

Kandidatarbete

EENX16-VT24-32

Tidig karaktärisering av stroke genom videoanalys, maskininlärning och ögonspårning

T. Svensson, E. Ström, S. Ollila, R. Khatiri och J. Westerberg.



8 maj 2024

Chalmers Tekniska Högskola

Abstract

Stroke is a major cause worldwide for both mortality and disability. Swift and precise diagnosis is key to optimize treatment, ensure patient outcome and save lives. This project seeks to utilize the prediction capabilities of machine learning algorithms, by applying it to extracted information from eye-tracking and video analysis, to ultimately predict stroke. The aim and intended result is to potentially improve the efficiency of prehospital care. The National Institutes of Health Stroke Scale (NIHSS) is a scale that is traditionally used to classify a stroke. By digitizing NIHSS and using it as an outline to identify eye-movement abnormalities typically induced by stroke, this can hopefully be achieved.

In this study, due to the scarce availability of patient data, a hybrid dataset was created containing both recorded and synthetic data. The recorded videos (n=99) consisted of healthy individuals simulating gaze palsy in accordance with the NIHSS guidelines. The synthetic data was necessary since certain eye movements, such as when a patient has issues with cranial nerve 3, are very difficult to mimic. Therefore, using digital animation tools (Blender and After Effects), videos with synthetic faces (n=65) were created simulate these symptoms. The recorded dataset were then validated by a stroke specialist. The algorithms chosen to interpret the combined datasets were convolutional neural networks (CNN), deep neural networks (DNN), gated recurrent units (GRU), support vector machines (SVM) and long short-term memory networks (LSTM).

The incorporation of a hybrid dataset expanded the amount of training data, an essential factor to improve the reliability of any machine learning model. LSTM achieved the best overall score in the study demonstrating an 88% accuracy rate, 87.7% sensitivity, 94.1% specificity, and an 86.7% F1-score, underscoring its future potential as a reliable diagnostic tool in prehospital setting.

In conclusion, the findings affirms that using machine learning and video analysis to digitize and classify stroke-induced eye movements has its merits. It could be transformative and be an effective future supplement to traditional stroke assessments. However, additional research and refinement is still required before any future application is viable.

Sammanfattning

Stroke är en ledande orsak till dödlighet och funktionsnedsättning globalt. Snabb och tillförlitlig diagnos är avgörande för att optimera behandlingen, öka patientsäkerheten och rädda liv. Detta projekt syftar till att använda maskininlärningsalgoritmer för att bedöma misstänkta strokefall genom att tillämpa dem på data från ögonspårning genom videoanalys. Målet och det avsedda resultatet är att potentiellt förbättra effektiviteten i prehospital vård. National Institutes of Health Stroke Scale (NIHSS) är en skala som traditionellt har används för att klassificera stroke. Genom att digitalisera NIHSS och använda den som en mall för att identifiera ögonrörelseavvikelser, en vanlig indikator på stroke, hoppas man uppnå detta.

I denna studie, på grund av bristen på patientdata, skapades en hybrid datamängd innehållande både verkliga och syntetiska data. De verkliga videorna (n=99) bestod av friska individer som simulerade blickförlamning i enlighet med NIHSS-riktlinjerna. De syntetiska datan var nödvändig eftersom vissa ögonrörelser, som när en patient har problem med kranialnerv tre, är mycket svår att härma. Därför användes digitala animationsverktyg (Blender och After Effects) för att skapa videor med syntetiska ansiktet (n=65) som simulerar dessa symtom. Den riktiga datamängden validerades sedan av en strokespecialist. De algoritmer som valdes för att tolka de kombinerade datamängderna var convolutional neural networks (CNN), deep neural networks (DNN), gated recurrent units (GRU), support vector machines (SVM) and long short-term memory networks (LSTM).

En hybrid datamängd utökade mängden träningsdata, en avgörande faktor för att förbättra tillförlitligheten hos alla maskininlärningsmodeller. LSTM uppnådde det bästa övergripande resultatet i studien och visade en noggrannhet på 88%, en känslighet på 87,7%, en specificitet på 94,1% och ett F1-värde på 86,7%, vilket understryker dess framtida potential som ett tillförlitligt diagnostiskt verktyg i prehospital miljö.

Sammanfattningsvis visar resultaten att tillämpningen av maskininläring och videoanalys för att digitalisera och klassificera strokeinducerade ögonrörelser erbjuder betydande fördelar. Denna teknik har potential att förändra och fungera som ett effektivt komplement till traditionella metoder för strokebedömning. Innan dessa tekniker kan implementeras i praktiken krävs dock ytterligare forskning och förfining av metoderna.

Förord

Denna rapport är vårt kandidatarbete vid sektionen Elektroteknik på Chalmers Tekniska Högskola och är i samarbete med gruppen Care at Distance - Remote and Prehospital Digital Health.

Vi vill framföra ett varmt tack till alla som stöttat oss i vårt arbete med detta projekt. Ett särskilt tack till vår handledare Hoor Jalo för konstant hjälp, respons och återkoppling, överläkare i neurologi Petra Redfors från Sahlgrenska Universitetssjukhus för återkoppling av inspelade videor samt vår examinator Stefan Candefjord för återkoppling av arbetet.

Teodor Svensson, Eddie Ström, Samuel Ollila, Robin Khatiri och Jacob Westerberg

Göteborg, maj 2024

Förkortningar och akronymer

- AI - Artificial Intelligence
- ASAP - Acute Support Assessment and Prioritizing
- CNN - Convolutional Neural Network
- convGRU - convolutional Gated Recurrent Neural Network
- DNN - Deep Neural Network
- DT - Decision Tree
- EOG - Electrooculogram
- FPS - Frames per second
- FWCI - Field-Weighted citation impact
- GDPR - The General Data Protection Regulation
- GRU - Gated Recurrent Units
- LSTM - Long Short Term Memory
- ML - Machine Learning
- NIHSS - National Institutes of Health Stroke Scale
- RF - Random Forrest
- SVM - Support Vector Machine
- ViPHS - Videostöd i PreHospitala Strokekedjan

Översättningar

- Artificiell intelligens - Artificial intelligence
- Avhopplager - Dropout-layer
- Bild för bild - Frame by frame
- Bilder per sekund - Frames per second
- Dataskyddsförordningen - The General Data Protection Regulation
- Djupfejk - Deepfake
- Dubbelriktad - Bidirectional
- F1-poäng - F1-score
- Förvirringsmatris - Confusion matrix
- Gaussianskt bruslager - Gaussian noise-layer
- Hyperparameteroptimering - Hyperparameter tuning
- K-delad korsvalidering - K-Fold cross validation
- Klassvikter - Class weights
- Klippvärde - Clip value
- Kompaktlager - Dense-layer
- Kärntyp - Kernel
- Maskeringslager - Masking-layer
- Maskininlärning - Machine learning
- Mjukmax - Softmax
- Nyckelbilder - Keyframes
- Satsnormalisering - Batch normalization
- Tillplattningslager - Flatten-layer
- Öppen källkod - Open source code
- Överanpassad - Overfitted

Innehåll

1	Inledning	1
1.1	Syfte	3
2	Avgränsningar	4
3	Samhälleliga och etiska aspekter	5
4	Litteraturstudie	6
4.1	Algoritmer	6
4.2	Syntetisk data	20
5	Analys	21
5.1	Metod	21
5.1.1	Urval av ögonfel och undersökningsmetod	21
5.1.2	Datainsamling	22
5.1.3	Dataanalysering	25
5.1.4	Algoritmer	28
5.1.5	Testning	32
5.2	Resultat	35
5.2.1	Översikt över algoritmers prestanda	43
5.3	Diskussion	44
5.3.1	Algoritmanalys	44
5.3.2	Datagenerering och bearbetning	47
5.3.3	Arbetets betydelse	48
5.3.4	Framtida arbete	49
5.3.5	Påträffade begränsningar	49
6	Slutsats	51

1 Inledning

Stroke är en av de ledande orsakerna till funktionsnedsättning och dödlighet globalt och utgör en betydande hälsoutmaning [1][2]. De skador som uppstår kan visa sig genom en mångfald av symptom vars karaktär och allvarlighetsgrad skiljer sig åt beroende på strokeens natur och den specifika hjärnregion som påverkas [1]. Vanliga symptom omfattar muskelsvaghet, tal- och språkstörningar, plötslig synförlust, samt förvirringstillstånd [1]. Givet strokeens varierande etiologi och symptombild är sjukdomen en komplex sjukdom att diagnostisera och behandla [3]. Trots komplexiteten, har det sedan 1990-talet gjorts betydande framsteg inom området för strokebehandling, tack vare ständigt framväxande teknologier och behandlingsmetoder [4]. Trots den förbättrade behandlingen och diagnostiseringen fortsätter risken för att få en stroke under sin livstid att öka och har sedan 20 år tillbaka ökat med 50% till nu var fjärde person [2]. Den ökade risken understryker faktumet att ännu mer forskning och även bättre behandling fortfarande behövs.

Stroke delas in i två kategorier - ischemisk stroke(85%) och hemorragisk stroke(15%) [5][6]. Strokeinsjuknandet är akut med neurologiska symptom. Den vanligaste typen, ischemisk stroke, uppstår till följd av en blodpropp som hindrar blodflödet i hjärnans kärl, vilket resulterar i syre- och näringsbrist för den påverkade hjärnvävnaden [7]. 24-46% av ischemisk stroke har ocklusion av hjärnans stora kärl vilket är en allvarligare form av ischemisk stroke [8]. Hemorragisk stroke är ett resultat av en bristning av blodkärl i hjärnan som resulterar i en blödning. Denna blödning bidrar till svullnad och ett ökat intrakraniellt tryck som skadar hjärncellerna [7]. Begreppet TIA står för transitorisk ischemisk attack och innefattar strokesymtom som helt går i regress inom 24 timmar, vilket sker till följd av att blodflödet tillfälligt minskar till någon del av hjärnan och kan vara en varningssignal för framtida stroke [5]. Förståelsen av olika strokevarianter är avgörande eftersom behandlingsmetoden och prognosen kan variera kraftigt beroende på strokeens typ och omfattning [1].

Snabb diagnostisering och behandling av stroke har stor betydelse för hur utfallet blir för patienten [9]. För behandling av ischemisk stroke finns två akuta huvudsakliga behandlingsmetoder - trombolys och trombektomi [10]. Gällande trombolys så måste behandling inledas inom 270 minuter (4.5 timmar), helst mycket snabbare, för att kunna ge så bra resultat för patienten som möjligt. Trombolys går ut på att med hjälp av läkemedel lösa upp en blodpropp i ett blodkärl. Ibland behöver man komplettera med en trombektomi, som går ut på att kirurgiskt dra ut blodproppen, och detta kan endast göras på utvalda sjukhus med neurointerventionellt center [10]. I Västra Götalandsregionen kan trombektomi endast göras på Sahlgrenska Universitetssjukhuset. Ifall det finns stora blödningsrisker, eller att tidsfönstret har passerat 4.5 timmar, så kan man istället göra enbart trombektomi. Vid en ocklusion av stora kärl krävs oftast en trombektomi med eller utan föregående trombolysbehandling [11]. För trombektomi är tidsfönstret för behandling längre (upp till 24 timmar). För att initiera tidig behandling är det viktigt med tidig bedömning av stroke så att patienten vid ischemisk stroke hinner få trombolys och eventuellt transporteras till sjukhus där trombektomi kan göras.

Med beaktande av den tidskritiska faktorn i behandlingen av stroke är vikten av bättre diagnostiska metoder stor. Forskningsframsteg visar att ögonrörelser tidigt kan ge indikationer på en stroke, vilket utgör en viktig del i den tidiga upptäckten och behandlingsplaneringen [12]. Specifikt kan en asymmetri i ögonrörelser eller deras reaktioner vara indikativt för neurologiska avvikelser som ofta associeras med stroke, vilket understryker deras värde i tidig bedömning [12]. För bedömningens ändamål analyseras saccadiska ögonrörelser, det vill säga ögats snabba, ryckiga rörelser, för att få fram rörelsens hastighet, precision och reaktionstid [12]. Avvikelser inom dessa parametrar kan inte bara indikera förekomsten av stroke utan även bidra till att lokalisera det drabbade området i hjärnan [13].

Visuell försummelse definieras som en nedsättning i förmågan att uppmärksamma, reagera på, eller orientera sig mot ny eller betydelsefull stimulering. Denna typ av försummelse är vanligt förekommande vid hjärnskador, där patienten möter utmaningar med att rikta uppmärksamhet mot eller reagera på stimuli belägna på ena sidan av synfältet, oftast den motsatta sidan av skadans placering [14]. Ögonspårningsteknik har framgångsrikt använts för att undersöka försummelsefenomenet genom direkt observation av den visuella

utforskningens natur och hur uppmärksamheten fördelas spatiellt [12]. Denna forskning har omfattat en bred variation av stimuli, exempelvis slumpmässigt placerade bokstäver, geometriska former samt vardagsfotografier och videoklipp som testats i olika utforskningssätt och experimentella upplägg. Enligt studier av Karnath tenderar individer med försummelse att under obunden visuell utforskning riktar sin uppmärksamhet cirka 15 grader mot den skadade hjärnhalvan [15]. Deras utforskande beteende karaktäriseras inte bara av en preferens för denna riktning, utan även av upprepade återfokuseringar på redan granskade områden, kortare ögonrörelser och förlängda fixeringstider [12].

Digitalisering av stroke diagnostik är ett relativt nytt område men har visat på goda resultat [16]. Vid identifiering av stroke används så kallade strokeskalor för att gå igenom symptom vid misstänkt stroke. Den vanligaste strokeskalan som används idag är National Institute of Health Stroke Scale (NIHSS) men fler skalor som till exempel Cincinnati Pre-hospital Stroke Scale (CPSS) och Face, Arm, Speech, Time test (FAST-test) finns [17][18][19]. En strokeskala som NIHSS mäter strokesymptomens svårighetsgrad och kartlägger symptom-bilden utifrån förutbestämda kriterier och är det som idag används för att bedöma misstänkt stroke [20]. Delar av strokeskalor har redan lyckats digitaliserats med videoanalys och öppnar upp för vidare forskning och användning inom sjukvården [16][21][22][23]. Ögonrörelser i strokeskalor fokuserar på horisontell rörelse. I horisontell ögonrörelse ingår isolerad blickpares, partiell blickpares och komplett blickpares [24]. Isolerad blickpares är när endast en specifik typ av rörelse är nedsatt, Isolerad blickpares är symptom som tyder på fel i kranialnerv 6 (abducens), 4 (trochlearis) eller 3 (oculomotorius) som alla påverkar ögonrörelser på olika sätt [25]. Partiell blickpares är när flera typer av rörelser är nedsatta och där paresen kan övervinnas själv eller med Doll's eye test. Komplet blickpares kan inte övervinnas.

För att diagnostisera stroke hos en medvetlös eller icke-responsiv patient kan ett Doll's eye-test, även känt som oculocephalic reflex, utföras. Testet innebär att patientens ögonlock lyfts och huvudet snabbt men varsamt vrids från sida till sida. Under testet observeras patientens ögon för att se ifall de vrider sig i motsatt riktning från huvudvridningen, vilket innebär att ögonen behåller en framåtriktad position [26]. Doll's eye-testet är användbart för att upptäcka skador och dysfunktioner i hjärnstammen som kan vara resultatet av en stroke [27]. Ett annat strokesymptom som hänger ihop med ögonrörelser är nyastagmus som kännetecknas av att ögonen involjänt gör små repetitiva rörelser [28][29]. Nyastagmus ingår inte i strokeskalan NIHSS.

Redan idag tas initiativ för att öka digitalisering i syfte att hjälpa sjukvården. Exempelvis finns projektet Videostöd i den PreHospitala Strokekedjan (ViPHS) som har installerat flertalet kameror i 12st ambulanser runt om i Västra Götalandsregionen för att kunna ha videokonsultation med specialister på sjukhuset [30]. Nutida tekniker, inklusive maskininlärning (ML) och artificiell intelligens (AI), har visat sig vara effektiva i att identifiera individer som tidigare drabbats av stroke genom att analysera saccadiska rörelser och ögonens fixeringstid [31]. En specifik studie rapporterar att med optimala förhållanden och den mest effektiva algoritmen som testades, Random Forrest (RF), kan en träffsäkerhet på upp till 88,45% uppnås, vilket belyser potentialen hos ML som ett verktyg i stroke-bedömning [31].

Användning av syntetisk data är ett relativt nytt område inom sjukvården som potentiellt kan revolutionera forskning och utveckling av medicinska lösningar [32]. Genom att generera simuleringar av patientdata kan forskare utveckla och experimentera och testa nya metoder och behandlingar. En studie har skapat syntetisk data för 19st olika dataset inom hälsa och sjukvård och jämfört maskininlärningsmetoder som tränats med den riktiga datan kontra den syntetiska datan [33]. Studien [33] visade att användningen av syntetisk data endast gav något sämre resultat än vid träning av riktig data med en skillnad i noggrannhet (accuracy) på 6-7 procentenheter. Däremot krävs det stor hänsyn till potentiella bias som kan förekomma vid skapande och användande av syntetisk data [32].

Inom ramen för initiativet Acute Support Assessment and Prioritizing (ASAP) ligger detta projekt i skärningspunkten mellan teknisk innovation och medicinsk forskning för att förbättra prehospital bedömning av stroke [34]. Projektet bygger på föregående forskningsresultat där videoanalys och ML har använts för att

analysera olika aspekter av NIHSS, vilket har visat på potentiella implementeringsvägar [16]. Med tanke på strokeans komplexitet och det tidskritiska behovet av intervention, framstår integrationen av AI och ML som ett betydande steg framåt för att stödja beslutsfattandet i prehospitala miljöer [35]. Ur ett akademiskt perspektiv är detta projekt inte bara en teknisk utmaning utan också en möjlighet att fördjupa förståelsen för samspelet mellan teknik och medicin. Det står i frontlinjen för forskning och utveckling inom strokevård och erbjuder en unik chans att undersöka hur tekniska innovationer, som AI och ML, kan integreras i kliniska flöden för att förbättra bedömningsprocesser. Projektet ger möjlighet att analysera, utvärdera och finjustera dessa teknologiska lösningar i verkliga scenarier, vilket bidrar till den akademiska kunskapsbasen och potentiellt förändrar framtida strokevård.

1.1 Syfte

Syftet med projektet "Tidig karaktärisering av stroke genom videoanalys, maskininlärning och ögonspårning" är att utveckla och förbättra metoder för tidig bedömning och karaktärisering av stroke i prehospitala miljöer, med särskild fokus på användning av videoanalys och ML-tekniker för att analysera ögonrörelser. Detta projekt följer NIHSS för att möjliggöra en snabbare och säkrare bedömning av stroke.

Det centrala målet är att utveckla en klassificeringsmodell som med hög noggrannhet och precision kan hjälpa med att bedöma och skilja mellan närvaron, frånvaron och graden av stroke. Inom ramen av detta projekt görs bedömningen genom att identifiera specifika mönster i ögonrörelser som är relevanta för strokebedömning. Processen för att uppnå detta mål inkluderar inspelning och skapande av verklig och syntetisk videodata som efterliknar strokepåverkade ögonrörelser.

För att tillgodose detta mål, kommer projektet att genomföra en litteraturstudie för att identifiera och utvärdera befintliga system och modeller som används idag för liknande ändamål. Genom användning av den manuella NIHSS-bedömningen, som för närvarande används i klinisk praxis, avses att klarlägga vilka typer av ögonrörelser som är mest relevanta och hur teknik kan användas för att effektivisera och förbättra diagnosprocessen.

Projektet strävar efter att komplettera och förbättra den prehospitala vården genom att potentiellt minska fördröjningar i behandling och därmed förbättra utfall för strokepatienter.

Slutgiltigen är problemställningen uppdelad i huvudsakliga två frågeställningar:

1. Är det möjligt att med maskininlärning digitalisera och analysera ögonrörelser för att bedöma stroke?
2. Är det möjligt att syntetiskt generera relevant data som kan användas för att träna och testa det framtagna systemet i att effektivt bedöma stroke?

2 Avgränsningar

Kandidatarbetets avgränsningar kan summeras i fyra punkter:

- Fokus på ögonrörelse
- Efterliknelse av stroke
- Fokus på programvara
- Etiska aspekter till följd av videokameror

Kandidatarbetet har endast fokuserat på hur ögat rör sig under akutfasen av stroke, och satt detta i kontrast mot frisk ögonrörelse. Stroke har en bred symtombild [36]. Alla karaktärsdrag kommer inte att tillgodoses som variabler i algoritmen/algoritmer som tas fram. Motiveringen bakom exkluderingen är att ett tidigare kandidatarbete var redan utfört som fokuserade på digitalisering och detektion av symptomen: arm- och benpares, ataxi, facialpares samt dysartri. Genom denna begränsning och med det tidigare arbetet som grund, så kan en mer djupgående analys göras utifrån det givna tidsspannet.

Projektet stog inför utmaningen att skapa anpassade algoritmer utan tillgång till stroke-data från riktiga patienter. Anledningen är de etiska godkännanden som krävs och utmaningarna med att inte störa sjukvårdspersonalens arbete eller försena behandlingen för strokepatienter i kritiska tillstånd. Istället efterliknade deltagare ögonrörelser som är förknippade med stroke. Efterlikningen skedde med god förberedelse och hjälpmedel som videodata. Även syntetisk data skapades där samma ögonrörelser efterliknades tillsammans med ögonrörelser som friska deltagare ej kan efterlikna. Videodatan godkändes av experter från Sahlgrenska som tydligt instruerade om hur ögonsymtom såg ut vid stroke. Efter det tränades och evaluerades ett flertal olika algoritmer på den framtagna datamängden. Det är således inte heller projektets avsikt att ta hänsyn till patienters medicinska eller biologiska data, som exempelvis blodtryck, ärftliga anlag eller sjukdomshistorik. Anledningen till att projektet inte använde sig av riktiga strokepatienter är för att detta skulle försvåra arbetet oerhört mycket. Då måste information samlas in efter instruktioner i ambulanser eller akuten. När individer spelas in i situationer som kränker deras integritet uppstår etiska dilemman. Dessa situationer kräver också juridiskt dokumentationsarbete. Åtgärder måste också göras för att försäkra att all data samlas säkert. Man skulle även skapa ett till distraktionsmoment för exempelvis ambulanspersonal eller läkare på akuten i en redan pressad och kritisk situation.

Kandidatarbetet hanterar endast mjukvara och inte hårdvara. Vad detta innebär är att det kommer konstrueras algoritmer som används för bedömningen. Däremot har inte arbetet hanterat val av hårdvara, som exempelvis val av kameror eller inbyggda system. Kandidatarbetet har inte heller hanterat hur denna utrustning ska hanteras på det mest effektiva sättet inom prehospitäl vård. Den integritetskränkande miljön som kan skapas på grund av kameror inom den prehospitäl vård är något som inte heller försökts förebyggas. Avgränsningens syfte är att lägga mer fokus på den faktiska mjukvaran. Om man återgår till kandidatarbetet syfte så handlar syftet primärt om att undersöka de framtagna algoritmernas effektivitet, och huruvida lovande för applicering kan tänkas bli. Kandidatarbetet ska inte tillgodose de etiska problemen i sin utvecklingsprocess som hade kunnat förhindra aspekter inom arbetet. De problem som följde med implementeringen tas upp och diskuteras, men undersöks inte ytterligare. Summan blir alltså att etiska problem av systemet tas upp, men vid detta tidiga stadiet av systemet så har projektet inte fokuserat på att överkomma dessa etiska problem. Detta projekt är ett bevis av koncept och ytterligare arbete och studier kan komma då de etiska aspekterna tas till hänsyn.

3 Samhälleliga och etiska aspekter

Projektet fokuserar på användningen av videoanalys, ML och ögonspårning för att karakterisera stroke. Det kommer då upp ett flertal viktiga samhälleliga och etiska aspekter som behöver tas i hänsyn till. Ett centralt område är patientens integritet och autonomi. Vid verklig implementering av projektet behöver procedurer införas för att få samtycke där patienter fullständigt informeras om hur deras data kommer att användas, lagras och skyddas. Denna process måste ge patienterna möjlighet att avstå utan några negativa konsekvenser. Med detta är det även viktigt att säkerställa att den insamlade datan anonymiseras för att skydda patienternas identitet och personliga uppgifter som enbart relevant personal har tillgång till. En aspekt att ta hänsyn till är att denna process kan komma med svårigheter i och med att patienter i dessa fall kan vara inkapabla till att ge sitt samtycke på grund av deras stroke, alternativt att de ej överlever.

Hantering av sådan känslig data kräver att åtgärder görs för att kunna skydda datan mot åtkomst av icke relevant personal samt brott mot datasekretess. Det är därför nödvändigt att följa dataskyddsförordningar som General Data Protection Regulation (GDPR) för att garantera att hanteringen av patientdata uppfyller standarder för integritet och sekretess [37].

Genom att förbättra möjligheterna för diagnostik och behandling av stroke med hjälp av ny teknologi framtiden i projektet, kan potentiellt vårdkvaliteten förbättras. Med den nya teknologin finns potential för positiva effekter för sjukvårdens arbetsbelastning, tidigare diagnostik, snabbare behandling och i slutändan bättre livskvalitet för patienter. Denna positiva inverkan kan då lägga mindre press på vårdpersonal och de kan allokera mer tid och fokus på andra aspekter som anses vara mer krävande. Här är det däremot nödvändigt att säkerställa att tekniken är tillförlitlig och inte leder till felaktig bedömning av stroke. Även teknologisk överbelastning och minskad interaktion mellan patient och vårdpersonal är något som behöver tas i hänsyn till. Ytterligare en viktig aspekt av forskningen är att säkerställa att teknologin som utvecklas är tillgänglig och användbar för en bred patientgrupp. Med denna breda tillgänglighet behöver olika socioekonomiska, geografiska och kulturella faktorer som kan påverka tillgång och användbarhet av tekniken tas hänsyn till.

Vid utformning av datainsamlandet är det viktigt att alla deltagare i projektet ger sitt samtycke till att bli inspelade och att inspelningarna kommer att sparas och bearbetas. Projektet kräver att deltagare behöver imitera vissa stroke-symptom. Deltagarna bör även vara fullt medvetna om syftet med forskningen samt ha rätt att när som helst avbryta sitt deltagande utan några negativa konsekvenser. Metoderna för stroke-imitationen ska även planeras och genomföras på ett sådant sätt som minimerar fysisk belastning då det är viktigt att säkerställa att forskningsmetoderna inte innehåller onödigt obehag. Vid stroke efterlikningen är det också avgörande att samarbeta med experter inom stroke för att säkerställa att metoder och utföranden är relevanta och korrekta.

Genom användning av videobaserad analys för att karakterisera stroke finns det möjlighet till att projektet bidra till en mer exakt och snabbare diagnos, vilket är avgörande för effektiv behandling. Resultatet kan potentiellt hjälpa till att minska tiden från att symptom av stroke visas till behandling. Projektet kan eventuellt ge värdefull insikt och användbar data för framtida forskning inom stroke och annan neurologisk diagnostik. En viktig fråga med användning av ML är risken för bias i algoritmerna vilket kan leda till felaktig bedömning. Det är därför nödvändigt att vara noggrann och övervaka samt att testa samtliga använda algoritmer. Algoritmerna behöver då testas med data tagna från stroke patienter för att säkerställa prestanda. Det är även väsentligt att överväga vem som bär ansvaret för beslut som tas av AI/ML-systemet. Genom att säkerställa att systemet kompletterar snarare än ersätter vårdpersonalens bedömningar kan en balans upprätthållas mellan teknologi och professionalism.

4 Litteraturstudie

Inledningsvis användes det tidigare årets kandidatarbete - "Early Characterization of Stroke Using Video Analysis and Machine Learning" [16] - för att få en översikt kring arbetet och området. Kandidatarbetet är planerat och konstruerat som en slags vidareutveckling och fördjupande av den ovannämnda kandidaten. Skillnaden från det äldre kandidatarbetet var det nya projektets bredd och omfattning. Det föregående kandidatarbetet hanterade flera olika symptom på olika kroppsdelar som kan orsakas av stroke och gjorde en sammanvägning mellan dessa. Detta täckte allt från ansiktsanalys, i både övre och nedre ansiktet, till röstsvårigheter. Detta årets kandidatprojekt är avskalat och endast fokuserat på ögats rörelse under stroke. Tanken är att arbetet ska fördjupas och ge mer insiktsfulla och givande resultat för att digitalisera mer av NIHSS. Således blev det tidigare kandidatarbetet en väldigt effektiv introduktion och en slags förlaga för detta kandidatarbete.

Efter denna grund genomfördes en litteraturstudie för att hitta data och information som passade vårt område, och för att få en överblick på forskningsfältet. För att hitta relevanta papper om stroke, videoanalys, ML och ögonspårning så gjordes detta med hjälp av scopus databas och google sökmotor. Litteraturen som söktes och efterfrågades skulle handla om applicerandet av ML inom stroke- och syndata. Scopus är en av de största databaserna för peer-review litteratur och en industriledande samling vetenskapliga artiklar och konferensartiklar [38]. Fortsättningsvis erbjuder Scopus avancerade sökningsinmatningar genom nyckelord med avancerade filtreringsalgoritmer. Google, till skillnad från akademiska databaser som Scopus, är en mer omfattande sökmotor som indexerar och gör alla sorters internetsidor tillgängliga [39]. Det är den mest använda sökmotorn och sorterar sidor med hjälp av avancerade algoritmer som avgör relevans och popularitet.

4.1 Algoritmer

I Scopus databas hittades bland annat en givande kartläggande studie [40]. Denna gav en bred överblick kring vilka algoritmer som var relevanta inom området. Tabell 1 och 2 gick att avläsa från den kartläggande studien.

ML / Aktivitet	Kodförståelse	Kodkritik	Parprogrammering	Debugging	Totalt
Support Vector Machine	4	0	0	0	4
Decision Tree	3	0	0	0	3
K-nearest Neighbours	1	1	0	0	2
Naïve Bayes	1	0	1	0	2
Random Forest	1	1	0	0	2
Ensemble Learning	1	0	0	0	1
Graph Embedding	0	0	0	1	1
K-means Clustering	1	0	0	0	1
Linear Regression	0	0	1	0	1

Tabell 1: Värmekarta av mjukvaruaktivitet x ML-teknik. Denna är tagen och översatt från kartläggningstudien [40]. Här sticker Support Vector Machine ut i antal gånger den blev applicerad.

Papper	Inmatning	Modell	Utmatning
ASW22 [Abbadi et al. 2022]	Linje- och fragmentnivåns blickdata	Decision Tree, Ensemble Learning	Mentalt krävande kodfragment
APSM21 [Al Maoklen et al. 2021]	Enskild fixeringstid	Random Forest	Programmerarens expertis (nybörjare eller expert)
HCC+21 [Hijazi et al. 2021]	Biometrisk egenskaper	K-nearest Neighbour	Kodgranskningskvalitet av kodregioner
HN21 [Harada and Nakayama 2021]	Ögonrörelseegenskaper	Support Vector Regression	Förmåga att läsa kod
AO20 [Ahsan and Obaidallah 2020]	Total fixeringstid	K-means clustering, K-nearest Neighbour, Support Vector Machine	Expertis hos nybörjarprogrammerare
VBMB20 [Vrzakova et al. 2020]	Blickdata + blickförskjutningsdata	Random Forest	Påverkan av kodgranskningen
ZSPSY19 [Zhang et al. 2019]	Ögonrörelsetrajektorier	Graph Embedding (node2vec)	Inbäddade rumspositioner
VR18 [Villamor and Rodrigo 2018]	CRQA-mätvärden	Naïve Bayes, Linear Regression	Framgång med parprogrammering
LHJNL18 [Lee et al. 2018]	Data från psykofysiologiska sensorer	Support Vector Machine	Programmerarens expertis och uppgiftssvårighet
FBMYZ14 [Fritz et al. 2014]	Ögonrörelsedata	Naïve Bayes, Support Vector Machine, Decision Tree	Uppgiftssvårighet

Tabell 2: Översikt av inmatning, utmatning och typ av ML-algoritm som var använd. Denna är tagen och översatt från kartläggningstudien [40]. Den visar vilka papper studien är tagen från, vilken modell som är anpassad, och vad in- och utmatning var.

Papper	Inmatning	Modell	Utmatning
HN21 [Harada and Nakayama 2021]	Ögonrörelseegenskaper	Support Vector Regression	Förmåga att läsa kod
FBMYZ14 [Fritz et al. 2014]	Ögonrörelsedata	Naïve Bayes, Support Vector Machine, Decision Tree	Uppgiftssvårighet

Tabell 3: Extraherade rader från tabell 2 som använder av SVM för att klassificera ögonrörelser.

Tabell 1 och 2 gav en god överblick kring vilka algoritmer som skulle tänkas vara relevanta för arbetet. Den som verkade vara mest central var Support Vector Machine (SVM), som således blev en aktuell kandidat för arbetet. Det är också tydligt i tabell 3 som är extraherad ur tabell 2. Några av de andra algoritmerna från kartläggningsstudien testades också utan framgång. De andra algoritmerna som valdes att implementeras hittades huvudsakligen genom generella googlesökningar i området som ej kan strukturerat redovisas i efterhand.

Vidare genomfördes en överskådlig litteraturstudie kring varje algoritms generella prestanda inom området ögonspårning. Relevanta algoritmer hittades genom scopus databas och dess prestanda placerades i tabeller. För varje enskild algoritm användes söktermerna i figurerna 1, 2, 3, 4 och 5. Sökningen tog lång tid eftersom de flesta källor inte hade kvantitativa måttal som efterfrågades. Uppskattningsvis så var 1/10 av de artiklar som verkade intressanta av värdefullt innehåll. De flesta källor innehöll annat sätt att mäta prestandan av algoritmerna som exempelvis pixelvis felmarginal. Alla källorna sorterades efter relevans och inget krav på peer-review ställdes. En generell regel var att ej använda artiklar som publicerats innan 2010 för att säkerställa aktuella insikter. Sedan var även antal referenser till andra artiklar på scopus en värdeful indikator på dess relevans. Däremot ställdes inga andra systematiska krav. Field-Weighted citation impact (FWCI) var inget heller som medräknades i urvalet men som redovisas. Särskilt svårt var det att hitta värdefulla artiklar kring Gated Recurrent Unit (GRU) och Deep Neural Network (DNN).

SVM

1 TITLE (SVM OR (Support AND Vector AND Machine) AND (GAZE OR EYE)) OR KEY (SVM OR (
 2 Support AND Vector AND Machine) AND (GAZE OR EYE))

Figur 1: Använda sök- och nyckelord för att hitta artiklar och konferenspaper på Scopus relaterade till SVM.

Algorithm, Nummer	Originaltitel	Författare, År	Studietyp	Datakaraktär	Citeringar	FWCI
SVM[41], 1	Human Identification using Linear Multiclass SVM and Eye Movement Biometrics	Namrata Srivastava et al., 2015	Retrospektiv studie	Empirisk data från EMVIC 2012 dataset, inspelad med 250Hz Ober2 eye-tracker	14 (55:e procentil)	0.51
SVM[42], 2	EOG Based Eye Movements and Blinks Classification using Irisgram and CNN-SVM Classifier	Ala'a Zyout et al., 2023	Prospektiv studie	Testning på EOG dataset, 10-55 grader horisontellt, 10-35 grader vertikalt ögonrörelse, och blinkningar	0	-
SVM[43], 3	Comparison of ANN and SVM for classification of eye movements in EOG signals	Lim Jia Qi, Norma Alias, 2018	Prospektiv studie	Empirisk testning på EOG signaler från 5 frivilliga personer	20 (97:e procentil)	4.90

Tabell 4: Övergripande beskrivning av SVM algoritmer och dess grad av inflytande.

Algorithm, Nummer	Noggrannhet	Precision	Specificitet (Specificity)	Känslighet (Sensitivity)	F1-poäng (F1-Score)	Kommentarer
SVM, 1	100% för 6 klasser, 91.67% för 12 klasser	-	-	-	-	Ögats positionsdifferens presterar bättre än hastighetsvektorer
SVM, 2	98.9%	96.8%-99.7%	99.5%-99.9%	98.2%-99.2%	98.0%-99.2%	Använder tredje ordnings kernel för klassificering
SVM, 3	69.75%	-	-	-	-	-

Tabell 5: Kvantitativa mått och kommentarer för SVM algoritmer.

Med sökorden i figur 1 hittades de tre studierna i tabell 4 och 5 som alla hade unika appliceringsätt av algoritmen SVM på ögonrörelsedata. Tabell 4 går genom övergripande vad studien handlar om och hur stort inflytande den har haft på forskningssfären. Tabell 5 har extraherat värdefull information och kvantitativa måttal som kan användas för att jämföra med det faktiska resultatet kandidatarbetet får, och varför eller varför inte projektet avviker från andra liknande projekt.

I studien skriven av Namrata Srivastava et al. presenterade en studie där en linjär multiklass-SVM användes för att identifiera individer baserat på deras unika ögonrörelsemönster [41]. Datan var insamlad genom att olika personer fick fixera på olika sorters visuell stimuli. Testningen av SVM-modellen gav som resultat en noggrannhet på 100% för 6 klasser och en noggrannhet på 91.67% för 12 klasser. Resultaten indikerade att ögonbaserade identifikationsprocesser är lovande för framtida applicering i kombination med EEG, ansiktsgenkänning och liknande tekniker. De kom även fram till att användande av ögats positionsdifferens var effektivt och presterade bättre än med användningen av hastighetsvektorer.

Artiklen skriven av Ala'a Zyout et al. handlade om så kallade *irisgram*, som är en tvådimensionell representation av kort-tids fourier transformering i formen av ögats iris [42]. Dessa irisgram kunde sedan användas som inmatning i en Convolutional Neural Network-algoritm (CNN). Sedan kunde resultatet från CNN inmatas in en SVM-algoritm som kategoriserade ögonrörelserna i kategorier blinkningar eller fyra olika sorters ögonrörelser. Den kombinerade prestandan av CNN-SVM var en noggrannhet av 96.2%. Däremot så hade SVM enskilt en noggrannhet på 98.9%. Det är värt att notera att en SVM-algoritm med tredje ordningens kernel var använd.

Studien skriven av Lim Jia Qi, Norma Alias handlade om insamlandet av EOG-data från 5 frivilliga personer [43]. Sedan blev datan förprocessad genom huvudsakligen tre metoder - statistiska parametrar, autoregressiva koefficienter med Burg metoden, och slutligen spektral effekttäthet med Yule-Walker metoden. Sedan blev datan placerad i ett artificiellt nätverk eller ett SVM-algoritm. Resultatet visade att SVM-algoritmen kombinerad med att datan blev förprocessad med statistiska parametrar gav högst noggrannhet på 69.75%.

DNN

1 TITLE (DNN OR (Deep AND Neural AND Network) AND (GAZE OR EYE)) OR KEY (DNN OR (Deep
2 AND Neural AND Network) AND (GAZE OR EYE))

Figur 2: Använda sök- och nyckelord för hitta artiklar och konferenspaper på Scopus relaterade till DNN.

Algoritm, Nummer	Originaltitel	Författare, År	Studietyp	Datakaraktär	Citeringar	FWCI
DNN[44], 1	Eye-gaze Estimation with HEOG and Neck EMG using Deep Neural Networks	Zhen Fu, Bo Wang, Fei Chen, Xihong Wu, Jing Chen, 2021	Prospektiv studie	Empirisk testning med både fix och rörligt huvud. Insamlad data inkluderade horisontell elektrookulografi (HEOG) och nackens elektromyografi (NEMG)	1 (42:e percentil)	0.32
DNN[45], 2	AutoML and Neural Architecture Search for Gaze Estimation	Adrian Bublea, Catalin Daniel Căleanu, 2022	-	Använde datasetten MPIIGaze (Appearance-based Gaze Estimation in the Wild) och CAVE (Columbia Gaze Data Set)	2 (79:e percentil)	1.38
DNN[46], 3	Gaze Fusion-Deep Neural Network Model for Glaucoma Detection	Sajitha Krishnan, J. Amudha, Sushma Tejwani, 2021	-	Visuellt fält och observationsförmåga från deltagare inmatat i fusion maps och sen klassificerat som glaukom eller frisk ögonrörelse	6 (92:e percentil)	2.81

Tabell 6: Övergripande beskrivning av DNN-algoritmer och dess grad av inflytande.

Algoritm, Nummer	Noggrannhet	Precision	Specificitet	Känslighet	F1-poäng	Kommentarer
DNN, 1	HEOG 71%, NEMG 35.7%, HEOG och NEMG 72.6%	-	-	-	-	HEOG 71% är den enda intressanta noggrannheten
DNN, 2	AutoKeras på CAVE 87%, Auto-sklearn på CAVE 21%, TPOT på CAVE 20%, FEDOT på CAVE 22%	-	-	-	-	Den intressanta är AutoKeras på CAVE 87% som motsvarar DNN
DNN, 3	87.5%	-	83%	100%	-	DNN modellen gav en noggrannhet på 87.5%

Tabell 7: Kvantitativa mått och kommentarer för DNN-algoritmer.

Med sökorden i figur 2 hittades de tre studierna i tabell 6 och 7 som alla hade unika appliceringsätt av algoritmen DNN på ögonrörelsedata. Tabell 6 går genom övergripande vad studien handlar om och stort inflytande den har haft på forskningssfären. Tabell 7 har extraherat värdefull information och kvantitativa mätetal som kan användas för att jämföra med det faktiska resultatet kandidatarbetet får, och varför eller varför inte projektet avviker från andra liknande projekt.

I en studie skriven av Zhen Fu et. al har det tittats på ögonens riktning och nackens electromyografi och kopplat det till hörapparaten hos hörselnedsatta för att kunna rikta hörapparaten mot någon som pratar

[44]. Riktningen av ljudupptagningen gör det enklare för hörselnedsatta att fokusera på den som pratar i en högljudd miljö. Studien använde och evaluerade DNN med både nack-electromyografi (NEMG) och horisontel electrooculogram (H-EOG) som gav en prestanda på 72.6% [44]. När DNN evaluerades med HEOG och NEMG separat gav de en noggrannhet på 71% respektive 35.7%. För detta projekt är det HEOGs noggrannhet (71%) som är i fokus då det handlar om ögonrörelse.

Artikeln skriven av Adrian Bublea, Catalin Daniel Căleanu handlar om att applicera två olika metoder för att hitta det mest optimala systemet för blickförutsägelse [45]. Det första systemet var Autokeras neurala arkitektur sökningsverktyg och det andra var AutoML. Autokeras använder NAS systemet för att automatiskt konfigurera och optimera nätverket. AutoML motsvarar mer traditionella ML-algoritmer. Studien jämförde dessa två tekniker på två dataset - Gaze Estimation in the Wild (MPIIGaze) och "Columbia Gaze Data Set" (CAVE). Autokeras på CAVE uppnådde en noggrannhet på 87%, Auto-sklearn på CAVE en noggrannhet på 21%, TPOT på CAVE en noggrannhet på 20%, och slutligen FEDOT på CAVE en noggrannhet på 22%. Det intressanta för oss är Autokeras på CAVE (87%) som motsvarar ett DNN.

I en annan studie skriven av Sajitha Krishnan et. al tittades det på ögonrörelsedata i form av fusionkartor hos subjekt med och utan glaukom för att separera dem in i två klasser av icke-glaukom och glaukom patienter [46]. Klassindelningen gjordes genom att lägga in fusionkartorna från deltagarna i en DNN modell som därefter tränades på datan och gjorde en klassificering. Resultatet gav en DNN-modell med en XGBoost klassificerare vilket gav en noggrannhet på 87.5% [46].

CNN

```

1 TITLE ( CNN OR (Convolutional AND Neural AND Network) AND (GAZE OR EYE)) OR KEY ( CNN OR
2 (Convolutional AND Neural AND Network) AND (GAZE OR EYE))

```

Figur 3: Använda sök- och nyckelord för hitta artiklar och konferenspapper på Scopus relaterade till CNN.

Algoritm, Nummer	Originaltitel	Författare, År	Studietyp	Datakaraktär	Citeringar	FWCI
CNN[42], 1	EOG Based Eye Movements and Blinks Classification using Irisgram and CNN-SVM Classifier	Ala'a Zyout et al., 2023	Prospektiv studie	Empirisk testning och validation på EOG dataset, 10-55 grader horisontellt, 10-35 grader vertikalt ögonrörelse, och blinkningar	0	-
CNN[47], 2	Gaze Prediction Based on Convolutional Neural Network	Abdulkader Helwan, Mohammad Khaled Sallam Ma'aitah, Selin Uzelaltinbulat, Mohamad Ziad Altobel, Manal Darwish, 2021	-	Empirisk testning på en allmänt tillgänglig databank. Använder bilder från Georgia Tech ansiktsdatabas, som inkluderade studenter som tittade rakt, höger och vänster.	14 (99:e procentil)	8.78
CNN[48], 3	A Framework For Human Behaviour Detection Using Combined Analysis of Facial Expression and Eye Gaze	Pankaj Jyoti Das, Anjan Kumar Talukdar, Kandarpa Kumar Sarma, 2019	-	Empirisk testning med ansiktsuttryck kombinerat med ögonrörelse. MUG databas som innefattar universella ansiktsuttryck.	0	-

Tabell 8: Övergripande beskrivning av CNN-algoritmer och dess grad av inflytande.

Algoritm, Nummer	Noggrannhet	Precision	Specifitet	Känslighet	F1-poäng	Kommentarer
CNN, 1	96.2%	-	-	-	-	-
CNN, 2	90.76% eller 88.76% med klassificering av blickriktning	-	-	-	-	-
CNN, 3	94%	-	-	-	-	-

Tabell 9: Kvantitativa mått och kommentarer för CNN-algoritmer.

Med sökorden i figur 3 hittades de tre studierna i tabell 8 och 9 som alla hade unika appliceringsätt av algoritmen CNN på ögonrörelsedata. Tabell 8 går genom övergripande vad studien handlar om och stort inflytande den har haft på forskningssfären. Tabell 9 har extraherat värdefull information och kvantitativa måttal som kan användas för att jämföra med det faktiska resultatet kandidatarbetet får, och varför eller varför inte projektet avviker från andra liknande projekt.

I en studie av Abdulkader Helwan et al. [47] fokuseras det på förutsägelsen av blickriktningar hos studenter i klassrum genom ansiktsbildanalys med hjälp av en CNN. Systemet identifierar om studenterna tittar mot läraren och kan signalera om deras uppmärksamhet behöver återfokuseras. Forskarna föreslår en algoritm för ögonrörelse baserad på ansiktsrörelser, där den upptäckta ögonregionen används som indata till CNN för att klassificera blicken som antingen rakt fram, åt höger eller vänster. Resultaten, erhållna från en offentlig databas, understryker CNNs effektivitet i att hantera multiklassificeringsproblem inom ögonspårning, vilket

kan ha implikationer för pedagogiska verktyg i klassrummet.

Ytterligare en relevant studie skriven av Pankaj Jyoti Das et al. [48] kombinerar ansiktsuttrycksigenkänning med ögonspårning för att förutsäga mänskligt beteende, som koncentrationsnivåer hos förare. Deras arbete använder en förtränad CNN-baserad modell på VGG16, kompletterat med Viola-Jones och Circular Hough Transform för ögonspårning. Resultatet är ett omfattande system som inte bara tolkar ansiktsuttryck utan också noggrant uppskattar blickens riktning. Med en noggrannhet på 94% för hela processen erbjuder deras metod en robust modell för realtidsanalys av mänskligt beteende, vilket kan ha breda tillämpningar inom HCI och MMI.

GRU

TITLE (GRU OR (Gated AND Recurrent AND Unit) AND (GAZE OR EYE)) OR KEY (GRU OR (Gated AND Recurrent AND Unit) AND (GAZE OR EYE))

Figur 4: Använda sök- och nyckelord för hitta artiklar och konferenspaper på Scopus relaterade till GRU.

Algoritm, Nummer	Originaltitel	Författare, År	Studietyp	Datakaraktär	Citeringar	FWCI
GRU[49], 1	Applying Eye Tracking with Deep Learning Techniques for Early-Stage Detection of Autism Spectrum Disorders	Ahmed Z.A.T., Albalawi E., Aldhyani T.H.H., et al., 2023	-	Empirisk testning, ögonspårningsdataset med 29 ASD (Autism Spectrum Disorder) och 30 TD (Typically Developing) barn, tidig karaktärisering autism	0	-
GRU[50], 2	Evaluation of e-learners' concentration using recurrent neural networks	Young-Sang Jeong, Nam-Wook Cho, 2023	Prospektiv studie	Empirisk testning med data från 92 studenter, bestämmer e-learners koncentration från blickriktning och mätpunkter i ansiktet	2 (71:e percentil)	0.88
GRU[51], 3	Deepfake Video Detection by Using Convolutional Gated Recurrent Unit	Yifeng Tu, Yang Liu, Xueming Li, 2021	-	Använder Celeb-DF(v2) dataset för att detektera deepfake videos, fokuserar på ConvGRUs förmåga	4 (66:e percentil)	0.82

Tabell 10: Övergripande beskrivning av GRU algoritmer och dess grad av inflytande.

Algoritm, Nummer	Noggrannhet	Precision	Specificitet	Känslighet	F1-poäng	Kommentarer
GRU, 1	97.49%	97.15%	98.40%	95.89%	98.04%	Samtliga kvantitativa mätetal går att räkna ut pga TP, FP, FN, TN finns tillgänglig för varje algoritm. Således räknades kvarstående mätetal - precision - ut.
GRU, 2	82.66%-84.31%	71.48%-90.63%	-	80.82%-84.62%	75.26%-87.85%	Mätalen varierar baserat på datakoncentration och RNN och FF lager.
GRU, 3	83.84%-94.56%	-	-	-	-	Skilnad i noggrannhet kommer från antal bilder (5-15), där bästa resultat togs när antalet bilder som användes var 10.

Tabell 11: Kvantitativa mått och kommentarer för GRU-algoritmer.

Med sökorden i figur 4 hittades de tre studierna i tabell 10 och 11 som alla hade unika applicerings sätt av algoritmen GRU på ögonrörelsedata. Tabell 10 går genom övergripande vad studien handlar om och stort inflytande den har haft på forskningsfären. Tabell 11 har extraherat värdefull information och kvantitativa mätetal som kan användas för att jämföra med det faktiska resultatet kandidat arbetet får, och varför eller varför inte projektet avviker från andra liknande projekt.

I en studien av Ahmed et. al [49] utvärderades prestandan hos fyra RNN-algoritmer för att uppskatta autismspektrumtillstånd hos barn via ögonspårning. Ett ögonspårnings dataset på 59 barn, varav 29 med autismspektrumtillstånd togs, datan tränades och testades av BiLSTM, LSTM, GRU och CNN-LSTM hybrid. I studien presterade LSTM bäst med en noggrannhet av 98.33%, GRU presterade näst bäst med en

noggrannhet på 97.94% och GRU och BiLSTM presterade sämst med 97.49% respektive 96.44%. Samtliga algoritmer visar på väldigt bra resultat och visar på att de är bra alternativ vid ML för ögonspårning.

En annan studie av Young-Sang Jeong och Nam-Wook Cho visar på liknande resultat då de utvärderar algoritmers förmåga att uppskatta koncentrationsförmåga hos studenter under online-lektioner genom att identifiera blick och ansiktsdrag [50]. Totalt extraherades 184 videos från 92 studenter med hjälp av OpenFace 2.0. Tre algoritmer undersöktes: dubbelriktad (bidirectional) RNN, LSTM och GRU där GRU presterade bäst med en noggrannhet mellan spannet 82.66% - 84.3%. Studien fortsätter även att beskriva hur andra modeller som CNN och temporal convolutional networks kan användas vid framtida studier.

I en studie av Yifeng Tu, Yang Liu och Xueming Li används CNN och convolutional GRU (convGRU) för att detektera djupfejk (deepfake) och evalueras med Celeb-DF(v2) dataset som är ett dataset med djupfejk syntetiska videor [51]. CNN användes för att först extrahera egenskaper för varje bild i djupfejk videorna, dessa bilder användes sedan för att träna en GRU modell för att kunna detektera ifall en video var djupfejk eller inte. Datasetet som användes bestod av 590 riktiga videor och 5639 djupfejk videor. Flera test gjordes med olika mängder bilder och det bästa resultatet som uppnåddes var 94.56% då 10 frames användes.

LSTM

```

1 TITLE ( LSTM OR (Long AND Short-term AND Memory) AND (GAZE OR EYE)) OR KEY ( LSTM OR (
2 Long AND Short-term AND Memory) AND (GAZE OR EYE))

```

Figur 5: Använda sök- och nyckelord för hitta artiklar och konferenspapper på Scopus relaterade till LSTM.

Algoritm, Nummer	Originaltitel	Författare, År	Studietyper	Datakaraktär	Citeringar	FWCI
LSTM[52], 1	Accuracy Improvement of Object Selection in Gaze Gesture Application using Deep Learning	M. Alfaroby E., Sunu Wibirama, Igi Ardiyanto, 2020	-	Data från ett tidigare blickorienterat projekt med ögonspåringsdata på 34 deltagare	7 (82:e percentilen)	1.49
LSTM[49], 2	Applying Eye Tracking with Deep Learning Techniques for Early-Stage Detection of Autism Spectrum Disorders	Ahmed Z.A.T., Albalawi E., Aldhyani T.H.H., et al., 2023	-	Empirisk testning, ögonspåringsdataset med 29 ASD (Autism Spectrum Disorder) och 30 TD (Typically Developing) barn, tidig karakterisering autism	0	-
LSTM[50], 3	Evaluation of e-learners' concentration using recurrent neural networks	Young-Sang Jeong, Nam-Wook Cho, 2023	Prospektiv studie	Empirisk testning med data från 92 studenter, bestämmer e-learners koncentration från blickriktning och mätpunkter i ansiktet	2 (71:e percentil)	0.88

Tabell 12: Övergripande beskrivning av LSTM algoritmer och dess grad av inflytande.

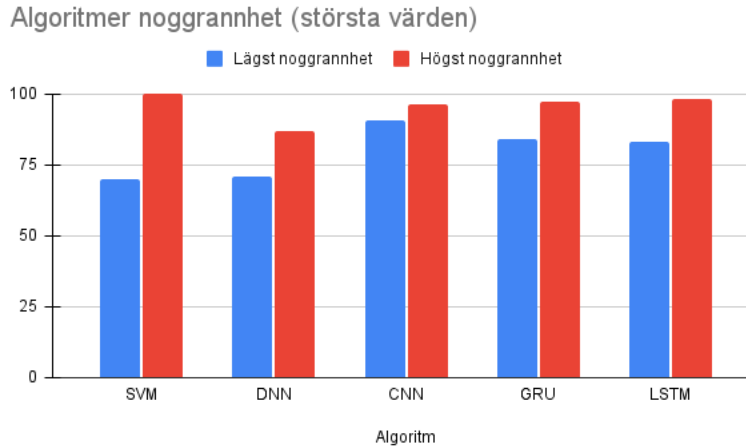
Algoritm, Nummer	Noggrannhet	Precision	Specifitet	Känslighet	F1-poäng	Kommentarer
LSTM, 1	96.17%	-	-	-	-	-
LSTM, 2	98.33%	98.11%	98.94%	97.25%	98.7%	Samtliga kvantitativa mätetal går att räkna ut pga TP, FP, FN, TN finns tillgänglig för varje algoritm. Således räknades kvarstående mätetal - precision - ut.
LSTM, 3	80.05%-83.18%	68.3%-89.53%	-	75.77%-84.37%	72.16%-86.93%	Mätalen varierar baserat på datakoncentration och RNN och FF lager.

Tabell 13: Kvantitativa mått och kommentarer för LSTM-algoritmer.

Prestandan för LSTM evalueras i studien [49] av Ahmed et. al och [50] av Young-Sang Jeong och Nam-Wook Cho där algoritmen uppnådde en noggrannhet på 98.33% respektive 80.05-83.13% i varsina områden. Dessa två studier presenterades tidigare under stycket om GRU.

I en studie [52] av Sunu Wibirama, M. Alfaroby E. och Igi Ardiyanto används LSTM-algoritmen för att lära in mönster från blick-data för objektval i en blickgestbaserad (gaze gesture) applikation. Objektval hos blick-baserade interaktiva applikationer sker främst genom att via en ögonspårare kontrollerar vilket objekt blicken stannar på under en längre tid, eller sackadiska rörelser från ett objekt till ett annat. För att öka hastigheten och noggrannheten av blickgest inmatningar tränades en LSTM-modell på ett dataset med 34 deltagare och en noggrannhet på 96.17% togs fram.

Med sökordet i figur 5 hittades de tre studierna i tabell 12 och 13 som alla hade unika appliceringsätt av algoritmen LSTM på ögonrörelsedata. Tabell 12 går genom övergripande vad studien handlar om och stort inflytande den har haft på forskningssfären. Tabell 13 har extraherat värdeful information och kvantitativa måttal som kan användas för att jämföra med det faktiska resultatet kandidatarbetet får, och varför eller varför inte projektet avviker från andra liknande projekt.



Figur 6: Sammanställning av algoritmernas prestanda under litteraturundersökningen av algoritmer. Denna figur visualiserar differensen mellan de högsta och lägsta noggrannheten i varje enskild algoritm. Den röda stapeln motsvarar den bästa noggrannheten för algoritmen, och den blåa motsvarar den sämsta. Det förekommer fall där en algoritm har presterat på ett spann vid ett enskilt appliceringstillfälle. Just denna sammanställning räknar med varje enskilds algoritm bästa resultat på detta spannet.

Figur 6 ger en grund till vad som kan förväntas i noggrannhet bland de olika algoritmerna under ungefärligt liknande omständigheter. Den som sticker ut mest bland algoritmerna är SVM och DNN som har relativt lägsta gräns noggrannhet i jämförelse med de andra algoritmerna. SVM presterar allra sämst i detta avseende. Däremot så visar anpassningar av CNN, GRU och LSTM relativt jämna och bra resultat utan större noggrannhetsspann vilket är ett gott tecken på att de har goda förutsättningar att prestera bra. Det högsta resultatet för SVM var väldigt högt och gav högsta noggrannheten av alla algoritmer som medverkade i litteraturstudien. LSTM, GRU och CNN presterade något lägre men fortfarande högt med noggrannheter på över 90%. Motsvarande figurer för andra kvantitativa måttal gick ej att konstruera på grund av brist på data.

En generell anmärkning av litteraturen är att många av studierna applicerade kombinerade ML-algoritmer för att ge förutsägelser kring ögonrörelser. I exempelvis SVM 2 så användes en hybridvariant mellan SVM och CNN. Först matades deras irisgram, som ser ut som ögats iris, in i CNN. Detta är en två-dimensionell representation av kort-tids fourier transformationer av EOG-signaler. Denna irisfigur visualiserar tidfrekvensers karaktärsdrag av ögonrörelser. Slutligen tas resultatet av CNN och placeras i SVM för att kategorisera olika ögonrörelser (blinkningar, kolla höger, etc). Sedan avgjordes modellens noggrannhet. Två slutsatser kan dras av detta. Deras kombinerade noggrannhet är till en grad missvisande i jämförelse med vårt projekt. Den andra slutsatsen är att det som motsvarar ögonrörelser och hur ögondata behandlas har en bred tolkning.

I många av studierna var också ögonrörelse bara en av faktorerna som matades som data i ML-algoritmen. Ett bra exempel är CNN 3 som tillgodosåg både ansikts- och ögonrörelser i sin modell. Eftersom ögonrörelse bara blir en faktor av resultatet kan således detta vara missvisande i avseende att bedöma algoritmens förmåga kring ögonrörelser. Alla dessa anmärkningar är viktiga för att inte ha för höga förväntningar på algoritmerna som tas fram. Det är viktigt att förstå litteraturstudiens begränsningar så att inte resultatet av studien upplevs prestera under standard i jämförelse med litteraturen.

Litteraturstudien ledde till insikten att uppgiften kräver algoritmer som är specialiserade på att hantera sekvensiell data. Denna typ av data kräver modeller som kan beakta tidsberoende relationer mellan tidsstegen i sekvensensekvensens olika delar. Ett fokus riktades mot mer avancerade maskininlärningsmetoder som kan hantera den sekvensiella naturen av datan. DNN och speciellt RNN, framstod som lovande kandidater. Dessa modeller är utformade för att känna igen och lära sig från mönster i data där tidigare händelser påverkar de

framtida, vilket är fallet med ögonrörelser i videosekvenser [53].

4.2 Syntetisk data

Det mest omfattande delen av litteraturstudien var undersökandet kring syntetisk data. Inledningsvis var två arbeten i fokus [54][55] som båda var författade av Erroll Wood et. al, med finansiering av Microsoft. Här användes Pytorch, ett bibliotek till python, för att parse och syntisera ansikten. Resultaten i undersökningarna verkade väldigt lovande, men implementering av arbetet visade sig väldigt svårt. Det var svårt att följa tillvägagångssättet som användes för att generera syntetisk data. Således gjordes försök att få kontakt med flera i forskningsteamet utan framgång. När detta misslyckades söktes det efter nya medel att generera syntetisk data. En artikel [56] som handlade om rendering av syntetiska bilder för ögonspårningevaluering inspirerade användandet av Blender för genereringen av syntetisk data. Studien använde sig av 3D modeller av ögon och huvuden för att rendera realistiska syntetiska bilder av ögon där det gick att noggrant bestämma parametrar som blickriktning och kameravinklar för att få utökad mängd data [56]. Ytterligare en artikel [57] visade ett program som genererar hyperrealistiska syntetiska bilder av ansikten. Problemet med programmet var att den krävde mycket processering och bar beräkningstung.

5 Analys

Kapitlet analys diskuterar metodiken och de tekniker som har använts under projektets gång. Detta innefattar allt från datainsamling till optimering av algoritmer. Slutligen diskuteras resultatet som är framtaget. Reflektion görs över de medel som användes under datagenereringen och jämförelser gjordes mellan algoritmerna och de olika typerna av data.

5.1 Metod

I det här kapitlet beskrivs projektet metod. Efter att litteraturstudien var genomförd så påbörjades inspelning av videor där symptom för stroke imiterades enligt NIHSS moment 2, där videorna sedan digitaliserades. Sedan skapades syntetisk data och den samlade datan användes för att träna och testa ML-algoritmer som skapades.

5.1.1 Urval av ögonfel och undersökningsmetod

Trots att projektet syftar till att undersöka en digitalisering av NIHSS så undersöktes möjligheterna att använda sig av andra ögonfel som är vanliga vid stroke. I diskussion med Petra Redfors, överläkare i neurologi vid Sahlgrenska Universitetssjukhus, bedömdes det att 2 olika ögonfel hade möjlighet att imiteras av friska personer; komplett blickpares eller partiell blickpares i båda ögonen. I diskussion med Redfors nämndes även andra symptom som inte går att imiteras av friska personer; isolerad blickpares och nystagmus.

Testning av ögonrörelse är en del av NIHSS och används för att undersöka blickpares samt partiell blickpares. Detta projekt följer moment 2, "Horisontell ögonrörelse" i NIHSS [24], som kan ses i tabell 14 nedan. Projektet genomfördes på så sätt att deltagaren instrueras att fokusera på ett finger eller en penna som förs från sida till sida för att testa deltagarens ögonmotorik.

Moment	Poängsättning
2. Horisontell ögonrörelse	Normal: +0 Partiell blickpares (som kan övervinnas själv eller med Doll's eye test), isolerad pares: +1 Komplett blickpares (kan inte övervinnas): +2

Tabell 14: NIHSS moment 2: Horisontell ögonrörelse [24].

5.1.2 Datainsamling

Datainsamling bestod av tre delar, inspelning av videos på friska deltagare som imiterade stroke, skapande av syntetiska data som simulerar ögonrörelser hos en strokepatient samt videobearbetning för att bättre integrera de olika datatyperna med varandra.

Videoinspelning

Videoinspelning genomfördes i en isolerad miljö med optimal belysning för att minimera skuggbildning som kunde påverka träningsdatan, se figur 7. Inspelningarna utfördes med en Sony A6100 kamera, utrustad med ett Sony E PZ 16-50mm F3.5-5.6 OSS objektiv. För att uppnå önskad ljussättning användes två Neewer LED 32 watt ljuskällor med en färgtemperatur på 4500 Kelvin, inställda på 70% av maximal ljusstyrka. Ljusspridare applicerades på båda ljuskällorna som var riktade mot subjektet. Användningen av ljusspridare syftade till att eliminera skarpa skuggor i ansiktet, inklusive de från näsan, vilka kunde täcka ögonen och därmed påverka analysen vid ögonspårning. Kameran monterades på ett trebensstativ med en maximal höjd av 1.3 meter och placerades cirka 1 meter från subjektet. Under inspelningarna imiterade deltagarna symptom för stroke av olika grad enligt skalan, varvid inspelat material sorterades och klassificerades enligt ett förutbestämt filträd för att säkerställa att datan inte blandades. Alla videoklipp spelades in i 50 bilder per sekund.

Totalt deltog sju personer som subjekt i inspelningen av verkliga data, varav fem var författarna till denna rapport. Insamlingsprocessen organiserades i flera sessioner, där varje subjekt medverkade i inspelningen av mellan tre till åtta videoklipp per session. Under dessa sessioner genomfördes ett test, benämnt "finger test", där subjektet var stillasittande och följde en penna med blicken åt höger och vänster, samtidigt som olika grader av strokesymptom imiterades enligt NIHSS. För videodata med klassificering 0 registrerades två olika typer av videoklipp. En typ där en penna rörde sig framför ansiktet och täckte ögonen, och en utan penna i bild. För videodata med klassificering 1 filmades subjektets blick då den initialt var riktad åt vänster alternativt åt höger, och sedan följde pennan enligt "finger testet", dock med begränsad ögonrörelse. För klassificering 2 upprätthöll subjektet sin blick åt höger eller vänster under hela videons längd för att efterlikna en högre grad av stroke enligt NIHSS. Syftet med denna metod var att systematiskt samla videodata för alla klassificeringar i en och samma session för varje deltagare, vilket säkerställde att subjektet bibehöll en optimal position i relation till kameran och ljussättningen, se figur 8 för en exempelbild. Totalt resulterade detta i 99 inspelade videor. Antalet videor för varje klassification ses i tabell 15.



Figur 7: Uppställningen av utrustning för videoinspelningar.



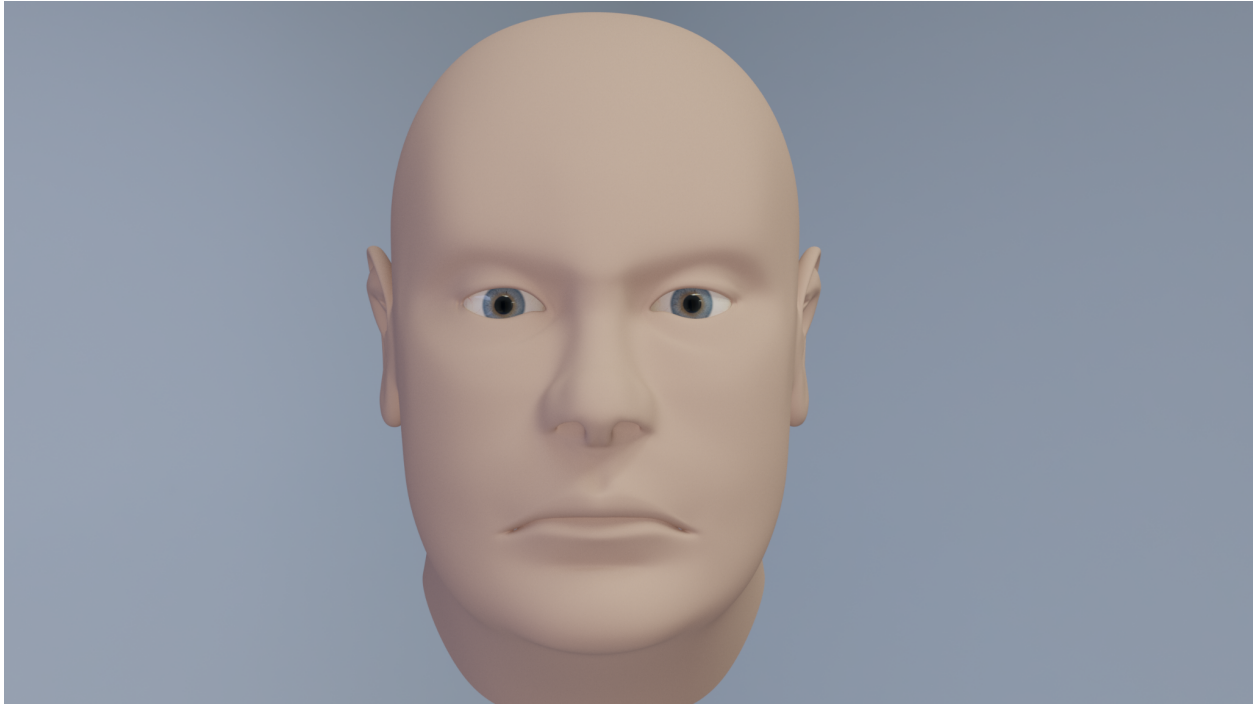
Figur 8: Exempelbild på videodata från klassificering 0 med all utrustning uppställd.

Syntetisk data

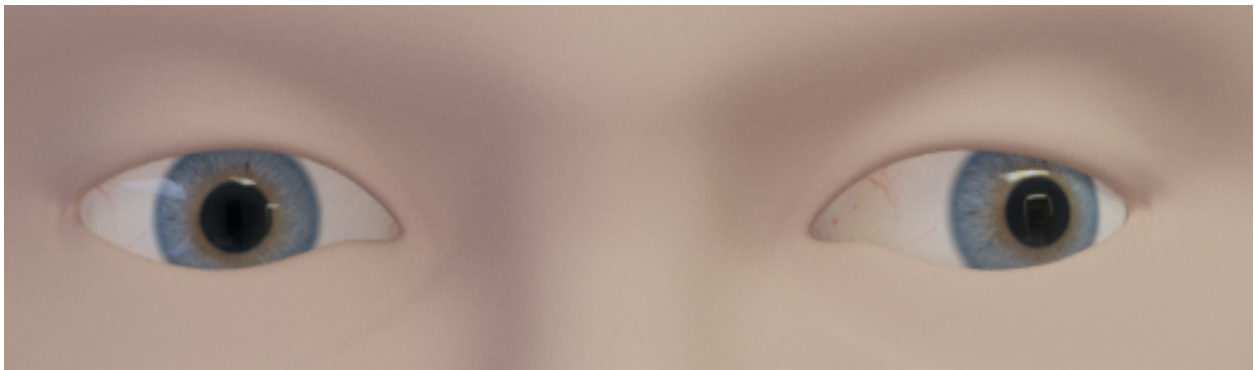
En metod utvecklades för att skapa detaljerade 3D-modeller av ögon och ett huvud genom att använda programvarorna Blender [58] och Adobe After Effects (AE) [59]. Modellen konstruerades för att kunna simulera realistiska ögonrörelser och generera syntetisk data. En serie videoklipp producerades där ögonmodellerna integrerades i ett syntetiskt huvud där animationen återspeglade de ögonrörelser som observeras vid olika typer av stroke. Detta genom att en 3D modell av ett öga samt huvud med ögonlocken öppna och tomrum för ögonen laddades ner från animationsstudio Turbosquid [60]. Dessa laddades upp i Blender där ögon placerades på korrekt plats i ansiktsmodellen. Då ögonen var två separata enheter styrdes de oberoende av varandra och videoklipp skapades genom att spara koordinater för ögonen vid särskilda nyckelbilder (keyframes). För att simulera olika symptom ändrades koordinaterna i X- samt Y-led för att uppnå önskade ögonrörelser. Videor av klassificering 0, 1 och 2 skapades och simulerade ögonrörelser för ingen blickpares, partiell blickpares och komplett blickpares. Utöver detta skapades även videor av klassificering 1 som simulerade symptom av nystagmus och isolerad blickpares, dessa gjorde upp 55% av klass 1 syntetiska videor. Alla videoklipp renderades med en längd på 160 bilder och en hastighet av 30 bilder per sekund, vilket gjorde dem alla 5.33 sekunder långa.

Efter renderingen i Blender genererades sammanlagt 65 mappar, där varje mapp innehöll 160 bilder. Dessa bilder omvandlades till videosekvenser i MP4-format med hjälp av AE. Inom AE skapades en ny komposition för varje bildsekvens. Dessa kompositioner renderades sedan sekventiellt för att exporteras som enskilda MP4-filer, vilket resulterade i 65 högkvalitativa syntetiska videosekvenser där alla var lika långa och med samma videoinställningar. Antalet videor för varje klassifikation och datatyp ses i tabell 15.

I figur 9 är en bild på modellen som användes för att skapa datan. I figur 10 visas ett exempel på hur det kan se ut när ögonen styrs separat.



Figur 9: 3D modell, syntetisk data.



Figur 10: Exempel på hur ögonen styrs separat.

Datotyp	Totalt	Klass 0	Klass 1	Klass 2
Inspelad	99	37	24	38
Syntetisk	65	20	18	27
Kombinerad	164	57	42	65

Tabell 15: Mängd använda videor för alla tre datatyper.

Videobearbetning

Efter processen av att samla både syntetisk och verklig data med alla olika klassificeringar för att träna modellerna bestämdes det för att den verkliga videodatan behövde konverteras för att passa ihop med den syntetiska i dess video inställningar. Videodatan hade blivit inspelat med 50 bilder per sekund medan den

syntetiska videodatan hade renderats till 30 bilder per sekund. På grund av denna skillnad i videoinställningar behövde all verklig inspelad videodata renderas och exporteras om och i den processen sänkas till 30 bilder per sekund. Denna process utfördes med Adobes programvara för videoredigering, Premiere Pro [61].

5.1.3 Dataanalysering

Ett öppen källkod-program (open source), GazeTracking, tillgängligt på GitHub har utnyttjats [62]. Programvaran använder algoritmer för bildbehandling och ML för att analysera live feed videos och identifierar riktningen användaren håller blicken samt ifall användaren blinkar. De fyra klassificeringar GazeTracking utför demonstreras i figur 11.

Denna process utför den genom att analysera pixelmönster, ögonens position och rörelser. Programmet ger hela tiden ut X- och Y-koordinater för höger respektive vänster pupill. Programmet är byggt med hjälp av OpenCV, Numpy och dlib varav OpenCV och dlib är viktiga bibliotek för bildbehandling och ML.

GazeTracking-programmet är uppbyggt av bildanalysfunktioner som kontinuerligt bearbetar en inkommande video för att detektera och följa ögonens positioner. Detta inkluderar steg för att först identifiera ansiktet i videon, därefter lokalisera ögonen och slutligen används modeller för att estimeras blickriktningen. Blinkningar kan identifieras genom att systemet observerar förändringar i ögonlockens position över tid för båda ögonen.

När videodatan var insamlad konverterades den till CSV-filer för att bli användbar ur ett ML syfte. För att göra detta skapades ett Python-program som tar utdatan från GazeTracking-systemet, alltså koordinaterna för respektive pupill, klassificering av blickriktning (höger, vänster samt blinkning) och lagrar det i en CSV-fil. Programmet gör detta för varje bild i videon och varje bilds utdata sparas på en egen rad, se figur 12. Programmet lade även till klippets klassificering som första tecken i filnamnet för att enkelt kunna komma åt den, se figur 13.



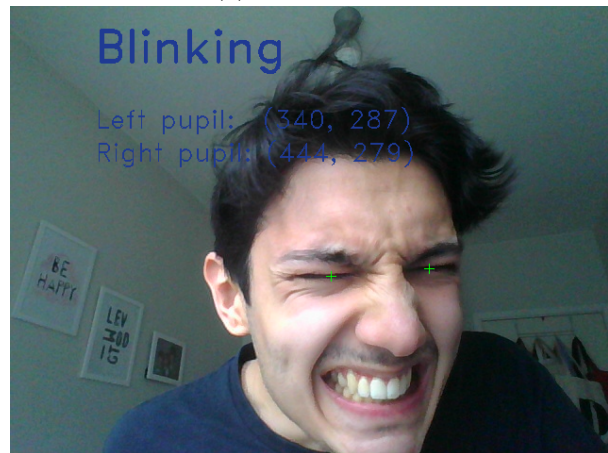
(a) Tittar framåt



(b) Tittar vänster



(c) Tittar höger



(d) Blinkar

Figur 11: De fyra olika klassificeringar GazeTracking-programmet gör.

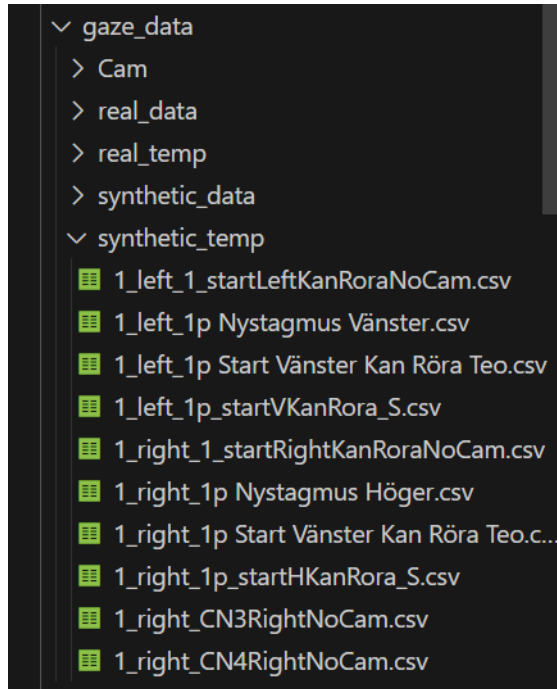
```

1 video_name,left_pupil,right_pupil,gaze_direction
2 C0001.MP4,"(860, 491)","(1106, 499)",center
3 C0001.MP4,"(862, 492)","(1106, 499)",center
4 C0001.MP4,"(861, 491)","(1105, 500)",center
5 C0001.MP4,"(863, 492)","(1107, 501)",center
6 C0001.MP4,"(863, 493)","(1107, 499)",center
7 C0001.MP4,"(862, 493)","(1108, 500)",center
8 C0001.MP4,"(864, 494)","(1108, 501)",center
9 C0001.MP4,"(865, 494)","(1109, 501)",center
10 C0001.MP4,"(865, 494)","(1110, 502)",center
11 C0001.MP4,"(865, 494)","(1110, 502)",center
12 C0001.MP4,"(865, 494)","(1108, 501)",center
13 C0001.MP4,"(865, 494)","(1109, 501)",center
14 C0001.MP4,"(866, 494)","(1109, 501)",center
15 C0001.MP4,"(865, 495)","(1109, 501)",center
16 C0001.MP4,"(866, 495)","(1109, 501)",center
17 C0001.MP4,"(867, 495)","(1111, 501)",center
18 C0001.MP4,"(868, 494)","(1118, 502)",center
19 C0001.MP4,"(868, 494)","(1119, 502)",center
20 C0001.MP4,"(869, 494)","(1119, 501)",left
21 C0001.MP4,"(869, 494)","(1112, 501)",center
22 C0001.MP4,"(869, 495)","(1113, 501)",center
23 C0001.MP4,"(870, 493)","(1119, 501)",center
24 C0001.MP4,"(870, 495)","(1115, 501)",center
25 C0001.MP4,"(870, 494)","(1114, 501)",center
26 C0001.MP4,"(871, 495)","(1120, 501)",left
27 C0001.MP4,"(872, 495)","(1122, 501)",left
28 C0001.MP4,"(873, 494)","(1121, 502)",left
29 C0001.MP4,"(874, 494)","(1117, 501)",left
30 C0001.MP4,"(874, 495)","(1117, 501)",center

```

Figur 12: Hur början av en klass 2 CSV-fil ser ut.

Varje video sparades i en egen CSV-fil, vilket underlättar felsökning, hantering och sortering av data och gör det enklare att isolera specifika typer av videor eller ögonfel för fördjupade studier. Denna uppdelning minskar även risken för datakorruption, eftersom skador på en fil inte leder till förlust av all insamlad data. Genom att ha data distribuerad över flera filer, kan specifika dataset enkelt väljas ut för olika forskningssyften utan att behöva bearbeta ett omfattande dataset, vilket ger en hög grad av flexibilitet när det gäller val av data för experiment och analyser.



Figur 13: Listan av mappar där CSV-filer placeras.

5.1.4 Algoritmer

Implementeringen börjar med att definiera sökvägar till de CSV-filer som genererats från videoanalysen. Dessa filer innehåller klassificering av blicken samt koordinaterna för båda pupillerna, vilket är avgörande för att träna modellerna.

Efter en omfattande granskning av den insamlade datan från CSV-filerna insågs det att en ytterligare transformation var nödvändig för att öka datans tolkbarhet. Transformationen involverade att konvertera X- och Y-koordinaterna för båda pupillerna till differenser mellan varje bilds koordinater, se figur 14.

$$\begin{bmatrix} 2. & 1. & 0. & 0. \\ -1. & -1. & -1. & 1. \\ 2. & 1. & 2. & 1. \end{bmatrix}$$

Figur 14: Exempel på omvandling av de fyra första raderna i figur 12 till differenser.

I vissa modeller var det nödvändigt att all data skulle ha samma längd för att den skulle kunna behandlas effektivt. För att uppnå detta fylldes kortare datasekvenser ut med ett påhittat värde, som inte förekommer naturligt i datamängden, tills de når önskad längd. Under analysen programmerades modellerna att ignorera de påhittade värdena som användes för att förlänga datasekvenserna.

Varje modell testades med tre olika datatyper. En med bara inspelad data, en med bara syntetisk data och en där typerna kombineras. Vid testning av inspelad data, tränades och testades modellen enbart med inspelad data (99st datafiler). Då modellerna testades med syntetisk data tränades den med all tillgänglig

syntetisk data (65st datafiler) och sedan testades den med all den inspelade datan. Vid det kombinerade fallet tränades och testades modellerna med kombinerad data (164st datafiler varav 60% var inspelad data).

För att säkerställa robustheten och tillförlitligheten hos de olika modeller som används i denna studie, har korsvalideringsmetoden K-delad korsvalidering (K-fold crossvalidation) tillämpats, detta med hjälp av `KFold` metoden från Pythonbiblioteket `sklearn.model_selection`. Denna valideringsmetod användes för att verifiera generaliserbarheten hos algoritmerna över olika delmängder av data. I arbetet implementerades 3-delad korsvalidering för alla algoritmer. Vid 3-delad korsvalidering innebär att datamängden delades upp i tre delmängder, där varje del i tur och ordning fungerade som ett valideringsset medan de övriga två delarna användes som träningsdata. Detta förfarande upprepades tre gånger, vilket säkerställde att varje datapunkt användes både för träning och validering. Prestandamått räknas ut för varje iteration och ett medelvärde av dessa tas. Exempel på hur 3-delad korsvalidering fungerar presenteras i punktlistan nedan:

- **Modell 1:** Tränad med del 1 + del 2, testad med del 3.
- **Modell 2:** Tränad med del 2 + del 3, testad med del 1.
- **Modell 3:** Tränad med del 1 + del 3, testad med del 2.

I flera av modellerna användes även Adam-optimiseraren, en optimeringsmetod för stokastisk gradientnedstigning [63] som genom iterativ träning uppdaterar modellvikterna vilket förbättrar modellens förmåga att klassificera data. Adam underlättade då det enkelt går att manipulera parametrar såsom inlärningshastighet. Alla modeller, med SVM som undantag, uppnådde ett värde för inlärningshastigheten genom att testa flera olika värden, både höga och låga. En högre inlärningshastighet får förlustvärdet hos modellen att inte konvergera, vilket leder till ett misslyckande i inlärningsprocessen, och en lägre inlärningshastighet gör hela processen onödigt långsam [64].

Hyperparameteroptimering genomfördes av samtliga algoritmer för säkerställa bästa resultat. De finns många faktorer bland algoritmer som påverkar deras prestanda, som exempelvis lärandehastighet, antal iterationer, och storleken på träningssetsen. Optimeringsprocessen gick inte till systematisk utan de flesta algoritmer testades på värden sporadiskt under kodningen.

CNN

Under utvecklingen av denna maskininlärningsalgoritm användes data uppdelade i 80% för träning och 20% för testning. Modellen inleds med ett Gaussianskt brus-lager (Gaussian noise-layer) som introducerar en liten mängd brus (0.01 standardavvikelse) i indata för att förbättra modellens robusthet genom att simulera variationer i verkliga data. Den första lagret i nätverket är ett 1D-konvolutionslager (Conv1D) med 64 filter och en kärnstorlek på 3. Detta lager använder ReLU (Rectified Linear Unit) som aktiveringsfunktion och är utrustat med L2-regularisering för att minska risken för överanpassning (overfitting) genom att begränsa storleken på vikterna [65].

Efter varje konvolutionslager följer ett satsnormalisering-lager (Batch Normalization) som normaliserar aktiveringarna från det föregående lagret för att påskynda träningen och förbättra konvergens. Efter normaliseringen tillämpas MaxPooling med poolstorlek 2 för att minska dimensionen på data och framhäva de viktigaste egenskaperna, följt av ett avhopp-lager (dropout layer) med en frekvens på 50% för att ytterligare förhindra överanpassning genom att slumpmässigt nollställa en del av neuronernas utgångar under träningen [66].

Det andra konvolutionslagret följer ett liknande mönster men med 128 filter för att djupare bearbeta och extrahera mer komplexa mönster från indata. Efter ytterligare maxpooling och avhopp appliceras ett

tillplattnings-lager (flatten-layer) som omvandlar den flerdimensionella utdatan från konvolutionsnätverken till en endimensionell vektor [67]. Denna vektor matas sedan in i ett fullt anslutet nätverk med 50 neuroner, också med ReLU-aktivering och L2-regularisering.

Slutligen avslutas modellen med ett kompakt-lager (Dense-layer) bestående av tre enheter med mjukmax-aktivering (softmax) för att utföra klassificeringen. Modellen använde en inlärningshastighet på 0.0001 och ett klippvärde (clipvalue) på 0.5 för att förhindra att gradientuppdateringarna blir för stora. Som förlustfunktion används kategorisk korsentropi.

DNN

DNN är en djup maskininlärningsmodell som bygger på arkitektur med flera lager, vilket möjliggör identifiering och inlärning av allt mer abstrakta mönster i data [68]. En stor fördel med DNN är dess förmåga att prestera bra på komplexa uppgifter såsom bildigenkänning och sågs därför som en bra kandidat till detta arbete [46]. Denna algoritmen består av flera gömda lager som bidrar till att modellen kan lära sig från komplex data [68]. DNN-modellen som skapades bestod av 3 tätt anslutna lager, med 64, 128 och 256 enheter och ReLU som aktiveringsfunktion. ReLU aktiveringen väljs för att effektivt hantera icke-linjäriteter i datan. Till modellen läggs även avhopp-lager efter varje kompakt-lager med en frekvens på 50%. Detta ökar modellens förmåga att generalisera bättre till ny data genom att förhindra att modellen blir för beroende av specifika egenskaper i träningsdatan. Det sista lagret i modellen är ett kompakt-lager med en mjukmax-aktivering, som är lämplig för multiklassklassificering. Under utvecklingen användes data uppdelade i 70% för träning och 30% för testning.

För att förbereda datan för modellen, extraheras och normaliseras funktionerna från CSV-filerna. Normaliseringen utförs med hjälp av *StandardScaler*, vilket säkerställer att datan har en lämplig skala för neural nätverksträning.

GRU

I denna studie testades GRU, en variant av RNN som är designad för att effektivare hantera sekvensdata genom sina uppdaterings- och återställningsgrindar [50].

Modellen implementerades med hjälp av Keras biblioteket i TensorFlow [69] och bestod av tre lager. Det första GRU-lagret bestod av 128 enheter och returnerade sekvensdata för vidare bearbetning. Detta följdes av ett andra GRU-lager med 64 enheter och sedan ett tredje GRU-lager med 32 enheter, där endast den sista utmatningen i sekvensen returnerades. Detta möjliggör en kondensering av all tidigare information i sekvensen till en enda vektor, vilket underlättar det slutliga klassificeringssteget [70]. Modellen avslutas med ett kompakt-lager bestående av tre enheter, vilket motsvarar antalet klasser i målvariabeln, med en mjukmax-aktiveringsfunktion för att generera en sannolikhetsfördelning över klasserna [71]. Modellen använde en inlärningshastighet på 0.0008. Som förlustfunktion användes kategorisk korsentropi, vilket är lämpligt för flerklass-klassificeringsproblem. Även här användes data uppdelade i 70% för träning och 30% för testning.

LSTM

LSTM-nätverk, en typ av återkommande RNN, har förmåga att lära sig och komma ihåg långa sekvenser av data [72]. Till skillnad från traditionella RNN:er kan LSTM lära sig beroenden från långa inmatningssekvenser. Denna egenskap gör LSTMs särskilt lämpade för tidsseriedata och sekvenser där kontext och temporal sekvens är avgörande för att förstå inmatningen. En fördel med LSTM-nätverk ligger i deras komplexa arkitektur som inkluderar flera grindar (inmatnings-, glömska- och utmatningsgrindar) som reglerar informationsflödet [49]. Dessa grindar bestämmer vilken information som behålls eller tas bort vilket gör att

nätverket kan behålla relevant information över långa perioder medan data som inte längre är användbar glöms bort.

Den använda datan delades in i 70% för träning och resterande 30% för testning. Modellen inkluderar ett maskeringslager (masking layer) för att ignorera de påhittade datapunkterna, vilket rensar bort oönskade ingångsvärden. Modellen innehåller tre lager av dubbelriktade LSTM nätverk. Det första av dessa lager har 128 neuroner, medan de efterföljande två lagren har 64 neuroner vardera. Mellan dessa LSTM-lager implementerades avhopp-lager med en avhoppningsfrekvens på 50%. För att ytterligare motverka överanpassning och förbättra modellens generaliseringsförmåga används kernelregularisering. Modellen använde en inlärningshastighet på 0.0005 och använde kategorisk korsentropi som förlustfunktion.

SVM

SVM är en övervakad maskininlärningsalgoritm som huvudsakligen används för klassificering och regressionsuppgifter [41][43]. Den sticker lite ut från de övriga ML-algortimerna och faller under kategorin linjära klassificerare, och grundidén går ut på att med hjälp av SVM-algoritmen hitta ett så kallat hyperplan. Det är hyperplanet som separerar datapunkterna och klassificerar dessa i olika klasser.

Den matematiska modellen för en linjär SVM beskrivs genom att först definiera ett hyperplan. Hyperplanet blir konstruerad med hjälp av den så kallade beslutsfunktionen som är definierad i ekvation 1.

$$f(\mathbf{x}) = \text{sign}((\mathbf{w}^T \mathbf{x}) + b) \quad (1)$$

\mathbf{w} motsvarar för viktvektorn, \mathbf{x} är en datapunkt eller träningsdata och slutligen b är biasen (basvektorn). Funktionen $f(\mathbf{x})$ tilldelar ett värde antingen +1 eller -1 (observera sign) till varje enskild datapunkt. Det som avgör värdet är på vilken sida om hyperplanet de befinner sig.

Den andra aspekten av SVM är att säkerställa att klasserna har ett så stort avstånd från varandra som möjligt. Detta går ut på att maximera marginalen. Marginalen maximeras genom matematisk minimera storleken (normen) på viktvektorn $\|\mathbf{w}\|$. Det är viktigt att alla datapunkter har blivit korrekt klassificerade och att de har en marginal som minst är 1. Detta verkställs genom ekvation 2.

$$y_i((\mathbf{w}^T \mathbf{x}_i) + b) - 1 \geq 0, \quad \forall i..n \quad (2)$$

där y_i står för datapunkt i . Genom att minimera $\|\mathbf{w}\|$ och att följa dessa ovannämnda regler så hittas det optimala hyperplanet. När marginalen är maximal och hyperplanet är hittat så kallas datapunkterna som ligger närmast hyperplanet för stödvektorer, eftersom de är avgörande för placeringen av hyperplanet.

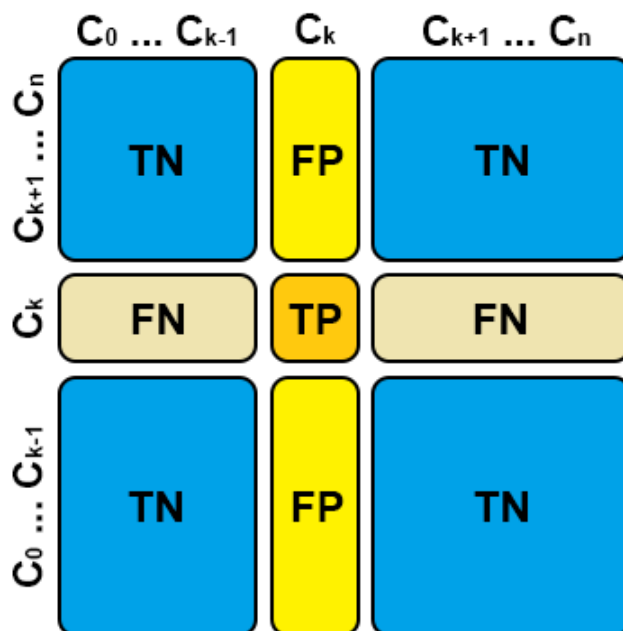
I fallet av ögonspårning så var SVM en bra kandidat som algoritm, och användningsområdet var en typisk situation där SVM brukar bli anpassad [40]. Detta var tydligt under kartläggningstudien i litteraturstudien. Om man algoritmen anpassas linjärt så blir blir kernelanpassning irrelevant. Planet behöver ej transformeras till högre dimensioner för att hitta ett hyperplan, och linjen motsvarar endast en rät linje. Mer avancerade algoritmer brukar öka komplexiteten och beräkningstiden markant. Vanliga kernelanpassningar som brukar användas är RBF(Radial Basis Function), polynomial och sigmoid [73]. Samtliga av dessa fyra av dessa olika anpassningar var studerade för att optimera SVM-algoritmen.

Kodmässigt så användes biblioteket sklearn för att implementera SVM. Genom sklearn kunde flera olika typer av kärnanpassningar appliceras effektivt. Hyperparameteroptimering, korsvalidering och alla olika typer av kvantitativa mätal kunde också appliceras genom sklearn. Vidare så är vanlig praxis för SVM är att ha 70% träningsdata och 30% testdata[42], och således blev detta det naturliga valet.

5.1.5 Testning

De prestandamått som används för att utvärdera modellens effektivitet inkluderar noggrannhet, precision, känslighet, specificitet och F1-poäng. Dessa mått ger insikt i modellens förmåga att korrekt identifiera och klassificera strokesymptom baserat på ögonrörelsedata. För att räkna ut dessa prestandamått används begreppen som förklaras och visualiseras nedan i figur 15:

- **Korrekt positiv (True Positive, TP)** för C1: Antalet C1-instanser som modellen korrekt klassificerar som C1.
- **Korrekt negativ (True Negative, TN)** för C1: Antalet icke-C1-instanser som modellen korrekt klassificerar som icke-C1.
- **Fel positiv (False Positive, FP)** för C1: Antalet icke-C1-instanser som modellen falskt klassificerar som C1.
- **Fel negativ (False Negative, FN)** för C1: Antalet C1-instanser som modellen falskt klassificerar som icke-C1.



Figur 15: Visualisering av mätvärdena TP, TN, FP och FN för icke-binär klassificering.

Dessa termer används för att beräkna de olika prestandamått enligt följande:

Noggrannhet (Accuracy): Noggrannhet mäter andelen korrekt klassificerade exempel (både TP och TN) i förhållande till det totala antalet exempel [74]:

$$\text{Noggrannhet} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Precision mäter andelen korrekt positiva klassificeringar bland alla positiva klassificeringar som modellen gör [74]:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Känslighet (sensitivity): Känslighet, eller sensitivity, mäter modellens förmåga att identifiera alla relevanta fall av den positiva klassen [74]:

$$\text{Känslighet} = \frac{TP}{TP + FN}$$

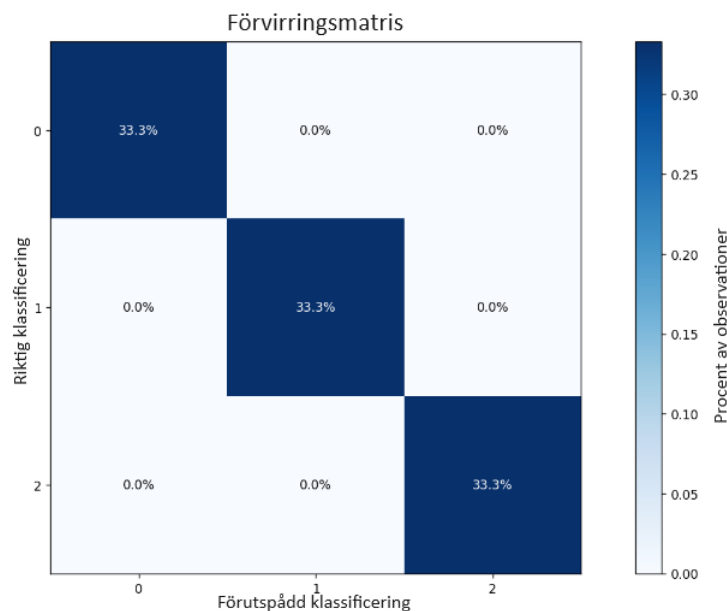
Specificitet: Specificitet, även känt som 'True Negative Rate', mäter modellens förmåga att korrekt identifiera negativa fall, det vill säga hur bra modellen är på att undvika falska positiva klassificeringar [74]:

$$\text{Specificitet} = \frac{TN}{TN + FP}$$

F1-poäng: F1-poängen är medelvärdet av precision och sensitivitet, vilket hjälper till att balansera mellan de två när det finns en klassobalans [74]:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Känslighet}}{\text{Precision} + \text{Känslighet}}$$

Då modellen klassificerar 0, 1 och 2 innebär det att den ej är binär och kan därför inte enbart klassificera stroke och inte stroke. Varje klass hanteras då separat och ett medelvärde tas över klasserna för att få macrovärdet över alla klasser [74]. Värdena för TP, TN, FP, och FN anpassas för varje specifik modell baserat på deras faktiska utfall. För att illustrera olika modellers resultat används nedanstående förvirringsmatriser och i figur 16 är ett exempel på hur en ideal sådan ser ut.



Figur 16: Exempel på ideell förvirringsmatris.

Algoritmerna testas även på deras förlust, som är ett mått på hur bra eller dåligt en algoritm presterar på sina förutspådda klassificeringar jämfört med de faktiska resultaten [75]. Förlusten är en indikation på avståndet mellan den förutsedda utgången och den verkliga utgången för varje datapunkt i träningsdatamängden [76]. Valideringsförlust och träningsförlust är två viktiga mått på prestanda där träningsförlusten mäter hur väl modellen presterar på träningsdatamängden, medan valideringsförlusten mäter prestandan på en separat valideringsdatamängd som modellen inte har sett under träning [75]. En låg träningsförlust indikerar att

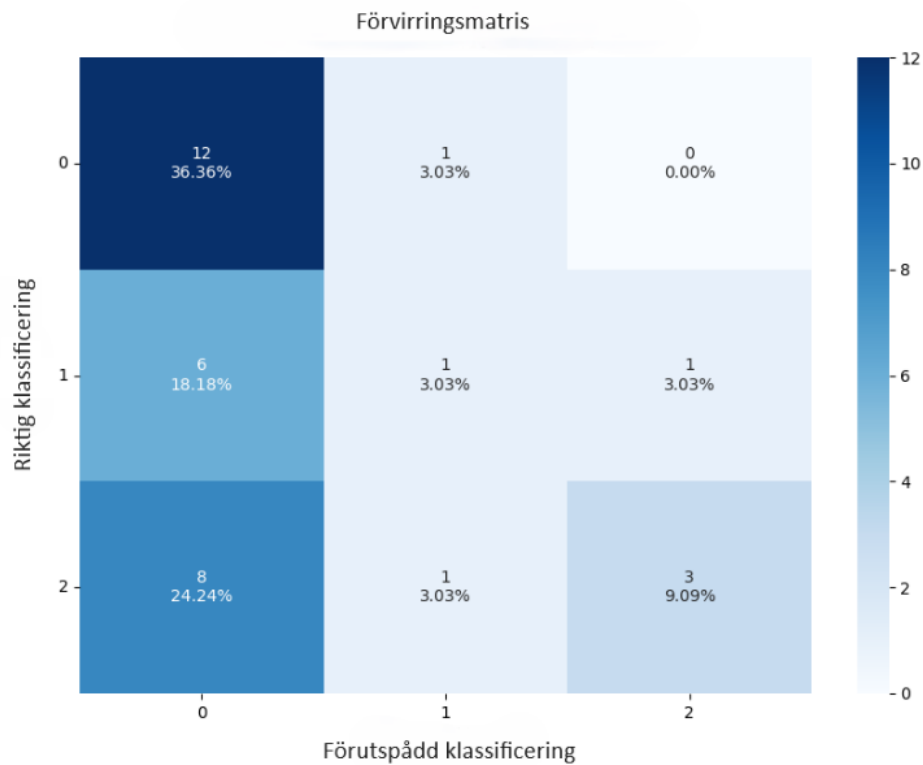
modellen presterar väl på träningsdatamängden, medan en låg valideringsförlust indikerar att modellen generaliserar väl till ny data [75]. Samtliga algoritmer evalueras på deras förluster för att se ifall en modell lider av överanpassning.

5.2 Resultat

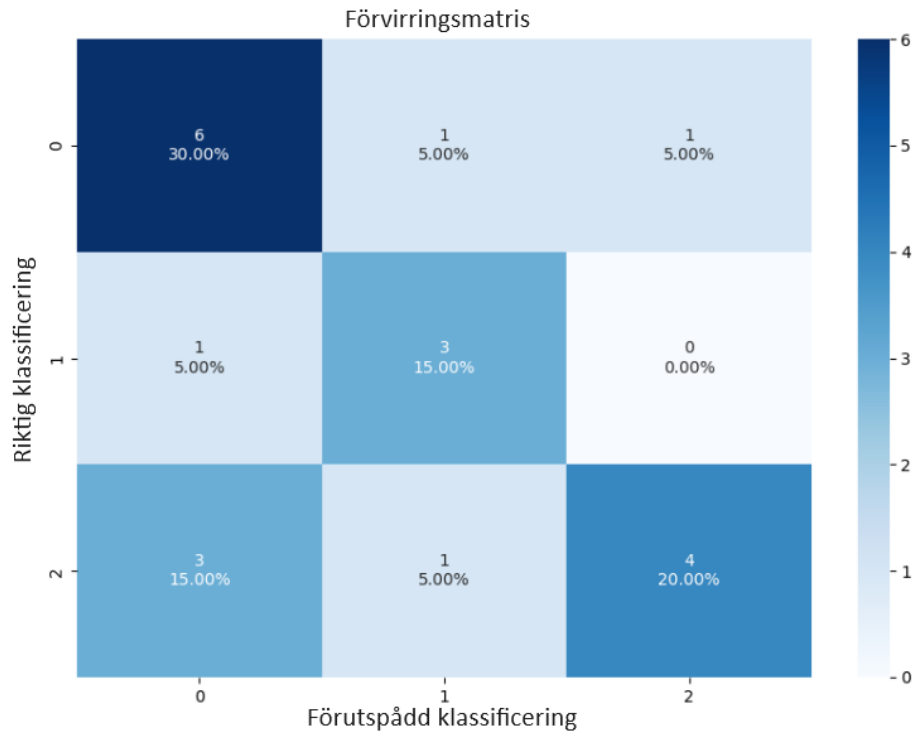
I detta avsnitt presenteras projektets resultat tillsammans med tillhörande tabeller och figurer. Systemets prestanda för samtliga algoritmer tas upp och slutligen redovisas dessa tillsammans i en gemensam tabell. Modellernas dataset bestod av 164 videor där 99 var inspelade och 65 syntetiskt skapade. Modellerna tränades och testades på både kombinerad och separerad användning av syntetisk och inspelad data.

CNN

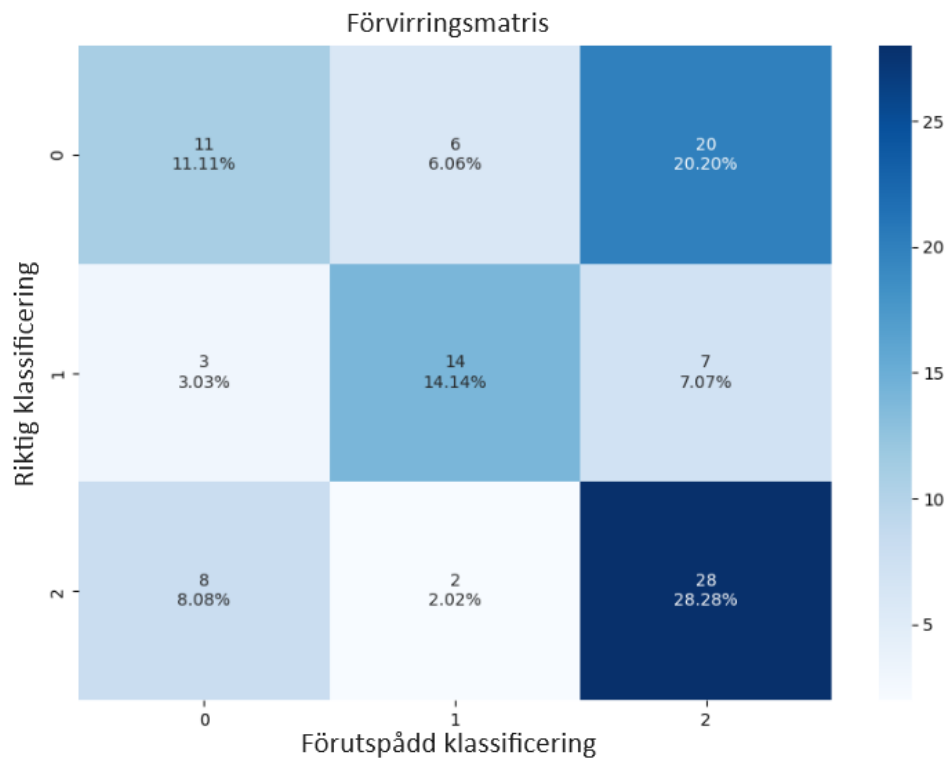
Modellen som använt CNN visade en varierande prestanda över träning på de olika dataseten där förvirringsmatriserna från träning med kombinerad data, inspelad data och syntetiska data återfinns i figur 17, 18 respektive 19. Modeller som tränats på olika data hade alla svårt att klassificera rätt samt att skilja på klass 0 & 2.



Figur 17: Förvirringsmatris för CNN-modellen med kombinerad data.



Figur 18: Förvirringsmatris för CNN-modellen med inspelad data.



Figur 19: Förvirringsmatris för CNN-modellen med syntetisk data.

Prestandamåtten för CNN-modellen presenteras i tabell 16, medan information om validerings- och träningsförlusterna visas i tabell 17. Modellen presterade överlägset bäst när den tränades på inspelad data, där den uppnådde en noggrannhet på 65.0%. Denna prestanda jämförs med resultaten från träning på syntetisk data och verklig data, vilka uppnådde noggrannheter på 53.53% respektive 48.48%. Träningsförlusten var högt för inspelad data med 0.82 medan syntetisk data uppnådde 0.69 och kombinerad data 0.55. Å andra sidan var valideringsförlusten för inspelad data lägst med 0.94 medan syntetisk data uppnådde 1.09 men och kombinerad data 1.75.

Datotyp	Noggrannhet	Precision	Känslighet	F1-Poäng	Specificitet
Inspelad	65.0%	66.67%	66.67%	64.96%	81.94%
Syntetisk	53.53%	54.83%	53.9%	52.83%	75.77%
Kombinerad	48.48%	51.5%	43.27%	39.07%	72.41%

Tabell 16: Prestanda för CNN-modellen med olika datatyper.

Datotyp	Valideringsförlust	Träningsförlust
Inspelad	0.94	0.82
Syntetisk	1.09	0.69
Kombinerad	1.75	0.55

Tabell 17: Förlust för CNN-modellen med olika datatyper.

DNN

DNN-modellen visade varierande prestandamått vid olika tränings-sessioner. De bästa uppnådda resultaten är presenterade i tabell 18, medan förlusterna finns i tabell 19. DNN hade överlägset bäst prestanda när den tränades på kombinerad data, där den nådde en noggrannhet på 45.36%. Detta är en markant förbättring jämfört med prestanda på enbart inspelad data och syntetisk data, där noggrannheten var 30% respektive 31.31%. Förlusten för kombinerad data och syntetisk data är höga med stor skillnad mellan validerings- och träningsförlusterna där kombinerad data har en skillnad på 15.7. Inspelad data uppnådde däremot jämnare och värden mellan dess förlusttyper och en lägre valideringsförlust.

Datotyp	Noggrannhet	Precision	Känslighet	F1-Poäng	Specificitet
Inspelad	30.0%	31.71%	33.33%	30.33%	65.28%
Syntetisk	31.31%	30.73%	31.11%	30.80%	65.11%
Kombinerad	45.36%	41.74%	40.94%	41.34%	73.55%

Tabell 18: Prestanda för DNN-modellen med olika datatyper.

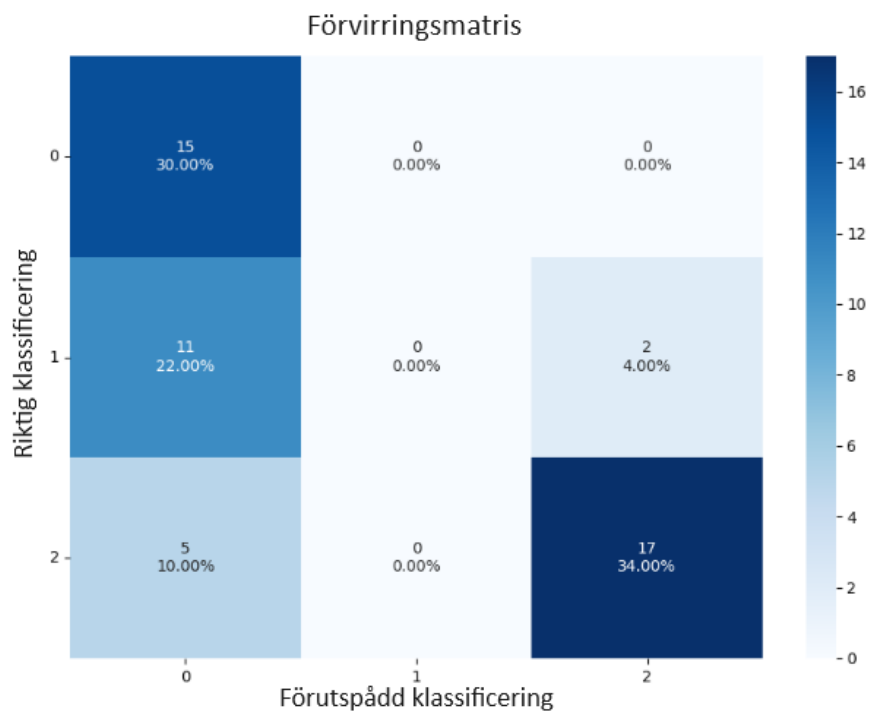
Datotyp	Valideringsförlust	Träningsförlust
Inspelad	1.38	1.27
Syntetisk	1.55	0.42
Kombinerad	15.8	0.01

Tabell 19: Förlust för DNN-modellen med olika datatyper.

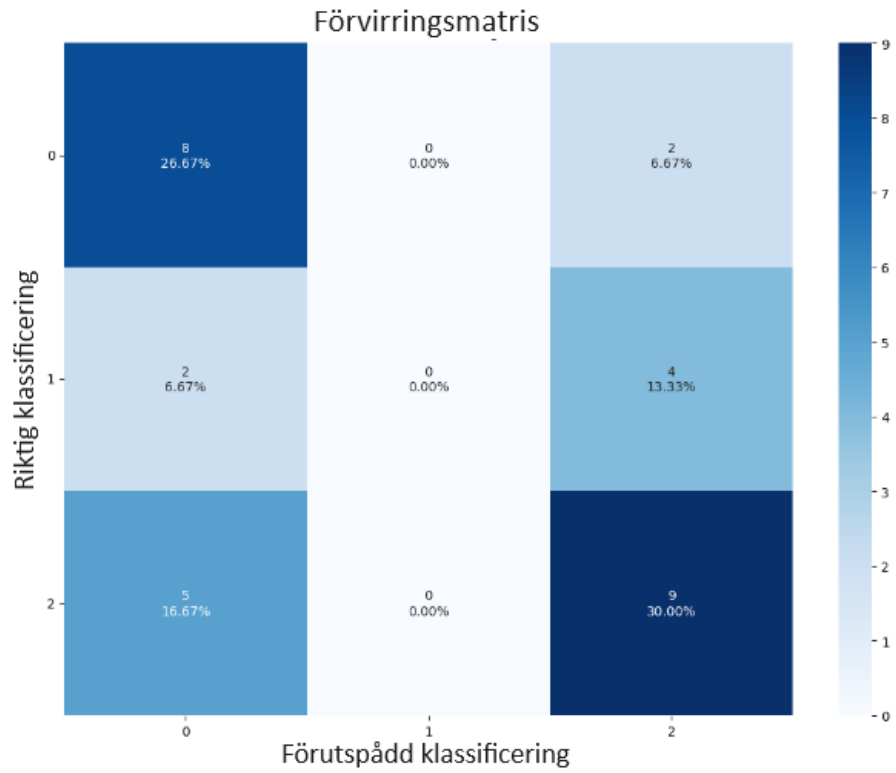
Resultaten från förvirringsmatriserna visade på väldigt varierande och ogynnsamma resultat. De observerade resultaten visade på en stor spridning av felklassificeringar och bristfällig noggrannhet hos modellen. Endast vid två tillfällen identifierades korrekt klass för majoriteten av testdatan, vid användning av kombinerad data identifierades klass 1 korrekt 57.14% av gångerna och klass 2 50% av gångerna.

GRU

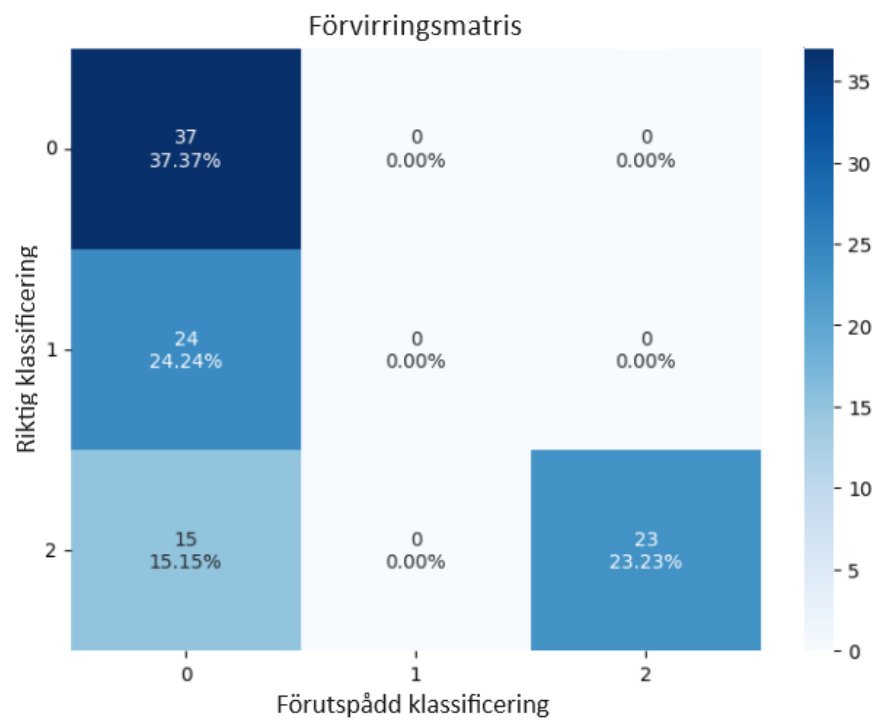
Förvirringsmatriserna för modellen baserad på GRU med träning på kombinerad data, inspelad data och syntetiska data presenteras i figur 20, 21 respektive 22. Genom att analysera dessa matriser framgår det att modellen har en viss förmåga att korrekt identifiera videor i klass 0 och klass 2. Däremot misslyckas den med att korrekt klassificera någon video av klass 1.



Figur 20: Förvirringsmatris för GRU-modellen med kombinerad data.



Figur 21: Förvirringsmatris för GRU-modellen med inspelad data.



Figur 22: Förvirringsmatris för GRU-modellen med syntetisk data.

I tabell 20 & 21 finns resultatet från GRU-modellen samt förlusterna för den. Högst resultat för GRU-modellen uppnåddes med kombinerad data med en noggrannhet på 64.0%, precision på 45.97%, en känslighet på 59.1%, en F1-poäng på 49.4%, och en specificitet på 82.4%. Detta innebär att modellen hade relativt jämna och höga resultat över samtliga prestandamått. Validerings- och träningsförlusten för GRU uppnådda nästan samma resultat över samtliga datatyper. Bäst presterade inspelad data med valideringsförlust 1.3 och träningsförlust 1.29. Kombinerad data presterade nästan lika med validerings- och träningsförlust på 1.32.

Datotyp	Noggrannhet	Precision	Känslighet	F1-Poäng	Specificitet
Inspelad	56.67%	37.77%	48.1%	42.03%	75.83%
Syntetisk	60.6%	49.57%	53.5%	46.97%	79.03%
Kombinerad	64%	45.97%	59.1%	49.4%	82.4%

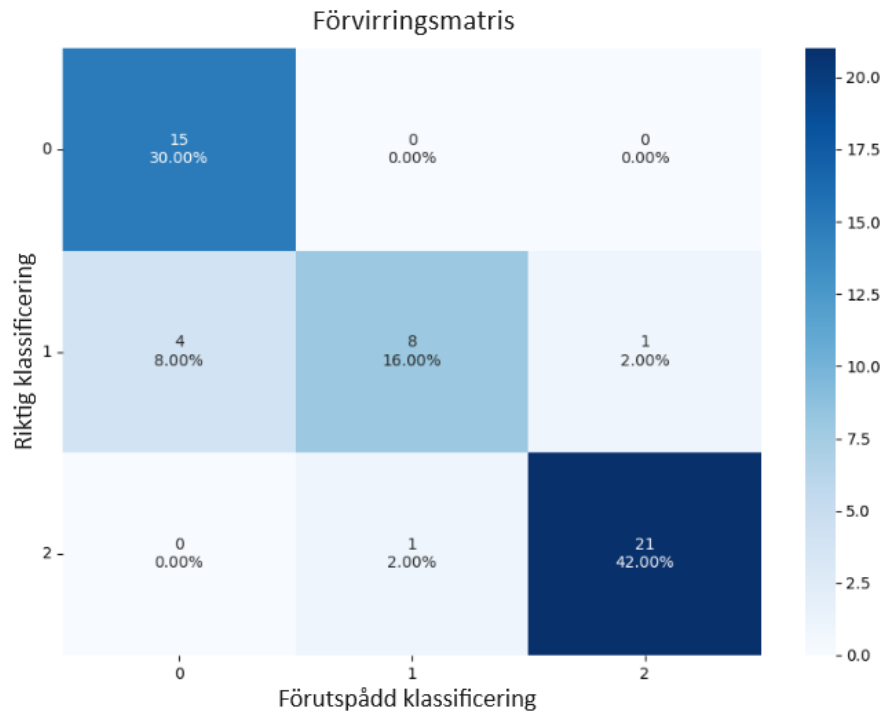
Tabell 20: Prestanda för GRU-modellen med olika datatyper.

Datotyp	Valideringsförlust	Träningsförlust
Inspelad	1.3	1.29
Syntetisk	1.33	1.53
Kombinerad	1.32	1.32

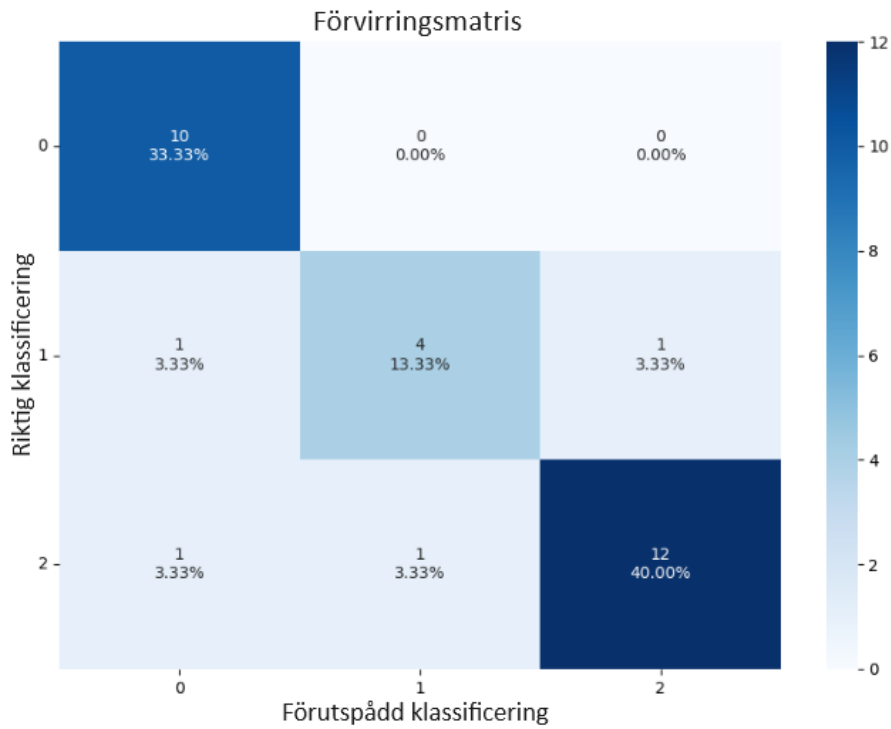
Tabell 21: Förlust för GRU-modellen med olika datatyper.

LSTM

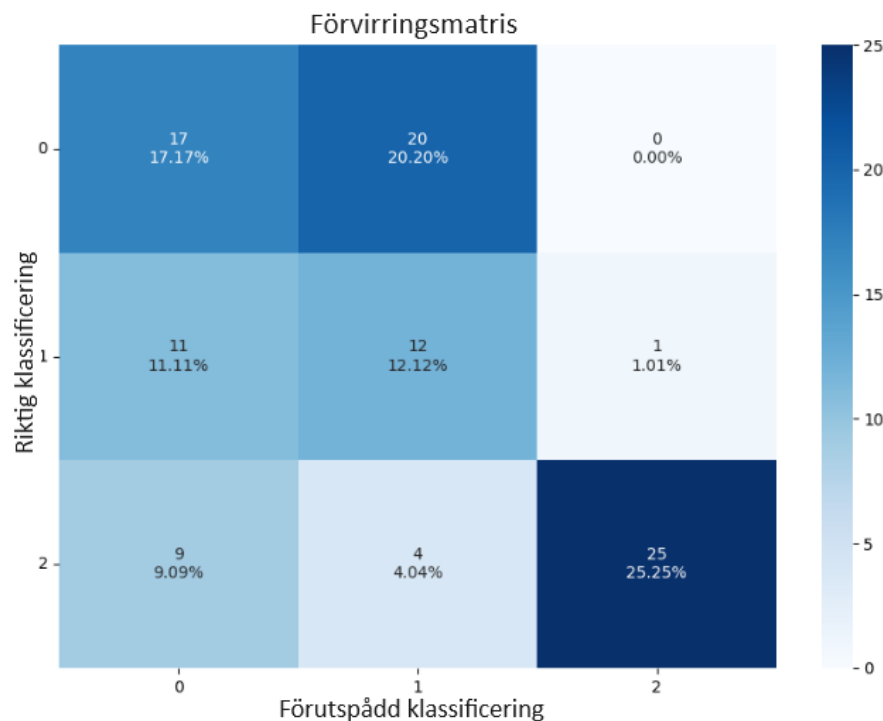
LSTM-modellen har implementerats för att effektivt hantera sekvensberoenden i ögonrörelsedata, vilket är avgörande för att identifiera dynamiska förändringar i pupillens position kopplade till olika strokeindikationer. Förvirringsmatriserna, som visas i figur 23 och 24, illustrerar modellens prestation när den tränats på kombinerad data respektive enbart inspelad data. Dessa matriser visar att modellen med relativt hög säkerhet kan klassificera alla klasser korrekt. Däremot, när modellen tränats enbart på syntetisk data, som presenteras i figur 25, uppvisar den svårigheter att med samma säkerhet klassificera alla klasser.



Figur 23: Förvirringsmatris för LSTM-modellen med kombinerad data.



Figur 24: Förvirringsmatris för LSTM-modellen med verklig data.



Figur 25: Förvirringsmatris för LSTM-modellen med syntetisk data.

Modellen uppvisade en bedömningsförmåga för stroke med en noggrannhet på 90.0% och en känslighet på 87.77% för kombinerad data. Med en precision på 85.67% och ett F1-poäng på 85.47% demonstrerar resultaten en hög tillförlitlighet i klassificeringen. Tabell 22 visar att modellen presterade bäst när den tränades på kombinerad data, med höga värden för noggrannhet och specificitet. Däremot gav träning på syntetisk data avsevärt lägre prestanda.

Tabell 23 belyser modellens förlustvärden under träning och validering över olika datatyper. Lägst förlustvärden nåddes av kombinerad data med 0.83 i valideringsförlust och 0.58 i träningsförlust. Förlustvärdena för syntetisk data och inspelad data var högre med relativt små skillnader mellan de olika förlusterna.

Datotyp	Noggrannhet	Precision	Känslighet	F1-Poäng	Specificitet
Inspelad	86.67%	85.2%	84.13%	84.17%	93.2%
Syntetisk	54.55%	58.46%	53.9%	54.73%	78.03%
Kombinerad	88.0%	87.77%	85.67%	85.47%	94.1%

Tabell 22: Prestanda för LSTM-modellen med olika datatyper.

Datotyp	Valideringsförlust	Träningsförlust
Inspelad	1.06	0.80
Syntetisk	1.40	1.43
Kombinerad	0.83	0.58

Tabell 23: Förlust för LSTM-modellen med olika datatyper.

SVM

Tabell 24 visar prestanda för SVM-modellen med polynomial-kärna över olika datatyper. De fyra olika varianter av SVM som testades var linjär, RBF, polynomial och sigmoid. Bland dessa presterade poly-kärnan i överlag bäst, och lyckades genomgående korrekt klassificera alla tre typer av data. En polynomial kärna använder ett polynom av given grad för att separera klasserna. I detta fallet använde den polynom av grad 2 för att klassificera inspelad och kombinerad data och ett polynom av graden 3 för att klassificera syntetisk data.

Linjär presterade exakt lika bra som poly-kärnan på verklig data men inte lika bra på den syntetiska och kombinerade datan. Linjär optimering hade särskilt låg noggrannhet på syntetisk data med ett resultat på 38.10%. RBF presterade allra bäst på syntetisk data med en noggrannhet på 80.95%. Däremot var dess precision (40.48%), känslighet (50.00%) och F1-poäng (44.74%) lägre, vilket tyder på en övergripande begränsning på att korrekt klassificera klasser av olika slag, och att datan är obalanserad. Sigmoid presterade sämst på klassificera i samtliga kategorier och således ointressant kernelanpassning för klassificering.

Korsvalidering var utnyttjat för att optimera algoritmen. I detta fallet var en 3-delad korsvalidering utnyttjad, vilket innebär att datamängden blir uppdelad i tre delar. De bästa korsvaliderings resultatet för poly-kärnan var 0.7 (inspelad data), 0.75 (syntetisk data) och 0.74 (kombinerad data). Hyperparameteroptimeringen gjordes för polynomialens "straffparameter" (C), graden, och slutligen gamma. Gamma hade värdena 0.01 (inspelad), 0.01 (syntetisk) och 0.1 (kombinerad). Straffparametern låg alltid på 1.

Datotyp	Noggrannhet	Precision	Känslighet	F1-Poäng	Specificitet
Inspelad	65.00%	65.87%	55.61%	58.52%	77.31%
Syntetisk	66.67%	50.79%	40.20%	44.85%	75.02%
Kombinerad	68.29%	50.62%	54.44%	52.01%	79.44%

Tabell 24: Prestanda för SVM-modellen med poly-kärna som presterade bäst med olika datatyper.

När resultatet från SVM-modelles klassificeringar analyserades kunde det uttydas att de prestanda-värden som den gett kan vara missvisande då test-datan bestod av en majoritet 1-klassad data, 73%. Modellen verkade dessutom klassificera majoriteten av 1- & 2-klassad data till klass 1 vilket innebär att den får en hög noggrannhet. Noggrannheten för att klassificera data med klass 0 och 2 när modellen tränats på kombinerad data var 20% respektive 33.33%.

5.2.1 Översikt över algoritmers prestanda

En översiktstabell som presenterar resultaten för varje modell när den är tränad på kombinerad data kan ses i tabell 25.

Modell	Noggrannhet	Precision	Känslighet	F1-Poäng	Specificitet
LSTM	90%	87.77%	85.67%	85.47%	94.1%
GRU	64%	45.97%	59.1%	49.4%	82.4%
SVM	68.29%	50.62%	54.44%	52.01%	79.44%
CNN	48.48%	51.5%	43.27%	39.07%	72.41%
DNN	45.36%	41.74%	40.94%	41.34%	73.55%

Tabell 25: Översikt av modellprestanda.

5.3 Diskussion

I detta avsnitt diskuteras först svagheter och styrkor med de olika algoritmerna och allt från insamlingen och genereringen av data till hanteringen av data. Därefter diskuteras framtida arbeten på projektet och potentiella utvecklingsområden. Slutligen diskuteras begränsningar i inspelad och syntetisk data.

5.3.1 Algoritmanalys

När man jämför resultatet med de resultaten i litteraturstudien, som kan ses i figur 6, finns det vissa likheter. Precis som förutspått så presterar DNN sämst bland de fem algoritmerna. Däremot så talade litteraturstudien för att de andra algoritmerna skulle prestera likartat, om inte SVM något sämre. Det fanns inget som talade för att LSTM skulle vara dominerade på något vis. Generellt så hade algoritmerna i litteraturstudien bättre resultat hos de kvantitativa måtten. Orsaken till denna skillnad är antagligen komplex och flerfaciterad, med allt från processad data till skillnad i användningsområde.

I planeringsfasen av projektet utforskades olika maskininlärningsalgoritmer för att effektivt analysera och klassificera de insamlade videodatamängderna. Initialt övervägdes enklare algoritmer såsom Decision Tree (DT) eller RF på grund av deras transparens och enkelhet i implementation. Emellertid stötte vi på utmaningar med dessa tillvägagångssätt, främst eftersom den tillgängliga datan består av sekvensiella observationer - varje video representerar en tidssekvens av ögonrörelser med varje rad från de konverterade CSV-filerna som en videoram med koordinater relativa till de andra. Simplare maskininlärningsmodeller såsom DT eller Naïve Bayes är inte kapabla att hantera sekvensiell data eller känna igen mönster över en följd av steg, vilket gör en analys av denna data med dessa modeller otillräcklig.

Med insikten om dessa begränsningar övergick vi till mer avancerade metoder för att säkerställa en noggrann och pålitlig dataanalys. Alla algoritmer utvärderades noggrant för att få fram pålitliga resultat, därför valdes k-delad korsvalidering. Användningen av 3-dels korsvalidering valdes med hänsyn till datamängdens storlek, där fler delar inte ansågs bidra med ytterligare värde. Denna metod möjliggjorde en balanserad utvärdering av modellernas prestanda över datamängden och reducerade risken för överanpassning. Genom denna metodik kunde pålitliga uppskattningar erhållas för modellernas noggrannhet, precision, känslighet, specificitet och F1-poäng.

En observation som fjordes är att skillnaderna mellan valideringsförlust och träningsförlust är relativt stora hos de flesta av våra algoritmer, med exempelvis LSTM vars förluster skiljer sig med 0.25 för kombinerad data. Anledningen på detta kan bero på det begränsade datasetet.

Tittar man på förvirringsmatrisen för CNN vid användandet av kombinerad data indikerar enligt figur 17 på att modellen har svårigheter att korrekt klassificera 0 och 1, men visar även svårigheter på att identifiera klass 2. Den visar däremot bättre förmåga att klassificera rätt då den endast tränats på inspelad- eller syntetisk data. Resultatet från figur 18 indikerar på att modellen kan identifiera skillnader mellan klass 0 och klass 1 och klassificera dem korrekt. Resultatet tyder även på att modellen har svårigheter med att identifiera klass 2 då den klassificerat många klass 2 som klass 0. Modellen visar på liknande resultat i figur 19, då den tränats på syntetisk data. Modellen identifierar skillnader mellan klass 1 och klass 0 men har klassificerat många klass 0 videor som klass 2, tvärtom från vid användandet av inspelad data.

DNN gav mycket varierande resultat varje gång modellen tränades. Förvirringsmatriserna och prestandan ändrades varje gång modellen testades och ingen konkret förklaring kunde hittas. Förvirringsmatriserna visade på väldigt varierande resultat och modellen kunde inte med säkerhet identifiera korrekt klass, speciellt då den tränades av enbart inspelad data. De dåliga resultaten kan bero på flera faktorer som obalanserad datamängd, komplexiteten hos klasserna eller brister i modellens arkitektur. Ytterligare en anledning till DNNs relativt dåliga prestanda kan vara att DNN inte hanterar sekvensiell data på samma sätt som RNN-algoritmer som är utformade just för att hantera sådana typer av data. Att modellen gav dåliga resultat

som ändrades mycket för varje gång den testades kan bero på att modellen bara gissar. Noggrannheten för inspelad och syntetisk data ligger båda runt 30%, i och med att bara tre olika klasser ges går det inte att utesluta att modellen inte lärt sig något från träningen. Då modellen tränades på kombinerad data visar den däremot på bättre resultat men på grund av den väldigt höga valideringsförlusten pekar det på att modellen lider av överanpassning.

I resultatet av LSTM-modellens prestanda tränad med olika datatyper framgår det att både den inspelade datan och den kombinerade datan ger goda resultat, vilket är synligt i förvirringsmatriserna och höga prestandamått. När det gäller träningen på endast syntetisk data uppvisar modellen stora svårigheter och har den relativt låga prestandan, med värden runt 55% för de flesta mått enligt tabell 22. Förvirringsmatrisen för modellen tränad på syntetisk data i figur 25 antyder att modellen främst har problem med att skilja mellan klass 0 och klass 1 men att modellen ofta korrekt igenkänner klass 2. Detta problem syns inte då den tränats på endast inspelad data, vilket tyder på att vissa egenskaper i de syntetiska videorna inte överförs väl till modellen, särskilt när det gäller att skilja mellan klass 0 och 1. Denna svårighet kan bero på att en viss typ av den syntetiska datan har mer särskiljbara symptom än andra typer. Specifikt partiell blickpares i båda ögon brukar följa samma mönster medan fel i kranialnerverna har relativt olika symptom. De varierande symptomen kan då påverka ML-modellens inlärningsprocess och bidra till sämre resultat. Denna observation av resultat visar begränsningarna med den genererade syntetiska datan och möjligtvis syntetisk data i allmänhet. Resultatet betonar därför behovet av noggrann konstruktion och validering av syntetisk data för att undvika att förlora viktiga nyanser som finns i inspelad data.

Resultatet antyder att LSTM-algoritmen är effektiv för att identifiera de olika klasserna, särskilt klass 2. Trots att träning på enbart inspelad data visar en något bättre förmåga att identifiera klass 0 jämfört med den kombinerade datan är skillnaden marginell. Denna skillnad kan delvis förklaras av en större mängd testdata när kombinerade data används, vilket kan ha bidragit till en ökad generalisering hos modellen. Dessutom framgår det att användningen av kombinerad data resulterar i högre noggrannhet och känslighet. Endast inspelad data har däremot en högre precision och F1-poäng. Trots att modellernas prestanda och deras förmåga att generalisera över olika datamängder är väldigt lika ses den tränad med kombinerad data som bättre. Anledningen till varför den modellens resultat väger tyngre är på grund av att den presterade på en lika hög nivå som då den tränats på endast inspelad data trots att ytterligare ögonrörelser som inte kunde imiteras fanns med i den syntetiska datan.

I GRUs förvirringsmatriser, figur 20, 21 och 22, kan det uttydas att den hade svårt att urskilja sekvenser med klass 1. Här misslyckas modellen, oavsett datatyp som användes för träning, med att korrekt identifiera någon sekvens av klass 1. Den felklassar även klass 2 som klass 0 väldigt frekvent när modellen tränades med syntetisk data och kombinerad data. Denna svårighet visar sig inte i lika stor grad vid användning av enbart inspelad data men prestandan är här överlag något sämre och visar på att den ej säkert kan klassificera videor av klass 2. Resultatet kan bero på att det finns en brist på inspelad data att använda och stärks ytterligare av att användning av kombinerad data presterar bättre än då bara inspelad data används.

Både GRU och LSTM uppvisar samma problem att de ej kan skilja mellan klass 0 och klass 1 korrekt, speciellt för GRU-modellen där problemen är större. Anledningen kan vara att LSTM har en högre komplexitet, vilket passar denna data bättre jämfört med GRUs lägre komplexitet. Att de båda har samma problem om än i olika grad kan peka på att vissa modellarkitekturer kan vara mer känsliga för kvaliteten och karaktären hos träningsdatan. Resultatet pekar dock på att LSTM-modeller har stor potential i applikationer som kräver noggrann analys av sekvensdata, vilket gör dem särskilt lämpliga för medicinska tillämpningar där tidiga och exakta diagnoser kan ha betydande påverkan på prehospital patientvård.

Anpassningen av SVM-algoritmens parametrar - huvudsakligen kärntypen (kernel) och klassvikterna - är centralt för hur algoritmen presterar. I fallet av SVM testades 4 olika typer av kernelanpassning: linjär, KBF, polynom och sigmoid. Det är intressant att kernelanpassningen av linjär och polynom presterade ex-

akt lika bra på den inspelade datan. Eftersom datan egentligen består av x- och y-värden för ögats position så är det rimligt att anta att datan går att separera med enklare SVM-algoritmer, alltså en linjär anpassning. Detta återspeglas följaktligen i att polynomial-kärnan presterar identiskt med den linjära anpassningen på den inspelade datan och att gamma var satt till 0.01 för polynomial-kärnan. Detta låga värde på gamma blir så att polynomial-anpassningen motsvarar en rät linje.

Valideringsförlust och träningsförlust är inte översättningsbart till SVM-sammanhang och vanligtvis inte sätt att som man använder för att beskriva förlust hos SVM-algoritmer. I de andra iterativa algoritmerna, speciellt i DNN eller CNN, är varje epok (epoche) den cykel genom hela datasettet därav vikterna hos modellen uppdateras. I SVM är konceptet av epok inte applicerbart. När hyperplanet är hittad så är algoritmen bestämd. Optimering handlar snarare om att framställa det bästa tänkbara hyperplanet i rätt dimension/rum som separerar klasserna med största marginalen.

LSTM-modellen har demonstrerat en förmåga att effektivt hantera sekvensberoende data, vilket är av yttersta vikt när man arbetar med ögonrörelsedata över tid. Denna förmåga att hantera sekvensdata med hög precision är avgörande för att identifiera subtila förändringar i pupillens position som kan indikera en stroke.

Förvirringsmatriserna, speciellt de för modellen tränad på kombinerad data i figur 23, bekräftar LSTMs förmåga att korrekt klassificera olika klasser med hög noggrannhet. Detta styrks av en noggrannhet på 90.0% och en specificitet på 94.1%, vilket är signifikant högre än de andra algoritmerna som undersökts. Sådana resultat indikerar att LSTM kan skilja mellan normala och patologiska ögonrörelsemönster med hög tillförlitlighet.

Vidare är förlustvärdena, både i termer av validerings- och träningsförlust, lägst för LSTM när den tränas på kombinerad data. Denna lägre förlust indikerar en bättre generalisering och robusthet hos modellen, vilket är essentiellt för att tillämpa modellen i verkliga scenarier där variation i data är oundviklig. Denna kombination av hög noggrannhet och låga förlustvärden understryker LSTMs effektivitet och pålitlighet.

LSTM överträffar också andra jämförbara modeller, såsom GRU och DNN, i hantering av komplexa datamängder där det är kritiskt att upprätthålla en kontinuerlig analys av tidsseriedata. Sammantaget pekar resultaten på att LSTM inte bara hanterar de utmaningar som finns i den syntetiska och verkliga datan väl, utan även excellerar i att dra nytta av den kombinerade datans bredare spektrum av egenskaper. Denna förmåga gör den särskilt värdefull i medicinska tillämpningar där precision och tillförlitlighet är viktiga komponenter för säkerställa patientens behandlingsutfall och säkerhet.

Jämförelse inspelad och syntetisk data

När de syntetiska videorna integrerades i den befintliga fil- och klassificeringssystemet ledde det till en förbättring av vårt dataset. Användningen av syntetisk data möjliggjorde en betydande ökning av datamängden tillgänglig för träning av ML-modellen, vilket är kritiskt för att uppnå hög noggrannhet och tillförlitlighet i strokekaraktäriseringen. Den precisa kontrollen av ögonen i modellen möjliggjorde även att noggrant simulera och analysera en bred variation av strokeinducerade ögonfel, vilket bidrog till en djupare förståelse och bättre modellprestanda.

I resultatet syns det att modellerna generellt presterat sämre då de tränats på endast syntetisk data. En notering som kan förklara den syntetiska datans relativt dåliga resultat till att identifiera klass 1 är att modellen som tränades med syntetisk data testades med inspelad data. Denna testningsmetodik innebär att videor med nystagnus samt kranialnerv 3, 4 och 6 ej blev testade. Sådana videos, som är indelade i fyra olika kategorier och har en begränsad mängd tillgängliga exempel representerar en viktig del som behöver validering för att säkerställa modellens tillförlitlighet. Hela 55% av klass 1 videor i syntetisk data blev inte testade. Detta ger en fördel till inspelad och kombinerad data, inspelad data varken tränas eller testas på

de typer av videor och kombinerad data både tränas och testas på dem. Även kombinerad data påverkas av att de fyra kategorier innehåller relativt få videor, det blir svårt att garantera att modellen blivit tillräckligt testad på varje kategori. Detta kan resultera i att modellens verkliga förmåga att generalisera över olika typer av data kan underskattas.

Sammanfattningsvis så har modellerna presterat bättre då de tränats på inspelad eller kombinerad data jämfört med att endast ha tränats på endast syntetisk data. Orsaken till detta är antagligen att den syntetiska data har täckt fler ögonfel och har därmed inte kunnat lära upp modellen tillräckligt för att ge korrekta klassificeringar. Samtidigt så visar resultaten att flera modeller har presterat bäst då den tränats på kombinerad data vilket tyder på att träningen gynnas utav den tillförda syntetiska datan. Även förlusterna i vissa modeller, främst LSTM, visar att modellerna presterar bättre med träning på kombinerad data. Det visar på att den syntetiska datan är viktig för modellens träning även om den inte på egen hand kan träna upp modellen till en acceptabel nivå.

5.3.2 Datagenerering och bearbetning

Efter en omfattande granskning av den insamlade datan från CSV-filerna insågs det att en ytterligare transformation var nödvändig för att öka datans tolkbarhet. Transformationen involverade att konvertera X- och Y-koordinaterna för båda pupillerna till differenser mellan varje bilds koordinater, se figur 14. Denna förändring ledde till en markant förbättring i algoritmernas prestation, inklusive ökad noggrannhet, precision, sensitivitet, specificitet och F1-poäng. Förbättringen kan tillskrivas det faktum att modellerna mer effektivt kan identifiera mönster i data när förändringar mellan bildrutor uttrycks i små numeriska värden, i motsats till när värdena enbart representerar de absoluta koordinaterna inom videons pixellager.

Anpassningen av bilder per sekund för den samlade videodatan till en enhetlig hastighet var kritisk för att säkerställa konsistens i analysen av ögonrörelser. GazeTracking-programmet, som användes för att extrahera ögonkoordinater från videon och konvertera dessa till CSV-format, bearbetar videor bild för bild ("frame by frame"). När den inspelade datan, inspelad med 50 bilder per sekund, ställdes mot den syntetiska datan renderad till 30 bilder per sekund, resulterade detta i en ojämn representation av ögonrörelserna i CSV-filerna. Det märktes tidigt att denna diskrepans skulle kunna leda till felaktiga tolkningar av ögonrörelsemönster, då ML-modellen kunde uppfatta identiska ögonrörelser som olika beroende på den varierande mängden av CSV-rader genererad per sekund.

Den här standardiseringen var därför avgörande för att undvika överrepresentation av rörelsedata från det högre antalet bilder per sekund hos den inspelade videodatan, vilket potentiellt kunde skapa bias i hur de olika modellerna tolkade och lärde sig från den inspelade datan jämfört med den syntetiska. Genom att harmonisera videodatainställningarna till en enhetlig hastighet av 30 bilder per sekund, säkerställdes att varje ögonrörelse, oavsett ursprung från inspelad eller syntetisk data, representerades likvärdigt i datamängden. Referens till metodiken som beskriver anpassningsprocessen finns i avsnitt 5.1.2.

Anledningen för att just 65 syntetiska videofiler skapades och inte färre var på grund av att hålla syntetiska datamängden någorlunda jämn med den inspelade datamängden och fler än 65 syntetiska videofiler ökade risken för överanpassning. Det märktes även att det fanns ett begränsat antal sätt att skapa samma rörelse med samma ansikte i de utvalda programvarorna vilket innebar att ytterligare videor av samma ögonfel skulle blivit kopior.

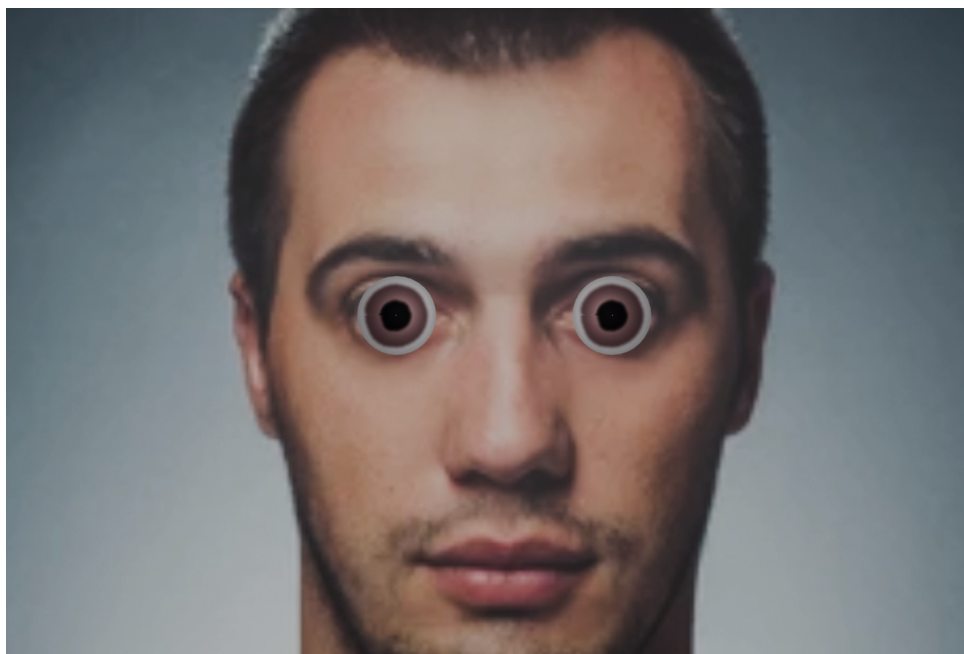
Ur tabell 15 framgår det att datasetet inte är balanserat och med en märkbar brist på videor för klass 1 jämfört med de andra klasserna. Det finns betydligt färre videor av klass 1 i jämförelse med klass 0 och klass 2, vilket kan påverka modellens förmåga att korrekt identifiera klass 1 och därmed påverka den övergripande prestandan hos klassificeringsmodellen. Denna obalans i datamängden kan leda till en snedvriden inlärning

vilket kan vara en anledning till att flera modeller hade svårt att identifiera videor av klass 1.

I början vid testning av algoritmer insågs det snabbt att det uppstod mycket problem med prestandan. Till en början kunde modellerna endast klassificera stroke av klass 1 enligt NIHSS och gjorde det på alla test. Att modellerna trots detta gav till synes bra resultat berodde troligtvis på att träningsdatan bestod av en majoritet stroke klass 1. Mycket av den syntetiska datan hade överdrivna kamerarörelser, vilket gjorde att modellerna kan ha blandat ihop klassificeringarna till bara klass 1. Exempelvis fick den algoritmen med den bästa prestandan en noggrannhet på 81.48%, vilket berodde på att 81.48% av videorna var av klass 1. Problemet löstes genom att spela in fler videor av klass 0 och 2, och ta bort några klass 1 av den syntetiska datan, som hade överdrivna kamerarörelser. Ny inspelning och borttagning av videor ledde till en mer pålitlig modell.

Ett annat problem som diskuterades vid skapandet av syntetisk data var variation på utseende. All syntetisk data använder samma 3D modell med samma ansiktsform samt samma ögon. Storlek av ögon och pupiller samt ögonens relativa position till ansiktet ändrades aldrig. Detta på grund av tid- och pengarrestriktioner, då det hade tagit lång tid att hitta bra passande modeller samt att de allra flesta modeller av sökt kvalité kostar pengar.

Nedan i figur 26 syns det första utkastet av en mall för den syntetiska datan. Denna prototypen fungerade på så sätt att GazeTrackings bildanalys lyckades detektera pupillerna och deras koordinater, men användes dock inte på grund av utseendet. Alltså var den slutgiltiga mallen för syntetisk data som presenterades i metodavsnittet, figur 9 en betydlig uppgraderingen.



Figur 26: Syntetisk data prototyp 1.

5.3.3 Arbetets betydelse

Arbetet representerar ett viktigt steg i forskningen om tidig diagnos av stroke, en ledande orsak till dödlighet och funktionsnedsättning världen över. Innovativa metoder utforskas genom att integrera videoanalys, maskinlärning och ögonspårning för att identifiera tidiga tecken på stroke. Denna forskning är avsedd att fungera som ett proof of concept för att visa potentialen i att digitalisera NIHSS och analysera ögonrörelseavvikelser

digitalt och använda som en tidig indikator på stroke. Genom att utveckla en hybrid datamängd som kombinerar verkliga och syntetiska data, syftar projektet till att förbättra pålitligheten och generaliserbarheten hos de maskininlärningsmodeller som tas fram. Även om dessa framsteg indikerar på att konceptet är möjligt krävs ytterligare studier och verifiering innan dessa metoder kan implementeras i klinisk praxis. Arbetet är viktigt för att främja tidig diagnos och behandling av stroke, vilket kan leda till betydande förbättringar i patientutfall och minska den samhällsmässiga bördan av denna allvarliga sjukdom.

5.3.4 Framtida arbete

Det är tydligt från resultaten att LSTM-modellen har potentialen för att nå en såpass avancerad och noggrann nivå av strokebedömning att den hade kunnat användas i prehospital vård. För framtida arbete hade LSTM-modellen gynnat av vidareutveckling som till en stor del omfattar att samla in verklig data samt att generera ytterligare och mer välgjord syntetisk data. Den syntetiska datan som genererats i detta arbetet har visat sig vara otillräcklig av flera anledningar, främst att antalet videor varit för få. För framtida utveckling hade det även behövts en bättre balans mellan klasserna inom den syntetiska datan samt att datasetet hade behövt utökas. Som det upplystes i avsnitt 5.3.1 är även vidare studie och analys av ögonrörelser vid specifika fall såsom partiell blickpares nödvändigt för att utveckla modellen och stödet den får från den syntetiska datan.

Utöver alla dessa utvecklingsområden hade arbetet även kunnat påbörja processen av att göra den användarvänlig för att förberedas för användning. Ett väl genomtänkt och enkelt användargränssnitt bör utvecklas för att ambulanspersonal i praktiken snabbt och effektivt ska kunna använda systemet på patienter.

För att uppnå en optimal modell som kan framstå som ett kraftfullt hjälpmedel för strokebedömning hade vidare implementering av ytterligare aspekter från NIHSS kunnat gynna systemet enormt. Dessa aspekter kan vara inom områden utanför ögon och ögonrörelser såsom dysartri, facialispares eller arm- och benpares. Digitalisering av bland annat dessa aspekter från NIHSS har redan studerats i tidigare arbete och även föregående kandidatarbete [16]. Genom att röra sig utanför ramen för ögonrörelser kan det redan utvecklade systemet i detta arbete, i samarbete med andra ML-modeller och insamlad data av dessa nya strokeaspekter, kunna föra arbetet till ett mer allmänt och säkert hjälpmedel till strokebedömning.

5.3.5 Påträffade begränsningar

Flera problem uppstod under arbetets gång som behövdes hanteras, speciellt vid skapandet av data då ett problem genom projektet har varit det begränsade datasetet. Dessa begränsningar är under detta avsnitt diskuterade och klargjorda.

Inspelad data

Skapandet av data var tidskrävande eftersom alla videor manuellt behövdes spelas in. Vissa imiteringar behövdes göras om ett flertal gånger för att resultatet skulle bli så bra som möjligt. Detta för att en del av symptomen var svårare än andra att efterlikna, till exempel Doll-eye test, som därför valdes att inte tas med. Andra symptom som nyastagmus och eye-muscle paresis testades också att efterliknas utan lyckade resultat, vilket gjorde att dessa symptom uteslöts. Dessa inkluderades dock i den syntetiska datan. Vissa typer av videor blev inte godkända av stroke-experten P. Redfors, detta tillsammans med lång tid mellan återkoppling gjorde att videor togs i flera omgångar vilket ökade mängden tid som krävdes. På grund av tidsbrist gjordes enbart 99st videor.

En annan svårighet med arbetet var att med ett bestämt mått avgöra hur bra deltagarnas efterliknelse av symptom motsvarar en faktiskt stroke. Denna felmarginal gick inte att uppskatta men med tanke på att datan verifierades av experter antogs datan vara tillräckligt tillförlitlig. Men ifall man hade testat ML-modellerna

med data från riktiga stroke patienter hade resultaten kanske varit annorlunda.

All videospelning gjordes i en nästintill perfekt miljö, utan störningar som rörelse och i bra ljus utan skuggor. Denna optimala miljö kan skilja sig åt vad som kan förekomma i en ambulans då miljön eventuellt inte alltid är så bra, speciellt om detta ska göras ute i trafik. Potentiella störningar i verkliga scenarion gör att arbetets träningsdata inte återspeglar verkligheten vidare bra vilket fås ta till hänsyn vid användning.

Vid skapande av träningsdata är det viktigt att tänka på att använda mycket olika mångfald för att få en bred träningsdata. Eftersom detta arbete endast analyserar ögonen var det viktigaste att ha mångfald på den kategorin. En bra mångfald hade varit att testa på flera deltagare med varierande storlek på ögon, pupiller och ögonens relativa position i ansiktet.

Syntetisk data

Syntetisk data kan tillhandahålla större och mer diversifierade datamängder vilket är särskilt värdefullt där tillgång till verkliga patientdata är begränsad eller otillgänglig. Dessutom möjliggör syntetisk data forskare att inkludera och modellera sällsynta eller specifika tillstånd som är svåra att efterlikna i verkliga situationer. Detta är av stor betydelse i strokeforskning där vissa ögonsymptom endast uppkommer under särskilda neurologiska förhållanden. Användningen av syntetisk data kan också bidra till att minska tiden det tar att initiera forskning och utveckling. I detta fall möjliggjorde det att ögonfel som inte kunde imiteras kunde skapas digitalt och därmed fortfarande ingå i datan. Metoden erbjuder ett kostnadseffektivt alternativ till dyra och tidskrävande datainsamlingar, och kan potentiellt förkorta utvecklingscykler vilket gör det till ett viktigt verktyg. För att ytterligare öka tillgängligheten och relevansen av syntetiska data, krävs kontinuerlig utveckling av genereringsverktygen för att de ska kunna skapa mer exakta och kontrollerbara datamängder. Ett framsteg i denna riktning skulle kunna revolutionera förutsättningarna för forskning inom många områden av medicinen, särskilt inom områden där etiska eller praktiska hinder begränsar insamlingen av verklig data såsom detta projekt.

Vid skapandet av syntetisk data uppstod många problem. Det alternativet som såg mest lovande ut var Microsoft FaceSynthetics (GitHub - microsoft/FaceSynthetics) [55], ett öppen källkod-verktyg tillgängligt på GitHub, som har gjort miljontals bilder på syntetiska 3D ansikten. Dock visade sig detta verktyg vara oförmöget att direkt generera videosekvenser med specifikt kontrollerade ögonrörelser som är essentiella för arbetets syfte.

Andra lovande möjligheter till att skapa syntetisk data hittades som NVlabs StyleGAN [57]. Problemet var att träna generatoren. Skaparnas av programmet har en beskrivning, som visas i tabell 26, om hur lång tid det skulle ta att processera träningsdata med en Tesla V100 GPU [57]. En sådan GPU kostar enligt priser givna av Microsoft 657 dollar/månad att hyra [77]. Investeringen i både tid och pengar var över projektets ramar.

GPUs	Högupplösning	Medelupplösning	Lågupplösning
1	41 days 4 hours	24 days 21 hours	14 days 22 hours
2	21 days 22 hours	13 days 7 hours	9 days 5 hours
4	11 days 8 hours	7 days 0 hours	4 days 21 hours
8	6 days 14 hours	4 days 10 hours	3 days 8 hours

Tabell 26: Tid för generera syntetisk data med Tesla V100 GPUs.

6 Slutsats

Projektet har visat sig framgångsrikt utvecklat en ML-modell för tidig karaktärisering av stroke genom videoanalys av ögonrörelser. Genom att använda både verklig och syntetisk videodata, som konverterats till sekvensiella koordinater och blicktillstånd i form av CSV-filer, har modellen tränats och validerats genom ett flertal olika metoder. Noggrann analys och optimering bidrog till att vår bäst presterande algoritm nådde en bra förståelse av hur ögonrörelsemönster korrelerar med olika strokegrader enligt NIHSS. Specifikt visade LSTM-algoritmen en framträdande prestanda med en noggrannhet på 90.0%, vilket översteg de andra testade algoritmer som till exempel CNN och SVM, där de respektive noggrannheterna var 48.48% och 68.29%. Denna höga noggrannhet hos LSTM kan tyda på dess förmåga att hantera sekvensdata över tid, vilket är kritiskt för att korrekt analysera de temporala mönstren i ögonrörelsedata.

Analysen av resultatet relaterar till projektets syfte, vilket var att utveckla metoder för tidig detektion och karaktärisering av stroke i prehospitala miljöer. Genom att generera och utnyttja syntetisk data tillsammans med inspelad verklig data kunde en väl optimerade ML-modell tränas. Resultaten indikerar att LSTM-modellen inte bara kan stödja nuvarande kliniska bedömningsmetoder utan också potentiellt förbättra snabbheten och precisionen i strokebedömningen, vilket kan minska tiden till behandling och därmed förbättra patientutfall.

Framtida forskning bör inriktas på att ytterligare förbättra LSTM-modellens tillförlitlighet och stabilitet genom kontinuerlig validering mot varierade datatyper. Denna testning på algoritmen bidrog till en djupare analys av modellens prestanda över olika stroke tillstånd. Vidare är det också av vikt att utforska modellens integrering i kliniska arbetsflöden, där den kan fungera som ett beslutsstöd för tidig diagnos och bedömning av stroke i prehospitala miljöer. Denna integrering kan innebära utvecklandet av ett program med ett användargränssnitt för ambulanspersonal att enkelt och effektivt använda i akuta diagnostiseringar.

Bevisligen, med resultaten som har åstadkommit, är det enligt projektets syfte möjligt att skapa relevant syntetisk data på ögonrörelser hos strokepatienter som kan användas för att både träna och testa ett ML-system. Det som har visats i detta arbetet är att syntetisk data fungerar bättre som ett tillägg till inspelad data för att träna en modell. Den genererade syntetiska datan spelade en stor roll i detta arbetet och bidrog till en djupare förståelse om hur modellerna lär sig från data i allmänhet och främst hur de presterade i olika strokefall.

Projektets resultat har potential att påverka framtida utveckling inom medicinsk diagnostik, särskilt för tillstånd där snabba och noggranna diagnoser är avgörande. Den slutgiltiga modellen i detta arbetet kan alltid förbättras med ett större och mer noggrant dataset där olika datatyper och klassificeringar är mer jämnt balanserade. Ytterligare utveckling och implementering kan leda till att denna teknologi erbjuder betydande förbättringar i vårdens effektivitet och patienters prognoser.

Referenser

- [1] P. Tuominen, *Stroke*, <https://www.1177.se/sjukdomar--besvar/hjarna-och-nerver/stroke-och-blodkarl-i-hjarnan/stroke/>, Tillgänglig: 1177 (hämtad 24, jan. 2024).
- [2] V. L. Feigin, M. Brainin, B. Norrving m. fl., "World Stroke Organization (WSO): Global Stroke Fact Sheet 2022," eng, *International Journal of Stroke: Official Journal of the International Stroke Society*, årg. 17, nr 1, s. 18–29, jan. 2022, (hämtad 6. maj, 2024), ISSN: 1747-4949. DOI: 10.1177/17474930211065917.
- [3] X. Hu och K. S. Sunnerhagen, *Stroke, rehabilitering*, <https://www.internetmedicin.se/rehabiliteringsmedicin/stroke-rehabilitering>, Tillgänglig: Internetmedicin (hämtad 5, feb. 2024).
- [4] P. B. Gorelick, "Primary and Comprehensive Stroke Centers: History, Value and Certification Criteria," *Journal of Stroke*, årg. 15, nr 2, s. 78–89, 2023, (hämtad 4, maj. 2024). URL: DOI:%2010.5853/jos.2013.15.2.78.
- [5] *Stroke*, <https://www.nhs.uk/conditions/stroke/>, Tillgänglig: NHS choices (hämtad 25, jan. 2024).
- [6] Socialstyrelsen, *Nationella riktlinjer för strokesjukvård*, (hämtad 6. maj, 2024), 2024. URL: <https://www.socialstyrelsen.se/kunskapsstod-och-regler/regler-och-riktlinjer/nationella-riktlinjer/riktlinjer-och-utvarderingar/stroke/%7D>.
- [7] *Stroke*, <https://www.hjart-lungfonden.se/sjukdomar/hjartsjukdomar/stroke/>, Tillgänglig: Hjärt- Lungfonden (hämtad 3, mar. 2024).
- [8] J. K. Nicholls, J. Ince, J. S. Minhas och E. M. L. Chung, "Emerging Detection Techniques for Large Vessel Occlusion Stroke: A Scoping Review," *Frontiers in Neurology*, årg. 12, s. 780324, jan. 2022, (hämtad 6. maj, 2024), ISSN: 1664-2295. DOI: 10.3389/fneur.2021.780324. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8796731/>.
- [9] H. J. Audebert, J. L. Saver, S. Starkman, K. R. Lees och M. Endres, "Prehospital Stroke Care," *Neurology*, årg. 81, nr 5, s. 501–508, 2013, (hämtad 4, maj. 2024). URL: DOI:%2010.1212/wnl.0b013e31829e0fdd.
- [10] L. Rosengren, *REGIONAL MEDICINSK RIKTLINJE*, <https://mellanarkiv-offentlig.vgregion.se/alfresco/s/archive/stream/public/v1/source/available/sofia/hs9766-305841775-120/surrogate/Trombolys%20och%20trombektomi%20vid%20ischemisk%20stroke.pdf>, Tillgänglig: Västra Götalandsregionen (hämtad 25, jan. 2024).
- [11] K. M. Keigher, "Large Vessel Occlusion in the Acute Stroke Patient: Identification, Treatment, and Management," eng, *Critical Care Nursing Clinics of North America*, årg. 32, nr 1, s. 21–36, mars 2020, (hämtad 6. maj, 2024), ISSN: 1558-3481. DOI: 10.1016/j.cnc.2019.11.007.
- [12] M. Delazer, M. Sojer, P. Ellmerer, C. Boehme och T. Benke, "Eye-tracking provides a sensitive measure of exploration deficits after acute right MCA stroke," *Frontiers in Neurology*, årg. 9, 2018, (hämtad 4, maj. 2024). URL: DOI:%2010.3389/fneur.2018.00359.
- [13] J. Bogousslavsky och O. Meienberg, "Eye-movement disorders in brain-stem and cerebellar stroke," *Archives of Neurology*, årg. 44, nr 2, s. 141–148, 1987, (hämtad 4, maj. 2024). URL: DOI:%2010.1001/archneur.1987.00520140013011.
- [14] *Visual Neglect - Eye Wiki*, https://eyewiki.aao.org/Visual_Neglect, (hämtad 4. maj, 2024), 2023.
- [15] H.-O. Karnath, "Spatial attention systems in spatial neglect," *Neuropsychologia*, årg. 75, s. 61–73, 2015, (hämtad 4, maj. 2024). URL: DOI:%2010.1016/j.neuropsychologia.2015.05.019.
- [16] A. Borg, V. Johansson, N. Larsson, M. Lorentzon, E. Thoreström och O. Tryggvasson, *Early Characterization of Stroke Using Video Analysis and Machine Learning*, Chalmers Tekniska Högskola, Göteborg, (hämtad 4, feb. 2024), 2023.
- [17] P. Lyden, *NIH Stroke Scale/Score (NIHSS)*, <https://www.mdcalc.com/calc/715/nih-stroke-scale-score-nihss>, Tillgänglig: Mdcalc (hämtad 26, jan. 2024), 2024.
- [18] R. Kothari, A. Pancioli, T. Liu, T. Brott och J. Broderick, "Cincinnati Prehospital Stroke Scale: Reproducibility and Validity," en, *Annals of Emergency Medicine*, årg. 33, nr 4, s. 373–378, april 1999, (hämtad 6. maj, 2024), ISSN: 01960644. DOI: 10.1016/S0196-0644(99)70299-4. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0196064499702994> (hämtad 2024-05-06).
- [19] H. Budinčević, A. Meštrović och V. Demarin, "Stroke Scales as Assessment Tools in Emergency Settings: A Narrative Review," en, *Medicina*, årg. 58, nr 11, s. 1541, okt. 2022, (hämtad 6. maj, 2024), ISSN:

- 1648-9144. DOI: 10.3390/medicina58111541. URL: <https://www.mdpi.com/1648-9144/58/11/1541> (hämtad 2024-05-06).
- [20] NIH strokeskala (NIHSS) / Institutionen för neurovetenskap och fysiologi, Göteborgs universitet, sv, (hämtad 7. maj, 2024), okt. 2022. URL: <https://www.gu.se/neurovetenskap-fysiologi/nih-strokeskala-nihss>.
- [21] T. Lee, E.-T. Jeon, J.-M. Jung och M. Lee, “Deep-Learning-Based Stroke Screening Using Skeleton Data from Neurological Examination Videos,” eng, *Journal of Personalized Medicine*, årg. 12, nr 10, s. 1691, okt. 2022, (hämtad 6. maj, 2024), ISSN: 2075-4426. DOI: 10.3390/jpm12101691.
- [22] C. M. Aldridge, M. M. McDonald, M. Wruble m. fl., “Human vs. Machine Learning Based Detection of Facial Weakness Using Video Analysis,” English, *Frontiers in Neurology*, årg. 13, juli 2022, (hämtad 6. maj, 2024), ISSN: 1664-2295. DOI: 10.3389/fneur.2022.878282. URL: <https://www.frontiersin.org/journals/neurology/articles/10.3389/fneur.2022.878282/full>.
- [23] Y. Zhuang, M. M. McDonald, C. M. Aldridge m. fl., “Video-Based Facial Weakness Analysis,” eng, *IEEE transactions on bio-medical engineering*, årg. 68, nr 9, s. 2698–2705, sept. 2021, (hämtad 6. maj, 2024), ISSN: 1558-2531. DOI: 10.1109/TBME.2021.3049739.
- [24] NIH strokeskala (NIHSS) / Institutionen för neurovetenskap och fysiologi, Göteborgs universitet, sv, (hämtad 7. maj, 2024), okt. 2022. URL: <https://www.gu.se/neurovetenskap-fysiologi/nih-strokeskala-nihss>.
- [25] R. D. Sanders, *Cranial nerves III, IV, and VI: Oculomotor function*, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2801485/>, Tillgänglig: Psychiatry (Edgmont (Pa.: Township)) (hämtad 5. mar, 2024).
- [26] E. Dishion och P. Tadi, *Doll’s eyes - statpearls - NCBI bookshelf*, <https://www.ncbi.nlm.nih.gov/books/NBK551716/>, Tillgänglig: National Library of Medicine (hämtad 24, jan. 2024).
- [27] D. P. Rastall och K. Green, “Deep learning in acute vertigo diagnosis,” *Journal of the neurological sciences*, årg. 443, s. 120454, 2022, (hämtad 4, maj. 2024). URL: DOI: %2010.1016/j.jns.2022.120454.
- [28] *What Is Nystagmus?* en, (hämtad 6. maj, 2024). URL: <https://my.clevelandclinic.org/health/diseases/22064-nystagmus>.
- [29] G. Mantokoudis, T. Wyss, E. Zamaro m. fl., “Stroke Prediction Based on the Spontaneous Nystagmus Suppression Test in Dizzy Patients: A Diagnostic Accuracy Study,” en, *Neurology*, årg. 97, nr 1, juli 2021, (hämtad 6. maj, 2024), ISSN: 0028-3878, 1526-632X. DOI: 10.1212/WNL.000000000012176. URL: <https://www.neurology.org/doi/10.1212/WNL.000000000012176>.
- [30] B. A. Sjöqvist, *Hur kan vi förkorta tiden till trombeaktomi i hela VGR genom att använda telemedicin med videokonferens*, <https://www.gu.se/sites/default/files/2023-12/ViPHS%20-%20Videost%C3%B6d%20i%20den%20PreHospitala%20Strokekedjan%20%20Sammanfattning%20och%20vidare%20utveckling%20PA%20Sj%C3%B6qvist.pdf>, Tillgänglig: Göteborgs Universitet (hämtad 4, maj. 2024), 2023.
- [31] Q. Lu m. fl., “Machine learning models for stroke detection by observing the eye-movement features under five-color visual stimuli in traditional Chinese medicine,” *Journal of Traditional Chinese Medical Sciences*, årg. 10, nr 3, s. 321–330, 2023, (hämtad 4, maj. 2024). URL: DOI: %2010.1016/j.jtcms.2023.06.003.
- [32] M. Giuffrè och D. L. Shung, “Harnessing the power of synthetic data in healthcare: innovation, application, and privacy,” en, *npj Digital Medicine*, årg. 6, nr 1, s. 1–8, okt. 2023, (hämtad 7. maj, 2024), ISSN: 2398-6352. DOI: 10.1038/s41746-023-00927-3. URL: <https://www.nature.com/articles/s41746-023-00927-3>.
- [33] D. Rankin, M. Black, R. Bond, J. Wallace, M. Mulvenna och G. Epelde, “Reliability of Supervised Machine Learning Using Synthetic Data in Health Care: Model to Preserve Privacy for Data Sharing,” EN, *JMIR Medical Informatics*, årg. 8, nr 7, e18910, juli 2020, (hämtad 7. maj, 2024). DOI: 10.2196/18910. URL: <https://medinform.jmir.org/2020/7/e18910>.
- [34] C. S. och J. H., *ASAP (Acute Support Assessment and Prioritizing) vid akut strokesjukvård*, (hämtad 4, maj. 2024), 2023. URL: <https://www.gu.se/sites/default/files/2023-12/ASAP%20%28Acute%20Support%20Assessment%20and%20Prioritizing%29%20vid%20akut%20strokesjukv%C3%A5rd%20Candefjord%20%2B%20Halo.pdf>.

- [35] M. Daidone, S. Ferrantelli och A. Tuttolomondo, *Machine learning applications in stroke medicine: advancements, challenges, and future prospectives*, (hämtad 4. maj. 2024), 2024. URL: DOI:%2010.4103/1673-5374.382228.
- [36] P. B. C. V. P. Campbell och P. P. M. Khatri, "Stroke," *The Lancet*, årg. 396, nr 10224, s. 129–142, 2020, (hämtad 4. maj. 2024). URL: DOI:%2010.1016/S0140-6736(20)31179-X.
- [37] *Dataskydd*, <https://www.imy.se/verksamhet/dataskydd/>, Tillgänglig: Integritetsskyddsmyndigheten: IMY (hämtad 24. jan. 2024), 2024.
- [38] Elsevier B.V., *Scopus*, <https://www.elsevier.com/products/scopus>, (hämtad 5. maj, 2024), 2024.
- [39] Wikipedia, *Google Search*, https://en.wikipedia.org/wiki/Google_Search, (hämtad 6. maj, 2024), 2024.
- [40] P. Kuang, E. Söderberg, D. C. Niehorster och M. Höst, "Applying Machine Learning to Gaze Data in Software Development: a Mapping Study," i *Proceedings of the 2023 Symposium on Eye Tracking Research and Applications*, ser. ETRA '23, (hämtad 6. maj, 2024), Tubingen, Germany: Association for Computing Machinery, 2023, ISBN: 9798400701504. DOI: 10.1145/3588015.3589190. URL: <https://doi.org/10.1145/3588015.3589190>.
- [41] N. Srivastava, U. Agrawal, S. K. Roy och U. S. Tiwary, "Human identification using Linear Multiclass SVM and Eye Movement biometrics," i *Proceedings of the 2015 Eighth International Conference on Contemporary Computing (IC3)*, (hämtad 4. maj, 2024), 2015, s. 365–369. DOI: 10.1109/IC3.2015.7346708. URL: <https://doi.org/10.1109/IC3.2015.7346708>.
- [42] A. Zyout, O. Alquraan, M. Alsalatie m.fl., "EOG Based Eye Movements and Blinks Classification Using Irisgram and CNN-SVM Classifier," i *Proceedings of the IEEE/CVF*, (hämtad 4. maj, 2024), 2023, s. 374–377. DOI: 10.1109/IICETA57613.2023.10351437. URL: <https://doi.org/10.1109/IICETA57613.2023.10351437>.
- [43] L. Qi och A. P. D. N. Alias, "Comparison of ANN and SVM for classification of eye movements in EOG signals," *Journal of Physics: Conference Series*, årg. 971, nr 1, s. 012012, 2018, (hämtad 4. maj, 2024). DOI: 10.1088/1742-6596/971/1/012012. URL: <https://doi.org/10.1088/1742-6596/971/1/012012>.
- [44] Z. Fu, B. Wang, F. Chen, X. Wu och J. Chen, *Eye-gaze Estimation with HEOG and Neck EMG using Deep Neural Networks*, (hämtad 4. maj, 2024), 2021. DOI: 10.13140/RG.2.2.35059.45609. URL: <https://doi.org/10.13140/RG.2.2.35059.45609>.
- [45] A. Bublea och C. D. Căleanu, "AutoML and Neural Architecture Search for Gaze Estimation," i *Proceedings of the 2022 IEEE 16th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, (hämtad 4. maj, 2024), 2022, s. 143–148. DOI: 10.1109/SACI55618.2022.9919471. URL: <https://doi.org/10.1109/SACI55618.2022.9919471>.
- [46] S. Krishnan, A. Joseph och S. Tejwani, "Gaze Fusion-Deep Neural Network Model for Glaucoma Detection," i *Advances in Visual Computing*, (hämtad 4. maj, 2024), 2021, s. 42–53, ISBN: 978-981-16-0418-8. DOI: 10.1007/978-981-16-0419-5_4. URL: https://doi.org/10.1007/978-981-16-0419-5_4.
- [47] A. Helwan och M. Ma'aitah, "Gaze Prediction Based on Convolutional Neural Network," i *Proceedings of the 2021 International Conference on Computer Vision*, (hämtad 4. maj, 2024), 2021, ISBN: 978-3-030-85989-3. DOI: 10.1007/978-3-030-85990-9_18. URL: https://doi.org/10.1007/978-3-030-85990-9_18.
- [48] P. J. Das, A. K. Talukdar och K. K. Sarma, "A Framework For Human Behaviour Detection Using Combined Analysis of Facial Expression and Eye Gaze," s. 154–160, 2019, (hämtad 4. maj, 2024). DOI: 10.1109/IESPC.2019.8902367. URL: <https://doi.org/10.1109/IESPC.2019.8902367>.
- [49] Z. A. Ahmed, E. Albalawi, T. Aldhyani, M. Jadhav, P. Janrao och M. Obeidat, "Applying Eye Tracking with Deep Learning Techniques for Early-Stage Detection of Autism Spectrum Disorders," *Data*, årg. 8, nr 11, s. 168, 2023, (hämtad 4. maj, 2024). DOI: 10.3390/data8110168. URL: <https://doi.org/10.3390/data8110168>.
- [50] Y.-S. Jeong och N.-W. Cho, "Evaluation of e-learners' concentration using recurrent neural networks," *The Journal of Supercomputing*, årg. 79, s. 1–18, 2022, (hämtad 4. maj, 2024). DOI: 10.1007/s11227-022-04804-w. URL: <https://doi.org/10.1007/s11227-022-04804-w>.

- [51] Y. Tu, Y. Liu och X. Li, “Deepfake Video Detection by Using Convolutional Gated Recurrent Unit,” i *Proceedings of the 2021 International Conference*, (hämtad 4. maj, 2024), 2021, s. 356–360. DOI: 10.1145/3457682.3457736. URL: <https://doi.org/10.1145/3457682.3457736>.
- [52] M. A. E., S. Wibirama och I. Ardiyanto, “Accuracy Improvement of Object Selection in Gaze Gesture Application using Deep Learning,” i *Proceedings of the 2020 12th International Conference on Information Technology and Electrical Engineering (ICITEE)*, (hämtad 4. maj, 2024), 2020, s. 307–311. DOI: 10.1109/ICITEE49829.2020.9271771. URL: <https://doi.org/10.1109/ICITEE49829.2020.9271771>.
- [53] *What is a neural network?* en-US, (hämtad 7. maj, 2024), jan. 2019. URL: <https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/>.
- [54] E. Wood, T. Baltrušaitis, C. Hewitt m. fl., *3D face reconstruction with dense landmarks*, (hämtad 6. maj, 2024), 2022. arXiv: 2204.02776 [cs.CV]. URL: <https://arxiv.org/abs/2204.02776>.
- [55] E. Wood, T. Baltrušaitis, C. Hewitt, S. Dziadzio, T. J. Cashman och J. Shotton, “Fake it till you make it: face analysis in the wild using synthetic data alone,” i *Proceedings of the IEEE/CVF international conference on computer vision*, (hämtad 4. maj, 2024), 2021, s. 3681–3691.
- [56] L. Łwirski och N. Dodgson, “Rendering synthetic ground truth images for eye tracker evaluation,” i *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA ’14, (hämtad 6. maj, 2024), Safety Harbor, Florida: Association for Computing Machinery, 2014, s. 219–222, ISBN: 9781450327510. DOI: 10.1145/2578153.2578188. URL: <https://doi.org/10.1145/2578153.2578188>.
- [57] *StyleGAN - Official TensorFlow Implementation*, en, (hämtad 16. apr, 2024). URL: <https://github.com/NVlabs/stylegan>.
- [58] B. Foundation, *blender.org - Home of the Blender project - Free and Open 3D Creation Software*, en, (hämtad 8. maj, 2024). URL: <https://www.blender.org/>.
- [59] “Adobe After Effects,” en, (hämtad 3. maj, 2024), 2024. URL: <https://www.adobe.com/se/products/aftereffects.html>.
- [60] “Turbosquid,” en, (hämtad 3. maj, 2024), 2024. URL: <https://www.turbosquid.com/>.
- [61] “Adobe Premiere Pro,” en, (hämtad 3. maj, 2024), 2024. URL: <https://www.adobe.com/se/products/premiere.html>.
- [62] A. Lamé, *GazeTracking*, en, (hämtad 4. maj, 2024). URL: <https://github.com/antoinelame/GazeTracking?tab=MIT-1-ov-file>.
- [63] K. Team, *Keras documentation: Adam*, en, (hämtad 22. apr, 2024). URL: <https://keras.io/api/optimizers/adam/>.
- [64] “Cognitive Big Data Intelligence with a Metaheuristic Approach,” en, (hämtad 5. maj, 2024), DOI: 10.1016/C2020-0-02004-9, 2022, 3.5.1 Learning rate. URL: <https://www.sciencedirect.com/topics/computer-science/learning-rate>.
- [65] *Regularization for Simplicity: L Regularization | Machine Learning*, en, (hämtad 07. maj, 2024). URL: <https://developers.google.com/machine-learning/crash-course/regularization-for-simplicity/12-regularization>.
- [66] K. Team, *Keras documentation: Dropout layer*, en, (hämtad 07. maj, 2024). URL: https://keras.io/api/layers/regularization_layers/dropout/.
- [67] *What is a Neural Network Flatten Layer?* en-US, (hämtad 7. maj, 2024), febr. 2024. URL: <https://www.geeksforgeeks.org/what-is-a-neural-network-flatten-layer/>.
- [68] A. Koutsoukas, K. J. Monaghan, X. Li och J. Huan, “Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data,” *Journal of Cheminformatics*, årg. 9, s. 42, juni 2017, (hämtad 7. maj, 2024), ISSN: 1758-2946. DOI: 10.1186/s13321-017-0226-y. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5489441/>.
- [69] *Keras: The high-level API for TensorFlow*, en, (hämtad 4. maj, 2024), 2023. URL: <https://www.tensorflow.org/guide/keras>.
- [70] S. Kostadinov, *Understanding GRU Networks*, (hämtad 8. maj, 2024), 2017. URL: <https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be>.
- [71] A. F. M. Agarap, “A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data,” i *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, ser. ICMLC ’18, (hämtad

8. maj, 2024), New York, NY, USA: Association for Computing Machinery, febr. 2018, s. 26–30, ISBN: 9781450363532. DOI: 10.1145/3195106.3195117. URL: <https://doi.org/10.1145/3195106.3195117>.
- [72] R. DiPietro och G. D. Hager, “Chapter 21 - Deep learning: RNNs and LSTM,” i *Handbook of Medical Image Computing and Computer Assisted Intervention*, ser. The Elsevier and MICCAI Society Book Series, S. K. Zhou, D. Rueckert och G. Fichtinger, utg., (hämtad 8. maj, 2024), Academic Press, jan. 2020, s. 503–519, ISBN: 9780128161760. DOI: 10.1016/B978-0-12-816176-0.00026-0. URL: <https://www.sciencedirect.com/science/article/pii/B9780128161760000260>.
- [73] Y. Yamasari, A. Qoiriah, N. Rochmawati, I. M. Suartana och O. Putra, “Exploring the Kernel on SVM to Enhance the Classification Performance of Students’ Academic Performance,” i *Proceedings of the 2022 International Conference on Vocational Education and Electrical Engineering (ICVEE)*, (hämtad 6. maj, 2024), IEEE, sept. 2022, s. 42–46. DOI: 10.1109/ICVEE57061.2022.9930405. URL: <https://doi.org/10.1109/ICVEE57061.2022.9930405>.
- [74] M. Grandini, E. Bagli och G. Visani, *Metrics for Multi-Class Classification: an Overview*, en, (hämtad 7. maj, 2024), 2020. DOI: <https://doi.org/10.48550/arXiv.2008.05756>. arXiv: 2008.05756 [stat.ML].
- [75] V. M, *What is the difference between Training Loss Validation Loss and Evaluation Loss*, en, (hämtad 7. maj, 2024), sept. 2023. URL: <https://medium.com/@penpencil.blr/what-is-the-difference-between-training-loss-validation-loss-and-evaluation-loss-c169ddeccd59>.
- [76] *Descending into ML: Training and Loss | Machine Learning*, en, (hämtad 7. maj, 2024). URL: <https://developers.google.com/machine-learning/crash-course/descending-into-ml/training-and-loss>.
- [77] *Virtual Machine series | Microsoft Azure*, en, (hämtad 16. apr, 2024). URL: <https://azure.microsoft.com/en-us/pricing/details/virtual-machines/series/>.