

Hibridni CNN-LSTM sistem globokega učenja za avtomatsko razvrščanje stanj testa devetih zatičev iz slike

Helena Domajnko, Žiga Špiclin, Žiga Bizjak

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, 1000 Ljubljana, Slovenija

E-pošta: hd2150@student.uni-lj.si

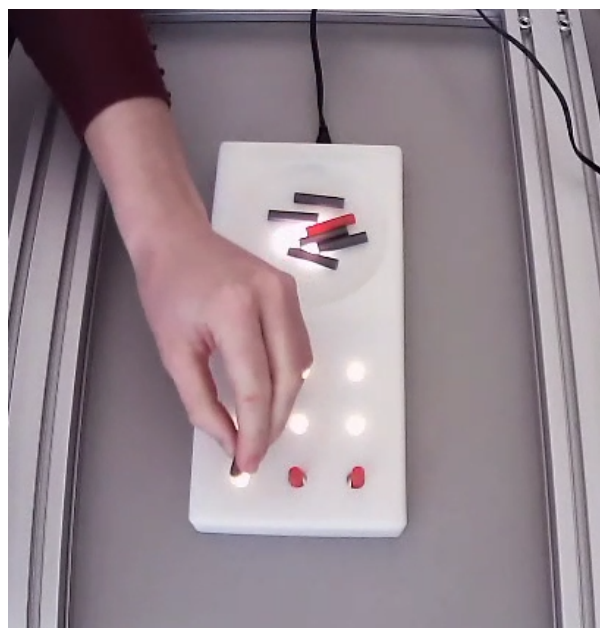
Hybrid CNN-LSTM Deep Learning System for Automatic Detection of Nine-Hole Peg Test States from Image

Abstract. The Nine-Hole Peg Test (9HPT) manually assesses finger dexterity, limiting its widespread, unsupervised use. To enable autonomous administration, this paper introduces a hybrid model dubbed Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM). The model classifies 9HPT video frames into states like "pin grabbing", "pin carrying", "pin releasing", and "empty hand" and also detects common errors such as grabbing multiple pins. Our architecture utilizes a pre-trained ResNet18 CNN for feature extraction, combined with a bidirectional LSTM for temporal analysis. On test data, the model achieved 63% accuracy. While not yet sufficient for fully autonomous clinical deployment, this result demonstrates potential for automated, objective, and scalable fine motor skill assessment and supervision of test subject adherence to test protocol execution.

1 Uvod

Test devetih zatičev (9HPT) je pogosto uporabljen in priznan merilni test za preprosto in hitro oceno fine motorike gornjih okončin, še posebej prstov in dlani [1]. Uporablja se pri bolnikih po možganski kapi, s Parkinsonovo boleznijo in drugimi nevrološkimi stanji, do preverjanja motorike pri otrocih in splošno v rehabilitaciji. Test je uveljavljen zaradi svoje enostavnosti, hitrosti izvedbe in sposobnosti, da zagotovi kvantitativno merilo spretnosti gornjih okončin.

Kljub široki uporabi in klinični relevantnosti pa se 9HPT trenutno izvaja ročno, z uporabo štoparice za merjenje časa in opazovanja morebitnih napak. Ta pristop prinaša vsaj dva izziva: (1) ročno merjenje je podvrženo variabilnosti ocenjevalcev oz. terapevtov, kar je posledica subjektivne ocene izvedbe in lahko vpliva na zanesljivost ter ponovljivost rezultatov; (2) ročna ocena je časovno in kadrovsko potratna, kar omejuje pogostost testiranja in obseg spremljanja napredka pacientov s tem testom. Trenutno na trgu ne obstaja široko dostopen in zanesljiv digitalni sistem za avtomatizirano izvedbo in oceno 9HPT, ki bi lahko nadomestil ročno validacijo. V [2] je predstavljena avtomatizacija testa na podlagi računalniškega vida z uporabo rokavic z barvnimi markerji. Dela [3, 4, 5] predstavljajo digitalne teste devetih zatičev (dNHPT), ki



Slika 1: Izvajanje testa devetih zatičev.

so dosegli dobre ocene zanesljivosti in podobno klinično uporabnost kot 9HPT. dNHPT tako lahko prispeva k objektivizaciji merjenja spretnosti rok brez izgube kompaktnosti in preprostosti, ki sta najpomembnejši lastnosti NHPT. Skupna slabost omenjenih del je bodisi sprememba osnovnega protokola izvedbe 9HPT bodisi nezmožnost avtonomne izvedbe.

Avtomatizacija validacije testa 9HPT bi omogočila avtonomno izvedbo in njegovo širšo uporabo v klinični praksi in raziskavah, naprimer v bolnišnicah ali rehabilitacijskih centrih. To bi povečalo dostopnost in pogostost meritev brez potrebe po stalni prisotnosti terapevta. Dodatna možnost je nenadzorovana uporaba 9HPT v domačem okolju. Omenjena širša uporaba bi pripeljala do obsežnejših in objektivnejših podatkov o dinamiki spremembe stanja pacientov, omogočila zgodnejše zaznavanje sprememb in morebitno prilagoditev terapije. V luči teh potreb pričujoče delo predstavlja hibridni CNN-LSTM model globokega učenja, namenjen avtomatskemu razvrščanju video posnetkov testa v ključna vmesna stanja in hkratio zaznavanje napak pri izvedbi.

2 Metodologija

2.1 Test devetih zatičev

Klasični test devetih zatičev se izvaja z uporabo standardizirane plošče z devetimi luknjicami in devetimi zatiči. Preiskovanec mora z uporabo ene roke (dominantne ali nedominantne, test se izvede za obe) čim hitreje prijeti posamezen zatič iz posodice in ga vstaviti v katerokoli od prostih luknjic na plošči. Ta postopek se ponovi za vseh devet zatičev, dokler niso vsi vstavljeni v luknjice. Ko so vsi zatiči v luknjicah, jih mora preiskovanec enega za drugim, z isto roko, prenesti nazaj v posodico. Protokol izvedbe in merske lastnosti testa so podrobneje predstavljene v preglednem članku [6].

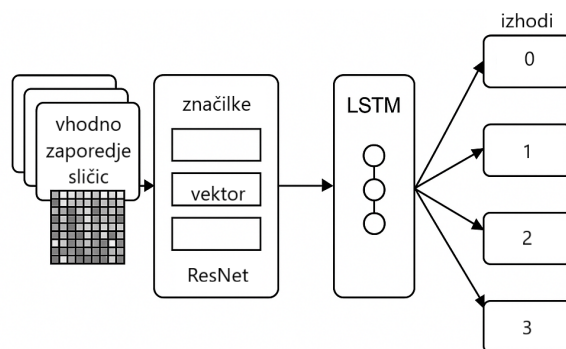
Ključni del klasičnega protokola je ročno merjenje časa in opazovanje izvedbe. Čas se začne beležiti s stoparico v trenutku, ko preiskovanec prične prijemati prvi zatič, in se zaključi, ko je zadnji zatič odložen nazaj v posodico. Med celotnim trajanjem testa mora biti prisoten izkušen ocenjevalec oz. terapevt, ki poleg beleženja časa pozorno opazuje izvedbo, beleži morebitne napake v izvedbi, kot so padec zatiča, prijem dveh zatičev hkrati ali uporaba druge roke, in daje navodila in spodbuja preiskovanca. Natančno beleženje časa in napak je ključno za objektivno oceno spretnosti in spremljanje napredka. Primer izvajanja testa je prikazan na Sliki 1.

2.2 Podatki

Ker tovrstni podatki (posnetki izvedbe 9HPT) niso javno dostopni, smo pričeli z zajemom in pripravo lastnega nabora podatkov. Zajeli smo 50 posnetkov s kamerama modela CMSXJ22A in petimi zdravimi raziskovanci. Pri tem je bila hitrost slihc (FR; ang. *Frame Rate*) 20 slihc na sekundo, ločljivost snemanja 640x480 pikslov in namestitvev kamer na levi in desni strani plošče na višini približno 30 cm. Pogled desne kamere na ploščo je viden na Sliki 1. Posnetki so bili zajeti v različnih svetlobnih pogojih in iz različnih zornih kotov, da bi zagotovili raznolikost posnetkov. Skupno trajanje uporabljenih posnetkov je bilo okoli 40 min.

Za anotacijo videoposnetkov smo uporabili orodje Label Studio [7]. Anotacija je obsegala prepoznavanje in označevanje dogodkov, kot so "prijemanje zatiča", "prenos zatiča", "odlaganje zatiča" in "prazna roka", poleg tega pa tudi zaznavanje pogostih napak, kot je prijetje več zatičev hkrati. Vsak videoposnetek je bil anotiran s strani enega anotatorja. Posnetke so zajeli in anotirali različni anotatorji. V vsakem posnetku je bilo osemnajst ponovitev vseh štirih faz, napako sočasnega prijema dveh zatičev je vsebovalo pet posnetkov.

Celoten nabor anotiranih podatkov smo razdelili na učni, validacijski in testni nabor v razmerju 70:15:15. Učni nabor je bil uporabljen za učenje modelov, validacijski nabor za optimizacijo hiperparametrov in zgodnje ustavljanje učenja, testni nabor pa za končno, neodvisno oceno uspešnosti modelov. Razdelitev je bila izvedena časovno ločeno, kjer so bili podatki razporejeni glede na zaporedje videoposnetkov.



Slika 2: Shema uporabljene hibridne arhitekture modela, ki združuje CNN, LSTM in klasifikator.

2.3 Arhitektura in optimizacija modela

Pristop k problemu razvrščanja akcij temelji na hibridni arhitekturi globokega učenja, ki združuje konvolucijske nevronske mreže (CNN) in dolgoročne kratkoročne pomnilniške mreže (LSTM), dopolnjene z naknadno obdelavo napovedi za izboljšanje robustnosti in točnosti.

Osnovno idejo modela prikazuje Slika 2. Za izločanje prostorskih značilnic iz posameznih slihc videoposnetka smo uporabili prednaučen model ResNet [8]. Arhitektura modela ResNet18 je predstavljena v [9]. ResNet, brez zadnje polnopolovozane plasti, preslika vsako vhodno slihc v vektor značilnic. Sekvence teh značilnic so nato dovedene v LSTM plast. LSTM modeli [10] so izjemno primerni za obdelavo sekvenčnih podatkov, kot so zaporedne slihc videoposnetka, saj lahko učinkovito zaznavajo in modelirajo dolgoročne odvisnosti. To je ključno za razumevanje prehodov med različnimi fazami dela v 9HPT, kot so "prijemanje zatiča", "prenos zatiča", "odlaganje zatiča" in "prazna roka".

Integracija CNN in LSTM je pogosta praksa pri analizi videoposnetkov [11] in omogoča upoštevanje tako prostorskih (značilnice posameznih slihc) kot tudi časovnih (prehodi med stanji) odvisnosti. V našem primeru so medsebojne odvisnosti med zaporednimi slihcami strogo definirane z logičnim zaporedjem faz: "prazna roka" → "prijemanje zatiča" → "prenos zatiča" → "odlaganje zatiča". Poleg tega ni dovoljeno vračanje na prejšnje stanje, vse slihc znotraj iste akcije pa morajo imeti enako oznako. Predvidevamo, da model, ki eksplicitno upošteva ta pravila, lahko doseže višjo točnost kot zgolj razvrščanje s CNN.

Izhod LSTM plasti je povezan s polnopolovozano plastjo, ki deluje kot klasifikator in ima toliko izhodov, kolikor je iskanih stanj v videoposnetku. Za učenje modela smo uporabili kriterijsko funkcijo križne entropije, optimizacijo pa smo izvedli z algoritmom Adam. Za obravnavo morebitne neuravnoteženosti razredov smo kriterijski funkciji podali uteži, izračunane na podlagi deleža vsakega stanja v celotnem naboru oznak. Za vizualizacijo rezultatov (npr. z matriko zamenjav) in izračun točnosti, smo končno napoved razreda določili z izbiro razreda z najvišjo posteriorno verjetnostjo na izhodu klasifikatorja.

2.3.1 Priprava podatkov

Po anotaciji smo zagotovili verodostojno povezavo med oznakami in pripadajočimi sličicami videoposnetkov. Bistvenega pomena je bilo ohraniti kronološki vrstni red sličic znotraj posameznih sekvenc, saj časovna komponenta predstavlja kritično informacijo za delovanje LSTM arhitekture. Podobno je bil ohranjen tudi vrstni red sekvenc med razdelitvijo podatkov na učni in validacijski nabor, s čimer smo preprečili uhajanje informacij (ang. *data leakage*) in zagotovili neodvisnost nabora za vrednotenje.

Pred vnosom v model so bile slike predhodno obdelane. Slike so bile preslikane na standardne dimenzije 300x300 pikslov in intenzitete pikslov normalizirane, kot je določeno za modele, učene na zbirki ImageNet. Takšne vhodne vrednosti zagotavljajo optimalno izkoriščanje naučenih značilnosti modela ResNet, kar izboljšuje stabilnost in učinkovitost procesa učenja.

2.4 Sekvenčna korekcija napovedi

Za odpravljanje očitnih napak in izboljšanje konsistentnosti v napovedih modela smo implementirali dodatno funkcijo za sekvenčno korekcijo napovedi. Ta funkcija zagotavlja, da ima vsaka identificirana faza minimalno dolžino treh zaporednih sličic, hkrati pa dosledno upošteva logični vrstni red faz. Funkcija je bila do sedaj implementirana samo za napovedovanje osnovnih štirih faz in ne omogoča zaznavanja napak v izvedbi.

V prvem koraku funkcija analizira dolžino vsakega segmenta z enako napovedano oznako. Če je segment daljši od treh sličic, ohrani svojo prvotno oznako. Krajšim segmentom pa funkcija dodeli novo oznako na podlagi konteksta sosednjih faz. V primeru, ko med razredom prejšnje in naslednje sličice manjka ena faza v logičnem zaporedju, se ta manjkajoča faza pripiše obravnavanemu segmentu. V nasprotnem primeru se odločitev sprejme glede na dolžino segmenta; naprimer, segmentu dolžine ena je pripisana prejšnja oznaka, sicer pa oznaka, ki ji sledi. Ta postopek učinkovito zmanjšuje napake v napovedih in povečuje skladnost rezultatov s pričakovanimi zaporedji akcij.

2.5 Razširitve in optimizacija arhitekture

Osnovna arhitektura predlagane rešitve dopušča več variacij, katerih vpliv na zmogljivost modela je podrobneje analiziran v poglavju 3. Cilj teh razširitev je bil optimizirati robustnost in točnost rešitve.

2.5.1 Izbira in konfiguracija konvolucijske nevronske mreže

Prvotno idejo o lastni implementaciji CNN arhitekture smo nadomestili z uporabo prednaučenega modela ResNet18 [8], ki je bil naučen na obsežni podatkovni bazi ImageNet [12]. Odločitev za uporabo prednaučenega modela temelji na njegovi dokazani sposobnosti ekstrakcije bogatih in smiselnih vizualnih značilnic. Preizkusili smo tudi model ResNet50, vendar nismo zabeležili bistvenega izboljšanja v končni točnosti. Zaradi manjše računske zahtevnosti in hitreje inference, smo se odločili za uporabo kompaktnejšega modela ResNet18.

Primerjali smo dva pristopa glede konfiguracije uteži CNN: (1) uporaba že naučenih uteži v vseh slojih (tj. zamrznjene uteži) in (2) doučitev uteži zadnjih nekaj slojev. Slednji pristop je rezultiral v počasnejši konvergenci učenja brez opaznih razlik v končni točnosti. Posledično smo se odločili za uporabo v celoti prednaučenega ResNet18 z zamrznjenimi utežmi, kar zmanjšuje kompleksnost učenja in preprečuje prekomerno prilagajanje (ang. *overfitting*) majhnemu naboru podatkov.

2.5.2 Obdelava večzornih podatkov

Zaradi razpoložljivosti sočasnih posnetkov iz dveh različnih zornih kotov, smo preučili možnosti za vključitev te komplementarne informacije v model. Prvotni pristop je predvideval izločanje značilnic iz slik vsake kamere ločeno in združevanje (ang. *concatenation*) dobljenih vektorskih predstavitev v enoten vhodni vektor za LSTM. Vendar so empirični rezultati, predstavljeni v poglavju 3, pokazali, da je ta način privedel do slabših rezultatov v primerjavi z obdelavo posnetkov iz posameznega zornega kota. Dodatni eksperimenti, ki so vključevali uporabo dveh ločenih CNN-jev za vsak vhodni posnetek iz posameznega zornega kota in doučitev uteži zadnjih slojev, prav tako niso prinesli izboljšanih rezultatov. Posledično smo se odločili, da posnetke iz obeh kamer uporabimo kot ločene vhode v model, kar se je izkazalo za učinkovitejši pristop.

2.5.3 Konfiguracija LSTM plasti

Za optimizacijo delovanja LSTM komponente modela smo vrednotili različne konfiguracije. Preizkusili smo velikosti skritega sloja LSTM 256 in 512 ter primerjali zmogljivost enosmernih ter dvosmernih LSTM arhitektur. Dvosmerni LSTM (ang. *Bidirectional LSTM*) je bil vrednoten z namenom upoštevanja tako preteklih kot tudi prihodnjih kontekstualnih informacij v sekvencah. Izbrali smo dvoplastni LSTM. Na podlagi rezultatov vrednotenja, ki so prikazani v poglavju 3, smo določili optimalno dimenzijo skritega sloja na 512 in se odločili za uporabo dvosmernega LSTM.

3 Rezultati

V tem poglavju so predstavljeni in analizirani rezultati empiričnega vrednotenja različnih konfiguracij modela, ki so vodile do izbire optimalne arhitekture. Validacijske in testne točnosti različic modela, predstavljenih v nadaljevanju, so zbrane v Tabeli 1.

3.1 Začetno vrednotenje in izzivi

Pred implementacijo LSTM komponente smo izvedli preliminarno vrednotenje razvrščanja akcij z uporabo zgolj prednaučenega modela ResNet18 v kombinaciji z naučenim klasifikatorjem. Model, ki je dosegel najvišjo validacijsko točnost približno 54 %, je na testnem naboru podatkov dosegel točnost 34 %. Matrika zamenjav rezultatov testiranja (Slika 3a) jasno prikazuje inherentno težavo, ki se je pojavljala skozi proces razvoja modela: kljub uporabi uteži v kriterijski funkciji je bil model nagnjen k napovedovanju dominantnega razreda, pogosto

Tabela 1: Rezultati različnih preizkušenih modelov.

Različica	Validacijska točnost / %	Testna točnost / %
učenje klasifikatorja	54	34
uporaba enega zornega kota	65	/
uporaba dveh zornih kotov	65	/
dimenzija skritega sloja LSTM 256	45	/
dimenzija skritega sloja LSTM 512	52	/
dvosmerni LSTM (dimenzija 512)	65	/
dolžina sekvence 20 sličic	65	60
dolžina sekvence 30 sličic	65	60
dolžina sekvence 40 sličic (izbrani model)	70	63
detekcija napake	41	/

najpogostejšega stanja. Pričakovali smo, da bo vključitev LSTM modula, ki omogoča modeliranje časovnih odvisnosti, pomagala pri blažitvi te pristranskosti in izboljšala točnost razvrščanja posameznih sličic.

3.2 Vpliv večzornih podatkov

Primerjava učenja modelov, ki obdelujejo enojni oziroma dvojni zorni kot, je prikazana na Sliki 4. Oba modela sta dosegla validacijsko točnost okoli 65 %. Vendar pa je bilo učenje modela z dvema vhomoma bistveno bolj nestabilno in nagnjeno k prekomernemu prilaganju. Na podlagi teh ugotovitev smo se za nadaljnje učenje odločili za model, ki obdeluje informacije iz enega samega zornega kota.

3.3 Optimizacija LSTM konfiguracije

Podrobno smo vrednotili vpliv različnih dimenzij skrite plasti v LSTM in primerjali zmogljivost enosmerne in dvosmerne LSTM. Preizkušene velikosti skrite plasti so bile 256 in 512, saj v obravnavanem primeru ne pričakujemo izjemno kompleksnih časovnih vzorcev, hkrati pa je na voljo omejena količina učnih podatkov. Izbrali smo velikost 512, ki je dosegla validacijsko točnost 52 %, kar je znatno bolje od velikosti 256, ki je dosegla 45 % validacijsko točnost. Dvosmerni LSTM je z validacijsko točnostjo 65 % očitno presegel zmogljivost enosmerne LSTM z dimenzijo 512. Matrika zamenjav za model z najboljšo validacijsko uspešnostjo je prikazana na Sliki 3b, kjer je že opazen izboljšan vzorec pravilnega odločanja v primerjavi s Sliko 3a.

3.4 Vpliv dolžine sekvence in končni model

Po določitvi optimalne konfiguracije modela smo nadalje raziskali vpliv dolžine vhodne sekvence na zmogljivost modela. Preizkusili smo dolžine sekvenc 20, 30 in 40 sličic, model pa smo učili na povečanem obsegu podatkov. Najboljšo zmogljivost je dosegla dolžina sekvence 40 sličic, ki je na validacijskem naboru dosegla 70 % točnost, na testnem naboru pa 63 %. Krajši sekvenci sta

imeli nekoliko nižjo točnost, in sicer okoli 65 % na validacijskem naboru in okoli 60 % na testnem naboru.

Končni, optimalni model vključuje prednaučen model ResNet18, obdeluje informacije iz ene kamere naenkrat, uporablja dvosmerni LSTM z dimenzijo skrite plasti 512 in dvema plastema ter sprejema sekvence sličic dolžine 40. Njegova testna uspešnost je znašala 63 %, pripadajoča matrika zamenjav pa je prikazana na Sliki 3c.

3.5 Detekcija napake

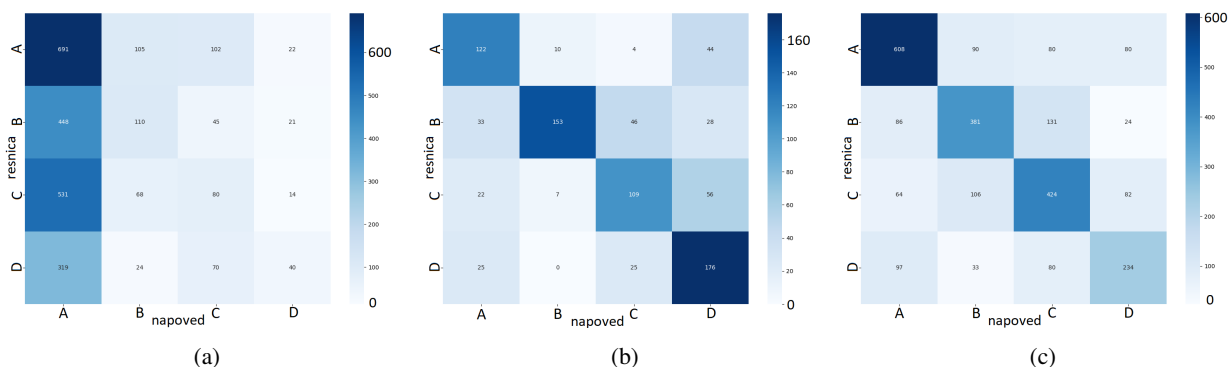
Za izvedbo nadgradnje, ki omogoča zaznavanje nedovoljenega prijema dveh zatičev, smo uporabili isto konfiguracijo modela kot za osnovni model. Model je bil ponovno naučen na naboru posnetkov, ki so vsebovali to specifično napako. Zaradi bistveno manjšega števila posnetkov, ki so vsebovali napake, v primerjavi s posnetki pravilno izvedenih akcij, so bili rezultati pričakovano slabši. Sličice iz štirih posnetkov, ki so vključevale napako, so bile uporabljene za učenje modela, medtem ko so bile sličice enega videoposnetka uporabljene za validacijo, ki je dosegla 41 % točnost. Zaradi pomanjkanja podatkov neodvisna testna množica za vrednotenje najboljšega modela ni bila na voljo. Uporaba funkcije za sekvenčno korekcijo napovedi (opisane v poglavju 2.3) v tem primeru ni bila ustrezna zaradi manjše determinističnosti pojavljanja napak. Za doseganje boljših rezultatov bi bilo nujno učenje na znatno večjem številu učnih posnetkov z napakami, poleg tega pa bi bila smiselna tudi prilagoditev oziroma dodajanje specifičnega naknadnega procesiranja napovedi za ta tip problema.

4 Razprava

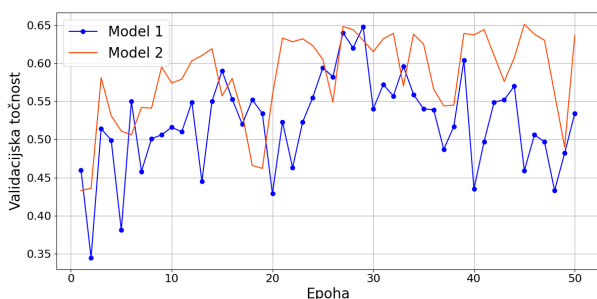
Analiza predstavljenih rezultatov razkriva pomembne vpoglede v zmogljivost in omejitve razvitega modela. Opazno je, da so bili rezultati na testnem naboru praviloma slabši od tistih, doseženih na validacijskem naboru. Ta diskrepanca izhaja iz metodologije razvoja, kjer so bile odločitve o arhitekturi in hiperparametrih modela sprejete na podlagi validacijskih rezultatov. Pri tem je pomembno poudariti, da je bil validacijski nabor sestavljen iz videoposnetkov istih avtorjev kot učni nabor, medtem ko so bili za končno testiranje uporabljeni povsem novi, neodvisni posnetki. Poleg tega so bili avtorji anotacij učnih, validacijskih in testnih posnetkov različni, kar lahko pomembno prispeva k negotovosti vrednotenja. Pomanjkanje zadostnega števila podatkov za povsem neodvisne testne množice na vsakem koraku razvoja modela je pomenilo, da so bili kompromisi nujni, kar potencialno pojasnjuje omenjeno razliko med validacijsko in testno točnostjo.

4.1 Alternativni pristopi in potencialne izboljšave

Poleg predstavljene hibridne arhitekture CNN-LSTM obstajajo tudi drugi alternativni pristopi, ki bi lahko ponudili izboljšave. Sem sodi zlasti uporaba 3D konvolucijskih nevronske mreže (3D CNN), ki so intrinzično zasnovane za obdelavo prostorsko-časovnih značilnic v videoposnetkih, in konvolucijskih LSTM (ConvLSTM) [13]. Obe metodi sledita podobni ideji združevanja informacij



Slika 3: Matrike zamenjav rezultatov testiranja. Oznake: A - "prazna roka", B - "prijemanje zatiča", C - "prenos zatiča", D - "odlaganje zatiča". (a) Matrika zamenjav rezultatov testiranja prednaučenega ResNet18 in naučenega klasifikatorja. Vidna je težava, na katero smo tekom razvoja modela večkrat naleteli, kjer je model kljub uporabi uteži v kriterijski funkciji nagnjen k napovedovanju prevladujočega razreda. (b) Matrika zamenjav rezultatov validacije dvosmernega LSTM z dimenzijo skrite plasti 512. (c) Matrika zamenjav rezultatov testiranja izbranega, najboljšega modela.



Slika 4: Primerjava krivulj učenja modela z dvema vhodnima sličicama (Model 1) in modela z individualno obravnavo sličic iz enega zornega kota (Model 2).

iz posameznih sličic in modeliranja odvisnosti med zaporednimi slikami. Možna je tudi kombinacija 3D CNN z LSTM [14] za kompleksnejše probleme. Raziskava teh alternativnih metod bi potencialno lahko privedla do izboljšave rezultatov, saj omogočajo zajemanje kompleksnejših prostorsko-časovnih korelacij.

V okviru obstoječega pristopa so bili identificirani nekateri dodatni elementi, ki bi potencialno izboljšali točnost modela, vendar niso bili uspešno implementirani v tej fazi razvoja. Mednje spadajo kodiranje položaja sličice v zaporedju (ang. *positional embedding*), ki bi omogočilo LSTM modulu boljše razumevanje časovne strukture dogajanja; prenašanje vrednosti skritih stanj oz. spomina LSTM med zaporednimi paketi (ang. *stateful LSTM*), kar bi bilo koristno za modeliranje zelo dolgih časovnih odvisnosti; in prilagojena kriterijska funkcija, ki bi poleg nepravilno označenih sličic kaznovala tudi nepravilne prehode med stanji, s čimer bi model neposredno usmerjala k učenju logičnih zaporedij akcij.

Dodatne raziskave in implementacije teh pristopov bi bile smiselne za morebitno povečanje robustnosti in točnosti modela. Predvsem pa je potrebno razširiti nabore podatkov, zlasti za redkejše dogodke in napake, za izboljšanje sposobnosti posploševanja modela in zmanjšanje razlike

med validacijsko in testno zmogljivostjo.

5 Zaključek

V tem članku smo razvili in ovrednotili hibridni model, ki združuje konvolucijske nevronske mreže (CNN) in dolgoročne kratkoročne pomnilniške mreže (LSTM) za avtomatsko razvrščanje akcij in zaznavanje napak pri manipulaciji zatičev pri 9HPT. Na podlagi zajema in anotacije lastnega nabora podatkov smo optimizirali arhitekturo modela. Ugotovljeno je bilo, da prednaučen ResNet18 učinkovito služi kot izločevalnik značilnic, dvosmerni LSTM pa optimalno modelira časovne odvisnosti. Pomembni dejavniki za izboljšanje točnosti so bili ločena obdelava posnetkov iz več zornih kotov in sekvenčna korekcija napovedi. Končni model je dosegel 63 % točnost na testnem naboru. Kljub doseženim rezultatom je razlika med validacijsko in testno točnostjo poudarila izzive pri posploševanju zmogljivosti modela. Prihodnje delo bo vključevalo raziskovanje naprednejših arhitektur (npr. 3D CNN, ConvLSTM) in pridobivanje obsežnejšega nabora podatkov, kar je ključno za izboljšanje robustnosti in zanesljivosti detekcije stanj 9HPT v realnih scenarijih.

Literatura

- [1] K. Oxford Grice, K. A. Vogel, V. Le, A. Mitchell, S. Muniz, M. A. Vollmer, "Adult Norms for a Commercially Available Nine Hole Peg Test for Finger Dexterity," *The American Journal of Occupational Therapy*, vol. 57, no. 5, pp. 570–573, 2003.
- [2] S. Fariss, *Automatic evaluation of the Nine Hole Peg Test for subjects with neurological pathologies through artificial vision approaches*, magistrsko delo, Politecnico di Torino, Corso di laurea magistrale in Mechatronic Engineering, 2020. Mentorica: G. Olmo.
- [3] E. Prochaska, M. Sedlmayr, "Advancements in Dexterity Measurements: the digital Nine Hole Peg Test: Validation results of a digital innovation in dexterity assessment," *Current Directions in Biomedical Engineering*, vol. 10, no. 4, pp. 510–513 2024.

- [4] E. Prochaska, E. Ammenwerth, "Validity and Reliability of a New Developed Digital Version of Nine Hole Peg Test," *IEEE Access*, vol. 11, pp. 97169-97176, 2023.
- [5] X. Jiang, M. McGinley, J. Johnston J, et al, "A digital version of the nine-hole peg test: Speed may be a more reliable measure of upper-limb disability than completion time in patients with multiple sclerosis," *Multiple Sclerosis Journal*, vol. 31, no. 1, pp. 81–92, 2024.
- [6] N. Čelofiga, U. Puh, "Merske lastnosti testa devetih zatičev pri pacientih z multiplo sklerozo in pacientih po možganski kapi," *Fizioterapija*, vol. 29, no. 1, pp. 35–44, 2021.
- [7] M. Tkachenko, M. Malyuk, A. Holmanyuk, N. Liubimov, *Label Studio: Data labeling software*, LabelStudio, 2020-2025. <https://github.com/HumanSignal/label-studio>. Dostopano: 28. 5. 2025.
- [8] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [9] F. Ramzan, M. U. Khan, A. Rehmat, S. Iqbal, T. Saba, A. Rehman, Z. Mehmood, "A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages Using Resting-State fMRI and Residual Neural Networks," *Journal of Medical Systems*, vol. 44, 2019.
- [10] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [11] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, S. W. Baik, "Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.
- [12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.
- [13] M. Venkataswamy, N. Lanier, A. Lam, *Temporally-Aware Deep Learning for Video Frame Classification*, Medium, 2021. <https://medium.com/convolutional-lstm-neural-networks-for-video-frame/neural-networks-for-classification-of-spatio-temporal-data-beb1693056f>. Dostopano: 10. 5. 2025.
- [14] X. Wang, L. Gao, J. Song, and H. Shen, "Beyond Frame-level CNN: Saliency-Aware 3-D CNN With LSTM for Video Action Recognition," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 510–514, 2017.