

Examensarbete

NACKADEMIN

Examensarbete

S.H.I.G.S

Simply a Highly Intelligent Gesture System

Student: Jimmy Kroneld

Utbildning och klass: Mjukvaruutvecklare, IoT (IoT22)

Handledare: Oscar Bexell

Datum och ort: 2024-01-11 Stockholm

Examensarbete

Sammanfattning

Detta examensarbete fokuserar på utvecklingen av "S.H.I.G.S" (Simply a Highly Intelligent Gesture System), ett innovativt system för handgestbaserad styrning av Spotify. I en värld där traditionella gränssnitt som röst- och pekskärmskontroller ofta är otillräckliga, syftar detta projekt till att erbjuda en alternativ lösning för musikstyrning, särskilt i köksmiljöer och för personer med talsvårigheter. Genom att använda teknologier som MediaPipe och OpenCV, har systemet utvecklats och testats på en Raspberry Pi 4 med en Raspberry Pi-kamera. Resultaten visar att systemet är tekniskt genomförbart, användarvänligt och pålitligt, med potential att förbättra användarupplevelsen i vardagliga situationer. Projektet understryker vikten av tillgänglighet och intuitivitet i teknologiska produkter och öppnar upp för framtida forskning och utveckling inom området för handgestbaserade gränssnitt.

Summary

This thesis focuses on the development of "S.H.I.G.S" (Simply a Highly Intelligent Gesture System), an innovative gesture-controlled system for Spotify. In a world where traditional interfaces like voice and touchscreen controls often fall short, this project aims to provide an alternative solution for music control, particularly in kitchen environments and for individuals with speech difficulties. Utilizing technologies such as MediaPipe and OpenCV, the system has been developed and tested on a Raspberry Pi 4 with a Raspberry Pi camera. The results demonstrate that the system is technically feasible, user-friendly, and reliable, with the potential to enhance user experience in everyday situations. The project emphasizes the importance of accessibility and intuitiveness in technological products and opens up avenues for future research and development in the field of gesture-based interfaces.

Examensarbete

Innehållsförteckning

Sammanfattning	2
Summary	2
Innehållsförteckning.....	3
Inledning	4
1.1 Bakgrund.....	4
1.2 Syfte.....	4
1.3 Frågeställningar	4
1.4 Metodbeskrivning.....	5
Använda Teknologier	5
2.1 MediaPipe	5
2.2 OpenCV.....	6
Resultat.....	6
3.1 Teknisk Implementering och Systemutveckling.....	6
3.2 Utmaningar och Lösningar	6
3.3 Gestigenkänning och Systemrespons	7
3.4 Användarvänlighet och Intuitivitet.....	7
3.5 Pålitlighet och Effektivitet.....	7
3.6 Sammanfattning av Resultat	8
Diskussion	8
Slutsatser	8
Referenser	9
Bilagor.....	10
1:Hand-Landmarks (Hand Landmarks Image, 2024)	10
2:Raspberry pi 4, fotograferad av Jimmy Kroneld	10
3:Raspberry pi kamera (Module 3) fotograferad av Jimmy Kroneld.....	11
4: Raspberry Pi-setup med kamera, fotograferad av Jimmy Kroneld	11
5:Play_Music, fotograferad av Jimmy Kroneld	12
6:Pause_Music, fotograferad av Jimmy Kroneld	12
7:Next_Song, fotograferad av Jimmy Kroneld	12
8:Previous_Song, fotograferad av Jimmy Kroneld	13
9:Increase_volume, fotograferad av Jimmy Kroneld.....	13
10:Decrease_Volume, fotograferad av Jimmy Kroneld	13

Examensarbete

Inledning

I den snabbt digitaliserande världen möter vi nya interaktionsutmaningar, särskilt i miljöer där traditionella gränssnitt som röst- och pekskärmkontroller är otillräckliga. Detta projekt syftar till att tackla dessa utmaningar genom att utveckla ett handgestbaserat system för att styra musikspelare, speciellt anpassat för användning i kök och för personer med talsvårigheter. Genom att erbjuda en alternativ interaktionsmetod siktar projektet på att öka tillgängligheten och användarvänligheten för musikstyrningsteknik. Denna rapport beskriver bakgrunden till projektet, detaljerar dess syfte och metodik, och presenterar de erhållna resultaten.

1.1 Bakgrund

I en alltmer uppkopplad värld, där smarta enheter blir centrala i våra liv, möter vi utmaningar inom användarinteraktion, särskilt i miljöer där traditionella styrmetoder inte är optimala. Detta är särskilt påtagligt i köksmiljöer och för personer med talsvårigheter, där röst- och pekskärmkontroller ofta faller kort. Denna bakgrund understryker behovet av att utforska och utveckla mer tillgängliga och effektiva interaktionsmetoder.

1.2 Syfte

Syftet med detta projekt är att utveckla och implementera ett innovativt system för musikstyrning baserat på handgester. Systemet syftar till att erbjuda en användarvänlig och tillgänglig lösning för musikstyrning i köksmiljöer och för personer med talsvårigheter, vilket bidrar till att öka tillgängligheten och förbättra den övergripande användarupplevelsen.

1.3 Frågeställningar

1. **Teknisk Implementering:** Hur kan handgestbaserad teknik med MediaPipe och OpenCV implementeras för att effektivt styra musikspelare i kök?

Examensarbete

2. **Användarinteraktion:** Vilka handgester är mest intuitiva och effektiva för användare när det gäller att styra musikuppspelning?
3. **Systemets Responsivitet:** Hur pålitligt och exakt kan systemet identifiera och svara på olika handgester under realtidsförhållanden?

1.4 Metodbeskrivning

Denna studie har fokuserat på att utveckla och testa ett handgestbaserat system för musikstyrning genom praktiska experiment och iterativa tester. Istället för att använda intervjuer eller enkäter, har projektet involverat direkt utveckling och utvärdering av handgester med hjälp av en Raspberry Pi 4 och en Raspberry Pi-kamera. Processen har inkluderat att självständigt identifiera och testa olika gester för att bedöma deras enkelhet och effektivitet i musikstyrningskontexten.

Genom att använda en uppsättning av handlandmarks som definierats av MediaPipe-teknologin från Google (se Bilaga 1 för en illustration av dessa landmarks), kunde jag utforska olika gesters effektivitet och användbarhet. Gesterna har valts och utvecklats för att vara enkla att utföra och tydliga att skilja från varandra. Detta arbete har inkluderat en process av personlig utvärdering och justering för att säkerställa att varje gest är intuitiv och praktiskt genomförbar.

OpenCV-tekniken har använts för bildanalys och igenkänning av handgester, och Spotify API har integrerats för att styra musikuppspelningen. Utvecklingsprocessen har bestått av design, kodning och noggrann självtester av systemet för att säkerställa dess pålitlighet och användarvänlighet. För en mer detaljerad översikt av projektets kod och struktur, se Github-projektet 'S.H.I.G.S' (Kroneld, 2024)

Använda Teknologier

2.1 MediaPipe

MediaPipe är en öppen källkodsplattform från Google, som erbjuder anpassningsbara maskininlärningslösningar för live och strömmande media. I detta projekt används MediaPipe för realtidsspårning av handrörelser och

Examensarbete

gestigenkänning. Det ger en robust ram för att upptäcka handlandmärken, vilket möjliggör för det geststyrda musikspelarsystemet att korrekt tolka olika handgester för musikstyrning. (Google, 2023)

2.2 OpenCV

OpenCV (Open Source Computer Vision Library) är ett öppen källkodsprogrambibliotek för datorseende och maskininlärning. OpenCV är utvecklat för att tillhandahålla en gemensam infrastruktur för datorseendeapplikationer och för att påskynda användningen av maskinperception i kommersiella produkter. I detta projekt används OpenCV för bildbehandling och fångande av videobilder från Raspberry Pi-kameramodulen 3 (se Bilaga 3). (OpenCV, 2023)

Resultat

3.1 Teknisk Implementering och Systemutveckling

Systemet utvecklades framgångsrikt genom att integrera OpenCV för bildanalys och igenkänning av handgester, samt Spotify API för musikstyrning, med en detaljerad uppbyggnad som visas i Bilaga 4. Användningen av Raspberry Pi 4 och en Raspberry Pi-kamera möjliggjorde effektiv spårning och tolkning av handgester i realtid (se Bilaga 2 och 3). Denna tekniska lösning visade sig vara robust och tillförlitlig under testerna.

3.2 Utmaningar och Lösningar

Under projektets gång stötte jag på några tekniska utmaningar:

- **Överlappande Gestproblem:** Ett initialt problem var att vissa handgester överlappade varandra, vilket ledde till svårigheter i att skilja dem åt med hög precision. För att hantera detta, initierade jag volymnivån till en fast nivå (50%) och begränsade systemet till att använda endast två gester för att öka eller minska volymen. Denna lösning förenklade gestigenkänningen och minskade risken för felaktig tolkning

Examensarbete

- **Känsliga Kamerakontakter:** Jag stötte på svårigheter med Raspberry Pi-kamerans känsliga kontakter. Efter att ha skadat en kabel, insåg jag vikten av att vara extra försiktig och rekommenderar att alltid ha en reservkabel tillgänglig.
- **Optimering av Gestigenkänningslogik:** Utvecklingen av gestigenkänningen krävde omfattande testning och justering. Genom att förstå sambanden mellan olika handpositioner och rörelser kunde jag skapa ett effektivt system för gestigenkänning. Dessutom underlättade minskningen av antalet gester för volymkontroll processen.

3.3 Gestigenkänning och Systemrespons

Gestigenkänningen var en central del av projektet. Systemet kunde korrekt identifiera och svara på en rad fördefinierade gester, inklusive de för att spela musik, pausa, byta låt och justera volymen (se Bilaga 5 till 10 för bilder av dessa gester). Genomsnittlig responsivitetstid var under 0,5 sekunder, vilket visar på systemets förmåga att snabbt och effektivt tolka användarens intentioner.

3.4 Användarvänlighet och Intuitivitet

De valda gesterna för projektet "S.H.I.G.S" har utformats för att vara intuitiva och lätta att utföra. Denna bedömning baseras på personliga tester och utvärderingar under utvecklingsprocessen. Även om en bredare användartestning inte har genomförts, indikerar dessa initiala tester att gesterna kan läras ut och förstås snabbt. Detta antyder att systemet har potentialen att vara användarvänligt och tillgängligt, även för personer utan tidigare erfarenhet av handgestbaserade gränssnitt. Vidare utvärdering och tester med en bredare användargrupp skulle dock vara nödvändiga för att fullständigt bekräfta dessa preliminära resultat.

3.5 Pålitlighet och Effektivitet

Systemets pålitlighet och effektivitet testades i olika miljöer för att säkerställa dess funktionalitet under varierande förhållanden. Resultaten visade att systemet bibehöll

Examensarbete

hög noggrannhet och pålitlighet även i miljöer med varierande ljusförhållanden och bakgrundsaktiviteter, vilket är avgörande för dess praktiska tillämpbarhet.

3.6 Sammanfattning av Resultat

Sammanfattningsvis visade resultaten att det handgestbaserade systemet för musikstyrning är tekniskt genomförbart, användarvänligt och pålitligt. Detta öppnar upp för möjligheter att använda liknande teknik i andra tillämpningar där traditionella gränssnitt är otillräckliga eller opraktiska.

Diskussion

Detta examensarbete har demonstrerat att handgestbaserade gränssnitt kan vara ett praktiskt och innovativt alternativ till traditionella styrmetoder för musikuppspelning. Genom att integrera OpenCV och MediaPipe har vi skapat en lösning som är både tillgänglig och intuitiv, vilket ger en möjlighet för personer med talsvårigheter eller i köksmiljöer att interagera med teknologi på ett nytt sätt. Samtidigt har vi identifierat utmaningar som systemets prestanda under varierande miljöförhållanden, vilket indikerar vikten av vidare utveckling och anpassning.

Med tanke på framtidens behov och tekniska framsteg, skulle nästa steg för projektet vara att utforska avancerade maskininlärningsmodeller som TensorFlow för att förbättra gestigenkänningen. Denna utvidgning skulle potentiellt kunna hantera en bredare variation av användarinteraktioner och tillhandahålla en ännu mer personlig och responsiv upplevelse.

Slutsatser

Projektet "S.H.I.G.S" har lagt en stabil grund för framtida innovation inom handgestbaserade användargränssnitt. Den nuvarande prototypen erbjuder en direkt och användarvänlig metod för musikstyrning, och jag ser en stor potential i att vidareutveckla systemet med en maskininlärningsdriven modell. Träning av en sådan modell skulle kräva en större datamängd och kan leda till en lösning som är mer flexibel och anpassningsbar för användarens unika gestmönster.

Examensarbete

Att ta itu med de utmaningar som presenterats under projektets gång, såsom systemets prestanda i olika miljöer, skulle kräva noggrann planering och en iterativ utvecklingsprocess. Denna process skulle omfatta samling av data, träning och validering av modellen, och skulle bidra till att ytterligare höja projektets värde och relevans.

I framtida arbete rekommenderar jag att utvidga forskningen för att inkludera användartester med en större och mer varierad grupp individer för att validera systemets effektivitet och användbarhet i verkliga scenarier. Genom att fortsätta att utforska och förbättra "S.H.I.G.S" kan jag bidra till en mer inkluderande och tillgänglig teknologisk framtid.

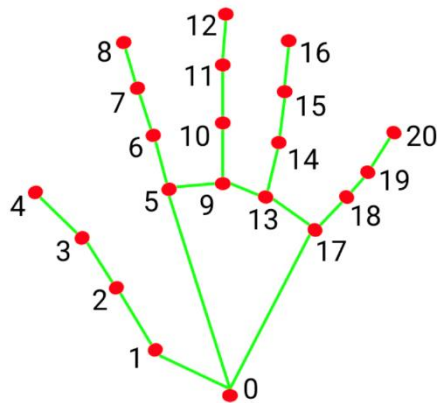
Referenser

- Google. (den 19 12 2023). *MediaPipe*. Hämtat från Google Developers: <https://developers.google.com/mediapipe>
- Hand Landmarks Image. (den 02 01 2024). Hämtat från Google Developers: <https://developers.google.com/static/mediapipe/images/solutions/hand-landmarks.png>
- Kroneld, J. (den 09 01 2024). *S.H.I.G.S*. Hämtat från GitHub: <https://github.com/Shallange/S.H.I.G.S>
- OpenCV. (den 19 12 2023). *About OpenCV*. Hämtat från OpenCV: <https://opencv.org/about/>
- Spotify. (den 26 12 2023). *Getting Started*. Hämtat från Spotify for Developers: <https://developer.spotify.com/documentation/web-api/tutorials/getting-started>
- Spotify. (den 26 12 2023). *Pause A User's Playback*. Hämtat från Spotify for Developers: <https://developer.spotify.com/documentation/web-api/reference/pause-a-users-playback>
- Spotify. (den 26 12 2023). *Refreshing Tokens*. Hämtat från Spotify for Developers: <https://developer.spotify.com/documentation/web-api/tutorials/refreshing-tokens>
- Spotify. (den 26 12 2023). *Set Volume For User's Playback*. Hämtat från Spotify for Developers: <https://developer.spotify.com/documentation/web-api/reference/set-volume-for-users-playback>
- Spotify. (den 26 12 2023). *Skip User's Playback To Next Track*. Hämtat från Spotify for Developers: <https://developer.spotify.com/documentation/web-api/reference/skip-users-playback-to-next-track>
- Spotify. (den 26 12 2023). *Skip User's Playback To Previous Track*. Hämtat från Spotify for Developers: <https://developer.spotify.com/documentation/web-api/reference/skip-users-playback-to-previous-track>
- Spotify. (den 26 12 2023). *Spotify Web API*. Hämtat från Spotify for Developers: <https://developer.spotify.com/documentation/web-api>
- Spotify. (den 26 12 2023). *Start A User's Playback*. Hämtat från Spotify for Developers: <https://developer.spotify.com/documentation/web-api/reference/start-a-users-playback>

Examensarbete

Bilagor

1:Hand-Landmarks



- | | |
|-----------------------|-----------------------|
| 0. WRIST | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP | 13. RING_FINGER_MCP |
| 3. THUMB_IP | 14. RING_FINGER_PIP |
| 4. THUMB_TIP | 15. RING_FINGER_DIP |
| 5. INDEX_FINGER_MCP | 16. RING_FINGER_TIP |
| 6. INDEX_FINGER_PIP | 17. PINKY_MCP |
| 7. INDEX_FINGER_DIP | 18. PINKY_PIP |
| 8. INDEX_FINGER_TIP | 19. PINKY_DIP |
| 9. MIDDLE_FINGER_MCP | 20. PINKY_TIP |
| 10. MIDDLE_FINGER_PIP | |

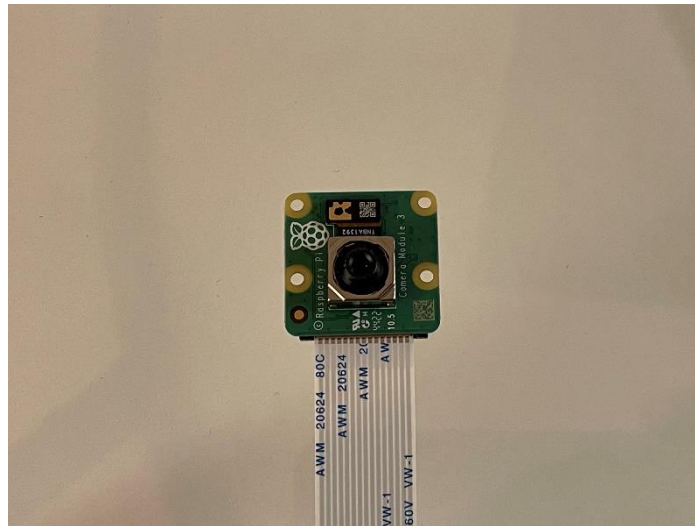
(Hand Landmarks Image, 2024)

2:Raspberry pi 4, fotograferad av Jimmy Kroneld

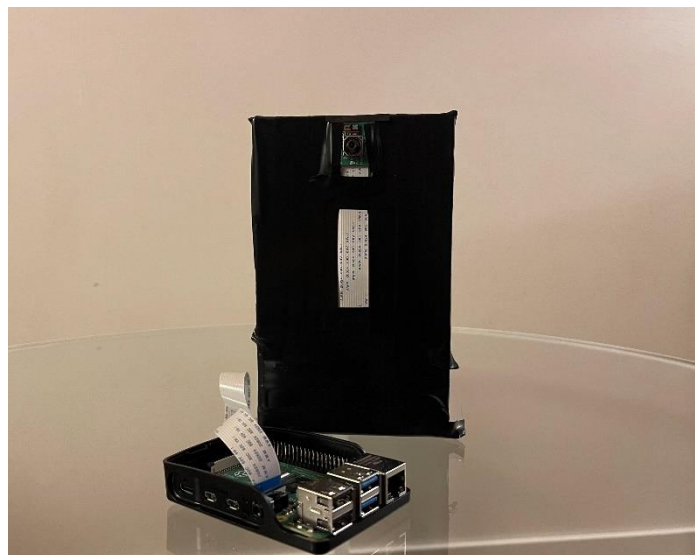


Examensarbete

3:Raspberry pi kamera (Module 3) fotograferad av Jimmy Kroneld

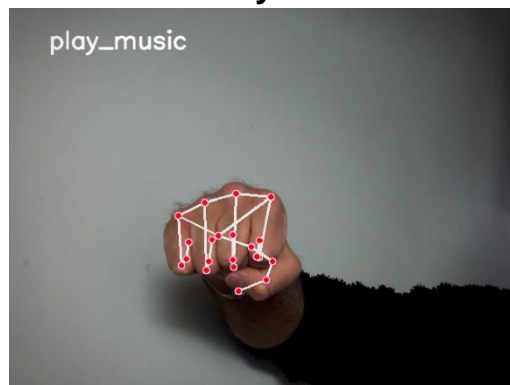


4: Raspberry Pi-setup med kamera, fotograferad av Jimmy Kroneld

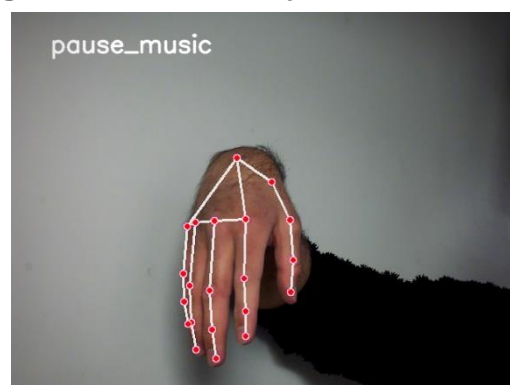


Examensarbete

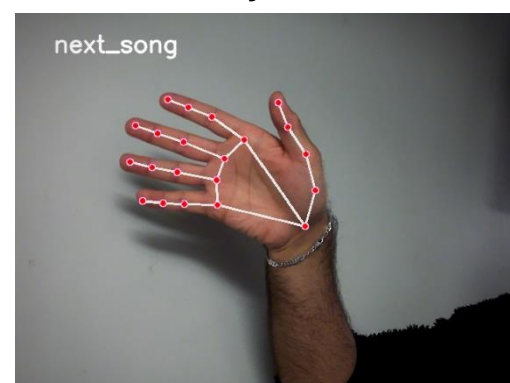
5:Play_Music, fotograferad av Jimmy Kroneld



6:Pause_Music, fotograferad av Jimmy Kroneld

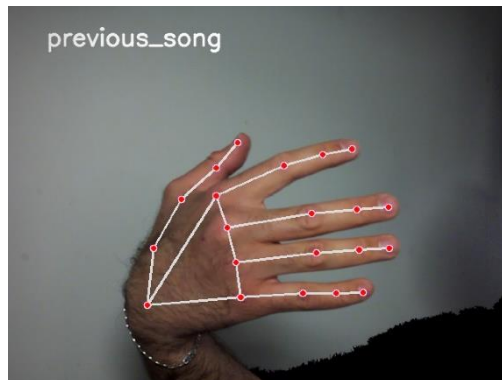


7:Next_Song, fotograferad av Jimmy Kroneld

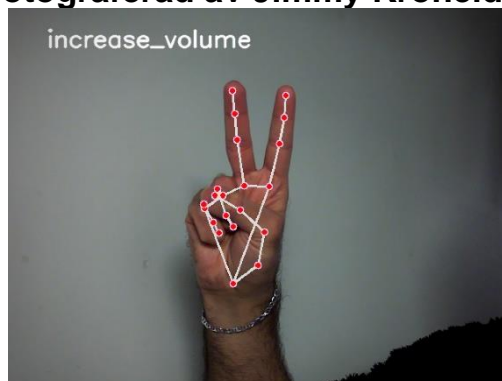


Examensarbete

8:Previous_Song, fotograferad av Jimmy Kroneld



9:Increase_volume, fotograferad av Jimmy Kroneld



10:Decrease_Volume, fotograferad av Jimmy Kroneld

