



# Avläsning av EEG-data från Emotiv Epoc X för ett SSVEP-baserat brain- computer interface för att spela Snake

Kandidatarbete i Data- och informationsteknik

Carl Erenstedt  
Alexander Ohlin  
David Persson  
Gustav Pihlquist  
Samuel Widén  
Carlos Yechouh



KANDIDATARBETE 2021

Avläsning av EEG-data från  
Emotiv Epoc X för ett SSVEP-baserat  
brain-computer interface för att spela Snake

Carl Erenstedt  
Alexander Ohlin  
David Persson  
Gustav Pihlquist  
Samuel Widén  
Carlos Yechouh



UNIVERSITY OF  
GOTHENBURG

---



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Institution för Data- och informationsteknik  
CHALMERS TEKNISKA HÖGSKOLA  
GÖTEBORGS UNIVERSITET  
Göteborg, Sverige 2021

Avläsning av EEG-data från Emotiv Epoc X för ett SSVEP-baserat brain-computer interface för att spela Snake

Carl Erenstedt Alexander Ohlin David Persson Gustav Pihlquist Samuel Widén  
Carlos Yechouh

© Carl Erenstedt, Alexander Ohlin, David Persson, Gustav Pihlquist, Samuel Widén,  
Carlos Yechouh 2021.

Handledare: Torbjörn Tjellén, Institutionen för data- och informationsteknik

Examinator: Graham Kemp, Institutionen för data- och informationsteknik

Kandidatarbete 2021

Institution för Data- och informationsteknik

Chalmers tekniska högskola och Göteborgs universitet

SE-412 96 Göteborg

Telefon +46 31 772 1000

Omslag: Bild av Emotiv Epoc X samt en monitor med spelet Snake

Typeset in L<sup>A</sup>T<sub>E</sub>X

Göteborg, Sverige 2021

## Sammanfattning

Denna rapport beskriver arbetet med att utforma ett BCI som kan styra datorspelet Snake i realtid. Spelet styrs genom att inducera steady state visually evoked potentials i hjärnan via flimrande stimuli av vissa frekvenser. Den resulterande elektriska aktiviteten i hjärnan läses sedan ut med hjälp av elektroencefalografi via Emotiv Epoc X. Slutligen tolkas informationen till specifika instruktioner med antingen Canonical Correlation Analysis eller ett Convolutional Neural Network och används för att styra spelet. Den resulterande programvaran visar potentialen av BCI i realtidskontrollerade videospel men ytterligare studier behövs för att hitta de optimala situationerna för bästa noggrannhet och responstid. Slutsatsen visar att Canonical Correlation Analysis fungerar bra till detta syfte, samt att Convolutional Neural Network inte passar lika bra då mängden data som krävs för träning inte var tillräcklig.

Nyckelord: BCI, EEG, Snake, SSVEP, CCA, Emotiv Epoc, CNN, responstid

## Abstract

This report describes the work of designing a BCI that can control the video game Snake in real time. The game is controlled by inducing steady state visually evoked potentials in the brain via flickering stimuli of certain frequencies. The resulting electrical activity in the brain is then read using electroencephalography via Emotiv Epoc X. Finally, the information is interpreted to specific instructions with either Canonical Correlation Analysis or a Convolutional Neural Network and can be used to control the game. The resulting software demonstrates the potential of BCI in real-time controlled video games, but further studies are needed to find the optimal situations for best accuracy and response time. Furthermore, this work shows that Canonical Correlation Analysis works well for the intended purpose, but that a Convolutional Neural Network did not perform comparably as the amount of data required for training was not sufficient.

Keywords: BCI, EEG, Snake, SSVEP, CCA, Emotiv Epoc, CNN, response time

## **Ett stort tack till**

Vår handledare Torbjörn Tjellén. Tack för ditt enorma engagemang, tålamod och intresse för vårt projekt. Tack för att du hållit privatlektioner för oss om signalteori, och lagt ned så mycket tid på att styra oss i rätt riktning. Vi vill även tacka Arne Linde och Lars Norén som hjälpte oss köpa in den nödvändiga hård- och mjukvaran som krävdes för detta arbete.

Carl Erenstedt, Alexander Ohlin, David Persson, Gustav Pihlquist, Samuel Widén,  
Carlos Yechouh, Gothenburg, juni 2021

# Innehåll

<b>Förkortningslista</b>	<b>viii</b>
<b>Figurer</b>	<b>x</b>
<b>1 Inledning</b>	<b>1</b>
1.1 Syfte . . . . .	2
1.2 Problemformulering . . . . .	2
1.3 Avgränsningar . . . . .	3
<b>2 Bakgrund</b>	<b>4</b>
2.1 BCI och EEG . . . . .	4
2.2 Hjärnan och hjärnvågor . . . . .	4
2.3 Historia . . . . .	5
2.4 Nutida applikationer . . . . .	5
2.5 10-20 systemet . . . . .	6
2.6 Emotiv Epoc X och EmotivPro . . . . .	6
<b>3 Teori</b>	<b>8</b>
3.1 Signalutlösare . . . . .	8
3.1.1 P300 - Event Related Potential . . . . .	8
3.1.2 Motor Imagery . . . . .	8
3.1.3 Steady state visually evoked potential . . . . .	9
3.2 Canonical Correlation Analysis . . . . .	9
3.3 Convolutional Neural Network . . . . .	9
3.3.1 Artificiella Neurala Nätverk . . . . .	9
3.3.2 Artificiell neuron . . . . .	10
3.3.3 Aktiveringsfunktioner . . . . .	10
3.3.4 Convolutional layer . . . . .	11
3.3.5 Pooling layer . . . . .	11
3.3.6 Träning av nätverket . . . . .	12
<b>4 Tidigare forskning</b>	<b>13</b>
4.1 Liknande arbeten med Snake och ett BCI . . . . .	13
4.2 Optimala frekvenser för SSVEP . . . . .	13
4.3 Vinkelavstånd mellan stimuli . . . . .	14
4.4 Storlek och form av stimuli . . . . .	14
4.5 Emotiv Epoc i vardagssituationer . . . . .	15



---

4.6	BCI-illiteracy . . . . .	15
<b>5</b>	<b>Metod</b>	<b>16</b>
5.1	Hårdvara . . . . .	16
5.2	Mjukvaruimplementation . . . . .	17
5.2.1	Signalutlösare . . . . .	17
5.2.2	EmotivPro och Databuffer . . . . .	17
5.2.3	Databehandling . . . . .	18
5.2.4	CCA-klassifikation . . . . .	19
5.2.5	CNN-klassifikation . . . . .	20
5.3	Datainsamling . . . . .	21
5.3.1	Gemensamma faktorer . . . . .	21
5.3.2	Datainsamling med enskilda frekvenser . . . . .	21
5.3.3	Datainsamling med flera frekvenser i sekvens . . . . .	21
5.3.4	Datainsamling med ökande frekvenser . . . . .	22
5.4	Sammanfattning av metodik och arbetsprocessen . . . . .	22
<b>6</b>	<b>Resultat</b>	<b>24</b>
6.1	Signalflödet genom hela systemet . . . . .	24
6.2	Resultaten av att spela Snake . . . . .	25
6.3	Klassifikationsresultat av CNN . . . . .	25
6.4	Klassifikationsresultat med CCA . . . . .	26
6.4.1	Klassifikationsresultat på noggrannhetstest . . . . .	27
6.4.2	Klassifikationsresultat på test av individuella frekvenser . . . . .	28
6.4.3	Undersökande av responstid vid byte mellan olika frekvenser . . . . .	31
6.5	Sammanfattning av resultat . . . . .	33
<b>7</b>	<b>Diskussion</b>	<b>34</b>
7.1	Förbättringsområden . . . . .	34
7.1.1	Annan hårdvara och mjukvara . . . . .	34
7.1.2	Alternativa signalutlösare till SSVEP . . . . .	35
7.1.3	Förbättringar av klassifikationsmetoden . . . . .	35
7.2	Felkällor . . . . .	36
7.2.1	Frekvensval . . . . .	36
7.2.2	Variation mellan olika personer . . . . .	36
7.2.3	Verifiering av frekvenser på skärm med oscilloskop . . . . .	37
7.3	Etiska aspekter utav BCI teknologi . . . . .	38
7.4	Kompetensbrister och förändringar om arbetet skulle göras om . . . . .	39
7.5	Nästa steg . . . . .	39
7.6	Sammanfattning av diskussion . . . . .	40
<b>8</b>	<b>Slutsatser</b>	<b>41</b>
	<b>Bibliografi</b>	<b>43</b>

# Förkortningslista

- BCI (Brain-Computer Interface)
- EEG (Elektroencefalografi)
- ITR (Information Transfer Rate)
- SSVEP (Steady State Visually Evoked Potential)
- MI (Motor Imagery)
- CNN (Convolutional Neural Network)
- CCA (Canonical Correlation Analysis)
- LSL (Lab Streaming Layer)

# Figurer

2.1	Bild av 10-20 systemet . . . . .	6
3.1	Exempel på ett enkelt neuralt nätverk. . . . .	10
3.2	Exempel på max pooling med ett filter och stride av storlek två. . . . .	12
5.1	Bild av Emotiv Epoc X . . . . .	16
5.2	Bild av 4 hexagoner med frekvenserna 6.66 (Vänster), 8.57 (Upp), 12 (Höger) och 15 (Ned). . . . .	18
5.3	Bild av Snake och Signalutlösare. . . . .	19
6.1	Övergripande bild på systemet. . . . .	24
6.2	Stapeldiagrammet visar samtliga testpersoners resultat vid 15 sekunders observation av stimuli. . . . .	27
6.3	Stapeldiagrammet visar resultat från testperson 5. . . . .	28
6.4	Antal sekunder datainsamling mot noggrannhet i procent . . . . .	30
6.5	Antal sekunder datainsamling (logaritmisk skala) mot noggrannhet i procent . . . . .	30
6.6	Försöksperson 2 med 4 sekunders frekvenslängd . . . . .	32
6.7	Försöksperson 5 med 4 sekunders frekvenslängd . . . . .	32
7.1	Oscilloskop för 6.66Hz stimuli . . . . .	37
7.2	Oscilloskop för 8.57Hz stimuli . . . . .	37
7.3	Oscilloskop för 12Hz stimuli . . . . .	38
7.4	Oscilloskop för 15Hz stimuli . . . . .	38

# 1

## Inledning

Under det senaste århundradet har mänsklighetens teknologiska framgång accelererat utan motstycke. Trots det är vår hjärna så komplex att vi ännu inte till fullo har förstått hur den fungerar. Världen har utvecklats markant, men vi interagerar fortfarande med den på samma sätt som vi gjort sedan stenåldern och tidigare. För att den mänskliga hjärnan ska kunna kommunicera med omvärlden så effektivt som möjligt behövs ett sätt att skapa en direktkanal till hjärnan. Därför är den teknologiska utvecklingen av BCI (Brain-Computer Interface) ett steg för att mänskligheten ska kringgå sina biologiska begränsningar.

Ett BCI är ett kommunikationsmedel som läser hjärnvågor, oftast i form av elektrisk potential [1]. En teknik som kan användas kallas EEG (Elektroencefalografi) och fungerar genom att läsa från elektroder som sitter på skalpen [2]. När en neuron<sup>1</sup> aktiveras skickas en elektrisk signal, som plockas upp av elektroderna. Den elektriska signalen översätts till en digital signal, som sedan skickas vidare för databehandling för att slutligen få ut ett kommando som tolkas av en dator. Hjärnvågorna används därmed för att skapa en helt ny kommunikationskanal mellan hjärna och dator. Användaren av ett BCI kan då kommunicera med datorer på ett sätt som inte kräver en traditionell muskelstyrd användarinmatning.

Detta arbete undersöker därför om ett BCI kan användas för att smidigt kontrollera spelet Snake. Snake är ett klassiskt enspelarspel som går ut på att styra en orm mot äpplen för att äta dem, och på så sätt få ormen att växa. Målet är att äta så många äpplen som möjligt, och spelet tar slut om ormen krockar med sig själv. Än så länge är användandet av BCI inom spel och underhållning både nytt och ineffektivt jämfört med traditionella former av datorinteraktion som möss och tangentbord. Andra arbeten som undersökt användningen av BCI har ofta endast partiell kontroll över ett system, eller tar genvägar som att pausa för att läsa in data [3]. I detta arbete kommer inga sådana genvägar utnyttjas, då Snake istället kommer kontrolleras i nära realtid med enbart ett BCI.

För att Snake skall fungera tillsammans med ett BCI måste implementationen av Snake anpassas. Eftersom det är hjärnvågorna som skall bestämma input till Snake istället för en traditionell muskelstyrd användarinmatning behövs en lösning för tolkningen av hjärnvågor. Detta arbete kommer därför att använda SSVEP (Steady

---

<sup>1</sup>En neuron, eller nervcell, ansvarar för överförandet och mottagandet av information i hjärnan.

State Visually Evoked Potentials)<sup>2</sup> för att styra Snake genom att titta på ett av fyra stimuli som blinkar i olika frekvenser. Frekvenserna detekteras i hjärnan, där varje unik frekvens motsvarar en instruktion i spelet.

Idag används BCI främst som ett forskningsverktyg eller som hjälpmedel till människor med rörelseproblem. Teknologin är fortfarande i utvecklingsfasen och har inte till fullo nått den kommersiella marknaden. Undersökandet av BCI som ett vardagligt verktyg kan bidra till dess framtida tillgänglighet. Med ett BCI som kan användas för realtidsavläsningar öppnas användningsmöjligheter inom både underhållning och personligt bruk. Detta leder till ett ekonomiskt incitament att investera i området och bidrar till vidare utveckling av teknologin.

## 1.1 Syfte

Syftet med arbetet är att avläsa och klassificera hjärnvågor från en person genom att konstruera ett BCI, och på så sätt styra ormen i spelet Snake. Arbetet har som mål att minimera responstid och maximera noggrannhet, samt att undersöka olika klassifikationsmetoder för att uppnå målet. Klassificeringarna översätts till instruktioner som bestämmer riktning i Snake utan annan datorinteraktion, vilket ska resultera i en spelbar implementation.

## 1.2 Problemformulering

Följande problemformulering utgår ifrån syftet och är formulerat kring de frågor som behövs för att besvara syftet och få en spelbar implementation av Snake.

- Vad är systemets responstid, d.v.s. hur lång tid tar det från att användaren vill utföra en handling tills att denna faktiskt utförs?
- Vad är förhållandet mellan responstid och noggrannhet?
- Vilka metoder kan användas för att klassificera EEG-data till instruktioner?
- Hur kan olika mängd EEG-data per klassifikation påverka noggrannhet och responstid?

För att Snake ska kännas smidigt och lättanvänt krävs det att responstiden är låg. Responstiden av systemet definieras som den förflutna tiden från att användaren observerar ett stimuli tills att systemet klassificerar EEG-datan. Då ett beslut fattas hos användaren ska den önskade händelsen ske på gränssnittet inom en tillräckligt kort tid för att användaren ska känna att denna är i kontroll av spelet. Ifall denna respons är långsam blir spelupplevelsen frustrerande, då handlingen sker senare än användaren hade tänkt sig. Responstiden får dock inte ha för stor negativ påverkan på noggrannheten, och vice versa. Därför krävs det att förhållandet mellan dessa

---

<sup>2</sup>SSVEP är en metod för signalutlösning genom blinkande stimuli. När en person observerar ett stimuli som blinkar med en viss frekvens upprepas denna frekvens i personens hjärnvågor. För mer information, se 3.1.3

variabler undersöks, så att noggrannheten är tillräcklig för att kontrollera spelet utan att fördröjningen på handlingar blir för hög.

Det kommer även att ske en undersökning av två olika klassifikationsmetoder, CCA (Canonical Correlation Analysis) och CNN (Convolutional Neural Network) vilka beskrivs i 5.2.4 respektive 5.2.5. Målet med denna undersökning är att bedöma metodernas applicerbarhet för ämnesområdet.

### **1.3 Avgränsningar**

Arbetet avser att anpassa klassifikationsmodellen till en generell lösning och modellen kommer därav inte kalibreras för varje individ. En modell som inte är anpassad till en specifik individ har potential att vara användbar av alla.

Detta arbete undersöker ej processen av att avläsa hjärnvågor med EEG, utan bygger endast på redan existerande kunskap och produkter inom området. Anledningen till detta är att färdiga kommersiella lösningar redan finns ute på marknaden samt att detta ligger utanför informationsteknikens omfattning.

# 2

## Bakgrund

I detta kapitel kommer bakgrunden till arbetet att beskrivas. Historia och förklaringar till arbetets centrala områden av BCI och hjärnvågor kommer presenteras. Standarder för EEG-avläsning och avläsaren som använts i denna rapport kommer även detaljeras.

### 2.1 BCI och EEG

En enkel beskrivning av ett BCI är som namnet avslöjar, ett gränssnitt mellan hjärna och dator. Det börjar med att elektriska signaler utlöses i hjärnan av en signalutlösningsteknik (se sektion 3.1), för att sedan genomgå databehandling. Innan databehandling brukar även datan som läses av elektroderna reduceras på något sätt, som till exempel att endast läsa data från de elektroder som ligger nära den hjärnlob vars aktivitet man är intresserad av. Databehandling utgörs av bearbetning, klassificering och koppling till specifika beslut. Dessa beslut kan sedan användas för att kontrollera en dator.

EEG är inte den enda metoden för avläsning av elektriska signaler från hjärnan. Alternativ till EEG inkluderar MEG (magnetoencefalografi), PET (positronemissionstomografi), fMRI (funktionell magnetresonanstomografi) och optisk avbildning [4]. Dock är dessa metoder mer invasiva, komplicerade, dyra, eller tar längre tid att ge resultat. Av dessa anledningar är EEG mest lämpad för detta arbete.

### 2.2 Hjärnan och hjärnvågor

Storhjärnan består av två hjärnhalvor, eller *hemisfärer*, som är ihopkopplade genom hjärnbalken. Storhjärnan delas vanligtvis in i fyra olika lober, som ansvarar över specifika hjärnfunktioner. Eftersom det finns två hemisfärer finns det också två av varje typ av lob, vilka är [5]:

- **Frontalloberna:**

Frontalloberna, som är de största loberna utgör den främre delen av storhjärnan. Här finns det motoriska centrum som kontrollerar kroppens rörelser. Även personligheten, koncentrationsförmågan och planeringsförmågan sitter här.

- **Temporalloberna:**  
Hjärnans temporallober sitter nära respektive tinning. Dessa kopplar bland annat ihop hörselintrycket med minnet, vilket bidrar till att musik kan ge en stark associering till minnen.
- **Parietalloberna:**  
I parietalloberna bearbetas och kombineras intrycken från våra sinnen och på så sätt tolkas en upplevelse. Dessa hjärnlobor är också mer aktiva vid lösning av logiska problem, samt uppfattar orienteringen av kroppen i rummet så att vi exempelvis kan sträcka oss efter föremål.
- **Occipitalloberna:**  
Occipitalloberna är de minsta av storhjärnans lobor och är belägna i den bakre delen av hjärnan, alltså vid bakhuvudet. Occipitalloberna ansvarar för att ta emot och bearbeta information från ögonens synceller.

## 2.3 Historia

På 1700-talet lade Luigi Galvani [6] grunden till elektrofysiologi som handlar om ström i biologiska vävnader. Detta användes sedan för att skapa det galvaniska elementet, som i sin tur ledde till batteriet.

Hans Berger forskade om hjärnan och utvecklade EEG som idag lägger grunden till ett BCI. Med EEG var det möjligt att upptäcka sjukdomar relaterade till hjärnan. Bergers arbete var inspirerat av läkaren Richard Cantons upptäckt av elektriska signaler i hjärnor hos djur, vilket Berger kunde applicera på människohjärnan istället [7]. Begreppet BCI uppkom redan på 70-talet, men på den tiden var EEG begränsat till laboratorier. De senaste 40 åren gick BCI från enkla kommunikationsprogram till sofistikerade applikationer [8].

I juni år 2004 blev Matthew Nagle ett exempel på vad BCI kan åstadkomma då han blev paralyserad från nacken nedåt. Med hjälp av BCI kunde Nagle använda en dator, styra musen, läsa sin e-mail och till viss del rita med endast sin hjärna som kommunikationsmedel. Nagle var den första människan i världen som fick ett BCI implanterat och blev ett levande bevis på att framtiden för teknologin har stor potential [9].

## 2.4 Nutida applikationer

Under 2010-talet ökade tillämpningen av BCI på privatmarknaden, istället för att endast vara ett forskningsmedel. Ett av de allmänt kända företagen som undersöker BCI är Elon Musks Neuralink, som i augusti 2020 demonstrerade hur de kunde avläsa hjärnsignaler från en gris och korrekt förutspå hur dess leder rör på sig. Neuralink är ett exempel på ett kirurgiskt BCI, då de använder sig av ett implantat direkt i hjärnan [10]. Det finns även icke-kirurgiska BCI, och ett exempel på det är Emotiv Epoc X. Denna typ av BCI kan läsa hjärnsignaler med hjälp av elektroder som har

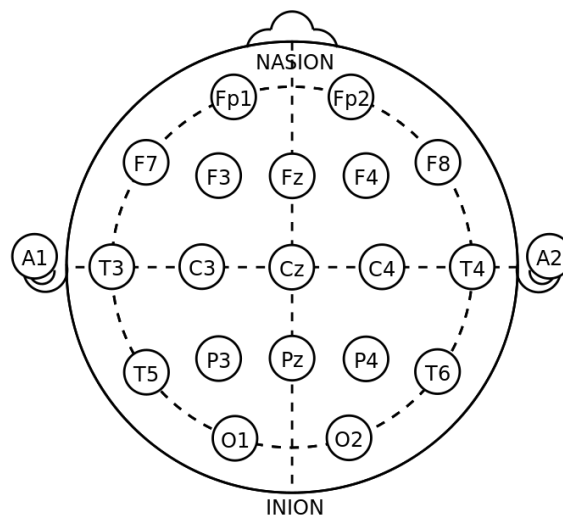


kontakt med skalpen [11].

Ytterligare en kommersiell produkt som finns på marknaden idag är Necomimi, som är ett par kattöron som fästs på huvudet. Detta BCI avläser sedan hjärnvågorna, och reagerar genom att styra och rotera öronen beroende på en persons känslor [12].

## 2.5 10-20 systemet

10-20 är ett internationellt standardiserat system för placering av elektroder på skalpen vid elektroencefalografi (se figur 2.1). Systemet är utvecklat för att standardisera testmetoder mellan olika studier så att resultat kan bli jämförda och återskapade för att verifieras. Systemets placering av elektroder är baserad på de olika regionerna i hjärnan. Varje position i systemet har ett specifikt namn för att identifiera vilken del av hjärnan som elektroden plockar upp data från. Dessa är pre-frontal (pF), frontal (F), parietal (P), occipital (O), och central (C). Varje position har även ett specifikt nummer, där jämna nummer är på högra sidan av huvudet (sett ovanifrån med näsan framåt) och udda nummer på den vänstra [13]. På så sätt kan man avläsa att position O1 går att finna på den vänstra delen av occipitalloben.



Figur 2.1: Bild av 10-20 systemet

## 2.6 Emotiv Epoc X och EmotivPro

Emotiv Epoc X är ett headset som används för att läsa av EEG-data via elektroder på skalpen. För att elektroderna skall få bättre kontakt med skalpen stoppas filt-tussar fuktade med saltlösning in i elektroderna. Epoc har tillgång till fjorton kanaler, noterade via 10-20 systemet. Dessa är: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 och AF4. Epoc sitter centralt på huvudet och elektroderna sitter jämnt fördelade på huvudet. Epoc har noterbart få elektroder över de delar av frontalloben som används för motorisk rörelse.

EmotivPro [11] är den tillhörande programvaran till Epoc och är nödvändig för att kunna interagera med Epoc och dess EEG-avläsningar. När Epoc sätts på huvudet ger EmotivPro ett värde på kontaktkvalitén, som beskriver hur väl EEG-data kan läsas av med Epoc. Det behöver säkerställas att detta värde är på 100%, så att alla elektroder sitter korrekt på skalpen och kan läsa av data. EmotivPro ger tillgång till ett flertal funktioner som bl.a. att se en graf över EEG-data, se olika känslvärden hos en individ, avläsa hur huvudet rör sig och strömma data till andra program. Liveströmmandet av EEG-signaler sker via protokollet LSL (Lab Streaming Layer) som är integrerat i EmotivPro.

# 3

## Teori

I detta kapitel kommer de teoretiska kunskaperna som är nödvändiga för detta arbete att detaljeras.

### 3.1 Signalutlösare

En signalutlösare är en metod för att utlösa en signal i hjärnan, vilken sedan avläses och används som indata till systemet. I detta avsnitt presenteras de vanligaste signalutlösarna för EEG-baserade BCI och en förklaring över hur de fungerar.

#### 3.1.1 P300 - Event Related Potential

P300-ERP är en respons i hjärnan som utlöses av att en person utsätts för stimuli via Oddball paradig. Oddball paradig innebär att en människa blir instruerad att leta efter ett specifikt stimuli men samtidigt blir utsatt för upprepade icke-sökta stimuli. Dessa stimuli brukar ofta bestå av blinkande bokstäver eller siffror men kan även vara auditiva eller känslobaserade. När användaren observerar det sökta stimuli utlöses en P300-respons i hjärnan. Namnet P300 har sitt ursprung i att responsen sker 300 millisekunder efter att det sökta stimuli har observerats.

Detta fenomen används ofta för att hjälpa nervskadade människor att kommunicera via en P300 speller, initialt skapad av Farwell *et al.* [14]. De utlöser en P300-respons genom att lysa upp rader och kolumner av bokstäver i slumpmässig ordning. Användaren fokuserar på en bokstav som det sökta stimuli vilken, när den lyser upp, utlöser en P300-respons. Sedan avläser de när P300-responsen utlöstes och kan därav beräkna vilken bokstav som användaren fokuserade på. Detta tillåter användaren att skriva ord utan att använda något annat än sina ögon.

#### 3.1.2 Motor Imagery

MI (Motor Imagery) är en metod där en individ mentalt föreställer sig eller simulerar en fysisk handling. Genom att en handling föreställs utlöses det signaler i hjärnan som reflekterar den faktiska rörelsen. Dessa signaler kan avläsas och den föreställda rörelsen slutleds. MI kräver oftast träning och testsessioner för att producera bra resultat. MI-data samlas individuellt per användare, och datan kan sedan klassificeras och användas för att kontrollera ett system [15].

### 3.1.3 Steady state visually evoked potential

SSVEP (Steady state visually evoked potential) är en metod för att utlösa en signal med hjälp av visuella stimuli som blinkar i en viss frekvens. Detta kan vara en eller flera olika stimuli. Genom att utsätta användaren för blinkande stimuli med olika frekvenser är det möjligt att identifiera vilket stimuli användaren fokuserar på i stunden, vilket kan göras genom att undersöka en avläst EEG-signal i hjärnan. Denna signal går att avläsa från den occipitala loben och kommer antingen matcha frekvensen av det visuella stimuli eller avläsas som en överton<sup>1</sup> av frekvensen. Signalen i hjärnan kan sedan klassificeras och används för att kontrollera ett system. De olika frekvenserna som användaren kan se på representerar olika handlingar som systemet kan utföra [16].

## 3.2 Canonical Correlation Analysis

CCA är en matematisk funktion som används inom statistik för att hitta korrelationen mellan två mängder,  $X$  och  $Y$ . Korrelationen mellan  $X$  och  $Y$  hittas genom följande formel:

$$r = \frac{a^T \cdot b \cdot \sum_{i=1}^n (x_i - \hat{x})(y_i - \hat{y})}{\sqrt{a^T \cdot \sigma_X^2 \cdot a} \cdot \sqrt{b^T \cdot \sigma_Y^2 \cdot b}}$$

Där  $a$  och  $b$  är linjära kombinationer som maximerar korrelationen.  $\sigma_X^2$  och  $\sigma_Y^2$  är variansen för  $X$  respektive  $Y$ . Formeln resulterar i vektorn  $r$  där varje index representerar den maximala korrelationen i motsvarande kolumnvektors index i  $X$ . Längden  $n$  på  $r$  är  $n = \min(\text{rank}(X), \text{rank}(Y))$ . Korrelationerna  $r_n$  anges:  $r_n \in [-1, 1]$  där  $r_n = 0$  motsvarar ingen korrelation,  $r_n = 1$  motsvarar perfekt korrelation ( $X = kY$ ) och  $r_n = -1$  motsvarar perfekt antikorrelerat ( $X = -kY$ ), där  $k$  är ett positivt reellt tal [17].

## 3.3 Convolutional Neural Network

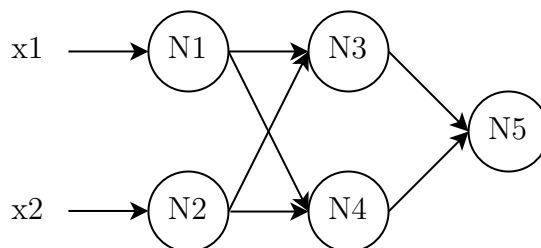
CNN är ett neuralt nätverk som används för att känna igen mönster i data och klassificera datan baserat på mönstren som upptäckts. CNN är en utbyggnad av artificiella neurala nätverk.

### 3.3.1 Artificiella Neurala Nätverk

Artificiella neurala nätverk är algoritmer som inspirerats av hjärnans struktur. Dessa algoritmer tar in data som de tränas på i syfte att känna igen mönster och kunna förutspå utdatan för liknande data. Neurala nätverk är ordnade i flera lager som är kopplade till varandra. Varje lager utgörs av ett antal neuroner. Datat börjar med

<sup>1</sup>En frekvens övertoner utgörs av alla heltalsmultiplar större än 1 av grundtonen till frekvensen  $f$ , alltså 2f, 3f, 4f osv.

att gå igenom inmatningslagret. Denna anknyts i vanliga fall till ett antal så kallade dolda lager. Slutligen når datan utgångslagret som ger slutresultatet [18].



Figur 3.1: Exempel på ett enkelt neuralt nätverk.

### 3.3.2 Artificiell neuron

Den minsta beståndsdel av ett neuralt nätverk är neuronerna. Det är en funktion som inspirerats av den biologiska neuronerna. En neuron är kopplad till neuronerna i lagret innan. Denna koppling har en associerad vikt. Vikten avgör neuronens inflytande vid inmatning i nästa neuron. Denna vikt multipliceras med neuronens utdata som summeras ihop med de andra neuronerna i inmatningslagret och matas in i neuronerna i nästa lager. Varje neuron i nästa lager har en associerad bias som adderas till resultatet. Neuronerna kan beskrivas enligt funktionen

$$z_j = \sum_i x_i w_{ij} + b_j$$

där  $x_i$  benämner indatan från de  $i$  neuronerna i det föregående lagret,  $w_{ij}$  benämner vikten från  $i$  till  $j$ , och  $b_j$  benämner bias-värdet för neuron  $j$ .

Resultatet av uträkningen,  $z_j$ , matar neuronerna in i en aktiveringsfunktion som avgör om neuronerna aktiveras eller inte. Detta beteende efterliknar den biologiska neuronens nervimpuls. Aktiveringen innebär alltså att neuronerna kommer överföra sin utdata vidare till nästa lager. Denna metod upprepas för varje lager och på så sätt propagerar datan genom nätverket i det som kallas *forward propagation*. Neuronerna i utmatningslagret motsvarar varje klass som nätverket försöker förutspå. Neuronerna med högst värde blir den mest betrodda klassen för inmatningsdatan [19].

### 3.3.3 Aktiveringsfunktioner

Aktiveringsfunktioner är viktiga i neurala nätverk därför att de kan introducera icke-linjärhet till nätverket och därmed ge nätverket möjligheten att känna igen mer komplexa mönster [20]. Valet av aktiveringsfunktion skiljer sig bland de olika typerna av lager. De dolda lagren använder vanligtvis samma aktiveringsfunktion. En vanlig aktiveringsfunktion för dolda lager är ReLU. Vanliga aktiveringsfunktioner för utgångslagret är softmax och sigmoid. En viktig egenskap som nätverket kräver är att aktiveringsfunktionerna är deriverbara. Detta har stor betydelse i backpropagation.

ReLU (Rectified Linear Unit) tar emot ett reellt värde och matar ut 0 om värdet är negativt eller själva inmatningsvärdet om detta är positivt. Funktionen lyder

$$f(x) = x^+ = \max(0, x)$$

Softmax tar emot en vektor med reella värden och matar ut en vektor av samma längd men med värden som summeras till 1. Värdena i resultatvektorn brukar tolkas som sannolikheten att en viss vektor tillhör en viss klass. Funktionen kan skrivas som

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad i = 1, 2, \dots, K$$

Sigmoid tar emot ett reellt värde och matar ut ett värde i intervallet 0 till 1. Funktionen har en S-formad kurva. För stora positiva tal genereras ett värde nära 1. För stora negativa tal genereras ett värde nära 0. Sigmoid-funktionen utgörs av följande ekvation:

$$S(x) = \frac{1}{1 + e^{-x}}$$

### 3.3.4 Convolutional layer

För att ett neuralt nätverk ska klassas som ett CNN behöver denna innehålla en eller flera convolutional layers. Convolution i CNN är inte den matematiska operationen convolution utan snarare en nära relaterad operation som kallas korskorrelation. Korskorrelation används för att bedöma likheten mellan två signaler. I signalbehandling används korskorrelation vanligtvis för att hitta mönster i en längre signal. Korskorrelation kan utföras över data i flera dimensioner. Datan i detta arbete kan betraktas som en endimensionell tidsserie bestående av flera kanaler. Korskorrelation utförs genom att glida en kärna (även känt som filter eller mask) över en vektor för att beräkna summan av produkterna på varje plats. Detta är även känt som glidande skalärprodukt. För att beräkna dimensionerna av den resulterande vektorn kan man använda följande formel

$$\frac{W - F + 2P}{S} + 1$$

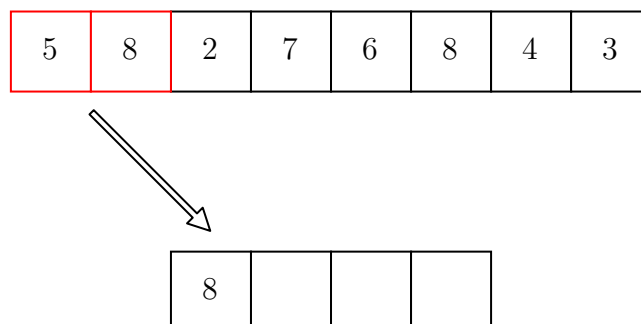
där  $W$  är vektorns storlek i dimensionen man beräknar,  $F$  är kärnans storlek i dimensionen man beräknar,  $P$  är antal vadderingspunkter<sup>2</sup> på vardera sida, och  $S$  benämner antal steg kärnan skiftas varje iteration (detta kallas även stride) [21].

### 3.3.5 Pooling layer

I CNN brukar ett pooling layer vanligtvis följa ett convolutional layer. Max pooling förminskar storleken på inmatningsvektorn genom att glida över en inmatningsvektor och mata ut en mindre vektor med maxvärdet i varje iteration. Max pooling tar

<sup>2</sup>vaddering kan man lägga i inmatningsvektorn för att ge kärnan möjligheten att glida över en större del av vektorn

två parametrar, filtrets storlek och stride. Max pooling används för att reducera vektorns storlek och därmed förminska komplexiteten i nätverket. Pooling leder även till en mer robust modell eftersom mönstren i data kan skifta utan att det påverkar igenkänning av mönster, även när små förvrängningar förekommer i datan [22].



Figur 3.2: Exempel på max pooling med ett filter och stride av storlek två.

### 3.3.6 Träning av nätverket

För att nätverket ska kunna känna igen mönster och förutspå resultatet av okänd data behöver nätverket tränas. Genom träning justeras vikterna i nätverket för att slutligen nå det optimala värdet för vikterna. Träningen är med andra ord ett optimeringsproblem. I detta arbete används optimeraren SGD (Stochastic Gradient Descent). Målet för SGD är att minimera en given förlustfunktion. SGD optimerar vikterna genom att minimera resultatet av förlustfunktionen. Detta innebär ett resultat så nära 0 som möjligt. Under träningens förlopp matas märkt data in i modellen. Denna träningsmetod kallas *supervised learning*. I utmatningslagret ges ett resultat och beroende på vilken aktiveringsfunktion man valt att använda i utmatningslagret får man olika typer av resultat. En typ av resultat är en sannolikhet för vilken klass nätverket tror att datan tillhör. Detta resultat jämförs med det faktiska resultatet av datan för att beräkna avvikelser, alltså förlusten. Samma data matas in flera gånger i ett visst antal epoker. En epok är en körning av hela datauppsättningen. Avvikelsen räknas ut och optimeraren justerar vikterna utifrån avvikelserna i det som kallas *backpropagation*.

Backpropagation är algoritmen som justerar vikterna och biasen för träningsdata efter att den gått igenom nätverket i forward propagation. Backpropagation börjar i utgångslagret och arbetar sig igenom nätverket fram till inmatningslagret [23].

# 4

## Tidigare forskning

I detta kapitel kommer ett antal liknande eller relevanta arbeten att presenteras. Dessa kommer jämföras med eller användas som motivation för ett antal val gjorda till detta arbete.

### 4.1 Liknande arbeten med Snake och ett BCI

Det finns andra liknande arbeten till detta där målet är att kontrollera spelet Snake med ett BCI. Ett sådant arbete är spelet BrainSnake, utvecklat av van Almkerk *et al.* [3] där två spelare använde sina alfavågor för att kontrollera Snake. Detta spel är en mer modern version av Snake där ormen rör sig i 360°, istället för implementationen i detta arbete då ormen endast kan svänga höger och vänster relativt till dess huvud. I van Almkerks spel har ena spelaren kontroll över att svänga till höger och den andra spelaren att svänga till vänster. Kontroll av Snake sker genom att en spelare blundar för att generera alfavågor i sin hjärna, vilket sedan märks av spelet och roterar ormens huvud. På grund av den tid som avläsning av alfavågor kräver pausas spelet medan ormen roteras, och den andra spelaren kan kommunicera när ormen ska sluta svänga. BrainSnake skiljer sig på flera sätt, då arbetet som beskrivs i denna rapport använder SSVEP som signalutlösare. Vad som även skiljer sig är att alla avläsningar sker i realtid. Då ITR (information transfer rate) och noggrannhet av avläsningar är en av de största problemen med BCI skapar realtidsavläsningar en stor skillnad i metodiken och resultaten mellan detta arbete och BrainSnake.

### 4.2 Optimala frekvenser för SSVEP

Tidigare studier är relativt enade när det kommer till vilka frekvenser som är mest effektiva för SSVEP-baserade BCI. En översiktsartikel fann att 10Hz och 16-18Hz gav störst evoked potential [24]. En senare studie, vars primära ändamål var att identifiera de mest passande frekvenserna för signalutlösning, konstaterar att 12-18Hz är mest passande, följt av 8-14Hz och 18-23Hz, och att frekvenser kring 15Hz har störst signal-brusförhållande<sup>1</sup> [25].

Resultaten varierar lite mer om man tittar på skillnaderna med att använda LED och

---

<sup>1</sup>Signal-brusförhållandet innebär den andel värdefull data i signalen jämfört med mängden brus.



LCD. LED verkar vara något effektivare, och har högre ITR [24], men detta skulle kunna vara på grund av att man använt stimuleringsfrekvenser som inte fungerar väl på en LCD-skärm (se [26]). Zhu *et al.* [24] nämner dock att fler tester behöver göras för att en generell slutsats skall kunna nås.

Det har gjorts några studier på vilka frekvenser som passar bäst till LCD-skärmar. Zhu *et al.* konstaterar att man inte kan uppnå frekvenser högre än halva skärmens uppdateringsfrekvens och påstår även att fel uppstår för frekvenser vars perioder inte är multiplar av 2 dividerat med skärmens uppdateringsfrekvens [24]. Anledningen till att dessa fel uppstår är att man behöver två bilder för att något ska kunna blinka, och för en konventionell LCD-skärm tar det  $\frac{2s}{60}$  att presentera 2 på varandra följande bilder, därav är tidsupplösningen  $\frac{2s}{60}$ . Senare visade dock Cecotti *et al.* [27] en metod som gör att man kan simulera frekvenser som har perioder av 1 genom uppdateringsfrekvensen, vilket även testades av Volosyak *et al.* [26].

### 4.3 Vinkelavstånd mellan stimuli

Ett BCI har oftast flera stimuli, och en tidig studie fann att man fick betydligt sämre resultat om vinkelavståndet <sup>2</sup> mellan stimuli var mindre än  $4.5^\circ$  [28]. Även senare studier har fått liknande resultat; Ng *et al.* har i två separata studier noterat att bättre noggrannhet uppnås när avståndet mellan stimuli är större än  $5^\circ$  [29] [30], och Zhang *et al.* fick bäst resultat för  $6^\circ$  till  $10^\circ$  avstånd och rekommenderar därför  $4^\circ$  till  $6^\circ$  avstånd för att få hög noggrannhet men ändå få plats med tillräckligt många stimuli [31].

### 4.4 Storlek och form av stimuli

I ett SSVEP-baserat BCI så kan storleken av stimuli påverka signal-brusförhållandet och klassificeringsnoggrannheten. Duszyk *et al.* undersökte magnituden av den utlösta elektriska potentialen för olika storlekar av stimuli, och fann att den största storleken som testades ( $3.57^\circ$ ) gav bäst resultat [32]. Detta stämmer överens med en tidigare studie av Ng *et al.*, som också fann att större stimuli gav bättre resultat, upp till ca  $7.1^\circ$  [29]. Ng *et al.* nämner dock att  $2^\circ$  är tillräckligt för att uppnå hög noggrannhet.

Det har ännu inte gjorts många undersökningar av hur stimulits form påverkar resultatet, men en studie av Duszyk *et al.* som undersökte de två vanligaste formerna på stimuli för SSVEP, cirkel och kvadrat, fann inte några statistiskt signifikanta skillnader mellan dem, även om de observerade något större magnitud med cirkelformade stimuli [32].

---

<sup>2</sup>Vinkelavståndet beskriver vinkeln mellan två punkter, sett från en specifik plats.

## 4.5 Emotiv Epoc i vardagsituationer

I en studie av Rehab Mahdi Alkhater undersöktes prestandan av en P300 speller med EEG-avläsaren Emotiv Epoc under användning i vardagliga situationer [33]. Målet var bland annat att undersöka den kommersiella användbarheten av Emotiv Epoc, samt att undersöka hur stavningshastigheten (antal avlästa bokstäver per minut) kan ökas. Denna studie visade att Epoc kan användas för att avläsa P300-signalerna och visar liknande resultat som mycket dyrare medicinska EEG-avläsare. Tester i denna studie utfördes i en vardagsituation, till skillnad från andra studier där testpersonen instrueras att sitta stilla och är isolerad från andra stimuli. Istället fick användaren röra sig men blev informerad om att rörelse kan påverka resultaten negativt. Dessutom visades det även att användarens motivation, trötthet och eventuella rörelser kan påverka resultaten negativt och leda till lägre noggrannhet. Dessa resultat har därför motiverat att Epoc kan användas både kommersiellt och eventuellt i en vardagsmiljö.

## 4.6 BCI-illiteracy

Ett eventuellt problem för användningen av BCI är att vissa personer är "BCI-illiterate", vilket beskrivs av Allison *et al.* [34]. Detta fenomen innebär oftast att de inte kan interagera med en specifik typ av BCI, något som tros uppkomma av att vissa individer inte kan framkalla den nödvändiga hjärnaktiviteten som krävs för kontroll. Detta uppkommer av individuella variationer i hur olika personers hjärnor är uppbyggda. Allison beskriver även hur detta kan ske hos en till synes frisk person, och därav noterar de att BCI-illiteracy kan vara medfött. Det har även uppskattats att, för de tre största formerna av signalutlösare (P300, SSVEP och MI), är runt 20% av befolkningen BCI-illiterate [34].

# 5

## Metod

I detta kapitel kommer metodiken för hela arbetet att tydliggöras. Först kommer metodiken för hur hårdvaran (Emotiv Epoc) förbereds före användning att beskrivas. Sedan kommer alla mjukvarukomponenter som samarbetar för att klassificera hjärnsignaler, sköta kommunikation mellan komponenter och visa Snake att detaljeras. Därefter följer en beskrivning av den datainsamling som skett i arbetet. Till sist presenteras en sammanfattning av processen som ledde fram till resultaten.

### 5.1 Hårdvara

Innan en person sätter på sig headsetet (som går att se i 5.1) behöver det förberedas. Filt-tussar som används till Epoc fuktas i en skål med saltlösning för att därefter fästas i elektrodhålen (se 2.5) på headsetet. Tussarna fästs i samtliga kanaler, förutom referenspunkterna P3 och P4, även om data från alla kanaler inte används.



Figur 5.1: Bild av Emotiv Epoc X

Epoc etablerar sedan kontakt med datorn via en Bluetooth-dongel och användaren sätter på sig headsetet. Sedan verifieras att kontaktkvaliteten i EmotivPro är 100%, så att den högsta kvalitén av data avläses. Därefter kontrolleras att avsedd samplingshastighet på 256Hz är aktiverad på headsetet (alternativ av 128Hz eller 256Hz).

Datorn som mjukvaran körs på är en 13"HP Spectre x360 med en fyrcärnig (åtta trådar) processor av modellen Intel i5-8250U. Skärmen har en uppdateringsfrekvens på 59.957Hz och en skärmapplösning på 1920x1080. En extern skärm används även, vilken har en uppdateringsfrekvens på 59.950Hz och en skärmapplösning på 1920x1200. För att öka prestanda av laptopen säkerhetsställs även att laddaren är inkopplad hela tiden.

## 5.2 Mjukvaruimplementation

Under denna rubrik kommer hela mjukvaruimplementationen att presenteras. Denna är uppdelad i fyra separata program: signalutlösare, EmotivPro, databehandling och klassifikation.

Det följande är en överskådlig genomgång av hela programflödet. Först startas en signalutlösare som visar användaren fyra stimuli på skärmen. EEG-data skickas sedan från Epoc med EmotivPro och LSL<sup>1</sup> till databehandling. Datan klassificeras till en instruktion av antingen CCA eller CNN och skickar denna till Snake, och upprepar sedan denna process till spelet tas slut.

Utöver det generella programflödet som används för att spela Snake finns det även alternativa användingssätt av signalutlösare och klassificeringsprocessen. Dessa används för att validera systemet och generera testresultat, som presenteras i 6.

### 5.2.1 Signalutlösare

Signalutlösning sker via ett separat program skrivet i C++ som använder sig av biblioteket GLFW<sup>2</sup>. Detta används för att visa SSVEP-stimuli, i form av vita blinkande hexagoner på en svart bakgrund (se bild 5.2). Hexagonernas storlek motsvarar ett vinkelavstånd på 7.9°, och avståndet mellan två olika stimuli (mätt från de närmaste kanterna) är ca 12.6°. Stimuli är formade som hexagoner då det ökar vinkelavståndet mellan dem något, och enligt tidigare studier finns det inga större skillnader på cirkulära och kvadratiske stimuli (se 4.4). Hexagonerna blinkar i frekvenserna 6.66Hz, 8.57Hz, 12Hz och 15Hz. Med hjälp av GLFW används även VSync (Vertical Synchronization), som synkroniserar mjukvaran med skärmens uppdateringsfrekvens, vilket säkerställer att hexagonerna blinkar i den frekvens som avses, givet att det inte finns andra felkällor. Validering av att dessa hexagoner visas med rätt frekvenser på skärmen gjordes även med ett oscilloskop (se sektion 7.2.3).

### 5.2.2 EmotivPro och Databuffer

Efter signalutlösning avläser Epoc EEG-data från skalpen. EmotivPro är den separata programvaran som används för att skicka datan från Epoc till databehandling. Datan samplas först i Epoc i 2048Hz, sedan används ett dual notchfilter på 50Hz och 60Hz samt ett low passfilter på 64Hz på datan för att ta bort brus<sup>3</sup>. Till sist nedsamplas datan till 256Hz innan den skickas till databehandling, strömmad med LSL.

I databehandling finns det en separat process kallad Databuffer som fångar och sparar ny data från EmotivPro. Här filtreras irrelevanta kanaler bort från datan,

---

<sup>1</sup>Ett bibliotek med funktioner som kan användas för att kommunicera med bl.a Epoc och skapa en ström av data till program i bl.a. Python, Java, C#.

<sup>2</sup>GLFW är ett open-source bibliotek med syfte att göra det enkelt att använda OpenGL, som i sin tur är ett grafikramverk.

<sup>3</sup>Brus är en störning i en signal som kan göra att den information som signalen innehåller blir förorenad.



Figur 5.2: Bild av 4 hexagoner med frekvenserna 6.66 (Vänster), 8.57 (Upp), 12 (Höger) och 15 (Ned).

och kvar blir datan från de kanalerna som sitter närmst occipitalloben, vilka är P7, O1, O2 och P8.

### 5.2.3 Databehandling

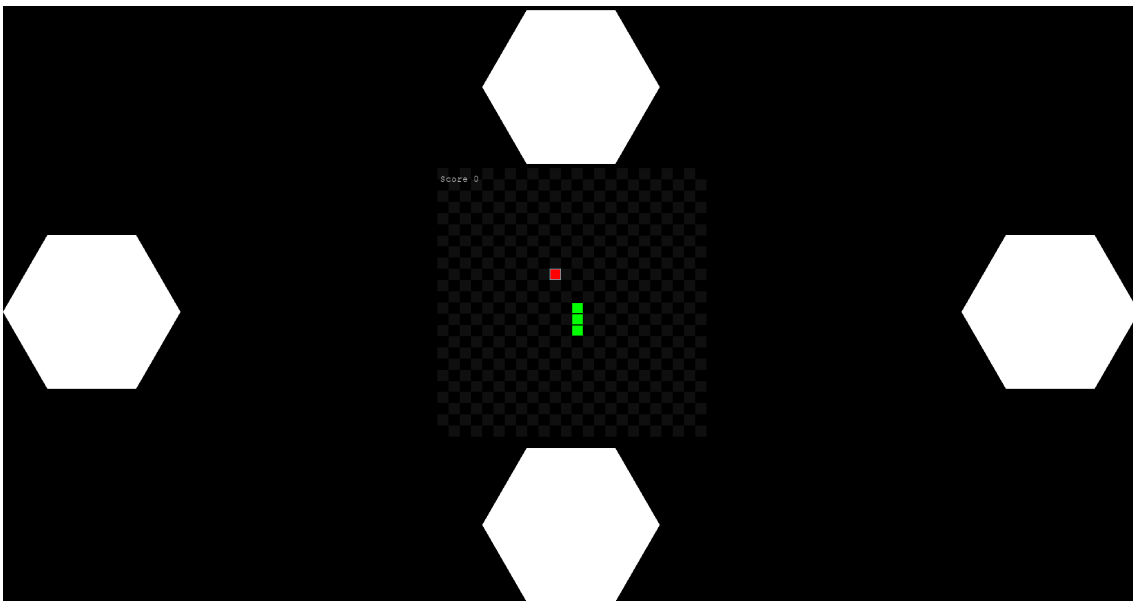
Databehandlingen är kärnan av systemet och är skriven i Python. Denna sköter all kommunikation mellan de olika mjukvarusystemen samt kör Snake. Den består av tre olika processer: Databuffern, som förklarades ovan, Beslutstagaren, som bearbetar och klassificerar data, och Snake-processen som kontrollerar och visar Snake. Beslutstagaren kan antingen använda CCA eller CNN som klassifikationsmetod (vilka beskrivs i sektion 5.2.4 respektive 5.2.5) varav med data från Databuffern räknar ut vilken instruktion som ska utföras. Mängden data som används till Beslutstagaren sköts även på två olika sätt, beroende på ifall klassificering sker diskret eller om flera klassificeringar sker kontinuerligt.

Med diskret klassificering väntar Beslutstagaren tills dess att den samlat upp en sekund värt av data (256 datapunkter med samplingshastighet på 256Hz). Sedan utförs en klassificering och all data rensas. Den riktning ormen ska åka skickas sedan till Snake-processen. Därefter upprepas processen tills spelet avslutas. Med denna metod kommer alla instruktioner vara en sekund fördröjda, då en hel sekund värt av ny data måste avläsas innan en ny instruktion kan klassificeras.

Till skillnad från ovan kan även klassificeringar skötas kontinuerligt av Beslutstagaren. Denna version väntar först tills att den har 256 datapunkter och gör då sin första klassifikation. Vad som skiljer sig från den diskreta versionen är att all data inte rensas. Istället rensas endast de äldsta 64 datapunkterna, varav systemet väntar tills den samlat över 64 nya datapunkter. Då utförs ytterligare en klassificering på dessa 256 datapunkter. Detta innebär (med 256 samplingshastighet) att fyra klassifika-

tioner utförs per sekund. Instruktioner till Snake-processen skickas vid regelbundna intervall, och den instruktion som klassificerats oftast sedan den senaste instruktionen är den som skickas. Längden av detta intervall bestämmer därav hur många instruktioner som kan skickas till Snake-processen per sekund. Till skillnad från den diskreta implementationen kan en datapunkt förekomma i mer än en klassifikation. Denna metod kan då tidigare märka ett frekvensbyte, d.v.s. en ny instruktion till spelet. Dock ökar mängden klassifikationer markant, vilket kan påverka systemets prestanda.

Det sista steget i systemet är att spela Snake med klassificerade instruktioner. Implementationen av Snake är en modifierad version av en existerande implementation, skriven i Python. När spelet körs placeras Snake över signalutlösarna (se bild 5.3) och ormen rör sig en ruta per sekund. Ormen rör sig i en riktning och om den äter ett äpple (röd kvadrat) blir den en ruta längre. Om ormen åker utanför skärmen dyker den upp på andra sidan och om den åker in i sig själv återställs spelet. Instruktioner kommer som den mest troliga frekvensen från Beslutstagaren. Frekvensen översätts till en riktning i spelet, d.v.s. vänster, upp, höger eller ned. Stimuli är även positionerade så att frekvensen som får ormen att svänga till höger är positionerad till höger om spelfönstret. Ifall ormen rör sig horisontellt och tar emot antingen en vänster- eller högersväng, kommer den fortsätta utan förändring. Däremot gör instruktionerna upp och ned att ormen svänger i den riktningen. Vice versa sker när ormen rör sig vertikalt.



Figur 5.3: Bild av Snake och Signalutlösare.

#### 5.2.4 CCA-klassifikation

Den första metoden som används i detta arbete för klassifikation är CCA. Denna kan ta emot data i både diskret och kontinuerlig form från Beslutstagaren. CCA-klassificeringen är implementerad i Matlab och använder sig av funktionen canon-

corr, Matlab's standardimplementation av CCA samt funktionen getHarmonics. Canoncorr tar in argumenten X och Y, där X är en matris innehållande den avlästa EEG-datan från alla kanaler som används i Epoc, där varje kolumnvektor i X är datan från en kanal. Y är en referensmatris som innehåller de förväntade värdena för X när användaren observerar ett visst stimuli, och som fylls med stimulits grundton och dess övertoner. Uppsättningen av övertoner genereras i funktionen getHarmonics och ger följande matris:

$$Y = \begin{bmatrix} \sin(2\pi f \frac{1}{sr}) & \cos(2\pi f \frac{1}{sr}) & \dots & \sin(2\pi f N_h \frac{1}{sr}) & \cos(2\pi f N_h \frac{1}{sr}) \\ \vdots & \vdots & & \vdots & \vdots \\ \sin(2\pi f \frac{N_p}{sr}) & \cos(2\pi f \frac{N_p}{sr}) & \dots & \sin(2\pi f N_h \frac{N_p}{sr}) & \cos(2\pi f N_h \frac{N_p}{sr}) \end{bmatrix}$$

Där  $f$  anger stimulits frekvens,  $sr$  anger samplingshastighet,  $N_h$  anger antalet övertoner och  $N_p$  anger antalet datapunkter. Korrelationerna  $r$  utvinns ifrån canoncorr, och en stark korrelation tyder på en hög sannolikhet att användaren fokuserar på stimulit med frekvensen  $f$ .

CCA anropas en gång för varje stimuli  $F_n$ . Den  $F_n$  som innehåller det största värdet  $r_n$  blir vald som den instruktion som det är troligast att användaren fokuserar på.

$$mF = \max(\text{abs}(F)), F = r_1, \dots, r_n$$

Denna frekvens är sedan den som skickas tillbaka till Beslutstagaren.

### 5.2.5 CNN-klassifikation

Det finns många tekniker för att klassificera EEG-data. Traditionellt har detta gjorts genom användningen av statistiska analysmetoder som CCA. Ett mer modernt tillvägagångssätt är användningen av neurala nätverk. Därför har även en lösning som använder sig av CNN utvecklats.

Följande är en beskrivning av CNN arkitekturen som implementerades med hjälp utav biblioteket Pytorch. Modellen som konstruerades består av två convolution layer samt två fullt anslutna linjära lager. Convolution layeren är uppsatta i block där datan går igenom convolution, batch normalization, aktiveringsfunktionen ReLU, och slutligen maxpooling. Storleken på kärnan för det första och andra convolution layeren är tre och fem. Storleken på maxpooling kärnan är två. Efter de båda convolution layeren matas datan vidare in i det första fullt anslutna linjära lagret följt av en dropout funktion för att förhindra överanpassning av träningsdatan. Slutligen matas datan in i det andra fullt anslutna linjära lagret och klassificering utförs med hjälp av aktiveringsfunktionen softmax. Resultatet av softmax funktionen är den frekvens som modellen tror datan tillhör.

Under träningen av nätverket användes förlustfunktionen crossentropyloss och optimeringsfunktionen stochastic gradient descent. Antalet epoker var tio och learningraten var 0.001.

## 5.3 Datainsamling

I denna rubrik kommer datainsamlingen som skett via systemet att detaljeras. Syftet med denna data är att för att undersöka hela systemets prestanda i resultatkapitlet och träna CNN-modellen. Först presenteras de gemensamma faktorerna hos alla datainsamlingar, sedan förklaras de tre datainsamlingarna på en, flera och ökande frekvenser.

### 5.3.1 Gemensamma faktorer

Försökspersonerna består av gruppmedlemmarna till detta arbete. Samtliga försökspersoner är av manligt kön, mellan 21 och 26 år (med medelålder av 23). Två av sex gruppmedlemmar har försämrad syn och bär glasögon.

Processen av datainsamling består av tre grundsteg. Steg ett är att etablera bra kontakt mellan försökspersonen och Epoc. Detta görs med syfte att försäkra att kontaktkvaliteten inte är en felkälla i inspelning av EEG-data. Steg två är att försökspersonen sitter i en bekväm stol och placerar huvudet 60cm från skärmen. Steg tre är minimering av externa ljuskällor. Alla lampor släcks för att undvika brus som annars uppstår på grund av lampornas blinkfrekvens. Gardiner dras ned för att mörklägga rummet ytterligare.

Försökspersonen försöker att blinka så få gånger som möjligt under inspelningen. Blinkningar skapar mycket brus och gör den inspelade datan mindre användbar.

### 5.3.2 Datainsamling med enskilda frekvenser

Data på enskilda frekvenser spelades först in. I detta fall används bara en hexagon i centrum av skärmen som visar en specifik frekvens. Försökspersonen tittade på denna hexagonen så länge de orkar, med ett maximum på 2-3 minuter. Processen repeterades för 6.66Hz, 8.57Hz, 12Hz och 15Hz. Alla dessa inspelningar sparas som separata filer där all data som spelades in under dessa minuter är loggad.

Denna data används för att träna CNN-modellen då varje datafil representerar en specifik målfrekvens. Det används även för att göra tester på olika noggrannheter för olika mängd data per CCA-klassifikation, vid både kontinuerliga och diskreta klassifikationer.

### 5.3.3 Datainsamling med flera frekvenser i sekvens

Följande är datainsamling på flera frekvenser, vilken används för jämförelser mellan noggrannheten av klassifikationer på olika frekvenser. Frekvenserna visas även i olika tidsintervall, från 1 till 5 sekunder. En klassificering sker varje sekund, alltså oberoende av hur länge testpersonen är ombedd att observera ett stimuli. Detta gör att datan som klassificeras är av samma längd och därmed jämförelsebar.

Insamlingen inleds med en första inspelning med syfte att uppnå en så bra noggrannhet som möjligt. Försökspersonen fokuserar på en stimuli i taget i en cirkulär



ordning där första stimulit är den åt vänster och ordningen rör sig sedan medurs. Försökspersonen observerar varje stimuli i 15 sekunder med en paus på 5 sekunder i bytet mellan stimuli. Denna sekvens upprepas två gånger.

Efter detta görs fem inspelningar till. I resten av inspelningarna ska varje stimuli observeras i slumpmässig ordning, men alla stimuli blir observerade lika många gånger i varje sekvens. Testpersonen observerar stimulit i  $n$  sekunder innan det är dags för en  $n$  sekunders paus, därefter presenteras nästa stimuli som ska observeras. Antalet sekvenser repeteras  $x$  gånger så att varje stimuli får totalt 15 sekunders observeringstid, detta med syftet att alla stimuli ska få lika mycket observationstid oberoende på  $n$ . Första inspelningen sker med  $n = 5$  (observation i fem sekunder) och  $n$  minskar med ett varje inspelning. Eftersom till exempel  $n = 4$  inte är jämnt delbart med 15, ger  $x = 3.75$ , som inte är ett heltal, avrundas  $x$  uppåt till 4. Alltså får  $n = 4$  och  $n = 2$  totalt 16 sekunder observationstid.

Efter inspelningen är färdig kombineras datan till procenthalter per frekvens per tidsintervall, som beskriver noggrannheten i att klassificera en specifik frekvens.

### 5.3.4 Datainsamling med ökande frekvenser

Data spelades även in där frekvensen som försökspersonen observerar varierar. Under inspelningen visades en hexagon i mitten av skärmen som blinkar med en specifik frekvens. Försökspersonen ska fokusera på denna hexagon under hela inspelningen, men pauser används för att de ska få vila ögonen. Denna hexagon byter mellan de fyra frekvenserna (6.66Hz, 8.57Hz, 12Hz, 15Hz). Varje frekvens visas i ökande mängd tid, från en till fem sekund per frekvens. Detta innebär att i första cykeln visas varje frekvens en sekund, i cykeln därpå visas alla frekvenser i två sekunder o.s.v. tills alla frekvenser visas i fem sekunder. Mellan varje tidsökning är det även en paus på 4 sekunder där försökspersonen kan vila ögonen och blinka. Totalt tar hela testet 80 sekunder.

Medan försökspersonen fokuserar på en stimuli körs klassificering på den EEG-data som avläses. Denna klassificering använder kontinuerlig CCA som metod. Samplingshastigheten är 256Hz och en ny klassifikation påbörjas efter att 64 nya datapunkter avlästs (då även de 64 äldsta kastas). Detta innebär att fyra klassifikationer sker per sekund. Den resultatet sparas sedan tillsammans med en tidsstämpel. Denna data kan användas för att iaktta fördröjningen från att en person observerar en frekvens till att den går igenom hela systemet och slutligen klassificeras.

## 5.4 Sammanfattning av metodik och arbetsprocessen

I denna rubrik beskrivs en sammanfattning av hela metodiken som ledde till systemet förklarad i ovanstående rubriker. Utöver de steg som beskrivs i följande stycken utfördes även studier på BCI, EEG, SSVEP, signalteori, CCA, CCN, vid tillfällen där mer kunskap behövdes för att fortsätta arbete.

Det initiala steget till detta arbete var att identifiera och konstruera en signalutlösare för att utlösa EEG-signaler i hjärnan. SSVEP visades vara den bästa kandidaten för att uppnå syftet och därmed implementerades de fyra blinkande hexagoner som SSVEP-stimuli. Denna valdes att implementeras i C++ då biblioteket GLFW gav enkel tillgång till VSync på fönstret med de blinkande hexagonerna.

Med Epoc på huvudet kunde nu EEG-datan innehållande frekvenserna som söktes avläsas. Efterföljande behövs denna data klassificeras, vilket ledde till att klassifikationsmetoderna CCA och CNN konstruerades. Då CNN-modellen behövde data för att tränas påbörjades även datainspelningen på specifika frekvenser. I detta steg testades även många olika faktorer för att avgöra vilken frekvens som hade bäst korrelation från CCA. Den som visade bäst resultat, alltså bäst noggrannhet, på att identifiera frekvenser i EEG-datan var att jämföra den största korrelationen hos alla frekvenser. Den största korrelationen hittades genom att jämföra den högsta korrelationen bland alla frekvensers grundtoner och övertoner.

När väl frekvenserna var valda skapades en implementation av Snake. Först skapades en implementation där endast två frekvenser användes, vilka motsvarade att svänga höger eller vänster relativt ormens huvud. Detta krävde dock en null-action, där ingen frekvens ska väljas då användaren inte avser att svänga. Att avgöra när en sådan situation sker var svårt, så därför skapades istället en implementation med fyra olika instruktioner, en för varje riktning. Denna är den slutgiltiga implementation som har beskrivits tidigare i detta kapitel.

När det slutliga systemet var skapat testades det genom att olika försökspersoner satte sig och försökte spela Snake. Spelandet fortsätter fram tills personen uttryckte att de inte hade kontroll över ormen eller tills ett antal äpplen var uppätta. Dessutom användes systemet för att spela in data, som beskrivet under tidigare rubriker. Denna data användes för att undersöka systemets prestanda och verifiera vilka noggrannheter och responstider som är möjliga. Med både praktiska tester av systemet och utvärdering av datan insamlad presenteras resultaten av detta arbete i nästföljande resultatkapitel.

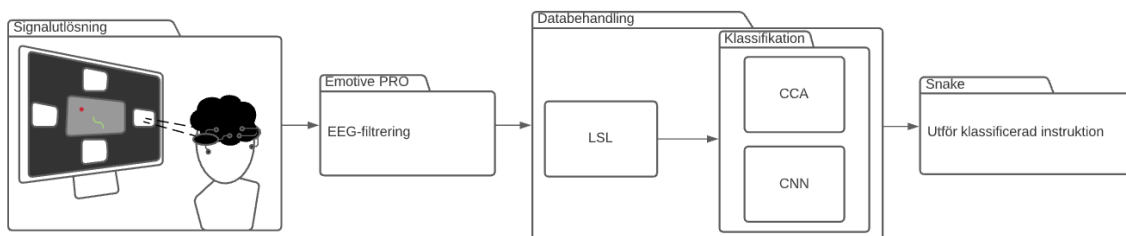
# 6

## Resultat

I detta kapitel kommer resultaten från både inspelad och realtidsdata att redogöras. Först presenteras en översikt av hela signalfödet i systemet, från att en frekvens visas på skärmen till att en instruktion utförs och visas i spelet. Sedan beskrivs applicerbarheten av olika klassifikationsmetoder till detta arbete och hur det är att spela Snake med programvaran. Slutligen presenteras ett antal tester som utförts med den insamlade datan.

### 6.1 Signalfödet genom hela systemet

Från att en blinkning visas på skärmen tills att Snake gör en handling så sker en kedja av händelser. Denna kedja går att se i 6.1 och kommer att förklaras i kronologisk ordning.



Figur 6.1: Övergripande bild på systemet.

Det första som sker är att stimuli visas på skärmen. De frekvenser som används som stimuli är 6.66Hz, 8.57Hz, 12Hz och 15Hz. Ett krav på frekvenserna är att bildskärmens uppdateringsfrekvens måste vara jämnt delbar med vardera stimulus frekvens, detta för att få en jämn blinkning med så lågt jitter<sup>1</sup> som möjligt. Om övertoner används får dessutom ingen frekvens eller dess övertoner vara samma som någon annan frekvens eller dess övertoner (se 3.1.3 för förklaring). Alltså får inte både 5Hz och 10Hz vara valda, då  $2 * 5 = 10$ . Frekvenserna som används valdes då de ligger så nära de mest effektiva frekvenserna som möjligt (se 4.2). 10Hz används inte då den gett sämre resultat i de tester som utförts, möjligen för att den ligger så pass nära 8.57Hz och 12Hz. Gällande form och storlek på stimuli var hexagoner

<sup>1</sup>Jitter innebär att en frekvens inte uppdateras med de jämna intervallen som den bör.

med en storlek som motsvarar ett vinkelavstånd av  $7.9^\circ$  (sett 60cm från skärmen) det slutgiltiga valet. Detta val baserades på tidigare studier, som beskrivs i 4.4.

Programvaran för signalutlösning startas och blinkar, medan användaren sitter rakt framför skärmen, d.v.s. betraktningvinkeln är  $0^\circ$ . Ögonen reagerar på stimulits blinkningar vilket i sin tur registreras av hjärnan, och när hjärnan bearbetar dessa signaler så kan det avläsas i form av elektrisk spänning på skalpen utanför occipitalloben. Epoc används för att läsa av denna spänning i realtid. Samplingshastigheten hos Epoc är 2048Hz, vilken sedan blir nedsamplad till 256Hz. Samplingshastigheten av 256Hz används istället för 128Hz för att öka mängden datapunkter, vilket förbättrar klassifikationsresultaten. Data från Epoc samlas sedan upp av Databuffern. Den data som har samlats klassificeras med en av klassificeringsmetoderna, CCA eller CNN, och resultatet av detta översätts till en handling. Vid observering av stimuli då Epoc samplar från hjärnan kan ett fenomen kallat vikningsdistorsion<sup>2</sup> ske, om samplingshastigheten är mindre än dubbelt så stor som den högsta övertonen. Om detta sker är frekvensen på skärmen inte densamma som den som klassificeras, vilket kan leda till att en felklassificering sker och därmed blir noggrannheten sämre. Den handling som klassificeras skickas till sist till Snake.

## 6.2 Resultaten av att spela Snake

I denna sektion presenteras resultaten från att alla försökspersoner testade och spela spelet. En person kände att denna kunde kontrollera ormen, medan resterande personer upplevde att de hade mindre kontroll. Personen som lyckades kontrollera ormen bäst kunde både köra i samma riktning under en längre period och svänga med hög noggrannhet. Detta ledde till att personen lyckades plocka upp ett flertal äpplen utan att dö. Denna person uttryckte dock att det är svårt att både se spelet och fokusera på ett stimuli samtidigt. Dessutom behövde försökspersonen byta fokus till ett nytt stimuli i förväg för att svänga, vilket ledde till att svängen ibland skedde för sent. Både diskret och kontinuerlig CCA provades för att spela Snake, men ingen väsentlig skillnad märktes i kontrollen, utan båda variationerna visade sig ha bra kontroll.

## 6.3 Klassifikationsresultat av CNN

De bristande resultaten från CNN kommer från att det inte finns tillräckligt med data för att träna modellen sådant att den ska identifiera de olika frekvenserna från EEG-datan. CNN-modellen tränades och testades på två dataset; ett inspelat med en samplingshastighet om 128Hz och ett inspelat med samplingshastigheten 256Hz, där båda använde 0.5 sekunders data per klassifikation. Det ena datasetet innehåller endast frekvenserna 10Hz och 12Hz medan det andra utgörs av frekvenserna 6.66Hz, 8.57Hz, 12Hz och 15Hz. Båda dataseten delades upp i ett tränings- respektive testdataset, varav träningsdatasetet innehöll 80% av den ursprungliga datan

<sup>2</sup>Vikningsdistorsion innebär att de höga frekvenserna "viks nedåt" och blir oskiljbara från lägre frekvenser.

och testdatasetet resterande 20%. Båda dataseten spelades in med en tidig version av signalutlösningsprogrammet som inte använde VSync. Varje dataset användes för att träna och testa modellen fem gånger.

Resultaten från båda dataseten påvisar att CNN-modellen inte lyckas klassificera datan korrekt. Det framgår av tabell 6.1 och 6.2, som visar den genomsnittliga noggrannheten och noggrannheten för varje frekvens, att nätverket gör till synes slumpmässiga klassificeringar. Nätverket ger olika resultat vid varje körning eftersom startvikterna är slumpmässigt inställda och dataseten blandas innan de delas in i tränings- och testdataset.

Tiden det tar att göra en klassifikation är försumbar, då det rör sig om 0.001 sekunder. Flaskhalsen ligger snarare i antalet datapunkter som behöver samlas in för att utföra en klassificering.

Körning	Noggrannhet (%)	10Hz (%)	12Hz (%)
1	50.3	54.1	46.7
2	50.9	48.0	53.7
3	50.1	52.3	47.3
4	51.2	42.6	59.8
5	50.9	47.3	54.4

Tabell 6.1: Resultat med 64 datapunkter (motsvarande 0.5s) per klassifikation. Nätverket tränades med 2856 datamängder och testades på 715 datamängder.

Körning	Noggrannhet (%)	6.66Hz (%)	8.57Hz (%)	12Hz (%)	15Hz (%)
1	21.5	24.8	13.8	27.9	18.9
2	23.3	30.6	13.2	15.4	36.8
3	24.3	17.3	15.4	42.1	22.7
4	22.3	23.8	8.8	29.8	28.8
5	24.9	19.2	16.9	31.2	33.3

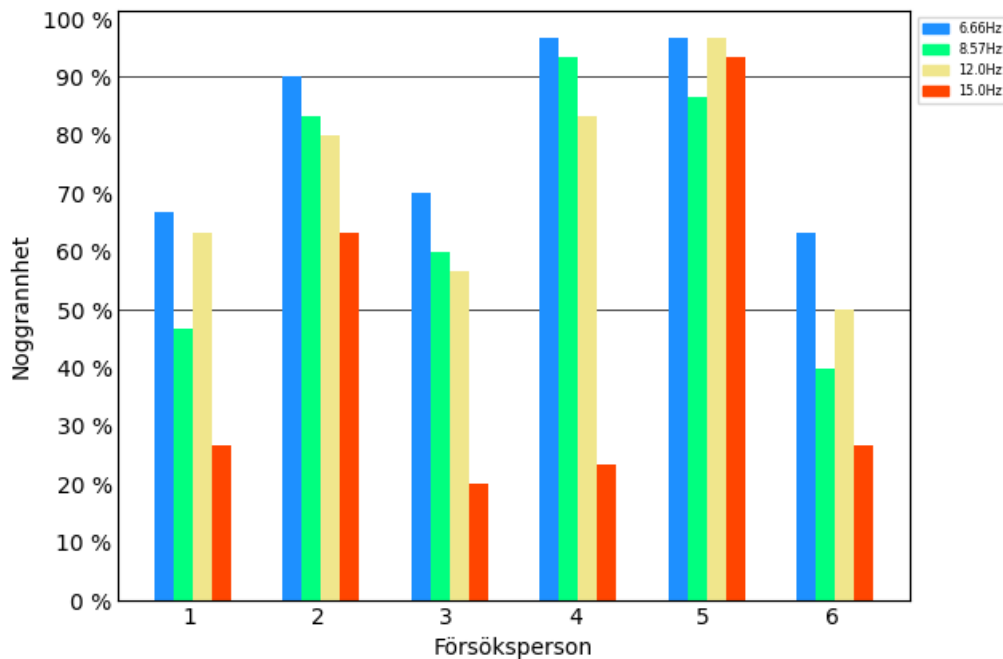
Tabell 6.2: Resultat med 128 datapunkter (motsvarande 0.5s) per klassifikation. Nätverket tränades med 982 datamängder och testades på 246 datamängder.

## 6.4 Klassifikationsresultat med CCA

Under denna rubrik presenteras resultaten av ett antal olika tester på klassifikationsnoggrannhet och responstid av CCA-klassificering, med data som är insamlat enligt 6.4.2. Först presenteras ett noggrannhetstest med data på flera frekvenser, vars inspelning förklaras i 5.3.3. Sedan visas resultat från inspelad data på individuella frekvenser och sist förklaras hur den minimala responstiden av systemet har uppskattats.

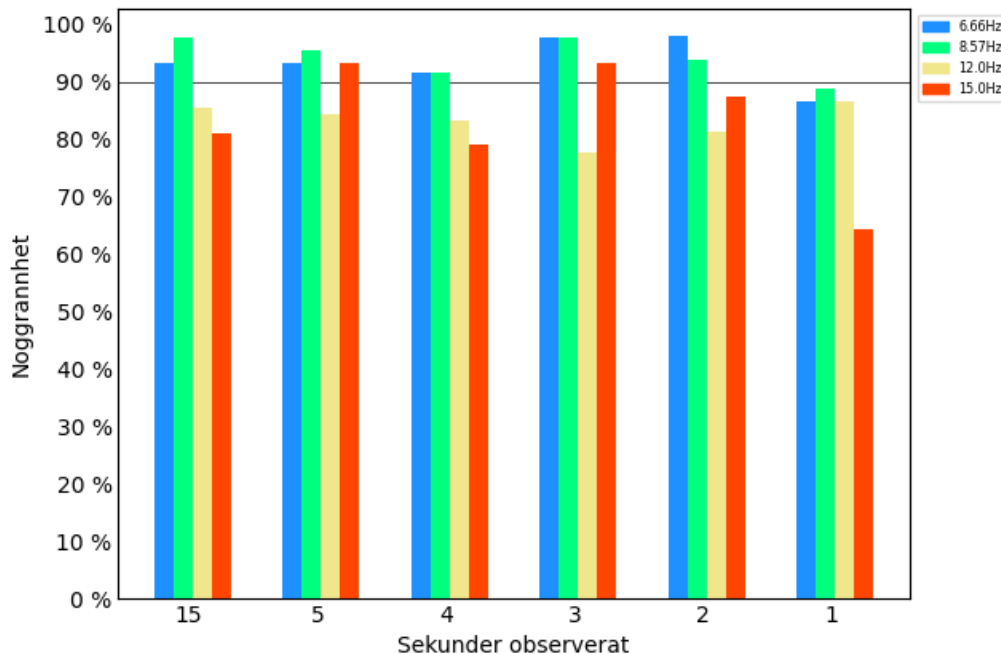
### 6.4.1 Klassifikationsresultat på noggrannhetstest

Testets huvudsyfte är att isolera sambandet mellan observationstid av ett stimuli och en korrekt klassificering. Detta är viktigt för att kunna påvisa att hög noggrannhet kan uppnås trots korta tidsintervall, om datan är representativ nog.



Figur 6.2: Stapeldiagrammet visar samtliga testpersoners resultat vid 15 sekunders observation av stimuli.

15 sekunders-testerna används för att få en uppfattning ifall noggrannheten ökar desto längre försökspersonen fokuserat på stimulit. Resultaten går att se i figur 6.2 där ett streck är draget vid 90% och 50% noggrannhet på y-axeln, för att göra det enklare att se hur testpersonerna presterat. För en klar majoritet av testpersonerna är noggrannheten för 15Hz väldigt låg medan övriga frekvenser har väldigt lika noggrannhet som är betydligt högre än 15Hz. Försöksperson 4 visar goda resultat för samtliga stimuli förutom 15Hz. Försöksperson 2 och 5 utförde även testet två gånger och visade bättre resultat under andra iterationen. Varför upprepade utförande av protokollet ger bättre resultat är oklart, men det kan vara så att träning av att observera stimuli leder till bättre resultat. Detta hade förklarat varför resultaten blir bättre vid senare försök. Vad som också stöder detta påstående är att de flesta testpersoner som utfört protokollet en gång ligger runt 50-60% noggrannhet. Eftersom det observerats en korrelation mellan antal gånger testet utförts och noggrannheten en försökspersonen uppnår så visas resultaten från den person som har utfört testet mest. Detta är för att skapa en bättre bild av de faktiska resultaten som uppnås efter träning. Försökspersonen som utfört testet mest är försöksperson 5, vars resultat går att se i figur 6.3 för alla tidsintervall.



Figur 6.3: Stapeldiagrammet visar resultat från testperson 5.

Ännu en gång finns en linje vid 90% som referens och vi kan se att en majoritet av försöken som är längre än en sekund befinner sig över denna. Vid en sekund är noggrannheten något lägre i resterande observationsperioder, där den huvudsakliga anledningen är att det finns en fördröjning från att försökspersonen fokuserar på stimuli och att potentialen visas i datan (se 6.4.3). För 12Hz så observeras dock inte en markant ökning av noggrannheten då observationstiden ökar från en sekund. Sammanfattningsvis från detta test går det att slutleda att för åtminstone en testperson är noggrannheten fortfarande hög vid lägre tidsintervall. För att komplettera dessa resultat bör även tester utföras på noggrannheten under en sekund, vilket bl.a. beskrivs i nästkommande rubrik.

#### 6.4.2 Klassifikationsresultat på test av individuella frekvenser

Under denna rubrik kommer testresultaten på data inspelad med ett stimuli i taget (se 5.3.2) att presenteras. För dessa test användes CCA som klassifikationsmetod samt ett diskret tidsfönster (varje datapunkt förekommer endast i en klassifikation). Samplingshastigheten för dessa inspelningar var 256Hz och datan spelades in med signalutlösaren på en extern monitor (se sektion 5.1). De fyra frekvenserna klassificerades både med inga och två övertoner per frekvens, för att testa övertonernas påverkan på resultatet. Datat spelades in på försöksperson 2 och 4, där båda spelade in två iterationer på 15 sekunder av data per frekvens.

Resultaten av testerna kan ses i tabell 6.3 och är med 256 datapunkter per klassi-

ficering, vilket representerar 1 sekund av data. Då detta test använder förinspelad data kan mängden data per klassifikation varieras, och mycket långa perioder som 16 sekunder per klassifikation kan testas. Det är värt att notera att bättre resultat kan uppnås ifall mer data används per klassificering, men att detta då hade krävt att Snake väntar på att mer än en sekund värt av data spelas in vid realtidsanvändning. För varje testperson klassificerades den data de hade spelat in, med en slutgiltig noggrannhet på andelen klassifikationer som blev korrekta. Den kombinerade raden på tabellen är resultaten från att all inspelad data används, från båda personerna. Datapunkter blandades inte mellan olika inspelningar (d.v.s. ingen klassificering skedde där olika delar av datan är på olika frekvenser), så att resultaten från klassifikationen kunde jämföras med de faktiska frekvenserna som testpersonerna hade observerat.

Försöksperson	Antal övertoner	Noggrannhet (%)
Person 2	0	45.1
Person 2	2	49.84
Person 4	0	45.3
Person 4	2	55.15
Kombinerad	0	45.07
Kombinerad	2	52.39

Tabell 6.3: Resultat med 256 datapunkter per klassifikation och med olika antal övertoner

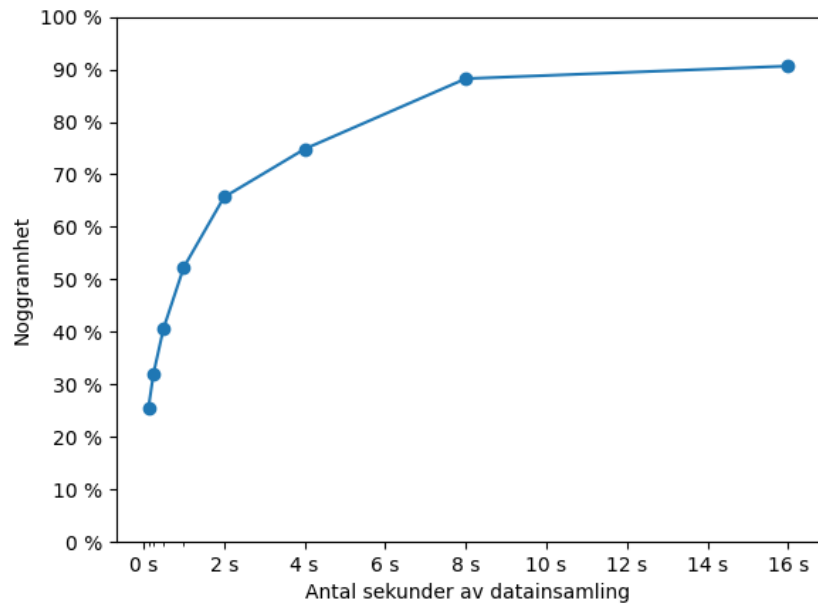
I den ovanstående tabellen syns det att det finns runt 50% chans att rätt frekvens väljs. Detta betyder dock inte att resultatet är slumpmässigt, då det finns fyra olika frekvenser som kan klassificeras. Noterbart är även att ifall inga övertoner används minskar noggrannheten med runt 5 procentenheter på samma data. Detta visar på att användandet av övertoner ger något förbättrade resultat.

Som tidigare nämnts blir noggrannheten bättre med fler datapunkter. I figur 6.4 visas hur mängden datapunkter korrelerar gentemot noggrannhet. Data används från alla frekvenser och inspelningar samt med 2 övertoner, där mängden data i varje klassifikation ändras mellan varje punkt. Mängden data är uttryckt i  $\frac{\text{datapunkter}}{\text{samplingshastighet}}$ , vilket visar hur länge man hade behövt vänta vid liveanvändning för att utföra en handling. Detta betyder att vid 8 sekunder används 2048 datapunkter för klassificering. Noggrannheten är uttryckt i procent. Förhållandet mellan X och Y i figur 6.4 verkar logaritmiskt och därav visas samma punkter med en logaritmisk skala på X-axeln i figur 6.5.

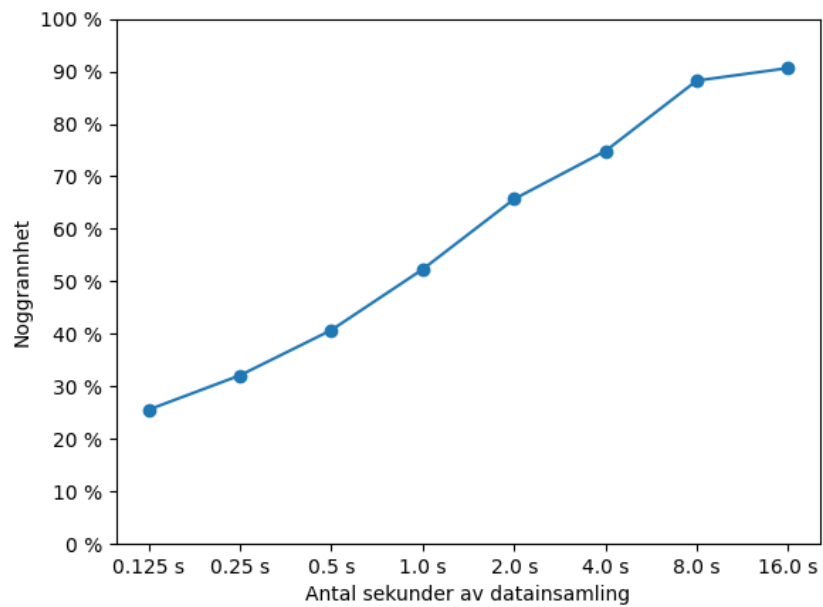
Figur 6.5 visar att det finns ett nära linjärt förhållande mellan noggrannhet och en fördubbling av data. Detta visar att noggrannheten bör öka då mängden data per klassifikation ökar. För att uppnå runt 90% noggrannhet krävdes det mellan 8 och 16 sekunder av data (2048 - 4096 datapunkter) per klassifikation.

Det är dock värt att ha i åtanke hur dessa resultat kommer endast från två försökspersoner och att resultaten hos andra kan variera drastiskt på grund av individuella variationer i hjärnan. Det är även värt att notera att försöksperson 5 inte var med





Figur 6.4: Antal sekunder datainsamling mot noggrannhet i procent



Figur 6.5: Antal sekunder datainsamling (logaritmisk skala) mot noggrannhet i procent

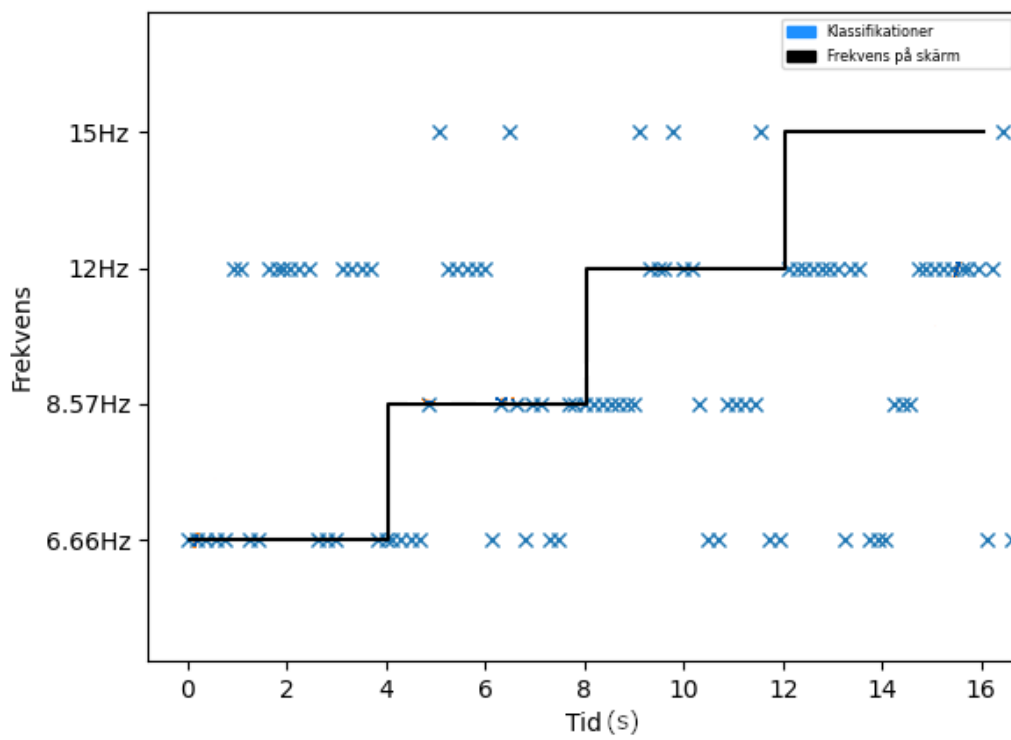
på dessa test, och denna har visat avvikande bra resultat gentemot resterande försökspersoner. Resultaten av detta test är inte att det alltid tar 16 sekunder för att få 90% noggrannhet, utan snarare att noggrannheten förbättras när mer data används.

### 6.4.3 Undersökande av responstid vid byte mellan olika frekvenser

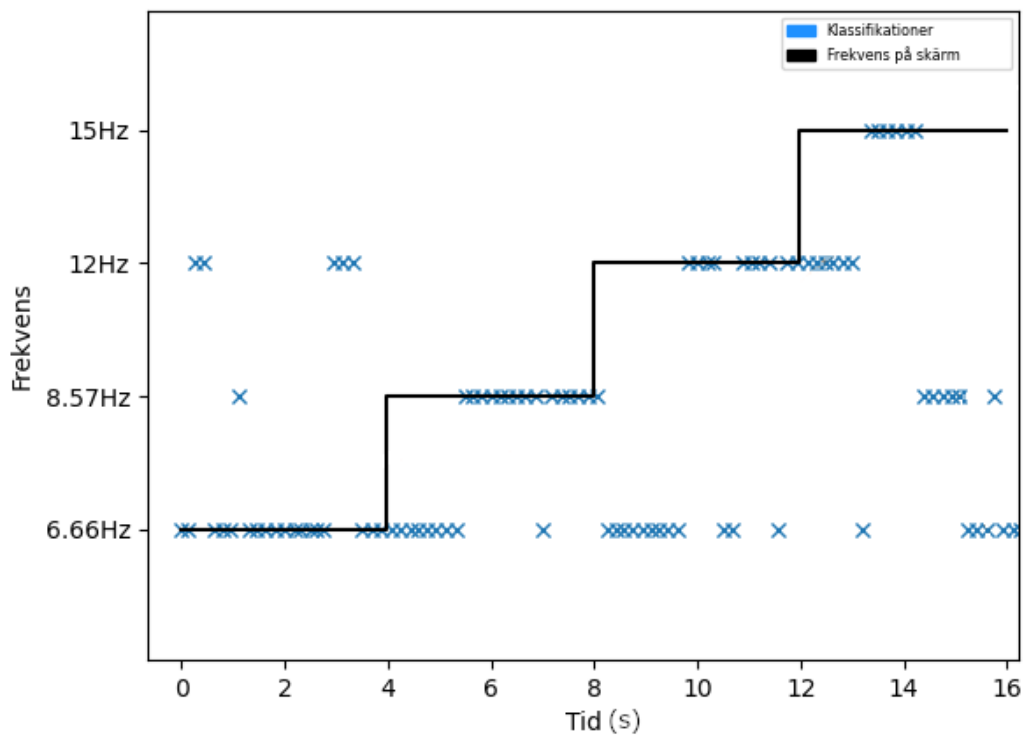
En undersökning av systemets responstid från det att användaren observerar stimuli med en ny frekvens, tills dess att denna klassificeras korrekt har utförts. Under denna rubrik kommer resultaten från denna undersökning att presenteras. Datan som används för denna undersökning beskrivs i sektion 5.3.4. Undersökningen utfördes på försöksperson 2 och 5, där försöksperson 2 gjorde testet tre gånger och försöksperson 5 gjorde testet fem gånger. Denna undersökning inkluderar dock inte ögats rörelse mellan stimuli, som vanligtvis sker under spelande av Snake, vilket kan vara en felkälla.

Resultaten från denna undersökning är presenterade i grafer med förväntade och faktiska klassifikationer till frekvenser, som går att se i figurerna 6.6 och 6.7. I båda dessa figurer visar X-axeln antalet sekunder sedan testet startade och Y-axeln de möjliga frekvenserna som en klassifikation kan resultera i. Den svarta linjen visar stimulits frekvens vid en viss tidpunkt och de blåa kryssen är klassifikationsresultat. Kryss som hamnar ovan eller under en linje är en felklassifikation. Till sist är områdena till höger och vänster om den svarta linjen är de tidsintervall då ingen frekvens visas på skärmen, d.v.s. paus.

Det första som kan slutledas från detta test är att försöksperson 5 hade bättre resultat än den andra försökspersonen. Detta går exempelvis att se i figurerna 6.6 och 6.7 där försöksperson 5 har större andel korrekta klassificeringar. Detta stämmer med resultaten från noggrannhetstesten i sektion 6.4.1, då försöksperson 5 även hade bäst resultat. Huvudresultatet från dessa test är att det verkar finnas en fördröjning från att ett stimuli visas till att motsvarande hjärnaktivitet avläses och en korrekt klassifikation utförs. Detta skapar en förskjutning av alla klassifikationer och går att se i båda figurerna, då det finns ett tomt område på den svarta linjen, mellan att en frekvens visas tills att den får en korrekt klassifikation. Genom att avläsa tidslängden av detta tomma område från alla resultat från detta test har fördröjningen uppskattats till mellan 0.5 och 1.5 sekunder. Detta innebär att varje gång en användare vill byta riktning på ormen finns det runt en sekunds fördröjning innan instruktionen når fram. Denna fördröjning kan alltså anses som den minsta möjliga responstiden på hela systemet. Det är en kombination av fördröjningen från ögat till hjärnan, hjärnan till avläsning och från avläsning till klassifikation. Det måste även samlas tillräckligt med data för att göra en korrekt klassifikation av den nya frekvensen, vilket ger ytterligare fördröjning till systemet. Det är osäkert hur stor andel dessa olika delar utgör av den totala fördröjningen. Det är dock troligt att tiden det tar för att samla tillräckligt med data för att avgöra att en ny frekvens har avlästs, vilket tar runt en sekund, tar upp majoriteten av responstiden. Oavsett vad som orsakar fördröjningarna kan man konstatera att systemets responstid ligger runt 0.5 till 1.5 sekunder.



Figur 6.6: Försöksperson 2 med 4 sekunders frekvenslängd



Figur 6.7: Försöksperson 5 med 4 sekunders frekvenslängd

Det finns dock ingen fördröjning på klassificering av 6.66Hz hos försöksperson 2 i figur 6.6. När frekvenserna visades både kortare och längre tid på skärmen skedde det vid flera tillfällen att 6.66Hz klassificerades direkt när frekvensen visades. Ett möjligt skäl till detta är att 6.66Hz är den frekvens som andra frekvensers felklassifikationer oftast resulterar i, och detta är mer sannolikt än att systemet klassificerar 6.66Hz betydligt snabbare än andra frekvenser. Detta hade även förklarat varför 6.66Hz alltid hade högst noggrannhet i noggrannhetstestet, beskrivet i 6.4.1, då osäkra klassifikationer hade blivit 6.66Hz och därav ökat dess noggrannhet. 15Hz, den högsta frekvensen, hade även sämst noggrannhet vilket sker orelaterat till hur länge frekvensen visades i testet. Noggrannheten för klassifikationer av 15Hz varierar dock mycket mellan försökspersoner; försöksperson 5 uppnådde mycket högre noggrannhet för 15Hz än försöksperson 2. Detta tyder på att det kan finnas individuell variation, och fler tester hade behövt utföras för att tydliggöra detta fenomen.

## 6.5 Sammanfattning av resultat

Sammanfattningsvis är resultaten mycket lovande, och spelresultaten visar en noggrannhet som gör Snake spelbart med CCA som klassifikationsmetod, om än med smärre förhinder. Dock visades CNN inte lika applicerbart till arbetet, då mängden data som krävs för träning inte finns tillgänglig. Det största problemet är responstiden av systemet, och denna responstid uppskattats till runt 0.5-1.5s, vilket försämrar spelbarheten. Detta gäller då användaren först börjar fokusera på ett nytt stimuli, men efter att användaren har tittat på samma stimuli under en period ökar noggrannheten avsevärt, så att Snake kan färdas i rätt riktning tills användaren väljer att svänga. När en person endast fokuserar på ett stimuli har en sekunds avläsning varit tillräckligt för att inte få så pass många felklassificeringar att spelupplevelsen förstörs. Dock har det visats att dessa resultat även varierar mellan olika personer och att vissa kan ha mycket svårare att kontrollera spelet än andra. Det har visats att både kontinuerlig och diskret CCA kan används för att spela spelet, och de har liknande noggrannhet som förbättras något om övertoner används.

I följande kapitel presenteras en diskussion av detta resultatkapitel och även tidigare metodkapitel. Eventuella felmarginaler och förbättringsområden till resultaten kommer att tas upp och diskuteras, ifall arbetet skulle gjorts om eller fortsättas.

# 7

## Diskussion

I detta kapitel kommer resultat och metodik diskuteras. De kommer relateras till arbetets syfte och diskutera förbättringsområden och felmarginaler som kan ha påverkat resultatet. I följande kapitel kommer slutsatser om hela arbetet att presenteras.

### 7.1 Förbättringsområden

Under denna rubrik kommer eventuella förbättringsområden till detta arbete presenteras. Dessa kan vara relevanta ifall arbetet hade gjorts om eller eventuellt utvecklats vidare. Först diskuteras hur användningen av annan hårdvara eller extern mjukvara kan leda till bättre resultat. Därefter beskrivs hur förbättringar eller alternativa implementationer av arbetets programvara potentiellt leder till bättre resultat.

#### 7.1.1 Annan hårdvara och mjukvara

Som visat tidigare i rapporten är avläsningsmöjligheterna hos den EEG-avläsare som använts direkt relaterade till hur bra resultat som kan produceras. Ett tydligt exempel på detta går att se i sektion 6.4.2, där mängden data som var tillgänglig för klassifikation ökade noggrannheten på resultaten. Därför är ett simpelt men effektivt steg till att förbättra resultaten från detta arbete att använda dyrare EEG-avläsare. Dessa har oftast högre samplingshastighet och mindre brus i signalen. Dock är ett relevant diskussionsområde i detta arbete ifall ett BCI kan användas i en vardagsituation för att spela spel, vilket inte är lika relevant ifall priset på avläsaren ökade markant. Att bättre resultat kan uppnås genom dyrare hårdvara har dock visats tidigare och är relativt uppenbart.

I början av arbetet identifierades en annan EEG-avläsare som ett alternativ till Epoc, nämligen en avläsare från OpenBCI [35]. Denna består av ett antal delar som måste monteras, till skillnad från den färdigmonterade Epoc. OpenBCIs mjukvara är även open-source, vilket gör den lättare att få tag i än EmotivPro och dess prenumeration. Rent hårdvarumässigt bör resultaten av OpenBCI inte skilja sig mycket från Epoc, då båda avläsarna har en samplingshastighet runt 250 Hz. Dock har OpenBCI en mer konfigurerbar uppsättning elektrodpositioner, vilket kan leda till att fler elektroder kan placeras över occipitalloben. Detta kan eventuellt förbättra klassifikationsnoggrannheten, då mer data är tillgänglig.

### 7.1.2 Alternativa signalutlösare till SSVEP

I detta arbete valdes SSVEP som signalutlösare. Det har visats att denna metod kan uppnå en mycket kort responstid med hög noggrannhet [36]. Dessutom har SSVEP redan applicerats för att skapa enkla spel [37] och har även visat positiva resultat i samband med Epoc [38]. Dock kan användning av andra signalutlösare eventuellt leda till bättre noggrannhet eller lägre responstid. De två alternativa signalutlösarna var att avläsa en P300 event related potential med en P300 speller eller att använda motor imagery. Det fanns dock problem med båda dessa alternativ.

Det var inte genomförbart att anpassa en P300 speller till detta arbete för att producera bra resultat. Tekniken har visat goda resultat som stöd till handikappade personer genom dess höga noggrannhet och användarvänlighet, men på grund av dess mycket höga svarstid är den inte applicerbar till detta arbete. Tidigare studier har visat att noggrannheten av P300 blir mycket låg för korta tidsintervall och man kan därför slutleda att den inte kan användas för att kontrollera Snake [14][39][40].

Motor Imagery avisades av andra skäl. Epocs placering av elektroder är inte välanpassat för att avläsa det motoriska cortexet av hjärnan. Därav var MI inte ett alternativ för detta arbete, eftersom de signaler som avläses för ett MI-baserat BCI kommer från det motoriska cortexet. Hade en annan avläsare valts, som exempelvis OpenBCIs lösning, hade det varit möjligt att en MI-implementation skulle generera högre noggrannhet på lika eller bättre responstid. Denna implementation skulle dock skilja sig markant från detta arbete, då MI och SSVEP använder olika delar av hjärnan och utlöser signaler på olika sätt.

### 7.1.3 Förbättringar av klassifikationsmetoden

Ett område där förbättring hade haft märkbar påverkan på detta arbete är en förbättrad klassifikationsprocess. Till detta arbete användes primärt CCA för att klassificera EEG-datan. Resultaten från CNN avisades då noggrannheten var låg och det inte fanns tillräckligt med data för att träna modellen. Dock kan förbättring i båda dessa metoder leda till bättre resultat.

Andra arbeten har även lyckats uppnå mycket bra resultat genom att använda CCA med en SSVEP-signalutlösare. Wai *et al.* lyckades få emot 90 till 95% noggrannhet på under en sekunds tidsintervall. Utöver att använda CCA-klassificering använde de även en metod kallad DTW (Dynamic time warping) för att klassificera sin data. De valde den mest lovande kandidaten från båda metoderna för att bestämma vilken frekvens som datan innehöll. Potentiellt kan då resultaten från detta arbete förbättras ifall andra klassificeringsmetoder som DTW samarbetar med CCA för att förbättra noggrannheten. Denna studie utfördes dock på ett förinspelat dataset vilket kan ha påverkat deras resultat. Dessutom hade dataseten data från 9 kanaler över den occipitala loben, till skillnad från detta arbete som endast hade tillgång till 4 kanaler. Dessa faktorer kan ha lett till resultat som är svårare att uppnå i realtid eller som behöver mer avancerad hårdvara (bättre datorer och bättre EEG-avläsare).

Ett annat möjligt förbättringsområde för CCA skulle vara att inte generera matri-

sen som indatan skall korrelera med endast baserat på målfrekvensen, utan istället anpassa den efter varje användare, till exempel genom att först köra ett testprotokoll. Den primära anledningen till att detta eventuellt hade ökat noggrannheten är att det finns individuella variationer i hjärnans uppbyggnad. Därav bör individuellt anpassade målfrekvenser undersökas ifall detta arbete hade fortsätts eller gjorts om.

Ett förbättringsområde av träningen till CNN-klassificeringen hade varit att bekräfta att signalutlösningssprogrammet faktiskt genererade korrekt frekvenser. EEG-datan som spelades in för träning av CNN nätverket använde sig av en tidig version av signalutlösningssprogrammet. Det visades senare att denna inte genererade korrekta frekvenser och ledde till förorenad EEG-data. Gruppen bestämde sig för att inte fortsätta lägga ner mer resurser för vidareutveckling och inspelning av ny data då CCA uppvisade lovande resultat. Tidigare arbeten har dock påvisat att CNN kan åstadkomma noggrannhet på upp emot  $99.86 \pm 0.27\%$  [41].

I arbeten där man utvecklar ett eget signalutlösningssverktyg och därmed behöver verifiera att det funkar är CNN inte det optimala valet av klassifikationsteknik. Anledningen till detta är att CNN (precis som andra neurala nätverk) kräver en stor mängd data den kan tränas på. Detta innebär att för varje justering av signalutlösningssverktyget behöver ny data spelas in för att träna nätverket på nytt. Värt att nämna är även att EEG-datan som CNN-nätverket tränades på spelades in av de försökspersoner som visade minst framgång i försöket att generera användbar EEG-data. För att förbättra CNN-modellen hade därför mer data behövts spelas in med fungerande signalutlösning. Skälet till att detta inte skedde under detta arbete var bristande tid och på grund av den rådande pandemin.

## 7.2 Felkällor

I denna sektion diskuteras kända och potentiella felkällor och hur dessa kan påverka resultaten.

### 7.2.1 Frekvensval

De frekvenser som valdes är baserade på tidigare studier och de krav som nämns i 4.2. Ytterligare test genomfördes av gruppen för att säkerställa att de valda frekvenserna gav bra resultat. Detta betyder inte nödvändigtvis att den kombination som valdes var optimal och det kan mycket väl finnas andra kombinationer som uppnår bättre resultat. Gruppen valde att inte pröva fler kombinationer då det fanns större potential för förbättring inom andra områden av projektet och kombinationen fungerade bra.

### 7.2.2 Variation mellan olika personer

Under test för noggrannhet upptäcktes det att det var stor skillnad mellan deltagarna. Utan någon tidigare träning eller erfarenhet presterade deltagarna olika bra, och detta i sin tur betyder att det går att ha naturlig talang för att använda sy-

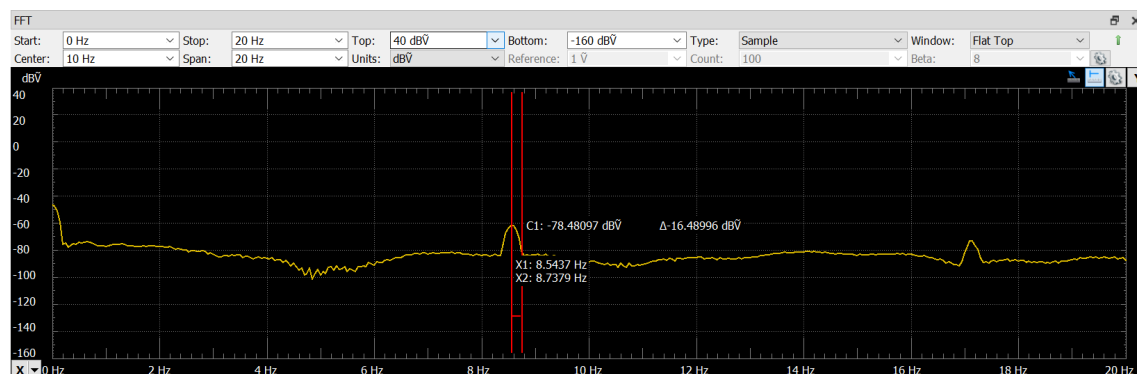
stemet. Detta lägger en grund för att träning eventuellt kan påverka resultatet hos en individ. Vad som förbättras med träning är svårt att säga men det kan vara till exempel en uppfattning om hur långt före en sväng som blicken måste flyttas till ett nytt stimuli, vilket troligtvis hade förbättrat kontrollen över spelet. Det finns också potential för förbättring genom att lära sig hur Snake generellt beter sig, alltså hur ofta rätt handling sker samt lära sig hur länge fokus bör läggas på en hexagon för att uppnå rätt handling.

### 7.2.3 Verifiering av frekvenser på skärm med oscilloskop

För att verifiera att de valda frekvenserna fungerar tillsammans med uppdateringsfrekvensen av skärmen gjordes en mätning med oscilloskop. Detta är viktigt då signalutlösningen från hjärnan kan påverkas ifall en frekvens inte uppdateras med den noggrannhet som är tänkt. VSync har använts för att undvika detta problem. På grund av att detta är en kritisk felmarginal gjordes dock ytterligare test för att säkerställa att detta inte var en felkälla. Resultaten av att använda oscilloskop går att se i följande figurer, för 6.66Hz i 7.1, 8.5Hz i 7.2, 12Hz i 7.3 och 15Hz i 7.4.

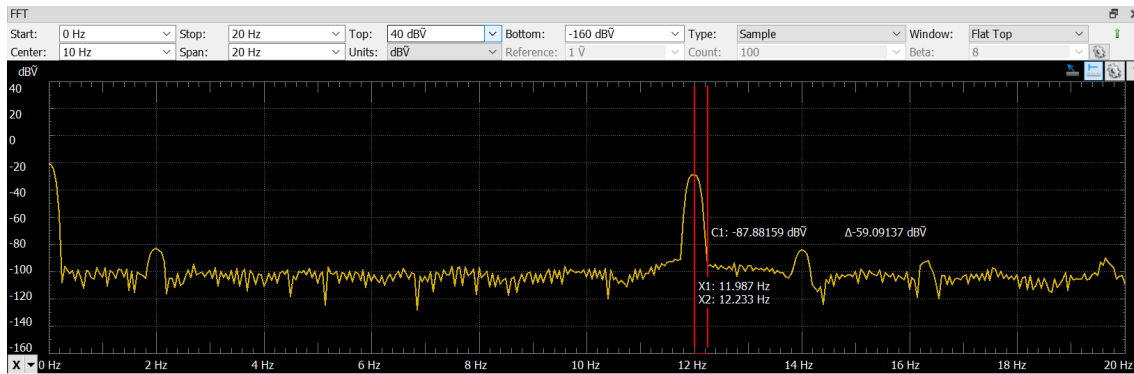


Figur 7.1: Oscilloskop för 6.66Hz stimuli

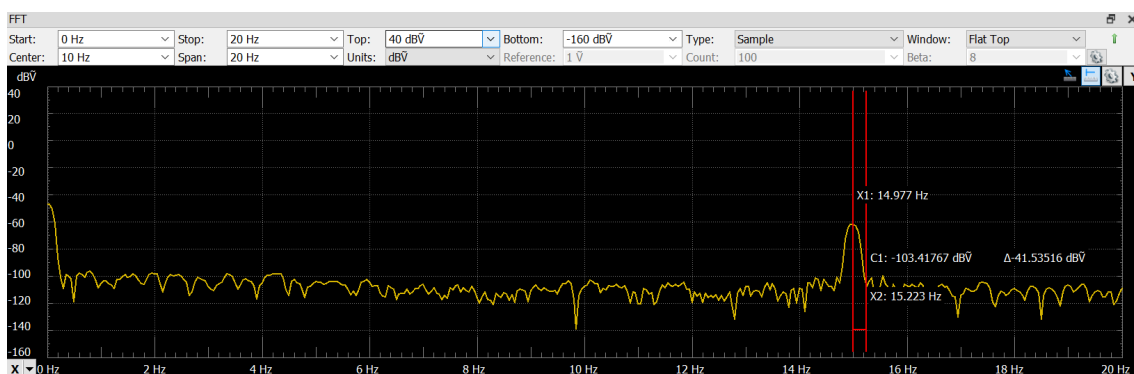


Figur 7.2: Oscilloskop för 8.57Hz stimuli





Figur 7.3: Oscilloskop för 12Hz stimuli



Figur 7.4: Oscilloskop för 15Hz stimuli

Ta figur 7.1 som exempel, som ska uppdateras med en frekvens på 6.66Hz. Enligt figuren gör den det men med ett väldigt lågt jitter. Samtliga frekvenser visar detta, vilket betyder att alla stimuli uppdateras med det jämna intervall som förväntas, och bör därav inte vara en kritisk felkälla.

Figurerna tyder också på en låg differens mellan topp och botten, på cirka 0.2-0.3Hz. Detta innebär att frekvenserna inte ligger för nära varandra och bör därför inte leda till felklassificeringar av anledning att de överlappar varandra.

### 7.3 Etiska aspekter utav BCI teknologi

I detta arbete finns det grund till att diskutera olika etiska aspekter. Det primära målet med arbetet är att kunna upptäcka och klassificera en respons till ett visuellt stimuli. Givet det förväntade utfallet av arbetet skapas ytterligare intresse för BCI. Detta kan leda till både kommersiella och medicinska framgångar som hade haft en positiv effekt på samhället. Å andra sidan finns det även etiska problem värda att diskutera. Det är ytterst osannolikt att direkt applikation av projektresultatet, givet dess avgränsningar, kan leda till etiska dilemman. Dock är det inte en omöjlighet och bör tas i åtanke. Till exempel har det visats att alla individers P300-vågor är relativt

unika. Detta betyder att om gruppen hade valt att analysera och klassificera P300-vågor hade arbetet kunnat anpassas för att identifiera en individ [42], vilket skulle kunna inskränka på personens integritet. En viktig fråga är även diskussionen kring ägandeskap av handlingar. Om en person menar att utföra en handling och BCI:t utför denna, vem är ansvarig för konsekvenserna? Hur hög måste noggrannheten vara för att handlingen (och därmed konsekvenserna) ska kunna knytas till personen? Dessa frågor är viktiga att diskutera och bör tas i åtanke vid fortsatt utveckling av arbetet.

## 7.4 Kompetensbrister och förändringar om arbetet skulle göras om

Gruppen hade liknande kompetenser från start då hälften av medlemmarna studerar det datavetenskapliga programmet på Göteborgs Universitet, och resterande det informationstekniska programmet på Chalmers. Utbildningarna är snarlika och erbjuder liknande kurser vilket innebar att alla gruppmedlemmar var ungefär lika oerfarna av BCI. Detta var inte ett problem utan kanske till och med en fördel då vi alla kunde dela med oss av vad vi lärde oss inom området och var på samma nivå.

I början av projektet fanns det inget standardiserad testprotokoll eller metod för inspelning. Eftersom inspelning och test kräver fysisk närvaro på campus så är det tidskrävande att spela in data och testa implementationen. Inspe­ling av mer data och tester konkurrerar även med varandra om den korta tid som kan spenderas på campus. Under en pandemi försöker också gruppen att ses fysiskt så sällan som möjligt. Att inte ha en standard för inspelning och tester innebar att gruppen fick åka in igen och upprepa testerna om något gått fel. Hade vi gjort detta igen hade vi haft en standard på hur inspelningar och tester ska utföras. Ett problem blev senare att tidigare inspelningar blev svåra att använda då olika inspelningar var i olika format. Idealt hade en struktur diskuterats från början och följts genom hela projektets gång. Detta är dock speciellt svårt när kunskaperna inom området är begränsade från start. Det är inte helt säkert att en standard hade hjälpt eftersom projektet förändrades mycket men det hade sannolikt varit mycket fördelaktigt.

Epoc hade en relativt lång leveransperiod i förhållande till projektlängden vilket skapade fördröjningar för arbetet. Konsekvenserna av förseningen blev att gruppen spenderade mer tid än vad som behövdes på bakomliggande forskning och litteraturstudier. Den tiden hade spenderats bättre på att börja implementera och testa. Hade gruppen gjort arbetet igen så hade vi beställt Epoc tidigare och gjort mer undersökningar kring mjukvaran för Epoc. Vi påbörjade processen tidigt så det hade varit svårt för oss att göra detta bättre under omständigheterna.

## 7.5 Nästa steg

Nästa naturliga steg för projektet hade varit att verifiera potentiella och kända felkällor och åtgärda dessa. Detta bör göras först för att sedan med trygghet kunna

utveckla projektet vidare utan att misstänka tidigare misstag i implementationen. Efter detta bör klassificeringen undersökas ytterligare för att hitta möjligheter för högre noggrannhet. Detta är en bra prioritering då beräkningstiden inte är en flaskhals och det finns utrymme för att utnyttja detta mer.

## 7.6 Sammanfattning av diskussion

Denna sektion sammanfattar de olika ämnen som har diskuterats i detta kapitel.

Gruppen hade väldigt lite erfarenhet av BCI i början av projektet, vilket dock aldrig var ett större problem men som gjorde vissa saker något mer utmanande. Det finns saker gruppen hade gjort annorlunda om projektet skulle göras om, så som bättre struktur kring tester och datainsamling.

Det existerar felkällor som vi är medvetna om och andra som vi misstänker. En sådan potentiell felkälla var att verifiera att de valda frekvenserna visas korrekt på den externa skärmen. Därför gjordes en mätning med oscilloskop. Samtliga resultat visade lågt jitter vilket betyder att frekvenserna fungerar bra och tyder på att signalutlösaren också fungerar som den ska. Gruppen hade undersökt felkällor ytterligare om det fanns mer tid, då det finns potential för bättre resultat. En annan EEG-avläsare hade eventuellt också producerat bättre resultat, men i detta projekt ville gruppen åstadkomma så bra resultat som möjligt med ett billigare headset. Att dyrare headset genererar bättre resultat är inte särskilt intressant; vad som har värde är att undersöka om det är möjligt att med billigare headset uppnå bra resultat.

Det utfördes även undersökningar kring andra signalutlösare men med den utrustning vi valde att arbeta med passade SSVEP bäst. Med annan avläsare och mer tid hade en MI-implementation varit möjlig och hade eventuellt producerat bättre resultat. Genom förbättring av klassifikationsmetoden hade resultaten kunnat förbättras ytterligare. Andra arbeten har använt sig av andra metoder i kombination med CCA för att uppnå bättre resultat, men det bör också tas i åtanke att utrustningen skiljer sig från vår utrustning och därmed också kan vara en faktor till att dessa studier uppnådde bättre resultat. Trots detta tror vi att klassifikationsmetoden är en av de största möjliga förbättringsområdena för detta projekt. Denna förbättringspotential finns också i CNN-modellen.

Nästa och sistkommande kapitel beskriver ifall arbetet uppnått sitt syfte och vilka slutsatser som kan dras från arbetet.

# 8

## Slutsatser

Syftet av arbetet var att undersöka hur hjärnvågor, med hjälp av ett BCI, kan klassificeras som en instruktion till Snake, med målet att minimera responstid och maximera noggrannhet. För att uppnå detta syfte skapades ett system där avläsningsmetoden EEG användes för att läsa elektriska potentialer från hjärnan via skalpen. Dessa elektriska potentialer utlöses från att personen observerade ett stimuli på en skärm i form av blinkande hexagoner vilket aktiverar en respons i hjärnan med samma frekvens som det observerade stimulit. Totalt användes fyra olika frekvenser som tolkas till fyra distinkta instruktioner för Snake. Potentialerna avlästes med hjälp av EEG-avläsaren Emotiv Eloc X, för att sedan klassificeras till en specifik instruktion som användes för att kontrollera ormen i Snake. Klassifikationsmetoderna som användes var antingen CCA eller CNN. För CCA implementerades dessutom både en diskret version, där en sekunds dataavläsning sker per klassifikation, och en kontinuerlig version. I den kontinuerliga versionen sker flera klassifikationer under en sekund, som kombineras till en instruktion baserat på vilken klassifikation som förekom oftast.

I den enklaste tolkningen av syftet, där det enda kravet är att hela systemet är tillräckligt responsivt för att uppfattas som kontrollerbar, kan syftet anses som uppnått. Vad som definieras som kontrollerbart kan variera från person till person, men uttryckt i Snake kan en tolkning vara förmågan att äta ett flertal äpplen utan att förlora och att styrningen inte upplevs som slumpmässig eller motsträvig. I en mer utvecklad tolkning av syftet har även andra faktorer undersökts. Dessa är tiden från att en person försöker skicka en instruktion till Snake tills att den sker i spelet, även känt som responstid. Förhållandet mellan responstid och noggrannhet har examinerats, och faktorer som om det finns ett minimum på responstid har även undersökts.

Till att börja med har det visats att en sekunds avläsning från hjärnan på en samplingshastighet av 256Hz är tillräckligt för att identifiera vilken frekvens en person observerar. Noggrannheten av denna identifikation varierar mellan olika personer, men har visats vara tillräcklig för att minst en person (försöksperson 5) kan kontrollera ormen och äta ett antal äpplen, med en icke-individbaserad korrelationsmatris i CCA. På så sätt har detta arbete resulterat i vad som kan kallas för en spelbar implementation av Snake, vilket var en del av det ursprungliga syftet. Vid tester av kortare tidsintervall för datainsamling, exempelvis en halv sekund, har noggrann-

heten minskat så pass mycket att personen med bäst kontroll uttryckte att denna har viss kontroll men att systemet är svårare att kontrollera och att fler felaktiga handlingar sker. Likartat beskrev de andra personerna som testade systemet att de hade mindre kontroll över systemet och svårare att kontrollera ormen. Å andra sidan, vid ökning av detta tidsintervall till över en sekund fick det smärre förbättringar på noggrannheten hos personen med mest kontroll. Sammanfattningsvis har ett tidsintervall för avläsning på sekund eller högre uppskattats som tillräckligt för att kontrollera systemet.

Gällande förhållandet mellan noggrannhet och responstid är slutsatsen att de är beroende av varandra. Detta innebär att en ökning av responstid även leder till en ökning i noggrannhet. Även motsatsen gäller, då en kort responstid även leder till en låg noggrannhet. Responstiden avgör hur mycket data som samlas in, och ökat antal datapunkter korrelerar med en ökad noggrannhet. Dock gäller detta inte i alla fall, och det finns tillfällen då en ändring i en faktor inte märkbart påverkade den andra vid användning av systemet. Vad som även har hittats är att detta förhållande inte är fast utan varierar från person till person. En person kan uppleva att de har kontroll över spelet vid ett tidsintervall av en sekund, medan en annan känner att handlingarna som utförs är helt slumpmässiga. Skälen till varför detta fenomen uppstår är ännu inte fastslagna, men det går att konstatera att de är konsekventa över flera experiment, då samma personer får liknande resultat. En potentiell förklaring av detta fenomen kan vara att de individer med mindre kontroll är BCI-illiterate, och därav inte kan generera hjärnvågorna nödvändiga för att kontrollera systemet. Ett annat skäl, som tidigare har diskuterats, är att det finns en övningsfaktor till resultaten, där vissa är "bättre" på att använda systemet.

I utvärdering av hela systemet har även en minsta möjliga responstid uppmätts. Denna beskriver fördröjningen från att en person observerar ett nytt stimuli tills att systemet kan utföra en korrekt klassifikation, och har uppskattats till mellan 0.5 - 1.5 sekunder. Denna fördröjning är mest relevant när användaren byter vilket stimuli de fokuserar på, och har inte någon stor påverkan vid fortsatt observation av samma frekvens. Fördröjningen består både av tiden det tar för hjärnan att reagera på det nya stimuli, och tiden det tar för systemet att lyckas klassificera den nya frekvensen. Att avläsa tillräckligt med data för att korrekt klassificera en frekvens har uppmätts ta runt en sekund, vilket förväntas utgöra majoriteten av responstiden.

För att slutleda ovanstående resultat har det som sagt använts olika klassificeringsmetoder, nämligen CCA och CNN. CCA visade sig bäst anpassad, då ingen träning eller stor datamängd krävs för att denna skulle leda till tillräckligt noggranna resultat på en kort tid. CNN å andra sidan klassificerar sämre på grund av för liten mängd träningsdata. För att ett neuralt nätverk ska fungera som klassifikationsmetod till likande arbeten krävs det antingen att data redan finns tillgänglig eller att mycket fokus läggs under arbetet på att spela in användbar data. Dessutom utvärderas resultaten av att skicka data på olika sätt till klassifikationsmetoderna. Antingen genom en bestämd mängd över en sekund eller kontinuerligt där data återanvändes och ett medelvärdesbeslut bestämdes. Skillnaden i noggrannheten mellan dessa metoder är inte stor, och det sker ingen märkbar skillnad på spelbarheten av

Snake mellan metoderna. Kontinuerlig klassificering har visats mer flexibel med hur lång tid som krävs innan en instruktion ska skickas till Snake, men utför även mer beräkningar vilket påverkar prestandan och klassifikationshastigheten.

Sammanfattningsvis var arbetet lyckat och en implementation där Snake kan köras med endast ett BCI som kommunikationsmedel har skapats. Denna implementation är spelbar vid en sekund per instruktion men kan förbättras med bättre noggrannhet vid kortare responstid. För att utveckla arbetet vidare bör potentiella och kända felkällor verifieras och åtgärdas. Sedan bör klassificeringen undersökas ytterligare för att hitta möjligheter för högre noggrannhet. Dessa förbättringar ska förhoppningsvis ta projektet närmre sitt mål av en spelupplevelse helt lik den där Snake spelas med klassisk kontroll.

# Litteratur

- [1] M. O. Krucoff, S. Rahimpour, M. W. Slutzky, V. R. Edgerton och D. A. Turner, "Enhancing Nervous System Recovery through Neurobiologics, Neural Interface Training, and Neurorehabilitation," *Frontiers in Neuroscience*, årg. 10, s. 584, 2016, ISSN: 1662-453X. URL: <https://www.frontiersin.org/article/10.3389/fnins.2016.00584>.
- [2] D. L. Schomer och F. H. L. da Silva, *Niedermeyer's Electroencephalography Basic Principles, Clinical Applications, and Related Fields: Basic Principles, Clinical Applications, and Related Fields*. Oxford, UK: Oxford University Press, nov. 2017, ISBN: 9780190228514. URL: <https://oxfordmedicine.com/view/10.1093/med/9780190228484.001.0001/med-9780190228484>.
- [3] M. van Almkerk, L. Brandl, R. Li, P. Romeo och M. Poel, "BrainSnake: Exploring Mode of Interaction in a Cooperative Multi-brain BCI Game Based on Alpha Activity,"
- [4] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller och T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clinical Neurophysiology*, årg. 113, nr 6, s. 767–791, 2002, ISSN: 1388-2457.
- [5] *Hjärnfonden - Om hjärnan*. URL: <https://www.hjarnfonden.se/om-hjarnan/>, (besöktes: 10.05.2021).
- [6] Piccolino och Marco, "Luigi Galvani and animal electricity: two centuries after the foundation of electrophysiology," *Trends in Neurosciences*, årg. 20, nr 10, s. 443–448, okt. 1997.
- [7] *The Brief History of Brain Computer Interfaces*, <http://www.brainvision.co.uk/blog/2014/04/the-brief-history-of-brain-computer-interfaces/>, Hämtat: 2021-02-07.
- [8] A. Kübler, "The history of BCI: From a vision for the future to real support for personhood in people with locked-in syndrome," *Neuroethics*, årg. 13, nr 2, s. 163–180, juli 2020.
- [9] V. Brower, "When mind meets machine," *EMBO reports*, årg. 6, nr 2, s. 108–110, febr. 2005.
- [10] E. Musk, *Neuralink Progress Update Summer 2020*, Youtube, aug. 2020. URL: <https://www.youtube.com/watch?v=DVvmgjBL74w>.

- 
- [11] *EmotivePros hemsida*. URL: <https://www.emotiv.com/emotivpro/>, (besöktes: 07.05.2021).
- [12] Neurowear, *Necomimi Brain Wave Cat Ears*. URL: <https://www.necomimi.com/>.
- [13] G. Klem, H. Lüders, H. Jasper och C. Elger, "The ten-twenty electrode system of the International Federation. The International Federation of Clinical Neurophysiology.," *Electroencephalography and clinical neurophysiology. Supplement*, årg. 52, s. 3–6, 1999.
- [14] L. Farwell och E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and Clinical Neurophysiology*, årg. 70, nr 6, s. 510–523, 1988.
- [15] H. Cho, M. Ahn, M. Kwon och S. Jun, "A Step-by-Step Tutorial for a Motor Imagery-Based BCI," s. 445–447, jan. 2018.
- [16] R. Singla, "Ssvep-based bcis," *Evolving BCI Therapy: Engaging Brain State Dynamics*, s. 91, 2018.
- [17] M. Nakanishi, Y. Wang, Y.-T. Wang och T.-P. Jung, "A comparison study of canonical correlation analysis based methods for detecting steady-state visual evoked potentials," *PloS one*, årg. 10, nr 10, e0140703, 2015.
- [18] C. Nicholson, "A Beginner's Guide to Neural Networks and Deep Learning," *Pathmind*, URL: <https://wiki.pathmind.com/neural-network>.
- [19] "Artificial Neural Network - Basic Concepts," *tutorialspoint*, URL: [https://www.tutorialspoint.com/artificial\\_neural\\_network/artificial\\_neural\\_network\\_basic\\_concepts.htm](https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_basic_concepts.htm).
- [20] S. Sharma, "Activation Functions in Neural Networks," *Towards Data Science*, 2017. URL: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.
- [21] D. Mishra, "Convolution Vs Correlation," *Towards Data Science*, 2019. URL: <https://towardsdatascience.com/convolution-vs-correlation-af868b6b4fb5>.
- [22] J. Brownlee, "A Gentle Introduction to Pooling Layers for Convolutional Neural Networks," *Machine Learning Mastery*, 2019. URL: <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>.
- [23] V. Bushaev, "How do we 'train' neural networks?" *Towards Data Science*, 2017. URL: <https://towardsdatascience.com/how-do-we-train-neural-networks-edd985562b73>.
- [24] D. Zhu, J. Bieger, G. Garcia Molina och R. M. Aarts, "A survey of stimulation methods used in SSVEP-based BCIs," *Computational intelligence and neuroscience*, årg. 2010, 2010.
- [25] R. Kuś, A. Duszyk, P. Milanowski, M. Łabecki, M. Bierzyńska, Z. Radzikowska, M. Michalska, J. Żygierewicz, P. Suffczyński och P. J. Durka, "On the quantification of SSVEP frequency responses in human EEG in realistic BCI conditions," *PloS one*, årg. 8, nr 10, e77536, 2013.



- [26] I. Volosyak, H. Cecotti och A. Gräser, "Impact of frequency selection on LCD screens for SSVEP based brain-computer interfaces," i *International Work-Conference on Artificial Neural Networks*, Springer, 2009, s. 706–713.
- [27] H. Cecotti, I. Volosyak och A. Gräser, "Reliable visual stimuli on LCD screens for SSVEP based BCI," i *2010 18th European Signal Processing Conference*, IEEE, 2010, s. 919–923.
- [28] S. Fuchs, S. K. Andersen, T. Gruber och M. M. Müller, "Attentional bias of competitive interactions in neuronal networks of early visual processing in the human brain," *NeuroImage*, årg. 41, nr 3, s. 1086–1101, 2008.
- [29] K. B. Ng, A. P. Bradley och R. Cunnington, "Stimulus specificity of a steady-state visual-evoked potential-based brain-computer interface," *Journal of Neural engineering*, årg. 9, nr 3, s. 036008, 2012.
- [30] —, "Effect of competing stimuli on SSVEP-based BCI," i *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2011, s. 6307–6310.
- [31] N. Zhang, Y. Liu, E. Yin, B. Deng, L. Cao, J. Jiang, Z. Zhou och D. Hu, "Retinotopic and topographic analyses with gaze restriction for steady-state visual evoked potentials," *Scientific reports*, årg. 9, nr 1, s. 1–10, 2019.
- [32] A. Duszyk, M. Bierzyńska, Z. Radzikowska, P. Milanowski, R. Kuś, P. Suffczyński, M. Michalska, M. Łabęcki, P. Zwoliński och P. Durka, "Towards an optimization of stimulus parameters for brain-computer interfaces based on steady state visual evoked potentials," *Plos one*, årg. 9, nr 11, e112099, 2014.
- [33] R. Alkhater, "Real-time detection of P300 brain events: brain-computer interfaces for EEG-based communication aids," diss., Auckland University of Technology, 2012.
- [34] B. Z. Allison och C. Neuper, "Could anyone use a BCI?" I *Brain-computer interfaces*, Springer, 2010, s. 35–54.
- [35] *OpenBCIs hemsida*. URL: <https://openbci.com>, (besöktes: 08.05.2021).
- [36] A. A. P. Wai, Y. Zhang, H. Guo, Y. Chi, L. Zhang, X.-S. Hua, S. W. Lee och C. Guan, "Towards a Fast Steady-State Visual Evoked Potentials (SSVEP) Brain-Computer Interface (BCI)," *arXiv preprint arXiv:2002.01171*, 2020.
- [37] E. Filiz och R. B. Arslan, "Design and Implementation of Steady State Visual Evoked Potential Based Brain Computer Interface Video Game," i *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*, 2020, s. 335–338.
- [38] I. Martišius och R. Damaševičius, "A prototype SSVEP based real time BCI gaming system," *Computational intelligence and neuroscience*, årg. 2016, 2016.
- [39] J. L. Sirvent, J. M. Azorín, E. Iáñez, A. Úbeda och E. Fernández, "P300-Based Brain-Computer Interface for Internet Browsing," i *Trends in Practical Applications of Agents and Multiagent Systems*, Y. Demazeau, F. Dignum, J. M. Corchado, J. Bajo, R. Corchuelo, E. Corchado, F. Fernández-Riverola, V. J. Julián, P. Pawlewski och A. Campbell, utg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, s. 615–622, ISBN: 978-3-642-12433-4.

- [40] E. Donchin, K. M. Spencer och R. Wijesinghe, "The mental prosthesis: assessing the speed of a P300-based brain-computer interface," *IEEE Transactions on Rehabilitation Engineering*, årg. 8, nr 2, s. 174–179, 2000.
- [41] N.-S. Kwak, K.-R. Müller och S.-W. Lee, "A convolutional neural network for steady state visual evoked potential classification under ambulatory environment," *PLOS ONE*, årg. 12, e0172578, febr. 2017.
- [42] X. Yang, J. Dai, H. Zhang, B. Wu, Y. Su, W. Chen och X. Zheng, "P300 wave based person identification using LVQ neural network," *Journal of Convergence Information Technology*, årg. 6, nr 3, s. 296–302, 2011.