TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Informaatika õppekava

Rasmus Vahtra

Mobiilirakenduse loomine tehisnärvivõrgule

Bakalaureusetöö (9 EAP)

Juhendaja: Tambet Matiisen

Tartu 2015

Mobiilirakenduse loomine tehisnärvivõrgule

Lühikokkuvõte:

Käesoleva töö raames valmis lahendus mobiilirakendusest, serveriliidesest ning tehisnärvivõrgust. Täpsemalt on seletatud projektis kasutatud tehisnärvivõrgu arhitektuuri ning ehitust, ühtlasi ka loodud lahenduse tööpõhimõtet. Samuti on välja toodud ka, et milleks selline lahendus üldse hea olla võiks.

Võtmesõnad:

Tehisnärvivõrgud, süvaõpe, Objective-C, mobiilirakendus

Title in English: Mobile application for artificial neural network

Abstract:

This paper describes the created project that consists of mobile application, server and artificial neural network. More details can be found about the architecture of the artificial neural network used in this project, but also the description of workflow of the given project. Several examples on why this project can be good are also presented.

Keywords:

Artificial neural networks, deep learning, Objective-C, mobile application

Sisukord

Sissejuhatus 4
1. Tehisnärvivõrgu idee 5
1.1 Tehisnärvivõrkude ajalugu 5
1.2 ImageNet võistlus 6
1.3 Projektis kasutatud tehisnärvivõrgu arhitektuur 7
1.4 CaffeNet tehisnärvivõrgu treenimine 12
1.5 Graafikaprotsessorite kasutamise eelis tehisnärvivõrkudega 12
2. Projekti lahendus 13
2.1 Mobiilirakendus 13
2.2 Serverirakendus 15
2.3 Tehisnärvivõrk 16
3. Tulemuste analüüs 17
3.1 Vastuse saamine 17
3.2 Vastuse täpsus 17
3.3 Ühenduse kiirused 19
4. Sarnased lahendused 20
4.1 Microsofti lahendus 20
4.2 TapTapSee 20
5. Projekti rakendatavus 21
5.1 Meditsiin 21
6. Edasiarenduse võimalused 22
6.1 Mitme tehisnärvivõrgu variandid 22
6.2 Realiseerida tehisnärvivõrk mobiilseadmes 22
6.3 Treenida uus närvivõrk seenteliikide klassifitseerimiseks 22
6.4 Mobiilirakenduse koodi optimiseerimine iOS-ile 22
6.5 Rakenduse laiendamine Android seadmetele 22
7. Kokkuvõte 23
8. Tsiteeritud teosed 24
Lisad 26
I. Terminid 26
II. Lähtekoodid 27
III. Litsents 28

Sissejuhatus

Projekti eesmärgiks on uurida tehisnärvivõrkude olemust ja töömehaanikat ning luua mobiilrakendus, mille abil saaks neid ära kasutada. Selleks uuriti lähemalt 2012. aastal korraldatud „ImageNet Large Scale Visual Recognition Challenge 2012“ konkursi võitjate loodud tehisnärvivõrku [1]. Uurimise käigus tehti selgeks, mida tehisnärvivõrk endast kujutab, kuidas konkreetne CaffeNet mudel loodud on ja kuidas seda on treenitud. Sama mudelit kasutati ka käesoleva töö käigus loodud lahenduse loomises, mis võimaldab teha mobiilirakendusega foto ning rakendus näitab, mis selle foto peal on. Selle jaoks pandi kõnealune tehisnärvivõrk spetsiaalsesse serverisse, sinna juurde Pythonis loodud serveriliides. Mobiilirakendus loodi Apple-i iPhone telefonile. Serveriliidese kaudu toimub suhtlus telefoni ja tehisnärvivõrgu vahel.

Töö esimese peatükis tutvustatakse lähemalt tehisnärvivõrgu ehitust, seletatakse, kuidas nad töötavad ning kuidas neid treenitakse. Lisaks tutvustatakse ka ImageNet võistlust, kus taolisi tehisnärvivõrke testitakse. Teises peatükis seletatakse lahti, kuidas tehtud projekt disainiti ning kuidas see täpselt töötab. Kolmandas peatükis analüüsitakse projekti tulemusi ning räägitakse vigadest, mis tehisnärvivõrgu puhul võivad kergelt tekkida. Neljandas peatükis on tutvustatud kahte sarnast lahendust. Viiendas peatükis räägitakse, kuidas antud lahendust ka kasulikult rakendada saaks. Kuuendas peatükis tuuakse välja osad võimalused, kuidas antud lahendust saaks edasi arendada ja paremaks teha.

1. Tehisnärvivõrgu idee

Selles projektis uuritakse lähemalt tehisnärvivõrku, mis on treenitud klassifitseerima pilte, ning luuakse mobiilirakendus, mille abil on võimalik mobiilikaameraga pilti teha ja tehtud pilt tehisnärvivõrgule analüüsimiseks saata. Selleks otstarbeks on kasutatud eeltreenitud tehisnärvivõrgu mudelit, mille disain põhineb „ImageNet Large Scale Visual Recognition Challenge 2012“ (ILSVRC) võistluse võitnud tehisnärvivõrgul. ILSVRC on iga-aastane võistlus, kus erinevate teadlaste loodud tehisnärvivõrgu lahendused proovivad saada piltide klassifitseerimisel kõige täpsemaid tulemusi. Kõnealuse mudeli disain saavutas esimese koha ning ületas märkimisväärselt varasemaid tulemusi. See oli oluliseks murdepunktiks tehisnärvivõrkude rakendamisel pildituvastuses.

1.1 Tehisnärvivõrkude ajalugu

Järgnev on lühikokkuvõte tehisnärvivõrkude ajaloost [2].

Aastal 1943 kirjutasid neurofüsioloog Warren McCulloch ja matemaatik Walter Pitts teooria selle kohta, kuidas neuron töötada võiks. Selle kirjeldamiseks disainisid nad lihtsa tehisnärvivõrgu vooluringina.

1949 kirjutas Donald Hebb oma teoses „The Organization of Behavior“ neuronite vahelistest ühendustest. Iga kord, kui kahe neuroni vahelist ühendust kasutatakse, muutub see tugevamaks, mis kirjeldabki põhimõtteliselt inimeste õppimisprotsessi.

1957 leiutas Frank Rosenblatt pertseptroni, kasutades McCulloch-Pittsi neuronit ja Donald Hebbi avastusi. Pertseptron on oluline leiutis, kuna see suudab kaalude abil sisendit mõjutada ning seeläbi ka õppida. Pertseptronid olid hiljem olulisel kohal tehisnärvivõrkude tekkes.

1959 arendasid Bernard Widrow ja Marcian Hoff mudeli nimega „MADALINE“. See oli esimene tehisnärvivõrk, mida rakendati reaalelulise probleemi lahendamiseks. Nimelt oli sellel ajal telefoniliinides palju kaja, „MADALINE“ oskas seda vähendada. Süsteem on küll väga vana tehnoloogiamõistes, kuid on siiani kasutuses.

1962 arendasid Bernard Widrow ja Marcian Hoff õppimismeetodi, mis uurib väärtusi enne, kui kohendab kaalusid. Idee seisneb selles, et kui üks pertseptron (õppimisalgoritm) võib tulemuseks saada suure vea, siis on võimalik see ära jaotada terve võrgu peale (või vähemalt lähedal seisvatele pertseptronidele).

1975 loodi esimene mitmekihiline tehisnärvivõrk.

1986 avaldavad David Rumelhart, Hinton ja Williams raamatu „Learning Internal Representation by Error Propagation“, mis propageerib tagasilevi (*backpropagation*) algoritmi mitmekihiliste tehisnärvivõrkude treenimiseks.

1989 rakendab LeCunn konvolutsioonilisi tehisnärvivõrke käsitsi kirjutatud numbrite tuvastamiseks. 1990. aastatel töödeldi selle süsteemi abil 10% Põhja-Ameerika riikides käibel olnud tšekkidest [10].

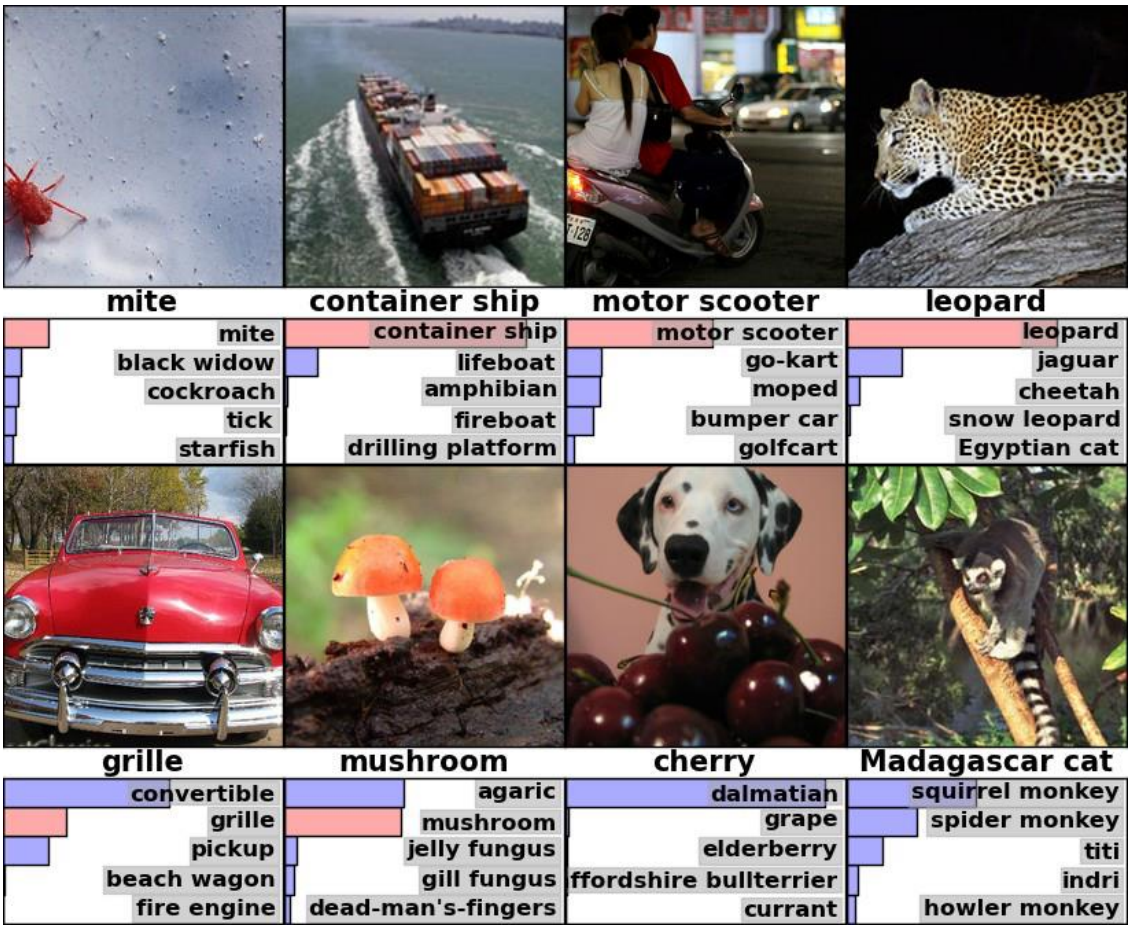
2006 näitab Geoffrey Hinton, kuidas saab eeltreenida sügavaid tehisnärvivõrke piirangutega Boltzmanni masinate (*Restricted Boltzmann Machines*) abil [10].

2012 – Alex Križevsky, Ilya Sutskever ja Geoffrey Hinton võidavad ImageNet võistluse [1]. Google kasutab süvaõpet Androidi telefonides kõnetuvastuse jaoks [17].

1.2 ImageNet võistlus

ILSVRC hindab erinevaid algoritme, mis tuvastavad ja klassifitseerivad piltidelt objekte. Võistlust korraldatakse selleks, et teadlastel oleks võimalik võrrelda oma tulemusi antud valdkonnas. Ühtlasi võimaldab selline võistlus efektiivselt ära kasutada piltide andmebaasi, mille loomiseks on kulunud väga palju töötunde, ning uurida, kui kaugele on jõutud arvutite „nägemises“.

Võistluse käigus peavad võistlusele esitatud lahendused tuvastama ja klassifitseerima etteantud piltidel olevaid esemeid.



Joonis 1. Näide sellest, kuidas tehisnärvivõrk piltidel olevaid objekte klassifitseerib. Pildi all on õige vastus, selle all viis parimat pakkumist tehisnärvivõrgu poolt.

Selleks kasutatakse käsitsi klassifitseeritud pilte ImageNet andmebaasist, kus on ligi 10 miljonit pilti 10 000 erinevas kategoorias. Võistluse jaoks kasutatakse 1,2 miljonit pilti 1000-st erinevast klassist. Sellest 50 000 pilti antakse valideerimiseks ning 100 000 pilti on reserveeritud testimiseks. Viimaseid enne testimist ei avalikustata.

Klassifitseerimise käigus peavad algoritmid väljastama viis tulemust tõenäosuse kahanemise järjekorras, mis kajastavad nende arvamust sellest, mis pildi peal on. Tulemusi hinnatakse selle järgi, mida algoritm pakkus kõige esimeseks ning kas õige objekt ka üldse üles leiti (kas on viie pakutud tulemuse seas).

Viimased kaks aastat hinnatakse võistlusel lisaks objekti klassifitseerimisele ka objekti asukoha tuvastamise täpsust. Olemas on veel eraldi koeratõugude määramine. Üks asi on aga kõikidel aastatel samaks jäänud ning see on TOP-1 ja TOP-5 veaarvutused. TOP-1

puhul vaadatakse, kas algoritm sai õige vastuse või mitte ning tulemus väljastatakse veaprotsendina (ehk siis kõik eksimused jagatud kõikide määramistega). TOP-5 veaarvutuse puhul aga vaadatakse, kas õige objekt oli algoritmi poolt pakutud tulemuste nimekirjas või mitte. Selle eesmärgiks on paindlikkus juhuks, kui piltidel on näiteks rohkem kui üks objekt ning algoritm need kõik ka üles leiab, kuid ei oska öelda, milline neist kõige tähtsam olla võiks. Sellisel juhul loetakse tulemus õigeks, kui õige objekt tulemuste nimekirjas on. Veaprotsent leitakse analoogselt TOP-1'le.

Kõige tähtsam tulemus ImageNet võistlusel ongi TOP-5. See on peamine võrdlusalus erinevate algoritmide vahel [4]. Kui vaadata sama võistluse viimaseid tulemusi, mis pärinevad aastast 2012, siis on selgelt näha, et tehisnärvivõrkude klassifitseerimisveaprotsent on oluliselt vähenenud [11].

1.3 Projektis kasutatud tehisnärvivõrgu arhitektuur

Selle projekti raames uuriti ja kasutati Alex Križevsky loodud arhitektuuril põhinevat peaaegu identset koopia võrgust, mis võitis ImageNet võistluse 2012. aastal. Võrk on loodud Caffe programmiga ning on eeltreenitud.

Võrk koosneb kaheksast kaaludega kihist, millest esimesed 5 on konvolutsioonilised ja viimased 3 täisühendatustega. Viimase täisühendustega kihi väljund suunatakse *softmax* kihti, mis arvutab vastavalt sisendile tõenäosused tuhandele erinevale klassile.

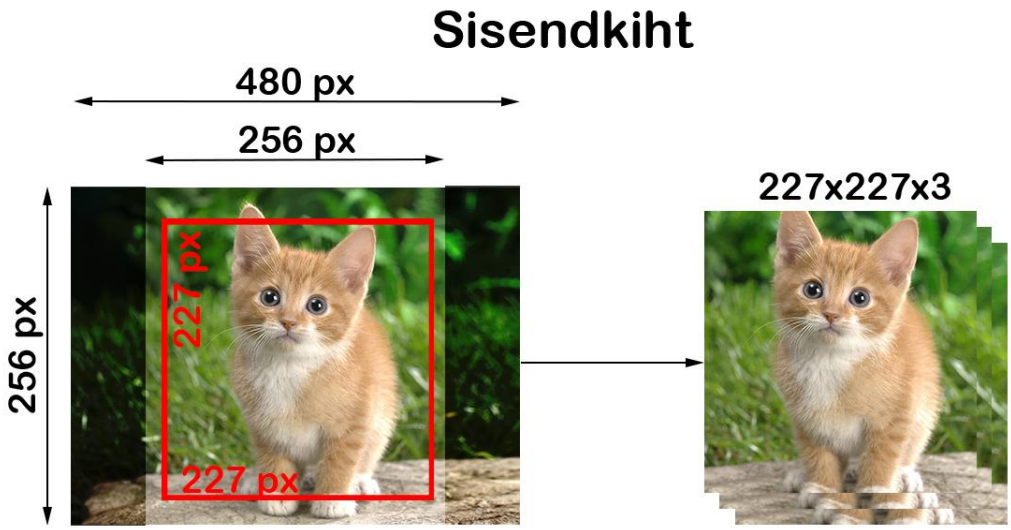
Esimesel, teisel ja viiendal konvolutsioonilisel kihil on veel *max-pooling* kihid, mis selekteerivad kõige aktiivsemad osad ning edastavad need järgmisesse kihti.

Esimeses konvolutsioonilises kihis võetakse ette antud pildist 227 x 227 x 3 suurune maatriks, kus x3 tähistab kolme värvikanalit (punane, roheline, sinine). Seda maatriksit analüüsitakse filtriga, mille suurus on 11 x 11 x 3 ning filtrisammuks on 4. Filtreid rakendatakse 96 tükki. Niimoodi toodetakse uus maatriks suuruses 55 x 55 x 96 (kus x96 tähistab uute maatriksite arvu – uus maatriks iga filtri kohta). Uus maatriks läheb edasi *max-poolingusse*, kus on filtrid suurusega 3 x 3, filtrisammuks 2 ning filtreid 96. Selle tulemusel tekib maatriks suurusega 27 x 27 x 96, mis saadetakse teise konvolutsioonikihti. Teises kihis analüüsitakse sisendit filtritega 5 x 5 x 96 (kus x96 tähistab kanalite arvu, mis tekkis esimeses konvolutsioonikihis) ning filtrisammuks on 1. Filtreid rakendatakse 256 tükki. Selle tulemusena tekib väljund, mis on suurusega 27 x 27 x 256. See suunatakse teise konvolutsioonikihi *max-pooling* kihist, mille filtri suurus on 3 x 3 ning filtrite arvuks 256 ja filtrisammuks 2. Selle tulemusel tekib väljund parameetritega 13 x 13 x 256.

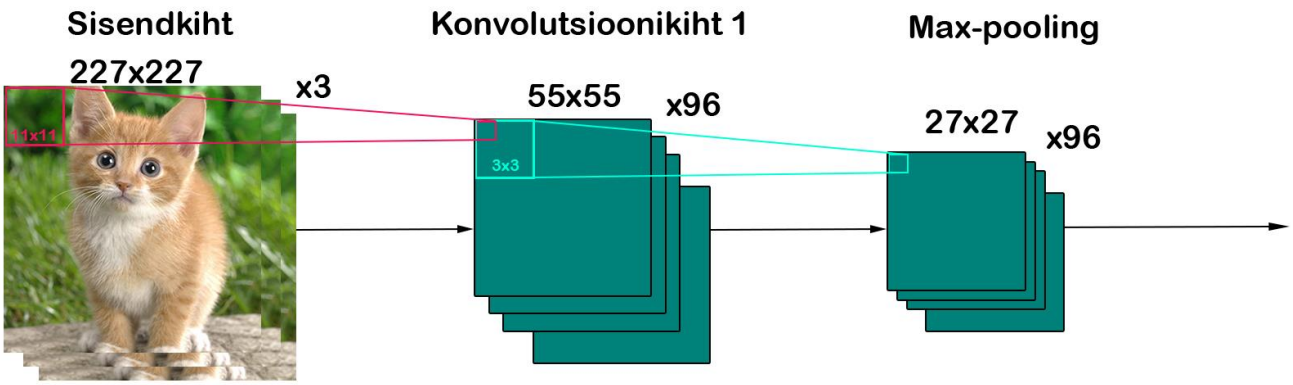
Kolmas, neljas ja viies konvolutsiooniline kiht on omavahel seotud ilma vahepealsete *max-pooling* kihtideta, ehk siis eelmise kihi väljund on järgmise kihi otsene sisend. Kolmandas konvolutsioonilises kihis võetakse vastu sisend suurusega 13 x 13 x 256, lisatakse ääris (*padding*) suurusega 1, kasutatakse filtreid suurusega 3 x 3 x 256, filtrite arvuks on 384 ning filtrisammuks 1. Neljandas kihis lisatakse ääris suurusega 1, kasutatakse filtreid suurusega 3 x 3 x 384, filtrite arvuks on 384 ning filtrisammuks 1. Viiendas kihis lisatakse ääris suurusega 1, kasutatakse filtreid suurusega 3 x 3 x 384, filtrite arvuks on 256 ning filtrisammuks 1. Pärast viiendat kihti on ka *max-pooling* kiht, kus rakendatakse filtrit suurusega 3 x 3, filtrite arvuks on 256 ning filtrisammuks 2. See toodab väljundi mõõtudega 6 x 6 x 256.

Kuues, seitsmes ja kaheksas kiht on täisühendustega kihid. Neil pole enam filtreid. Kuuendas kihis surutakse saadud sisend väljundisse 4096 väärtusega, sest seal on 4096 võrgusõlme. Seitsmendas kihis võetakse vastu sisend 4096 väärtusega ning väljastatakse väljund 4096 väärtusega. Kaheksandas kihis võetakse samuti vastu sisend 4096 väärtusega

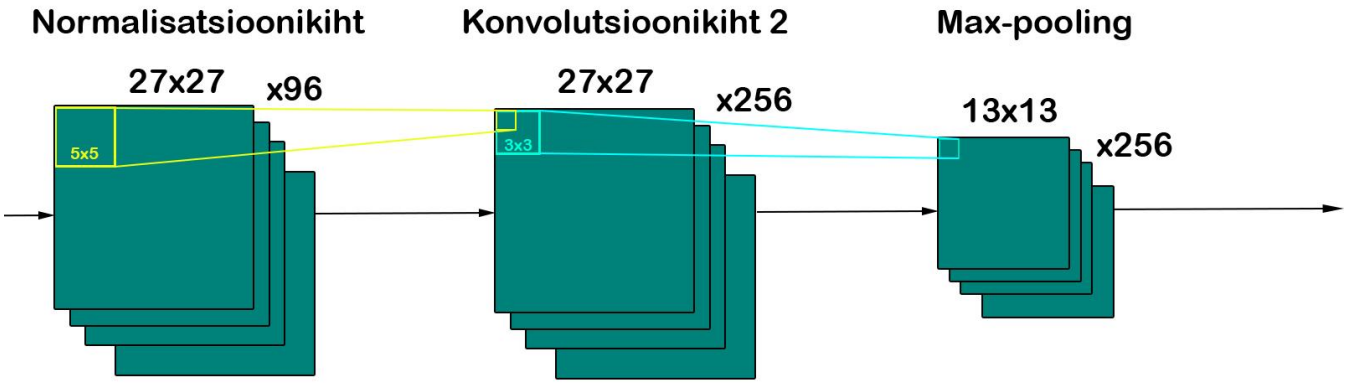
ning surutakse see tõenäosuskihis 1000 erinevaks väärtuseks, mis ongi lõpuks kogu võrgu lõppväljundiks. Viimased 1000 väärtust esindavad neid klasse, mida antud võrk on treenitud ära tundma. Sõltuvalt pildil olevast infost saab mingi väärtus rohkem „lööke“ ning muutub seeläbi aktiivsemaks (väärtus suureneb). Lõpptulemusena tähistab iga väärtus vastava klassi tõenäosust.



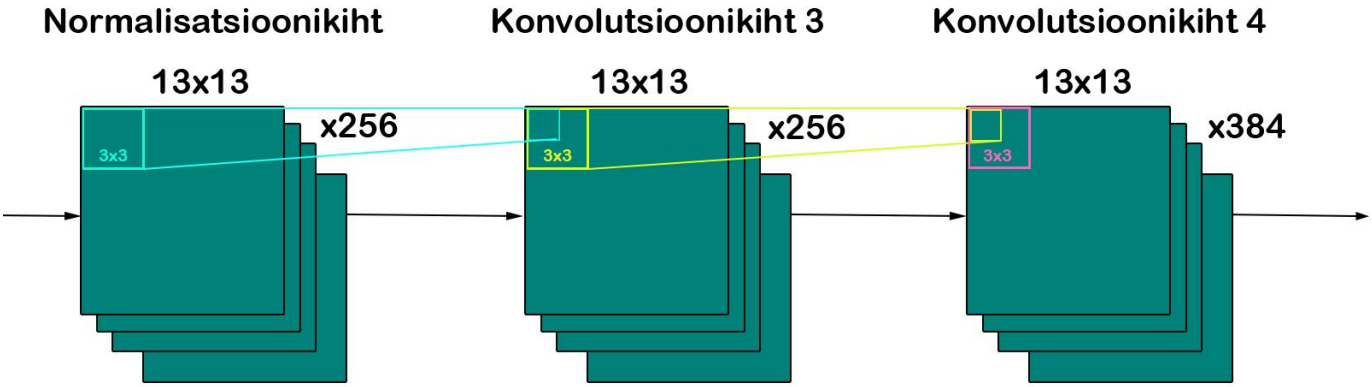
Joonis 2. Sisendkihi töötlus. Esialgsest pildist lõigatakse välja 256x256 ruut, millest treenimisel omakorda lõigatakse juhuslikult välja 227x227 osa, et treeningandmete hulka suurendada ja muuta võrk vähem tundlikuks objekti asukoha suhtes.



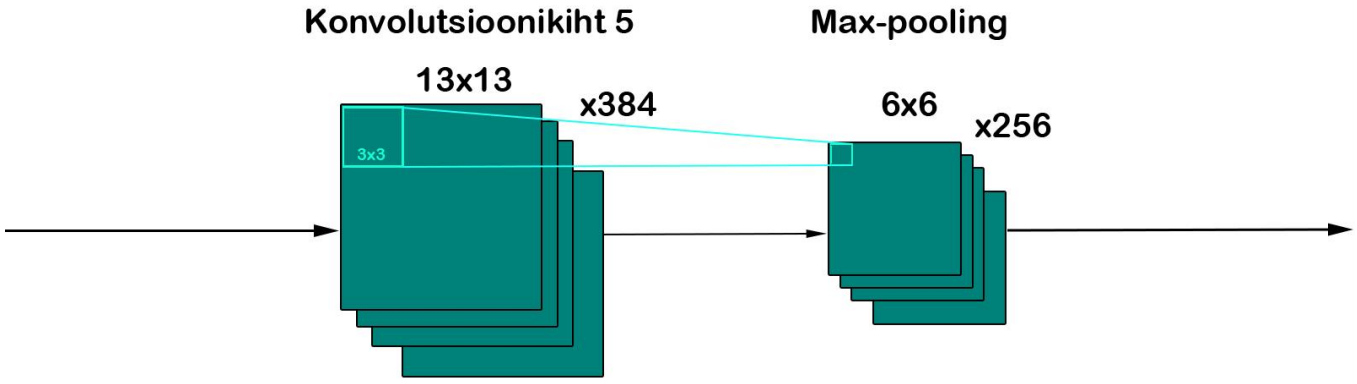
Joonis 3. Joonis näitab, kuidas toimub pildianalüüsimine esimestes kihtides. Filtrite projektsioonid on siin ja edaspidi erinevat värvi parema eristuse huvides.



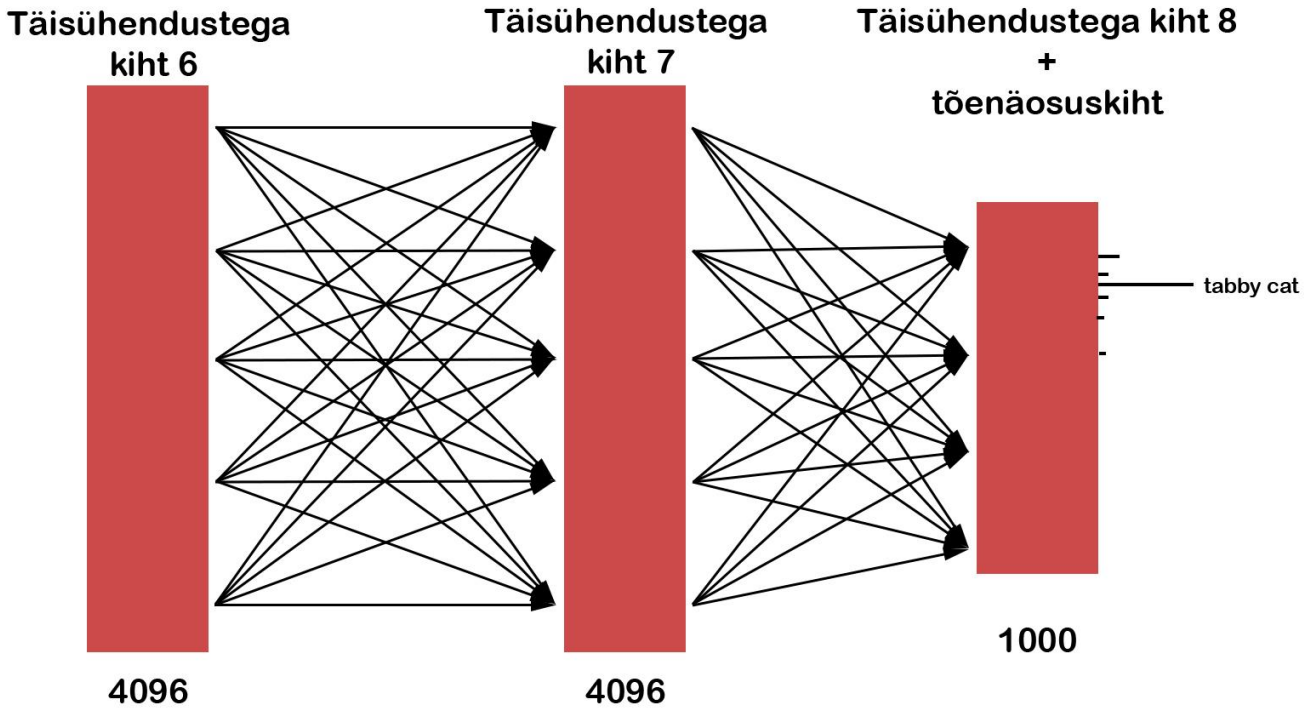
Joonis 4. Analüüsimine liigub teise kihti.



Joonis 5. Protsessid kolmandas ja neljandas kihis.



Joonis 6. Viienda kihi protsessid.



Joonis 7. Täisühendustega kihid. Nooled kujutavad ühendusi erinevate võrgusõlmede vahel.

Tabel 1. Projektis kasutatud tehisnärvivõrgu arhitektuur.

Kihid	**Laius**	**Kõrgus**	**Sügavus**	**Ääris (*padding*)**	**Filtri laius**	**Filtri kõrgus**	**Filtri sügavus**	**Filtrite arv**	**Filtri samm**	**Väljund**	**Kaalusid**	**Vabaliikmeid**	**Parameetreid**
conv1	227	227	3		11	11	3	96	4	290400	11x11x3x96	55x55	37873
pool1	55	55	96		3	3		96	2	69984			
conv2	27	27	96	2	5	5	96	256	1	186624	5x5x96x256	27x27	615129
pool2	27	27	256		3	3		256	2	43264			
conv3	13	13	256	1	3	3	256	384	1	64896	3x3x256x384	13x13	884905
conv4	13	13	384	1	3	3	384	384	1	64896	3x3x384x384	13x13	1327273
conv5	13	13	384	1	3	3	384	256	1	43264	3x3x384x256	13x13	884905
pool5	13	13	256		3	3		256	2	9216			
fc6	6	6	256							4096	6x6x256x4096	4096	37752832
fc7	4096									4096	4096x4096	4096	16781312
fc8	4096									1000	1000x4096	1000	4097024
												Kokku	**62381253**

1.4 CaffeNet tehisnärvivõrgu treenimine

Kuna siin projektis on kasutatud Caffe tarkvaral põhinevat tehisnärvivõrku, siis keskendutakse CaffeNet näidisvõrgu treenimisele.

Treenimise jaoks peab olema andmestik, antud juhul pildid, mida saab siis tehisnärvivõrgule klassifitseerimiseks anda. CaffeNeti lahenduse puhul on treeningandmed eraldi kaustas, kus nad on jaotatud klasside kaupa. Eraldi on olemas fail, kus on kirjas, et mis pildiga tegu on. Lisaks treeningandmetele on olemas ka valideerimisandmed. Viimaseid enam treeninguks ei kasutata, vaid nende abil hinnatakse treenimistulemusi. See aitab analüüsida, et milliseid hüperparameetreid peaks muutma treenimise ajal, et oleks võimalik saavutada parem tulemus. Kõige lõpuks toimub testimine, kus lastakse tehisnärvivõrgul klassifitseerida teatud hulga pilte ning hinnatakse tulemusi. Kuigi olemuselt on valideerimine ja testimine sarnased, kasutatakse valideerimist treenimise ajal, et oleks võimalik treenimise ajal vastavaid korrektsioone teha, ning testimist selleks, et saaks aimu lõpptulemusest.

Väljundi puhul kasutavad mõlemad võrgud *softmax* kihti. Treenimisel arvutatakse selle põhjal veafunktsioon (*loss function*) ning muudetakse võrgu kaalusid tagasilevi (*backpropagation*) algoritmi abil. Testimisel on kasutusel veel täpsuskiht (*accuracy*), mida kasutatakse täpsuse hindamiseks testandmestiku korral [16].

1.5 Graafikaprotsessorite kasutamise eelis tehisnärvivõrkudega

Kuigi tänapäevased protsessorid (*central processing unit*, CPU) on väga võimsad ning nende võimsus kasvab ajas veelgi, on graafikaprotsessorid (*graphical processing unit*, GPU) teatud ülesannete täitmisel siiski kordades kiiremad. Seepärast kasutataksegi vahel CPU-de asemel GPU-sid või kasutatakse mõlemaid korraga. Põhjus miks see nii on, on selles, et CPU-d on skalaararhitektuuriga - disainitud ühe instruktsiooni haldamiseks korraga ühel ajahetkel. Kuigi paljud CPU-d on saanud endale superskalaar arhitektuuri ehk neile on antud lisatuumasid, et oleks võimalik täita rohkem instruktsioone korraga [12], ei ole tulemused võrreldavad GPU-dega.

GPU-d põhinevad SIMD arhitektuuril (*Single Instruction Multiple Data*), mis võimaldavad neil sama tüüpi operatsioone täita samaaegselt [13]. Kuna pildianalüüs on hästi paralleliseeritav protsess, millega GPU-d väga hästi hakkama saavad, on loogiline ka neid võimalusel pildianalüüsi kaasata [3].

2. Projekti lahendus

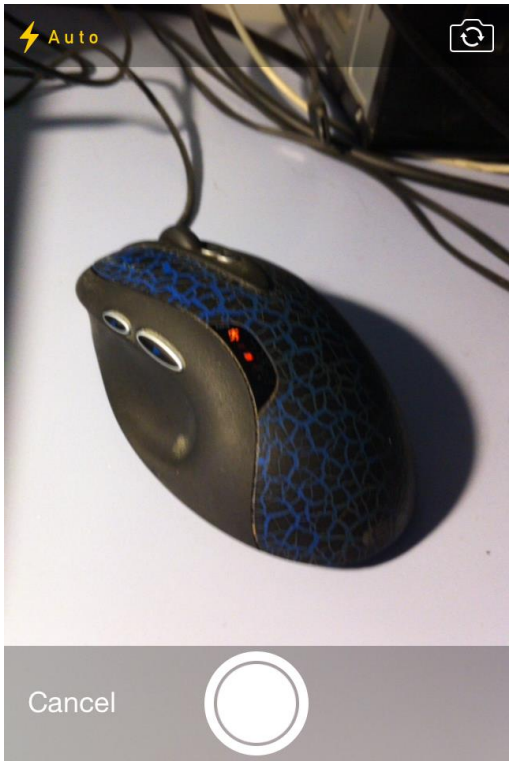
Käesolev projekt koosneb kolmest osast: mobiilirakendusest, serveriliidesest ja tehisnärvivõrgust. Mobiilirakendus ja serveriliides loodi spetsiaalselt selle töö tarbeks, tehisnärvivõrk, nagu eelnevalt ka juba öeldud, oli eelnevalt treenitud.

Mobiilis oleva rakenduse abil on võimalik kasutada mobiilikaamerat ning tehtud pilt siis kindlale aadressile saata. Selle aadressi taga on server, millel on käima pandud serveriliides, mis sissetulevad päringud vastu võtab. Kui päring on kohal, loetakse sealt pildiinfo ning antakse edasi võrgule. Võrk saab seejärel analüüsima hakata ning väljastab serveriliidesele oma tulemuse. Serveriliides konverteerib saadud tulemused inimesele loetavateks protsentväärtusteks, paneb saadud tulemused kokku eelnevalt defineeritud klasside listiga ning saadab vastuse päringuna tagasi mobiilile. Mobiilis võetakse see vastu ning kuvatakse ekraanile.

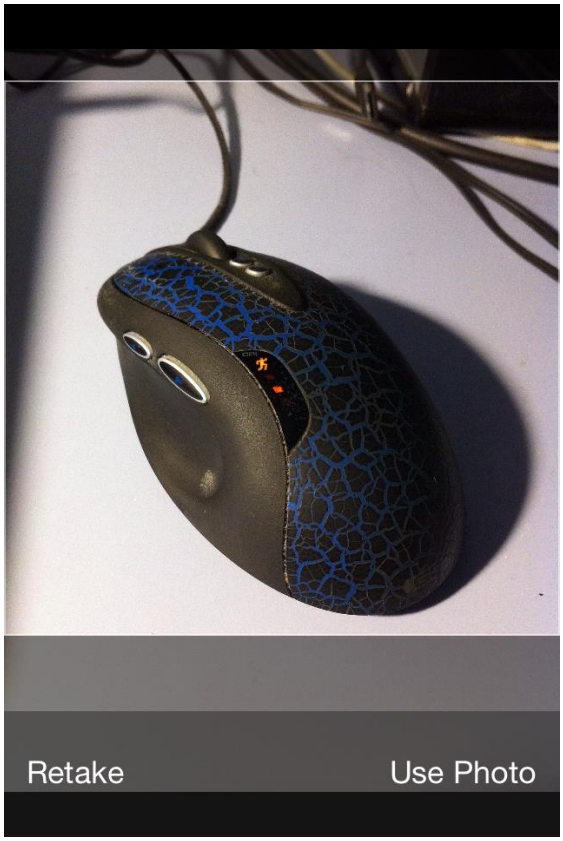
2.1 Mobiilirakendus

Mobiilirakendus on loodud Objective-C's iOS platvormile. Programm on ehitatud ühe näidiskoodi peale, mis demonstreeris *UIImagePickerController* klassi kasutamist. Näidis kood on loodud Rafael Garcia poolt [8]. Töö raames muudeti ja täiustati graafilist liidest, lisati HTTP päringute funktsioon, realiseeriti stringitöötlus ning kasutati näidiskoodis väljatoodud klassimeetodeid piltide tegemiseks ning loodi meetodid piltide saatmiseks HTTP päringutega. Programm toetab ennekõike iPhone mobiiltelefone. iPadi ja iPod Touchi tuge ei ole testitud.

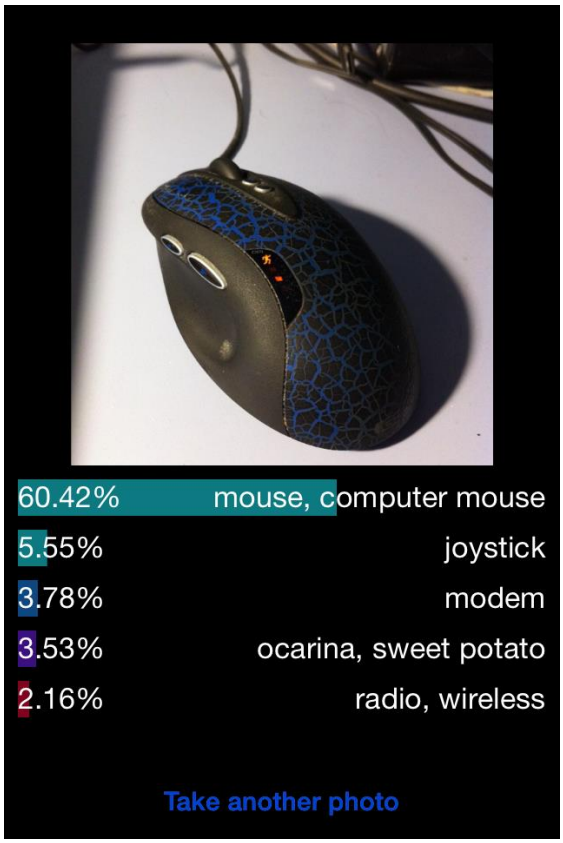
Oma olemuselt on rakendus väga lihtne ning intuitiivne. Rakenduse käivitamisel lülitutakse kohe pildistamisrežiimi, et soovitud objektist pilti teha.



Joonis 8. Pildistamisrežiim



Joonis 9. Pärast pildistamise kuvatakse selline ekraan. Saab kas uuesti pilti teha vajutades *Retake* või saata see foto vajutades *Use Photo*.



Joonis 10. Pilt on tehisnärvivõrgu poolt analüüsitud ning vastus mobiiltelefoni saadetud.

Soovi korral võib ka suumida ning vähese valguse korral lülitatakse automaatselt sisse ka välk. Kui pilt tehtud, kuvatakse see ekraanile ning pakutakse varianti, kas uus pilt teha (*Retake*) või kasutada seda, mis ees on (*Use Photo*). Soovi korral saab ka veel selles olekus pilti suumida, kui soovitakse valida mingit konkreetset objekti pildilt.

Kui pildiga ollakse rahul, siis võib vajutada *Use Photo* ning pilt laetakse serverisse üles. Vastuse tuleku aeg sõltub ühenduse ja pildianalüüsi kiirusest. Kui pilt on üles laetud, jääb programm ootele, millest ka ekraanil teada antakse. Kui vastus serverist on käes, siis töödeldakse tulnud vastust ekraanil kujutamiseks sobivaks. Viimase sammuna kuvataksegi vastus ekraanile koos võetud pildiga. Vastusena näidatakse viis kõige tõenäolisemat pakkumist tehisnärvivõrgu poolt koos tõenäosusprotsentide, klassi nime ning vertikaalsete tulpadega. Pärast seda on võimalus uus pilt teha vajutades nupule *Take another photo*.

Programmis kasutatav API toetab ka pildialbumist pildivalimist, et oleks võimalik varem tehtud pilte saata tehisnärvivõrgule, kuid antud töös ei peetud selle realiseerimist vajalikuks. Kui kunagi peaks aga rakendust edasi arendama iOS platvormil, siis on selline variant igatahes olemas.

2.2 Serverirakendus

Serveriks on riistvaraliselt tavaline arvuti, mis on kohendatud serverilaadsete ülesannete täitmiseks. Serveris on Linuxi operatsioonisüsteem, mille peal jookseb programmeerimiskeeles Python loodud liides. Läbi selle liidese toimub suhtlus tehisnärvivõrgu ja mobiilirakenduse vahel. Pythoni liides on loodud eelnevalt valminud HTTP näidiskoodi peale [18]. Liidesesse on lisatud meetodid tehisnärvivõrguga suhtlemiseks, piltide vastuvõtmiseks, tehisnärvivõrgult saadud vastuste tõlgendamiseks ning korrektses formaadis vastuse ehitamiseks.

Liides pannakse käima nii nagu iga teinegi Pythonis tehtud programm. Käivitamise hetkel pannakse käima ka tehisnärvivõrk ning jäädakse kuulama eelnevalt määratud pordi taha. Mobiilirakendusest saab pilti saata serverisse ainult siis, kui see saadakse õigele pordile ja kui pordiliiklus on lubatud.

Server on enamuse ajast ootel ega tee midagi erilist. Kui server päringuga pildi saab, teostatakse teatud kontrollid. Esimesena kontrollitakse, kas üldse mingi fail saadi ja et ei oleks midagi muud saabunud. Teisena kontrollitakse, kas failiga tulid ka mingid andmed kaasa. Seejärel salvestatakse pilt ajutiselt kõvakettale ning antakse tehisnärvivõrgule. Tehisnärvivõrk proovib etteantud failist pilti välja lugeda. Kui see ebaõnnestub, siis antakse teada, et saadud failist polnud võimalik lugeda piltidele omast infot. Kui aga kõik klapib ning tehisnärvivõrk saab etteantud failist pildiinfo kätte, teostatakse analüüs ning väljastatakse tõenäosused 1000-le erinevale klassile, millest selekteeritakse 5 suurimat väärtust. Need väärtused on alati kahedimensioonilises massiivis iga klassi jaoks kindlatel indeksitel. Kahedimensiooniline on massiiv seepärast, et vastusena väljastatakse massiiv, milles sisalduvad massiivid, milles omakorda on tõenäosuseväärtused 1000-le klassile. Viimaseid massiive võib olla rohkem kui üks juhul, kui analüüsiti mitut pilti järjest. Seda juhtumit aga siin projektis ei käsitletud ning vastused küsitakse alati esimese massiivi indeksist 0. Vastus pannakse kokku ühte kindla formaadiga stringi ning saadetakse tagasi aadressile, kust pilt algselt pärines.

Kui serveris on olemas ka graafikaprotsessorid, siis on võimalik projektis kasutatav tehisnärvivõrk seadistada neid kasutama tavaliste protsessorite asemel.

2.3 Tehisnärvivõrk

Tehisnärvivõrguks on Caffe tööriistaga loodud võrk. Arhitektuuriks on Alex Križevsky loodud tehisnärvivõrk, mis võitis ImageNet võistluse 2012 aastal. Kasutatud tehisnärvivõrgust on lähemalt räägitud peatükis 2.3.

3. Tulemuste analüüs

Võib öelda, et projekti lõppeesmärk sai täidetud ning teostatud lahendus töötab tõrgeteta. Paraku käivad kaasas aga teatud mööndused.

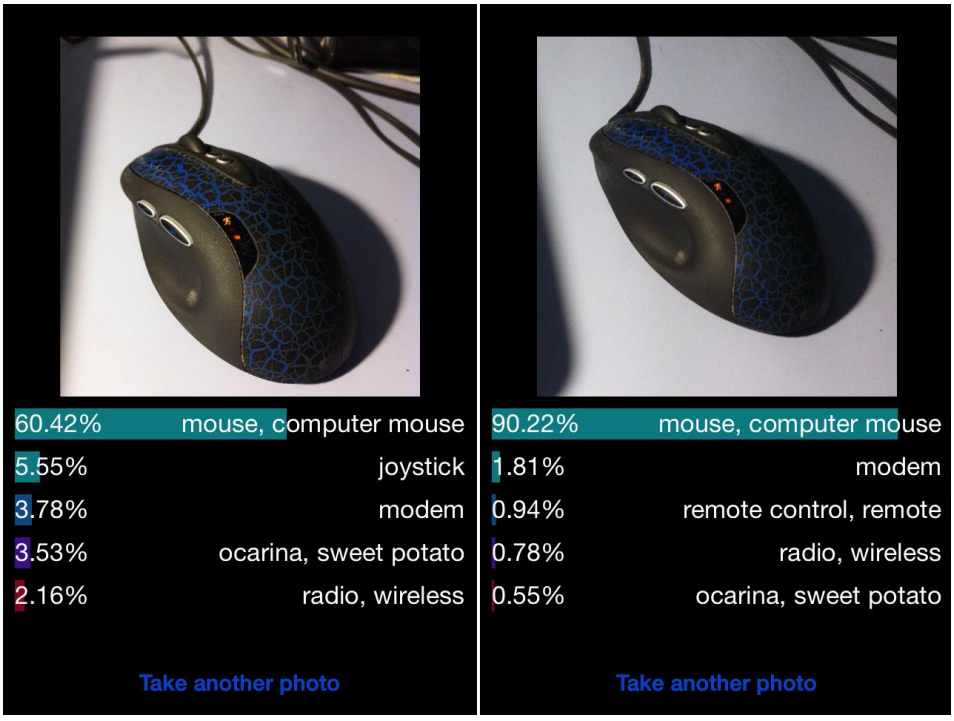
3.1 Vastuse saamine

Lahendus eeldab mingisuguse internetiühenduse olemasolu. Mobiilirakendus ise pilte ei analüüsi, seega kui internetiühendus puudub või on liiga kehva kvaliteediga, siis ei saa ka mingit mõistlikku vastust, sest ei ole võimalik saada ühendust tehisnärvivõrguga. Kuna aga mobiilirakendus on võrdlemisi töökindlaks tehtud, siis sellega seoses muid erilisi piiranguid ei ole.

Kindlasti peavad lisaks internetiühendusele töötama ka server, server peab olema ligipääsetav ning tehisnärvivõrk peab olema valmis pilte aktsepteerima. Kui mingi nendest lülidest ei tööta korralikult, siis ei pruugi ka vastust saada

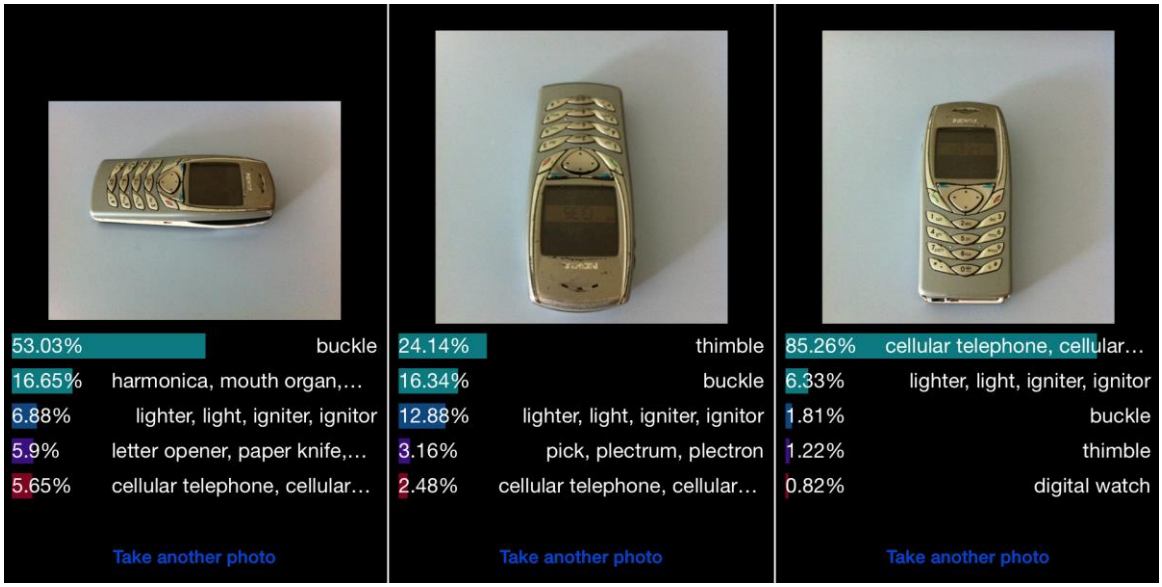
3.2 Vastuse täpsus

Kui internetiühendus on olemas ning kontakt tehisnärvivõrguga saavutatud, siis võib kindlalt öelda, et vastus ka tuleb (testimise käigus pole veel olnud situatsiooni, kus kõikide ühenduste olemasolul pole veel vastust tulnud). Tehisnärvivõrgu vastus sõltub aga väga palju juba pildist endast. Arvesse lähevad siin pildi suurus, suuruste suhe, valgus, nurk, objektide arv pildil. Lisaks sellele sõltub vastus ka pildistatud objektide omadustest, näiteks kuju ja orientatsioon.



Joonis 11. Pildistatud on täpselt sama eset. Siit võrdlusest on näha, kuidas pildistamisnurka ainult natukene muutes tõusis tõenäosusprotsent arvutihiire suhtes peaaegu 30%.

Lisaks sõltuvad tulemused ka sellest, mis pidi on mingit objekti pildistatud. Treenimise ajal pööratakse vahel osad pildid teistpidi horisontaalteljel. Seda tehakse selleks, kuna maailmas me ei vaatle kõik objekte alati ühte pidi. Näiteks kahesuunalisel tänaval liiguvad autod mõlemat pidi. Merel laevaga sõites mööduvad teised laevad nii ühelt kui ka teiselt poolt. Paraku aga sellega asi piirdubki – pilte ei pöörata näiteks vertikaalteljel. Üldiselt vaatleme objekte õiget pidi ehk teisisõnu, vaatleme neid nii nagu gravitatsioon lubab. Pilte autodest pole mõtet peegeldada vertikaalteljel, sest autod ei sõida taevas. Samuti hoiame peaaegu kõiki seadmeid alati ühte pidi – me ei keera neid pea peale. Seetõttu pole ka mõtet võrku treenida selliseid objekte tuvastama. On aga olukordi, kus seadmed paistavad ka tavaolukordades tagurpidi. Näiteks kui mobiiltelefon on laua peal. Sõltuvalt sellest, kuidas see sinna pandud on ning vaatleva inimese vaatenurgast, võib mobiiltelefon tunduda kas õigetpidi või tagurpidi.

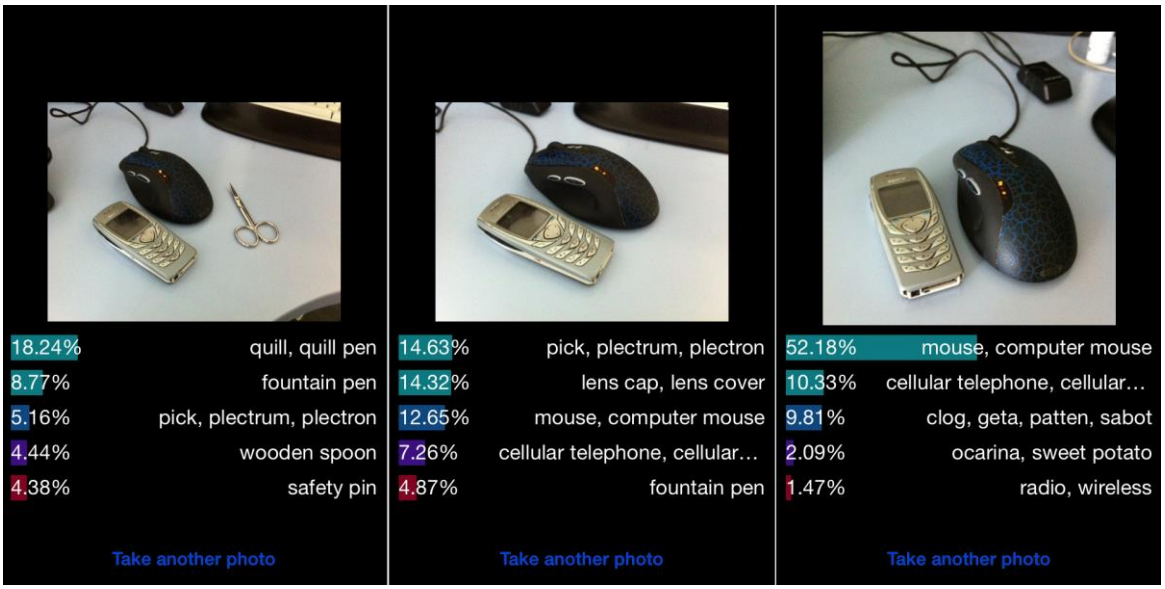


Joonis 12. Vasakpoolsel pildil on mobiiltelefon n-ö küljega. Keskmisel pildil on mobiiltelefon tagurpidi. Parempoolsel pildil on mobiiltelefon õigetpidi. Kuigi tegemist sama objektiga, on vastused vägagi erinevad.

Jooniselt 12 on väga hästi näha, mis vahed tekivad sõltuvalt sellest, mis asetusega objekt pildile on jäänud. Kuigi keskmisel pildil, kus mobiiltelefon on tagurpidi, on võrk suutnud mobiiltelefoni ka tuvastada ning TOP-5 kategoorias loetaks see õigeks, on 2,48% tulemus siiski väga väike võrreldes parempoolse pildiga, kus mobiiltelefon on õigetpidi ning tõenäosusprotsent 85,26%. Vasakpoolsel pildil sarnaneb mobiiltelefon võrgu arvates aga väga palju pandlaga (*buckle*). Inimese jaoks ei ole reeglina probleem objekte tuvastada ükskõik millises asendis, kuid tehisnärvivõrgud seda ei suuda ning neid tuleb selleks vastavalt treenida.

Vastuse täpsus sõltub ka veel näiteks sellest, mitu objekti pildil on ning kas tehisnärvivõrk on treenitud ka neid klassifitseerima. Joonisel 13 on vasakpoolsel pildi demonstreeritud, kui pildi peal on fookuses kolm objekti, millest ühte ei ole võrku treenitud klassifitseerima (kääre). Kuna pilt on tehtud ka teatud nurga alt (ei ole otse), siis lõpptulemusena ei ole võrk mitte ühtegi vastust õigesti saanud. Keskmisel pildil on käärid eemaldatud, kuid pildistamisnurk on rohkem külje peal. TOP-5 skaalas loetaks vastus õigeks, kuna mõlemad fookuses olevad objektid tundis võrk ära (mobiiltelefon, arvutihiir), kuid esimesele kohale ja isegi teisele kohale on pakutud objekte, mida pildil tegelikult ei esine. Segadusse võivad ajada siin ka pildi ülemises servas nähtavad mustad objektid (mis tegelikult on heliregu-

laator ja klaviatuuri käetugi). Parempoolsel pildil on tehtud pilt nii-öelda otsevaates, kus fookuses on mobiiltelefon, arvutihiir ning ka heliregulaator (seda heliregulaatorit ei ole võrk treenitud klassifitseerima). Tulemused on sellistes tingimustes kohe paremad, kus arvutihiir on esimesel kohal 52,18% ning mobiiltelefon teisel kohal 10,33%.



Joonis 13. Parempoolsel pildil on fookuses kolm objekti, millest ühte ei oska võrk tuvastada (käärid). Keskmisel pildil on käärid eemaldatud, kuid pilt on tehtud rohkem küljepealt. Parempoolsel pildil on pilt tehtud nii-öelda otsevaates. Siit on ka näha, et kõrvaliste objektide olemasolu ei takista hea pildi korral fookuses olevate objektide klassifitseerimist.

3.3 Ühenduse kiirused

Kui mobiiltelefon on ühendatud WiFi võrku, siis reeglina infovahetuses serveri ja mobiiltelefoni vahel probleeme ei esine. Mobiiltelefonid ei ole aga alati WiFi võrgus, vaid kasutavad vahel ka mobiilset internetti. Viimane kehtib eriti juhul, kui mobiiltelefoniga ollakse kusagil väljas, näiteks metsas, ning kuna terve projekti üks eesmärkidest oligi võimaldada objektide pildistamist ükskõik kus, siis on seda ka teatud määral arvesse võetud.

Mobiilrakendusega tehtava pildi suurus on 640 x 640 pikselit, mis võimaldab palju väiksemaid pildifaili suurusi. Pildifaili suurused jäävad keskmiselt vahemikku 100 - 150 KB. Ka kõige minimaalsema 3G ühenduse korral saab pildi edukalt saadetud ning 3G on peaaegu igal pool Eestis saadaval. Kehvemate ühendustüüpide korral ei pruugi programm enam korralikult töötada.

Andmeside kvaliteedi kõikumisi silmas pidades saadetakse serverist vastusena ainult üks string. String sisaldab viie tõenäoseima klassi nimetusi ning nende tõenäosusi protsentväärtustena. Stringi sisse on pandud spetsiaalsed tähemärgid, mille abil oskab programm saadud vastuse õigesti lahti harutada ning siis kogu info õigetesse lahtritesse panna ekraanile.

4. Sarnased lahendused

4.1 Microsofti lahendus

2014. aastal iga-aastasel Faculty Summit üritusel esitles Microsoft oma arengut tehisintellekti vallas. Nende lahenduses on 2 miljardit ühendust (võrdluseks selles töös kasutatud lahenduses on 60 miljonit) ning on väidetavalt kaks korda täpsem ning 50 korda kiirem kui teised lahendused. Microsofti treenitud süsteem oli õpetatud tuvastama koeratõuge. Treenimiseks kasutati väidetavalt materjali sellistelt lehekülgedelt nagu Flickr, kuid arvestades koeratõugude rohkust ning heaks treenimiseks vajaminevat materjali, on alust arvata, et kasutati ka ImageNet andmebaasi, eriti kuna ImageNetil on olemas suur hulk erinevaid pilte koertest. Töötas nende lahendus aga samal põhimõttel nagu käesolevas projektis: soovitud objektist tehakse mobiiltelefoniga pilt, viimane saadetakse tehisnärvivõrgule analüüsimiseks ning vastus väljastatakse lõpuks mobiiltelefonis. Demonstratsiooniks oli kohale toodud paar koera, keda publik ees tuvastati.

Microsofti üheks eesmärgiks oli uurida, et kas ühenduste suurendamine suurendab ka määramistäpsust. [5]

4.2 TapTapSee

Aastal 2014 väljastas Image Searcher, Inc oma programmi mobiiltelefonidele nimega TapTapSee. Programm on peamiselt suunatud vaegnägijatele, kellel on probleeme normaalse nägemisega. Antud programm võimaldab neil teha soovitud objektist pilti, ning pärast analüüsimist annab programm teada, kas häälega või tekstiga, et millega on tegu. Image Searcher väidab, et nende programm kasutab piltide analüüsimiseks pildituvastusalgoritme, kuid ühtlasi ka inimeste abi. [6][7]

5. Projekti rakendatavus

Projekti idee seisneb selles, et mobiilkaamerat kasutades oleks võimalik teha soovitud esemest pilt ning siis lasta see tehisnärvivõrgul ära tuvastada. Praegu pole taolised mobiilirakendused, mis käesoleva projekti raames tehti, väga laialt levinud. Teada on küll Microsofti tehtud lahendus, mis oskab pildi pealt koera liiki tuvastada ning ka TapTapSee, mis kasutab inimeste abi. Mõlemast on täpsemalt juttu peatükis 4.

Üks võimalik kasutusala projektis tehtud rakendusele on pildipõhine otsing – näita mulle poode, kus selliseid tooteid müüakse. Teoorias kätkeks selline lahendus endas tehisnärvivõrku, mis oskaks pildistatud objekti klassifitseerida, ning eraldi andmebaasi asjade kohta, mida kuskil poodides müüakse. Kui potentsiaalne klient leiab endale meelepärase eseme, kuid soovib teada, kas sama eset või sarnaseid esemeid kuskil mujal veel müüakse, siis saaks kasutada kirjeldatud lahendust.

Teine kasutusviis oleks pimedate inimeste abistamine objektide äratundmisel – telefon ütleb häälega, mis objekt pildi peal on. Tehisnärvivõrkude ja käesoleva programmi idee baasil saaks luua lahendusi, mis aitaksid näiteks vaegnägijaid. Ideelt on tegemist üpriski sarnase lahendusega nagu seda on TapTapSee, kuid selle vahega, et enam ei kasutata inimesi, et kirjeldada pildil toimuvat, vaid ainult tehisnärvivõrke. Selle puuduseks on küll asjaolu, et tehisnärvivõrgud ei suuda kirjeldada pildil toimuvat, vaid ainult klassifitseerida pildil olevaid objekte. TapTapSee üks eeliseid on see, et vastusena tulev tekst tihtipeale ka kirjeldab lühidalt, mis pildil toimub, kuid see eest ei oska klassifitseerida objekte nii täpselt, kui seda suudaks tehisnärvivõrk. Ühtlasi eeldab TapTapSee lahendus mingisugust inimtööjõudu. Rakendades ainult tehisnärvivõrke, see probleem kaoks.

5.1 Meditsiin

Meditsiin on lai valdkond ning seal tegelikult juba kasutataksegi tehisnärvivõrke. Kuigi nende panus meditsiini on praegu võrdlemisi väike, ei saa seda siiski mainimata jätta, eriti kuna tehisnärvivõrkude potentsiaal meditsiinis on väga suur. Usutakse, et lähiaastatel aitavad tehisnärvivõrgud lahendada olulise hulgal meditsiiniga seotud probleeme, nagu näiteks diagnoosid, biokeemilised analüüsid ja ravimitootmine [14].

Üks spetsiifilisem ala, kus panustatakse väga palju just taolistele tehisnärvivõrkudele, mida siin projektis kasutati, on rinnavähikoest mitoosiliste protsesside otsimine. Selleks puhuks on spetsiaalselt treenitud üks võrk, mis oskab neid protsesse tuvastada piltide pealt, mille peal on kudede läbilõiked. Tegemist on üpris keerulise ülesandega, millega treenimata inimesed hakkama ei saaks ning isegi inimesed, kes on sellega palju tegelenud, ei suuda alati kindlalt öelda, kas tegu on mitoosiliste protsessidega või mitte. Korralikult treenitud tehisnärvivõrgud suudavad aga selliseid ülesandeid täita ning seda palju kiiremini, hoides kokku aega ja raha [15].

Antud projekti oleks võimalik rakendada näiteks nahahaiguste tuvastamiseks, mida iga inimene saaks teha oma kodus. Selle jaoks saaks treenida spetsiaalse võrgu, mis oskaks tuvastada ja klassifitseerida erinevaid nahal esinevaid laike. Idee poolest võiks olla nii, et inimene teeb mingist naha osast pilti ning võrk saadab vastuse, et millega tegu võib olla. Koos vastusega võivad tulla vastavalt tulemustele ka instruktsioonid, et kuidas teha kindlaks, kas tegemist on ikkagi pakutud diagnoosiga või mitte ning edasised juhised arstiga konsulteerimiseks.

Need on aga ainult üksikud näited sellest, mis kõik oleks võimalik antud lahendust kasutades. Kindlasti leiaks võimalusi ning kasutusalasid veel rohkem.

6. Edasiarenduse võimalused

6.1 Mitme tehisnärvivõrgu variandid

Üks variant, kuidas olemasolevat lahendust täiendada saaks, oleks mitme erineva tehisnärvivõrgu ülesseadmine ning mobiilirakenduses vastavate valikute loomine nende kasutamiseks. Praegune tehisnärvivõrk on eelnevalt treenitud klassifitseerima kindlat 1000 erinevat objekti ImageNet andmebaasist. Võimalusi on aga palju rohkem ning eeltreenituid tehisnärvivõrke on treenitud ka muudeks spetsiifilisemateks ülesanneteks. Idee olekski selles, et mobiilirakenduses saaks enne või pärast pilditegemist valida, et millisele tehisnärvivõrgule pilti saata soovitakse. Variant võiks olla ka mitmele saata, et oleks parem tulemusi võrrelda.

6.2 Realiseerida tehisnärvivõrk mobiilseadmes

On olemas ka kohendatud tehisnärvivõrgud, mis töötavad otse mobiilseadmel, näiteks mobiiltelefonis. Sellised võrgud on kõvasti lihtsustatud ja nende tulemused ei ole päris võrreldavad tavalistel arvuti- või serveriarhitektuuridel töötavate võrkudega, kuid arvestades süsteemi lihtsust, on tegemist tõsiseltvõetava ideega. Sellisel juhul ei pea eraldi töötama kuskil mingi server, mille peal võrk töötab. Pole vaja ka ühenduda kuhugi, vaid kogu töö tehakse mobiilseadmes ära. [9]

6.3 Treenida uus närvivõrk seenteliikide klassifitseerimiseks

Esialgne plaan, mida kahjuks teha ei õnnestunud, oli treenida uus tehisnärvivõrk seente liikide klassifitseerimiseks. Eduka treeningu korral oleks võimalik uus võrk panna soovitud serverisse ning teha mobiiliprogrammis vastavad muudatused, et uut võrku oleks võimalik kasutada. Töötava lahenduse korral oleks võimalik metsas seenel käies tuvastada mobiiltelefoniga, et mis seentega tegu on. See töötaks siiski eeldusel, et on olemas mingisugune internetiühendus.

6.4 Mobiilirakenduse koodi optimiseerimine iOS-ile

Praegu kasutatav lähtekood on modifitseeritud ja täiendatud variant Rafael Garcia tehtud demonstratsioonist, kuidas kasutada *UIImagePickerController* klassi. Kõige õigem oleks teha uus projekt ning järgides praegu kasutuses oleva lähtekoodi struktuuri, disainida uus lähtekood, mis oleks puhtam, loogilisem ning optimaalsem.

6.5 Rakenduse laiendamine Android seadmetele

Projekti tegemise käigus otsustati teha mobiilirakendus iOS platvormile. Testimiseks piisas sellest täiesti, kuid edasisteks uuringuteks ja tuleviku laiatarbele mõeldes tasuks kaaluda Androidi versiooni tegemist.

7. Kokkuvõte

Tehisnärvivõrgud imiteerivad neuronite vahelisi ühendusi, mis esinevad inimeste ajus. Võrke on võimalik treenida erinevateks ülesanneteks, näiteks piltidelt objektide klassifitseerimiseks, jäljendades sellega nägemismeelt. Kui muidu toimuvad treenimised ja testimised ainult arvutites koos eelnevalt tehtud ja töödeldud piltidega, siis selle töö käigus loodi lahendus, mis aitaks vähemalt testimise tuua sammukese lähemale sellele, kuidas inimesed igapäevaselt maailma näevad. Selle lahendusega saaks testida, kuidas treenitud võrk saaks hakkama maailmas, mida inimesed iga päev tajuvad. Selle tulemusel oleks võimalik välja töötada palju erinevaid lahendusi, mis võiksid inimestele ka kuidagi kasulikud olla, näiteks aidata vaegnägijaid objektide tuvastamisel. Ühtlasi saaks minna sammu edasi Microsofti lahendusest, mis klassifitseerib koeratõuge, ning miks mitte disainida lahendus, millega oleks võimalik kõiki metsloomi klassifitseerida. Nii saaks näiteks tuvastada, et millise linnuga on tegemist, mis kala õnge otsa jäi või mis võõrloomaga on tegemist. Inimene pole võib-olla taolist looma varem näinud, kuna näiteks tegemist võib olla võõrliigiga, kuid võrke on võimalik varakult treenida taolisteks juhtumiteks ning ühegi looma liik ei jääks sellisel juhul mõistatuseks, kui mobiiltelefon olemas on.

Kaugemale tulevikule mõeldes, selles töös tehtud samm võiks kujutada endast laborikatsetest edasiliikumist ning reaaleluliste katsete algust. Kuna tehisnärvivõrkude puhul on sisuliselt tegemist tehisintellektile sarnaste süsteemidega, on huvitav teada, kuidas need hakkama saavad laboritest väljaspool. Kunagi oleks võimalik ehitada robot, mis näiteks kannab erinevaid treenituid tehisnärvivõrke endaga kaasas. Kõik võrgud võivad olla treenitud erinevateks ülesanneteks, aga miks mitte ka üksteist täiendavateks. Selline võrgustik oleks juba mitu sammu lähemal inimajule ning selline robot võiks teoreetiliselt juba päris hästi hakkama saada inimestega samas ruumis.

8. Tsiteeritud teosed

[1] ImageNet, all results tables, http://image-net.org/challenges/LSVRC/2012/results.html [01.04.2015]

[2] History: The 1940's to the 1970's, Neural Networks, http://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html [01.04.2015]

[3] nVidia, „What is GPU accelerated computing?" http://www.nvidia.com/object/what-is-gpu-computing.html [02.04.2015]

[4] ImageNet, http://image-net.org/challenges/LSVRC/2012/index#introduction [09.04.2015]

[5] Chansanchai, Athima, „Microsoft Research shows off advances in artificial intelligence with Project Adam", http://blogs.microsoft.com/next/2014/07/14/microsoft-research-shows-advances-artificial-intelligence-project-adam/ [12.04.2015]

[6] „Description of TapTapSee", http://www.wonderbaby.org/articles/taptapsee-app-review [10.04.2015]

[7] TapTapSee, https://twitter.com/taptapsee/status/342033507587088384 [10.04.2015]

[8] Garcia, Rafael, „Build a Simple Camera App Using UIImagePickerController", http://www.appcoda.com/ios-programming-camera-iphone-app [04.04.2015]

[9] „How to run a state of the art image classifier on a iPhone", http://libccv.org/post/how-to-run-a-state-of-the-art-image-classifier-on-a-iphone/ [04.04.2015]

[10] Hintoni, Geoffrey, Coursera.org kursus loeng 5 slaid nr 14, https://www.coursera.org/course/neuralnets [14.05.2015]

[11] Wu, Ren, et al. "Deep Image: Scaling up Image Recognition." *arXiv preprint arXiv:1501.02876* (2015), lk 6, tabel 3, http://arxiv.org/pdf/1501.02876v2.pdf [11.05.2015]

[12] Faraz, Ahmed, „Superscalar architecture", presentatsioon, http://www.math.nsysu.edu.tw/~lam/MPI/lecture/SUPERSCALAR_ARCHITECTURE.ppt [12.05.2015]

[13] Rosenberg, Ofer, OpenCL Dev. Team, „Introduction to GPU Architecture", http://haifux.org/lectures/267/Introduction-to-GPUs.pdf [12.05.2015]

[14] Siganos, Dimitrios, „Neural Networks in Medicine", http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/ds12/article2.html [12.05.2015]

[15] Cireşan, Dan C., et al. "Mitosis detection in breast cancer histology images with deep neural networks." *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013*. Springer Berlin Heidelberg, 2013. 411-418. http://people.idsia.ch/~ciresan/data/miccai2013.pdf [12.05.2015]

[16] Caffe, „Brewing ImageNet", http://caffe.berkeleyvision.org/gathered/examples/imagenet.html [14.05.2015]

[17] Vanhoucke, Vincent, „Speech Recognition and Deep Learning“, http://googleresearch.blogspot.com/2012/08/speech-recognition-and-deep-learning.html [14.05.2015]

[18] Stackoverflow, „Simple Python server to save file“, http://stackoverflow.com/questions/13146064/simple-python-webserver-to-save-file [20.03.2015]

Lisad

I. Terminid

API Tegemist programmiliidesega, mille kaudu on võimalik olemasoleva programmiga suhelda.	API Application Programming Interface, used for developing extensions to already existing programs.
Python Üks paljudest programmeerimiskeeltest, sarnane Javale ja C++'ile, kuid süntaksilt lihtsam.	Python Programming language, similar to Java and C++, but with easier syntax.
Objective-C Programmeerimiskeel, mis on arendatud Apple-i poolt nende operatsioonisüsteemide OS X ja iOS jaoks.	Objective-C Programming language developed by Apple for their operating systems OS X and iOS.
3G Mobiilse ühenduse tüüp, tähistab kolmandat generatsiooni. On kiirem kui varasemad ühendused, kuid aeglasem kui 4G.	3G Mobile telecommunication technology. 3G is short for third generation. Is faster than older Technologies, but slower than 4G.
Softmax Funktsiooni nimetus, mida kasutatakse antud tehisnärvivõrgu viimases kihis.	Softmax Name of a function used in the last layer of the artificial neural network used in this project.
Max-pooling Protsessi nimetus, mida kasutatakse max-pooling kihtides.	Max-pooling Name of a process used in max-pooling layers.
HTTP Tegemist protokolliga, mille abil toimub suur osa interneti andmevahetusest	HTTP Hypertext Transfer Protocol. Widely used protocol in internet traffic.

II. Lähtekoodid

- Mobiilirakenduse lähtekood kui ka serveriliides on olemas lisana, aga ka aadressil http://kodu.ut.ee/~coldy/Bach/.
- Tehisnärvivõrku on võimalik hankida järgnevalt aadressilt https://github.com/BVLC/caffe. Nimekiri kõikidest vajaminevatest programmidest ning juhendid leiab siit http://caffe.berkeleyvision.org/. Vägagi tõenäoliselt tuleb serveriliideses muuta seda osa, mis paneb tehisnärvivõrgu käima, kuna neid Caffe mudeleid uuendatakse.

III. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina **Rasmus Vahtra** (sünnikuupäev: 11.04.1992)
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose **Mobiilirakenduse loomine tehisnärvivõrgule**,
(*lõputöö pealkiri*)

mille juhendaja on Tambet Matiisen,
(*juhendaja nimi*)

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **14.05.2015**