CHALMERS

Utveckling av AR-applikation med AI-glasögon för förbättrad bilkvalitetsuppfattning

Examensarbete inom högskoleingenjörsprogrammet
Datateknik och Mekatronik

Erik Linder, Serok Tunc

Institutionen för data- och informationsteknik

CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2025
www.chalmers.se

EXAMENSARBETE 2025

Utveckling av AR-applikation med AI-glasögon för förbättrad bilkvalitetsuppfattning

Erik Linder

Serok Tunc

CHALMERS

Institutionen för data- och informationsteknik

CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2025

Utveckling av AR-applikation med AI-glasögon för förbättrad bilkvalitetsuppfattning
Erik Linder
Serok Tunc

Handledare på Chalmers: Sakib Sistek, Institutionen för data- och informationsteknik
Handledare på Intended Future: Kostas Stylidis, VD för Intended Future
Examinator: Nicholas Smallbone, Institutionen för data- och informationsteknik

Examensarbete 2025
Institutionen för data- och informationsteknik
Chalmers Tekniska Högskola
SE-412 96 Göteborg
Telefon +46 31 772 1000

Omslagsbild: AI-glasögon “Frame” som använts för detta projekt [41].
Göteborg 2025

Utveckling av AR-applikation med AI-glasögon för förbättrad bilkvalitetsuppfattning
Erik Linder
Serok Tunc
Institutionen för data- och informationsteknik
Chalmers Tekniska Högskola

Sammanfattning

Detta projekt undersöker ett alternativt sätt att interagera med en AI-klient utvecklad av startupföretaget Intended Future, som analyserar bilder på bilar och returnerar en uppskattning av upplevd kvalitet. Det nuvarande sättet att använda klienten innebär att användaren manuellt tar ett foto, till exempel med en mobiltelefon, och laddar upp det via ett webbgränssnitt för att få AI:ns textbaserade tolkning. För att möjliggöra en mer sömlös och handsfree-interaktion har projektet utvärderat att istället använda AR via AI-glasögon som ett alternativt sätt att interagera med en sådan AI-klient.

En mobilapplikation utvecklades för att kommunicera med Frame, ett par open-source AI-glasögon från Brilliant Labs. Applikationen parar med Frame via Bluetooth och låter användaren ta bilder med glasögonens inbyggda kamera. Bilderna visas i mobilens gränssnitt, skickas till ett API, och den resulterande texten returneras och kan presenteras via Frames display, via mobiltelefonens gränssnitt, och/eller genom text-to-speech. Användaren kan även välja mellan olika interaktionssätt, som att ta bilder genom att ta på Frame eller använda en knapp på mobiltelefonens gränssnitt, välja mellan att använda Frames kamera eller mobilkameran, eller välja om texten ska visas i Frames display eller inte.

Syftet är att utvärdera användbarheten av denna typ av interaktion genom användartester. Testresultatet visade en preferens för att ta bild genom att ta på glasögonen framför att använda en knapp på mobilapplikationens UI, samt en preferens att ta del av svarsinformationen i form av text via glasögonens display.

Abstract

This project explores an alternative method of interacting with an AI-client developed by the startup company Intended Future, which analyzes images of cars and returns an estimation of perceived quality. Currently, users manually take a photo, typically with a smartphone, and upload it via a web interface to receive the AI's text based interpretation. To enable a more seamless and hands-free interaction, the project evaluated the use of AR via AI-glasses as an alternative way to interact with the AI-client.

A mobile application was developed to communicate with Frame, a pair of open-source AI-glasses from Brilliant Labs. The application pairs with Frame via Bluetooth and allows the user to take pictures using the built-in camera in the glasses. These images are displayed in the app interface, sent to an API, and the resulting text can be presented via Frame's display, the smartphone interface, and/or through text-to-speech. The user can also choose between different interaction modes, such as capturing an image by tapping the glasses or by pressing a button in the app, selecting between using Frame's camera or the phone's camera, and whether or not to show the text on Frame's display.

The aim is to evaluate the usability of this type of interaction through user testing. The test results showed a preference for capturing images by tapping the glasses rather than using a button in the app's UI, as well as a preference for receiving the response information in the form of text displayed on the glasses.

Förord

Vi vill rikta ett stort tack till alla som på olika sätt varit involverade i detta projekt och bidragit till dess genomförande. Ert stöd, engagemang och värdefulla synpunkter har hjälpt oss att nå ett bättre resultat. Ett särskilt tack vill vi rikta till vår handledare Sakib Sistek, vars kunskap och vägledning varit värdefullt under projektets gång.

Erik Linder, Serok Tunc, Juni 2025

Förkortningar

Förkortningar som används i denna rapport finns listade nedan i alfabetisk ordning:

AI - Artificial Intelligence

API - Application Programming Interface

AR - Augmented Reality

HCI - Human-Computer Interaction

HMD - Head-Mounted Display

HUD - Heads-Up Display

HHD - Handheld Display

LLM - Large Language Model

SDK - Software Development Kit

TTS - Text-To-Speech

UI - User Interface

UX - User Experience

Innehållsförteckning

Figurer

1. Inledning (bakgrund)

Intended Future är ett företag som utvecklar AI-tjänster för att analysera fordondesign. Analysen syftar till att vara ett verktyg för att bättre möta den potentiella kundens förväntningar, och ge svar på hur en kund uppfattar kvaliteten på den betraktade produkten. Här innefattas skillnader i preferenser mellan olika globala marknader, och vad som passar konsumenter gällande funktioner samt estetik. Detta kan vara till hjälp för att i designstadiet få vägledning i hur kvaliteten av en produkt uppfattas redan innan produkten går ut i marknaden. Alternativt kan det användas för att väga hur diverse kostnadsreduceringar i design skulle påverka uppfattningen av slutprodukten. Uppgiften i detta projekt är att utveckla en app som ska ta företagets tjänster för att analysera fordondesign och applicera det till AI-glasögon. Glasögonen ska via augmented-reality ge användaren information av AI-tjänstens utvärdering av en bils kvalitet, design och estetik i helhet.

1.1 Syfte

Intended Futures AI-agent syftar till att enklare få en objektiv uppfattning av fordondesign. Istället för att komma i kontakt med människor inom arbetsområdet kan man använda AI-agenten för att få feedback på designen. Den utvecklade applikationen till AI-glasögonen skall hjälpa användarna att få informationen som behövs via glasögonen direkt. Applikationen kopplar till den redan färdiga LLM-baserade agenten som via glasögonen ska göra det simpelt och effektivt för användaren att få agentens uppskattning av fordonets design med hänsyn till dess funktioner, specifikationer och kvalitetsindikatorer.

1.2 Mål

- **Utforma och utveckla en AR-applikation:** Skapa en användarvänlig augmented reality (AR) app som är förenlig med open-source AI-glasögon och kan visa bilinformation för att förbättra användarens uppfattning av bilens kvalitet.
- **Förbättra användarinteraktion:** Utnyttja AI för att leverera insikter om bilens funktioner, specifikationer och kvalitetsindikatorer, vilket berikar användarens upplevelse i miljöer som bilutställningar.
- **Utvärdera användarens uppfattning:** Bedöm hur AR-applikationen påverkar användarnas uppfattning om bilkvalitet jämfört med traditionella metoder för informationsöverföring.

1.3 Avgränsningar

- All data gällande bilkvalitet tillhandahålls av företaget Intended Future och genereras inte av oss. Vår applikation kopplar samman en AI-agent med AI-glasögonen. All tolkning och bedömning av bilkvalitet utförs således av Intended Futures AI-modell.

2. Augmented Reality Litteraturöversikt

Augmented reality (AR) är teknologi som sammanbinder digital data med verkligheten i realtid. Detta görs genom att överlägga digital information över det användaren ser framför sig i den fysiska miljön. Det skapar en möjlighet att förstärka den verkliga världen genom att använda sig av digital data vilket ger AR-teknologin möjligheten att vara ett hjälpmedel som har potential att revolutionera många arbetsområden, framförallt fordonsindustrin.

Augmented reality som förbinder verkligheten med digital data skiljer sig från virtual reality (VR). Med VR är användaren i en simulerad miljö som inte är verklig, medan AR förstärker den verkliga miljön med hjälp av datagenerering som används, dessa ska inte förväxlas [1]. Användningen av AR brukar oftast göras genom Head-up Display (HUD), Head-mounted Display (HMD) eller Handheld Display (HHD). HUD används oftast i fordon genom att visa digital information för föraren medan de kör utan att distrahera dem. HMD visar digital information genom att ha apparater på huvudet/ansiktet som AR-glasögon och AR-hjälm. HHD är oftast surfplattor och mobiltelefoner som visar den digitala informationen genom skärmen och visar den verkliga miljön genom en kamera [1][2].

2.1 Användar uppfattning rörande AR och produktkvalitet

Med AR-teknologin som blir allt vanligare och används mer i vardagen är det viktigt att veta om hur människor uppfattar och påverkas av att använda sig av det. Human-computer interaction (HCI) är ett forskningsområde där människors interaktion med datorer och digitala system undersöks. För att veta vad en användare av till exempel en mobilapp föredrar i design och funktionalitet är viktigt för att tillfredsställa användningen [3]. Med augmented reality dyker det upp nya frågor gällande hur människors upplevelse är med AR:s integration i olika industrier. Mycket av HCI gäller hur människor interagerar med en skärm, som en dator, men eftersom AR används för att förstärka den verkliga miljön med digital information kan upplevelsen vara annorlunda. Det är viktigt att veta om hur AR påverkar människor både i hur användbart det är men även dess påverkan på människors kognitiva/mentala belastning.

Eftersom AR kan användas av konsumenter vid köp av varor för att få en bättre syn på vad det är som ska köpas, hjälper det med beslutstagandet. Med detta kan människor få en

berikande upplevelse av en fysisk produkt [4]. Med AR är potentialen för kund engagemanget mycket större och leder till en positiv påverkan i beslutsfattande [5].

2.2 AR inom fordonsindustrin

Genom att integrera AR med fordonsindustrin förekommer en utveckling att använda och utnyttja möjligheterna AR skapar. Fordonsindustrin är väldigt bred och har många aspekter där AR kan användas. Det finns dock fortfarande många svårigheter kring integrationen med AR trots att det finns en konstant utveckling [6]. Mycket av applikationerna inom fordonsindustrin testas fortfarande och används inte som standard.

Eftersom fordonsindustrin är väldigt bred kan användningen av AR delas in i följande:

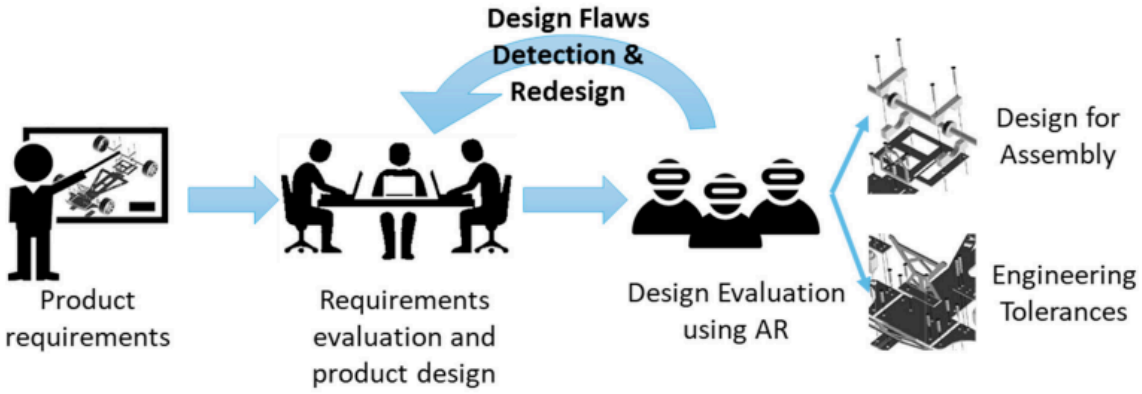
- Design
- Produktion
- Försäljning och marknadsföring
- Användning av konsument

Användningen av AR i de olika områdena varierar och är ett verktyg som förstärker det som redan finns. Att applicera det i till exempel fordonsdesign eller inom försäljning görs på olika sätt, det blir dock densamma i att det resulterar i kostnadseffektivitet, tidsbesparing, ökad kvalitet [1][6][7].

2.2.1 Design

Design av fordon har många aspekter som måste uppmärksammas, allt från dess estetik till dess funktionalitet. För att få den eftersträvade designen måste det konstant bearbetas och tas hänsyn till vad olika konsumenter eftertraktar. Med hjälp av AR kan denna process effektiviseras och till och med förbättras.

När designprocessen görs är prototypen en väldigt viktig fas. Eftersom det är tidskrävande och mindre kostnadseffektivt är det svårt med en stor mängd prototyper. Med hjälp av AR kan denna fas reduceras väldigt mycket. Designers och ingenjörer kan få en virtuell prototyp istället för en fysisk. Prototypen kan konstant modifieras och samtidigt ge en realistisk insikt på storlek, funktion och estetik [8][9]. Med AR som hjälpmedel sammansätts den digitala miljön med den fysiska, vilket hjälper till att upptäcka brister som kan förekomma i de senare stadierna i designprocessen.



Figur 2.1: Förslag på hur designfas kan se ut med AR, hämtad från [9]

AR gör det möjligt för bland annat ingenjörer och designers att enklare kunna samarbeta. Parterna i ett designprojekt över olika geografiska områden får en möjlighet att i realtid kunna se digital data som läggs över det de ser i den fysiska miljön. Det kan vara en 3D Figur av ett fordon men även data som läggs över en fysisk bil. De kan då tillsammans göra modifieringar och ha diskussioner utan att tillsammans vara på plats [9][10].

Projektet i denna rapport fokuserar på bildesign med hjälp av AR. Intended Future är ett företag som bl.a. arbetar med en AI-klient som innehåller data om konsumenters preferenser på bildesign [11]. Genom att koppla klienten till AR-glasögon som har en kamera, blir det möjligt att få digital feedback på en fysisk bil.

2.2.2 Montering och underhåll

Inom fordonsindustrin är mycket av produktionen redan automatiserad. Trots detta finns det fortfarande många delar som inte görs av automatiserade maskiner. AR kan komma i användning i dessa områden och underlätta för operatörer och tekniker.

Vid montering kan AR användas på olika sätt beroende på hur avancerade applikationerna är. Operatörer kan få digitala instruktioner som visar var olika komponenter ska fästas vid montering. Detta kan göras genom pilar eller genom att highlighta olika komponenter vid montering [1]. Det finns delar av monteringar som kan vara komplexa där AR kan vara till hjälp. Ett exempel är justering av olika karossdelar i en bil där tolerans, estetik och funktionalitet är viktigt [12]. I artikeln beskrivs hur operatörer använder sig HMD för att få instruktioner för olika monteringsmoment. AR användes som ett hjälpmedel för operatörerna

och det resulterade bland annat i att bättre upptäcka avvikelser, tidsbesparing och ökad effektivitet.

Användningen av AR behöver inte begränsas till komplexa monteringar och fordonsproduktion. Många monteringsmoment är simpla och upprepande, även bland dessa arbetsmoment kan AR komma till nytta. Eftersom monteringar kan vara monotona blir operatörernas mentala belastning stor. Utöver att uppnå förbättringar i själva produktionen, d.v.s. kvalitet, produktionseffektivitet samt tidsbesparingar, är AR ett verktyg som kan hjälpa tekniker och operatörers kognitiva belastning [13]. I artikeln har ett experiment gjorts i avseendet att undersöka den kognitiva belastningen vid användandet av AR för de anställda. Det visade sig att den kognitiva belastningen minskade med 10% vid användning av AR. Med detta visas att användningen av AR inom olika industrier har en positiv påverkan på användarna samt att det ger en förbättring för både produktion och de anställdas välmående.

Underhåll och montering inom fordonsindustrin kan vara komplexa och krävande i specifika kunskaper. Då utveckling av maskiner och annan teknologi inom fordonsproduktion konstant utvecklas blir processen allt mer komplex. Detta ger en begränsning till antalet kunnande inom de komplexa områdena då det kan bli för dyrt och tidskrävande att träna upp och lära ut de olika arbetsmomenten. För att underlätta träningsprocessen kan AR användas [14]. Eftersom augmented reality ger en konstant visuell guide i arbetsmoment kan det implementeras i träning för tekniker. Ett exempel är hur det kan implementeras för introducering av komponenter, funktionssimulering samt montering och demontering [15].

2.2.3 Försäljning och marknadsföring

När ett nytt fordon släpps ut måste det marknadsföras. Marknadsföringen är viktig men inte alltid enkel att göra. Genom att visa fordonets unika funktionalitet och estetik för konsumenter får de en bättre förståelse av hur fordonet är. Även i detta område har AR sin användning.

När fordon visas upp på showroom-miljöer begränsas utrymmet och visning av funktionalitet. Utrymmet kan öka genom att inte ha fysiska fordon, utan holografiska bilder som tydligt visar upp bilens estetik. Men AR kan också sammansmältas med fysiska fordon genom att få tillgång till olika färger, visa upp fordonets funktionalitet och även ge interaktiva demonstrationer. Eftersom konceptet av AR är att förstärka den verkliga miljön appliceras det i marknadsföringen på samma sätt, den förstärker showroom-miljön [8][16][17].

Ett exempel är Toyotas AR-app som kan användas av konsumenter på deras bilar [18]. Genom att använda appen ger det konsumenter en möjlighet för bättre förståelse av hur fordonet fungerar. Användaren får interaktiva element som visar olika detaljer på bilen och ger relevant information.

2.2.4 Användning av konsument

Integreringen av AR i fordon har blivit allt vanligare och fortsätter att utvecklas. Teknologin fungerar som ett hjälpmedel och underlättar upplevelsen för förare som bidrar till bättre och säkrare körning [19].

Användningen i dagsläget är främst HUD som visar upp hastighet och navigation. Detta görs genom en display för föraren att se, men som är sammansmält med verkligheten. Föraren ser alltså den digitala informationen på fordonets framruta. Navigationen visar upp pilar och detaljer på var fordonet ska köra. Eftersom det måste göras på ett säkert sätt för att inte distrahera och täcka för mycket av förarens synfält är de digitala bilderna små. Föraren kan därmed köra fordonet säkert och med en förbättrad upplevelse. Fördelen med användningen av simpla HUDs är att föraren inte behöver göra något för att få informationen som visas. Den digitala informationen syns där föraren redan tittar när de kör, men täcker heller inte förarens synfält.

Med hjälp av överläggning av digital information gör det möjligt för föraren att köra mer säkert. Varningar med viktiga detaljer kan visas upp utan att behöva påverka var föraren tittar medan de kör. Med mer avancerad teknologi kan AR även användas till advanced driver-assistance system (ADAS). Ett exempel är att ha system som gör det möjligt att få information om hinder och trafikanter. Med sådana system kan den viktiga informationen delas med föraren genom AR och även visa upp relevant data som kan göra det enkelt för föraren genom att få vägledning [20].

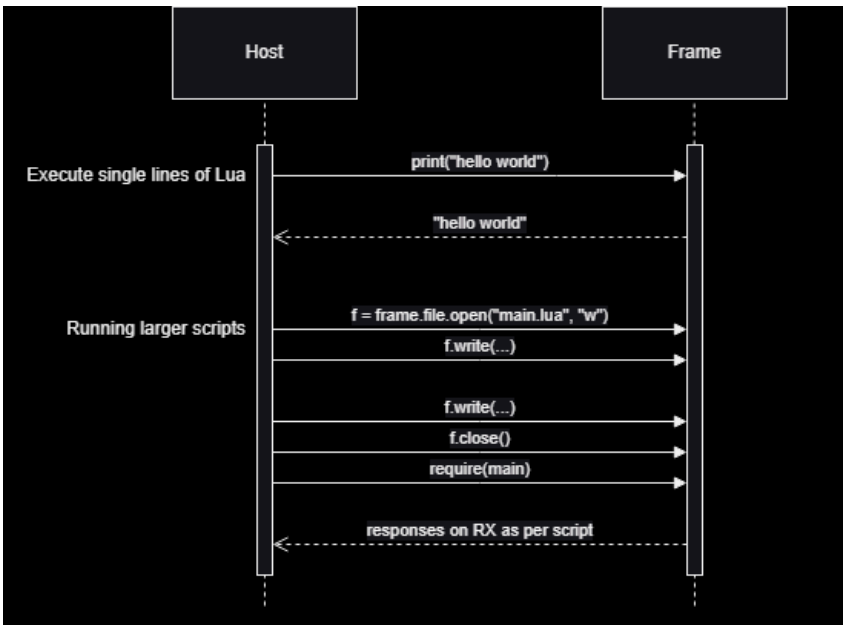
3. Teknisk Bakgrund

3.1 Frame AI-glasögon från Brilliant Labs

Frame [21] är ett par smarta glasögon utvecklade av företaget Brilliant Labs. Glasögonen är utrustade med en 640x400 OLED mikrodisplay som användaren ser genom ett prisma i den högra linsen, en 720p RGB-kamera placerad i glasögonens brygga samt en mikrofon för att uppfatta tal. Utöver detta har Frame en triaxiell accelerometer, vilket möjliggör registrering av rörelser. Detta kan användas för att låta användare interagera med glasögonen genom att tappa på dem. Brilliant Labs Frame utgör ett open source-alternativ inom kategorin AI-glasögon på marknaden.

Glasögonen kommer utrustade med en multimodal AI-assistentapp som heter Noa, där bearbetning av text-, ljud- och bilddata sker via ett externt Application Programming Interface (API). Användarna är dock inte begränsade till att endast använda Noa. Brilliant Labs uppmuntrar användare att utveckla egna applikationer, exempelvis för att koppla glasögonen till andra AI-klienter beroende på syftet med applikationen.

Glasögonen kommunicerar via Bluetooth 5.3 till eller från en dator- eller mobiltelefonapp. På glasögonen exekveras vad som kallas en “main event loop” som kör events baserad på lagrad Lua-kod i Frame, och på applikationen som kopplar till Frame exekveras en “main application loop” som hanterar applikationslogiken. Mellan dessa separata program sker kommunikation genom asynkrona meddelanden via Bluetooth som kallar olika funktioner.



Figur 3.1: Sekvensdiagram för Frames exekvering av Lua-kod. Hämtad från [21]

Vid länkning till glasögonen från en app sker en initial injektion av Lua-kod som glasögonen använder för att till exempel visa textsträngar eller enklare bilder på sin OLED-mikrodisplay. Ett flertal rekommenderade SDKs existerar för utveckling av dator- eller mobiltelefonappar, där frameworket Flutter rekommenderas för utveckling av Android-appar.

I Brilliant Labs Software Development Kit (SDK) för utveckling i Flutter finns kodbibliotek för kontakt på lägre nivå med Frame, och även bibliotek med standardfunktioner för utveckling av Android/iOS-applikationer med Frame [22]. Dessa kommer med exempel av mobilapplikationer som använder dessa funktioner, som kan användas för vidare utveckling av mer avancerade applikationer. Två sådana bibliotek och exempel som varit centrala för detta projekt är simple_frame_app och frame_vision_api, bägge utvecklade av CitizenOne hos Brilliant Labs [23][24][25][26][27].

3.2 Flutter och Dart

Flutter är ett open-source framework utvecklat av Google som lanserades 2017. Flutter designades för att skapa kraftfulla och snabba web-applikationer, desktop-applikationer (macOS, Windows och Linux), och mobilapplikationer (iOS och Android). Därmed kan samma kodbas användas för att kompilera program till många olika målplattformar [28][29][30][31].

Flutter-appar skrivs i programmeringsspråket Dart. Dart är ett objektorienterat språk med klassbaserad struktur liknande språk som Java och C# [32]. Dart har stöd för så kallad “hot reload”, vilket innebär att utvecklare kan uppdatera källkod och genast se resultatet i en redan körande app vid testning [30]. Dart kan kompilera maskinkod till ARM, x64, eller RISC-V för desktop-, backend-, eller mobila applikationer [30].

En av Flutters mest framträdande egenskaper är dess sätt att strukturera användargränssnitt (UI). Användargränssnittet byggs upp genom att så kallade widgets arrangeras i en trädlik hierarki. Detta tillvägagångssätt möjliggör en snabb och flexibel design samt omstrukturering av gränssnittet, vilket underlättar utvecklingsarbetet för programmeraren [29][30].

3.3 Lua

Lua är ett programmeringsspråk och open-source programvara som skapades i Brasilien 1993. Språket är lättviktigt och kan enkelt integreras med olika program och system, detta gör att den ofta används som ett inbäddat skriptspråk i samband med andra programmeringsspråk. Det är enkelt att använda Lua för att utvidga program skrivna i andra programmeringsspråk. Användningen av Lua i applikationer tar inte mycket utrymme eftersom den är liten men fortfarande effektiv. Programmeringsspråket gör det möjligt för användare att endast inkludera det som är nödvändigt vid deras specifika tillämpning. Detta är en fördel för hårdvara med begränsade resurser, som till exempel Brilliant Labs “Frame” [33][34].

3.4 Git

Git är en open-source programvara för versionskontroll vid programutveckling, utvecklat 2005 av Linus Torvalds. Syftet med programmet är att följa, göra, eller ångra ändringar i källkoden över tid. Detta görs med hjälp av en så kallad repository, vilket är en samling av alla mappar och filer som ingår i projektet samt historiken av alla tidigare förändringar som gjorts i projektet. Git körs lokalt på datorn, med hela den lokala repositoryns historik tillgängligt lokalt [35].

Ett vanligt sätt att använda Git är att dela in projektet i olika grenar (“branches”) där huvudgrenen av projektet vanligtvis kallas “main”. Användaren kan skapa nya grenar från existerande grenar, vilket ger möjlighet att arbeta på nya funktioner och samtidigt hålla en fungerande version av koden oförändrad i en annan gren. Grenar kan också sammanfogas

till samma gren och man kan gå fram eller tillbaka i historiken av ändringar i olika grenar som laddats till repositoryn [35].

3.5 GitHub

GitHub är en molnbaserad tjänst för att lagra och hantera Git-repositorier, utvecklat av Microsoft. En lagrad Git-repository på GitHub kan därför ha flera medarbetare och kan antingen vara öppen för allmänheten eller privat. Genom att repositoryn lagras online kan varje medarbetare klona en kopia av repositoryns nuvarande innehåll till sin egen dator för att arbeta lokalt med koden, förslagsvis på en egen gren. Ändringar på repositoryn kan sedan skickas tillbaka och sammanfogas med den gemensamma versionen på GitHub [36].

Vid arbete med kod i en Git-repository används många operationer. Ett urval av de vanligaste operationerna, och sekvensen som dessa görs i, kan ses i Figur 3.2. Med “git add” lägger användaren till de filer med ändringar som skall läggas till servern. När de filer som skall läggas till har lagts till används “git commit” med ett meddelande som förklarar vad som ändrats. Vid “git push” uppdateras den centrala repositoryn med de filer som valts. Användare kan också använda “git pull” för att ladda ned den nuvarande versionen av den centrala repositoryn, och använder “git checkout” för att byta mellan olika grenar av repositoryn [36][37].

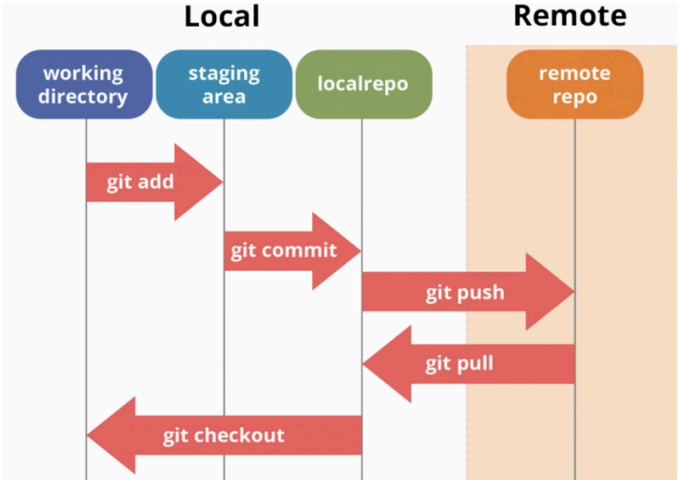


Fig 3.2: Sekvens av GitHub-kommandon för att: Lägga till filer att lagra (git add), Låsa in ändringarna med ett meddelande (git commit), Pusha ändringarna till det centrala repot (git push), Ladda ned ändringar av repositoriet (git pull), samt byta branch (git checkout) [37].

3.6 Android Studio

Android Studio är den officiella utvecklingsmiljön för utveckling av Android appar. Den är därför bra att använda då den har många verktyg och paket som hjälper till med att arbeta med Android applikationer. Det finns många programmeringsspråk som stöds av Android Studio som Java, Kotlin och Flutter/Dart. Det finns stöd för att använda en inbyggd mobiltelefonemulator, och vid utveckling av Android-applikationer i ramverket Flutter finns det möjlighet för realtidskompilering, som båda är bra verktyg för att underlätta utvecklingsarbete [38][39].

4. Metod

4.1 Utveckling av programvara

En mobilapplikation som kommunicerar med Frame utvecklades i ramverket Flutter med programmeringsspråket Dart. Som utgångspunkt användes ett applikationsexempel från Brilliant Labs dokumentation (frame_vision_api), som i sin tur bygger på två öppna bibliotek (simple_frame_app och frame_vision_app). Dessa bibliotek tillhandahåller standardfunktioner för att hantera kommunikationen med Frame och är tillgängliga via GitHub. Utvecklingen av mobilapplikationen genomfördes i utvecklingsmiljön Android Studio.

Eftersom vi under projektets gång saknade tillgång till Intended Futures AI-klient, användes även en enkel Python-server från Brilliant Labs dokumentation (frame_vision_api_impl) för att simulera kommunikationen med ett bildtolkande API. Servern tar emot bilder och svarar med slumpvist utvalda textsträngar ur en lista, i syfte att efterlikna den funktion en AI-klient skulle ha haft.

4.2 Inhämtning av kunskap om ramverk och programmeringsspråk

Då vi saknade tidigare erfarenhet av ramverket Flutter och programmeringsspråket Dart, inleddes projektet med en inlärningsfas för att tillgodogöra den kunskap somkrävdes för att påbörja utvecklingen av mobilapplikationen. Arbetet inleddes med studier av den officiella Flutter-dokumentationen och dess tillhörande interaktiva guider med fokus på ny syntax och Flutters UI- och widget-arkitektur samt hur denna uppdateras och hanteras. Under projektets gång kompletterades kunskapen fortlöpande genom sökningar i dokumentation, på tekniska forum och på sociala plattformar som YouTube och Discord.

4.3 Utformning av mobilapplikationen

En mobilapplikation utvecklades för att para med och upprätthålla bluetooth-kommunikation med Frame. Applikationen kan kommunicera till Frame att visa text på linsens display, samt ta emot bilder tagna med glasögonens kamera. Användaren kan ta bilder med Frame genom att tappa på glasögonen, eller via en knapp på mobilapplikationens användargränssnitt. Det finns även möjlighet att ta bilder med mobiltelefonens kamera istället. Användaren kan mata

in en IP-adress till ett externt API, som applikationen använder som destination för att skicka bilder för utvärdering och tar emot textbaserade svar från API:et.

Applikationen utrustades med ett antal andra funktioner, vilka kan delas in i funktioner som skulle utvärderas av testpersoner, respektive funktioner vars främsta syfte var att underlätta användandet av både applikationen och glasögonen.

4.4 Utvärdering av användaruppfattning

Utvärdering av resultatet gjordes genom användartester med en enkät där svaren utformades efter en Likertskala med fem svarsalternativ. Testpersonerna var studenter på Chalmers campus.

4.5 Arbetsflöde

Majoriteten av utvecklingen genomfördes via parprogrammering, antingen vid samma dator eller via skärmdelning över videosamtal. Vi valde denna metod då den visade sig vara mer effektiv än att arbeta individuellt, särskilt eftersom Flutter var nytt och att vi endast hade tillgång till ett par Frame-glasögon för testning.

Arbetsflödet präglades av flera likheter med ett agilt arbetssätt, såsom versionshantering via GitHub, stegvis utveckling av nya funktioner samt löpande feedback och anpassning av applikationen. Ett strikt agilt arbetssätt användes dock inte, då vi inte arbetade i tidsbestämda sprints. Detta val gjordes för att öka flexibiliteten i tidsplaneringen, och eftersom vi träffades regelbundet under veckorna integrerades de moment som normalt hanteras i sprintmöten istället i vardagsarbetet när det passade.

5 Genomförande

5.1 Planering och samling av teoretisk information

Vid början av projektet planerade vi genomförandet av projektet i olika delmoment i en grov tidslinje, i form av ett GANTT-schema (se GANTT-schema i Appendix B). Detta hjälpte oss att få en bättre överblick över ungefärliga hållpunkter för projektets planering. Detta var nödvändigt eftersom många delar av arbetet var nytt och mycket ny information samlades under projektets gång.
Utöver arbetet med mobilapplikationen som var kärnan i hela projektet skulle vi också samla teoretisk information om AR allmänt och hur det kan användas inom fordonsindustrin. Det vi gjorde var att läsa forskningsartiklar om hur AR används idag och framtida trender gällande arbetet med AR. Det var därför viktigt att kunna arbeta med det teoretiska i samband med huvuddelen av projektet och vi såg därför till att göra en bra planering där arbetet mellan dessa kunde göras.

Ursprungligen var det planerat att vår mobilapplikation skulle kopplas med en AI-klient. Eftersom arbetet med att koppla en AI-klient till en mobilapplikation behövde vi veta mer om hur allmänna LLM:s fungerar och hur dessa kan användas på ett bra sätt. Vi fick som förslag av vår handledare på Intended Future (Kostas) att gå igenom några av Andrej Karpathys filmer på Youtube som bidrog till bra information om hur LLM fungerar och gav oss en bredare insikt på hur det kan användas [40].

5.2 Utförande

Målet var att skapa en applikation som sammankopplar AR-glasögon med en AI-klient för visning av bilinformation. Informationen kommer från AI-klienten, som svar på en bild som tagits med glasögonen och skickats via applikationen. I samråd med handledare Kostas Stylidis beslutades att en mobilapplikation vore mest intressant ur användarsynpunkt eftersom Frame kommunicerar med en applikation via Bluetooth, och glasögonen behöver kontinuerlig kontakt med applikationen för att styra glasögonens funktioner.

5.2.1 Val av programmeringsspråk

Ingen projektmedlem hade tidigare kunskap om Brilliant Labs Frame vid inledningen av projektet, och vi saknade initialt fysisk tillgång till glasögonen. Denna första tid lades till stor del på litteratursökning om ämnet i stort, men också på att undersöka hur Frame fungerar och möjliga vägar att gå för att skapa en mobilapp som kan koppla till Frame.

Brilliant Labs dokumentation var vid projektets start otydlig angående vilka SDK:s som är fungerande eller inte, och dokumentationen har under projektets gång sett många förändringar. Eftersom det var svårt att avgöra vilka verktyg som bör användas för glasögonen var det viktigt att utforska det som passade bäst till detta projekt.

Till en början funderade vi över möjligheten att använda Unity för utveckling av en mobilapplikation, då det fanns en SDK för Unity på Brilliant Labs dokumentation och en av projektmedlemmarna var bekant med Unity sedan tidigare. Det verkade dock vid närmare undersökning som att det inte fanns mycket stöd för utveckling i Unity. Vi övervägde också kort möjligheten att utveckla en webbplats med Web SDK:n tillgänglig från Brilliant Labs dokumentation, för att möjligtvis skapa en webbplats som en mobiltelefon kan använda sig av. Web SDK:n verkade dock inte vara i fungerande skick.

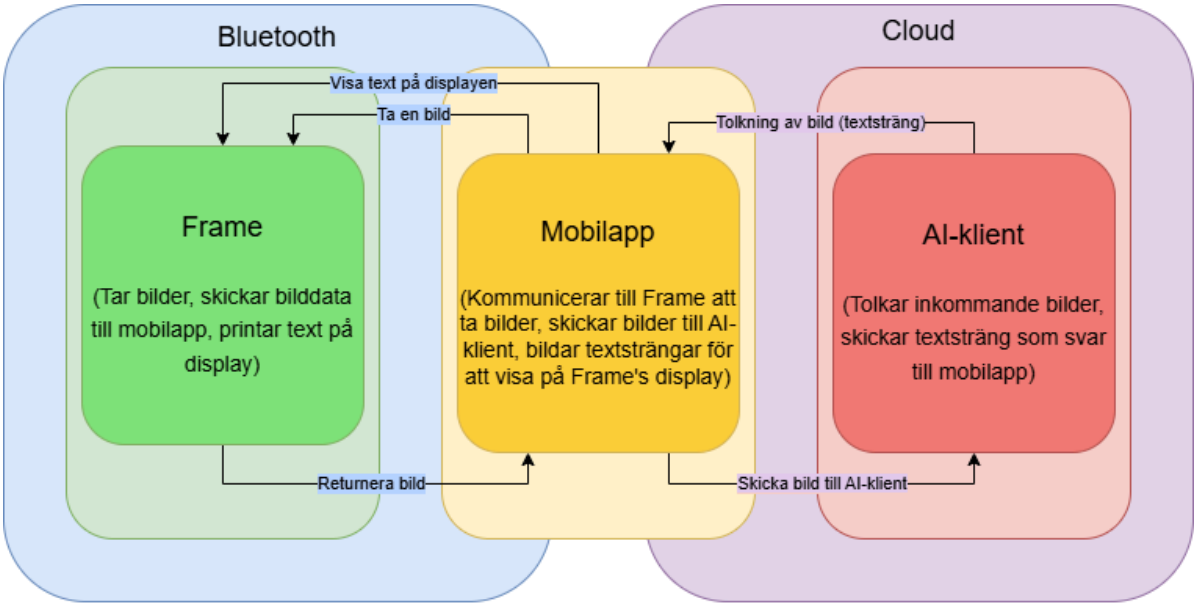
Den SDK som verkade ha mest stöd var Python, för utveckling av desktop-applikationer. Därtill fanns en SDK för utveckling av mobilapplikationer i ramverket Flutter. I denna SDK fanns två bibliotek för anslutning på lägre nivå, samt två bibliotek med många standardfunktioner. På dessa biblioteks GitHub-sida står det att läsa *“This is a work-in-progress personal project with limited support and frequent breaking changes”*, men i brist av andra alternativ beslutade vi att den bästa riktningen skulle vara att använda Flutter och Dart för att bygga en mobilapplikation.

5.2.2 Initial applikationsstruktur

Vår utformade plan var att skapa en mobilapplikation som kontaktar och parar med Frame och som står för “main application loop” i systemet. Mobilapplikationen skickar kommandon och tar emot information via Bluetooth till och från Frames main event loop. Mobilapplikationen skulle också skicka bilder som tas med glasögonen till ett API för en AI-klient som analyserar bilderna, samt ta emot AI-klientens svar gällande mätvärden och övrig analys gällande bilden.

Exakt vilken information som svaret från AI-klienten skulle innehålla eller hur formatet på svaret skulle se ut var inte givet vid projektets start. Därför var planen att tills vidare arbeta under antagandet att svaret skulle komma som en textsträng som innehåller korta mätvärden, som mobilapplikationen sedan skickar till Frame för att visa på dess display.

Den preliminära planeringen av systemets generella arkitektur kan ses i Figur 5.1.



Figur 5.1: Generellt arkitekturdiagram över preliminär plan av applikationens struktur.

5.2.3 Inledande tekniska utmaningar och lösningar

När utvecklingsmiljön var färdigställd och kunde kompilera Flutter-applikationer korrekt, inleddes arbetet med att utveckla en mobilapplikation som kommunicerar med Frame. Vid genomgång och experimentering av Brilliant Labs SDK för Flutter-utveckling lade vi märke till exempelprojektet frame_vision_api, som bedömdes vara en lämplig startpunkt för det egna projektet. Detta exempelprojekt följde en snarlik struktur som kan ses i Figur 5.1. Koden kompilerade efter mindre ändringar i versioner som utvecklingsmiljön använder till en mobilapplikation som startade, men trots omfattande försök lyckades vi inte etablera en fungerande anslutning mellan applikationen och Frame.

På Discord-communityn för utveckling av Frame-applikationer rapporterade många andra användare om liknande problem att para till Frame, och saknade en lösning. Det spekulerades bland användare i att felet kunde ligga i buggar eller ändringar som uppstått i externa lågnivå-bibliotek, som simple_frame_app-4.0.1 använder sig av. Vi genomförde därför löpande versionsuppdateringar och ändringar i kod och utvecklingsmiljö för att applikationen skulle fungera.

I samband med detta arbete kom vi att förlita sig allt mindre på Brilliant Labs dokumentation, eftersom dokumentationen inte hade uppdaterats i takt med utvecklingen och var föråldrad och svår att följa. Merparten av arbetet i detta skedet gick istället till att steg för steg genomsöka koden i de olika biblioteken för att få programmet att fungera.

När kontakten mellan Frame och applikationen väl fungerade började vi experimentera med att ta bilder som skickas till ett externt API för svar. Eftersom vi inte hade tillgång till Intended Futures API behövde vi simulera AI-klientens del i vår pipeline från bildtagning till visning av information. För detta användes en enklare Python-server som substitut, som fanns som ett exempel från Brilliant Labs med mindre ändringar, byggd med Python-frameworket FastAPI. Bilden som tas skickas till Python-servern, som kontrollerar att det är en bild och skickar tillbaka ett textsvar från en lista med textsträngar. Tanken är att det i framtiden enkelt ska gå att byta destinationen för bildöverföringen, så att bilden istället skickas till Intended Futures API.

På grund av behovet att felsöka kontakten med Python-servern under arbetet utvecklades en separat mobilapplikation i Flutter, kallad “test_server_response”. Syftet var att eliminera så många övriga felfaktorer som möjligt under felsökningsarbetet. Applikationen gavs endast grundläggande funktionalitet: att spara en API-adress, låta användaren välja en bild från mobilens bildgalleri, skicka bilden till det angivna API:et, och visa serverns textsvar. Denna enklare applikation användes upprepade gånger under projektets gång för att verifiera att testmiljön fungerade vid arbete vid olika fysiska platser.

5.2.4 Ändringar och extra funktionalitet för förbättrad användarvänlighet

Härnäst gjordes många ändringar i applikationens UI. För att frigöra så mycket utrymme som möjligt på appens startskärm, där bild och text visas, togs flera widgets som var överflödiga för vårt projekt bort. Den frigjorda ytan gav mer utrymme att visa bilden som tas med Frame,

samt den sträng med mätvärden som applikationen får som svar från servern som visas både på mobiltelefonens UI och i Frames glasögondisplay.

Vi lade även märke till att det fanns skillnader i hur olika individer såg Frame-displayen beroende på invididuell ansiktsform. Delar av glasögondisplayen kunde ligga i synfältet för en användare men utanför synfältet för en annan användare. Därför utvecklades ett lättanvänt system för att med fyra knappar enkelt flytta grundpositionen av texten på displayen, så att olika testanvändare skulle få en rättvis bedömning av appens funktioner vid utvärdering.

Vid utveckling av denna funktion hittades en bugg i sendMessage()-metoden kring hur x- och y-koordinater skickas till Frame. Sättet som koordinaterna skickades saknade återkoppling i den Lua-kod som kör på Frame. sendMessage() var en av de metoder som hade sett många refaktoreringar under de många versionsuppdateringarna och i samband med detta hade funktionaliteten för x- och y-position samt färgval för texten försvunnit. När felet hittats löstes det relativt snabbt genom mindre ändringar i programmets interna Lua-bibliotek med kod som matas in i Frame vid uppstart.

Därefter skapades en Drawer-meny med ett eget widget-träd som fungerar som en meny för inställningar och nås via en hamburgerikon från övre vänster del av applikationens startskärm. Knapparna för att hantera textens position flyttades in i UI-trädet för hamburgermenyn. Det är också i denna menyn som knappar för att aktivera eller deaktivera diverse funktioner i programmet senare placerades.

5.2.5 Implementering av testfunktioner

Användaren kan ta ett kort med Frames kamera genom att tappa på Frame ett antal gånger, vilket registreras av med hjälp av Frames inbyggda rörelsesensor. Vi testade olika antal tappningar för korttagningsfunktionen. En tappning var för lite och ledde ofta till att Frame tog bilder av slumpmässiga huvudrörelser, vid tal, eller tuggningar av användaren. Tre tappningar beslutades vara för mycket, eftersom fototagning är centralt för användningen och eftersom användaren troligen vill ha funktionen så snabbt tillgänglig som möjligt. Därför beslutades att korttagning skulle ske vid dubbeltappning av Frame.

En av funktionerna vi ville utvärdera var användarens preferens för att ta ett foto: genom att tappa på Frame, eller via en knapp i mobilapplikationen. För detta ändamål skapades en

knapp längst ner på applikationens förstasida. Originalkoden för bildtagning som kallades vid tappning på Frame refaktorerades för att skapa en separat metod (takeFramePhoto()) som startar kedjan från bildtagning till visning av textsvaret från servern på Frame. Denna metod kallas nu både när användaren dubbeltappar på Frame, eller när användaren trycker på fotoknappen på mobilapplikationen.

I vanliga fall används Intended Futures AI-klient genom att användaren tar bilder med till exempel en mobilkamera, och användaren skickar sedan bilderna via en webbplats till ett API för AI-klienten. Vi såg en möjlighet att integrera denna form av användande direkt i vår mobilapplikation, genom att lägga till en extra funktion som gör det möjligt att helt enkelt byta till mobilens inbyggda kamera för fototagning, för att testa användares preferens. Detta gjordes genom import av bibliotek för hantering av mobilens kamerafunktioner och för att hantera tillåtelser i mobiltelefonen (vilket är nödvändigt för att användaren ska kunna acceptera användning av mobilkameran). Istället för att gå igenom proceduren där användaren laddar in varje bild som tas via en webbplats, så automatiserades processen i vår mobilapplikation till att automatiskt skicka den tagna bilden via samma pipeline som används för bilder tagna med Framekameran så fort bilden tas.

Bildtagning med mobiltelefonens kamera görs med metoden takeMobilePhoto() istället för takeFramePhoto(). Ytterligare två metoder - getFrameCameraUI() och getMobileCameraUI() - skapades även för att returnera olika widgetgrenar i Flutters widgetträd, för de delar av applikationens framsida som behöver skilja sig vid användning av de olika kamerorna.

Vi återanvände en funktion från exempelapplikationens inställningar som vi tyckte förbättrade användarvänligheten - en funktion som låter användaren rikta Frames kamerariktning uppåt eller nedåt. Flutters widgetstruktur gjorde det enkelt att göra plats för och flytta denna funktion till vår nya hamburgermeny.

I slutskedet av projektet fick vi besked att svaret från Intended Futures AI-klient skulle komma i form av en längre löpande text än vi hade planerat för. I samband med detta fick vi en förfrågan om det var möjligt att även lägga till och utvärdera en text-till-tal-funktion (text-to-speech, TTS) i applikationen för uppläsning av längre texter. Detta gjordes genom import av biblioteket flutter_tts.dart som använder sig av operativsystemets inbyggda TTS-funktion. Liksom de andra funktionerna kan användaren sätta på och stänga av TTS-funktionen med en knapp i applikationens inställningar. En knapp lades även till på

användargränssnittet för att enkelt kunna avbryta en pågående TTS-uppläsning vid behov. Denna knapp syns endast om TTS-funktionen är vald.

Frontsidan utgörs i den färdiga applikationen av en batteriwidget för Frames batteri i övre höger hörn, en hamburgerikon som leder till inställningarna, utrymme för bild och text (som kan scrollas vid behov), en knapp för fototagning, samt en så kallad “Footer”-widget vid botten av skärmen med knappar för att hantera anslutningen till Frame.

5.3 Systemets huvuddelar

5.3.3 Frame

AR-glasögonen “Frame” kommunicerar med mobilapplikationen genom att skicka bilder och tar emot textsträngar som visas upp i dess display. Anslutningen till mobilapplikationen sker via bluetooth. Figur 5.2 visar hur det kan se ut i glasögon-displayen för användaren, där den visar information om bilen som bilden är tagen av. Med hjälp av Frame får användaren den digitala informationen direkt. Beroende på vilken server som är anslutet till mobilapplikationen är det möjligt för informationen som visas upp vara mer detaljerad och specifik. Själva Frame har i uppgift att kunna kommunicera med mobilapplikationen genom att ta bilder och visa relevant information i samband med att möjliggöra ändringar på informationen beroende på inställningar från mobilapplikationen.

Figur 5.2: Bild tagen genom Frames högra lins som visar ett exempel av texten i glasögondisplayen.

5.3.4 Mobilapplikation

Mobilapplikationen kommunicerar både med AR-glasögonen och en server. Initiering av kommunikationen sker från mobilapplikationen där adressen till servern skrivs in och bluetooth-anslutningen till glasögonen görs. Genom att ta emot bilder från glasögonen (alternativt tas bild med mobilkamera), skickas bilden vidare till servern som svarar med en textsträng. Textsträngen hanteras av mobilapplikationen och visas upp både i användargränssnittet och AR-glasögonen. Eftersom mobilapplikationen är utrustad med olika funktioner och inställningar varierar vad som kommuniceras och visas upp mellan de olika enheterna och användargränssnittet.

5.3.5 Server-API

Kommunikationen mellan mobilapplikationen och servern görs med API. Servern är skriven i programmeringsspråket Python och tar emot bilder från mobilapplikationen och skickar tillbaka en textsträng. Arbetet med servern är gjort så att den ska kunna ersättas med en AI-klient. I nuläget, under detta arbete, skickar servern slumpmässigt en textsträng som är taget ur en lista av textsträngar.

5.4 Applikationens funktioner

5.4.1 Justering av textens position

Textvisning från Frame var varierande beroende på användaren. I vissa fall kunde användaren se texten som visades upp och i andra fall inte. Det skapades därför en offsetfunktion som gav användaren möjlighet att flytta var display-texten skulle visas. Det är tänkt att användaren ska ändra dessa inställningar vid början av användningen av mobilapplikationen.

5.4.2 Ändra kamerans vinkel

Applikationen har funktionalitet för att justera riktningen av Frames kamera uppåt eller nedåt från grundläget. Denna inställning sköts av en slider-widget i inställningsmenyn och var en inställning som adapterades från exempelapplikationen frame_vision_api.

5.4.3 Alternativ UI-knapp för bildtagning med Frame

Applikationen ger användaren möjlighet att ta foton med Frames kamera genom att antingen tappa på Frame, eller genom att använda en knapp på nedre delen av mobiltelefonens användargränssnitt.

5.4.4 Visning av text på Frames display

Användaren kan välja om funktionen för att visa texten som kommer som svar på en bild från API:t skall visas på Frames display. Denna funktion kan användas både till kortare texter, till exempel för visning av olika mätvärden, eller för längre löpande texter där användaren kan trippel-tappa Frame för att hoppa framåt i texten, eller fyrdubbel-tappa Frame för att hoppa bakåt i texten.

5.4.5 Text-to-speech

Appen har stöd för TTS-uppläsning av textsvaren från API:t, vilket erbjuder ett alternativt sätt att ta till sig informationen. Eftersom att Frame inte har några högtalare sker all ljuduppspelning via mobiltelefonens högtalare. Detta är en begränsning i den befintliga hårdvaran, och under tester uppmanades därför användare att bortse från detta och istället föreställa sig att uppläsningen sker direkt via glasögonen utan att mobiltelefonen behöver vara till hands.

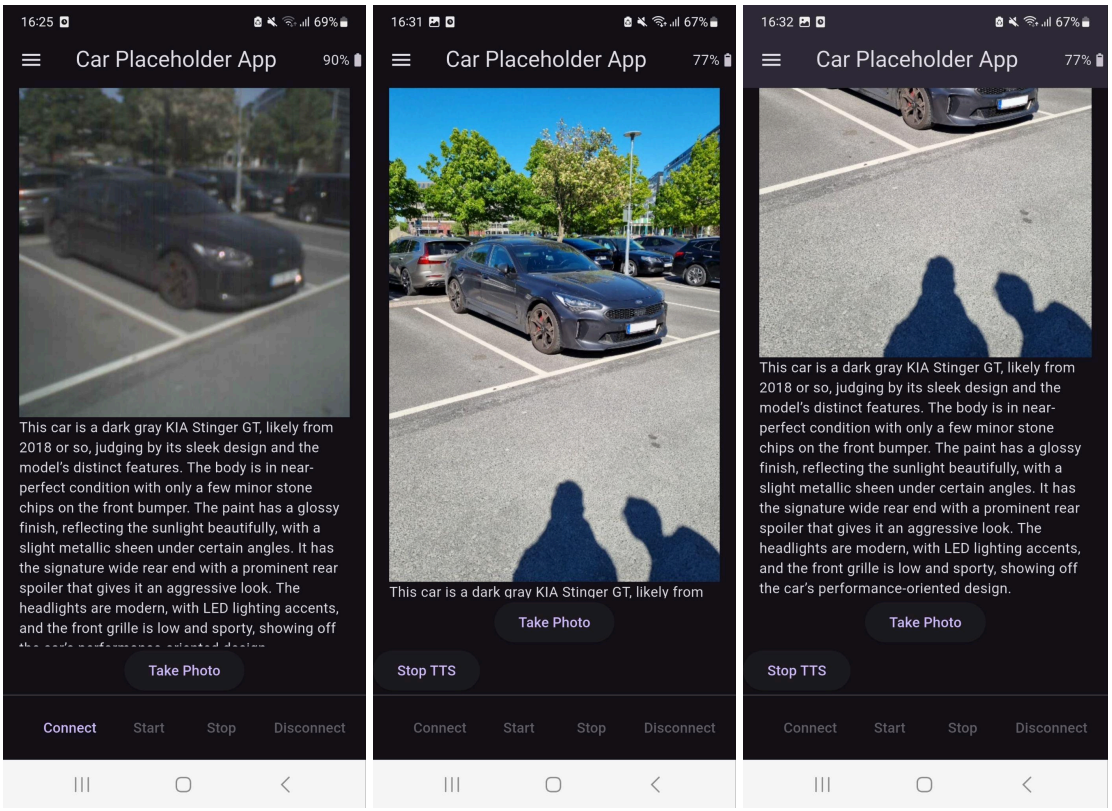
5.4.6 Val mellan Frames kamera eller mobiltelefonens kamera

Användaren kan välja mellan att använda kameran på glasögonen eller använda mobiltelefonens kamera. Detta gör det möjligt för användaren att byta mellan dessa lägen beroende på vad som passar bäst i olika situationer. Det kan vara svårt att ta en bild på många delar med bara AR-glasögonen, och tillgång till en mobilkamera gör det möjligt att ta bilder från mer vinklar. Med tanke på att Frames kamera har betydligt sämre kvalitet jämfört med en mobilkamera, och att projektgruppen under lång tid hoppades kunna koppla mobilapplikationen till Intended Futures API för faktisk AI-bildtolkning, ansågs möjligheten att ta bilder med mobilkameran vara en värdefull funktion att inkludera. Huvudsyftet med att inkludera möjligheten att ta bilder med mobiltelefonens kamera var dock att kunna utvärdera skillnaden i användning mellan att fotografera med en glasögonkamera och att använda en mobilkamera.

5.5 UI

5.5.1 Applikationens huvudsida

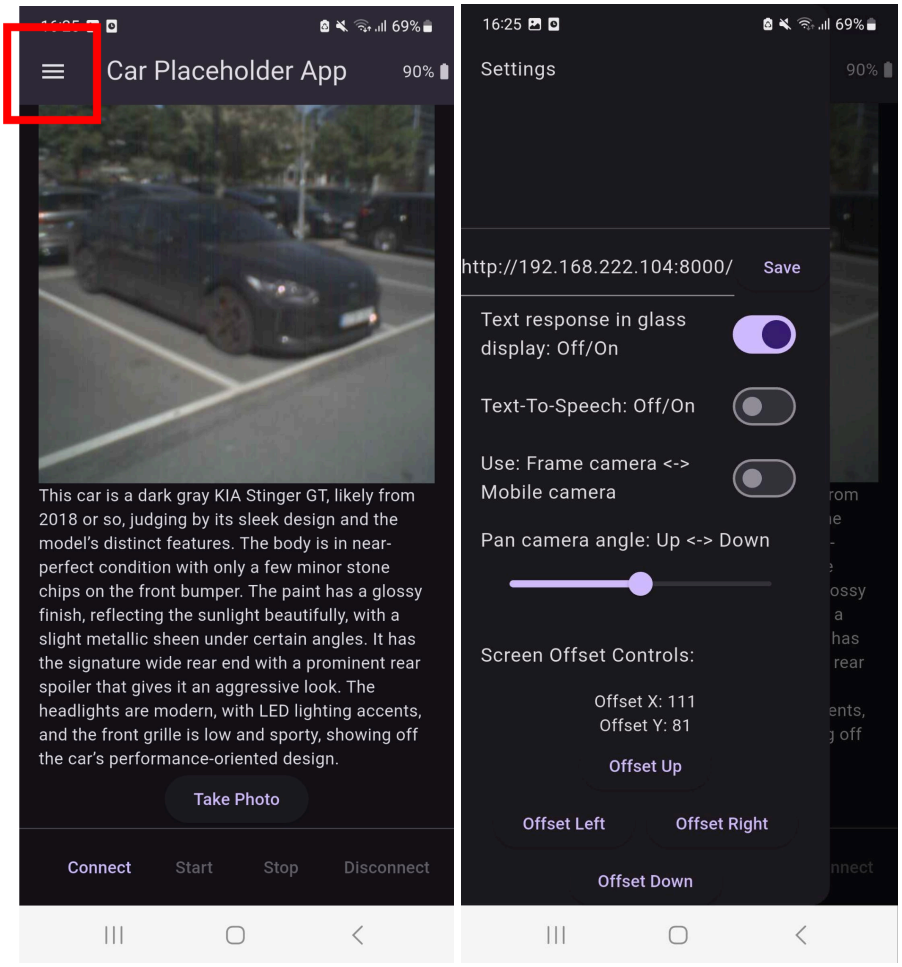
Applikationens huvudsida är uppdelad i olika delar. Längst upp visas ett hamburgerikon på vänster sida som öppnar sidan för applikationens inställningar. På höger sida visas AR-glasögonens batterinivå. När en bild tas, antingen med glasögonen eller med mobilkamera, visas bilden upp på övre delen av huvudsidan för att visa användaren vad de tagit bild på. Texten som visas upp i glasögondisplayen visas samtidigt också i användargränssnittet under bilden som visas. Det finns även en UI-knapp med texten “Take Photo” som ger användaren möjlighet att ta en bild. Bilden tas antingen genom mobilkameran eller med glasögon-kameran beroende på vad användaren har valt i inställningarna. Ifall "text-to-speech"-funktionen är igång har användaren möjlighet att genom UI-knappen “Stop TTS” att avsluta den pågående text-to-speech-uppläsningen. I botten av användargränssnittet finns utrymme för användaren att ansluta och koppla ur med Frame samt att starta mobilapplikationen efter att Frame är anslutet. Se Figur 5.3.



Figur 5.3. Från vänster till höger: Applikationens huvudsida vid bildtagning med Frames kamera, huvudsidan vid bildtagning med mobilkameran då TTS-funktionen är påslagen, och exempel på att huvudsidan har rullbar bild och text vid behov.

5.5.2 Placering av inställningar

Inställningarna för applikationen är alla inuti en meny. Som markerad med en röd ruta i Figur 5.4 finns användargränssnittet för inställningar tillgängligt att öppna på huvudsidan av applikationen. Längst upp i menyn för inställningar finns ett adressfält för användaren att skriva in vilket klient/server som den ska kopplas ihop med. Under adressfältet kan användaren sätta på och av 3 switchar där varje switch styr en funktion. Funktionerna är, uppifrån och ned, en switch som styr om texten ska visas i glasögonen, text-to-speech-funktionen och användning av mobilkamera eller glasögon-kamera. För att ändra på kamerariktningen finns en slider under de 3 switcharna som flyttar kamera riktningen upp eller ner. Den nedersta delen av hamburgermenyn innehåller offsetfunktionen. Där kan användaren ändra på var texten ska visas i glasögonen med hjälp av de fyra knapparna “Offset- up,down, right och left”. Offsetkoordinaterna visas också i inställningarna, se högra bilden av Figur 5.4.



Figur 5.4: Placering av hamburgerikon på vänster bild och hur användargränssnitt ser ut för inställningar på höger bild.

5.6 Testning av användare

Applikationen utvärderades som ett möjligt gränssnitt för att möjliggöra användarinteraktion med en AI-klient som tolkar bilder av bilar. Testningen av applikationen genomfördes av frivilliga studenter på Chalmers campus den 15:e, 16:e, och 19:e maj 2025.

Testet inleddes med en kort introduktion då testpersonerna fick tydliga instruktioner om Intended Futures AI-klient och dess användningsområde, en förklaring av sättet som användare använder AI-klienten i nuläget, och att vårt projekt var tänkt att utvärdera alternativa sätt som en användare skulle kunna interagera med denna AI-klient.

Det förklarades att vi inte hade tillgång till Intended Futures API och att testet därför utfördes med hjälp av en egen server som körde på en närstående laptop, och att textsvaret i sig därför inte var viktigt. Testpersonen skulle också bortse från möjliga begränsningar i glasögonens hårdvara, såsom skillnad i bildkvalité mellan bilder tagna med Frames kamera och bilder tagna med mobiltelefonens kamera, avsaknad av högtalare, eller eventuella problem att helt korrekt se texten i displayen.

Efter introduktionen fick testpersonen prova Frame och mobilapplikationens samtliga funktioner under en provperiod som vanligtvis varade i flera minuter, i syfte att ge användaren möjlighet att bekanta sig med systemet.

Efter användartestet fick testpersonen utvärdera sin upplevelse genom att besvara ett frågeformulär bestående av fyra frågor. Testpersonerna uppmanades att ställa frågor vid eventuella oklarheter av frågornas utformning. Utvärderingen fokuserade på användarens uppfattning kring följande aspekter:

- Vad användaren tyckte om att ta foton genom att tappa på Frame, jämfört med att använda fotoknappen i mobilapplikationens användargränssnitt.
- Om det var en förbättring att få svarstexten visad på Frames display, eller inte.
- Användarens intryck av att använda en TTS-funktion för att ta del av svarstexten.
- Vad användaren tyckte om att använda glasögonen för att ta foton, jämfört med att använda mobiltelefonens kamera.

Svaren på svarsenkäten följde en femgradig Likertskala, där varje fråga hade samma svarsalternativ:

- Mycket bättre

- Lite bättre
- Ingen skillnad
- Lite sämre
- Mycket sämre

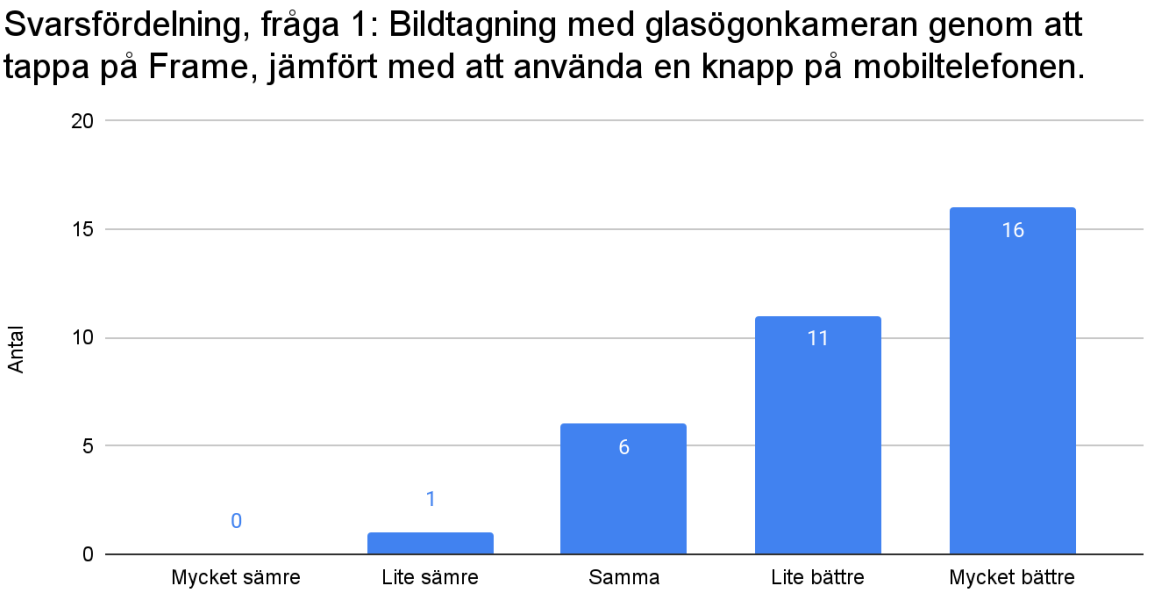
Enkäten som användes finns presenterad i Appendix A.

6. Resultat

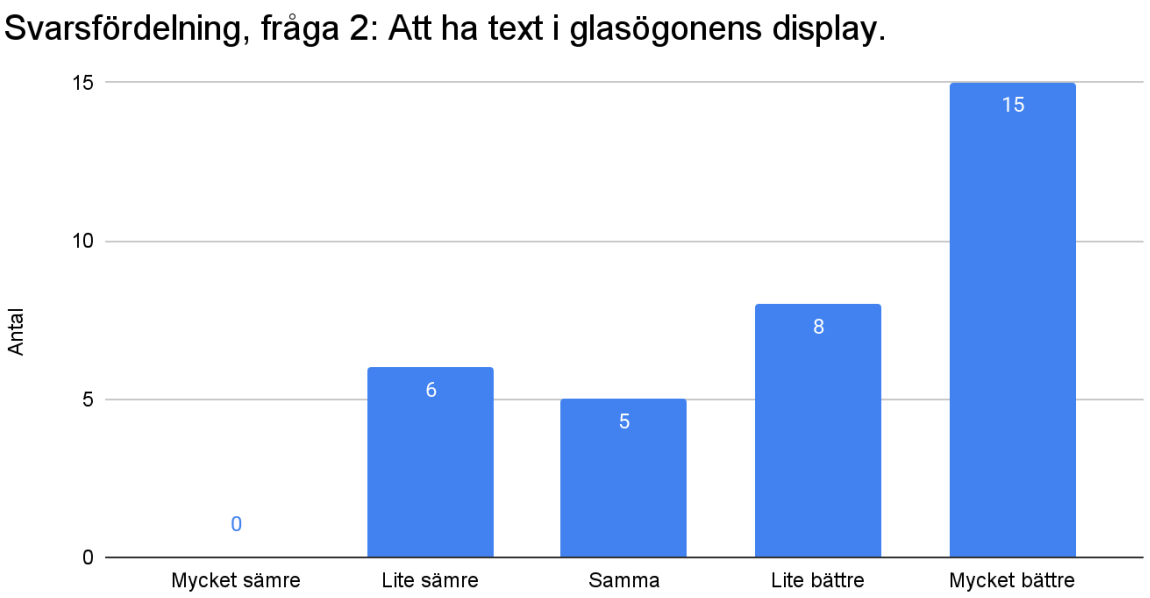
6.1 Resultat av användarutvärdering

Användarutvärderingen visade en tydlig preferens till att använda glasögonens kamera genom att tappa på glasögonen jämfört med att använda knappen på mobilens användargränssnitt (Figur 6.1). Även mottagandet av svarstexten direkt på glasögonens display föredrogs av de flesta framför andra sätt att ta till sig informationen (Figur 6.2).

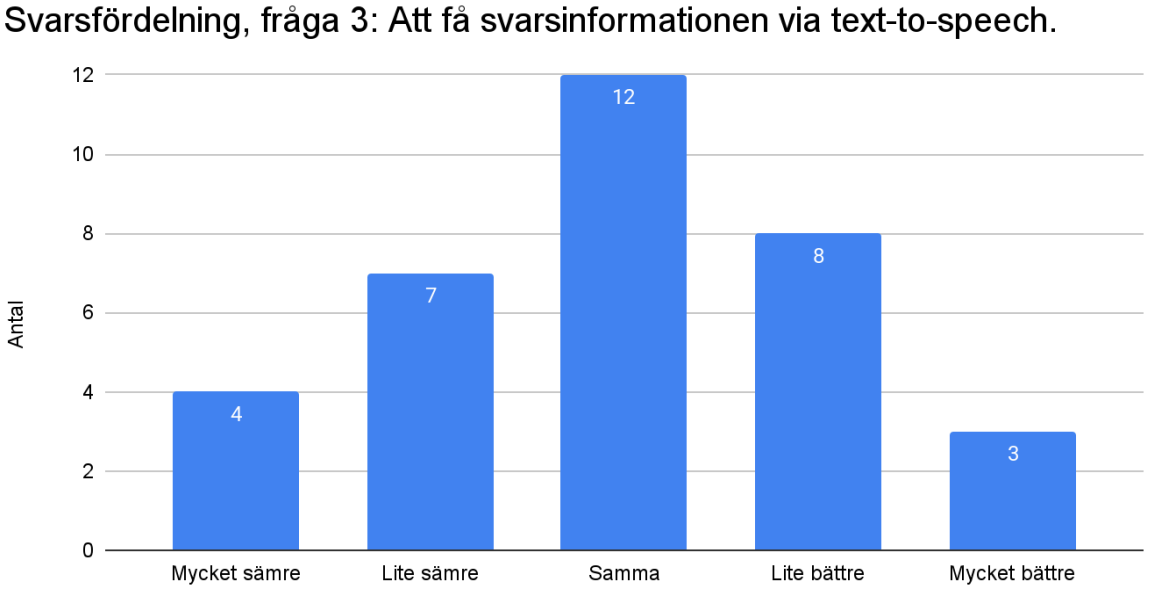
När det gäller preferenser för att få svarstexten uppläst via text-to-speech, respektive valet mellan att använda mobilens eller glasögonens kamera för bildtagning, var svaren mer jämnt fördelade (Figur 6.3 och Figur 6.4).



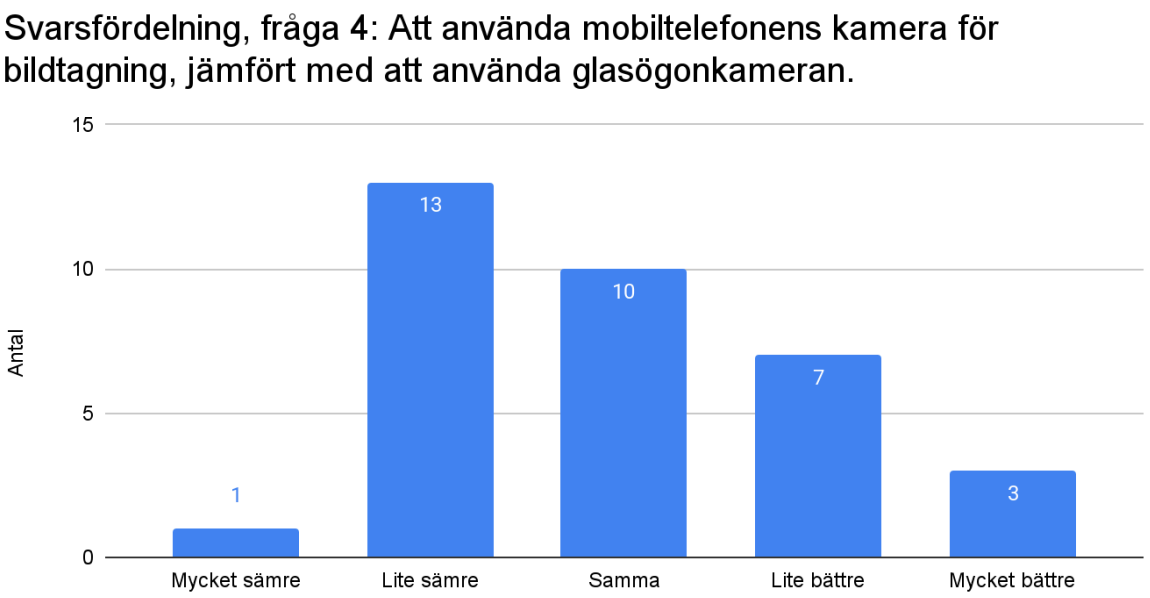
Figur 6.1: Svarsfördelning av enkätfråga 1 - Bildtagning med glasögonkameran genom att tappa på Frame, jämfört med att använda en knapp på mobiltelefonen.



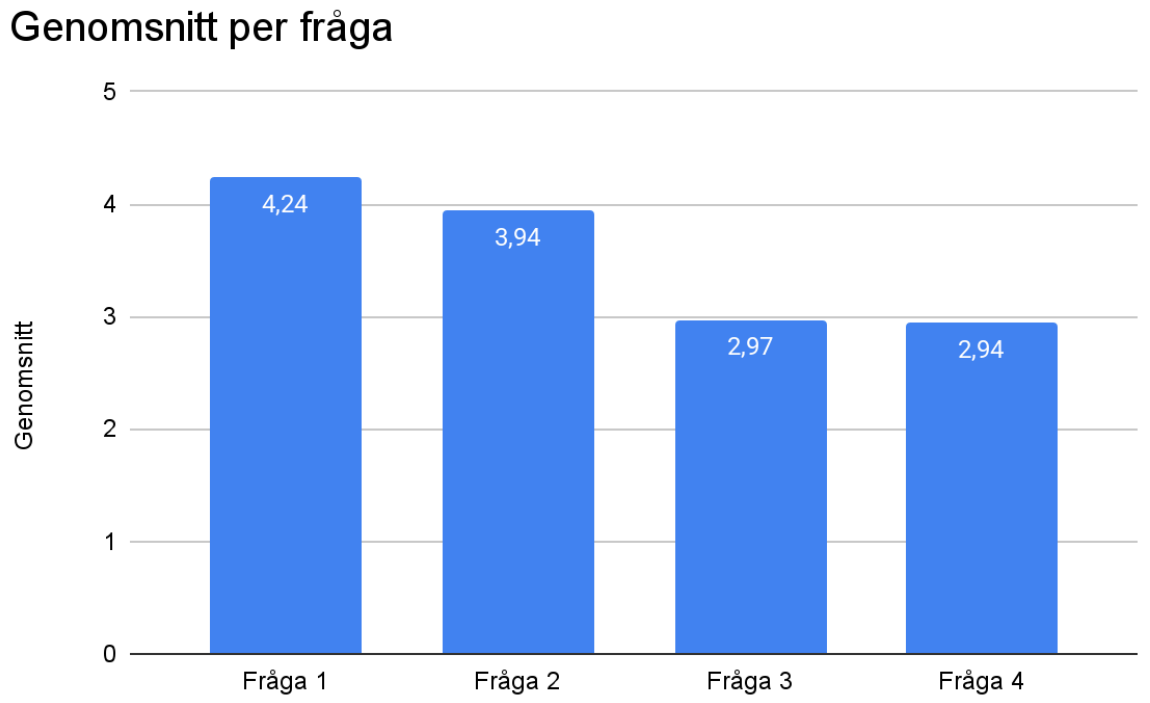
Figur 6.2: Svarsfördelning av enkätfråga 2 - Att ha text i glasögonens display.



Figur 6.3: Svarsfördelning av enkätfråga 3 - Att få svarsinformationen via text-to-speech.



Figur 6.4: Svarsfördelning av enkätfråga 4 - Att använda mobiltelefonens kamera för bildtagning istället för att använda glasögonkameran.



Figur 6.5: Visualisering av svaren på de fyra frågorna med hjälp av ett medelvärde. Här har de fem svaren getts ett numeriskt värde enligt följande: “Mycket sämre”: 1, “Lite sämre”: 2, “Samma”: 3, “Lite bättre”: 4, “Mycket bättre”: 5.

De individuella svaren per enkät finns representerat som en tabell i Appendix.

6.2 Potentiella felkällor

6.2.1 Avsaknad av expertis hos testpersoner

Testpersonerna i utvärderingen var studenter vid Chalmers och saknade specifik expertis inom fordondesign eller relaterade områden. Det kan därför ha varit svårt för dem att identifiera potentiella brister i mobilapplikationen ur ett professionellt användnarperspektiv, till exempel sett ur en bildesigners behov. Trots att det under testet förklarades vilken typ av information AI-klienten förväntas leverera, samt hur glasögonen kan användas för att ge en fördjupad upplevelse vid exempelvis bilutställningar, kan det ändå ha varit svårt för deltagarna att ge relevanta synpunkter.

6.2.2 Viljan att ge ett “snällt” resultat

Trots uppmaningar om att vara så ärliga som möjligt finns det en risk att testpersonerna gav mer positiv feedback än vad de egentligen ansåg. Eftersom både testledare och testpersoner var studenter kan det ha uppstått en form av social önskvärdhet, där de som gjorde testet ville vara vänliga eller undvika att ge kritik. En annan möjlig förklaring är att deltagarna trodde att en positiv respons skulle gynna projektet, och därför kan ha valt att uttrycka sig mer positivt än deras faktiska upplevde.

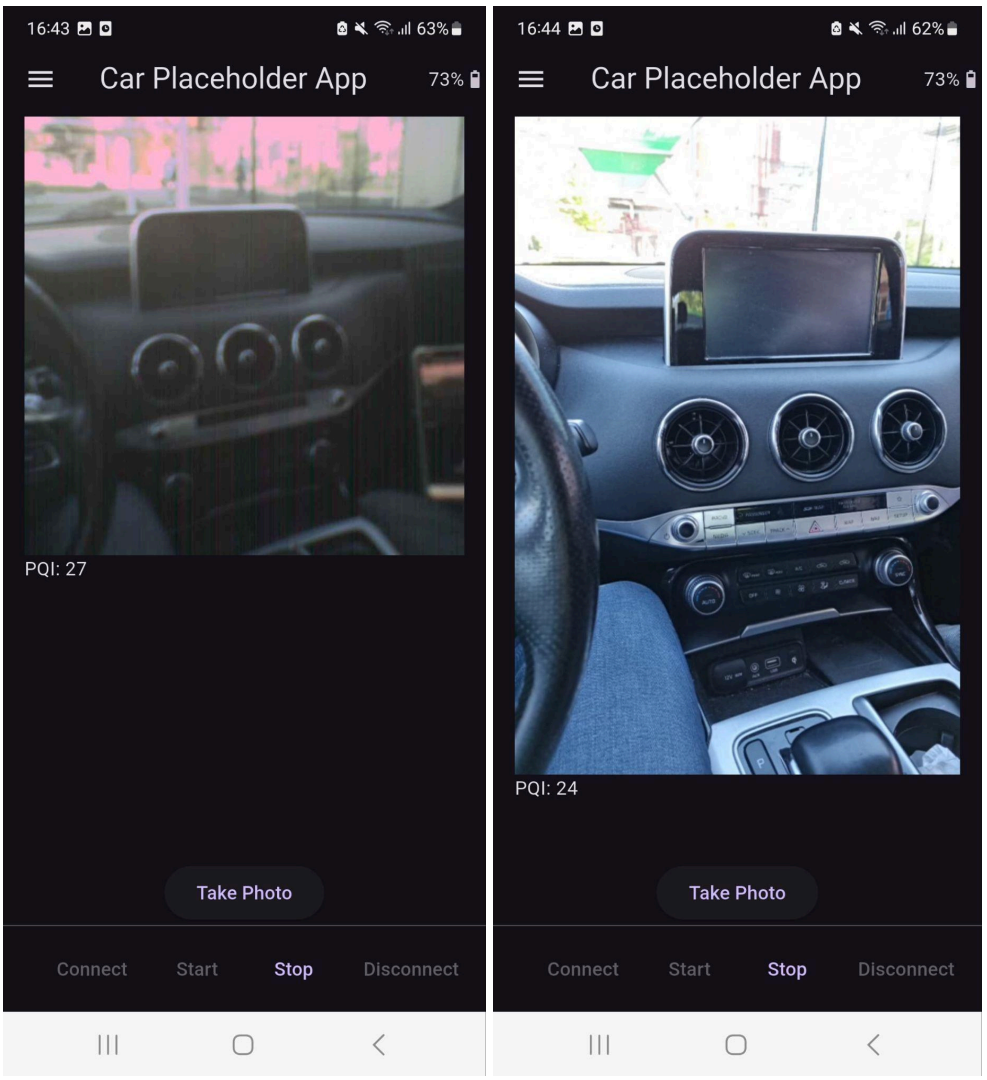
6.2.3 Svårighet att förbise brister hos hårdvara

Även om det vid upprepade tillfällen under testerna förklarades att testet inte var menat att utvärdera hårdvaran (Brilliant Labs Frame), utan sättet att använda glasögonen med en AI-klient, är det ändå möjligt att den sämre fotokvalitén, vissa användares svårighet att ta bilder genom att tappa “korrekt” på glasögonen, eller avsaknad av inbyggda högtalare, har färgat vissa användares uppfattning och därmed även svar.

7. Diskussion

7.1 Hårdvara

Eftersom projektet var tänkt att kopplas till en AI-klient som skulle överföra mer konkret data till applikationen och AR-glasögonen vet vi inte om Frame är rätt val för detta. Som i Figur 7.1 syns en väldigt tydlig skillnad på kvalitet mellan kameran från glasögonen och en mobilkamera. För att en AI-klient ska kunna göra en analys på vad bilden visar kanske det krävs högre bildkvalitet. Detta vet vi dock inte eftersom vi inte hade tillgång till en AI-klient under projektet. För att få en bättre uppfattning av bilkvalitet genom AR-displayen hade det uppskattats mer med möjligheten att visa mer än en textsträng på displayen. Eftersom detta inte var möjligt på grund av det begränsade utrymmet och funktionalitet var en textsträng bästa möjliga lösning.



Figur 7.1: Skillnad i kvalitet mellan Frames kamera och en mobilkamera.

7.2 Mjukvara

Eftersom Frame var ett av huvuddelarna i detta arbete var mobilapplikationen väldigt beroende av Brilliant Labs dokumentation. Med bristande SDK som kontinuerligt uppdateras var det svårt att avgöra hur kommunikationen och vilka funktioner skulle tillämpas. Det var bestämt att vi skulle arbeta med en mobilapplikation och vi var därför tvungna att förhålla oss till dokumentationer och SDK för Flutter. I projektets mellanfas uppdaterade Brilliant Labs sina SDK där vissa bibliotek uppdaterades och tagits bort eftersom de inte fungerade. Arbetet gick smidigt efter att ha fått en bättre förståelse för hur utveckling med Frame gick till samt vilka funktioner som kan implementeras.

En viktig begränsning i projektet var att vi inte hade tillgång till en faktisk AI-klient. Även om kommunikationen till servern var utformad för att efterlikna en AI-klient, hade tillgången av en fungerande AI-klient troligen haft en bättre påverkan på testresultaten. Om vi haft möjligheten till att se in i vilka analyser och svar en verklig AI-klient kan generera, hade vi kunnat göra en mer bättre bedömning av mobilapplikationens användbarhet.

7.3 Etisk diskussion

Med en väl fungerande klient som kan generera fördelaktig respons, skulle mobilapplikationen kunna effektivisera arbetet för de som designar bilar. AR kan självklart inte ersätta en fordonsdesigner, men det kan underlätta jobbet genom att ge en översikt över fördelar och nackdelar med en bildesign. Därmed kan designern enklare kunna fördjupa sig inom vad som kan förbättras samt vad resurser och energi ska läggas på. En eventuell nackdel med denna teknologi kan dock vara att efterfrågan av de som designar bilar kan minska ifall tekniken blir mycket framgångsrik. Detta kan leda till att efterfrågan bara går till experter och inte till de med motsvarande kunskapsnivå som enkelt kan ersättas av AI och AR. En viktig aspekt att ta hänsyn till är att den feedback som ges på designförslag kommer från en AI och inte från en människa. Om AI:n föreslår förändringar utifrån ett rent estetiskt perspektiv är det viktigt att förstå att det yttersta ansvaret fortfarande ligger på människan. Eftersom fordon är produkter med höga säkerhetskrav kan val på design som påverkar funktionalitet eller säkerhet ge allvarliga konsekvenser. Det är därför viktigt att granska och utvärdera AI-genererade förslag ur både ett säkerhets- och funktionsperspektiv.

7.4 Metodkritik

Detta projekt hade möjligtvis kunnat genomföras med ett annat tillvägagångssätt än det som valdes. Utvecklingen av mobilapplikationen baserades på ett exempelprojekt från Brilliant Labs GitHub, vilket fungerade som ett skelett. Projektet byggdes vidare på denna befintliga struktur och anpassades efter de funktioner som redan fanns integrerade.

Ett alternativt tillvägagångssätt hade varit att bygga applikationen helt från grunden. Det är dock svårt att i efterhand avgöra om det hade lett till ett bättre eller sämre resultat. Att börja från ett tomt projekt hade potentiellt kunnat ge oss mer flexibilitet och låtit oss bestämma mer av applikationens arkitektur. Samtidigt blir användningen av ett existerande projekt tillgång till en viss struktur och snabbare uppstart.

En faktor som försvårar bedömningen är att SDK:n och dokumentationen förändrades flera gånger under projektets gång. Bland annat togs SDK:n för Unity och webapplikationer bort. Svårighet att följa dokumentationen hade påverkat ett projekt byggt från grunden, och i slutändan gör vi bedömningen att ett sådant tillvägagångssätt inte hade lett till någon förbättring.

7.5 Reflektion över möjliga förbättringar

Det finns flera potentiella förbättringar som kan göras i den nuvarande versionen av mobilapplikationen. I början av projektet diskuterade vi olika alternativ för hur bildtagning med Frame-glasögonen skulle implementeras. Istället för att användaren manuellt tar en bild åt gången, funderade vi på att låta glasögonen kontinuerligt ta bilder med ett visst tidsintervall och uppdatera informationen som visas i displayen. Syftet med denna lösning var att förenkla användarupplevelsen vid undersökning av en bil, genom att utesluta behovet av att manuellt aktivera bildtagning. Denna funktion hade troligen varit relativt enkel att implementera med hjälp av en timer som kontinuerligt kallar metoden för bildtagning med Frame, vilket automatiskt är startpunkten för pipelinen för bildleverans och svarsvisning. Denna användningsfunktion valdes i slutändan bort, men hade möjligtvis kunnat vara något att utvärdera.

En mer enkel funktion att implementera skulle vara att ha med en kamera “preview” där användaren kan se vad de ska ta bild på vid användning av mobilkameran. En förhandsvisning visas innan bilden tas, vilket gör att användaren kan se var kameran är

riktad i realtid. När bilden väl är tagen visas resultatet kort på skärmen, varefter användargränssnittet automatiskt återgår till kamerans förhandsvisning.

Objektorienteringen i koden kan förbättras genom att skapa klasser för olika funktioner och hantering av kommunikation för att ge koden en bättre struktur. Vidare finns det vissa tekniska brister i hanteringen av kamerafunktionaliteten. Applikationen anropar troligen inte kamerans dispose()-funktion i alla fall där det bör göras, alltså att systemresurser inte frigörs korrekt och därmed används i onödan.

Fallet där användaren skulle neka till kamerans behörighet hanteras inte i nuläget. Enligt modern UX-praxis (UX - User Experience) och etablerade rekommendationer för hantering av behörigheter bör applikationen i detta fall visa ett meddelande som förklarar att kamerabehörighet krävs för att ta bilder med mobilkameran, samt instruera användaren om hur behörigheten kan aktiveras i mobiltelefonens inställningar. Sedan att förfrågan om kamerabehörighet återkommer när användaren ställer in för att använda mobilkameran i menyn för inställningar. Att förbättra denna hantering skulle höja användarvänligheten.

Källförteckning

[1] L. Elhattab, J. Khairalla, R. Al-Attar, S. Albert, N. Shorim and E. Eliwa, "Augmented Reality Applications in the Automotive Industry," 2023 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 2023, pp. 357-364.[Online]. Tillgänglig: https://ieeexplore.ieee.org/document/10278357?denied=. Hämtad 2025-03-15

[2] Kore, Understanding the different types of AR devices. UX Collective, 2018. [Online]. Tillgänglig: https://uxdesign.cc/augmented-reality-device-types-a7668b15bf7a. Hämtad 2025-05-02.

[3] J. M. Carroll, Human Computer Interaction - brief intro. Interaction Design Foundation - IxDF, 2014. [Online]. Tillgänglig: https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro. Hämtad 2025-05-02

[4] Billinghurst, Mark & Clark, Adrian & Lee, Gun, A Survey of Augmented Reality. Foundations and Trends® in Human-Computer Interaction, (2015). [Online]. Tillgänglig: https://www.researchgate.net/publication/277637661_A_Survey_of_Augmented_Reality. Hämtad 2025-05-02

[5] Y. M. Amin et al., Human-Computer Interaction in Augmented Reality Environments: Design Principles and User Experiences. 2024 36th Conference of Open Innovations Association (FRUCT), Lappeenranta, Finland, 2024. [Online]. Tillgänglig: https://ieeexplore.ieee.org/document/10749913. Hämtad 2025-05-02

[6] Boboc, Razvan & Florin, Girbacia & Butila, Eugen, The Application of Augmented Reality in the Automotive Industry: A Systematic Literature Review. Applied Sciences. (2020). [Online]. Tillgänglig: https://www.researchgate.net/publication/342357985_The_Application_of_Augmented_Re. Hämtad 2025-03-27

[7]I. Adamska, “8 uses of augmented reality in automotive manufacturing”. Nsflow. (2023). [Online]. Tillgänglig: https://nsflow.com/blog/8-uses-of-augmented-reality-in-automotive-manufacturing. Hämtad 2025-03-27

[8] E. Wallace, “How Augmented Reality is Shaping EV Development and Design”. RTinsight. (2024). [Online]. Tillgänglig: https://www.rtinsights.com/how-augmented-reality-is-shaping-ev-development-and-design/. Hämtad 2025-04-03
[9] D. Mourtzis, V. Zogopoulos, E. Vlachou, Augmented Reality supported Product Design towards Industry 4.0: a Teaching Factory paradigm, Procedia Manufacturing, Volume 23, 2018, Pages 207-212.[Online].
Tillgänglig: https://www.sciencedirect.com/science/article/pii/S2351978918304906. Hämtad 2025-04-03

[10]UX Bulletin, Future of Brainstorming: How AR is Revolutionizing Design Collaboration. (2022). [Online]. Tillgänglig: https://www.ux-bulletin.com/ar-design-collaboration/#toc_What_is_AR_Design_Collaboration . Hämtad 2025-04-03

[11] Intended Future. Officiell webbplats. [Online]. Tillgänglig: https://www.intendedfuture.ai/. Hämtad 2025-04-27

[12] M. D. Mura, G. Dini, An augmented reality approach for supporting panel alignment in car body assembly. Journal of Manufacturing Systems, Volume 59, 2021, Pages 251-260. [Online]. Tillgänglig:https://www.sciencedirect.com/science/article/pii/S0278612521000595. Hämtad 2025-04-17

[13]Atici-Ulusu, H., Ikiz, Y. D., Taskapilioglu, O., & Gunduz, T. (2021). Effects of augmented reality glasses on the cognitive load of assembly operators in the automotive industry. International Journal of Computer Integrated Manufacturing, 34(5), 487–499. [Online]. Tillgänglig:https://www.tandfonline.com/doi/epdf/10.1080/0951192X.2021.1901314?. Hämtad 2025-04-17

[14]Gavish, N., Gutiérrez, T., Webel, S., Rodríguez, J., Peveri, M., Bockholt, U., & Tecchia, F. (2013). Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks. Interactive Learning Environments, 23(6), 778–798. [Online]. Tillgänglig:https://www.tandfonline.com/doi/full/10.1080/10494820.2013.815221#d1e242. Hämtad 2025-04-17

[15] F.R. Pusda, F.F. Valencia, V.H. Andaluz, V.D. Zambrano, Training Assistant for Automotive Engineering Through Augmented Reality. Springer, Cham. (2019). [Online]. Tillgänglig:https://link.springer.com/chapter/10.1007/978-3-030-25999-0_13. Hämtad 2025-04-17

[16] Future visual, How Augmented Reality is Driving Change in Car Dealerships. [Online]. Tillgänglig: https://www.futurevisual.com/blog/augmented-reality-car-dealerships/. Hämtad 2025-04-29

[17] Philipp A. Rauschnabel, Barry J. Babin, M. Claudia tom Dieck, Nina Krey, Timothy Jung, What is augmented reality marketing? Its definition, complexity, and future. Journal of Business Research, Volume 142, 2022, Pages 1140-1150. Tillgäng: https://www.sciencedirect.com/science/article/pii/S0148296321010043. Hämtad 2025-04-29

[18] Brandwidth, “Toyota AR Demo video”. Youtube. (2019). [Online]. Tillgänglig: https://www.youtube.com/watch?v=xBnyWWECHac. Hämtad 2025-05-5

[19] J. L. Gabbard, G. M. Fitch and H. Kim, "Behind the Glass: Driver Challenges and Opportunities for AR Automotive Applications," in Proceedings of the IEEE, vol. 102, no. 2, pp. 124-136, Feb. 2014. [Online]. Tillgänglig: https://ieeexplore.ieee.org/abstract/document/6704805. Hämtad 2025-04-29

[20] Z. Wang, K. Han and P. Tiwari, "Augmented Reality-Based Advanced Driver-Assistance System for Connected Vehicles," 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 2020, pp. 752-759. [Online]. Tillgänglig: https://ieeexplore.ieee.org/abstract/document/9283462. Hämtad 2025-04-29

[21]: Brilliant Labs, “Dev”. [Online]. Tillgänglig: https://brilliant.xyz/pages/developers. Hämtad 2025-03-02

[22] Brilliant Labs, “Building Apps with the frame SDK”. [Online]. Tillgänglig: https://docs.brilliant.xyz/frame/frame-sdk/. Hämtad 2025-03-02

[23] Brilliant Labs. Officiell webbplats. [Online]. Tillgänglig: https://brilliant.xyz/. Hämtad 2025-03-02

[24] simple_frame_app. Dokumentation. pub.dev. [Online]. Tillgänglig: https://pub.dev/documentation/simple_frame_app/latest/. Hämtad 2025-03-27

[25] CitizenOneX. simple_frame_app [GitHub-repository]. [Online]. Tillgänglig: https://github.com/CitizenOneX/simple_frame_app. Hämtad 2025-03-27

[26] CitizenOneX. frame_vision_api [GitHub-repository]. [Online]. Tillgänglig: https://github.com/CitizenOneX/frame_vision_api. Hämtad 2025-03-27

[27] CitizenOneX. Github-användare. [Online]. Tillgänglig: https://github.com/CitizenOneX. Hämtad 2025-03-27

[28] Google, “Build for any screen”. Flutter.dev. [Online]. Tillgänglig: https://flutter.dev/ Hämtad 2025-04-02

[29] Google, “Welcome to the Flutter API reference documentation!”. Flutter SDK. [Online]. Tillgänglig: https://api.flutter.dev/ Hämtad 2025-04-02

[30] Dart.dev. Officiell webbplats. [Online]. Tillgänglig: https://dart.dev/ Hämtad 2025-04-15

[31] Flutter-artikel på ars technica [Online]. Tillgänglig: https://arstechnica.com/gadgets/2018/02/google-starts-a-push-for-cross-platform-app-development-with-flutter-sdk/. Hämtad 2025-05-02

[32] Dart.dev, “Classes”. (2025). [Online]. Tillgänglig: https://dart.dev/language/classes. Hämtad 2025-04-15

[33] Lua, “About”. (2024). [Online]. Tillgänglig: https://www.lua.org/about.html. Hämtad 2025-05-05

[34] C. Stieg, “Why Lua Is So Popular — & What You Can Build With It”. Lua Code Academy. (2022). [Online]. Tillgänglig: https://www.codecademy.com/resources/blog/what-is-lua-programming-language-used-for/. Hämtad 2025-05-05

[35] Atlassian, “Getting git right”. [Online]. Tillgänglig: https://www.atlassian.com/git. Hämtad 2025-05-10

[36] Github Docs, “About repositories”. Github. [Online]. Tillgänglig: https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories. Hämtad 2025-05-10

[37] M. Nemerever, “Git, GitHub, & Workflow Fundamentals”. Dev.to. (2019). [Online]. Tillgänglig: https://dev.to/mollynem/git-github--workflow-fundamentals-5496. Hämtad 2025-05-10

[38] Google Developers, “Build and run your app”. Android Studio. (2025). [Online]. Tillgänglig: https://developer.android.com/studio/run. Hämtad 2025-05-05

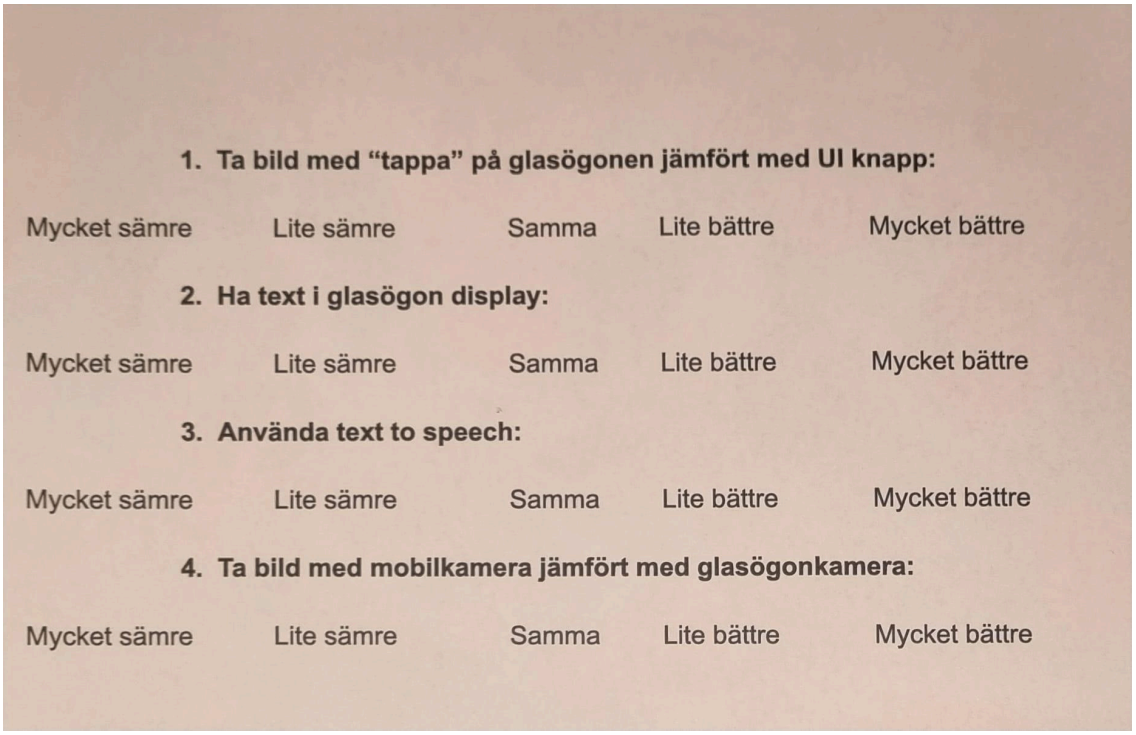
[39] Google, “Android Studio and IntelliJ”. Flutter docs. (2025). [Online]. Tillgänglig: https://docs.flutter.dev/tools/android-studio. Hämtad 2025-05-05

[40] A. Kaparthy. Youtube-kanal. [Online]. Tillgänglig: https://www.youtube.com/@AndrejKarpathy/featured. Hämtad 2025-05-5

[41] T. Lastovich, “Brilliant Labs Frame Review”. Lastovich.com, June 10, 2024. [Online]. Tillgänglig: https://lastovich.com/writing/are-smart-glasses-finally-here-brilliant-labs-frame. Hämtad 2025-06-11

Appendix A

Svarsenkät som användes vid utvärderingen:



1. **Ta bild med "tappa" på glasögonen jämfört med UI knapp:**

Mycket sämre Lite sämre Samma Lite bättre Mycket bättre

2. **Ha text i glasögon display:**

Mycket sämre Lite sämre Samma Lite bättre Mycket bättre

3. **Använda text to speech:**

Mycket sämre Lite sämre Samma Lite bättre Mycket bättre

4. **Ta bild med mobilkamera jämfört med glasögonkamera:**

Mycket sämre Lite sämre Samma Lite bättre Mycket bättre

Tabell med enkätsvar per fråga

U	Q1	Q2	Q3	Q4
1	5	4	3	2
2	3	3	2	3
3	5	5	4	2
4	3	2	2	2
5	5	5	2	3
6	5	4	4	2
7	5	5	3	4

8	5	5	2	2
9	5	5	4	4
10	4	2	1	5
11	4	2	4	3
12	4	3	3	3
13	3	3	2	2
14	5	5	3	2
15	3	2	3	2
16	5	5	4	4
17	5	5	3	2
18	5	5	3	2
19	4	4	3	3
20	4	5	5	3
21	4	2	3	4
22	5	5	1	2
23	3	3	2	3
24	4	4	3	3
25	4	4	5	4
26	5	4	1	3
27	4	2	3	2
28	4	5	1	5
29	2	3	3	5
30	5	5	4	4
31	5	4	2	4

32	5	5	4	1
33	3	4	4	3
34	4	5	5	2

Appendix B

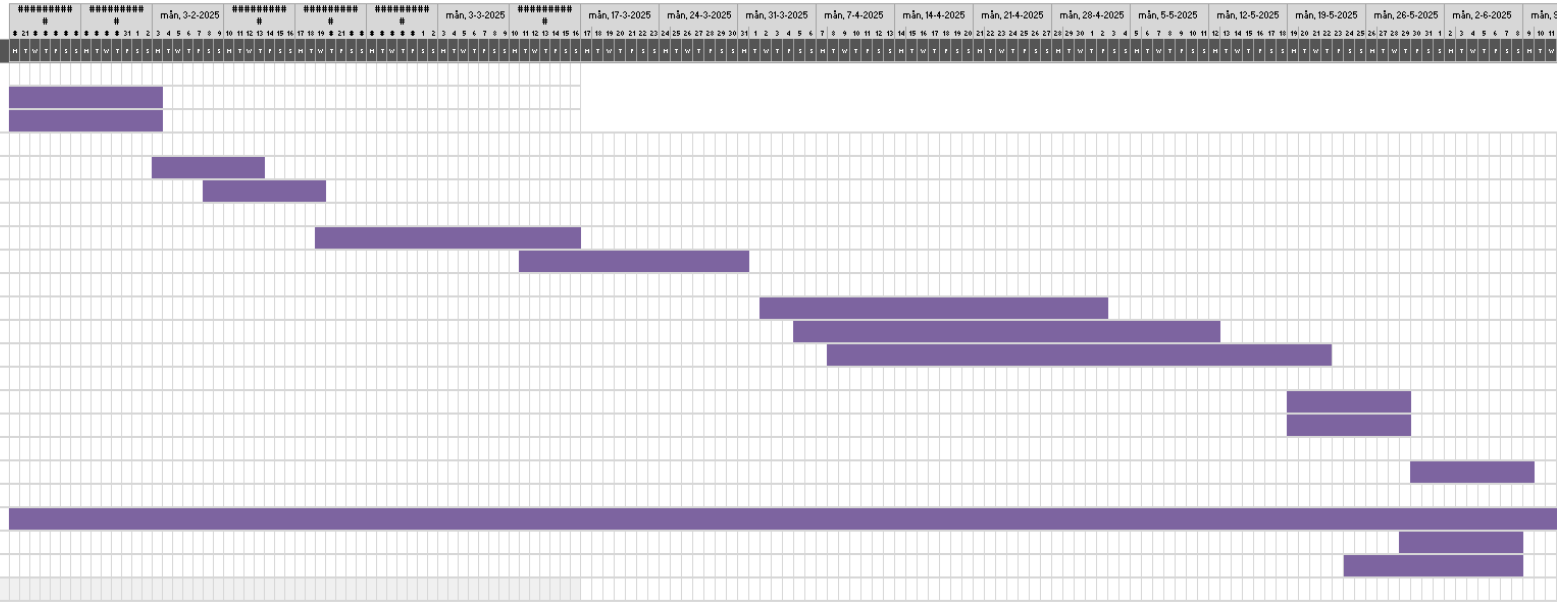
GANTT-schema:

TASK	START	END
Inledande del		
Planeringsrapport	mån, 20-1	mån, 3-2
Information och kontrakt från Intended Future	mån, 20-1	mån, 3-2
Litteraturöversikt		
Undersök befintliga AR-applikationer inom bilindustrin.	mån, 3-2	tor, 13-2
Teorier om uppfattning rörande produktkvalitet o AR-teknologi.	lör, 8-2	ons, 19-2
Teknisk utveckling		
Använddet öppna SDK:t från AI-glasögonens GitHub-repositor	ons, 19-2	sön, 16-3
Utveckla AR-applikationen som känner PQ attributer och visar	tis, 11-3	mån, 31-3
Applikationsfunktioner		
Identifiering av en bils tillbehör	ons, 2-4	fre, 2-5
Informationsöverlagring	lör, 5-4	mån, 12-5
Interaktiva element	tis, 8-4	tor, 22-5
Testning av användare		
Rekrytera deltagare för testning i en simulerad showroom-miljö.	mån, 19-5	tor, 29-5
Använd enkäter och intervjuer för att samla data.	mån, 19-5	tor, 29-5
Dataanalys		
Bedöm kvalitativ feedback	fre, 30-5	mån, 9-6
Uppgifter från skolan		
Skriftlig projektrapport	mån, 20-1	lör, 14-6
Egen presentation	tor, 29-5	sön, 8-6
Opponering	lör, 24-5	sön, 8-6

Utveckling av Augmented reality-applikation

Chalmers Mekatronik Examensarbete
Serok Tunc

Project Start: mån, 20-1
Display Week: ######



TASK	START	END
Inledande del		
Planeringsrapport	######	mån, 3-2
Information och kontrakt från Intended Future	######	mån, 3-2
Litteraturöversikt		
Undersök befintliga AR-applikationer inom bilindustrin.	mån, 3-2	tor, 13-2
Teorier om uppfattning rörande produktkvalitet o AR-tekn	lör, 8-2	ons, 19-2
Teknisk utveckling		
Använddet öppna SDK:t från AI-glasögonens GitHub-repc	ons, 19-2	sön, 16-3
Utveckla AR-applikationen som känner PQ attributer ocl	tis, 11-3	######
Applikationsfunktioner		
Identifiering av en bils tillbehör	ons, 2-4	fre, 2-5
Informationsöverlagring	lör, 5-4	######
Interaktiva element	tis, 8-4	tor, 22-5
Testning av användare		
Rekrytera deltagare för testning i en simulerad showroom-	######	tor, 29-5
Använd enkäter och intervjuer för att samla data.	######	tor, 29-5
Dataanalys		
Bedöm kvalitativ feedback	fre, 30-5	mån, 9-6
Uppgifter från skolan		
Skriftlig projektrapport	######	lör, 14-6
Egen presentation	tor, 29-5	sön, 8-6
Opponering	lör, 24-5	sön, 8-6

INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2025
www.chalmers.se

CHALMERS