmazás. Emellett érdemes hangsúlyozni, hogy az alkalmazás nem csupán a teljesen vak felhasználók számára készült. A gyengén vagy részlegesen látó felhasználók számára is rendkívül hasznos, akár azoknak, akik látási nehézségekkel küzdenek betegségük vagy más tényezők miatt. Az alkalmazás jelenlegi funkcionalitása is jelentős segítséget nyújthat számukra.

6.1 Elért eredmények és elemzése

Az elkészült munka olyan elemeket tartalmaz, amelyek segíthetik a látássérült felhasználókat a mindennapi életben. Az alkalmazás számos kiemelkedő funkciót kínál, amelyekhez a legmodernebb technológiai megoldásokat alkalmazza, és ezáltal képes egyesíteni a látássérült felhasználók mindennapi tevékenységeit, így például támogatja az útvonaltervezést, *VoiceOvert* és saját hangalapú applikáción belüli navigációs lehetőséget, saját objektumok ("kedvenc"

tárgyak) megkeresését a valós idejű objektum keresés / szkennelés funkció segítségével vagy az objektumok felismerését és a szövegek kinyerését majd azok akár hangalapú ismertetését, ami a kommunikáció és az információhoz jutás szempontjából kulcsfontosságú. Az objektumok valós idejű felismerése és szkennelése szintén alapvető fontosságú. Az alkalmazás képes objektumokat felismerni és kinyerni belőlük a szövegeket, amelyeket a felhasználók hallhatnak vagy megjeleníthetnek. Ez számos területen segíthet, például az élelmiszer címkék, táblák vagy közlekedési jelek olvasásában. Az alkalmazás kiemelkedő jellemzője a natív mesterséges intelligencia könyvtárak integrálása, amelyek lehetővé teszik a virtuális asszisztensek és hangalapú kommunikációs megoldások használatát. Az alkalmazás nem csupán az információ és kommunikáció forrásait kínálja, hanem egyedi funkcionalitást is biztosít, amely lehetővé teszi a felhasználók számára a hangalapú kommunikációt saját és a környezetükkel. Az alkalmazás az aktuális technológiai trendeket követi és a látássérült felhasználók teljes körű támogatását nyújtja. Ugyanakkor az ipari és vállalati környezetekben is kiemelkedően hasznos lehet, amire lehet építkezni amennyiben a térképek és objektumok felismerésének és elemzésének képességeit a munkavégzés vagy a tervezés során alkalmazzák. A munka az általános életminőség javítása mellett hozzájárul a látássérült felhasználók önállóságához és hatékonyabb mindennapi tevékenységekhez, saját tárgyaik megkereséséhez de akár gyengén vagy gyengébben látó felhasználók számára is segítséget nyújthat az objektum szkennelés és a rajta lévő szöveg felolvasása.

6.2 Kapcsolódó már létező megoldások

A kutatásom során csak egy megbízható alkalmazást találtam, amely kapcsolódó megoldást kínál, és ez az alkalmazás a “Be My Eyes” [27] [28] nevet viseli. Az alkalmazás hasonlóképpen működik, mint a kutatás során fejlesztett. A működése a következő, bárki, aki 17 éves vagy idősebb, regisztrálhat felhasználóként vagy önkéntesként, a regisztráció után van lehetőség kapcsolatfelvétellel. A hívások összekapcsolása a nappali időzóna és a nyelv alapján történik (melynek várakozási ideje 15 másodperc). Amikor egy hívást fogadnak, élő egyirányú videó indul, amely lehetővé teszi az önkénteseknek, hogy lássák, mi van a felhasználó (látásban sérült személy) kamerája előtt, és szóbeli támogatást nyújtsanak, az alkalmazásnak több, mint 3 millió önkéntessel rendelkezik. Ennek következtében felmerült bennem a kérdés, hogy a tématerület és a funkciók elemzése kiemelten fontos, hisz ezen kívül nem találatam hasonló mobil applikációt iOS platformra.

6.3 Jövőbeli fejlesztési irányok

A jövőbeli fejlesztési irányok szempontjából számos lehetőség és útmutatás felmerül, amelyek hozzájárulhatnak az alkalmazás javításához és bővítéséhez. Az egyik ilyen terület a modell optimalizálása. Az optimalizált modellek lehetővé teszik a kisebb méretű alkalmazást, ami felgyorsítja a letöltést és a futást a felhasználók számára. Ezenkívül az optimalizált modellek kevesebb erőforrást igényelnek, ami energiahatékonyabbá teszi az alkalmazást, nem mellesleg így a valós idejű objektum detektálás is pontosabb és megbízhatóbb adatokkal tudna szolgálni a felhasználó számára. A szerver kommunikációt illetően jövőbeli irányai közé tartozik továbbá egy saját szerverrel való kiegészítés lehetősége. Ennek segítségével a rendszer gyorsabb és megbízhatóbb kapcsolatot építhet ki a felhasználók és a szerver között. Ez lehetővé teszi az alkalmazásnak, hogy a legfrissebb információkat és adatokat szállítsa az felhasználóknak, és még jobb felhasználói élményt nyújtson. Az alkalmazás lokalizációja is fontos jövőbeli irány lehet. Több nyelv támogatásával az alkalmazás szélesebb körű felhasználói bázist érhet el, és lehetőséget biztosít a globális terjeszkedésre. Az alkalmazás lokalizációja a felhasználók nyelvéhez és kultúrájához való alkalmazkodást jelenti, és a felhasználók számára egyéni élményt nyújt az alkalmazás használata során, a szövegfelolvasási funkcióra is vonatkozóan. A térkép funkció még hatékonyabbá és hasznosabbá tehető azzal, hogy kiegészítjük egy videó kapcsolattal vagy egy önálló funkción keresztül is elérhetővé tesszük. A videó kapcsolat lehetővé tenné a felhasználók és segítők közötti valós idejű kommunikációt, ami rendkívül hasznos lenne a problémák megoldásában (hasonlóan, mint a "Be My Eyes"). A látássérült felhasználók az alkalmazáson keresztül megoszthatnák élő videóképet a segítőkkel, akik azonnal láthatnák és értelmezhetnék a környezetet, majd irányítást és segítséget nyújthatnának. Ezen felül, ha a videó kapcsolatot külön funkción keresztül kínáljuk, az lehetőséget teremt az alkalmazásnak arra, hogy további funkciókat is magában foglaljon. Például az alkalmazás kiegészíthető további kommunikációs eszközökkel, így a felhasználók és segítők könnyen cseveghetnek, képeket és videókat oszthatnak meg, vagy akár hangfelismerési lehetőségeket is bevezethetünk a jobb és sokoldalúbb támogatás érdekében. Az ilyen videó kapcsolat vagy önálló kommunikációs funkción keresztül a térkép funkció még interaktívabbá válna, és széleskörűbb segítséget nyújthatna a látássérült felhasználók számára a mindennapi életük során. Az újítások által az alkalmazás még inkább hozzájárulhat a látássérült emberek életminőségének javításához és nagyobb függetlenségükhöz.

7. Elkészült alkalmazás bemutatása

Az elkészült alkalmazás bemutatására youtube videókat készítettem, amelyekkel az alkalmazás telepítését és működését szeretném megmutatni.

- Az alkalmazás szimulátoron való bemutatása és a háttérrendszerek ismertetése. [29]
 - <https://youtu.be/wQh2VjNhVuA>
- Az alkalmazás fizikai eszközön való telepítése és autentikációs folyamat. [30]
 - <https://youtube.com/shorts/ITouEsw8mkQ?feature=share>
- Az alkalmazásban elérhető interaktív mód és VoiceOver rövid bemutatása. [31]
 - <https://youtu.be/x1EKSoDwHgg>
- Az alkalmazás objektum szkennelés mesterséges intelligenciával működő funkciójának bemutatása. [32]
 - <https://youtu.be/SHcgXmA4BQw>
- Az alkalmazás térkép funkciója és használata egy valós példán keresztül. [33]
 - <https://youtu.be/VeV8trq0PBM>
- Az alkalmazás objektum detektálása mesterséges intelligenciával támogatott és a meglévő objektumok alapján keresés és összehasonltás funkció bemutatása. [34]
 - <https://youtu.be/1s1MvMK9JlU>

8. Irodalomjegyzék

- [1] D. T. Alex, "Github," [Online]. Available: <https://github.com/DuT0mi/Thesis>.
- [2] Apple, "apple.com," [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines>.
- [3] Apple, "apple.com," [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/accessibility>.
- [4] Apple, "apple.com," [Online]. Available: <https://developer.apple.com/documentation/docc>.
- [5] Realm, "github.com," [Online]. Available: <https://github.com/realm/SwiftLint>.
- [6] Medium, "medium.com ", "Mastering Android MVVM Architecture: Developers' Guide " [Online]. Available: <https://medium.com/@mutebibrian256/mastering-android-mvvm-architecture-developers-guide-3271e4c8908b>
- [7] Medium, "medium.com ", "Mastering Android MVVM Architecture: Developers' Guide " [Online]. Available: <https://medium.com/cr8resume/viper-architecture-for-ios-project-with-simple-demo-example-7a07321dbd29>
- [8] Apple, "apple.com," [Online]. Available: <https://developer.apple.com/documentation/combine>.
- [9] Wikipedia, "Separation of concerns," [Online]. Available: https://en.wikipedia.org/wiki/Separation_of_concerns.
- [10] hmlongco, "github.com," [Online]. Available: <https://github.com/hmlongco/Resolver>.
- [11] Apple, "apple.com," [Online]. Available: <https://developer.apple.com/documentation/coredata/>.

- [12] Apple, "apple.com," [Online]. Available:
https://www.apple.com/voiceover/info/guide/_1121.html.
- [13] Apple, "apple.com," [Online]. Available: <https://docs.swift.org/swift-book/documentation/the-swift-programming-language/properties/#Property-Wrappers>.
- [14] Apple, "apple.com," [Online]. Available:
<https://developer.apple.com/documentation/vision/>.
- [15] Apple, "apple.com," [Online]. Available:
<https://developer.apple.com/documentation/coreml>.
- [16] Apple, "apple.com," [Online]. Available:
<https://developer.apple.com/documentation/metal>.
- [17] Apple, "apple.com," [Online]. Available: <https://developer.apple.com/machine-learning/api/>.
- [18] ultralytics, "docs.ultralytics.com," [Online]. Available:
<https://docs.ultralytics.com/models/yolov3/>.
- [19] viso, "viso.ai," [Online]. Available: <https://viso.ai/deep-learning/yolov3-overview/>.
- [21] Apple, "apple.com," [Online]. Available:
<https://developer.apple.com/documentation/vision/vnrequest>.
- [22] Wikipedia, "Levenshtein distance," [Online]. Available:
https://en.wikipedia.org/wiki/Levenshtein_distance.

- [23] "stackoverflow.com," [Online]. Available:
<https://stackoverflow.com/questions/4057513/levenshtein-distance-algorithm-better-than-onm>.
- [24] Apple, "apple.com," [Online]. Available:
<https://developer.apple.com/documentation/vision/vnfeatureprintobservation>.
- [25] Apple, "apple.com," [Online]. Available: <https://developer.apple.com/ios/>.
- [26] Apple, "apple.com," [Online]. Available:
<https://developer.apple.com/documentation/mapkit/lookaroundpreview>.
- [27] Apple, "apple.com," [Online]. Available: <https://support.apple.com/hu-hu/guide/iphone/iph65703a702/ios>.
- [28] B. M. Eyes, "Be My Eyes," [Online]. Available: <https://apps.apple.com/us/app/be-my-eyes/id905177575>.
- [29] B. M. Eyes, "bemyeyes.com," [Online]. Available: <https://www.bemyeyes.com/>.
- [30] D. T. Alex, "youtube.com," [Online]. Available: <https://youtu.be/wQh2VjNhVuA>.
- [31] D. T. Alex, "youtube.com," [Online]. Available:
<https://youtube.com/shorts/ITouEsw8mkQ?feature=share>.
- [32] D. T. Alex, "youtube.com," [Online]. Available: <https://youtu.be/x1EKSoDwHgg>.
- [33] D. T. Alex, "youtube.com," [Online]. Available: <https://youtu.be/SHcgXmA4BQw>.
- [34] D. T. Alex, "youtube.com," [Online]. Available: <https://youtu.be/VeV8trq0PBM>.
- [35] D. T. Alex, "youtube.com," [Online]. Available: <https://youtu.be/1s1MvMK9JIU>.