



KUNNEN OOGBEWEGINGEN EN KNIPPEREN VIA EEN CHROME-EXTENSIE VERTAALD WORDEN NAAR WEBSITE-INTERACTIES?

INTERNE PROMOTOR: PIETER-JAN BEECKMAN

EXTERNE PROMOTOR: MATHIEU BRAEKEVELD

ONDERZOEKSVRAAG UITGEVOERD DOOR

TIBO MESSIAEN

VOOR HET BEHALEN VAN DE GRAAD VAN BACHELOR IN DE

MULTIMEDIA EN CREATIVE TECHNOLOGIE

Howest | 2023-2024

Woord vooraf

Deze bachelorproef markeert het hoogtepunt van mijn studie Multimedia en Creative Technologie aan de Hogeschool Howest in Kortrijk.

Het onderwerp dat ik heb gekozen voor deze bachelorproef is voortgekomen uit een concrete onderzoeksvraag die me bijzonder intrigeerde: "Kunnen oogbewegingen en knippen via een Chrome-extensie vertaald worden naar website-interacties?" Deze vraag weerspiegelt mijn passie voor eye-tracking technologieën en mijn achtergrond in web development.

Deze bachelorproef bespreekt eerst de research en technische demo uit de module 'Researchproject'. De demo bestond uit het gebruiken van mijn zelf geschreven Chrome-extensie voor het aansturen van éénder welke website.

Gedurende dit proces ben ik enorm dankbaar geweest voor de begeleiding en expertise van mijn promotor, Pieter-Jan Beeckman. Zijn waardevolle inzichten en feedback hebben mijn onderzoek verrijkt en mijn begrip verdiept. Ook wil ik zeker Arthur d'Hooge en Mathieu Braekeveld bedanken om feedback te geven op mijn behaalde resultaten.

Daarnaast wil ik mijn dankbaarheid uiten aan alle medestudenten en docenten die mij hebben geholpen tijdens het realiseren van mijn research project.

Met trots presenteer ik deze bachelorproef als een belangrijke mijlpaal in mijn academische en professionele ontwikkeling.

Tibo Messiaen
12/02/2024

Abstract

Deze studie is gemaakt rond de onderzoeksvraag: "Kunnen oogbewegingen en knippen via een Chrome-extensie vertaald worden naar website-interacties?"

Het onderzoek omvatte de ontwikkeling en het testen van een Chrome-extensie die in staat is om oogbewegingen en knippen te interpreteren als invoercommando's voor navigatie en interactie op websites. Belangrijke factoren die van invloed zijn op de evaluatie van het onderzoek zijn de nauwkeurigheid en betrouwbaarheid van het oogvolgsysteem, de gebruiksvriendelijkheid van de interface en de mogelijke privacy zorgen die gepaard gaan met het verzamelen en verwerken van gevoelige biometrische gegevens.

Belangrijke overwegingen voor het advies omvatten het waarborgen van robuuste privacy maatregelen, het optimaliseren van de nauwkeurigheid en responsiviteit van het oogvolgsysteem, en het bieden van aanpasbare instellingen om diverse gebruikersbehoeften en -voorkeuren tegemoet te komen.

Tot slot, hoewel de technologie veelbelovend is voor het verbeteren van de web toegankelijkheid, zijn verdere verfijning en grondige tests noodzakelijk om technische uitdagingen en ethische overwegingen aan te pakken voordat breed gebruik mogelijk is.

Inhoudsopgave

| | |
|---|----|
| Woord vooraf | 1 |
| Abstract..... | 2 |
| Inhoudsopgave | 3 |
| Figurenlijst..... | 5 |
| Lijst met afkortingen | 6 |
| Verklarende woordenlijst | 7 |
| 1 Inleiding..... | 9 |
| 2 Research..... | 10 |
| 2.1 Hoe maak je een Chrome-extensie? | 10 |
| 2.2 Welke website interacties zijn er mogelijk via een Chrome-extensie? | 12 |
| 2.3 Hoe gebruik je eye tracking in de Chrome-extensie?..... | 14 |
| 2.4 Is externe software nodig bij het gebruik van eye tracking binnen de Chrome-extensie? | 14 |
| 2.5 Wat zijn de voor- en nadelen van het gebruiken van een standaard camera t.o.v eye tracking glasses? | 15 |
| 2.6 Hoe kan je oog bewegingen registreren? | 16 |
| 2.7 Hoe kan je acties registreren om bepaalde interacties uit te voeren? | 18 |
| 3 Technisch onderzoek | 19 |
| 3.1 Website pagina maken | 20 |
| 3.2 Website omzetten naar Chrome-extensie..... | 20 |
| 3.3 Eye tracking uit testen | 21 |
| 3.3.1 Webgazer..... | 22 |
| 3.3.2 GazeCloudAPI | 23 |
| 3.3.3 Eye tracking keuze | 24 |
| 3.4 Face landmark Detection uittesten | 25 |
| 3.4.1 PyGaze..... | 25 |
| 3.4.2 MediaPipe Face landmark Detection..... | 26 |
| 3.4.3 Face landmark Detection keuze | 26 |
| 3.5 Eye tracking & Face landmark Detection toevoegen aan Chrome-extensie..... | 26 |
| 3.6 Interacties opstellen..... | 27 |
| 3.6.1 Scrollen nabootsen | 27 |
| 3.6.2 Muis klik nabootsen..... | 28 |
| 3.6.3 Dubbel klik nabootsen | 28 |
| 3.6.4 Terug knop nabootsen | 29 |
| 3.6.5 Markeren van element die bekeken worden..... | 30 |
| 3.7 Gebruiker acties registreren | 31 |
| 3.8 Chrome-extensie instellingen..... | 32 |
| 4 Reflectie..... | 35 |
| 4.1 Resultaten technisch onderzoek..... | 35 |
| 4.1.1 Sterke punten | 35 |
| 4.1.2 Zwakke punten | 36 |
| 4.2 Bruikbaarheid projectresultaat | 36 |
| 4.3 Overweging implementatie bedrijfscontext..... | 37 |
| 4.4 Alternatieven | 38 |
| 4.5 Economische en maatschappelijke meerwaarde..... | 38 |
| 4.5.1 Economische meerwaarde..... | 38 |
| 4.5.2 Maatschappelijke meerwaarde..... | 39 |
| 4.5.3 Conclusie..... | 39 |

| | | |
|-------|---|----|
| 4.6 | Suggesties vervolgonderzoek..... | 40 |
| 4.6.1 | Handsfree typen..... | 40 |
| 4.6.2 | Firefox extensie ontwikkelen..... | 40 |
| 4.6.3 | Verbetering van Eye-tracking met Interpolatie..... | 40 |
| 4.6.4 | Conclusie..... | 40 |
| 5 | Advies | 41 |
| 5.1 | Bruikbaarheid en toepasbaarheid | 41 |
| | Toepassingen in de Privésector | 41 |
| | Toepassingen in de Bedrijfswereld | 42 |
| | Belangrijke overwegingen | 42 |
| 5.2 | Concrete aanbevelingen | 42 |
| 5.2.1 | Toepassingsbereik | 42 |
| 5.2.2 | Toegankelijkheid | 42 |
| 5.2.3 | Gebruikersgericht Ontwerp | 43 |
| 5.3 | Stappenplan | 44 |
| 5.3.1 | Stap 1: Onderzoeken | 44 |
| 5.3.2 | Stap 2: Uittesten..... | 44 |
| 5.3.3 | Stap 3: Ontwikkelen..... | 44 |
| 5.3.4 | Stap 4: Integreren | 44 |
| 5.3.5 | Stap 5: Afwerken | 45 |
| 6 | Conclusie | 46 |
| 6.1 | Gebruikte technologieën | 46 |
| 6.2 | Eindresultaat..... | 47 |
| 6.3 | Onderzoeksvraag | 48 |
| 6.4 | Suggesties vervolg onderzoek..... | 48 |
| 7 | Literatuurlijst..... | 49 |
| 8 | Bijlages | 51 |
| 8.1 | Verslag Faktion | 51 |
| 8.2 | Verslag Goomyx..... | 53 |
| 8.3 | Installatie & gebruikershandleiding | 56 |
| | Chrome extension | 56 |
| | Eye-tracking Chrome extension installation manual | 57 |
| | Eye-tracking Chrome extension user manual..... | 60 |

Figurenlijst

| | |
|--|----|
| Figuur 1 - schema overzicht deelvragen | 10 |
| Figuur 2 – service worker code | 11 |
| Figuur 3 - detectie om scrollen te activeren | 12 |
| Figuur 4 - klik naboots functie..... | 12 |
| Figuur 5 - dubbel klik naboots functie | 13 |
| Figuur 6 - terug knop functie..... | 13 |
| Figuur 7 - Schema overzicht over hoe oog bewegingen worden geregistreerd | 16 |
| Figuur 8 - mediapipline output data | 17 |
| Figuur 9 - chrome-extensie pop-up | 18 |
| Figuur 10 - eye tracking extension schermafbeelding | 19 |
| Figuur 11 - manifest.json code overview..... | 21 |
| Figuur 12 - eye tracking test website - Webgazer - schermafbeelding..... | 22 |
| Figuur 13 - eye tracking test website - GazeCloudAPI - schermafbeelding | 23 |
| Figuur 14 – voordelen en nadelen van WebGazer en GazeCloudAPI | 24 |
| Figuur 15 - pygaze blink detection | 25 |
| Figuur 16 - scroll actie code | 27 |
| Figuur 17 - klik naboots functie code..... | 28 |
| Figuur 18 - dubbel klik naboots functie code | 29 |
| Figuur 19 - terug actie code regel | 29 |
| Figuur 20 - element markeren functie code | 30 |
| Figuur 21 - onderscheiden van knipper acties code | 31 |
| Figuur 22 - instellingen om acties aan interacties te koppelen..... | 32 |
| Figuur 23 - ophalen van opgeslagen instellingen code | 33 |
| Figuur 24 - opslaan van instellingen code | 33 |
| Figuur 25 - acties de gekoppelde interactie laten uitvoeren..... | 34 |
| Figuur 26 - schema overzicht van de verschillende stappen..... | 44 |
| Figuur 27 - Overzicht van de gebruikte technologieën | 46 |

Lijst met afkortingen

| | |
|------|-----------------------------------|
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| CPU | Central Processing Unit |
| CSS | Cascading Style Sheets |
| GPU | Graphics Processing Unit |
| HTML | Hypertext Markup Language |
| JSON | JavaScript Object Notation |

Verklarende woordenlijst

| | |
|-----------------------------|---|
| Chrome-extensie | Een kleine softwaretoepassing die kan worden toegevoegd aan de Chrome-browser om extra functionaliteit toe te voegen of het gedrag van de browser aan te passen. |
| Chrome web store | Een online marktplaats waar gebruikers extensies, thema's en apps kunnen vinden en installeren die speciaal ontworpen zijn voor de Google Chrome-webbrowser. |
| Service worker | Een script dat op de achtergrond draait in de webbrowser en wordt gebruikt om verschillende taken uit te voeren, zoals het beheren van cache, pushmeldingen en achtergrondtaken, om de prestaties en functionaliteit van webapplicaties te verbeteren. |
| Eye-tracking | Het proces van het meten en volgen van de bewegingen van de ogen om informatie te verkrijgen over waar iemand naar kijkt of hoe zijn of haar aandacht zich verplaatst. |
| Interacties op websites | Acties die gebruikers uitvoeren tijdens het gebruik van een website, zoals klikken op links, invullen van formulieren, scrollen, etc. |
| Navigatie | Het proces van het verplaatsen van de ene webpagina naar de andere binnen een website of van de ene website naar de andere |
| Toegankelijkheid op het web | Het vermogen van websites en web toepassingen om te worden gebruikt door mensen met verschillende niveaus van vermogen en beperkingen, waaronder mensen met een beperking. |
| Machine Learning Pipeline | Een machine learning-pijplijn is een reeks onderling verbonden gegevensverwerkings- en modelleringsstappen die zijn ontworpen om het proces van het bouwen, trainen, evalueren en implementeren van machine learning-modellen te automatiseren, standaardiseren en stroomlijnen |
| Face Landmark detection | Een techniek die wordt gebruikt om specifieke punten op het gezicht van een persoon te identificeren en te lokaliseren, zoals de locatie van de ogen, neus en mond. Dit wordt vaak gebruikt in toepassingen voor gezichtsherkenning, emotieanalyse en augmented reality. |

Interpolatie

Interpolatie is het afleiden van nieuwe datapunten binnen het bereik van een verzameling bekende discrete datapunten onder de veronderstelling van een zekere relatie tussen die punten.

1 Inleiding

Vrijwel iedereen maakt wel eens gebruik van het internet, zij het voor vermaak of uit noodzaak, zoals het maken van een online reservering. Hoewel dit voor de meeste mensen moeiteloos verloopt, kan het voor anderen een uitdaging vormen. Personen met verlammingen of beperkingen die het gebruik van een computermuis bemoeilijken, kunnen hierbij problemen ondervinden. Daarom ben ik geïnteresseerd geraakt in het vinden van oplossingen waarmee deze handelingen kunnen worden uitgevoerd zonder een computermuis te hoeven gebruiken. Dit bracht me bij eye-tracking als veelbelovende mogelijkheid. Ook stem herkenning lijkt een bijzonder goede oplossing te zijn. Maar dit kan voor een aantal problemen zorgen als je in een publieke ruimte zit. Andere mensen horen wat je aan het doen bent en als deze mensen ook aan het praten zijn kan dit als verkeerde informatie opgevangen worden door de stem herkenning. Daarom lijkt eye-tracking als de meer discrete optie.

Hier uit is de volgende onderzoeksvraag voortgekomen: "Kunnen oogbewegingen en knipperen via een Chrome-extensie vertaald worden naar website-interacties?"

Het onderzoek omvatte zeven onderdelen die worden opgesomd in deelvragen:

1. Hoe maak ik een Chrome-extensie?
2. Welke website interacties zijn er mogelijk via een Chrome-extensie?
3. Hoe gebruik je eye tracking in de Chrome-extensie?
4. Is externe software nodig bij het gebruik van eye tracking binnen de Chrome-extensie?
5. Wat zijn de voor- en nadelen van het gebruiken van een standaard camera t.o.v eye tracking glasses?
6. Hoe kan je de oog bewegingen registreren?
7. Hoe kan je acties registreren om bepaalde interacties uit te voeren?

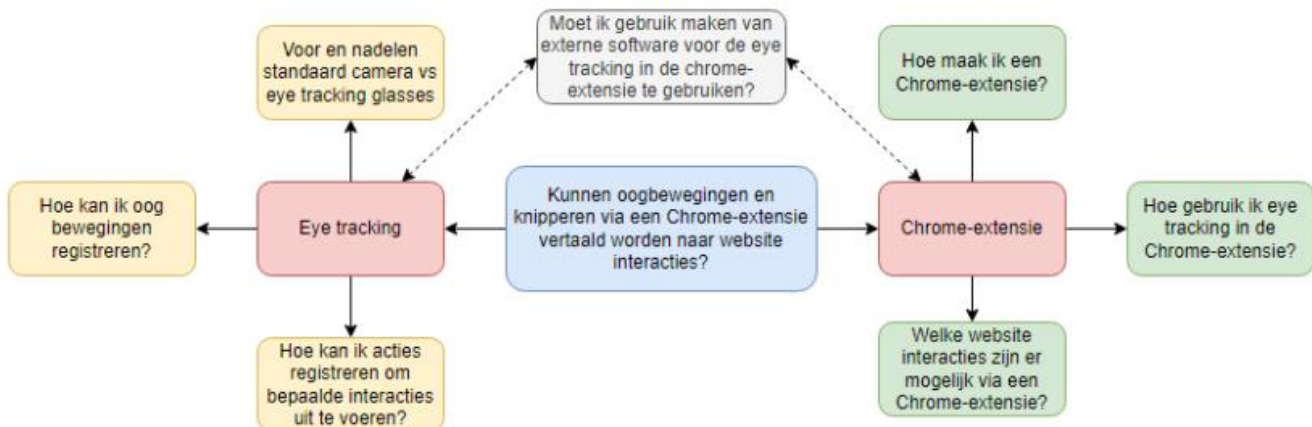
In deze bachelorproef wordt het technisch onderzoek uitvoerig besproken, inclusief de gebruikte technologieën en hoe deze samenwerken. Het uiteindelijke prototype zal ook worden geëvalueerd samen met experts uit het werkveld. Ik zal zelf ook reflecteren op het gevoerde onderzoek. Vervolgens zal er advies worden gegeven voor degenen die zelf aan de slag willen gaan met deze oplossing. Tot slot zal er een conclusie worden geformuleerd voor deze bachelorproef.

2 Research

Dit hoofdstuk biedt een uitgebreid overzicht van het onderzoek dat is uitgevoerd voor het project. Het onderzoek richt zich op het bepalen van de haalbaarheid van het project en het identificeren van de beste bibliotheken en benaderingen om het gewenste resultaat te bereiken. Het omvat een diepgaande verkenning van verschillende technische vraagstukken die van belang zijn voor het succesvol ontwikkelen van de Chrome-extensie voor eye-tracking.

Een Chrome-extensie is een kleine softwaretoepassing die functionaliteit toevoegt aan de Google Chrome-webbrowser. Deze extensies kunnen worden gedownload en geïnstalleerd vanuit de Chrome Web Store. Ze bieden diverse functies, zoals adblockers, wachtwoordmanagers, en tools voor productiviteit, die de gebruikerservaring verbeteren en aanpassen aan persoonlijke behoeften.

Hier is een overzicht van de technische vraagstukken die in dit hoofdstuk zullen worden behandeld. (Figuur 1)



Figuur 1 - schema overzicht deelvragen

2.1 Hoe maak je een Chrome-extensie?

De beslissing om een Chrome-extensie te gebruiken voor de implementatie van de eye-tracking technologie was gebaseerd op de wens om toegang te hebben tot een breed scala aan websites, niet beperkt tot zelfgemaakte sites.

Maar hoe wordt zo'n Chrome-extensie gemaakt? Het antwoord op deze vraag is te vinden in de officiële documentatie van Chrome for Developers. Hierin wordt stap voor stap uitgelegd hoe een eenvoudige voorbeeldextensie kan worden gemaakt. Ook wordt uitgelegd hoe deze extensie kan worden geladen als ontwikkelaar. Door de 'Ontwikkelaarmodus' in te schakelen, kunnen zelfgemaakte extensies worden gebruikt en getest zonder deze in de Chrome Web Store* te plaatsen. [1]

De initiële stappen lijken op het ontwikkelen van een standaard webpagina. Eerst wordt een HTML-pagina gemaakt om de lay-out van de extensie te definiëren. Optioneel kan een CSS-bestand worden toegevoegd voor styling, maar in dit specifieke project is ervoor gekozen om de styling rechtstreeks in de HTML-pagina in te voegen vanwege de beperkte omvang. Een JavaScript-bestand wordt echter wel gekoppeld voor de implementatie van specifieke functionaliteiten.

Tot op dit punt is het proces vergelijkbaar met het maken van een standaard webpagina. Voor de omzetting naar een Chrome-extensie is echter een extra bestand vereist: het "manifest.json" bestand. Dit bestand is van cruciaal belang voor een Chrome-extensie, aangezien het metadata bevat over de extensie, zoals de naam, versie, beschrijving, icoon en welke webpagina's de extensie mag aanpassen of toegang toe mag hebben. Het manifest voorziet Chrome van de nodige informatie om de extensie te installeren, configureren en uit te voeren.

Een volgende onderdeel dat belangrijk is om alle acties uit te kunnen voeren van een Chrome-extensie is een service worker*. Dit is een JavaScript-bestand dat toegang heeft tot alle pagina's die geopend zijn in je Chrome-browser. Dit bestand kan verschillende acties uitvoeren met deze pagina's. Ook de uitleg over deze service worker* is te vinden in de officiële documentatie van Chrome for Developers. Via deze service worker is het mogelijk om zelf Javascript bestanden te injecten* in de website pagina. Hierdoor heb je volledige controle over deze pagina en kan je aanpassingen toebrengen waar nodig. [2] [18] (Figuur 2)

```
chrome.runtime.onMessage.addListener((request, sender, sendResponse) => {
  if (
    request.action === 'simulatePageUp' || request.action === 'simulatePageDown' || request.action === 'startEyeTracking'
    || request.action === 'startBlinkDetection' || request.action === 'stop'
  ) {
    // Find the currently active tab
    chrome.tabs.query({ active: true, currentWindow: true }, function (tabs) {
      if (tabs.length > 0) {
        const tabId = tabs[0].id;
        // Send a message to the content script to simulate Page Up or Page Down
        chrome.scripting.executeScript({
          target: { tabId: tabId },
          function: (action) => {
            // Add Custom CSS - Function
            const Add_Custom_CSS = (css) => (document.head.appendChild(document.createElement('style')).innerHTML = css);
            // Create Custom Element - Function
            function Create_Custom_Element(tag, attr_tag, attr_name, value) { ...
            }
            // *** Action ***
            if (action === 'simulatePageUp') { ...
            } else if (action === 'simulatePageDown') { ...
            } else if (action === 'startEyeTracking') {
              // if script is already loaded, do nothing
              if (document.getElementById('GazeCloudApiScript')) {
                console.log('GazeCloudApi.js already loaded');
                return;
              }
              // Create script element
              const localScriptURL = chrome.runtime.getURL('GazeCloudApi.js');
              const scriptElement = document.createElement('script');
              scriptElement.id = 'GazeCloudApiScript';
              scriptElement.src = localScriptURL;
              // Set up onload event to ensure the script is fully loaded
              scriptElement.onload = function () { ...
            };
            // Append the script element to the head
            document.head.appendChild(scriptElement);
            } else if (action === 'startBlinkDetection') { ...
            } else if (action === 'stop') { ...
          }
        },
        args: [request.action],
      });
    });
  }
});
```

Figuur 2 – service worker code

Tenslotte is er ook nood aan een algemene opslagplaats om verschillende zaken in op te kunnen slaan. Zoals de informatie over welke acties de gebruiker koppelt aan welke interacties. Hiervoor kan geen gebruik maken van de locale opslag van een specifieke website, omdat de extensie op alle mogelijke websites moet werken. Gelukkig kunnen deze gegevens opgeslagen worden in de locale opslag van de Chrome-Extensie. Via de Chrome Storage API* kunnen gegevens weggeschreven en opgehaald worden. [3]

2.2 Welke website interacties zijn er mogelijk via een Chrome-extensie?

Vooraleer te starten met de ontwikkeling van dit project is het van cruciaal belang om grondig te onderzoeken welke functionaliteiten haalbaar zijn via een Chrome-extensie. [4]

Een van de belangrijkste aspecten van dit onderzoek is het verkennen van de mogelijkheid tot interactie met websites via de extensie. Specifiek zijn de volgende interacties van belang:

1. Scrollen

De eerste functionaliteit die is onderzocht, betreft het vermogen om webpagina's op en neer te scrollen. Dit kan relatief eenvoudig worden gerealiseerd door gebruik te maken van het 'window' element en de 'scrollBy()' functie aan te roepen om het scroll-effect te simuleren. Met deze functie kan de hoeveelheid en de richting van het scrollen worden aangegeven door respectievelijk een positief of negatief getal mee te geven. Dit biedt de mogelijkheid om een soepele en gecontroleerde scroll ervaring te creëren binnen de Chrome-extensie. [5]

(Figuur 3)

```
// If looking at top of the page, scroll up
if (GazeData.GazeY < 100) {
    window.scrollBy(0, -scrollSpeed);
}
// If looking at bottom of the page, scroll down
if (GazeData.GazeY > screenHeight2 - 100) {
    window.scrollBy(0, scrollSpeed);
}
```

Figuur 3 - detectie om scrollen te activeren

2. Klikken

Een andere nuttige interactie die is onderzocht, betreft het simuleren van muisklikken op webpagina's. Met behulp van JavaScript kunnen we het 'document' object gebruiken en de 'click()' methode aanroepen om een muisklik na te bootsen. Deze methode kan worden toegepast op specifieke elementen op de pagina, waardoor we interactie kunnen simuleren zoals een gebruiker die op een knop klikt. Hierbij kunnen we aangeven op welk element we willen klikken, en vervolgens wordt de actie uitgevoerd alsof deze door een gebruiker zelf werd uitgevoerd. [6] (figuur 4)

```
function EmulateClick() {
    console.log('EmulateClick');
    const element = document.elementFromPoint(
        GazePointer.getBoundingClientRect().x,
        GazePointer.getBoundingClientRect().y,
    );
    if (element) {
        element.click();
    }
}
```

Figuur 4 - klik naboots functie

3. Dubbel klikken

Het proces voor het nabootsen van dubbelklikacties op webpagina's verschilt van dat voor gewone klikacties. In plaats van de standaard 'click()' methode te gebruiken, maken we gebruik van een 'dblclick' mouse event en de 'dispatchEvent()' methode. Met deze benadering kunnen we dubbelklikacties simuleren door een dubbelklik-event te creëren en dit vervolgens af te vuren op specifieke elementen op de pagina. Hierdoor kunnen we hetzelfde gedrag genereren als wanneer een gebruiker daadwerkelijk twee keer achter elkaar op een element zou klikken. [7] (figuur 5)

```
function EmulateDoubleClick() {
  console.log('EmulateDoubleClick');
  const element = document.elementFromPoint(
    GazePointer.getBoundingClientRect().x,
    GazePointer.getBoundingClientRect().y,
  );
  if (element) {
    var doubleClickEvent = new MouseEvent('dblclick', {
      bubbles: true,
      cancelable: true,
      view: window,
    });
    element.dispatchEvent(doubleClickEvent);
  } else {
    console.log('No element found');
  }
}
```

Figuur 5 - dubbel klik naboots functie

4. Terug knop

Een andere interessante functionaliteit die is onderzocht, betreft het nabootsen van de 'terug' actie op webpagina's. In tegenstelling tot het simuleren van klikacties met behulp van events, maken we hier gebruik van een specifieke methode: 'window.history.back()'. Door deze methode aan te roepen, kunnen we de browser terug laten navigeren naar de vorige pagina in de geschiedenis. Dit creëert hetzelfde effect als wanneer een gebruiker op de 'terug' knop van de browser zou klikken. Deze methode kan handig zijn om gebruikersinteractie te automatiseren of om de navigatie-ervaring van de gebruiker te verbeteren binnen een Chrome-extensie of een webapplicatie. [8] (Figuur 6)

```
window.history.back();
```

Figuur 6 - terug knop functie

Dit zijn de vier interacties die in dit project aanbod komen om nagebootst te worden via de Chrome-extensie. Gelukkige blijkt dit mogelijk te zijn.

2.3 Hoe gebruik je eye tracking in de Chrome-extensie?

Het ontwikkelingsproces begon met het opzetten van een testwebsite om de eye-tracking-functionaliteit te kunnen testen, een essentiële eerste stap in het project. Dit bood de mogelijkheid om de eye-tracking geïsoleerd te evalueren, waardoor potentiële problemen konden worden geïdentificeerd zonder de invloed van andere factoren zoals de Chrome-extensie of specifieke websiteomgevingen. Het testen op een afzonderlijke testwebsite bood een zekere mate van verzekering dat eventuele problemen gerelateerd waren aan de eye-tracking software zelf. [9] [10] [11] [12]

Na het succesvol implementeren van de eye-tracking-functionaliteit op de testwebsite, was de volgende stap om deze functionaliteit te integreren in een Chrome-extensie. De aanpak die werd gebruikt om de eye-tracking aan de testwebsite toe te voegen, was relatief eenvoudig: een JavaScript-bestand werd toegevoegd aan de website om de functionaliteit te activeren. Echter, in plaats van dit bestand rechtstreeks aan de Chrome-extensie toe te voegen, moest het worden geïnjecteerd in het actieve Chrome-tabblad waarop de extensie werd gebruikt. Dit werd bereikt door de extensie zelf, die automatisch dit JavaScript-bestand injecteert in het huidige actieve Chrome-tabblad bij het openen van de extensie.

2.4 Is externe software nodig bij het gebruik van eye tracking binnen de Chrome-extensie?

Deze vraag richtte zich voornamelijk op de haalbaarheid van het afhandelen van eye-tracking via de Chrome-extensie, en of het noodzakelijk zou zijn om aparte software te gebruiken voor het registreren van oogbewegingen van de gebruiker. [13]

Beide opties zijn mogelijk. Enerzijds zijn er geavanceerde software-oplossingen die de oogbewegingen van de gebruiker op het scherm kunnen registreren, ongeacht de activiteiten die de gebruiker op zijn computer uitvoert. Deze software biedt diepgaande inzichten in het kijkgedrag van de gebruiker en kan waardevol zijn voor onderzoek naar gebruikerservaringen, marketingstrategieën en zelfs gezondheidszorgtoepassingen.

Anderzijds is er de mogelijkheid om alleen de oogbewegingen van de gebruiker op de websitepagina te volgen. Deze benadering vereist geen aparte software en kan worden bereikt door een eye-tracking JavaScript-bibliotheek toe te voegen aan de webpagina die door de Chrome-extensie wordt gebruikt.

Het kiezen tussen deze twee benaderingen hangt af van de specifieke vereisten van het project en de gewenste diepgang van de eye-tracking analyse. Voor sommige toepassingen kan de geavanceerde software de voorkeur hebben, terwijl voor andere situaties de meer gerichte benadering via de Chrome-extensie voldoende kan zijn. Het is belangrijk om de beschikbare opties te overwegen en de benadering te kiezen die het beste past bij de doelstellingen en de context van het project.

In het geval van dit project is het voldoende om enkel de oogbewegingen van de gebruiker te registreren op de websitepagina zelf. Het toevoegen van extra software zou een overbodige extra stap zijn voordat je deze Chrome-extensie kunt gebruiken.

2.5 Wat zijn de voor- en nadelen van het gebruiken van een standaard camera t.o.v eye tracking glasses?

Een cruciale overweging aan het begin van dit project was de keuze tussen het gebruik van een standaard webcam of eye-tracking glasses. Het antwoord op deze vraag bepaalde immers de koers van het verdere onderzoek. [14] [15]

Na een grondige analyse van de voordelen en nadelen van beide mogelijkheden zijn volgende bevindingen vastgesteld:

1. Nauwkeurigheid

Webcam: De nauwkeurigheid van een standaard webcam kan variëren, afhankelijk van factoren zoals resolutie en frame rate. Het kan moeite hebben om kleine oogbewegingen nauwkeurig vast te leggen.

Eye-tracking glasses: Eye-tracking glasses bieden over het algemeen een hogere nauwkeurigheid omdat ze direct de oogbewegingen van de gebruiker volgen zonder enige beperkingen van de cameraresolutie.

2. Eenvoudigheid

Webcam: Het gebruik van een standaard webcam is over het algemeen eenvoudiger omdat het geen extra hardware vereist en gemakkelijk toegankelijk is.

Eye-tracking glasses: Het gebruik van eye-tracking glasses kan complexer zijn vanwege de benodigde kalibratie en het dragen van extra apparatuur.

3. Kosten

Webcam: Een standaard webcam is meestal kosteneffectiever omdat het een breed beschikbare en relatief goedkope oplossing is.

Eye-tracking glasses: Eye-tracking glasses kunnen aanzienlijk duurder zijn vanwege de gespecialiseerde technologie en hardware die nodig is.

4. Functionaliteit

Webcam: Een webcam kan geschikt zijn voor basis eye-tracking toepassingen, maar kan beperkt zijn in geavanceerde functionaliteiten zoals het volgen van pupilverwijding.

Eye-tracking glasses: Eye-tracking glasses bieden geavanceerde functionaliteiten, zoals het volgen van pupilverwijding, die waardevol kunnen zijn voor specifieke toepassingen zoals neurowetenschappelijk onderzoek.

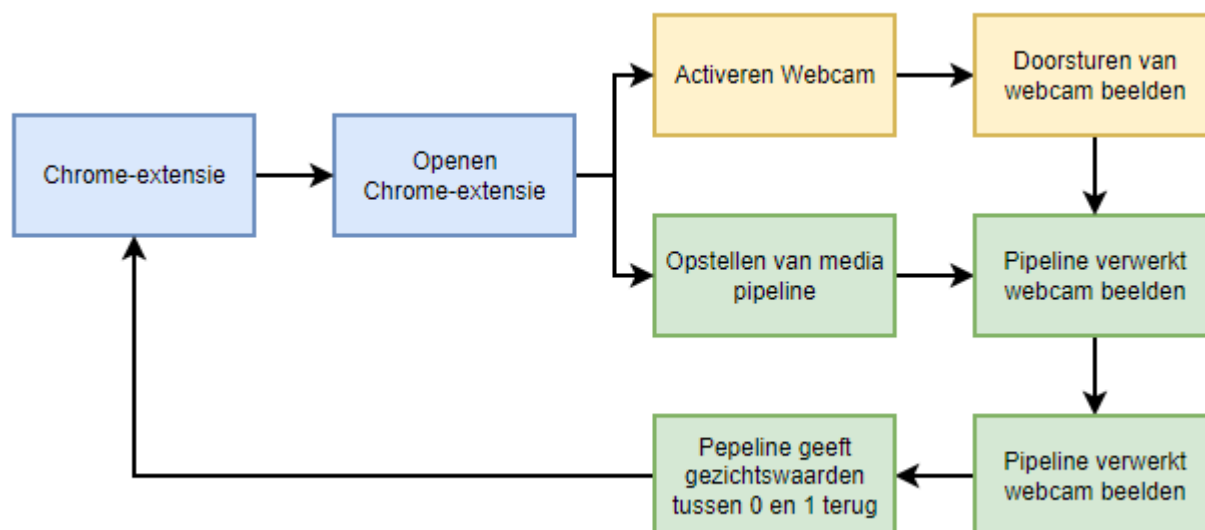
Deze bevindingen hebben mij geholpen bij het maken van een weloverwogen beslissing over welke eye-tracking methode het meest geschikt is voor mijn project. Het is belangrijk om deze aspecten zorgvuldig af te wegen, aangezien eenmaal genomen beslissingen moeilijk ongedaan te maken zijn en een grote invloed kunnen hebben op de verdere ontwikkeling van het onderzoek.

De keuze viel op het gebruik van een standaard webcam om twee belangrijke redenen: de wijdverbreide beschikbaarheid onder gebruikers en de focus op gebruiksvriendelijkheid.

De keuze voor een standaard webcam werd voornamelijk ingegeven door het feit dat de meeste mensen al over een dergelijk apparaat beschikken. Hierdoor wordt de toegankelijkheid vergroot en wordt de noodzaak voor extra aankopen of installaties geëlimineerd, waardoor het voor een breder publiek mogelijk wordt om deel te nemen aan het onderzoek of gebruik te maken van de Chrome-extensie.

2.6 Hoe kan je oog bewegingen registreren?

Om de oogbewegingen van de gebruiker te registreren, wordt gebruik gemaakt van de webcam. Deze webcam filmt de gebruiker tijdens het gebruik van de extensie, waarbij de beelden live worden verwerkt door een lokaal opgestelde machine learning pipeline*. Deze pipeline wordt geactiveerd bij het inladen van de Chrome-extensie, die een JavaScript-bestand injecteert in de huidige actieve webpagina. Dit script richt vervolgens de pipeline op, die een taak voor gezichtsoriëntatiepuntdetectie uitvoert. Deze taak retourneert verschillende waarden tussen 0 en 1 voor verschillende aspecten van het gezicht, waaronder diverse metingen voor de ogen. [16] [17] (Figuur 7 en 8)



Figuur 7 - Schema overzicht over hoe oog bewegingen worden geregistreerd



Figuur 8 - mediapipline output data

2.7 Hoe kan je acties registreren om bepaalde interacties uit te voeren?

Om de acties van de gebruiker te registreren, wordt de data geanalyseerd die wordt ontvangen van de gezichtsoriëntatiepuntdetectie. Op basis van deze waarden kan worden vastgesteld welke gezichtsuitdrukkingen de gebruiker vertoont. [17]

De geselecteerde acties voor registratie zijn gericht op het knippen van de gebruiker. Door gebruik te maken van de waarden 'EyeBlinkLeft' en 'EyeBlinkRight', kan worden bepaald of de gebruiker aan het knippen is. Een hoge waarde voor 'EyeBlinkLeft' en een lage waarde voor 'EyeBlinkRight' geven aan dat de gebruiker knipt met zijn linker oog, en vice versa voor een knipoog met het rechter oog. Wanneer beide waarden tegelijkertijd hoog zijn, duidt dit op een normale knipperslag, die automatisch wordt genegeerd vanwege het natuurlijke karakter ervan. Echter, er wordt wel gecontroleerd op het snel achter elkaar knippen met beide ogen, wat resulteert in de derde actie: de dubbele knipper.

Deze drie knipperacties zijn: "linker knipoog", "rechter knipoog" en "dubbele knipper". Deze acties op zichzelf resulteren niet automatisch in interacties op de webpagina. Elke actie moet nog worden gekoppeld aan een specifieke interactie. Dit kan worden gedaan door de Chrome-extensie te openen. Wanneer de Chrome-extensie wordt geopend, verschijnt er een pop-up in de rechter bovenhoek van de pagina met de beschreven acties. Elk van deze acties kan vervolgens worden gekoppeld aan een gewenste interactie op de webpagina (Figuur 9).

Een vierde actie die de gebruiker kan uitvoeren, is simpelweg naar de pagina kijken. Op basis van de locatie waar de gebruiker op de pagina kijkt, kan worden bepaald of er moet worden gescrold of niet. Deze actie wordt afzonderlijk behandeld en verloopt niet via de gezichtsoriëntatiepuntdetectie, maar via de eye-tracking.

Als de gebruiker naar de onderkant van zijn scherm kijkt, wordt de pagina automatisch naar beneden gescrold. De scrol snelheid is constant en de pagina blijft scrollen totdat de gebruiker zijn blik weer naar het midden van het scherm verplaatst. Op dezelfde manier zal de pagina omhoog scrollen als de gebruiker naar het bovenste deel van zijn scherm kijkt, weer tot de gebruiker zijn blik terugbrengt naar het midden van het scherm.

Deze scrol snelheid kan opnieuw worden ingesteld via de pop-up van de Chrome-extensie (Figuur 9), waardoor gebruikers de mogelijkheid hebben om de scrol ervaring aan te passen aan hun persoonlijke voorkeuren en gebruiksgemak.

Eye tracking extension

Eye tracking

Stop

Settings

Scroll speed



Blink options

Left eye blink

Back ▼

Right eye blink

Click ▼

Double eye blink

Double Click ▼

Figuur 9 - chrome-extensie pop-up

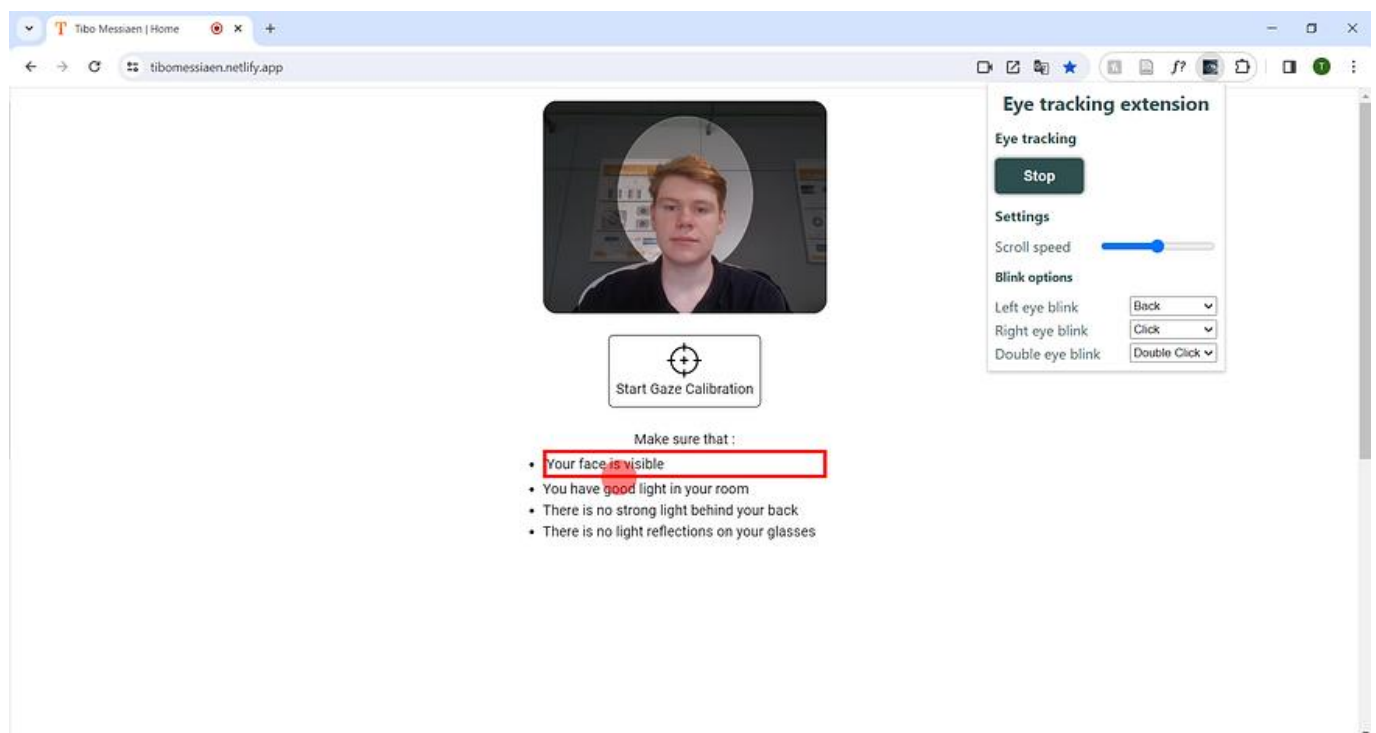
3 Technisch onderzoek

Het eindproduct van dit technische onderzoek is een volledig werkende **Chrome-extensie** die via **eye-tracking** en **gezichtsoriëntatiepunctdetectie** de gebruiker in staat stelt om elke website aan te sturen. (Figuur 10)

In de bijlage vindt u een installatie- en gebruikershandleiding die stap voor stap uitlegt hoe het project, besproken in deze bachelorproef, geïnstalleerd en gebruikt kan worden. In dit hoofdstuk wordt uitgebreid ingegaan op de geschreven code, de bevindingen, de denkwijze, en de keuze voor de gebruikte technologieën.

Het ontwikkelen van deze Chrome-extensie vereiste een grondige analyse van verschillende technologieën en methoden voor het volgen van oogbewegingen en gezichtsoriëntatiepunctdetectie. Door de juiste combinatie van deze technologieën toe te passen, kon een robuuste en gebruiksvriendelijke extensie worden gecreëerd die een nieuwe dimensie toevoegt aan de interactie tussen gebruiker en webpagina's.

Deze bachelorproef beoogt niet alleen de functionaliteit van de extensie te demonstreren, maar ook inzicht te bieden in het ontwikkelingsproces, de technische uitdagingen en de besluitvorming achter de gebruikte technologieën. Het doel is om bij te dragen aan zowel de academische kennis als de praktische waarde voor toekomstige ontwikkelaars en onderzoekers op het gebied van interactieve technologieën.



Figuur 10 - eye tracking extension schermafbeelding

3.1 Website pagina maken

Het ontwikkelen van een Chrome-extensie begint met het creëren van een standaard webpagina, die zal fungeren als de popup die de gebruiker te zien krijgt bij het openen van de extensie in Chrome. De eerste stap in het proces van het maken van een webpagina is het opstellen van een **HTML-bestand**. Dit kan worden bereikt door een bestand aan te maken met een geschikte naam en deze te voorzien van de extensie .html. Dit bestand bepaalt de visuele weergave voor de gebruiker. Op dit moment omvat dit slechts een kleine witte popup in de rechterbovenhoek van het scherm, met de naam van de extensie als titel, samen met ruimte voor instellingen die in latere stadia van dit project zullen worden ontwikkeld.

3.2 Website omzetten naar Chrome-extensie

Het proces van het omzetten van een website naar een Chrome-extensie omvat het opstellen van een **manifest.json-bestand**, waarin verschillende essentiële gegevens over de extensie worden opgenomen. (Figuur 11)

Een cruciaal aspect van het manifest.json-bestand is het specificeren van de manifestversie, die de correcte werking en compatibiliteit van de extensie waarborgt. Voor deze extensie is gekozen voor versie 3, de meest recente beschikbare versie op dit moment, om ervoor te zorgen dat de extensie zo lang mogelijk kan worden gebruikt. Versie 1 is al niet meer bruikbaar en versie 2 zal ook in 2024 stoppen met werken.

Naast de manifestversie kunnen ook details over de extensie worden opgenomen, zoals de **naam**, **icoontjes** en **beschrijving**. Het specificeren van een versienummer maakt het mogelijk om verschillende versies van de extensie uit te brengen en bij te houden.

Verder vereist het manifest.json-bestand ook de specificatie van een **"service worker"**, een bestand dat op de achtergrond blijft werken, zelfs wanneer de extensie niet actief is. Dit draagt bij aan een naadloze gebruikerservaring en biedt mogelijkheden voor offline functionaliteit.

Een ander belangrijk aspect is het opgeven van de **"default popup"**, waarbij de naam van de HTML-pagina wordt vermeld die standaard wordt geladen wanneer de extensie wordt geopend. Voor deze extensie is dit bijvoorbeeld "popup.html".

Daarnaast moeten ook de **Content Security Policy (CSP)**, web accessible resources en content scripts worden opgenomen in het manifest.json-bestand. De Content Security Policy bepaalt welke bronnen kunnen worden geladen en uitgevoerd binnen de extensie, terwijl **web accessible resources** worden gespecificeerd om ervoor te zorgen dat externe bronnen correct worden geladen. Content scripts kunnen ook worden toegevoegd om interactie met webpagina's mogelijk te maken en functionaliteit toe te voegen aan specifieke sites.

Door al deze stappen te volgen en de juiste informatie op te nemen in het manifest.json-bestand, kan een website effectief worden omgezet naar een functionele Chrome-extensie.

```

"manifest_version": 3,
"name": "Page Scroll Extension",
"version": "1.0",
"description": "An extension to scroll the active tab using Page Up and Page Down buttons.",
"permissions": ["storage", "declarativeContent", "activeTab", "scripting"],
"content_security_policy": {
  "extension_pages": "script-src 'self'; object-src 'self'; script-src-elem 'self' 'unsafe-inline' https://api.gazerecorder.com/GazeCloudAPI.js"
},
"web_accessible_resources": [
  {
    "resources": ["GazeCloudApi.js", "MediapipeFaceLandmarker.js"],
    "matches": ["<all_urls>"]
  }
],
"icons": {
  "16": "img/icon16.png",
  "48": "img/icon48.png"
},
"action": {
  "default_popup": "popup.html"
},
"background": {
  "service_worker": "background.js"
},
"content_scripts": [
  {
    "matches": ["<all_urls>"],
    "js": ["content.js"],
    "web_accessible_resources": [
      {
        "resources": ["GazeCloudApi.js", "MediapipeFaceLandmarker.js"],
        "matches": ["<all_urls>"]
      }
    ]
  }
]

```

Figuur 11 - manifest.json code overview

3.3 Eye tracking uit testen

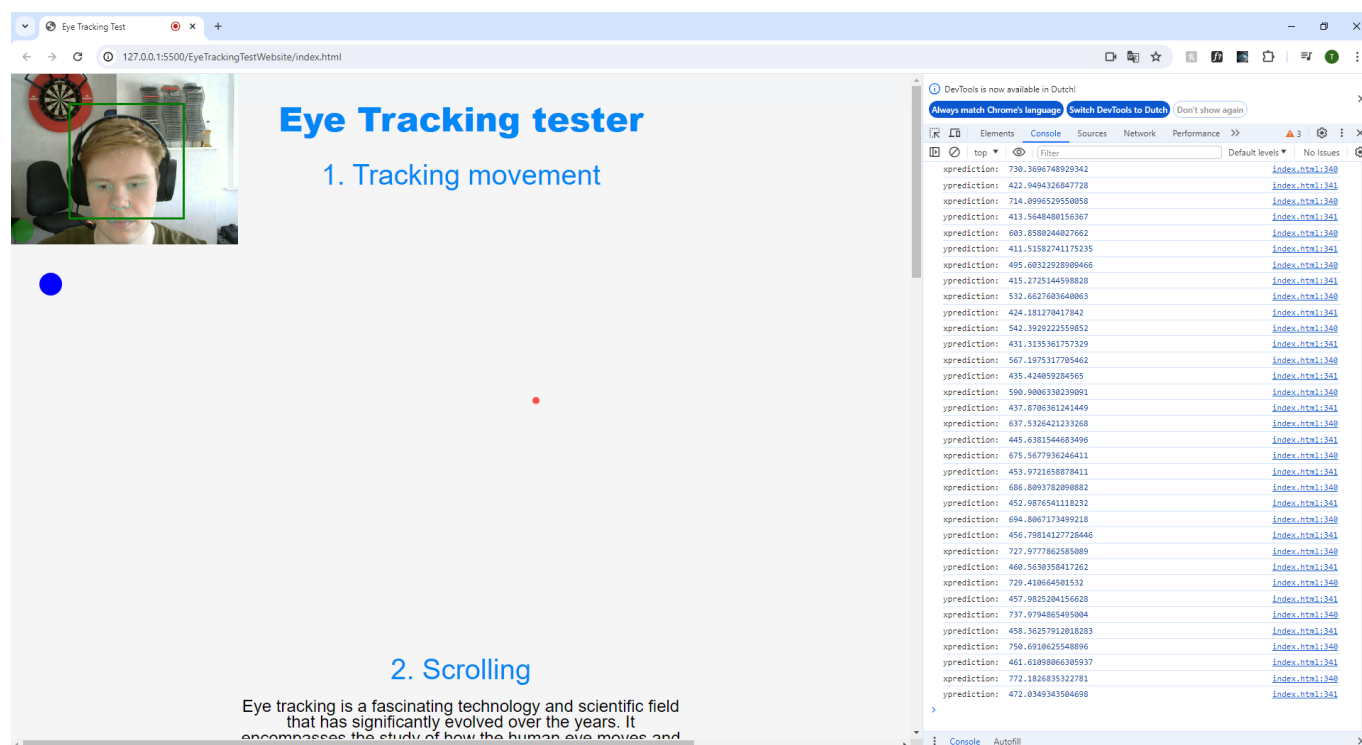
Eye tracking vormt een essentieel onderdeel van deze Chrome-extensie en is wellicht het meest cruciale aspect ervan. Om een weloverwogen keuze te maken voor de eye-tracking software, zijn verschillende opties grondig getest. WebGazer en GazeCloudAPI zijn twee van de uitgebreid onderzochte mogelijkheden voor het live registreren van oogbewegingen met behulp van een computer webcam.

Deze twee opties zijn zorgvuldig vergeleken door middel van tests op een speciaal ontwikkelde **testwebsite**. Deze website is op maat gemaakt om de functionaliteit van de eye-tracking software grondig te evalueren en te vergelijken.

3.3.1 Webgazer

WebGazer is een **JavaScript-bibliotheek** die in 2016 is ontwikkeld en zich origineel heeft gebaseerd op een onderzoek dat is uitgevoerd door Brown University. Deze software wordt gebruikt voor het uitvoeren van **real-time eye-tracking** direct in de webbrowser. Met WebGazer kunnen ontwikkelaars eenvoudig **oogbewegingen** van gebruikers **registreren** en **analyseren** terwijl ze een website bezoeken. De bibliotheek maakt gebruik van de webcam van de gebruiker om de positie van de ogen te volgen en zo te bepalen waar de gebruiker naar kijkt op het scherm. WebGazer is een handige tool voor het verbeteren van de gebruikerservaring, het verzamelen van analytische gegevens over gebruikersgedrag en het ontwikkelen van interactieve web toepassingen die reageren op oogbewegingen.

Deze JavaScript-bibliotheek is toegevoegd aan de zelf opgestelde website om dit eens grondig uit te testen. Wanneer WebGazer wordt geactiveerd, opent het de webcam van de gebruiker en begint het met het vastleggen van videobeelden om **gezichtsdetectie** uit te voeren. Hierdoor kan het de locatie van het gezicht en de positie van de ogen van de gebruiker bepalen aan de hand van **gezichtspunten**. Zodra de ogen zijn gedetecteerd, analyseert WebGazer hun positie om te **voorspellen waar de gebruiker naar kijkt** op het scherm. Deze voorspelling wordt weergegeven door een klein rood puntje op het scherm van de gebruiker. Hoewel de nauwkeurigheid van deze voorspelling aanvankelijk mogelijk niet optimaal is, kan deze worden verbeterd door gebruikers te vragen te klikken op de plaats waar ze daadwerkelijk naar kijken, waardoor de algoritmen van WebGazer kunnen worden aangepast en verfijnd. (Figuur 12)

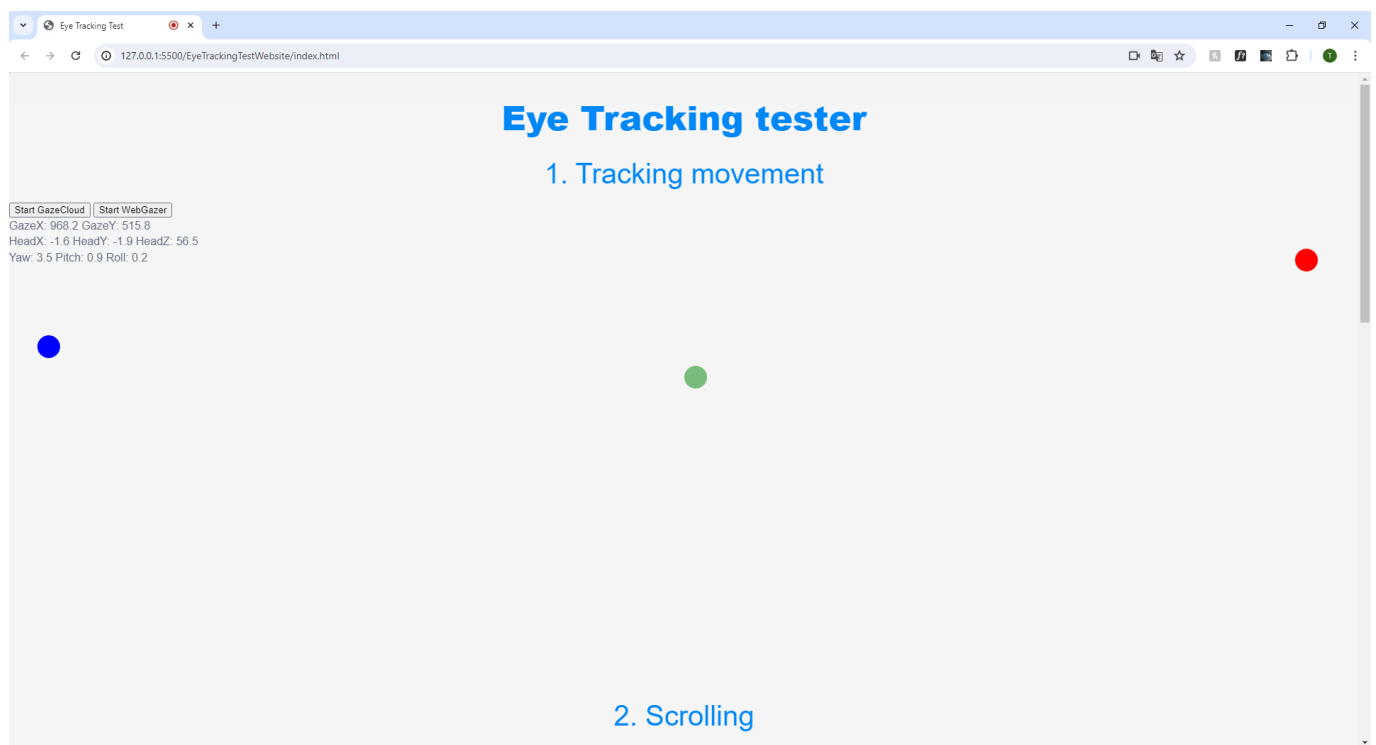


Figuur 12 - eye tracking test website - Webgazer - schermafbeelding

3.3.2 GazeCloudAPI

GazeCloudAPI is een **JavaScript-bibliotheek** die onderdeel uitmaakt van GazeRecorder, een bedrijf die verschillende eye-tracking api's heeft. Deze software wordt gebruikt voor het uitvoeren van **real-time eye-tracking** direct in de webbrowser. Met GazeCloudAPI kunnen ontwikkelaars eenvoudig **oogbewegingen** van gebruikers **registreren en analyseren** terwijl ze een website bezoeken. De bibliotheek maakt gebruik van de webcam van de gebruiker om de positie van de ogen te volgen en zo te bepalen waar de gebruiker naar kijkt op het scherm. GazeCloudAPI is een handige tool voor het verbeteren van de gebruikerservaring, het verzamelen van analytische gegevens over gebruikersgedrag en het ontwikkelen van interactieve web toepassingen die reageren op oogbewegingen.

Deze JavaScript-bibliotheek is toegevoegd aan de zelf opgestelde website om dit eens grondig uit te testen. Wanneer GazeCloudAPI wordt geactiveerd, opent het de webcam van de gebruiker en begint het met het vastleggen van videobeelden om gezichtsdetectie uit te voeren. Hierdoor kan het de locatie van het gezicht en de positie van de ogen van de gebruiker bepalen aan de hand van gezichtspunten. Zodra de **ogen zijn gedetecteerd**, analyseert GazeCloudAPI hun **positie** om te **voorspellen waar de gebruiker naar kijkt** op het scherm. Deze voorspelling word weergegeven door een groene punt op het scherm van de gebruiker. Hoewel de nauwkeurigheid van deze voorspelling aanvankelijk mogelijk niet optimaal is, zal deze veel verbeteren bij het afleggen van de **ingebouwde kalibratie** test. Deze test word automatisch gestart bij het activeren van GazeCloudAPI. Als er gebruik gemaakt word van de **gratis demo versie** van GazeCloudAPI zal deze wel maar voor twee minuten meer werken na de kalibratie. Daarna komen er geen voorspellingen meer binnen. (Figuur 13)



Figuur 13 - eye tracking test website - GazeCloudAPI - schermafbeelding

3.3.3 Eye tracking keuze

WebGazer en **GazeCloudAPI** bieden beide unieke voordelen en hebben hun eigen set van nadelen. Het is essentieel om deze aspecten te vergelijken om een weloverwogen keuze te maken.

Zowel WebGazer als GazeCloudAPI zijn gratis beschikbaar, hoewel de **gratis** versie van GazeCloudAPI beperkt is tot een demo die slechts **twee minuten** werkt, er is ook een betalende versie waarbij je 400 euro betaald waarbij je 10.000 minuten aan eye-tracking tijd krijgt. Dit zijnde voor niet commercieel gebruik. Beide bibliotheken zijn relatief **eenvoudig** te implementeren in het project en bieden ingebouwde **kalibratiemogelijkheden**. Echter, de kalibratie van WebGazer vereist interactie van de gebruiker door te klikken met de muis. Dit kan de gebruikerservaring beïnvloeden, vooral voor diegenen die een volledig handsfree bediening willen hebben. (Figuur 14)

Na zorgvuldige afweging lijkt **GazeCloudAPI de meest geschikte keuze** voor dit project te zijn. Dit komt voornamelijk doordat de **kalibratie** van WebGazer interactie met de muis vereist, wat de volledig **handsfree** bediening van de Chrome-extensie belemmert. Bovendien is GazeCloudAPI **sneller** dan WebGazer, wat resulteert in een vloeiendere en responsievere gebruikerservaring. De snelheid van GazeCloudAPI kan van cruciaal belang zijn voor het nauwkeurig volgen van oogbewegingen en het bieden van real-time feedback aan de gebruiker.

| WebGazer | |
|-------------------------|-----------------------------|
| Voordelen | Nadelen |
| Gratis | Kalibratie via muis klikken |
| Open-source | Traag |
| Simpel te implementeren | |
| Onbeperkte duur | |

| GazeCloudAPI | |
|-------------------------------------|---------------------------------|
| Voordelen | Nadelen |
| Gratis (demo) | Gratis versie beperkt tot 2 min |
| Open-source | Kalibratie word niet onthouden |
| Simpel te implementeren | |
| Ingebouwde kalibratie test via ogen | |
| Snel | |

Figuur 14 – voordelen en nadelen van WebGazer en GazeCloudAPI

3.4 Face landmark Detection uittesten

Voor het **registreren van de acties** van de gebruiker is ervoor gekozen om ook naar de ogen van de gebruiker te kijken, specifiek naar het knipperen van de ogen. Door te **knipperen** met het **linker-** of **rechteroog**, of door snel twee keer met beide ogen te knipperen, kan de gebruiker verschillende acties oproepen.

Om deze acties te registreren, **is aparte software** nodig die verschillende punten van het gezicht uitleest en bepaalt of de gebruiker aan het knipperen is. PyGaze en de Face Landmark Detection van MediaPipe zijn twee van dergelijke mogelijkheden.

PyGaze is een tool die specifiek is ontworpen voor eye-tracking in psychologisch onderzoek, met een focus op nauwkeurigheid en geavanceerde analyse van oogbewegingen. Aan de andere kant biedt MediaPipe's Face Landmark Detection een krachtige oplossing voor het detecteren van gezichtskenmerken, waaronder de positie van de ogen, op basis van machine learning-modellen.

Beide opties zijn apart getest om te beoordelen hoe goed ze werken en of ze gemakkelijk te implementeren zijn in het project. Dit evaluatieproces zal helpen bij het bepalen welke software het meest geschikt is voor het registreren van oogbewegingen en het activeren van gebruikersacties binnen de Chrome-extensie.

3.4.1 PyGaze

PyGaze is een **Python-bibliotheek** die wordt gebruikt voor het ontwikkelen van experimenten en toepassingen op het gebied van oogbeweging onderzoek. Met PyGaze kunnen onderzoekers oogbewegingen registreren, analyseren en visualiseren tijdens het uitvoeren van experimenten. De bibliotheek biedt verschillende functies en tools voor het aansturen van eye-trackers, het presenteren van visuele stimuli en het registreren van oogbewegingsgegevens. (Figuur 15)



Figuur 15 - pygaze blink detection

3.4.2 MediaPipe Face landmark Detection

MediaPipe Face Landmark Detection is een geavanceerde tool voor het real-time detecteren en volgen van gezichtskenmerken. Het maakt gebruik van **machine learning-modellen** om nauwkeurig punten op het gezicht te identificeren, zoals de ogen, neus en mond. Met deze technologie kunnen ontwikkelaars interactieve toepassingen creëren die gezichtsgerichte analyse en tracking mogelijk maken, zoals gezichtsfilters, gezichtsherkenning en augmented reality.

Bovendien biedt MediaPipe verschillende **codevoorbeelden** die **illustreren** hoe deze tool kan worden gebruikt. Een van deze voorbeelden is gericht op web toepassingen en omvat functionaliteiten zoals het uitlezen van webcambeelden van de gebruiker. Het voorbeeld biedt een overzicht van alle gedetecteerde gezichtspunten en voorspelt **gezichtsuitdrukkingen** en **bewegingen** met een getal tussen **0** en **1**. Een waarde dicht bij 1 geeft een hogere kans aan dat de uitdrukking of beweging van toepassing is, terwijl een waarde van 0 aangeeft dat deze niet van toepassing is.

3.4.3 Face landmark Detection keuze

Na het uitvoerig testen van zowel PyGaze als de Face Landmark Detection van MediaPipe, werd al snel duidelijk dat **MediaPipe** de **superieure keuze** was. Niet alleen bleek MediaPipe **aanzienlijk sneller** te zijn, maar het leverde ook veel **nauwkeurigere** resultaten op.

Bovendien is de Face Landmark Detection van MediaPipe beschikbaar als een **JavaScript-bibliotheek**, waardoor het eenvoudig kan worden geïntegreerd in websites. Wat het nog aantrekkelijker maakte, is dat deze tool ook de mogelijkheid biedt om zijn verwerkingskracht te laten draaien op de **CPU** in plaats van de **GPU**, waardoor de **prestaties** nog verder worden **verbeterd**.

3.5 Eye tracking & Face landmark Detection toevoegen aan Chrome-extensie

Het proces van het integreren van de gekozen eye-tracking software, GazeCloudAPI en Face landmark Detection (MediaPipe), in de Chrome-extensie vereist zorgvuldige **stappen** om ervoor te zorgen dat deze bibliotheken **correct functioneren** in het actieve Chrome-tabblad waar de extensie wordt gebruikt.

In eerste instantie worden de GazeCloudAPI en Face landmark Detection-code gedownload en **lokaal aan het project toegevoegd** als JavaScript-bestanden. Vervolgens worden deze bestanden **via de service worker toegevoegd**. De service worker zoekt eerst naar het **actieve Chrome-tabblad**. Nadat dit tabblad is geïdentificeerd, wordt de ID ervan opgehaald en bijgehouden. Deze ID wordt vervolgens doorgegeven aan het **executeScript-commando** van Chrome, waardoor de verdere acties op het juiste tabblad worden uitgevoerd. In deze functie worden de lokale GazeCloudAPI- en Face landmark Detection-codes **als scriptelementen toegevoegd** aan dit tabblad. Bovendien worden HTML- en bijbehorende CSS-elementen toegevoegd aan de pagina, die dienen als markeringen om aan te geven waar de gebruiker op het scherm kijkt.

De functie die het **GazeCloudAPI-bestand als script inlaadt**, wordt uitgevoerd wanneer een actieverzoek met de naam "startEyeTracking" wordt ontvangen. Op vergelijkbare wijze wordt de functie die het Face landmark Detection-script inlaadt, uitgevoerd wanneer een actieverzoek met de naam "startBlinkDetection" wordt ontvangen. Deze **"startEyeTracking"** en **"startBlinkDetection" commando's** worden **automatisch doorgegeven** bij het laden van de Chrome-extensie, waardoor deze functies automatisch worden gestart wanneer de extensie wordt geopend.

3.6 Interacties opstellen

Nu dat de eye-tracking en Face landmark Detection zijn toegevoegd aan de extensie is het tijd om de effectieve acties die moeten worden uitgevoerd toe te voegen. Dit zijn de basis acties die de gebruiker moet kunnen uitvoeren bij het gebruiken van een website. **Scrollen, klikken, dubbel klikken en terug navigeren.**

3.6.1 Scrollen nabootsen

Om de functionaliteit toe te voegen waarmee gebruikers via de Chrome-extensie kunnen scrollen op een website, wordt eye-tracking data gebruikt. Door enkele code-regels toe te voegen aan het GazeCloudAPI-bestand wordt gecontroleerd of de gebruiker naar de **boven- of onderkant** van het scherm **kijkt**. Wanneer de gebruiker naar de bovenste 100 pixels van het scherm kijkt, wordt er **omhoog gescrold**. Op dezelfde manier, als de gebruiker naar de onderste 100 pixels kijkt, wordt er naar **beneden gescrold**. (Figuur 16)

Hiermee krijgt de gebruiker de mogelijkheid om op en neer te scrollen in de website door simpelweg naar boven of beneden te kijken.

```
// If looking at top of the page, scroll up
if (GazeData.GazeY < 100) {
    window.scrollTo(0, -scrollSpeed);
}
// If looking at bottom of the page, scroll down
if (GazeData.GazeY > screenHeight2 - 100) {
    window.scrollTo(0, scrollSpeed);
}
```

Figuur 16 - scroll actie code

3.6.2 Muis klik nabootsen

Om een muisklik na te bootsen, kan de **click()** functie van een HTML-element worden gebruikt. Het is echter noodzakelijk om te bepalen op welk HTML-element deze klik moet worden gesimuleerd. Dit element kan worden verkregen door de **document.elementFromPoint()** functie aan te roepen en de coördinaten door te geven van waar de gebruiker naar kijkt. Deze functie retourneert het HTML-element dat zich op die specifieke positie bevindt. Als de **click()** functie vervolgens op dit element wordt toegepast, lijkt het alsof de gebruiker op dat element heeft geklikt met zijn muis. Deze code regels zijn toegevoegd aan het Face landmark Detection bestand, aangezien deze functie zal worden aangesproken door een knipper actie van de gebruiker. (Figuur 17)

```
function EmulateClick() {
  console.log('EmulateClick');
  const element = document.elementFromPoint(
    GazePointer.getBoundingClientRect().x,
    GazePointer.getBoundingClientRect().y,
  );
  if (element) {
    element.click();
  }
}
```

Figuur 17 - klik naboots functie code

3.6.3 Dubbel klik nabootsen

Om een **dubbelklik** na te bootsen, is een iets andere aanpak nodig dan bij een normale klikfunctie. In plaats van simpelweg de **click()** functie te gebruiken, moet er een **MouseEvent** worden aangemaakt en vervolgens worden uitgevoerd via de **dispatchEvent()** functie.

Het proces begint met het identificeren van het HTML-element waarop de dubbelklik moet worden uitgevoerd. Vervolgens wordt een nieuw **MouseEvent** aangemaakt met het type **"dblclick"** (dubbelklik). Dit **MouseEvent** omvat alle relevante informatie over de dubbelklik, zoals de coördinaten van de cursor op het moment van de klik.

Nadat het **MouseEvent** is gemaakt, wordt het uitgevoerd op het gewenste HTML-element door middel van de **dispatchEvent()** functie. Hierdoor wordt de dubbelklik gesimuleerd, alsof de gebruiker twee keer snel achter elkaar op dat specifieke element heeft geklikt. (Figuur 18)

Deze code regels zijn opnieuw toegevoegd aan het Face landmark Detection bestand, aangezien deze functie zal worden aangesproken door een knipper actie van de gebruiker.

Het nabootsen van een dubbelklik vereist dus een iets complexere benadering dan een normale klik, maar met behulp van een **MouseEvent** en de juiste afhandeling via **dispatchEvent()** kan dit effectief worden bereikt.


```
function EmulateDoubleClick() {
  console.log('EmulateDoubleClick');
  const element = document.elementFromPoint(
    GazePointer.getBoundingClientRect().x,
    GazePointer.getBoundingClientRect().y,
  );
  if (element) {
    var doubleClickEvent = new MouseEvent('dblclick', {
      bubbles: true,
      cancelable: true,
      view: window,
    });
    element.dispatchEvent(doubleClickEvent);
  } else {
    console.log('No element found');
  }
}
```

Figuur 18 - dubbel klik naboots functie code

3.6.4 Terug knop nabootsen

Om de terug actie van de browser na te bootsen, wordt de JavaScript-functie **window.history.back()** gebruikt. In tegenstelling tot het nabootsen van een klik of dubbelklik, waarbij interactie met een specifiek HTML-element vereist is, maakt deze functie direct gebruik van de browsergeschiedenis om terug te gaan naar de vorige pagina. (Figuur 19)

Wanneer deze functie wordt aangeroepen, wordt de browserinformatie over de sessiegeschiedenis van de gebruiker gebruikt om terug te keren naar de vorige pagina in de browsergeschiedenis. Dit simuleert hetzelfde gedrag als wanneer de gebruiker op de terugknop van de browser zou klikken.

Het gebruik van window.history.back() vereenvoudigt het proces van het terugkeren naar de vorige pagina aanzienlijk, omdat het direct toegang biedt tot de browsergeschiedenis en de terug actie uitvoert zonder dat er specifieke HTML-elementen of coördinaten nodig zijn. Hierdoor kan de Chrome-extensie de gebruiker een naadloze en intuïtieve browse-ervaring bieden, waarbij de terugactie moeiteloos kan worden uitgevoerd met een simpele aanroep van de JavaScript-functie.

Deze code regels zijn opnieuw toegevoegd aan het Face landmark Detection bestand, aangezien deze functie zal worden aangesproken door een knipper actie van de gebruiker.

```
window.history.back();
```

Figuur 19 - terug actie code regel

3.6.5 Markeren van element die bekeken worden

Het markeren van een HTML-element is een manier om aan de gebruiker duidelijk te maken naar welk element hij of zij kijkt en waarop mogelijk een actie zal worden uitgevoerd.

Om te bepalen naar welk element de gebruiker kijkt, wordt opnieuw gebruikgemaakt van de **document.elementFromPoint()-functie**, waarbij de **coördinaten** van het punt waar de gebruiker naar kijkt worden meegegeven. Het element dat door deze functie wordt geretourneerd, wordt vervolgens gemarkeerd.

De markering van het element wordt gedaan door het element een rode rand te geven, waardoor het duidelijk wordt welk element op dat moment is geselecteerd en waarop de actie zal worden uitgevoerd. (Figuur 20)

Deze functionaliteit is geïmplementeerd in het GazeCloudAPI-bestand, aangezien het markeren van de elementen afhankelijk is van het element waar de gebruiker naar kijkt. Hierdoor kan de gebruiker **visueel** worden **geïnformeerd** over welk element op het scherm wordt **gefocust**, wat de gebruikerservaring verbetert en de interactie met de Chrome-extensie intuïtiever maakt.

```
function HighlightElement(GazePointerElement) {
  try {
    const element = document.elementFromPoint(
      GazePointerElement.getBoundingClientRect().x,
      GazePointerElement.getBoundingClientRect().y,
    );
    if (element) {
      // Reset old element back to original border
      if (oldElement != null) oldElement.style.border = oldElementBorder;
      // Set new element to be the old element
      oldElement = element;
      // Set the new border to be the old border
      oldElementBorder = element.style.border;
      // Set the new border to be red
      element.style.border = '4px solid red';
    }
  } catch (e) {
    console.log(e);
  }
}
```

Figuur 20 - element markeren functie code

3.7 Gebruiker acties registreren

Voor het uitvoeren van alle interacties in deze Chrome-extensie zijn er drie acties die nog moeten worden aangestuurd door een knippen met de ogen. Deze acties zijn gedefinieerd als een linker knipoog, rechter knipoog en een dubbele knippering binnen dezelfde seconde.

Om te bepalen wanneer deze acties plaatsvinden, wordt gebruikgemaakt van de 'eyeBlinkLeft' en 'eyeBlinkRight' waarden van de MediaPipe Face Landmark Detection. Door naar deze waarden te kijken, kan worden vastgesteld welke van de drie acties er wordt uitgevoerd. Voor de linker knipoog actie wordt gedetecteerd of de '**eyeBlinkLeft**' waarde **hoger is dan 0.5** en de '**eyeBlinkRight**' waarde **lager is dan 0.5**. Voor de rechter knipoog actie wordt het tegenovergestelde verwacht. Bij een dubbele knippering worden beide 'eyeBlinkLeft' en 'eyeBlinkRight' waarden geëvalueerd, waarbij wordt gecontroleerd of beide waarden boven 0.5 liggen binnen dezelfde seconde. (Figuur 21)

Het is belangrijk op te merken dat de vorige knipper wordt bijgehouden en dat hiernaar wordt gekeken om te bepalen of **er effectief wordt geknipperd** en of de ogen **niet** worden **dichtgehouden**. Dit voorkomt dat de acties meerdere keren worden uitgevoerd als gevolg van een aanhoudende oogsluiting. Zo wordt een nauwkeurige en betrouwbare detectie van knipperacties gegarandeerd.

Op deze manier zijn er drie knipper acties waarmee de gebruiker interacties zal kunnen aansturen.

```
const valueEyeBlinkLeft = blendShapes[0].categories
  .map((shape) =>
    | shape.categoryName === 'eyeBlinkLeft' ? shape.score : null,
  )
  .filter((shape) => shape !== null)[0];
const valueEyeBlinkRight = blendShapes[0].categories
  .map((shape) =>
    | shape.categoryName === 'eyeBlinkRight' ? shape.score : null,
  )
  .filter((shape) => shape !== null)[0];

// Special Event 1: if the user blinks with the left eye only
if (
  valueEyeBlinkLeft > 0.5 &&
  valueEyeBlinkRight < 0.5 &&
  OldValueEyeBlinkLeft < 0.5 &&
  OldValueEyeBlinkRight < 0.5
) { ...
}

// Special Event 2: if the user blinks with the right eye only
if (
  valueEyeBlinkLeft < 0.5 &&
  valueEyeBlinkRight > 0.5 &&
  OldValueEyeBlinkLeft < 0.5 &&
  OldValueEyeBlinkRight < 0.5
) { ...
}

// Special Event 3: if the user blinks with both eyes twice in a row (within 1 second)
if (
  valueEyeBlinkLeft > 0.5 &&
  valueEyeBlinkRight > 0.5 &&
  OldValueEyeBlinkLeft < 0.5 &&
  OldValueEyeBlinkRight < 0.5
) { ...
}

OldValueEyeBlinkLeft = valueEyeBlinkLeft;
OldValueEyeBlinkRight = valueEyeBlinkRight;
```

Figuur 21 - onderscheiden van knipper acties code

3.8 Chrome-extensie instellingen

Verschillende acties en interacties zijn gedefinieerd, maar tot nu toe zijn ze **nog niet aan elkaar gekoppeld** om de gewenste gedragingen aan te sturen. Om dit te bereiken, is besloten om instellingen toe te voegen aan de Chrome-extensie, waardoor gebruikers deze aan elkaar kunnen koppelen volgens hun persoonlijke voorkeuren.

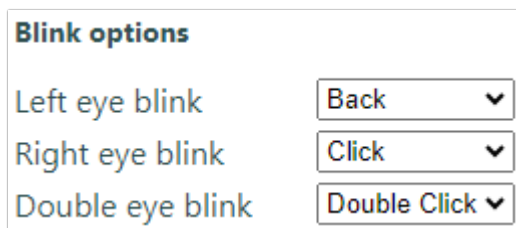
Bij het openen van de Chrome-extensie worden deze instellingen weergegeven, waarbij gebruikers via een **dropdownmenu** kunnen kiezen welke actie aan welke interactie moet worden gekoppeld. (Figuur 22)

Deze instellingen worden beheerd via de opslag van de Chrome-extensie zelf, die wordt aangeroepen via de **Chrome Extension Storage API**. Op deze manier zijn kunnen deze instellingen altijd opgehaald worden onafhankelijk van de website waarop de extensie gebruikt wordt. (figuur 23) (figuur 24)

In de code van de gebruikersacties wordt nu gecontroleerd **welke interactie** aan elke actie is gekoppeld, waardoor de juiste interactie kan worden uitgevoerd op basis van deze koppelingen. (figuur 25)

Ook is er nog een optie voorzien om de scrol snelheid aan te passen. De gebruiker kan via een schuiver bepalen of de scrol snelheid sneller of trager moet gebeuren. Ook deze instelling wordt in de Chrome-extensie opslag bijgehouden.

Dit zorgt voor een **flexibele** en **aanpasbare** ervaring, waarbij gebruikers de mogelijkheid hebben om de Chrome-extensie aan te passen aan hun **individuele behoeften** en **voorkeuren**.



| Blink options | |
|------------------|----------------|
| Left eye blink | Back ▼ |
| Right eye blink | Click ▼ |
| Double eye blink | Double Click ▼ |

Figuur 22 - instellingen om acties aan interacties te koppelen

```

// Get data from storage
chrome.storage.sync.get([request.key], function (result) {
  console.log('Value currently is ' + result[request.key]);
  console.log(result[request.key]);

  // Set data in localStorage from current active tab
  chrome.tabs.query({ active: true, currentWindow: true }, function (tabs) {
    if (tabs.length > 0) {
      const tabId = tabs[0].id;

      // Send a message to the content script to set data in localStorage
      chrome.scripting.executeScript({
        target: { tabId: tabId },
        function: (key, value) => {
          // Set data in localStorage
          console.log('key', key);
          console.log('value', value);
          localStorage.setItem(key, value);
        },
        args: [request.key, result[request.key]],
      });
    }
  });

  // Send response
  sendResponse({ message: result[request.key] });
});

```

Figuur 23 - ophalen van opgeslagen instellingen code

```

// Set data in storage
if (request.action === 'storeDataToChromeStorage') {
  // Log action
  console.log('storeDataToChromeStorage');

  // Save it using the Chrome extension storage API.
  chrome.storage.sync.set({ [request.key]: request.value }, function () {
    console.log('Value is set to ' + request.value);
  });

  // Set data in localStorage from current active tab
  chrome.tabs.query({ active: true, currentWindow: true }, function (tabs) {
    if (tabs.length > 0) {
      const tabId = tabs[0].id;

      // Send a message to the content script to set data in localStorage
      chrome.scripting.executeScript({
        target: { tabId: tabId },
        function: (key, value) => {
          // Set data in localStorage
          localStorage.setItem(key, value);
        },
        args: [request.key, request.value],
      });
    }
  });

  // Send response back to sender
  sendResponse({ message: 'Value is set to ' + request.value });
}

```

Figuur 24 - opslaan van instellingen code

```

// Special Event 1: if the user blinks with the left eye only
if (
  valueEyeBlinkLeft > 0.5 &&
  valueEyeBlinkRight < 0.5 &&
  OldValueEyeBlinkLeft < 0.5 &&
  OldValueEyeBlinkRight < 0.5
) {
  console.log('Special Event 1: if the user blinks with the left eye only');
  GazePointer.style.backgroundColor = 'red';
  // Get leftEyeBlinkAction from local storage
  const leftEyeBlinkAction = localStorage.getItem('leftEyeBlinkAction');
  if (leftEyeBlinkAction == 'none') {
    console.log('No action set for left eye blink');
  } else if (leftEyeBlinkAction == 'click') {
    EmulateClick();
  } else if (leftEyeBlinkAction == 'doubleClick') {
    EmulateDoubleClick();
  } else if (leftEyeBlinkAction == 'back') {
    window.history.back();
  } else {
    console.log('Invalid action set for left eye blink');
  }
}
}

```

Figuur 25 - acties de gekoppelde interactie laten uitvoeren

4 Reflectie

In dit hoofdstuk wordt er gereflecteerd op de resultaten van het technische onderzoek naar de vraag: "Kunnen oogbewegingen en knipperen via een Chrome-extensie vertaald worden naar website-interacties?" Dit wordt gedaan in combinatie met feedback van zowel Arthur d'Hooge, een AI- en backend-ontwikkelaar bij Hitlab, een bedrijf dat onderzoek doet naar eye-tracking, als Mathieu Braekeveld, medeoprichter en projectmanager bij JobFunders, een opkomend bedrijf dat AI-technologie inzet om carrières te bevorderen.

4.1 Resultaten technisch onderzoek

4.1.1 Sterke punten

Een sterk punt van dit project is de hoge mate van toegankelijkheid voor gebruikers. Het enige dat een gebruiker hoeft te doen om aan de slag te gaan, is het toevoegen van de Chrome-extensie. Zodra dit is gebeurd, kunnen websites worden aangestuurd via webcam-gebaseerde eye-tracking. De ingebouwde camera van een laptop is voldoende om de oogbewegingen te registreren. Er is geen extra hardware, zoals een eye-tracking camera of bril, nodig. Dit zorgt ervoor dat de extensie volledig kosteloos kan worden gebruikt, waardoor het een zeer toegankelijke en betaalbare oplossing is voor een breed scala aan gebruikers. Zowel Arthur als Mathieu vinden dit een zeer belangrijk en sterk punt van de de extensie.

Een ander sterk punt van dit project is dat het een Chrome-extensie betreft, en niet een specifieke website die aangestuurd kan worden door eye-tracking. Hierdoor kunnen gebruikers elke willekeurige website bedienen met behulp van eye-tracking, zonder beperkingen tot één platform of applicatie. Deze flexibiliteit maakt de extensie veelzijdig inzetbaar en toepasbaar op een breed scala aan websites, wat de bruikbaarheid en waarde van de tool aanzienlijk vergroot. Dit punt wordt opnieuw benadrukt door zowel Arthur als Mathieu.

Een bijkomend voordeel van dit project is dat de kalibratie van de eye-tracking volledig via de webcam gebeurt. Het kalibratieproces vereist enkel het volgen van een aantal bolletjes die heen en weer bewegen op het scherm, zonder dat er interactie met een muis of ander invoerapparaat nodig is. Dit handsfree kalibratieproces verhoogt de toegankelijkheid en gebruiksvriendelijkheid van de extensie, waardoor gebruikers zonder aanvullende hulpmiddelen of ingewikkelde handelingen eenvoudig kunnen beginnen met het gebruiken van de eye-tracking functionaliteiten.

Een ander sterk punt van dit project is de mogelijkheid voor gebruikers om de extensie naar hun eigen voorkeuren in te stellen. Via de instellingen kan de gebruiker onder andere de scrol snelheid aanpassen en bepalen welke acties aan welke interacties worden gekoppeld. Deze aanpasbaarheid geeft de gebruiker meer controle over de ervaring, waardoor de extensie beter kan worden afgestemd op individuele behoeften en voorkeuren. Dit draagt bij aan een gebruiksvriendelijkere en gepersonaliseerde interactie met verschillende websites.

4.1.2 Zwakke punten

Een tekortkoming in de extensie volgens Arthur is het ontbreken van een manier om tekst te kunnen typen op een handsfree manier. Arthur heeft voorgesteld om dit probleem aan te pakken door gebruik te maken van eye-tracking of eventueel spraaktechnologie. Met deze extra toevoeging zou de extensie een volledig afgewerkt product worden, waardoor gebruikers niet alleen websites kunnen navigeren en bedienen, maar ook tekst kunnen invoeren zonder fysieke interactie. Dit zou de toegankelijkheid en bruikbaarheid van de extensie verder vergroten, met name voor gebruikers met mobiliteitsbeperkingen of andere beperkingen die het gebruik van traditionele invoerapparaten bemoeilijken.

Mathieu merkte op dat er meer kan worden ingezet op de gebruiksvriendelijkheid door betere instructies te geven aan de gebruiker over wat ze moeten doen en wat de extensie kan. Het aanbieden van een duidelijke introductie over de verschillende mogelijkheden, zoals scrollen, klikken en andere interacties, zou de gebruikerservaring aanzienlijk verbeteren. Door gebruikers stap-voor-stap te begeleiden en hen te informeren over de functionaliteiten van de extensie, kunnen ze sneller en effectiever leren hoe ze deze moeten gebruiken. Dit draagt bij aan een soepelere en meer intuïtieve interactie met de extensie, wat de algehele tevredenheid en efficiëntie verhoogt.

4.2 Bruikbaarheid projectresultaat

Het was even afwachten of deze Chrome-extensie een realistische manier zou zijn om websites te bedienen of dat het meer een gimmick zou zijn. Zowel Arthur als Mathieu zijn ervan overtuigd dat dit project zeker meer is dan een gimmick en dat deze extensie aanzienlijke potentie heeft. Er zouden wel nog een paar kleine aanpassingen moeten worden gemaakt om de extensie volledig af te maken en optimaal bruikbaar te maken in diverse situaties.

Voor Arthur zou deze extensie bijzonder nuttig kunnen zijn als er een manier wordt toegevoegd om tekst te kunnen typen via eye-tracking of via stemherkenning. Dit zou de functionaliteit verder uitbreiden en de extensie toepasbaar maken in nog meer scenario's. Typen via eye-tracking zou bijvoorbeeld handig zijn voor mensen met mobiliteitsbeperkingen, terwijl stemherkenning een snelle en efficiënte manier biedt om tekstinvoer te vergemakkelijken. Arthur benadrukt dat deze toevoegingen de toegankelijkheid van de extensie zouden vergroten en het bruikbaar zouden maken voor een bredere doelgroep.

Mathieu denkt dat de extensie nu al effectief kan worden gebruikt om eenvoudig op het internet te surfen, zoals voor het lezen van het nieuws. Hij ziet de extensie als een waardevol hulpmiddel voor mensen die regelmatig hun handen vol hebben of voor wie het gebruik van een muis en toetsenbord niet praktisch is. Bijvoorbeeld, in situaties waar gebruikers snel door webpagina's moeten navigeren zonder hun activiteiten te onderbreken, kan de extensie een aanzienlijke verbetering van de efficiëntie bieden.

Daarnaast stelde Arthur voor om gebruik te maken van interpolatie bij het uitmappen van de datapunten waar de gebruiker naar kijkt. Dit is een techniek om datapunten te voorspelen op basis van bestaande data. Dit is een bekende techniek binnen de eye-tracking. Dit zou leiden tot een vloeiendere gebruikerservaring doordat de tracking nauwkeuriger en consistentere wordt. Interpolatie kan helpen om kleine onnauwkeurigheden in de eye-tracking data te corrigeren, wat resulteert in een stabielere en betrouwbaardere cursorbeweging op het scherm. Hierdoor wordt de interactie met websites intuïtiever en minder vermoeiend voor de gebruiker.

Arthur adviseerde ook om de knippertijd van de gebruiker als trigger voor acties te gebruiken in plaats van enkel het knippen zelf. Dit zou de acties iets vertragen, maar voorkomt onbedoelde activeringen door onwillekeurig knippen. Arthur stelde voor om hiervoor 1 seconde te voorzien. Door een minimale tijdsduur voor het sluiten van de ogen te vereisen, kunnen gebruikers meer controle uitoefenen over de triggers, wat leidt tot een nauwkeurigere en betrouwbaardere activering van functies. Deze aanpassing zou vooral nuttig zijn in situaties waar precisie cruciaal is, zoals bij het selecteren van kleine elementen op een webpagina of het uitvoeren van complexe opdrachten.

Samengevat, hoewel er nog enkele verbeteringen nodig zijn, zijn zowel Arthur als Mathieu het erover eens dat deze extensie een veelbelovende en praktische oplossing biedt voor handsfree navigatie op het web. De voorgestelde verbeteringen, zoals tekstinvoer via eye-tracking of stemherkenning, interpolatie voor nauwkeurigere tracking, en verfijnde knipperdetectie, kunnen de extensie transformeren van een innovatief concept naar een essentieel hulpmiddel voor dagelijkse internetgebruikers en professionals in diverse werkomgevingen.

4.3 Overweging implementatie bedrijfscontext

Zowel Arthur als Mathieu zijn overtuigd dat deze extensie zowel voor privégebruik als in een bedrijfsomgeving kan worden ingezet. Arthur ziet de extensie nuttig in situaties waarin werknemers hun handen niet vrij hebben om extra taken uit te voeren. Bijvoorbeeld, in de context van bandwerk kunnen medewerkers instructies op een scherm lezen zonder hun handen te gebruiken. Hij benadrukt dat de handsfree bediening een significant voordeel kan bieden in productielijnen, waar efficiëntie en snelheid cruciaal zijn. Door de extensie te gebruiken, kunnen werknemers hun taken uitvoeren zonder te hoeven stoppen om een muis of toetsenbord te gebruiken, wat de workflow en productiviteit kan verbeteren.

Mathieu deelt een vergelijkbare visie en ziet toepassingen voor de extensie in situaties zoals kassawerk of order picking, waar medewerkers hun handen bezet hebben. Hij wijst erop dat kassamedewerkers vaak constant in beweging zijn en multitasken, wat betekent dat een handsfree oplossing zoals deze extensie hun werk aanzienlijk kan vergemakkelijken. Daarnaast benadrukt hij dat de extensie ook bruikbaar is in omgevingen waar simpelweg geen ruimte is voor een toetsenbord en muis, zoals bij het bedienen van schermen die aan een muur hangen. In magazijnen of distributiecentra waar order pickers constant bezig zijn met het verplaatsen van producten, kan de extensie hen helpen sneller te reageren op veranderingen in bestellingen of routes zonder hun huidige activiteit te onderbreken.

Deze flexibiliteit maakt de extensie waardevol in diverse werkomgevingen waar handenvrije bediening essentieel is. Het vermogen om snel en efficiënt te reageren op visuele informatie zonder fysieke inputapparaten te gebruiken, kan leiden tot een meer gestroomlijnde en effectieve werkomgeving. Bovendien biedt de extensie de mogelijkheid om taken uit te voeren in situaties waar traditionele invoerapparaten onpraktisch of onbruikbaar zijn, wat de toegankelijkheid en bruikbaarheid in uiteenlopende bedrijfsscenario's vergroot. Arthur en Mathieu zijn het erover eens dat deze technologie een innovatieve oplossing biedt voor moderne werkplekken, waar handsfree interactie steeds relevanter wordt.

4.4 Alternatieven

Een alternatieve manier om de eye-tracking van deze extensie te verbeteren, is door gebruik te maken van specifieke eye-tracking camera's of zelfs eye-tracking brillen. De prijs van zo een camera loopt tussen de 100 en 1000 euro en kan zelfs oplopen tot 10.000 euro als het over een echte professionele bril gaat. Een bekende maker van deze producten is Tobii. De Tobii eye tracker 5 is een voorbeeld van zo een camera en verkoopt rond de 300 euro. Dit zou de gebruikerservaring aanzienlijk kunnen veranderen. Het voordeel hiervan is dat de resultaten nauwkeuriger zouden zijn. In het geval van een eye-tracking bril hoeft de gebruiker niet in een vaste, goed belichte positie voor een webcam te zitten, maar kan hij of zij vrij rond bewegen zonder aan nauwkeurigheid in te boeten. Mathieu wees erop dat sommige gebruikers tijdens hun werk niet recht voor een webcam kunnen zitten, wat de inzet van een eye-tracking bril aantrekkelijk maakt. [20]

Arthur is echter van mening dat het toevoegen van een hardware-optie om de eye-tracking te verbeteren de extensie niet ten goede zou komen. Hoewel dit inderdaad zou leiden tot betere resultaten, acht hij de kosten niet gerechtvaardigd. Eye-tracking brillen zijn nog steeds erg duur, wat de gebruiksvriendelijkheid zou schaden in plaats van de kwaliteit significant te verhogen. Hij benadrukt dat de huidige webcam-gebaseerde oplossing al een goede balans biedt tussen kosten en prestaties, waardoor het project toegankelijk en betaalbaar blijft voor een breed scala aan gebruikers.

4.5 Economische en maatschappelijke meerwaarde

Dit project biedt aanzienlijke economische en maatschappelijke voordelen. Door de integratie van eye-tracking technologie in een Chrome-extensie, kunnen zowel bedrijven als individuen profiteren van verbeterde efficiëntie en toegankelijkheid.

4.5.1 Economische meerwaarde

De economische voordelen van deze extensie zijn duidelijk, vooral in bedrijfsomgevingen waar tijdsbesparing cruciaal is. In veel bedrijven voeren werknemers repetitieve taken uit waarvoor handen vrij moeten zijn, zoals bandwerk, kassawerk of order picking. Door gebruik te maken van eye-tracking technologie, kunnen deze werknemers instructies en informatie op een scherm lezen zonder hun handen te hoeven gebruiken. Dit kan de workflow aanzienlijk verbeteren en de productiviteit verhogen, wat uiteindelijk leidt tot kostenbesparingen.

4.5.2 Maatschappelijke meerwaarde

Naast de economische voordelen biedt deze extensie ook belangrijke maatschappelijke meerwaarde, met name voor mensen met fysieke beperkingen. Voor veel mensen met mobiliteitsbeperkingen kan het gebruik van een muis en toetsenbord een uitdaging zijn. Door het aanbieden van een handsfree manier om websites te navigeren, opent deze extensie nieuwe mogelijkheden voor deze doelgroep om het internet te verkennen en te gebruiken.

Het internet is een essentiële bron van informatie, communicatie en entertainment. Voor mensen met fysieke beperkingen kan toegang tot deze bron via conventionele methoden beperkt of onmogelijk zijn. Met de implementatie van eye-tracking technologie, kunnen deze gebruikers nu zelfstandig surfen, lezen en interacteren met webinhoud zonder afhankelijk te zijn van anderen. Dit bevordert hun zelfstandigheid en integratie in de digitale wereld.

Bijvoorbeeld, iemand met een ernstige motorische beperking kan nu eenvoudig door nieuwsartikelen bladeren, online aankopen doen, of deelnemen aan sociale media interacties, allemaal door oogbewegingen te gebruiken. Dit vergroot hun autonomie en verbetert hun kwaliteit van leven aanzienlijk.

4.5.3 Conclusie

De economische en maatschappelijke voordelen van deze Chrome-extensie zijn duidelijk. Door tijd te besparen en de efficiëntie te verhogen, kan de extensie bedrijven helpen hun operationele kosten te verlagen. Tegelijkertijd biedt het een inclusieve oplossing voor mensen met fysieke beperkingen, waardoor zij op een gelijkwaardige manier kunnen deelnemen aan de digitale wereld. Deze combinatie van voordelen onderstreept de waarde en potentie van het project, en toont aan dat het meer is dan een technologische gimmick - het is een praktische, waardevolle tool met brede toepassingen.

4.6 Suggesties vervolgonderzoek

Er zijn verschillende interessante onderzoekrichtingen die de functionaliteit en toegankelijkheid van deze Chrome-extensie verder zouden kunnen verbeteren. Hier volgen enkele potentiële gebieden die nader onderzoek verdienen.

4.6.1 Handsfree typen

Een belangrijke onderzoekrichting zou kunnen zijn hoe gebruikers het meest effectief handsfree kunnen typen. Dit kan door het integreren van verschillende technologieën zoals eye-tracking en spraakherkenning. Het onderzoeken van de beste methoden voor handsfree typen kan leiden tot verbeterde functionaliteiten binnen de extensie, waardoor gebruikers niet alleen kunnen navigeren, maar ook teksten kunnen invoeren zonder hun handen te gebruiken. Dit is bijzonder nuttig voor mensen met motorische beperkingen en zou de algehele bruikbaarheid van de extensie aanzienlijk vergroten. Door het combineren van spraakcommando's met nauwkeurige oogbewegingen, kan een efficiënte en intuïtieve handsfree typemethode worden ontwikkeld.

4.6.2 Firefox extensie ontwikkelen

Een andere waardevolle onderzoekrichting zou kunnen zijn het aanpassen van de Chrome-extensie om als Firefox-extensie te dienen. Niet alle gebruikers maken gebruik van de Chrome-browser; daarom zou het uitbreiden van de compatibiliteit naar Firefox een bredere gebruikersbasis bereiken.

4.6.3 Verbetering van Eye-tracking met Interpolatie

Arthur heeft gesuggereerd om te onderzoeken of het gebruik van interpolatie voor de eye-tracking data de gebruiksvriendelijkheid zou kunnen verbeteren. Interpolatie kan helpen om de tracking van oogbewegingen vloeiender en nauwkeuriger te maken. Door de punten waar de gebruiker naar kijkt te interpoleren, kunnen kleine schommelingen in de data worden gladgestreken, wat resulteert in een stabielere en betrouwbaardere gebruikerservaring. Dit onderzoek zou zich richten op het ontwikkelen en testen van verschillende interpolatie-algoritmen om de meest effectieve methode te vinden voor het verbeteren van de eye-tracking prestaties. Verbeterde nauwkeurigheid en soepelheid in eye-tracking kunnen de algehele ervaring van de gebruiker aanzienlijk verbeteren, vooral tijdens langdurig gebruik.

4.6.4 Conclusie

Deze mogelijke onderzoekrichtingen bieden veel potentieel om de functionaliteit en toegankelijkheid van de eye-tracking extensie verder te verbeteren. Door handsfree typen te optimaliseren, de extensie compatibel te maken met Firefox, en de nauwkeurigheid van eye-tracking data te verbeteren door middel van interpolatie, kan de extensie worden verfijnd tot een nog krachtiger en inclusiever hulpmiddel voor gebruikers. Verdere studies in deze richtingen zullen niet alleen de technologie verbeteren maar ook de gebruikerservaring verrijken, waardoor de extensie een nog waardevoller instrument wordt voor een breed scala aan toepassingen en gebruikersgroepen.

5 Advies

In dit hoofdstuk worden concrete aanbevelingen voor het werkveld gepresenteerd op basis van het onderzoek. De nadruk ligt op de bruikbaarheid en toepasbaarheid van de voorgestelde oplossingen. Adviezen uit het werkveld zijn besproken en geïntegreerd in de aanbevelingen. Er wordt een gedetailleerd stappenplan geboden om de onderzoeksresultaten effectief te implementeren. Tot slot worden de speciaal ontwikkelde tools voor deze implementatie geïntroduceerd.

5.1 Creëren van een chrome-extensie

Bij het maken van een chrome-extensie is het belangrijk dat je de goed overweg kan met html, css en javascript. Dit zijn de basis onderdelen om aan een website te werken. Vandaar dat deze ook aanbod komen bij het maken van een chrome-extensie.

Het is belangrijk om te weten waarop je wilt focussen. Wil je een extensie creëren die werkt voor alle websites of eerder specifiek voor één bepaalde website. Beide opties zijn mogelijk maar eisen een andere aanpak. Bij het focussen op één vaste website kan je specifieke elementen van die website aanspreken om bepaalde aanpassingen of acties uit te voeren. Als je van plan bent om een extensie te ontwikkelen voor alle websites kan je niet zomaar specifieke element aansturen van een website. In dit geval kan je beter zelf elementen toevoegen aan de website en die dan aanspreken.

Tenslotte is het ook belangrijk om indien mogelijk met de laats mogelijke manifest versie te werken. Dit zorgt ervoor dat de extensie zo lang mogelijk ondersteund zal worden.

5.2 Bruikbaarheid en toepasbaarheid

Met de constante evolutie van AI-technologieën worden webcam-gebaseerde eye-tracking opties steeds beter. Ook Apple heeft aangekondigd dat ze bezig zijn met nieuwe toegankelijkheidsfuncties, waaronder eye-tracking . [19] Deze ontwikkelingen zorgen ervoor dat applicaties zoals deze Chrome-extensie steeds bruikbaar en toegankelijker zullen worden, aangezien de technologie steeds verder verbetert en mensen steeds meer vertrouwd raken met het concept van eye-tracking.

Toepassingen in de privésector

In de privésector kan deze Chrome-extensie een bijzondere rol spelen in het verbeteren van de levenskwaliteit van mensen met fysieke beperkingen. Door gebruik te maken van eye-tracking technologie kunnen deze gebruikers zelfstandig op het internet surfen zonder fysieke interactie met een toetsenbord of muis. Dit biedt hen een nieuwe vorm van onafhankelijkheid en vergroot hun toegang tot digitale informatie en diensten. De extensie stelt hen in staat om taken zoals het lezen van nieuws, online shoppen en sociale media te gebruiken zonder de noodzaak van handmatige input, wat een enorme stap vooruit is in termen van digitale inclusie en gelijkheid.

Toepassingen in de bedrijfswereld

In de bedrijfswereld biedt deze extensie ook aanzienlijke voordelen. Ze kan vooral nuttig zijn in situaties waarin werknemers hun handen niet vrij hebben om extra taken uit te voeren. Bijvoorbeeld, in productieomgevingen kunnen werknemers instructies op een scherm lezen zonder hun werk te onderbreken. In logistieke sectoren zoals orderpicking en kassawerk, waar werknemers constant hun handen gebruiken voor andere taken, kan eye-tracking technologie hen in staat stellen om gemakkelijk informatie te bekijken en eenvoudige handelingen uit te voeren zonder fysieke interactie met een computer.

Belangrijke overwegingen

Hoewel de toekomstperspectieven veelbelovend zijn, is het belangrijk om ook enkele tekortkomingen te erkennen. Een significante beperking is dat gebruikers een webcam nodig hebben en recht voor deze moeten zitten om gebruik te kunnen maken van de extensie. Dit kan ongemakkelijk zijn voor sommige gebruikers en beperkt de flexibiliteit van waar en hoe de extensie gebruikt kan worden.

Bovendien zijn er nog veel mensen die niet bekend zijn met Chrome-extensies, wat een drempel kan vormen voor het gebruik van deze technologie. Voor deze gebruikers kan de installatie en het gebruik van de extensie verwarrend zijn, wat de adoptie van deze technologie kan belemmeren. Het verstrekken van duidelijke instructies en gebruiksvriendelijke handleidingen kan helpen om deze barrière te verminderen, maar het blijft een uitdaging die moet worden aangepakt.

5.3 Concrete aanbevelingen

5.3.1 Toepassingsbereik

De keuze om deze applicatie als een Chrome-extensie te ontwikkelen, is een strategische beslissing die de toegankelijkheid en veelzijdigheid van de tool aanzienlijk vergroot. Aangezien de extensie in de browser werkt, is deze geschikt voor alle websites die via Chrome worden bezocht. Dit maakt de tool breed inzetbaar en nuttig in een groot aantal contexten zonder de noodzaak van aanpassingen voor specifieke websites.

5.3.2 Toegankelijkheid

Een ander sterk punt van deze applicatie is de toegankelijkheid. Door enkel gebruik te maken van de ingebouwde camera van een laptop of een eenvoudige webcam, wordt de drempel voor gebruik verlaagd. Gebruikers hoeven geen extra hardware aan te schaffen, wat de kosten laag houdt en het gebruiksgemak verhoogt. Dit maakt de extensie aantrekkelijk voor een breed publiek, waaronder mensen met fysieke beperkingen die anders geen toegang zouden hebben tot dergelijke technologieën. Arthur en Mathieu vinden de toegankelijkheid een zeer belangrijk en goed onderdeel van dit project.

5.3.3 Gebruikersgericht Ontwerp

Bij het ontwikkelen van deze extensie is het belangrijk om een balans te vinden tussen gebruiksvriendelijkheid en toegankelijkheid enerzijds, en kwaliteit en nauwkeurigheid anderzijds. Hieronder volgen enkele aanbevelingen voor beide benaderingen:

Gebruiksvriendelijkheid en toegankelijkheid

Eenvoudige installatie en configuratie: Zorg ervoor dat de installatie van de extensie en de initiële configuratie intuïtief en eenvoudig zijn. Dit kan worden bereikt door een duidelijke, stapsgewijze handleiding en een gebruiksvriendelijke interface.

Handsfree kalibratie: Behoud de huidige handsfree kalibratiemethode waarbij gebruikers hun ogen slechts hoeven te volgen. Dit verhoogt de toegankelijkheid en maakt het mogelijk voor gebruikers om snel en zonder extra hulpmiddelen aan de slag te gaan.

Flexibele instellingen: Bied gebruikers de mogelijkheid om de instellingen van de extensie aan te passen aan hun persoonlijke voorkeuren, zoals de snelheid van scrollen en de toewijzing van acties aan knipperbewegingen. Dit verhoogt de gebruiksvriendelijkheid en maakt de tool aanpasbaar aan verschillende behoeften.

Kwaliteit en nauwkeurigheid

Geavanceerde eye-tracking technieken: Overweeg het implementeren van geavanceerde eye-tracking technieken zoals interpolatie om de nauwkeurigheid van de tracking te verbeteren. Dit kan leiden tot een soepelere en betrouwbaardere gebruikerservaring. Dit is aangeraden door Arthur.

Duidelijke feedback mechanismen: Voorzie duidelijke visuele feedback over waar de gebruiker naar kijkt en welke acties worden uitgevoerd. Dit helpt fouten te minimaliseren en verhoogt de betrouwbaarheid van de interacties.

Verbeterde Knipperdetectie: Integreer geavanceerde methoden voor het detecteren van knipperbewegingen, zoals het meten van de duur van het sluiten van de ogen, om onbedoelde activeringen te voorkomen en de precisie van de acties te verbeteren.

5.4 Stappenplan

Het ontwikkelen van een Chrome-extensie om websites te besturen met behulp van eye-tracking vereist een zorgvuldig begin, aangezien dit de basis vormt voor het uiteindelijke resultaat van de extensie. Hieronder worden de stappen beschreven die je kunt volgen om dit proces effectief aan te pakken. (Figuur 26)



Figuur 26 - schema overzicht van de verschillende stappen

5.4.1 Stap 1: Onderzoeken

Het eerste wat je moet doen is het bepalen van je doelpubliek. Dit kan variëren van mensen met fysieke beperkingen tot werknemers in specifieke bedrijfsomgevingen zoals kassa's of productielijnen. Afhankelijk van dit publiek, zoek je naar de meest geschikte eye-tracking software. Deze software moet nauwkeurig en betrouwbaar zijn, en gemakkelijk te integreren met een Chrome-extensie. Daarnaast moet je ook een manier vinden om gezichtsbewegingen te herkennen en deze gegevens te verwerken.

5.4.2 Stap 2: Uittesten

Maak een eenvoudige test applicatie om de gekozen eye-tracking software uit te proberen. Dit helpt om te bepalen of de software precies doet wat je zoekt en hoe het werkt. Gebruik deze test applicatie om de nauwkeurigheid en gebruiksvriendelijkheid van de eye-tracking functies te evalueren. Test ook de mogelijkheid om gezichtsbewegingen te herkennen en hoe deze gegevens kunnen worden geïntegreerd in een functionele applicatie.

5.4.3 Stap 3: Ontwikkelen

Nu je een goed begrip hebt van de eye-tracking software, kun je beginnen met het ontwikkelen van de Chrome-extensie. Bepaal de algemene werking en functies die je met de extensie wilt bereiken. Ontwerp de gebruikersinterface en zorg ervoor dat de extensie eenvoudig te gebruiken is. Definieer ook de interacties die gebruikers kunnen hebben met de extensie, zoals het navigeren door websites en het uitvoeren van acties met behulp van eye-tracking.

5.4.4 Stap 4: Integreren

Voeg de reeds geteste eye-tracking software toe aan de Chrome-extensie. Zorg ervoor dat de integratie soepel verloopt en dat de eye-tracking functies naadloos werken binnen de extensie. Test de extensie uitgebreid om te verifiëren dat alle functies correct werken en dat de gebruikerservaring optimaal is. Pas waar nodig de code aan om de prestaties te verbeteren en eventuele bugs op te lossen.

5.4.5 Stap 5: Afwerken

Pas de toegevoegde software aan om deze precies te laten werken zoals je wilt en om ervoor te zorgen dat deze goed samenwerkt met andere delen van de extensie. Dit kan inhouden dat je bepaalde parameters wijzigt, nieuwe functies toevoegt of bestaande functies verfijnt. Zorg ervoor dat de eye-tracking functionaliteiten nauwkeurig blijven werken en dat de extensie blijft voldoen aan de behoeften van het doelpubliek. Voer grondige tests uit om te controleren of alle onderdelen van de extensie goed samenwerken en of de prestaties optimaal zijn.

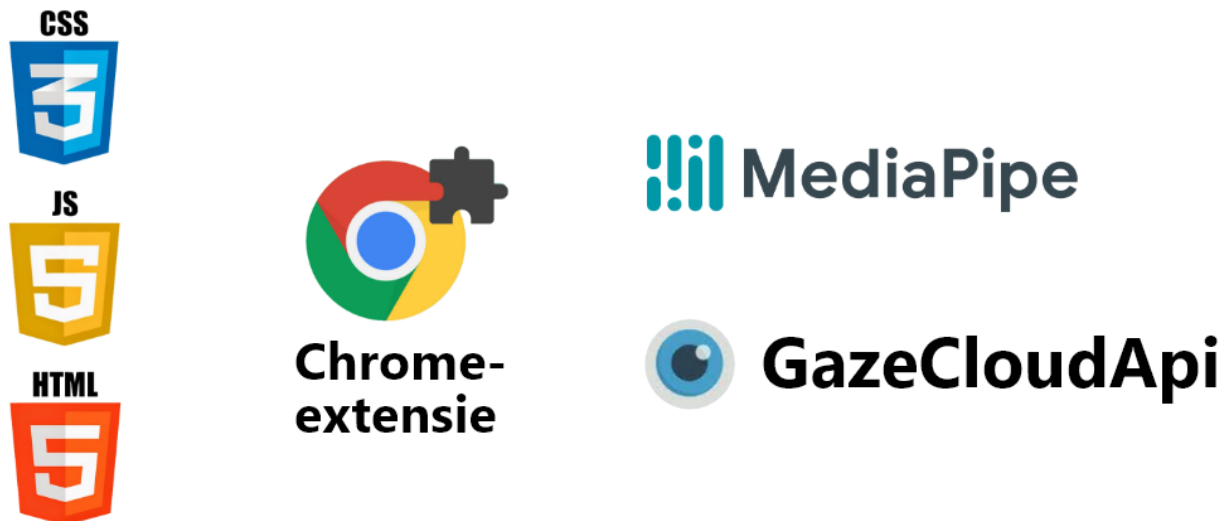
Door dit stappenplan te volgen, kun je een effectieve en gebruiksvriendelijke Chrome-extensie ontwikkelen met geavanceerde eye-tracking functionaliteiten, geschikt voor zowel privé- als zakelijke toepassingen.

6 Conclusie

In dit hoofdstuk wordt teruggeblikt op alle voorgaande hoofdstukken en wordt de onderzoeksvraag “Kunnen oogbewegingen en knipperen via een Chrome-extensie vertaald worden naar website-interacties?” beantwoord.

6.1 Gebruikte technologieën

De gebruikte technologieën hiervoor zijn als volgt:



Figuur 27 - Overzicht van de gebruikte technologieën

De ontwikkeling van een **Chrome-extensie** zorgt ervoor dat de tool toegankelijk is voor een breed scala aan websites zonder beperkingen tot specifieke platformen of applicaties. Dit maakt de extensie veelzijdig inzetbaar en toepasbaar op elke website die via Chrome wordt bezocht, wat de bruikbaarheid en waarde van de tool aanzienlijk vergroot.

De keuze om **Face Landmark Detection** van MediaPipe voor gezichtskenmerken detectie en oogbeweging tracking in deze bachelorproef te gebruiken, is gebaseerd op verschillende belangrijke aspecten. Ten eerste is het een JavaScript-bibliotheek die eenvoudig aan een website kan worden toegevoegd. Dit maakt de integratie naadloos en vermindert de ontwikkeltijd. Daarnaast is MediaPipe aanzienlijk sneller dan andere opties zoals PyGaze. Er waren ook verschillende codevoorbeelden beschikbaar die aantonen hoe deze JavaScript-bibliotheek gebruikt kan worden, wat de leercurve verkort. Bovendien biedt MediaPipe de flexibiliteit om te kiezen tussen het gebruik van de CPU of GPU, wat belangrijk is om de prestaties te optimaliseren.

GazeCloudApi is gekozen als eye-tracking software omdat het een JavaScript-bibliotheek is die gemakkelijk kan worden geïntegreerd in een website. Deze keuze werd ook beïnvloed door de snelheid, aangezien GazeCloudApi sneller is dan vergelijkbare opties zoals WebGazer. Een andere belangrijke factor was de handsfree kalibratie mogelijkheid van GazeCloudApi, die een aanzienlijk gebruiksgemak biedt ten opzichte van andere oplossingen..

6.2 Eindresultaat

Dit project heeft geresulteerd in een Chrome-extensie die bij het openen twee belangrijke JavaScript-bibliotheken injecteert in de actieve Chrome-pagina. De implementatie van de Face Landmark Detection en GazeCloudApi bibliotheken stelt de extensie in staat om de subtiele oogbewegingen van de gebruiker te detecteren via de webcam. Hiermee wordt een nieuwe dimensie van interactie met het web mogelijk gemaakt.

Zodra deze bibliotheken zijn geïnjecteerd, wordt een informatief scherm weergegeven op de website, waarop het camerabeeld van de gebruiker wordt getoond samen met stapsgewijze instructies om de beste positie voor de camera te vinden. Deze eenvoudige setup zorgt ervoor dat de extensie gemakkelijk toegankelijk is voor gebruikers van verschillende niveaus van technische bekwaamheid.

Na het volgen van deze eenvoudige instructies kan de gebruiker op de knop "Start Gaze Calibration" klikken, waarna een kalibratietest wordt gestart. Tijdens deze test moet de gebruiker een rode bol op het scherm volgen terwijl deze beweegt, wat de nauwkeurigheid van de eye-tracking bijwerkt op basis van de individuele oogbewegingen van de gebruiker. Deze kalibratietest draagt bij aan een gepersonaliseerde gebruikerservaring en verbetert de nauwkeurigheid van de interacties met de website.

Zodra de kalibratie is voltooid, keert de gebruiker terug naar de originele websitepagina. Hier wordt de locatie van de blik aangegeven door een bolletje dat de bewegingen van de ogen van de gebruiker volgt. Bovendien worden de specifieke elementen op de website waar de gebruiker naar kijkt, gemarkeerd met een rode rand, waardoor de gebruiker zich bewust is van de focuspunten op de pagina.

Met de gratis versie van GazeCloudApi kan de gebruiker gedurende de komende twee minuten vrij navigeren op de website. De intuïtieve interface van de extensie biedt gebruikers de mogelijkheid om handsfree te scrollen door simpelweg naar de boven- of onderkant van de pagina te kijken. Bovendien biedt de extensie een pop-up venster waarin de gebruiker de website-interacties kan aanpassen aan knipperbewegingen. Dit geeft gebruikers de flexibiliteit om de interacties aan te passen aan hun specifieke behoeften en voorkeuren.

Samengevat biedt deze Chrome-extensie een geavanceerde en gebruiksvriendelijke manier voor gebruikers om websitefuncties te bedienen via eye-tracking. De combinatie van kalibratie, real-time blikdetectie en aanpasbare interactiemogelijkheden maakt het mogelijk om op een natuurlijke en intuïtieve manier met het web te communiceren, waardoor de toegankelijkheid en het gebruiksgemak worden vergroot.

6.3 Onderzoeksvraag

In deze bachelorproef is er onderzoek gedaan naar de vraag: “Kunnen oogbewegingen en knippen via een Chrome-extensie vertaald worden naar website-interacties?” Het antwoord op deze vraag is ja. Doormiddel van gebruik te maken van de webcam en software om oogbewegingen en knippen uit te lezen kan de Chrome-extensie website interacties uitvoeren net zoals een gebruiker deze zelf zou uitvoeren.

6.4 Suggesties vervolg onderzoek

Een mogelijke richting voor vervolgonderzoek is het verkennen van geavanceerdere eye-tracking hardware in plaats van de standaard webcam die momenteel wordt gebruikt in de Chrome-extensie. Door gebruik te maken van gespecialiseerde eye-tracking hardware kan de nauwkeurigheid en responsiviteit van de eye-tracking-functionaliteit worden verbeterd. Deze hardware kan speciaal ontworpen zijn voor het volgen van oogbewegingen en kan geavanceerde technologieën bevatten die meer gedetailleerde en precieze gegevens kunnen leveren dan een standaard webcam. Mogelijkheden hiervoor zijn scherm gebaseerde eye-tracking camera's die je aan je scherm kan vastmaken of brillen met ingebouwde eye-tracking camera's om in elke richting te kunnen kijken. Tobii is een bekend voorbeeld van dit soort producten. [9]

Een ander interessant aspect om te onderzoeken is de mogelijkheid om handsfree tekstinvoer mogelijk te maken met behulp van eye-tracking. Dit zou gebruikers in staat stellen om tekst op een natuurlijke en intuïtieve manier in te voeren, simpelweg door naar de letters op een virtueel toetsenbord te kijken. Door het combineren van geavanceerde eye-tracking technologie met slimme algoritmes voor tekstvoorspelling en autocorrectie, kan handsfree tekstinvoer een revolutionaire functie worden die de toegankelijkheid van computersystemen aanzienlijk vergroot, vooral voor mensen met mobiliteitsbeperkingen. Gazeboard is een voorbeeld van zo een eye-tracking tool om tekst in te kunnen geven. GazeBoard toont alle letters op het scherm en zoemt langzaam aan in op de letters waar je naar aan het kijken bent, om zo te bepalen welke letters je wilt ingeven. [21]

Het onderzoeken van deze mogelijkheden vereist een diepgaande analyse van de beschikbare eye-tracking hardware op de markt, evenals het ontwikkelen van nieuwe software-algoritmes en gebruikersinterfaces om handsfree tekstinvoer te ondersteunen. Daarnaast is het belangrijk om gebruikerstests uit te voeren om de bruikbaarheid en effectiviteit van deze nieuwe functies te evalueren en te verfijnen.

Door verder te bouwen op het huidige onderzoek en te investeren in nieuwe technologieën en toepassingen, kunnen we nieuwe manieren ontdekken om de interactie tussen gebruikers en computersystemen te verbeteren en de toegankelijkheid voor alle gebruikers te vergroten.

7 Literatuurlijst

- [1] Chrome for Developers , “Development basics”, [Online].
Beschikbaar: <https://developer.chrome.com/docs/extensions/mv3/getstarted/development-basics/>.
[Geraadpleegd op 8 november 2023]
- [2] Chrome for Developers , “Handle events with service workers”, [Online]. Beschikbaar:
<https://developer.chrome.com/docs/extensions/mv3/getstarted/tut-quick-reference/>
[Geraadpleegd op 8 november 2023]
- [3] Chrome for Developers, “chromo.apslag | API”, [Online]
Beschikbaar: <https://developer.chrome.com/docs/extensions/reference/api/storage?hl=nl>
[Geraadpleegd op 20 november 2023]
- [4] dev.to, “How to Change a webpage UI with a Chrome Extension”, [Online]
Beschikbaar: <https://dev.to/opensauced/how-to-change-a-webpage-ui-with-a-chrome-extension-jn8>
[Geraadpleegd op 8 november 2023]
- [5] w3schools “Window scrollBy()”, [Online]
Beschikbaar: https://www.w3schools.com/jsref/met_win_scrollby.asp
[Geraadpleegd op 18 november 2023]
- [6] geeksforgeeks “How to simulate a click with javascript”, [Online]
Beschikbaar: <https://www.geeksforgeeks.org/how-to-simulate-a-click-with-javascript/>
[Geraadpleegd op 20 november 2023]
- [7] developer.mozilla.org “MouseEvent: MouseEvent() constructor – Web API’s | MDN”, [Online]
Beschikbaar: <https://developer.mozilla.org/en-US/docs/Web/API/MouseEvent/MouseEvent>
[Geraadpleegd op 18 november 2023]
- [8] w3schools “Window history.back()”, [Online]
Beschikbaar: https://www.w3schools.com/jsref/met_his_back.asp
[Geraadpleegd op 18 november 2023]
- [9] Tobii, “What can you do with eye tracking ?”, [Online]. Beschikbaar: <https://www.tobii.com/learn-and-support/get-started/what-can-you-do-with-eye-tracking> [Geraadpleegd op 10 november 2023]
- [10] Gazrecorder, “Try it FREE”, [Online]. Beschikbaar: <https://app.gazrecorder.com/>
[Geraadpleegd op 10 november 2023]
- [11] Webgazer.js “Democratizing Webcam Eye Tracking on the Browser”, [Online].
Beschikbaar: <https://webgazer.cs.brown.edu/> [Geraadpleegd op 15 november 2023]
- [12] W. Wang, “Integrating GazeCloudAPI, a high accuracy webcam based eye-tracking solution, into your own web-app”, [Online]. Beschikbaar: <https://medium.com/@williamwang15/integrating-gazecloudapi-a-high-accuracy-webcam-based-eye-tracking-solution-into-your-own-web-app-2d8513bb9865#:~:text=As%20can%20be%20seen%20from> [Geraadpleegd op 15 november 2023]
- [13] is er externe software nodig voor eye-tracking
- [14] iMotions, “Webcam-Based Eye Tracking vs. an Eye Tracker [Pros & Cons] - iMotions”, [Online].
Beschikbaar: <https://imotions.com/blog/learning/best-practice/webcam-eye-tracking-vs-an-eye-tracker/#:~:text=While%20dedicated%20eye%20trackers%20offer>
[Geraadpleegd op 10 november 2023]

- [15] iMotions, "10 Free Eye Tracking software Programs [Pros and Cons]", [Online] Beschikbaar: <https://imotions.com/blog/insights/trend/free-eye-tracking-software/> [Geraadpleegd op 10 november 2023]
- [16] opencv, "OpenCV", [Online]. Beschikbaar: <https://opencv.org/> [Geraadpleegd op 15 november 2023]
- [17] MediaPipe Studio, "Face Landmark Detection", [Online] Beschikbaar: https://mediapipe-studio.webapps.google.com/demo/face_landmarker [Geraadpleegd op 15 november 2023]
- [18] M. Burrell, "Injecting Content Scripts in Chrome Extensions: Statically vs Programmatically", [Online]. Beschikbaar: <https://medium.com/@fullstackmatt/injecting-content-scripts-in-chrome-extensions-statically-vs-programmatically-763ba90e6fc3> [Geraadpleegd op 20 november 2023]
- [19] Apple, "Apple announces new accessibility features including Eye Tracking, Music Haptics, and Vocal Shortcuts", [Online] Beschikbaar: <https://www.apple.com/newsroom/2024/05/apple-announces-new-accessibility-features-including-eye-tracking/> [Geraadpleegd op 20 mei 2024]
- [20] iMotions, "Eye Tracker Prices – An Overview of 15+ products", [Online]. Beschikbaar: <https://imotions.com/blog/learning/product-news/eye-tracker-prices/> [Geraadpleegd op 25 mei 2024]
- [21] GazeBoard, "Try it FREE", [Online]. Beschikbaar: <https://gazerecorder.com/gazeboard/> [Geraadpleegd op 25 mei 2023]

8 Bijlages

8.1 Verslag Faktion

Verslag van de Gastspreking door Jeroen Boeye, PhD – Head of Machine Learning

Datum: 10 januari 2024

Spreker: Jeroen Boeye, PhD

Onderwerp: Toepassing van Kunstmatige Intelligentie bij Faktion

Inleiding

Op 10 januari 2024 gaf Jeroen Boeye, PhD en hoofd van Machine Learning bij Faktion, een gastspreking waarin hij de aanpak en principes van Faktion op het gebied van kunstmatige intelligentie (AI) deelde. In deze notities worden zijn inzichten en ervaringen besproken.

Ervaring en Filosofie

Jeroen Boeye beschikt over zeven jaar ervaring in de AI-industrie. Hij benadrukt het belang van een perfecte balans tussen AI-toepassing en software engineering om succesvolle AI-projecten te realiseren. Dit vormt de kern van de werkmethode bij Faktion.

Spin-offs van Faktion

Faktion heeft verschillende spin-offs ontwikkeld die zich richten op specifieke AI-oplossingen:

- **Chatlayer:** Een platform voor chat- en voicebots, ontworpen om communicatie te automatiseren.
- **MetaMaze:** Een platform voor intelligente documentverwerking, gericht op het efficiënter maken van documentbeheer en -analyse.

Platforms en Ventures voor Klanten

Tijdens de gastspreking noemde Jeroen Boeye enkele platforms en ventures die Faktion voor hun klanten heeft gebouwd:

- **Proximus Interact:** Een platform voor interactieve klantcommunicatie.
- **Jane:** Een AI-gedreven oplossing voor klantinteracties.
- **Rigzi:** Een project gericht op het verbeteren van bedrijfsprocessen door middel van AI.
- **Chroligo:** Een venture die zich richt op geavanceerde data-analyse.

Stappen naar een Succesvol AI-project

1. Valideer de Business Case

Het eerste en belangrijkste aspect van een AI-project is het valideren van de business case. Dit zorgt ervoor dat het project een duidelijk doel en toegevoegde waarde heeft.

2. Is de Data Beschikbaar?

Een cruciale stap is het evalueren of de benodigde data beschikbaar is. Jeroen Boeye legt uit dat de aanwezigheid van data geen zwart-wit situatie is; het gaat om een goede balans tussen de kwaliteit en kwantiteit van de data.

Wat als de Data Niet Beschikbaar Is?

Wanneer data niet beschikbaar is, raadt hij aan om creatieve manieren te vinden om synthetische data te genereren. Hoewel echte data altijd de voorkeur geniet, kan synthetische data een waardevol alternatief zijn.

3. Snel Falen, Sneller Leren

Het principe van 'fail fast, learn faster' is essentieel in AI-projecten. Dit betekent dat men snel moet experimenteren en leren van fouten om sneller tot verbeterde oplossingen te komen.

4. Communicatie

Effectieve communicatie is cruciaal voor het succes van AI-projecten. Regelmatige updates en duidelijke communicatie met alle betrokkenen zijn noodzakelijk om het project op koers te houden.

Belang van Data Pre-processing

Data pre-processing is een uitdagende maar fundamentele stap in elk AI-project. Jeroen Boeye benadrukt het belang van het goed beheersen van dit proces, aangezien het de basis vormt voor alle verdere analyses en modellering.

Behandel Data Niet als een Statistisch Middel

Data moet dynamisch worden behandeld. Dit houdt in dat men soms data moet creëren (VRR) of de taak moet aanpassen afhankelijk van de beschikbaarheid en kwaliteit van de data.

Conclusie

De gastspreking van Jeroen Boeye bood waardevolle inzichten in de aanpak van Faktion bij de toepassing van AI. De nadruk op de balans tussen AI en software engineering, het creatief omgaan met dataproblemen, en het belang van snelle iteratie en effectieve communicatie zijn cruciale elementen voor succes. Door deze principes te volgen, kunnen bedrijven effectievere en efficiëntere AI-oplossingen ontwikkelen en implementeren.

8.2 Verslag Goomyx

Verslag van de Gastspreking door Alex Christiaens van Goomyx

Datum: 23 januari 2024

Spreker: Alex Christiaens, Co-founder van Goomyx

Onderwerp: Hoe Goomyx Projecten Ontwikkelt voor Bedrijven met een Sociale Impact

Inleiding

Op dinsdag 23 januari 2024 gaf Alex Christiaens, co-founder van het bedrijf Goomyx, een gastspreking waarin hij uitlegde hoe Goomyx projecten uitvoert voor bedrijven met een sociale impact. Hij besprak het volledige proces van projectontwikkeling, de uitdagingen die ze tegenkomen en hoe ze deze overwinnen. Daarnaast gaf hij inzicht in hun klantenbestand en welke soort klanten ze in de toekomst willen aantrekken.

Klanten en Projecten

Alex begon de presentatie door enkele van hun klanten en projecten te bespreken:

1. **Remote Lab Connection voor Vives:** Een project gericht op het verbeteren van afstandsonderwijs.
2. **Flexicap:** Stelt KMO's en particulieren in staat hun energieverbruik te analyseren en mogelijke verbeteringen te identificeren.
3. **Sport R:** Een project voor patiëntopvolging.

Projectontwikkelingsproces

Alex beschreef de typische stappen die Goomyx volgt bij het ontwikkelen van een project:

1. **Probleemstelling omzetten naar een functionele analyse:** Identificeren van de behoeften en eisen van het project.
2. **Wireframes:** Visuele schetsen van de gebruikersinterface en de structuur.
3. **Klikbare mock-up:** Een interactieve prototype die uitgebreid wordt getest voordat de ontwikkeling begint.
4. **Ontwikkeling:** Het omzetten van de klikbare mock-up naar een volledig functioneel product, waarbij eventuele noodzakelijke aanpassingen worden doorgevoerd. Deze fase omvat ook uitvoerig testen.
5. **Oplevering:** Het voltooide product wordt afgeleverd en geïmplementeerd.

Onderzoek en Aanpak bij Verschillende Projecten

Alex illustreerde hoe Goomyx onderzoek uitvoert naar de behoeften van hun projecten en hoe ze omgaan met beperkte budgetten:

Voorbeeldprojecten

1. Vives Remote Lab Connection Tool:

- Gericht op middelbare en hoge scholen.
- Oplossing: Gebruik van een Raspberry Pi met een Apache-server om kosten te besparen en toch effectieve afstandsonderwijsoplossingen te bieden.

2. Konect VZW:

- Doel: Mensen met mentale beperkingen helpen bij het oefenen van sollicitatievaardigheden.
- Uitdagingen: Klein budget en bruikbaarheid voor mensen met mentale beperkingen.
- Oplossingen:
 - Gebruik van iconen voor herkenbaarheid.
 - Implementatie van een lichte framework (Svelte) en Google Sheets als database.
 - Gebruik van magic links voor eenvoudige authenticatie via Firebase.

3. PhonAID:

- Onderzoek naar vroege detectie van stemstoornissen met AI.
- Doel: Detectie van aandoeningen zoals heesheid, stemknobbels en tumoren, geïnspireerd door AI-detectie van hartritmes en huidaandoeningen.

Intern Onderzoek en Kennisdeling

Naast projectwerk voert Goomyx ook intern onderzoek uit om hun kennis te verbreden. Elke maand wijden ze een dag aan onderzoek, waarvan de bevindingen worden bijgehouden in een kennisdatabase in Notion.

Onderzoeksmethoden

- Gebruik van Google, YouTube en ChatGPT.
- Deelname aan workshops.
- Het experimenteren met en implementeren van nieuwe technologieën in projecten.

Vraag- en Antwoordsessie

Tijdens de vraag- en antwoordsessie werd gevraagd waar Goomyx hun projecten deployt. Alex gaf aan dat ze voornamelijk Netlify en Raspberry Pi's gebruiken. Ook besprak hij de uitdagingen van zelfstandig ondernemerschap, met name het vele papierwerk.

Conclusie

Alex Christiaens gaf een inspirerende kijk op hoe Goomyx als startup projecten ontwikkelt en onderzoek uitvoert. Hij benadrukte het belang van grondig onderzoek, creatieve oplossingen voor budgetbeperkingen en het openstaan voor nieuwe technologieën. Zijn presentatie toonde duidelijk aan dat succesvol projectmanagement en innovatie hand in hand gaan, zelfs binnen beperkte middelen.

Met deze inzichten biedt Goomyx een waardevol model voor andere startups en bedrijven die streven naar sociale impact door middel van technologische oplossingen.

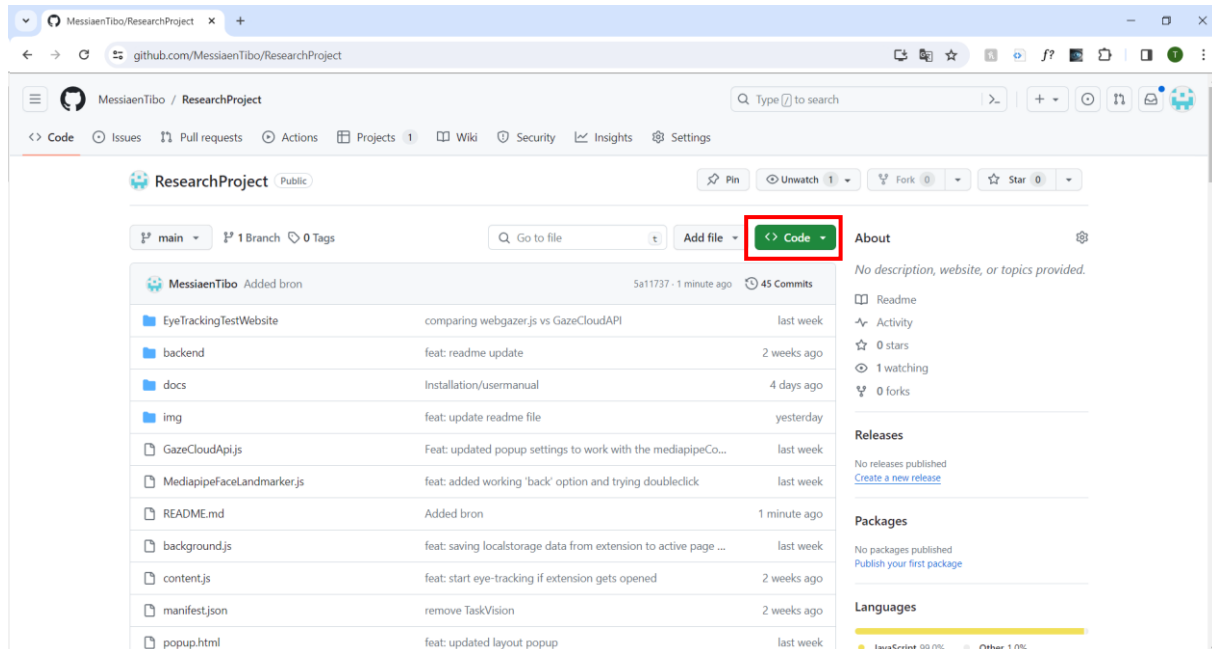
8.3 Installatie & gebruikershandleiding

Chrome extension

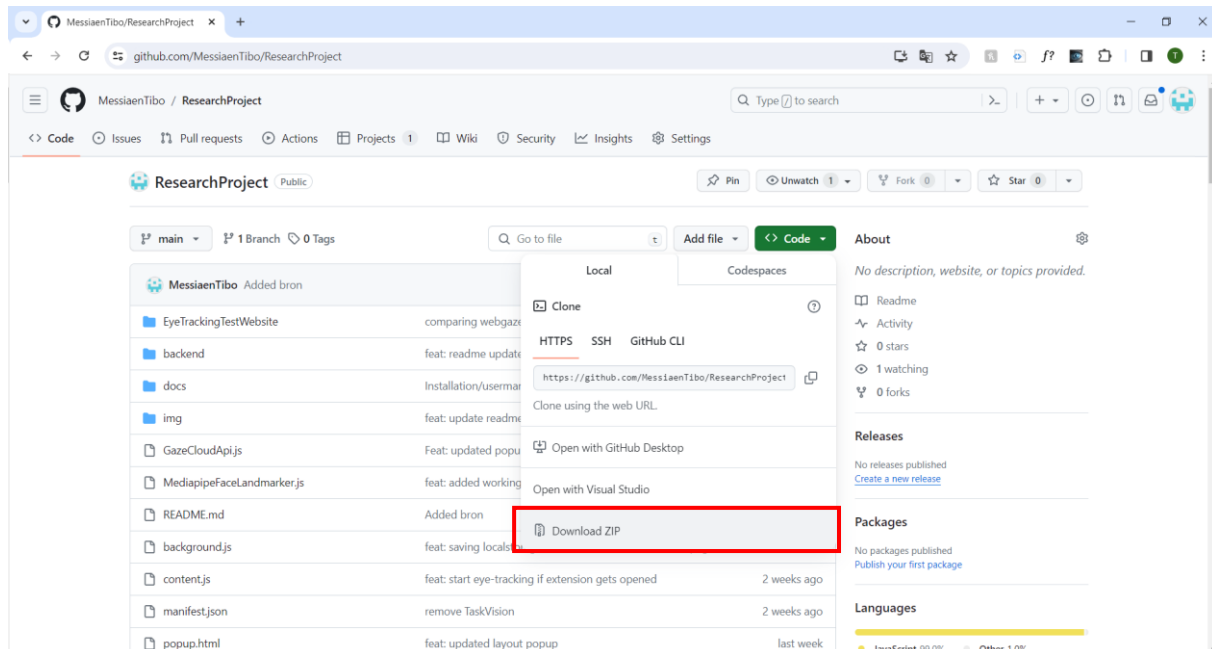
Eye-tracking Chrome extension installation manual

Downloading the code from github

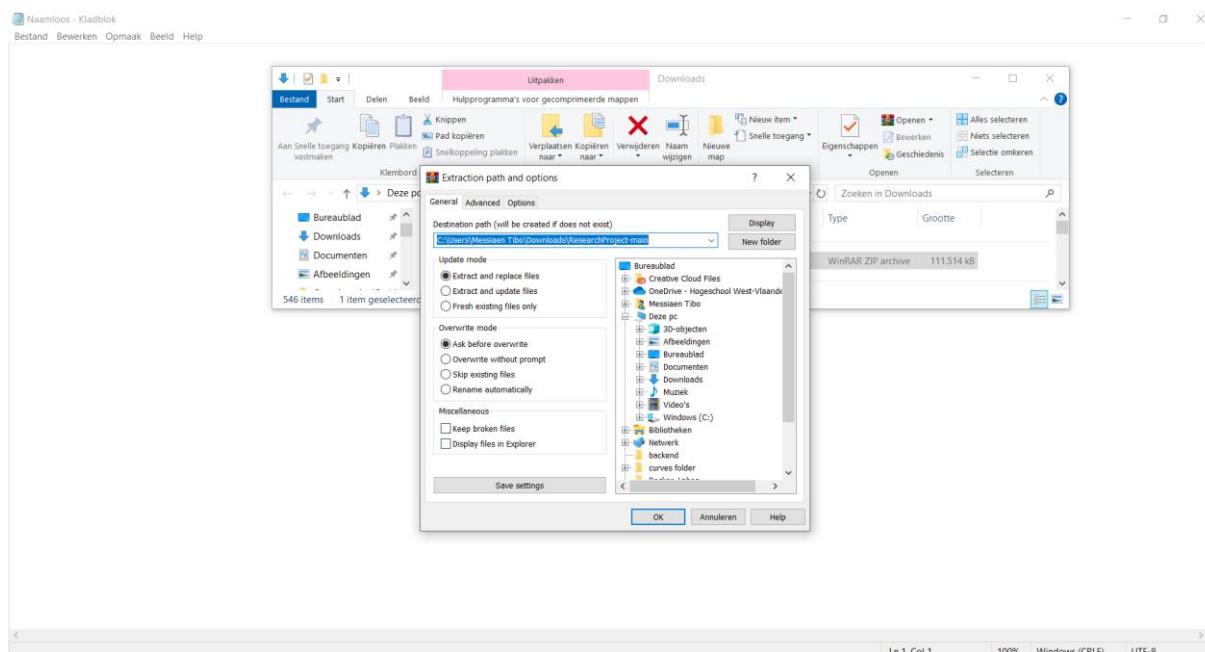
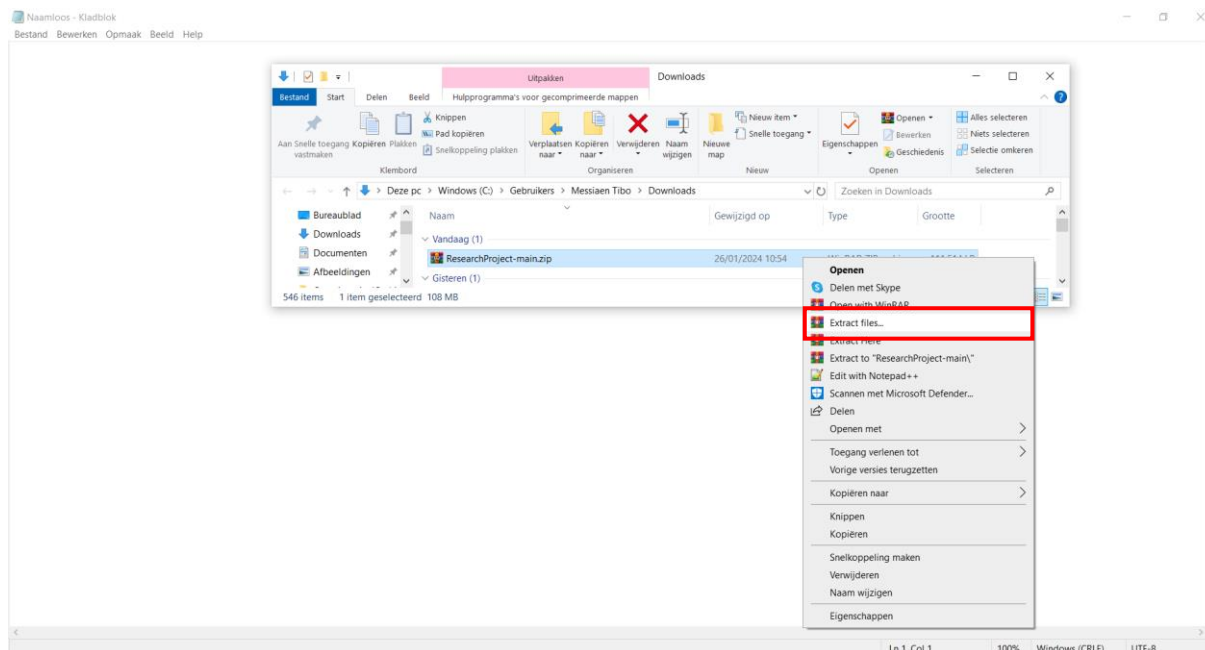
Go to: <https://github.com/MessiaenTibo/ResearchProject>



Click on the code button and download the zip file

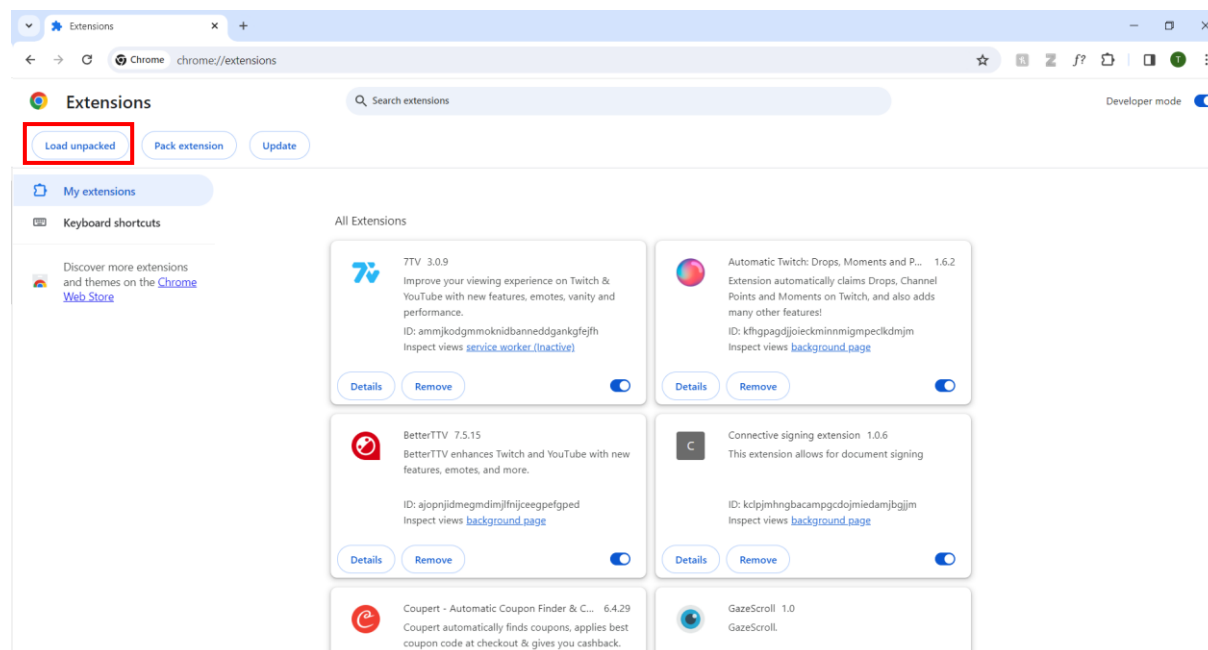


Go to the downloaded zip and extract the file to a folder of choice

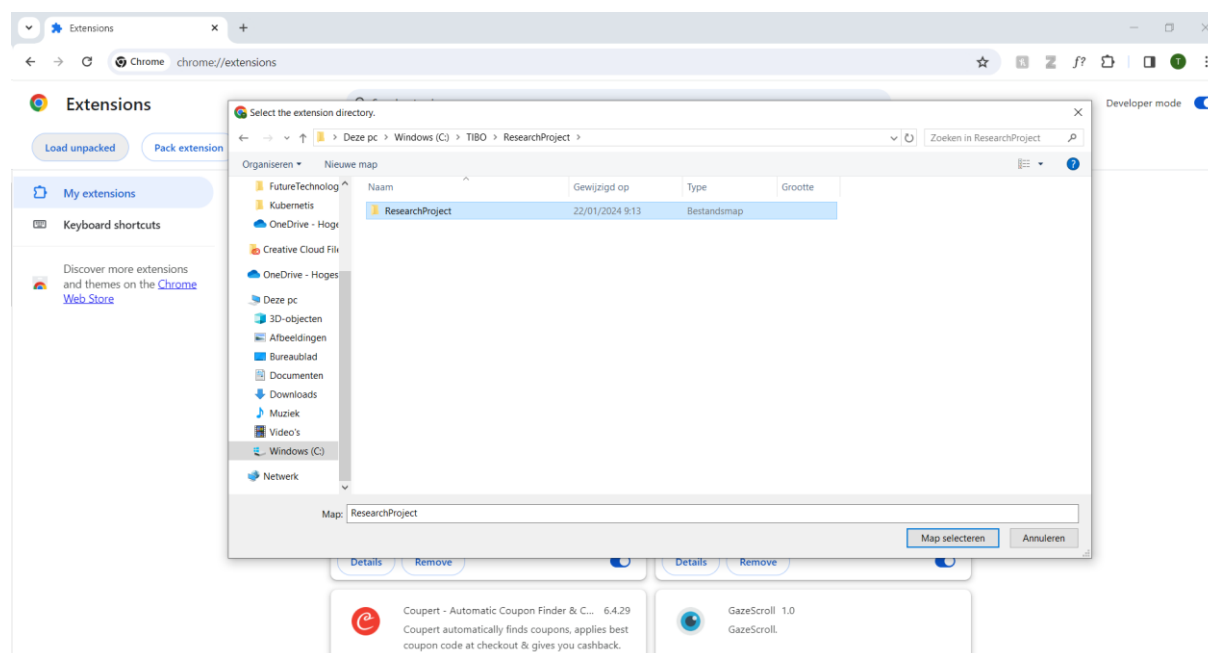


Add the crhome extension

Go to “chrome://extensions/” in a chrome browser



Click in the “Load unpacked” button
And open the just downloaded code map

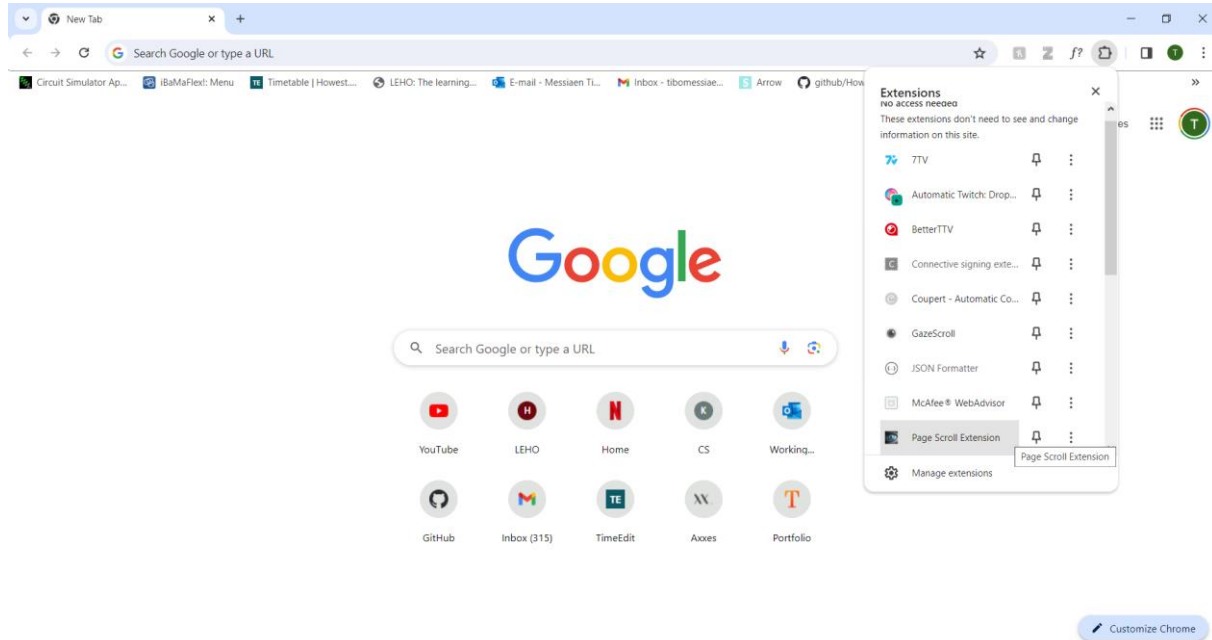


Eye-tracking Chrome extension user manual

Open the extension

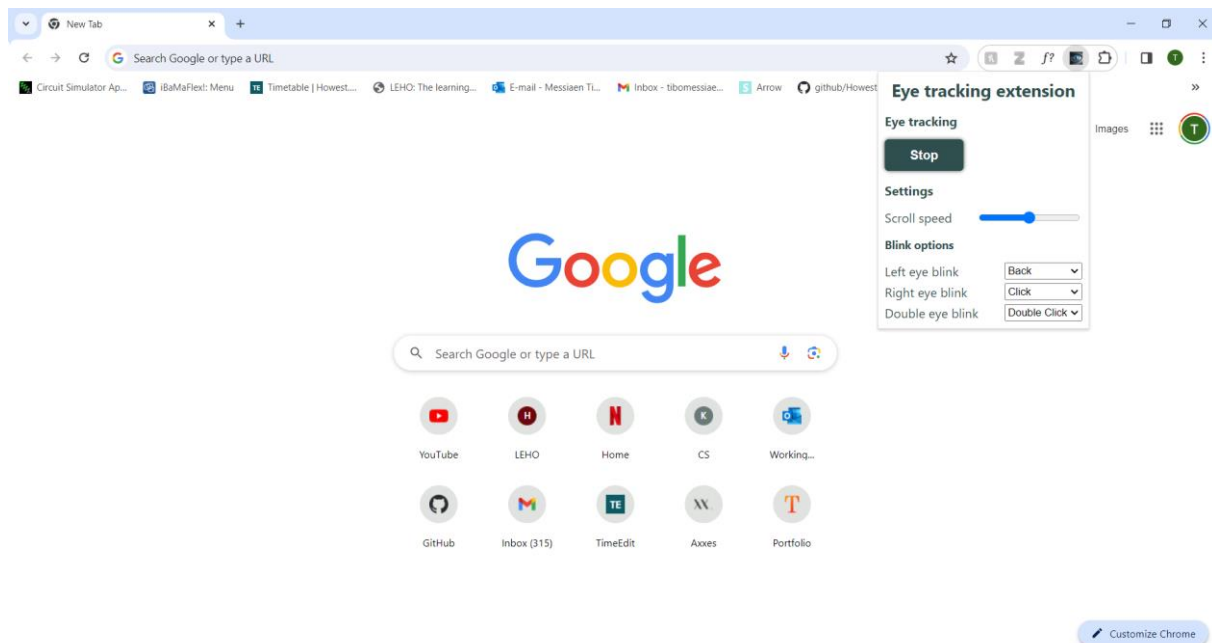
Open a Chrome browser and click on the extension button

Click on the desired extension



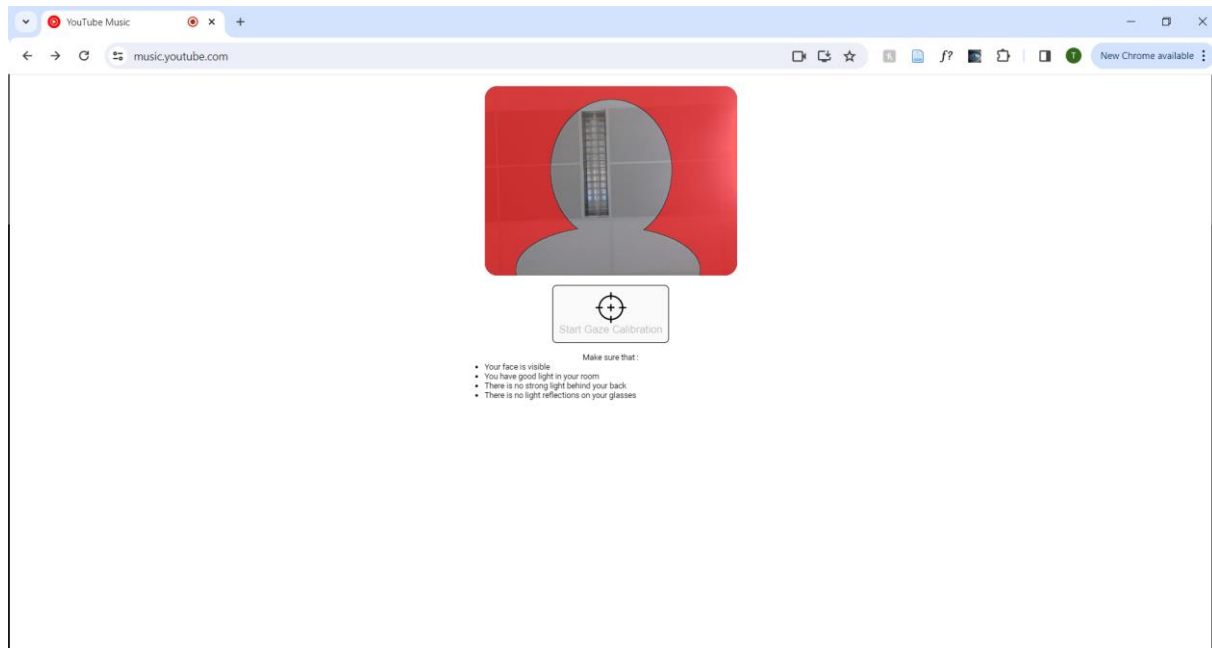
The extension opens and is ready to be used.

You can also customize the scrolling speed and blink options

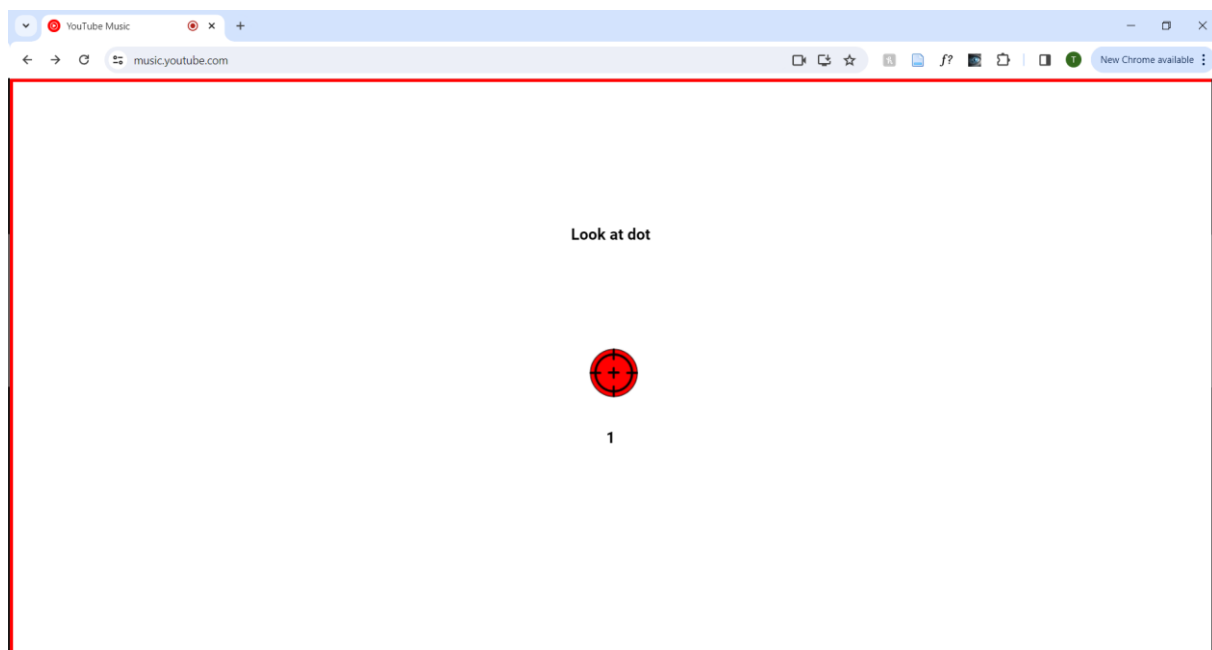


Start Gaze Calibration

After opening the chrome extension this screen will appear on the current website. You can now start the calibration process by pressing the "Start Gaze Calibration" button.



Look at the red dot and follow the instructions to calibrate the eye tracking



Ready to interact with the website

Now you are ready to go and Interact with the page.

You can select page elements by looking at them. A red border will appear around it.

Based on the selected blink options you can click, double click or go back to the previous window.

