

TARTU ÜLIKOOL

Arvutiteaduse instituut

Informaatika õppekava

Andry Nõgols

Mobiilirakendus seente tuvastamiseks

Bakalaureusetöö (9 EAP)

Juhendaja: Sven Aller

Tartu 2019

Mobiilirakendus seente tuvastamiseks

Lühikokkuvõte:

Bakalaureusetöö eesmärk on luua mobiilirakendus Androidile, mis suudab pildilt tuvastada Eesti metsades kasvavaid seeni ning tõsta seeläbi inimeste seeneteadlikkust.

Töös kirjeldatakse seente määramisega seotud probleeme, tuues välja rakenduse vajalikkus. Samuti tutvustatakse probleemi lahendamiseks kasutatavate tehisnärvivõrkude tööpõhimõtet ja uuritakse sarnasel põhimõttel töötavaid olemasolevaid lahendusi. Töös kirjeldatakse nii tehisnärvivõrgu mudeli kui ka mobiilirakenduse loomise protsesse ning selleks kasutatud tehnoloogiliste vahendeid ja tööriistu. Töö lõpus analüüsitakse saadud tulemusi ning arutletakse rakenduse edasiarendamise võimaluste üle.

Võtmesõnad:

Mobiilirakendus, tehisnärvivõrk, pildituvastus, Android

CERCS:

P175 (Informaatika, süsteemiteooria)

Mobile application for mushroom identification

Abstract:

The aim of this bachelor thesis is to develop an Android mobile application, which identifies mushrooms in Estonian forests based on its picture and thus improves people's knowledge of mushrooms.

In this thesis problems related to mushroom identification and necessity for an application are described. Also the working principles of artificial neural networks that are used for image recognition are explained and some already existing solutions described. Furthermore, the processes of developing a mobile application and an artificial neural network model are described, as well as used technologies and tools for both of them. In the end of the thesis the results are analyzed and further development opportunities are discussed.

Keywords:

Mobile application, artificial neural network, image recognition, Android

CERCS:

P175 (Informatics, systems theory)

Sisukord

Sissejuhatus.....	5
1. Seente liigitamine.....	6
1.1. Mobiilirakendused seente tuvastamiseks	7
1.1.1. Mushroom Identifier	7
1.1.2. Rozpoznání hub	8
1.1.3. Mushrooms app.....	8
2. Tehisnärvivõrgud	9
3. Rakenduse kirjeldus	13
3.1. Tehisnärvivõrgu mudel	13
3.1.1. Andmed.....	13
3.1.2. Mudeli koostamine.....	14
3.1.3. Lõpliku mudeli kirjeldus.....	16
3.2. Mobiilirakendus	18
3.2.1. Rakenduse ülesehitus	19
3.2.2. Rakenduse kasutamine.....	20
3.3. Kasutatud tehnoloogiad.....	21
3.3.1. TensorFlow	22
3.3.2. Keras	22
3.3.3. OpenCV	23
3.3.4. Android Studio.....	23
4. Tulemused.....	24
4.1. Testimine	24
4.2. Hinnang.....	25
4.3. Edasiarendamise võimalused	26
Kokkuvõte.....	28
Kasutatud kirjandus	29
Lisad.....	32
1. Andmete rohkuse jaotus üle liikide.....	32
2. Katsetatud mudelite arhitektuurid.....	33
3. Segadusmaatriksi analüüs testhulgal.....	34
Lihthitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks.....	35

Sissejuhatus

Eesti riik on suurelt jaolt tuntud oma metsarohkuse poolest ning metsatööstus on üks Eesti majanduse aluseid. Samuti on metsasaadused, sealhulgas seened, paljudele Eesti elanikele sümboolse tähendusega. Seenel käiakse nii hobi korras kui ka lihtsalt toidulaua rikastamise eesmärgil, kuid paljude inimeste jaoks on seente müügist saadud tulu ka elulise tähtsusega. Sellest tulenevalt on oluline, et seenel käimisega tegelevad inimesed oleksid teadlikud korjatavatest seeneliikidest, nende välimusest ning mürgisusest. Eesti elanike seeneteadlikkuse tõstmine ning eksperimentaalne huvi pildituvastuse vastu on ajendiks, et luua infotehnoloogiline lahendus, mis lihtsustaks erinevate Eesti metsas kasvavate seente identifitseerimist.

Käesoleva bakalaureusetöö eesmärk on luua Androidi rakendus, mis võimaldaks tehismärgivõrkude abil loodud mudeli põhjal tuvastada pildil oleva Eesti metsades kasvava seene liiginimetuse. Töö sisaldab endas vastava mudeli jaoks vajalike andmete (seente pildid, liiginimetused) kogumist, mudeli loomist ning treenimist ja mudeli põhjal saadud tulemuse kasutajale informatiivsel kujul esitamist.

Töö koosneb neljast peatükist. Töö esimeses peatükis antakse ülevaade seente ning nende liigitamise kohta, tuuakse välja probleemi tähtsus ning kirjeldatakse juba olemasolevaid seente liigitamiseks loodud lahendusi. Töö teises peatükis seletatakse seente tuvastamiseks kasutatavate tehismärgivõrkude tööpõhimõtet. Töö kolmandas peatükis selgitatakse rakenduse ülesehitust ning kirjeldatakse täpsemalt kõiki tervikliku rakenduse alamosi. Lisaks tuuakse välja rakenduse loomisel tehtud tehnoloogilised valikud. Töö neljandas peatükis kirjeldatakse rakenduse testimist ning hinnatakse saadud tulemusi. Lisaks arutletakse rakenduse edasiarendamise võimaluste üle.

1. Seente liigitamine

Planeedil Maa arvatakse olevat ligikaudu 2,2 kuni 3,8 miljonit erinevat seeneliiki, millest 2018. aasta seisuga on teadlased ära kirjeldanud umbes 144000 [1]. Seente liigirikkus ja mitmekesisus on tohutu ning nii kuuluvad nende liikide hulka näiteks paljud seenelistele tuttavad kübarseened kui ka erinevad hallitused. Kõnekeeles kasutatakse mõistet “seen” meelevaldselt ning selle all mõeldakse enamasti selliseid seeni, mis moodustavad palja silmaga nähtavaid viljakehi [2]. Käesoleva töö raames on edaspidi seene all mõeldud samuti just selliseid seeni.

Võrdlemisi suuri silmaga nähtavaid viljakehi moodustavaid seeni on maailmas ligikaudu 14000 [3], millest mitmed tuhanded kasvavad ka Eestis [4]. Eesti metsades kasvavatest seentest on söödavaid umbes 400 liiki ning mürgised ligikaudu 200 liiki [5]. Mürkseened on liigiti vähemal või suuremal määral mürgised. Osade mürkseente puhul avaldub mürgisus ainult teatud tingimustel, näiteks seente tarbimine koos alkoholiga või nende söömine ilma eelneva kupatamiseta [6]. Mürkseene söömine võib põhjustada inimesel kerge või tõsise mürgistuse [7].

Sõltuvalt mürgistusest võib mürgise seene söömine kahjustada inimese kesknärvisüsteemi, põhjustada seedeekstrakti häireid või tekitada tõsiseid maksa ja/või neerukahjustusi. Viimase tagajärjeks on tihti inimese surm. Seenemürgistused on eriti ohtlikud, sest tõsisematel mürgistustel on tihti väga pikk peiteaeg (kaks-kolm päeva) ning mürgistuse avastamise hetkeks võib olla juba liiga hilja, kuna paljudele mürgistustele puudub vastumürk.

Seenel käies tuleb seega olla eriti ettevaatlik ja täielikult teadlik seeneliikidest, mida korjata plaanitakse. Vähekogenud seenelisele võib olla soovitud seeneliigi tuvastamine tõsiseks katsumuseks, sest eksimuse korral võib asi lõppeda surmaga. Sellest tulenevalt on vajalik, et seenelistele oleks loodud käepäraseid vahendeid, mis neid liigi kindlaksmääramisel aitaks.

Üheks enim levinumateks abisteks seente identifitseerimisel on erinevad seeneraamatud, mis aitavad inimestel seeni tuvastada visuaalselt seentele omaste tunnuste (nt. jalg, kübar, rõngas) põhjal. Seeneraamatud on enamasti kättesaadavad paberkandjal, kuid tänu tehnoloogia arengule on need leidnud tee ka seeneliste mobiiltelefonidesse. Nii on loodud erinevaid mobiilirakendusi kõikvõimalike seenelisi abistavate funktsioonidega paljudes keeltes.

1.1. Mobiilirakendused seente tuvastamiseks

Mobiiltelefonidele on loodud mitmeid erinevaid rakendusi, mis sisaldavad seenepilte ning lühikirjeldusi. Niisugused rakendused pakuvad tihti kasutajatele lihtsalt seente nimistus otsingu- ja sirvimisvõimalusi. Samuti on loodud rakendusi, mis lisaks seentega tutvumisele aitavad kasutajal tuvastada neid seeni visuaalselt seentele omaste tunnuste põhjal.

Lisaks käsitsi väliste tunnuste põhjal tuvastamisele, on loodud ka rakendusi, mis suudavad seeni tuvastada pildi põhjal automaatselt, kasutades sarnaselt käesolevale tööle tehismärgivõrke. Kahjuks ei leidu selliste rakenduste seas ühtegi eestikeelset rakendust või ei ole need avalikult kättesaadavad. Üheks suureks probleemiks võib arvustuste kohaselt pidada ka taoliste rakenduste seente määramise täpsust. Üheks selle võimalikuks põhjuseks võib olla liiga suur geograafiline haare. Rakendused üritavad tuvastada seeneliike üle kogu maailma, kus leidub tuhandeid liike erinevaid seeni, mistõttu võib kõvasti kannatada tuvastamise täpsus. Käesolev töö keskendub ainult Eesti metsades kasvavatele seentele, olles seega eestikeelne ja proovides saavutada sarnaste rakenduste hulgas parem täpsus nende seente hulgas.

Järgnevalt on kirjeldatud kolme Android platvormil tasuta kättesaadavat mobiilirakendust, mille peamiseks funktsiooniks on pildi põhjal seente tuvastamine.

1.1.1. Mushroom Identifier

“Mushroom Identifier” on nii Android [8] kui ka iOS [9] platvormidel tasuta kättesaadav mobiilirakendus, mis kirjelduse kohaselt peaks suutma pildi põhjal eristada rohkem kui 900 erinevat seeneliiki üle maailma. Rakendus üritab tuvastada kasutaja sisendpildi põhjal vastava seene liigi, kuvades tulemuseks kõige tõenäolisemate seente nimekirja. Samuti on kuvatud tuvastatud liigi lühikirjeldus ja teave söödavuse/mürgisuse kohta. Lisaks seente tuvastamisele võimaldab rakendus salvestada leitud seente asukohti kaardil, lahendada seentega seotud küsimustikke, lisada ja vaadata seente müügikuulutusi ning sirvida seente andmekogu.

Rakendus sisaldab endas mitmeid kasulikke funktsioone, kuid seente tuvastamise funktsioon eksis internetist võetud piltidega põgusal katsetamisel tulemusega väga tihti. Rakendus ei saanud kohati hakkama väga isikupärase välimusega Eestis kasvavate seente tuvastamisega (näiteks kevadkogrits, harilik kukeseen, punane kärbseseen). Tihti oli õige tulemus küll võimalike

tõenäoliste seente hulgas, kuid mitte esikohal. Kuna rakendus andis tulemuseks igasuguse pildi korral palju erinevaid tõenäolisi seeni, siis oleks kindlasti pidanud olema iga seene kohta antud ka vastav tõenäosuse hinnang, et kasutajal oleks mingisugunegi aimdus rakenduse täpsusest.

1.1.2. Rozpoznání hub

“Rozpoznání hub” on tasuta kättesaadav Android rakendus [10] seente tuvastamiseks. Rakendus annab pildi/piltide põhjal tulemuseks nimekirja erinevatest seentest koos tõenäosushinnangutega. Samuti on kuvatud tuvastatud liigi lühikirjeldus ja teave söödavuse/mürgisuse kohta. Rakendus kasutab parema tulemuse saamiseks seene leiukoha informatsiooni GPS asukoha näol ning võimaldab ühe tuvastamise raames sisestada mitu pilti.

Rakendus sai internetist võetud piltidega põgusal katsetamisel üsna hästi hakkama Eestis kasvavate isikupärase välimusega seente tuvastamisega, kuid eksis kohati natukene sarnasemate liikide (näiteks erinevad riisikad) tuvastamisel. Õige tulemus on tihti küll nimekirjas, kuid mitte esimesel kohal. Lisaks esineb rakenduse kasutajaliideses kohati väiksemaid vigu.

1.1.3. Mushrooms app

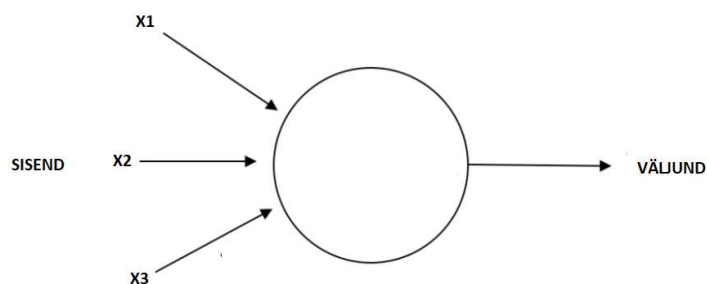
“Mushrooms app” on tasuta kättesaadav Android rakendus [11], mis sisaldab endas maailma 210 kõige enam levinud seene lühikirjeldust. Rakendus võimaldab neid seeni tuvastada käsitsi seene väliste tunnuste põhjal või automaatselt kasutaja telefonikaamera sisendi põhjal. Automaatse pildituvastuse põhjal kuvatakse tulemuseks seente nimekiri ja nende tõenäosuste hinnangud. Samuti on kuvatud tuvastatud liigi lühikirjeldus ja teave söödavuse/mürgisuse kohta. Lisaks on rakendusel kaardi funktsionaalsus, mille abil saab kasutaja näha leitud seente asukohta kaardil.

Rakendus sai internetist võetud piltidega põgusal katsetamisel enamasti hästi hakkama Eestis kasvavate isikupärase välimusega seente tuvastamisega, kuid eksis tihti visuaalselt sarnaste seente puhul. Õige tulemus on tihti küll nimekirjas, kuid mitte esimesel kohal. Erinevalt teistest sarnastest rakendustest puudub sellel võimalus tuvastada seeni varem tehtud piltidelt. Seene tuvastamiseks tuleb kaamera suunata seenele, misjärel rakendus üritab seda tuvastada reaalselt. Sellest tulenevalt ei suuda rakendus vahel pildil olevat seent tuvastada, kuna kasutaja käsi väriseb liialt või seen ei ole rakenduse jaoks sobival kaugusel.

2. Tehisnärvivõrgud

Tehisnärvivõrgud on üks masinõppe vormidest, mille tööpõhimõtte sarnaneb kohati inimese ajule – sellest ka nimetus [12]. Esimest korda esitleti tehisnärvivõrkude ideed aastal 1944, kuid omal ajal ei saavutanud see meetod suurt populaarsust, sest tehisnärvivõrgud nõuavad suurt arvutusjõudlust ja treeningandmete hulka, mida toona veel ei olnud [13]. Viimasel kümnendil toimunud graafikakaartide ning protsessorite arvutusvõimsuse hüppeline kasv, erinevate kogutud andmete mahu tohutu suurenemine ning masinõpet lihtsustava tarkvara loomine on praeguseks teinud tehisnärvivõrkudest enneolematult populaarse masinõppe meetodi [12].

Tehisnärvivõrk koosneb neuronitest. Neuron on matemaatiline kontseptsioon, mis saab sisendiks arvud, teeb nendega mingid arvutused ning tagastab ühe arvu [14] (joonis 1).



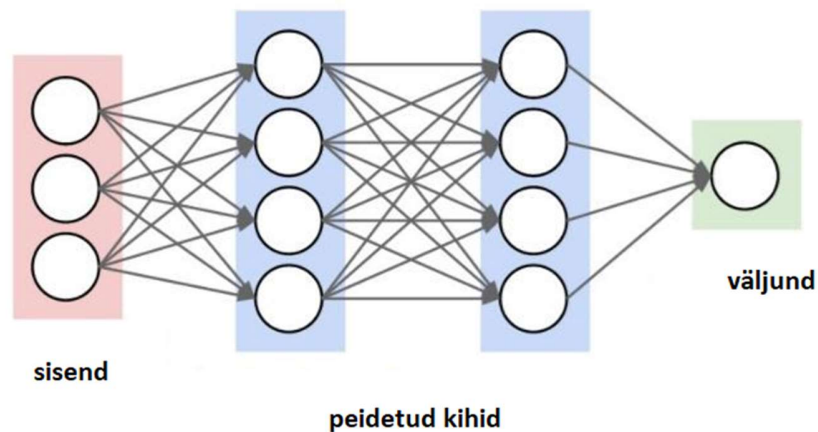
Joonis 1. Tehisnärvivõrgu neuron [15].

Neuronisse sisenemisel omistatakse igale sisendile kaal ning nende põhjal arvutatakse varem defineeritud matemaatilise mittelineaarse aktiveerimisfunktsioon väärtus [16]. Aktiveerimise eesmärgiks on mudelisse mittelineaarsuse sissetoomine [17], sest vastasel juhul oleks närvivõrgu sisend ja väljund omavahel seotud ühe lineaarse funktsiooniga, kuid enamik keerukaid andmeid ei ole lineaarselt eraldatavad. Lisaks on aktiveerimisfunktsioonide eesmärk hinnata, kas ja mis määral vastav neuron lõpptulemust mõjutab. Selle saavutamiseks on kasutatavatel aktiveerimisfunktsioonidel lävifunktsioonidele sarnane omadus jagada tulemused sisendi põhjal kaheks (mõjutab tulemust, ei mõjuta tulemust) [18]. Üheks sobivaks ja enamkasutatavaks aktiveerimisfunktsiooniks praktikas on näiteks ReLU funktsioon (joonis 2), sest masinal on selle väärtust lihtne arvutada. Samuti on ReLU funktsioon mittelineaarne ja tagab, et positiivse sisendi korral neuron aktiveeritakse, vastasel juhul mitte [17].

$$ReLU = \max(0, x)$$

Joonis 2. ReLU funktsioon, kus sisendiks on x .

Neuronite hulk moodustab kihi ning omavahel ühendatud sisendkiht, peidetud kihid ja väljundkiht omakorda moodustavadki tehisnärvivõrgu (joonis 3). Sisendiks on mudeli treenimisel varasemalt kogutud treeningandmed ning väljundiks tehisnärvivõrgu arvutuste järel saadud andmetele vastavad väärtused [16]. Ühte sisendandmete tehisnärvivõrgu läbimist kutsutakse edasileviks.



Joonis 3. Kihid moodustavad tehisnärvivõrgu [19].

Pärast tehisnärvivõrkudes sisendi edastamist ja sellele vastava väljundi saamist on võimalik väljundit võrrelda õigete tulemustega. Erinevuse hindamiseks kasutatakse matemaatilist kaofunktsiooni, mille tulemus on positiivne arvuline hinnang ennustuste ja õigete tulemuste vahel. Binaarse klassifitseerimise (õige või vale, 0 või 1) puhul kasutatakse tihti ristentroopia kaofunktsiooni (joonis 4). Ristentroopia kaofunktsioon saab sisendiks 2 tõenäosusjaotust (ennustused, tulemused) ning tagastab nende kahe tõenäosusjaotuse lahknevust hindava arvu [20]. Mida suurem on vastav väärtus, seda rohkem tõenäosusjaotused teineteisest erinevad – seda suurem on ennustuste ja õigete tulemuste erinevus.

$$H(p, q) = - \sum_i p_i \log q_i$$

Joonis 4. Ristentroopia valem, kus p on tõesed väärtused ja q ennustatud väärtused.

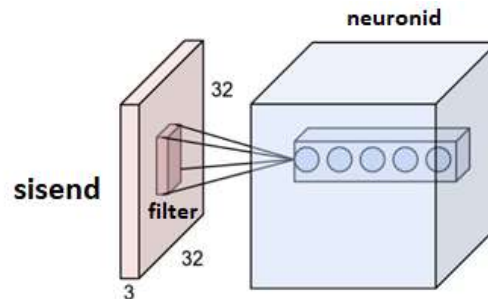
Tehisnärvivõrgu eesmärgiks on õigete tulemuste ennustamine ehk seega kaofunktsiooni väärtuse minimeerimine. Selle jaoks kasutatakse tehisnärvivõrkudes tagasilevi ja gradientlaskumist. Gradient iseloomustab funktsiooni muutumise kiirust ja selle arvutamiseks on vaja arvutada funktsiooni parameetrite osatuletised. Tagasilevialgoritm liigub närvivõrgus tagant ette (väljundkihist sisendkihi poole) arvutades järjest kõikide parameetrite osatuletised ning saades tulemuseks väärtused, mis iseloomustavad parameetrite muutmise suunda nii, et kaofunktsioon väheneks [21].

Tehisnärvivõrgu mudeli treenimine koosneb edasilevi ja tagasilevi kordustest. Edasilevi käigus arvutab algoritm parameetritele sisendile vastava tulemuse. Tagasilevi käigus muudetakse neid parameetreid parema tulemuse saavutamise suunas. Neid tegevusi korratakse seni, kuni edasilevi järel saadud ennustustäpsus ja kadu on saavutanud vastavalt maksimaalse ja minimaalse väärtus ning edasine treenimine tulemusi enam ei paranda.

Tehisnärvivõrkudel on võime leida keeruliste andmete vahel seaduspära, korduvaid mustreid ja trende, mille leidmisega inimene või teised masinõppe algoritmid hakkama ei saa [22]. Seega on tehisnärvivõrkude kasutamine otstarbekas selliste probleemide korral, mille olemus ise on raskesti mõistetav. Üheks selliseks probleemiks on näiteks pildituvastus. Pildid on mitmemõõtmelised andmed ning programmeerijal on väga raske kirjeldada kõiki võimalike tunnuseid, mida pildilt eristada [12]. Siinkohal on probleemi lahendamiseks mõistlik kasutada tehisnärvivõrke, mis suudavad tuvastada need tunnused iseseisvalt olemasolevate näidete põhjal.

Pildituvastamiseks kasutatakse enamasti konvolutsioonilisi närvivõrke, mis on üks tehisnärvivõrkude liik. Konvolutsioonilised närvivõrgud erinevad klassikalistest närvivõrkudest selle poolest, et lisaks klassikalisele neuroneid sisaldavale kihile kasutatakse nendes konvolutsioonilisi kihte. Konvolutsiooniline kiht koosneb sarnaselt klassikalisele kihile neuronitest, kuid kogu sisend ei ole juhitud läbi kõigi neuronite. Selle asemel kasutatakse konvolutsioonilises kihis kaalude asemel fikseeritud suurusega risttahuka kujulisi alasid ehk

filtreid. Filtreid liigutatakse üle terve sisendi (pildi) ning igal liigutusel juhitakse alaline sisend läbi vastavate neuronite (joonis 5) [19].



Joonis 5. Konvolutsioonilise kihi tööpõhimõte [19].

Piltide puhul sõltub andmete hulk pildi suurusest. Seega on suurte mõõtmetega piltide puhul arvutatavate kaalude (parameetrite) arv klassikalise tehisnärvivõrgu puhul väga suur ja nõuab seetõttu ka suurt arvutusjõudlust. Filtrite kasutamise loogika silub seda probleemi, sest neuronid on terve pildi asemel ühendatud ainult kindla pildi alamosaga (joonis 5) [19]. Lisaks on piltandmete puhul selline lähenemine üsna loogiline, sest enamasti esinevad tunnused pildidel sarnaselt filtritele väikeste aladena (näiteks näo puhul silm või auto puhul ratas). Kasutades aga ainult klassikalisi närvivõrke otsitakse tunnuseid pildi kõikide osade vahel.

3. Rakenduse kirjeldus

Töö praktiline pool koosneb kahest osast: tehisnärvivõrgu mudeli loomisest ja treenimisest ning mobiilirakenduse loomisest. Tehisnärvivõrgu mudeli eesmärk on sisendi (pildi) põhjal tuvastada seal leiduva seene liik, andes tulemuseks tõenäosusjaotuse üle võimalike seeneliikide. Mobiilirakenduse eesmärk on tehisnärvivõrgu mudeli ja kasutaja sisendi sidumine ning mudelist saadud tulemuse töötlemine ning informatiivsel kujul esitamine. Peatüki lõpus on välja toodud mõlema osa loomisel kasutatud tehnoloogilised vahendid.

3.1. Tehisnärvivõrgu mudel

Tehisnärvivõrgu mudeli loomine on töö kontekstis kõige suurema tähtsusega, sest just mudeli abil toimub seene liigi tuvastamine. Käesolev peatükk kirjeldab mudeli treenimiseks vajalikke andmeid ning nende kogumise ja eeltöötlemise protsesse. Samuti on kirjeldatud lõpliku mudeli struktuur ning sobiva mudelini jõudmiseks tehtud eelnevad katsetused.

3.1.1. Andmed

Tehisnärvivõrgu mudeli treenimiseks kogutud piltide andmestik koosneb 8004 erinevast pildist 43 erineva seeneliigi kohta (lisa 1). Valitud seeneliigid on sagedased Eestis kasvavad söögi- ja mürkseened [5,23]. Andmed on kogutud andmebaasidest Mushroom World, Mushroom Observer, MushroomExpert.com, eElurikkus ning kasutades Google'i piltide otsingumootorit.

Andmebaasidest Mushroom World, MushroomExpert.com ja eElurikkus andmete kogumiseks on töö autor loonud vastavad andmekaeve skriptid programmeerimiskeeles Python ning nende lähtekood on avalikult kättesaadav autori GitHub repositooriumis [24]. Mushroom Observer andmebaasist on juba varasemalt andmekaevet teostatud ning käesolevas töös kasutati neid samu andmeid [25]. Lisaks on baasandmestiku suurendamiseks käsitsi pilte alla laetud Google'i otsingumootorist, kasutades selleks Google Chrome'i laiendust Image Downloader.

Mudeli treenimisel on andmed jagatud treeninghulgaks (70%), valideerimishulgaks (15%) ja testhulgaks (15%). Treeningandmete vähesuse tõttu on selle kunstlikuks suurendamiseks kasutatud treenimise käigus reaalses andmete juurde genereerimist, kasutades selleks Kerase

teegi ImageDataGenerator klassi. See võimaldab reaajas iga epohhi käigus vaatluse all oleva partii andmete (juhuslikku) muutmist erinevate pilditöötluse võtete abil [26].

Andmestiku suurendamiseks on andmete peal treenimise käigus juhusikult kasutatud järgnevaid pilditöötluse võtteid:

- 1) pildi pööramine;
- 2) pildi nihutamine mööda horisontaaltelge;
- 3) pildi nihutamine mööda vertikaaltelge;
- 4) pildi kallutamine;
- 5) pildi skaleerimine;
- 6) pildi horisontaalne peegeldamine;

Lisaks on kõik mudelis kasutatavad andmed viidud suurusele 96 x 96 pikslit. Iga kanali (punane, roheline, sinine) pikslitel põhjal on arvutatud vastava kanali keskmine pikslite väärtus ning standardhälve. Seejärel on andmed normaliseeritud kõikide kanalite pikslitelt vastava kanali keskmise piksli väärtuse lahutamise ja vastava kanali standardhälbega läbi jagamise teel. See tagab, et andmete (antud juhul pikslite) väärtuste keskmine väärtus oleks 0 ja standardhälve 1.

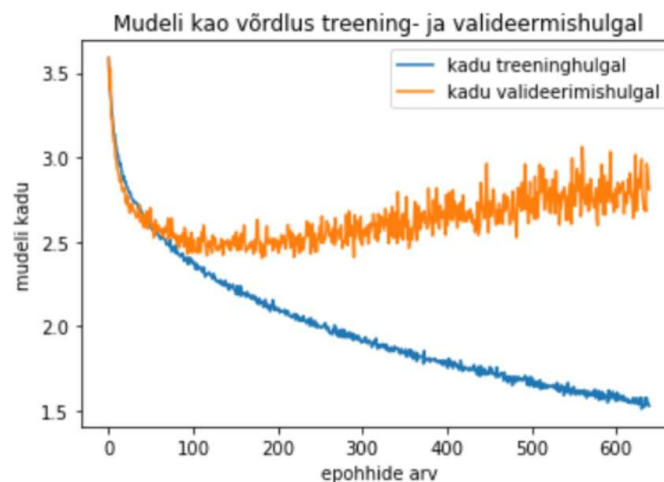
3.1.2. Mudeli koostamine

Mudeli arhitektuuri disainimisel on aluseks võetud mõned juba varasemalt loodud konvolutsioonilised tehismärgivõrgud nagu näiteks LeNet, AlexNet ja VGGNet [27]. Sellised närvivõrgud on aga üsna sügavad ja nõuavad suurt arvutusjõudlust, mida autori kasutuses olev üks sülearvuti ei võimalda või nõuab see tohutult aega. Seega on mudeli loomisel proovitud leida tasakaal mudeli sügavuse ja ajalise kestvuse vahel, hoides samas varasemate mudelitega sarnaseid jooni. Mudeleid on treenitud seni, kuni pole enam märgata täpsuse kasvu treening- ega valideerimishulgal või kui kadu valideerimishulgal ja treeninghulgal hakkavad liikuma graafikul erinevates suundades. Kõikide katsetatud mudelite puhul on välja toodud täpsus, et kõige tõenäolisem ennustus on õige. Mudeli loomise näite lähtekood on kättesaadav autori GitHub repositooriumis [24].

Kõige esimese mudeli (Mudel 1) loomise eesmärgiks on saada esialgne tulemus, mida edasise arendamise käigus täiustada. Mudeli disainimisel on eeskujuks VGGNet [27]. Katsetatud on

sellele sarnast peitkihtide paigutust, kuid mille sügavust on võimalikult palju minimeeritud, et ajalist kulu vähendada (lisa 2a). Mudeli treenimisel on läbitud 4300 epohhi, milleks kulus aega ligikaudu 37 tundi. Treeninghulga andmete põhjal saavutatud täpsus on 35,7% ja testhulga andmete põhjal saavutatud täpsus on 23,1%. Nende täpsuste vahe ei ole eriti suur ning seega tundub, et mudel suudab võrdlemisi hästi treeningu põhjal üldistusi teha, kuid treeninghulgal saavutatud täpsus on küllaltki madal. Sellest tulenevalt on järgmistel katsetusel tõstetud mudeli parameetrite arvu, mis eelduste kohaselt võiks tõsta mudeli saavutatavat täpsust vähemalt treeninghulga andmete põhjal.

Järgneval katsel (Mudel 2) on parameetrite arvu tõstmiseks kihtide paigutus jäetud samaks, kuid suurendatud on Conv2D ja Dense kihtide suurusi (lisa 2b). Selle tulemusena tõusis mudeli parameetrite arv 3 korda. Mudeli treenimisel läbiti 640 epohhi, milleks kulus aega ligikaudu 12 tundi. Treeninghulga andmete põhjal saavutatud täpsus on 56,1%, mis on ootuspäraselt parem kui eelmisel mudelil ning täpsus testhulga andmete puhul on 30,4%.



Joonis 6. Teise mudeli kao võrdlus treening- ja valideerimishulgal.

Analüüsides mudeli kadu epohhide raames nii treening- kui valideerimishulgal (joonis 6), on näha, et kaod treeninghulgal ja valideerimishulgal hakkavad lahkema. Selline nähtus viitab ülesobitamisele. Mudeli üldistusvõime parandamiseks ja ülesobitamise vältimiseks on töös proovitud lisada Dropout kiht, mis jätab arvestamata juhuslikult valitud neuronid. Neuronite arvestamata jätmisel lõhutakse neuronites varem õpitud seaduspärasid. Selle tulemusena ei suuda

model nii efektiivselt õppida treeningandmete vahel olevat seaduspära ja vähendab eelduste kohaselt ülesobitamise riski.

Järgneval katsel (Mudel 3) on mudeli teiste kihtide paigutused ja suurused jäetud samaks ning lisatud on vaid üks Dropout kiht (lisa 2c). Mudelit treenimisel on läbitud 160 epohhi, milleks kulus aega ligikaudu 10 tundi. Treening- ja testhulga andmete põhjal saavutatud täpsused on vastavalt 48% ja 33%. Ootuste kohaselt tõusis testhulga andmete põhjal saavutatud täpsus ligikaudu 3%.

Lisaks liigi klassifitseerimisele proovis autor kõiki eelpool mainitud mudeleid, et klassifitseerida ainult liigi mürgisust. Selleks jagas autor andmed nelja rühma: söödav, mürgine, alkoholiga mürgine, kupatatult söödav kuid muidu mürgine [6]. Autor treenis kõiki varasemalt kirjeldatu mudeleid uuel jaotusel, kuid kõigil kordadel jäi täpsus testhulgal alla 60%. Nelja klassi ennustamise puhul on juhusliku õigesti arvamise tõenäosus 25% ning sellest tulenevalt võib saavutatud täpsust pidada küllaltki madalaks. Seega on autor otsustanud mürgisust klassifitseerivat mudelit rakenduses mitte kasutada. Üheks kehva tulemuse põhjuseks võib pidada andmete liiga suurt erinevust ühe klassi raames ning vähest andmete hulka. Sellest tulenevalt ei suuda model leida üldistavaid tunnuseid klasside eristamiseks.

3.1.3. Lõpliku mudeli kirjeldus

Lõpliku mudeli loomisel kombineeris autor kõiki mudeli koostamisel proovitud tehnikaid ning üritas saavutada võimalikult suurt täpsust testhulga andmete põhjal. Mudeli koostamise käigus suutis autor mudeli täpsust järjepidevalt tõsta (tabel 1).

<i>Mudel</i>	<i>Parameetrite arv</i>	<i>Täpsus treeninghulgal</i>	<i>Täpsus testhulgal</i>
<i>Mudel 1</i>	9,051,755	35,7%	23,1%
<i>Mudel 2</i>	27,151,115	56,1%	30,4%
<i>Mudel 3</i>	27,151,115	48,0%	33,0%
<i>Lõplik</i>	2,539,147	45,5%	37,1%

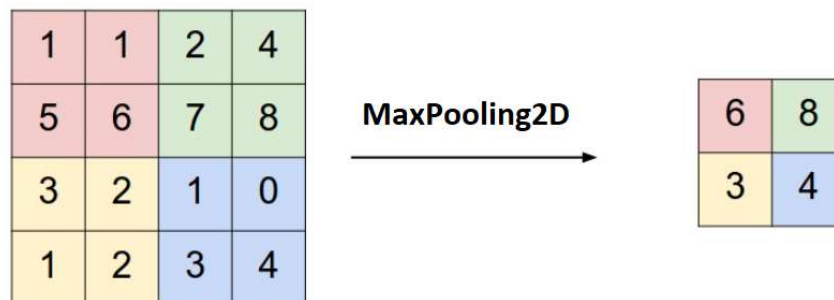
Tabel 1. Katsetatud mudelite võrdlus.

Lõplik närvivõrgu mudel, mida kasutatakse rakenduses seente tuvastamiseks, saavutas testhulgal täpsuse 37,1% (joonis 7).

SISEND
Conv2D_64
ReLU
Conv2D_64
ReLU
MaxPooling2D
Dropout (0.1)
Conv2D_32
ReLU
Conv2D_32
ReLU
BatchNormalization
MaxPooling2D
Dropout (0.1)
Flatten
Dense_96
ReLU
Dense_43
Softmax
VÄLJUND

Joonis 7. Lõpliku mudeli arhitektuur.

Mudeli treenimisel on läbitud 210 epohhi, milleks kulus aega ligikaudu 18 tundi. Mudel koosneb kahest konvolutsioonilistele närvivõrkudele omasest kihtide plokist (Conv2D, ReLU, MaxPooling2D). Conv2D kiht tähendab konvolutsioonilist kihti ning ReLU, et kihi neuronitel on kasutatud ReLU aktiveerimisfunktsiooni. Konvolutsiooniliste kihtide vahel kasutatakse tihti MaxPooling2D kihti, mis vaatab fikseeritud suurusega alasid eelmise kihi väljundist, jättes alles igast alast maksimaalse väärtuse (joonis 8). Selle tulemusena kitsenevad eelmise kihi väljundi mõõtmeid ning väheneb parameetrite hulk, mis omakorda vähendab ülesobitamist ja kiirendab algoritmi tööd [19].



Joonis 8. MaxPooling2D kihi tööpõhimõte [19].

Lisaks on konvolutsiooniliste kihtide vahel kasutatud Dropout kihti, mis jätab arvestamata juhuslikult valitud neuronid. Selle tulemusena ei õpi mudel treeningandmete vahel olevat seaduspära nii spetsiifiliselt ja suudab eelduste kohaselt saavutada seeläbi uute andmete peal parema tulemuse. Teises konvolutsioonide plokis kasutatakse ka BatchNormalization kihti, mis normaliseerib vahetulemused, eesmärgiga kiirendada treenimist.

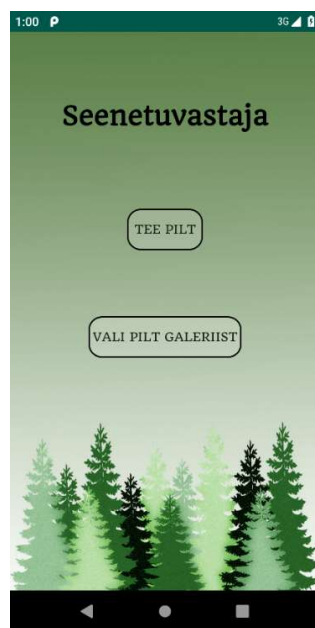
Pärast konvolutsioone teeb Flatten kiht mitmemõõtmelised andmed ühemõõtmelisteks, et need edastada klassikalistesse närvivõrkude (Dense) kihtidesse. Viimases Dense kihis on aktiveerimisfunktsioonina kasutatud softmax funktsiooni (klassifitseerijat). Softmax kasutab mitme klassi ristentroopiat, mis on ristentroopia edasiarendus ning tagastab tõenäosusjaotuse üle 43 klassi ehk seeneliigi. Mudeli väljundiks ongi 43 seeneliigi tõenäosused saadud sisendile.

3.2. Mobiilirakendus

Mobiilirakendus on loodud Android 9 versiooni jaoks, kuid minimaalse versioonina on lubatud ka Android 6. Rakenduse eesmärk on kasutaja sisendi (pildi) põhjal seeneliigi tuvastamine. Kasutaja saab teha pildi telefonikaameraga või valida juba olemasoleva pildi telefoni pildigaleriist. Kuna seeneliike hakkab tuvastama eelnevalt loodud tehismärgivõrk, siis on mobiilirakenduse puhul tegu kasutajaliidesega, mis võimaldab mugavat suhtlust kasutaja ja närvivõrgu mudeli vahel.

3.2.1. Rakenduse ülesehitus

Rakenduse funktsioonide hulka kuuluvad kasutajapoolne pildi tegemine/valimine ning selle põhjal ennustatud seeneliikide kuvamine. Seega koosneb rakendus kahest lihtsast vaatest: peamenüü vaade (joonis 9) ja ennustuste vaade (joonis 10). Kõik vaadetes kasutatud pildid on võetud Pixabay pildikogust [28] ning neid võib vabalt kasutada. Kogu rakenduse lähtekood on avalikult kättesaadav autori GitHub repositooriumis [24]



Joonis 9. Peamenüü vaade.



Joonis 10. Ennustuste vaade.

Android rakenduste arendusel vastab igale ekraanivaatele eraldi toiming (klass) [29]. Peamenüü funktsioone juhtiva toimingu ülesanded on:

- 1) kasutajalt sisendi küsimine;
- 2) sisendi tehiskäsitluse mudeli sisendkihi kujule viimine;
- 3) sisendi tehiskäsitluse mudelile edastamine;
- 4) sisendile vastava tehiskäsitluse tulemuse saamine ja ennustuste toimingule edastamine;

Kasutaja sisend on kas telefonikaameraga tehtud või telefoni galeriist valitud pilt. Mõlemal juhul on vaja järjestada baidid pildis ümber nii, et see sobituks käsitluse mudeli pildiformaadiga. Lisaks lahutatakse igalt pikslilt keskmine väärtus ja jagatakse standardhälbega, et andmed oleksid sarnaselt tehiskäsitluse mudeli treeningandmetele normaliseeritud. Keskmised

väärtused ja standardhälbed on arvutatud treeninghulga andmetel iga piksli positsiooni (punane, roheline, sinine) kohta eraldi. Seejärel saab pildi edastada tehisnärvivõrgu mudelile, mis on varasemalt treenitud ning mobiilirakenduse ressursina kaasas. Kõik need tegevused on kirjeldatud töö autori kirjutatud *Classifier.java* klassis.

Tehisnärvivõrgu mudelile edastatud sisendi tulemuseks on 43 erineva seeneliigi tõenäosused. Kõik võimalikud liigid on töö autor rakendusse sisse kodeerinud ning iga liik on esitatav *Mushroom.java* klassi isendina. Isenditele rakendusesiseseks ligipääsemiseks on autor loonud andmepääsu objekti, mis on klassi *MushroomDAO.java* isend. Peamenüü toimingus seotakse tulemuseks saadud tõenäosused vastavate *Mushroom.java* klassi isenditega ning edastatakse ennustuste toimingule.

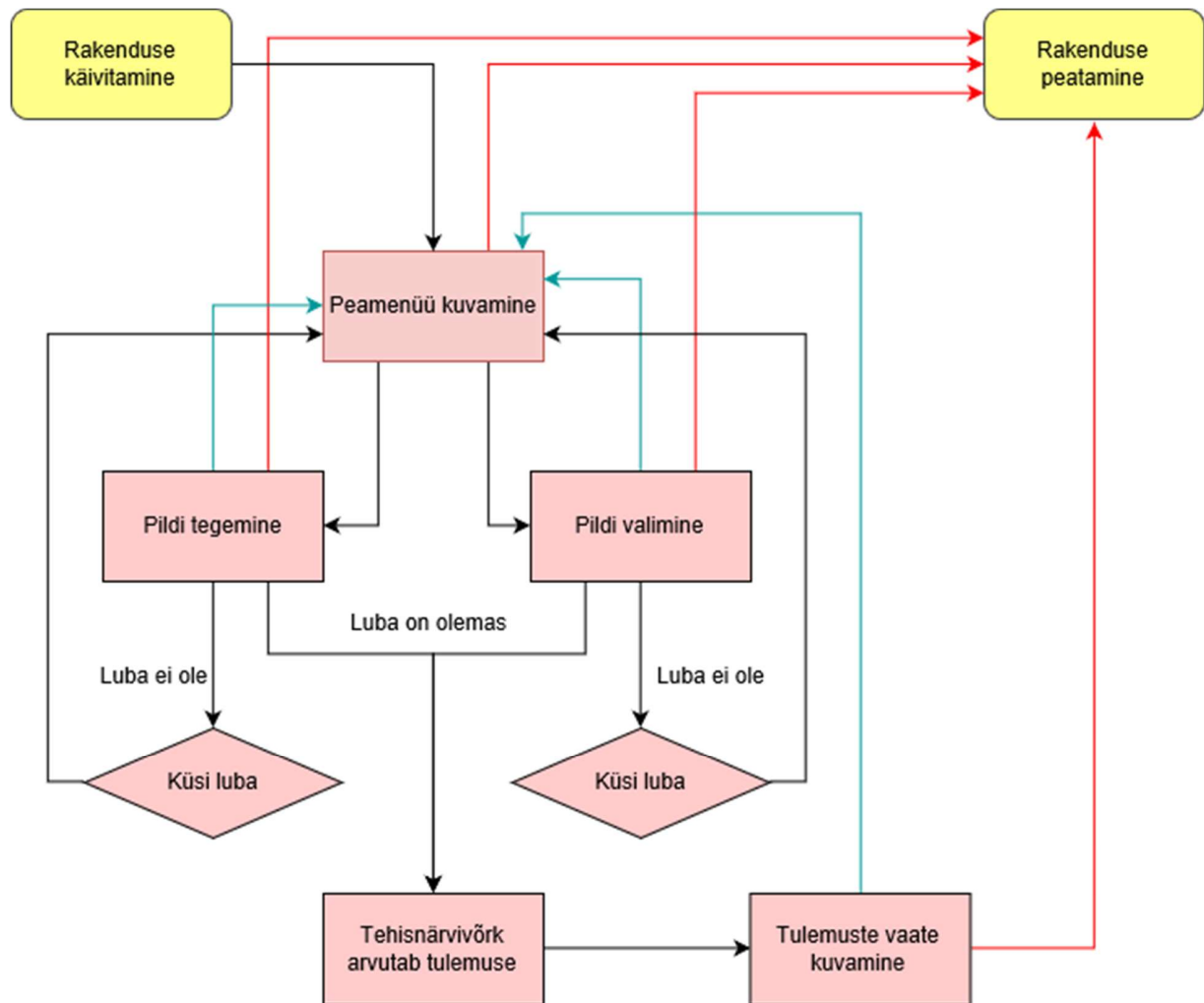
Ennustuste toimingu ainus eesmärk on kuvada kasutajale peamenüü toimingult saadud tulemused töödeldud kujul. Kuna ennustuse korral on väga paljude liikide tõenäosuste väärtus 0, siis on tulemuste hulgast välja filtreeritud need, mille tõenäosus on alla 1%. Allesjäänud tulemustest valitakse 5 kõige suurema tõenäosusega seeneliiki. Selleks, et ennustuste vaates kuvada *Mushroom.java* klassi isendid ühtse järjendina (joonis 10), on autor loonud *MushroomAdapter.java* klassi, mis seob omavahel rakendusse sisse kodeeritud seene liigi ennustuste vaatega.

3.2.2. Rakenduse kasutamine

Rakenduse avamisel suunatakse kasutaja peamenüü vaatesse (joonis 9). Peamenüüst saab kasutaja valida, kas soovib teha pildi koheselt telefonikaameraga või valida galeriist mõne juba varasemalt tehtud meelepärase pildi. Esmakordselt käivitamisel küsib rakendus luba vastavalt telefonikaamera kasutamiseks või galerii piltidele ligipääsemiseks. Seni, kuni kasutaja ei anna rakendusele neid õigusi, ei saa neid rakenduse funktsioone kasutada. Pärast õiguste andmist ja meelepärase pildi tegemist/valimist suunatakse kasutaja ennustuste vaatesse. Selleks ajaks on pilt taustal edastatud tehisnärvivõrgule ning vastavate seeneliikide tõenäosused leitud.

Ennustuste vaates (joonis 10) kuvatakse kasutajale informatsioon sisendile vastava kuni viie kõige tõenäolisema seeneliigi kohta. Kuvatavad parameetrid on liigi eestikeelne nimetus, ladinakeelne nimetus ja mürgisus. Liigid, millele autoril õnnestus leida selline vastav pilt, millele ei tule rakendusesiseselt viidata, on kuvatud ka vastav pilt (ligikaudu 80% liikide korral).

Võimalikud mürgisuse klassid on: mürgine, söödav, kupatatult mürgine kuid muidu söödav, alkoholiga mürgine [6]. Kasutaja saab tulemuse vaatamiseks kerida seente nimekirja üles ja alla. Uue pildi sisestamiseks saab kasutaja vajutada tagasinoolt, misjärel suunatakse kasutaja tagasi peamenüü vaatesse.



Joonis 11. Rakenduse kasutamine, kus punased jooned tähistavad „Kodu“ nupu vajutamist, sinised jooned tagasinoole vajutamist ning mustad jooned klassikalist töövoogu.

3.3. Kasutatud tehnoloogiad

Tehisnärvivõrke on võimalik programmeerida keeles Python, kasutades selleks vaid keelde sisse ehitatud funktsioone ja andmetüüpe. Nullist närvivõrgu mudelite loomine nõuab palju tööd ning kuna tehisnärvivõrgud kasutavad suurt arvutusjõudlust siis on väga oluline, et need oleksid programmeeritud töötama efektiivselt. Seega on töös kasutatud juba spetsiaalselt masinõppe

probleemide lahendamiseks loodud tarkvara ning vahendeid, mis võimaldaksid lihtsalt närvivõrgu mudeleid luua, treenida ja kasutada. Kõigi tehnoloogiate valikul on ajendiks autori varasem kogemus nende tehnoloogiate kasutamisega.

3.3.1. TensorFlow

Tensorflow on avatud lähtekoodiga teek matemaatiliste arvutuste tegemiseks., millel on liidesed erinevates programmeerimiskeeltes nagu näiteks Python, C++ ning Java. Teek on loodud Google Brain meeskonna poolt eesmärgiga lihtsustada masinõppe ja tehiskärvivõrkudega seotud uurimuste teostamist, kuid seda saab rakendada ka paljude teiste matemaatiliste probleemide lahendamisel. TensorFlow'd saab jooksutada nii protsessoril kui ka graafikakaardil ning TensorFlow abil loodud tehiskärvivõrgu mudeleid saab kasutada erinevatel platvormidel nagu näiteks Linux, Windows, macOS, Android ja iOS [30].

TensorFlow teeki on töös kasutatud tagasüsteemina tehiskärvivõrgu mudeliga seotud arvutuste tegemiseks. Mobiilirakenduses on tehiskärvivõrgu mudeliga töötamiseks kasutatud TensorFlow kergekaalulisemat lahendust TensorFlow Lite, mis on loodud spetsiaalselt mobiili- ja manusseadmete jaoks [31]. Teek on kasutusele võetud, sest tegemist on kõige enam kasutatava masinõppeteegiga ja seega leidub selle kasutamise kohta kõige enam materjale.

3.3.2. Keras

Keras on avatud lähtekoodiga tehiskärvivõrkude teek, mis on kirjutatud programmeerimiskeeles Python. Teek on loodud projekti ONEIROOS raames, eesmärgiga võimaldada teadustöös lihtsat ja kiiret kärvivõrkudega eksperimenteerimist ja prototüüpide loomist [32]. Keras on universaalne ja paindlik ning võimaldab loodud kärvivõrkude mudeleid rakendada paljudel erinevatel platvormidel (iOS, Android, veeb, Google Cloud, JVM, Raspberry Pi). Lisaks on Keras kergesti ühilduv tuntud sügavõppe tagasüsteemidega nagu näiteks TensorFlow, CNTK ja Theano [33].

Kerases teeki on töös kasutatud TensorFlow eessüsteemina, et kirjeldada mugavalt tehiskärvivõrgu mudeli ehitust ja teostada mudeli treenimisega seotud toiminguid. Teegi valiku põhjuseks on rohke materjalide hulk ja kasutajasõbralikkus.

3.3.3. OpenCV

OpenCV on avatud lähtekoodiga arvutinägemise ja masinõppe teek, mis on kirjutatud programmeerimiskeeltes C ja C++. Teek sisaldab üle 2500 optimeeritud algoritmi ning on peamiselt loodud reaalajas visuaalse informatsiooni töötlemiseks, kuid võimaldab sealhulgas teostada ka paljusid erinevaid pildi töötlemisega ja tuvastamisega seotud protseduure. OpenCV toetab Linux, Windows, Android ja Mac OS operatsioonisüsteeme ning teegil on liidesed programmeerimiskeeltele C++, Python, Java ja MATLAB [34].

Teeki on töös kasutatud piltide maatriksi kujul sisse lugemiseks ja piltide mõõtmete reguleerimiseks. Teek on valitud, kuna seda selle kasutamine on tehtud väga lihtsaks.

3.3.4. Android Studio

Android Studio on 2013. aastal Google I/O konverentsil tutvustatud IntelliJ IDEA'1 põhinev tasuta kättesaadav integreeritud arenduskeskkond Android rakenduste loomiseks [35]. Lisaks IntelliJ koodiredaktorile ja tööriistadele sisaldab Android Studio palju erinevaid Android rakenduste arendamist hõlbustavaid võimalusi [36]. Android Studio abil saab Android rakendusi kirjutada programmeerimiskeeltes Java ja Kotlin.

Android Studio keskkonnas on Java keeles programmeeritud kogu mobiilirakendus. Keskkond on valitud seetõttu, et sisaldab kasutajaliidese loomist toetavat visuaalset tööriista ning võimaldab rakenduse kohest konstrueerimist ja katsetamist.

4. Tulemused

Rakendus on valminud kevadel, mil enamik Eestis leiduvaid seeneliike veel ei kasva. Sellest tulenevalt on keeruline läbi viia kasutusjuhu lähedast testimist – teha metsas telefoniga seentest pilte ja võrrelda tulemusi. Mudeli ennustustulemusi on testitud ja analüüsitud ainult baasandmestikust eraldatud testhulga peal.

4.1. Testimine

Testhulk, millel mudeli täpsust testiti, moodustab baasandmestikust 15% ja sinna kuulub 1201 pilti. Tehisnärvivõrgu mudel saavutas testhulgal täpsuse 37,1%. Rakendus kuvab iga ennustuse korral maksimaalselt 5 kõige tõenäolisemat seeneliiki, kusjuures kuvatakse ainult sellised liigid, mille tõenäosus on suurem kui 1%. Seega on ka oluline teada, kui suur on tõenäosus, et õige tulemus on nende maksimaalselt 5 kõige tõenäolisema liigi seas (tabel 2).

	<i>TOP 1</i>	<i>TOP 2</i>	<i>TOP 3</i>	<i>TOP 4</i>	<i>TOP 5</i>
<i>Täpsus</i>	37,1%	49,5%	58,6%	66%	71,2%

Tabel 2. Õige vastuse viie kõige tõenäolisema ennustuse hulka sattumise tõenäosus..

Testhulgal (TOP 1) tulemuste hindamiseks genereeris autor ennustuste ja õigete vastuste põhjal segadusmaatriksi ning analüüsis järgmisi tunnuseid (lisa 3):

- 1) täpsus, mis hindab, kui tihti oli ennustus õige, kui ennustati kindlat liiki;
- 2) saagis, mis hindab, kui tihti ennustati kindlat liiki, kui see oli õigeks vastuseks;
- 3) liigi esinemist testhulgas;
- 4) liigi ennustamise kordi;

Segadusmaatriksit uurides selgub, et mudel ei saa üldse hakkama põhja-külmaseene, männi-kivipuraviku, kitsemampeli ja pruuni sametpuraviku tuvastamisega. Nende liikide puhul on nii täpsuse kui ka saagise väärtus 0, mis tähendab, et mudel ei suutnud ühtegi korda pildilt neid liike õigesti tuvastada. Lisaks ei üritanud närvivõrgu mudel neid liike pakkuda teiste seente korral. Kuna mudel ei suuda neid seeni ära tunda ning ei aja neid ka teiste seentega segamini, siis võib järeldada, et mudel ei õppinud neid seeni üldse ära tundma.

Lisaks eelmainitud liikidele on väga madala täpsuse ja saagisega seepheinik ning sügiskogrits. Sarnaselt eelmistele liikidele ei suuda mudel ka neid seeni hästi tuvastada, kuid lisaks sellele pakub mudel neid väga tihti teisteks seenteks. Sellest võib järeldada, et mudel on nende seente puhul õppinud treeninghulga põhjal ära sellised tunnused, mis testhulgas olevatele uutele piltidele ei üldistu.

Kõige paremini oskab mudel ennustada punast kärbseseent. Punase kärbseseene puhul on nii täpsus kui saagis võrdlemisi suured (vastavalt 0,83 ja 0,78). Selle liigi esinemiskordade arv testhulgas on keskmine ning mudel ei paku seda ka massiliselt teisteks liikideks. Sarnaselt ka mitmetele teistele seeneliikidele, on punane kärbseseene väga ainulaadse välimusega (punane ja valged täpid) ning võib arvata, et mudel on need tunnused küllaltki hästi ära õppinud. Lisaks punasele kärbseseenele on kõrgemad täpsuse ja saagise väärtused ka kevadkogritsal, kollasel pilvikul ja vereval vöödikul.

Osade liikide puhul esineb kõrge täpsuse väärtus, kuid madal saagis. Niisugune olukord tähendab seda, et mudel tegi kindlat liiki ennustades tihti õige otsuse. Samas ei suutnud mudel liigi kasuks otsustada paljudel juhtudel, kui liik pildil tegelikult oli. Sellisteks liikideks on näiteks aasšampinjon ja lilla ebaheinik. Vastupidiselt madala täpsusega ja kõrge saagisega on näiteks valge kärbseseene. See tähendab, et mudel suutis enamasti liigi pildilt tuvastada, kui see seal oli, kuid pakkus seda väga tihti valesti teisteks liikideks.

Ülekaalukalt kõige rohkem esinemisi testhulgas ja ka ennustamise kordi on soomustindikul ja rohelisel kärbseseenel. Mõlemad liigid on ka baasandmestikus arvukuselt väga kõrgetel kohtadel. Soomustindikul on liikide seast kõige kõrgem saagis, kuid ligikaudu 2 korda madalam täpsus. Mudel suudab soomustindiku enamikel kordadel pildilt tuvastada, kuid ajab seda segamini teiste liikidega. Rohelise kärbseseene puhul on saagis keskmine, kuid täpsus üsna madal. Kuigi baasandmeid selle liigi kohta on võrreldes teiste liikidega palju, ei suutnud mudel seda siiski korralikult ära õppida.

4.2. Hinnang

Käesoleva töö eesmärk oli luua mugava kasutajaliidesega mobiilirakendus, mis suudaks tuvastada Eesti metsades kasvavaid seeni.

Autori ning nelja eesti keelt kõnelevate tudengite hinnangul on loodud rakendus lihtsasti kasutatav ja arusaadav. Kasutajal on võimalik kiirelt sisestada omapoolne telefonikaameraga tehtud või galeriist valitud pilt ning saada koheselt sisendpildi põhjal tulemus. Niisugust sujuvust ja kohest vastuse kuvamist ei pakkunud mitte ükski juba olemasolev rakendus.

Testhulga tulemuste (tabel 2) ja segadusmaatriksi (lisa 3) analüüsile toetudes võib öelda, et seente tuvastamise täpsus on liiga madal. Rahule võib jääda punase kärbseseene ennustamise tulemustega, mis on stabiilsed ja küllaltki kõrged. Kõrgemaid tulemusi saab täheldada veel paari liigi puhul, kuid enamike liikide puhul on mudeli ennustusvõime halb.

Kõige tõenäolisema ennustuse õigeks osutumise (TOP 1) täpsus on 37,1%. Autori hinnangul võiks tulemusega rahule jääda, kui see täpsus oleks võrdne praeguse TOP 5 täpsusega (tabel 2). Seejuures võiks õige vastuse viie kõige tõenäolisema ennustuse hulka sattumise (TOP 5) täpsus olla vähemalt 90%. Sellisel juhul oleks vähemalt üheksal korral kümnest õige vastus rakenduse ennustuste hulgas.

Praeguste tulemuste põhjal ei ole rakendus piisavalt täpne, et seda saaks kasutada seeneliikide kindlaksmääramisel. Rakendusest võib olla siiski abi, kui kasutada seda seente tuvastamiseks koos teiste olemasolevate vahenditega. Töö käigus suudeti tehiseärrvivõrgu mudeli koostamisel mudeli täpsust katsetuste käigus jooksvalt tõsta ning tulemuse põhjal on alust arvata, et rakendusest on võimalik välja arendada seenehuvilistele väga kasulik rakendus.

4.3. Edasiarendamise võimalused

Rakenduse kasutamisel söögi- ja mürkseentel vahet tegemiseks on kasutaja jaoks kõige olulisemal kohal ennustamise täpsus, sest vale ennustuse uskumine võib lõppeda tõsiste tervisehädadega. Rakenduse hetkelist täpsust (testhulgal 37,1 %) võib sellest seisukohast pidada küllaltki madalaks, sest ligikaudne oht inimese tervisele on iga ennustuse korral umbes 60%. Seega on üheks tähtsamaks rakenduse edasiarenduseks on täpsuse tõstmine.

Täpsust aitaksid tõsta andmete juurde kogumine ning tehiseärrvivõrgu mudeli täiustamine ja mahukam treenimine. Mudeli treenimine suure arvutusjõudlusega masinatel või mõnes serveripargis võimaldaks oluliselt suurendada ärrvivõrgu mudeli parameetrite arvu. Sügavate ärrvivõrkude treenimine võib suurendada ka mudeli täpsust [27]. Täpsust suurendaks ka ilmselt

mitme pildi lisamise võimalus. Sellisel juhul saaks arvutada mitme pildi tulemuste keskmise ja kuvada vastavad tõenäosused. Kuna kasutajale on väga oluline teave seene mürgisuse kohta, siis oleks kasulik ka sellise närvivõrgu treenimine ja kasutamine, mis seeneliikide asemel ennustaks liigi mürgisust.

Rakendust võib kasutada ka hariduslikul eesmärgil, näiteks seeneliikide tundma õppimisel. Sellisel juhul on lisaks heale täpsusele oluline ka ennustusega kaasnev informatsioon seeneliigi kohta. Lisaks liigi nimetusele ja mürgisusele aitaksid kaasa ennustusel kuvatavad kvaliteetsed illid ning pikem kirjeldus ennustatud liigi kohta. Isegi kui ennustus ei ole ideaalne, oleks kasutajal võimalus kiirelt võrrelda seent teiste (sarnaste) liikidega ning õppida tundma nende iseloomulike jooni. Hetkel üritab rakendus ennustada 43 erinevat seeneliiki, kuid Eestis leidub rohkem erinevaid seeneliike ning seega on võimalik suurendada ka ennustatavate liikide hulka, kui sellega suudetakse ka jätkuvalt tagada hea täpsus.

Lisaks seeneliikide tuvastamisele, mis on rakenduse peamine funktsioon, on üheks edasiarenduse võimaluseks funktsioonide juurde lisamine. Turu-uuringute ja mükoloogidega koostöö tegemise tulemusena leiab kindlasti tähtsaid funktsioone, mis seenelistele kasulikud oleksid. Funktsioonide ning teiste võimalike keelte lisamisel ja geograafilise haarde suurendamisel saaks rakendusest potentsiaalselt arendada välja kõik-ühes seenelise rakenduse, mis aitaks seenehuvilisi individuaalselt ja hõlbustaks seeneliste vahelist koostööd üle kogu maailma.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli luua mugava kasutajaliidesega mobiilirakendus, mis suudaks tuvastada mitmeid Eesti metsades kasvavaid seeni.

Töö raames loodi Android rakendus, mis proovib tuvastada 43 erinevat Eesti metsades kasvavat seeneliiki. Pildilt liikide tuvastamiseks kasutab rakendus konvolutsioonilist tehismärgivõrgu mudelit, mis samuti loodi ja treeniti töö raames. Rakendus on lihtsa ja arusaadava välimusega ning seda on väga mugav ja kiire kasutada.

Valminud mobiilirakendus võimaldab kasutajal teha pilt telefonikaameraga või valida see telefoni galeriist. Seejärel kasutab rakendus loodud märgivõrgu mudelit, et arvutada välja pildi põhjal võimalike seeneliikide tõenäosused. Rakendus kuvab kasutajale maksimaalselt viie kõige tõenäolisema seeneliigi eestikeelse nimetuse, ladinakeelse nimetuse, mürgisuse ning enamikel juhtudel ka illustreeriva pildi.

Töö tulemusena valminud rakendus ei saavutanud seeneliikide määramisel sellist täpsust, mis lubaks seda kasutada liikide määramisel. Rakendusest võib olla siiski abi, kui kasutada seda seente tuvastamiseks koos teiste olemasolevate vahenditega. Töö käigus suudeti tehismärgivõrgu mudeli koostamisel mudeli täpsust katsetuste käigus jooksvalt tõsta ning tulemused annavad alust arvata, et rakendusest on võimalik välja arendada seenehuvilistele väga kasulik rakendus.

Kasutatud kirjandus

- [1] Royal Botanic Gardens, Kew. State of the World's Fungi, 2018.
https://stateoftheworldsfungi.org/2018/reports/SOTWFungi_2018_Full_Report.pdf
(10.05.2019)
- [2] Yellow Elanor. Mushrooms, Fungi, Toadstools: What's the Difference?, 2015.
<http://www.yellowelanor.com/mushrooms-fungi-toadstools-whats-the-difference/>
(10.05.2019)
- [3] Chang S. T. The World Mushroom Industry: Trends and Technological Development.
International Journal of Medicinal Mushrooms, 2006.
https://www.researchgate.net/publication/247855342_The_World_Mushroom_Industry_Trends_and_Technological_Development (10.05.2019)
- [4] Saar I. Seente asend elupuul ja süstemaatika 1. *Eesti Loodus*, 2015.
http://mukoloogiauhing.ut.ee/sites/default/files/mukoloogiauhing/files/2015_saar_el31.pdf
(10.05.2019)
- [5] Raitviir A. Väike mürkseeneraamat. Maalehe raamat, 2005.
- [6] Keizer G. J. Seente entsüklopeedia, 2006.
- [7] Saar I. Mürkseened ja seenemürgistused Eestis. *Eesti Loodus*, 2010.
<http://eestiloodus.horisont.ee/index.php?id=3201> (10.05.2019)
- [8] Google Play. Mushroom Identify - Automatic picture recognition.
<https://play.google.com/store/apps/details?id=com.pingou.champignouf&hl=en>
(10.05.2019)
- [9] Apple Store. Mushroom Identifier.
<https://itunes.apple.com/us/app/champignouf/id1227854971?ls=1&mt=8> (10.05.2019)
- [10] Google Play. Mushroom identification from photos.
<https://play.google.com/store/apps/details?id=com.mushroom.recogniser> (10.05.2019)
- [11] Google Play. Mushrooms app.
<https://play.google.com/store/apps/details?id=bazinac.aplikacenahouby&hl=en>

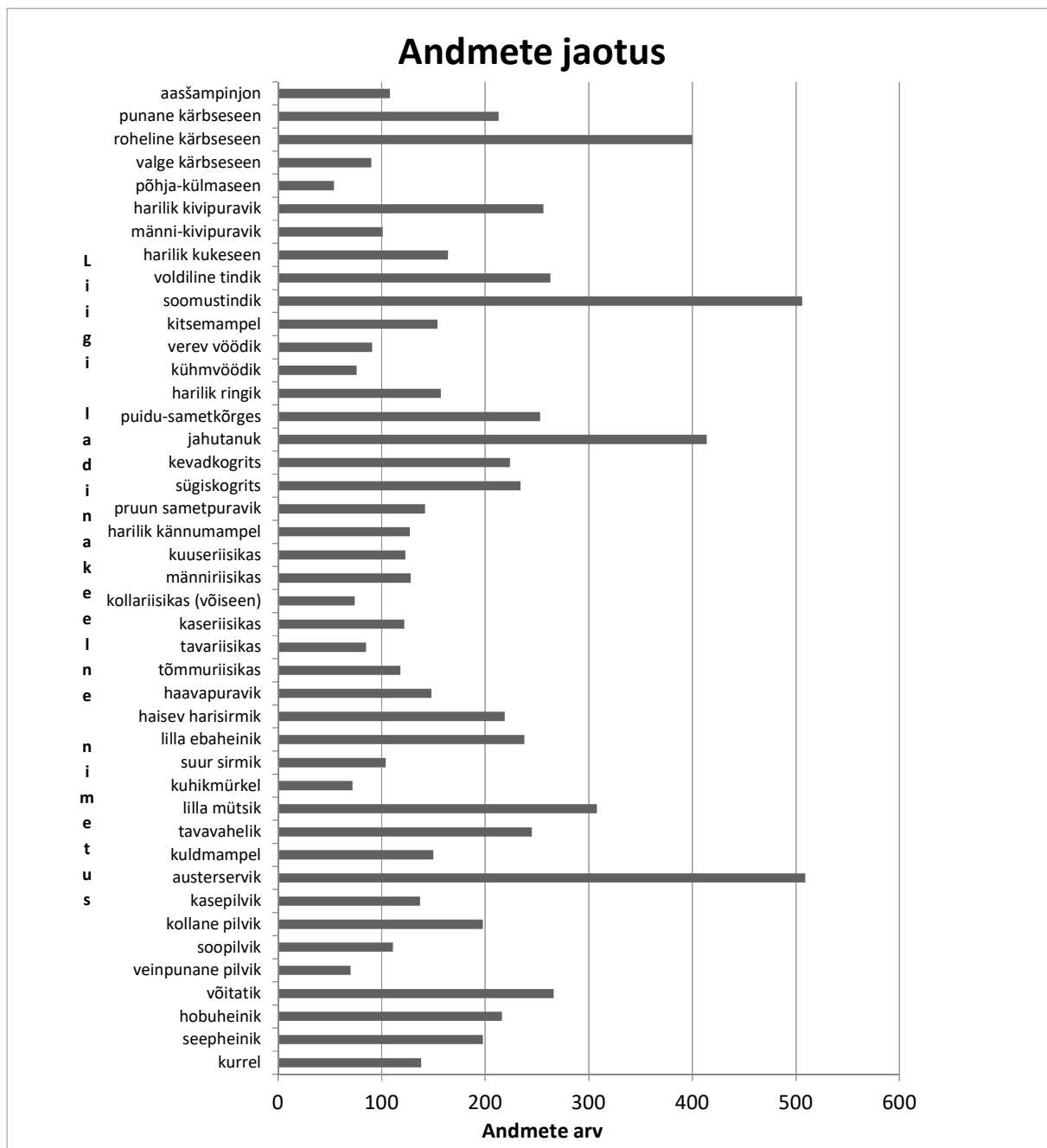
(10.05.2019)

- [12] Physics World. Neural networks, explained, 2018. <https://physicsworld.com/a/neural-networks-explained/> (10.05.2019)
- [13] Hardesty L. Explained: Neural networks. *MIT News*, 2017.
<http://news.mit.edu/2017/explained-neural-networks-deep-learning-0414> (10.05.2019)
- [14] Nielsen M. Using neural nets to recognize handwritten digits, 2018.
<http://neuralnetworksanddeeplearning.com/chap1.html> (10.05.2019)
- [15] The Data Science Blog. A Quick Introduction to Neural Networks, 2016.
<https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/> (10.05.2019)
- [16] Sanjeevi M. Chapter 7: Artificial neural networks with Math, 2017.
<https://medium.com/deep-math-machine-learning-ai/chapter-7-artificial-neural-networks-with-math-bb711169481b> (10.05.2019)
- [17] Hao Z. Activation Functions in Neural Networks, 2017.
https://isaacchanghau.github.io/post/activation_functions/ (10.05.2019)
- [18] Sharma A. Understanding Activation Functions in Neural Networks, 2017.
<https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0> (10.05.2019)
- [19] CS231n Convolutional Neural Networks for Visual Recognition.
<http://cs231n.github.io/convolutional-networks/> (10.05.2019)
- [20] Hao Z. Loss Functions in Neural Networks, 2017.
https://isaacchanghau.github.io/post/loss_functions/ (10.05.2019)
- [21] Backpropagation Algorithms in Artificial Neural Networks, 2018
<https://rubikscode.net/2018/01/22/backpropagation-algorithm-in-artificial-neural-networks/> (10.05.2019)
- [22] Stergiou C., Sigamos D. Neural Networks.
https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html (10.05.2019)
- [23] Raitviir A. Väike söögiseeneraamat. Maalehe raamat, 2006.

- [24] GitHub. Seenetuvastaja. <https://github.com/andry1001/seenetuvastaja> (10.05.2019)
- [25] GitHub. mushroomobser-dataset. <https://github.com/bechtle/mushroomobser-dataset> (10.05.2019)
- [26] Kulkarni T. D., Whitney W., Kohli P., Tenenbaum J. B. Deep Convolutional Inverse Graphics Network, 2015. <https://arxiv.org/pdf/1503.03167.pdf> (10.05.2019)
- [27] Amidi A., Amidi S. The evolution of image classification explained. <https://stanford.edu/~shervine/blog/evolution-image-classification-explained> (10.05.2019)
- [28] Pixabay. <https://pixabay.com/> (10.05.2019)
- [29] Android Developers. Documentation. Activity. <https://developer.android.com/reference/android/app/Activity> (10.05.2019)
- [30] Unruh A. What is the TensorFlow machine intelligence platform?, 2017. <https://opensource.com/article/17/11/intro-tensorflow> (10.05.2019)
- [31] TensorFlow Lite. <https://www.tensorflow.org/lite/guide> (10.05.2019)
- [32] Keras. <https://keras.io/> (10.05.2019)
- [33] Keras documentation. Why use Keras? <https://keras.io/why-use-keras/> (10.05.2019)
- [34] OpenCV. About. <https://opencv.org/about.html> (10.05.2019)
- [35] Guy R., Haase C. Realm Academy. Android: A Developer's History, 2017. <https://academy.realm.io/posts/romain-guy-chet-haase-android-developers-history/> (10.05.2019)
- [36] Android Developers. Android Studio. Meet Android Studio. <https://developer.android.com/studio/intro> (10.05.2019)

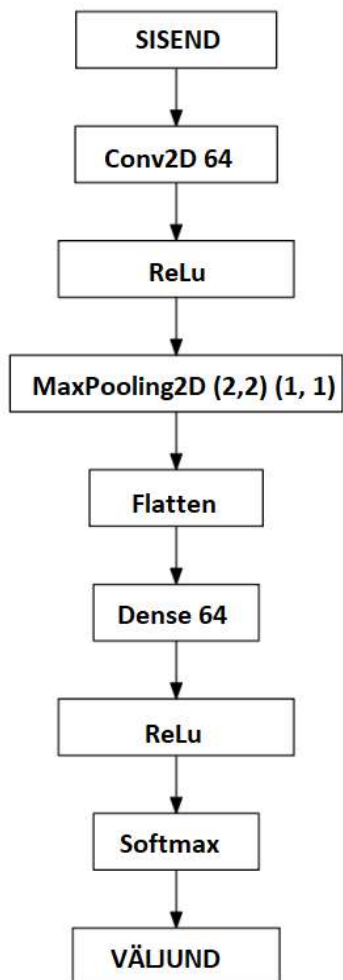
Lisad

1. Andmete rohkuse jaotus üle liikide

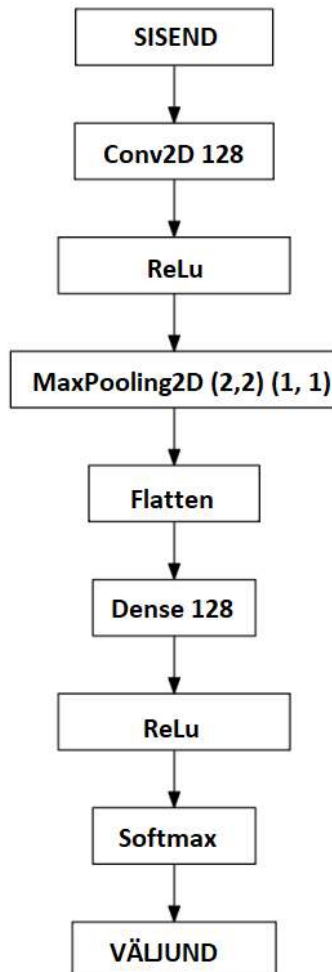


2. Katsetatud mudelite arhitektuurid

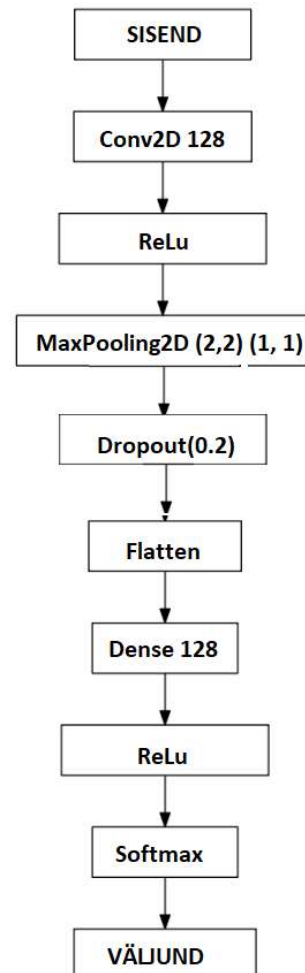
a) Mudeli 1 arhitektuur



b) Mudeli 2 arhitektuur



c) Mudeli 3 arhitektuur



3. Segadusmaatriksi analüüs testhulgal

Liigi nimi	Täpsus	Saagis	Esinemiste arv	Ennustuste arv
aassampinjon	0.67	0.1	20	3
austerservik	0.45	0.37	81	66
haavapuravik	0.21	0.19	16	14
haisev harisirmik	0.49	0.46	39	37
harilik kivi puravik	0.28	0.13	38	18
harilik kuke seen	0.5	0.64	28	36
harilik kännumampel	0.33	0.12	24	9
harilik ringik	0.29	0.5	26	45
hobuheinik	0.23	0.37	30	47
jahutanuk	0.55	0.35	63	40
kasepilvik	0.18	0.21	24	28
kaseriisikas	0.5	0.14	22	6
kevadkogrits	0.67	0.56	32	27
kitsemampel	0	0	28	5
kollane pilvik	0.54	0.84	32	50
kollariisikas (võiseen)	0.25	0.29	17	20
kuhikmürkel	0.23	0.33	9	13
kuldmampel	0.4	0.33	24	20
kurrel	0.22	0.31	13	18
kuuseriisikas	0.83	0.28	18	6
kühmvöödik	0.33	0.1	10	3
lilla ebaheinik	0.67	0.11	36	6
lilla mütsik	0.47	0.32	50	34
männi-kivi puravik	0	0	12	3
männiriisikas	0.37	0.41	17	19
pruun samet puravik	0	0	18	1
puidu-samet kõrges	0.84	0.4	40	19
punane kärbseseen	0.83	0.78	32	30
põhja-külmaseen	0	0	9	1
roheline kärbseseen	0.25	0.56	66	147
seepheinik	0.08	0.2	25	60
soomustindik	0.42	0.87	70	146
soopilvik	0.64	0.43	21	14
suur sirmik	0.38	0.17	18	8
sügiskogrits	0.15	0.12	25	20
tavariisikas	0.2	0.11	9	5
tavavahelik	0.15	0.09	34	20
tõmmuriisikas	0.31	0.5	18	29
valge kärbseseen	0.24	0.73	15	45
veinpunane pilvik	0.75	0.27	11	4
verev vöödik	0.83	0.5	10	6
voldiline tindik	0.26	0.27	37	39
võitatik	0.38	0.38	34	34

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Andry Nõgols,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose Mobiilirakendus seente tuvastamiseks,

mille juhendaja on Sven Aller,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Andry Nõgols

10.05.2019