

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Karl-Erik Kanal**

# Eesti lindude pildiklassifitseerija koostamine eeltreenitud mudeli baasil

**Bakalaureusetöö (9 EAP)**

Juhendaja: Markus Kängsepp, MSc

Tartu 2023

# **Eesti lindude pildiklassifitseerija koostamine eeltreenitud mudeli baasil**

## **Lühikokkuvõte:**

2022. aasta seisuga puudus Eesti lindudele kohandatud mudel, kus pildilise sisendi kaudu oleks võimalik olnud määrata linnuliiki. Töö eesmärk on luua mudel, mis võimaldab algajal Eesti linnuvaatlejal kiiresti kontrollida, keda ta üles pildistas. Selleks koostatakse töö raames uus andmestik, mille peal mudelit treenida. Peale seda õpetatakse eeltreenitud mudel ümber tuvas-tama Eesti linnuliike, kasutades valminud andmestikku ning erinevaid andmete rikastamise meetodeid. Lõpus analüüsitakse valminud mudelit ning tuuakse välja aspekte, mida saaks tulevikus uue mudeli tegemise puhul parandada.

## **Võtmesõnad:**

Klassifitseerija, siirdeõpe, tehisnärvivõrgud, masinõpe, linnud

**CERCS:** P176 Tehisintellekt

# **Creating an Estonian bird image classifier using a pretrained model**

## **Abstract:**

At the time of writing in 2022, an image classification model did not exist for Estonian birds. The work aims to create a classifier for Estonian bird species, allowing beginner Estonian bird hobbyists to check the bird species they captured. A new dataset was created to complete this model. After that, a pretrained image classification model was retrained to classify Estonian birds using the dataset and different data augmentation methods. The completed model is analysed at the end of the thesis, and potential improvements are proposed.

## **Keywords:**

Classifier, transfer learning, artificial neural network, machine learning, birds

**CERCS:** P176 Artificial intelligence

# Sisukord

Sissejuhatus .....	4
1 Teoreetiline taust .....	5
1.1 Praegused võimalused Eesti linnuliikide tuvastamiseks .....	5
1.1.1 Linnumäärajad.....	5
1.1.2 Häältuvastus .....	5
1.1.3 Pildituvastus .....	6
1.2 Andmete rikastamine pilditöötluks .....	6
1.2.1 Andmete rikastamise taust .....	6
1.2.2 Tavalisemad pildi rikastamise võtted .....	7
1.3 Konvolutsioonilised närvivõrgud .....	10
2 Metoodika.....	13
2.1 Andmete kogumine.....	13
2.2 Mudel Inception V3.....	14
2.3 Treenimine.....	15
2.4 Muud proovitud mudelid .....	17
3 Tulemused .....	19
3.1 Valminud mudeli analüüs .....	19
3.2 Mudeli avalikustamine ja väärtuste kaart .....	20
3.3 Mida oleks võinud teha teisiti ja edasised arenguvõimalused .....	21
Kokkuvõte .....	22
Viidatud kirjandus .....	23
Lisad.....	26
I. Andmestike ülesehitus.....	26
II. Klassifitseerimise aruanne .....	28
III. Litsents.....	30

## Sissejuhatus

Olen olnud lapsest saati huvitatud lindudest ning soovisin oma lõputöö teema siduda lindudega. Teema uurimisel selgus, et Eesti lindude kohta puudub selline mudel, kus pildilise sisendi kaudu oleks võimalik määrata linnuliiki. Olen täheldanud, kuidas avalikes gruppides, nagu Linnuhuvilised Facebookis, postitatakse sageli pilt linnust ja küsitakse, kellega tegu on. Loodav mudel võimaldaks algajal linnuvaatlejal kiiresti kontrollida, keda ta üles pildistas ja võiks aidata lahendada seda probleemi.

Töö eesmärk on treenida pildilise sisendiga mudel, mis suudaks tuvastada 50 tavalisemat Eesti linnuliiki. Loodav mudel peaks suutma tuvastada õige liigi vähemalt 75% täpsusega. Töö käigus luuakse ka treeningandmestik ja testandmestik, mille peal hakatakse mudelit ümber treenima ning hindama. Arvestades, et töö jaoks tuleb uus andmestik luua, on 50 linnuliiki mahu poolest mõistlik arv ning 75% täpsus mitmeklassilise mudeli puhul saavutatav eesmärk, mille poole püüelda. Valminud mudelit oleks tulevikus võimalik edasi treenida rohkem liike tuvastama ning seda oleks võimalik integreerida erinevatesse lindudega seotud mobiilirakendustesse või veebilehtedesse.

Töö on jaotatud kolmeks suuremaks peatükiks. Esimeses peatükis kirjeldatakse olemasolevaid viise, kuidas linde saab Eestis määrata, kuidas pilte saab mudeli parema treenimise jaoks muuta ning ka sellest, kuidas konvolutsioonilised närvivõrgud toimivad. Teises peatükis kirjeldatakse töö käigus läbitud etappe nagu andmetike loomine, eeltreenitud mudeli valimine ja selle ümberõpetamine tuvastamiseks Eesti linde. Lisaks põhjendatakse protsessi käigus tehtud valikuid ning kasutatud tehnoloogiaid. Kolmandas peatükis analüüsitakse tehtud töö tulemusi ja tehakse järeldusi eesmärkide saavutamise ning loodud mudeli täpsuse kohta.

# 1 Teoreetiline taust

Selles peatükis tutvustatakse praeguseid lindude tuvastamise võimalusi Eestis ning kirjeldatakse tavalisemaid piltide töötlemise protseduure, mida pildilise mudeli andmete puhul rakendatakse. Lisaks tutvustatakse ka pildiõppeks kasutatavate konvolutsiooniliste närvivõrkude ülesehitust.

## 1.1 Praegused võimalused Eesti linnuliikide tuvastamiseks

19.08.2022 seisuga kuulub Eesti lindude nimestikku kokku 400 loodusliku päritoluga meil või naabermaades püsiva asurkonna moodustanud linnuliiki [1]. Nende tuvastamiseks on võimalikest abivahenditest lindude määramisel praegu peamiselt kolm varianti: linnumäärajad, häältuvastusrakendused ja pildituvastust pakkuvad rakendused.

### 1.1.1 Linnumäärajad

Linnuvaatlejate ustavaks abivahendiks on kaua aega olnud välimäärajad. Tänapäeval on saada-val mitmed erinevad paberkujul määrajad, millest tuntuim on Svenssoni jt [2] loodud “Euroopa ja Vahemere lindude välimääraja”. Kuigi leidub juba mitmeid mobiilseid määrajaid, siis on määramisraamatud ikkagi veel linnuhuviliste seas olulisel kohal.

Mobiilirakendustest on olemas konkreetselt Eesti lindude jaoks võimalik Android<sup>1</sup> telefonile tõmmata rakendus Eesti Linnud<sup>2</sup>. Selles rakenduses on võimalik otsida linde tunnuste järgi ning ka lugeda infot lindude kohta ning kuulata ka nende häälotsusi. Ingliskeelse variandina on võimalik osta endale eelnimetatud Svenssoni jt [2] määraja põhjal loodud rakendus Collins Bird Guide<sup>3</sup>.

### 1.1.2 Häältuvastus

Eesti linnuhäälte tuvastamiseks on loodud Proekspert AS poolt rakendus nimega Siuts<sup>4</sup>. Rakendusega on võimalik salvestada loodusest linnuhäält, mille siis mobiilirakendus ära tuvastab ning pakub 5 võimalikku vastet tõenäosustega. Rein Kuresoo on täheldanud, et rakendust kasutavad õpilased on julgemad vastama küsimustele laulva liigi kohta, kuid nendib siiski, et arenguruumi rakendusel veel on [3]. Ingliskeelsetest variantidest on hea valik Cornelli ülikooli ja Chemnitz'i tehnikaülikooli koostööprojektina valminud rakendus BirdNET<sup>5</sup>.

---

<sup>1</sup> Mobiiltelefoni operatsioonisüsteem

<sup>2</sup> <https://play.google.com/store/apps/details?id=ee.eoy.key.mobile.eestilinnud> Vaadatud 03.05.2023

<sup>3</sup> <https://play.google.com/store/apps/details?id=com.natureguides.birdguide> Vaadatud 03.05.2023

<sup>4</sup> <https://play.google.com/store/apps/details?id=ee.proekspert.siuts&hl=et&gl=US> Vaadatud 03.05.2023

<sup>5</sup> [https://play.google.com/store/apps/details?id=de.tu\\_chemnitz.mi.kahst.birdnet&hl=et&gl=US](https://play.google.com/store/apps/details?id=de.tu_chemnitz.mi.kahst.birdnet&hl=et&gl=US) Vaadatud 03.05.2023

### 1.1.3 Pildituvastus

Pildituvastamise võimalustega rakendustest on kättesaadaval ainult välismaised variandid. Kõige võimsam saadaval olev rakendus on Picture Bird<sup>6</sup>, millega on võimalik tuvastada üle 10 tuhande liigi, kuid rakendus on samas tasuline. Veel üks tuntum variant on Cornelli ornitoloogialabori poolt välja töötatud rakendus Merlin Bird ID<sup>7</sup>, kuid rakendus keskendub põhiliselt Põhja-Ameerika lindudele. Võimalik on ka kasutada Google'i loodud Google Lensi<sup>8</sup>, mis on muidu päris võimekas linnumääraja, kuid kuna see on disainitud tuvastama ka mitmeid muid asju pildi pealt, siis ei pruugi see vahepeal pildi pealt lindu tuvastada või üles leida.

## 1.2 Andmete rikastamine pilditöötluseks

Selleks, et lindu pildi pealt tuvastavad rakendused häid tulemusi tagastaksid, on vaja rakenduses peituvat mudelit treenida suure hulga andmete peal. Tihti võib juhtuda nii, et olemasolevatest andmetest jääb treenimise jaoks väheks, seega tuleb pilte mehhaaniliselt juurde toota. Järgnevalt kirjeldatakse võimalikke viise, kuidas neid saab lisaks tekitada.

### 1.2.1 Andmete rikastamise taust

Andmete rikastamine (ingl *data augmentation*) on andmehulga täiendamine uute sarnaste andmetega, mis on saadud olemasolevate andmete muutmise teel. Piltide puhul tähendab see erinevaid toiminguid nagu pööramine, lõikamine, värvi muutmine, udustamine, kombineerimine jne. Suurem eesmärk selle protsessi puhul on üldisemate andmete juurde tekitamine, mis omakorda vähendab masinõppe mudeli ülesobitamist<sup>9</sup>.

Lemley jt [4] on liigitanud andmete rikastamise võtted kaheks: juhendatud ja juhendamiseta rikastamiseks (ingl *supervised/unsupervised augmentation*). Juhendamiseta rikastamine on tavaliste pildimuutmisvõtete kasutamine sõltumata märgendist (ingl *label*) nagu loetletud varasemalt. Juhendamisega rikastamine on näiteks kahe sama märgendiga pildi kombineerimine üheks uueks sama märgendiga pildiks. Juhendamine on vajalik selleks, et veenduda, kas uus loodud pilt tõepoolest sobitub märgendiga. Selle töö puhul kasutatakse juhendamiseta rikastamise viisi, sest piltide kombineerimine uueks pildiks on keerukas ning on ebakindel, kas see mudeli tulemust parandaks.

---

<sup>6</sup> <https://picturebirdai.com/> Vaadatud 03.05.2023

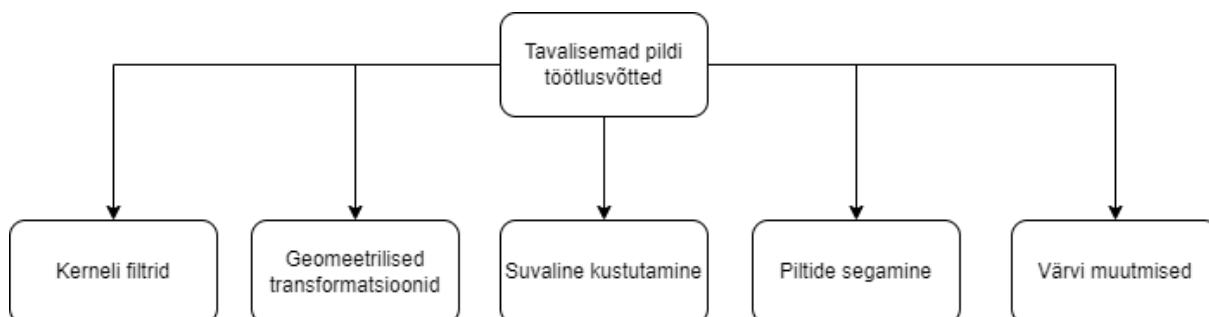
<sup>7</sup> <https://merlin.allaboutbirds.org/photo-id/> Vaadatud 03.05.2023

<sup>8</sup> <https://play.google.com/store/apps/details?id=com.google.ar.lens&hl=et&gl=US> Vaadatud 03.05.2023

<sup>9</sup> Mudeli liigne kohandamine treeningandmetele, mille tulemusena mudel ei ennusta uutel andmetel piisavalt hästi.

### 1.2.2 Tavalisemad pildi rikastamise võtted

Shorten jt [5] on oma artiklis loonud taksonoomia, mis liigitab piltide rikastamise võtted erinevatesse kategooriatesse. Selle järgi on tavalisemad võtted liigitatud viide erinevasse alamkategooriasse (vt joonis 1).



Joonis 1. Pildi töötlusvõtete jaotus. Adapteeritud artiklist [5].

Järgnevalt kirjeldame selle joonise järgi erinevaid piltide rikastamise viise.

#### Geomeetrilised transformatsioonid

Geomeetriliste transformatsioonide alla kuuluvad tehnikad nagu pildi nurga alla seadmine, skaaleerimine, ümberpööramine või nihutamine (vt joonis 2). Neid protsesse on lihtne teostada ning saab ära kasutada keerukamate algoritmide loomiseks. Yang jt [6] sõnul peetakse transformatsioone üldjuhul märgendeid säilitavaks protsessiks, sest nende teostamisel muutuvad vaid tunnuste asukohad. Siiski tasub tähelepanelik olla, et mõned tunnused pildi raamidest ära ei kaoks.



Joonis 2. Näide pildist enne ja pärast geomeetriliste transformatsioonide rakendamist.

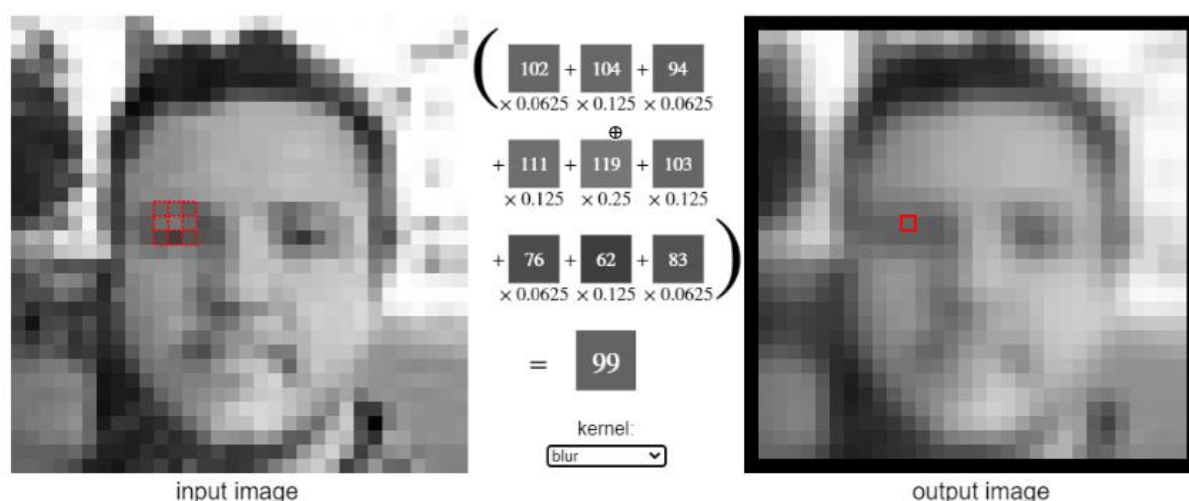
Lisaks tasub Shorteni jt [5] sõnul vaadata ülesande konteksti, sest näiteks *kass vs koer* pildi tuvastamise ülesannete puhul on need operatsioonid ohutud, kuid numbrite tuvastamisel nagu 6 vs 9 puhul võib märgend kaotsi minna.

## Värvi muutmised

Pildi värvi muutmised on tavaline praktika andmete rikastamise puhul. Kuna pilt on arvuti jaoks üks suur arvudest koosnev maatriks, siis on võimalik teke nagu OpenCV<sup>10</sup> kasutades pildi piksleid muuta soovitud tulemuse saavutamiseks. Pikslite manipuleerimisega saab näiteks pilti muuta kontrastsemaks või muuta pildi värvitooni. Yang jt [6] nendivad, et kuigi värviga mängimine säilitab pildi esialgsed mõõtmed ja kuju, siis värviliste tunnuste puhul võivad toimingud mudeli segadusse ajada. Seega, ülesannetega, kus mingi objekti värv on oluline tunnus, tasub ettevaatlik olla.

## Kerneli filtrid

Kerneli filtreid kasutatakse pilditöötles pildi udustamiseks, teravustamiseks ja ka äärte tuvastamiseks. Meetod seisneb selles, et võetakse  $n \times n$  maatriks, kus sees on filter ning nii-öelda libistatakse see üle pildi. Sisuliselt korrutatakse pildi sees olevad numbrilised väärtused filtri väärtustega (vt joonis 3).



Joonis 3. Näide kerneliga pildi udustamisest. Väljavõte lehelt [7].

Tulemusena saab luua andmeid, mis aitavad mudelil paremini toime tulla udusemate piltidega või teravustada pilte selleks, et tunnused paremini nähtavad oleksid.

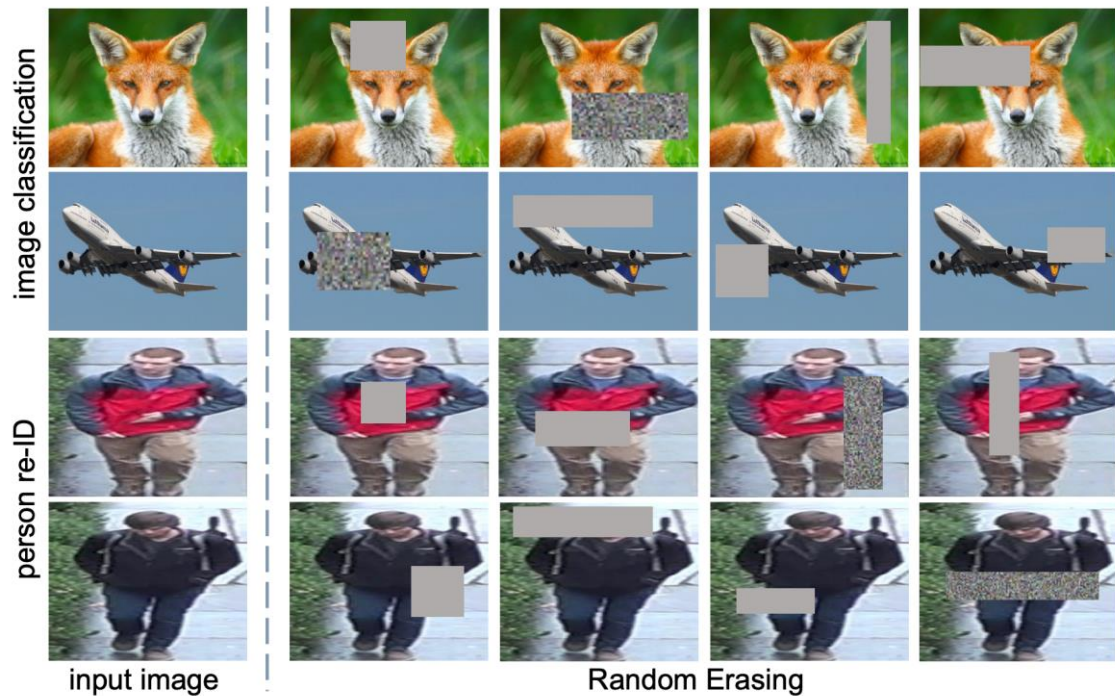
## Suvaline kustutamine

Suvaline kustutamine on Zhong jt [8] poolt välja arendatud lahendus, mis seisneb pildi seest suvalise ruudu välja lõikamises ning asendamises suvalist värvi pikslitega (vt joonis 4). Selle

<sup>10</sup> <https://docs.opencv.org/4.x/index.html> Vaadatud 08.05.2022



protsessi tulemusena tekitatakse andmed, mis aitavad mudelil paremini toime tulla oklusiooniga<sup>11</sup>. Lisaks vähendab see ka mudeli puhul ülesobitamist.

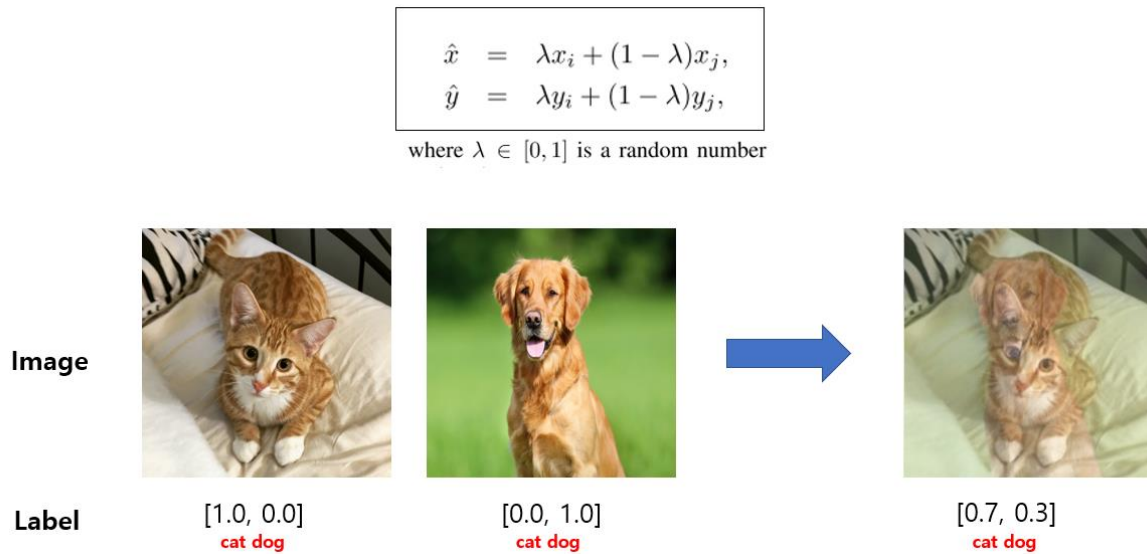


Joonis 4. Näide suvalisest kustutamisest [8].

## Piltide segamine

Piltide omavaheline segamine andmete rikastamiseks on idee, mis seisneb kahe või enama pildi kombineerimises, et seeläbi luua uusi treeningandmeid. Shorten jt [5] on pakkunud võimalikuks seletuseks, miks see strateegia töötab, et suurenenud andmehulgas tulevad rohkem esile robustsed tunnused nagu jooned ja ääred. See tähendab, et mudel saab pärast treenimist paremini uute andmetega hakkama, sest ebamääraselt moodustatud pildid muudavad treeningandmeid üldisemaks, mis läbi väheneb mudelil ülesobitamise oht. Ühed tuntumad pildi segamise strateegiad on mixup [9] ja CutMix [10] ning nende põhjal on edasi arendatud palju muid piltide kombineerimise meetodeid. Mixupi idee on võtta kaks pilti ning suvalise kaalu järgi määrata pildi nähtavus ning seejärel panna üks pilt teise peale (vt joonis 5). CutMix aga toimib nii, et ühest pildist lõigatakse suvalise suurusega ruut välja ning asendatakse osaga teisest pildist (vt joonis 6).

<sup>11</sup> Oklusioon masinõppes tähendab juhte, kus meid huvitav objekt pole täielikult nähtav.



Joonis 5. Näide mixupi abil rikastamisest [11].



Joonis 6. Näide CutMixi abil loodud pildist. Väljavõte artiklist [10].

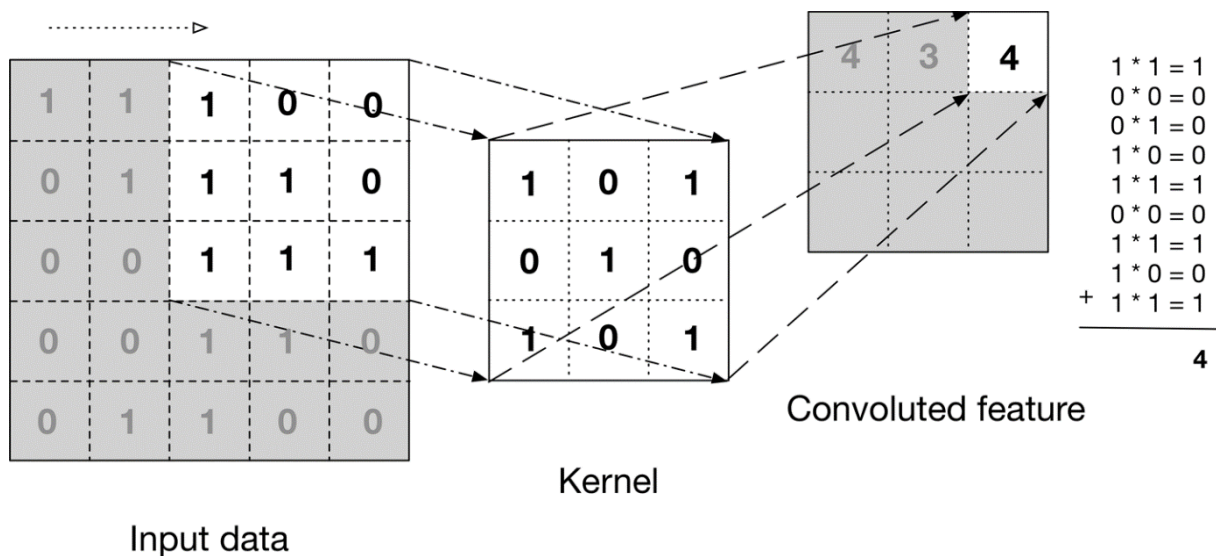
Yun jt [10] leidsid oma uurimuses, et nende poolt välja pakutud CutMix algoritmi kasutades saadud tulemused on klassifitseerimisel paremad kui Mixupi kasutades. Lisaks toimis see ka paremini objekti lokaliseerimise ülesandel, kus leiti, et Mixupi üldisuse tõttu keskendus mudel väiksemale osale pildist. Siit võib järeldada, et piltide segamise algoritmid pidevalt arenevad ning täiustavad teineteist, seega tasub jälgida, millised uued algoritmid välja tulevad.

### 1.3 Konvolutsioonilised närvivõrgud

Pildiliste mudelite treenimiseks kasutatakse konvolutsioonilisi närvivõrke. Laiemalt võime jaotada konvolutsioonilise närvivõrgu kaheks osaks: tunnuste tundma õppimise osaks ning klassifitseerimise osaks. Peatüki tekst põhineb Mandali [12] ning Saha [13] loodud juhendite põhjal.

Tunnuste õppimiseks kasutavad närvivõrgud filtreid ehk kerneleid. Kuna pilt on arvuti jaoks kui üks suur maatriks numbraid, siis on võimalik defineerida omakorda väiksem maatriks, mis hakkab üle pildi sammhaaval liikuma ja tuvastama tunnuseid. See tähendab, et kernel liigub  $n$  sammu kaupa paremale kuni jõuab pildi ääreni ning peale seda liigub  $n$  sammu alla ja hakkab

uuesti paremale liikuma. Kerneli ja valitud pildi alal olevad arvud korrutatakse elementhaaval omavahel vastavate arvudega ning liidetakse kokku. Saadud tulemus sisestatakse uude maatriksisse ehk tunnuste kaarti (vt joonis 7).

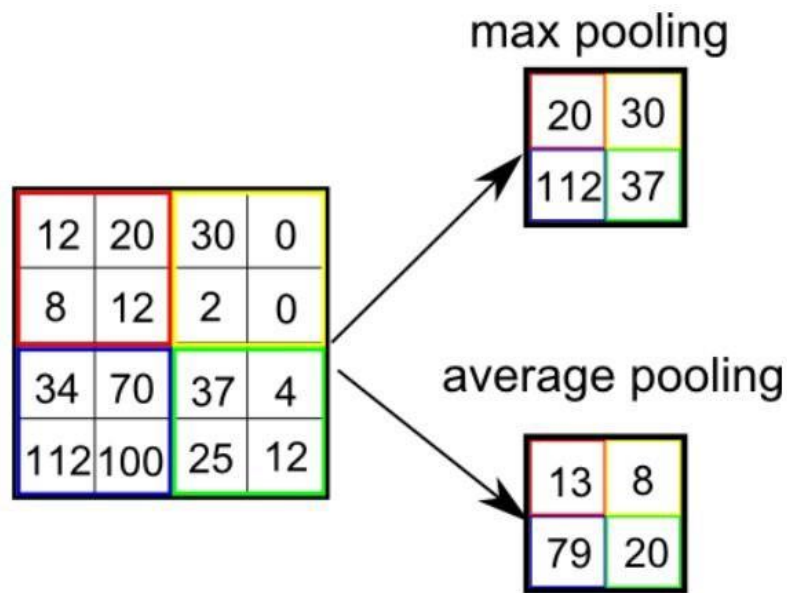


Joonis 7. Tunnuse leidmine pildi pealt kasutades kernelit [12].

Saadud tunnuste kaart saadetakse edasi järgmisele konvolutsioonilisele kihile. Närvivõrgu alguses tuvastavad filtrid lihtsamaid tunnuseid nagu jooni või nurki, mis närvivõrgus edasi minnes muutuvad keerulisemate tunnuste tuvastamise, nagu tervete objektide, filtriteks. Keerulisemate objektide tuvastamiseks kasutatakse ära eelnevaid filtreid. Näiteks lindu tuvastava närvivõrgu puhul saab kombineerida nokka tuvastava ja silma tuvastava filtri kokku pead tuvastavaks filtriks.

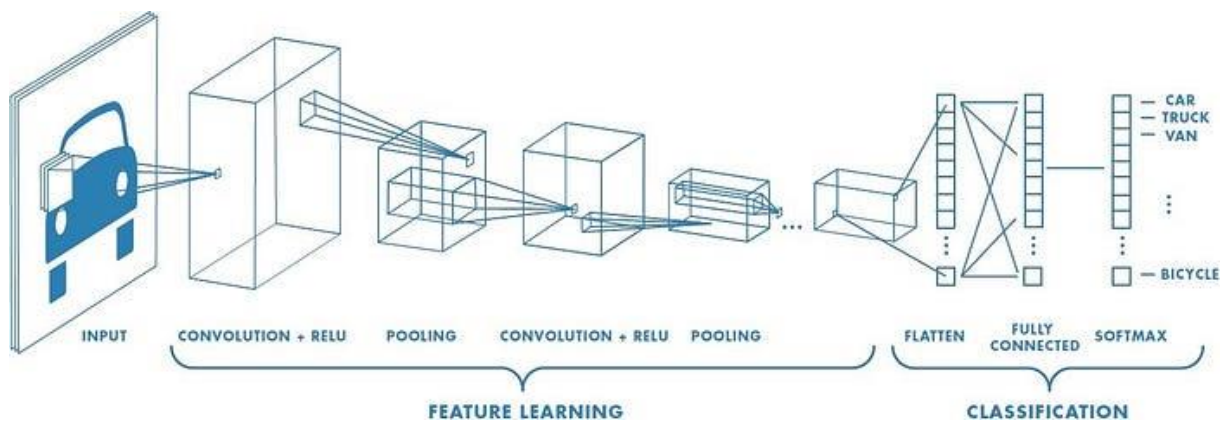
Arvutusliku mahu vähendamiseks kasutatakse konvolutsioonilistes närvivõrkudes ka ReLU [14] funktsiooni ehk mittenegatiivset lineaarfunktsiooni. ReLU funktsioon asendab tunnuste kaardis kõik negatiivsed väärtused nullidega, mis muudab järgnevad arvutused tunnускаardiga kiiremaks.

Lisaks konvolutsioonilisele kihile, mis sisaldab filtreid, kasutatakse närvivõrkudes ka ahenduskihte (ingl *pooling layer*). Nende eesmärk on tunnuste kaardi suurust vähendada, et seeläbi arvuti jaoks arvutuslikku mahtu vähendada. Tunnuskaardi ahendamiseks kasutatakse kerneleid, mis töötavad sarnaselt konvolutsioonilises kihis kasutatavatele filtritele. Tavalisemad kaks ahenduskihi tüüpi on maksimumi võtmine ja keskmise võtmine (vt joonis 8), kus filter valib välja kattuvalt alalt vastavalt, kas maksimum väärtuse või kõigi arvude keskmise väärtuse ning paigutab need uude tunnускаarti.



Joonis 8. Maksimumi ja keskmise ahenduskihi toimimine [12].

Peale konvolutsiooniliste kihtide läbimist on vaja tunnuskaart kuidagi klassifitseerida. Selleks muudetakse tunnuste kaart ühedimensionaalseks järjendiks, mis antakse edasi täissidusatele (ingl *dense*) kihtidele. Viimase kihina konvolutsioonilises närvivõrgus on softmax-kiht, mis tagastab tõenäosused iga võimaliku klassi kohta, mis näitavad kui tõenäoline on, et pilt kuulub nimetatud klassi.



Joonis 9. Tavaline konvolutsioonilise närvivõrgu struktuur [13].

Kõigi eelnevate komponentide kokkupanemisel saamegi kätte klassikalise konvolutsioonilise närvivõrgu struktuuri, mis on valmis pildi pealt objekte tuvastama (vt joonis 9).

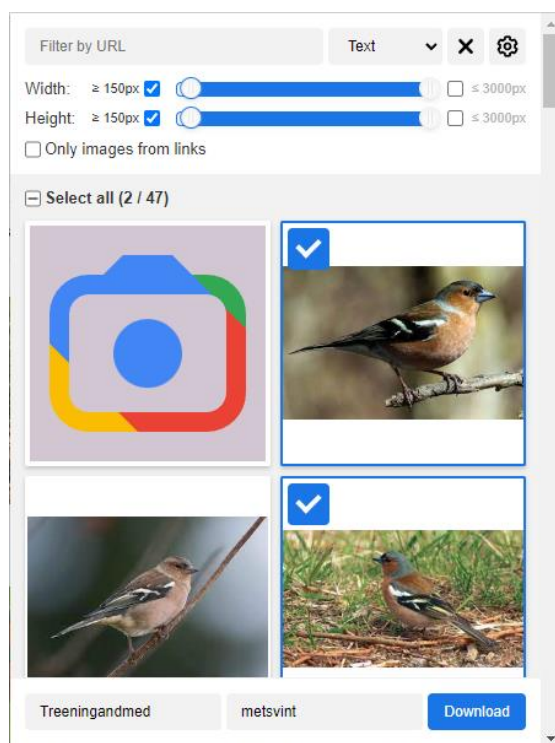
## 2 Metoodika

Selles peatükis kirjeldatakse mudeli treenimise jaoks andmestike loomist, treenimise jaoks vaadatud mudeleid ning lõpliku mudeli treenimise protsessi, kasutades siirdeõpet.

### 2.1 Andmete kogumine

Kuigi internetis leidub mitmeid avalikke andmestikke lindude klassifitseerimise jaoks, siis valdav osa neist keskenduvad piirkondadele, mille kattuvus Eesti linnuliikidega on väike. Seega mudeli edukaks treenimiseks oli vaja luua eraldi treening- ja testandmestik. Aja ja ressursi kokuhoidmiseks sai eesmärgiks seatud koguda pilte 50 tavalisema Eestis kohatava liigi kohta, tuginedes autori kogemusele. Eelkõige valiti tavalisemad aia- ja metsalinnud, kuid ka mõned veelinnud, sest need võiksid olla liigid, keda igapäevaselt on võimalik sagedamini näha. Täpsemalt valituks osutunud liikidega on võimalik tutvuda lisas I.

Piltide kogumiseks kasutati veebilehitseja Google Chrome'i laiendust Image Downloader<sup>12</sup>, mis võimaldab veebilehel olevaid pilte korraga alla laadida. Lisaks saab määrata alla laetavate piltide suurused, anda neile ühine nimi ja määrata kaust, kuhu pildid salvestatakse (vt joonis 10).



Joonis 10. Näide Image Downloaderi kasutusest.

<sup>12</sup> <https://chrome.google.com/webstore/detail/image-downloader/cnpniohnpfhjhjhaiiggeabnkhjhpaldj> Vaadatud 08.05.2023



Piltide hankimiseks kasutati Google otsingu tulemusel saadud piltide lahtrit. Kuna Image Downloader tõmbab pilte otse lehelt, siis saadud pildid on vähendatud suurus, mitte oma algsuurus. Tulenevalt sellest jäeti alles ainult pildid, mis on vähemalt 150 x 150 pikslit suured. See suurus sai ka hiljem mudeli sisendi suuruseks. Pilte üritati valida selliselt, et linnud oleksid võimalikult erinevate nurkade all, kuid samas piisavalt hästi näha, et mudel oskaks treenimisel leida liigile omaseid tunnuseid. Juhtudel, kus liigi emaslind ja isaslind on selgelt eristatavad, nagu näiteks metsvint või koduvarblane, prooviti võimalikult võrdne arv pilte mõlemast soost alla laadida. Lisaks tõmmati andmestikku ka pilte lendavatest lindudest ja noorlindudest. Näiteks erinevate röövlindude puhul on oluline, et mudel suudaks ka lendavaid linde tuvastada, sest tihti peale näeb neid just lendamas ning tiivaaluse järgi on neid hea määrata.

Peale piltide tõmbamist tuli andmestikud märgendada ning selleks kasutati Microsofti rakendust Excel<sup>13</sup>. Andmestikku loodi neli erinevat tulp: linnu eestikeelne nimi, ladinakeelne nimi, ingliskeelne nimi ning indeksi tulp, et järke pidada, mitmenda pildi juures ollakse. Testandmestikus märgendati ära ka linnu sugu, kus seda määrata oli võimalik. Kuna lõplik mudel linnu sugu ei ennusta, siis ajalistel kaalutlustel jäeti ära soo märgendamine treeningandmestikus.

Kokku laeti andmestiku tegemisel alla 6993 pilti, millest 5926 pilti läksid treeningandmestikku ning 1067 pilti testandmestikku. Treeningandmestikust omakorda eraldati 20% piltidest valideerimisandmestikku. Lõplik jaotus on seega 5926 pilti treeninguks, 1186 valideerimiseks ja 1067 testimiseks. Liigiti piltide jaotusega on võimalik tutvuda lisas I, tabel 4. Kuna andmestikku alla laetud piltide puhul pole autorite luba küsitud, siis andmestikku avalikuks teha ei saa.

## 2.2 Mudel Inception V3

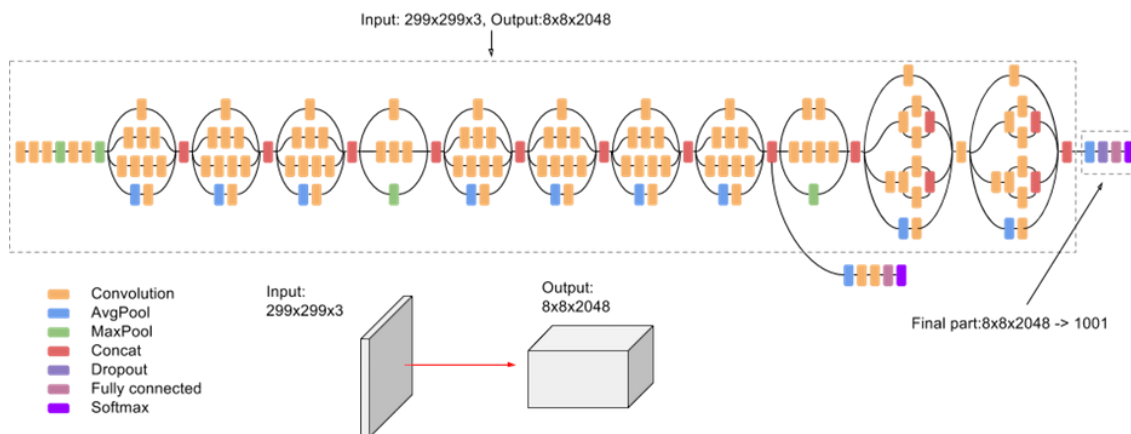
Inception V3 on Szegedy jt [15] poolt väljamõeldud mudeli arhitektuur (vt joonis 11), mis on tõhusam ja täpsam edasiarendus oma samanimelistest eelkäijatest Inception V1 [16] ja Inception V2 [17]. Treenituna ImageNet [18] andmestikul saavutas mudel parema top-5<sup>14</sup> ja top-1 vea kui varasemalt loodud mudelid nagu GoogLeNet [16] ja PReLU [19]. Lisaks on mudel näidanud häid tulemusi madalamate resolutsioonidega piltidel. Näiteks resolutsioonil 79 x 79 leidsid Szegedy jt [15], et mudel saavutas top-1<sup>15</sup> täpsuse (ingl *accuracy*) 75.2%. Seega suudab mudel ka hakkama saada pildi pealt väiksemate objektide tuvastamisega.

---

<sup>13</sup> <https://www.microsoft.com/en-us/microsoft-365/excel> Vaadatud 08.05.2023

<sup>14</sup> Protsent, mis näitab, mitu korda on õige märgend puudunud mudeli 5 kõige tõenäoslikuma pakkumise seast.

<sup>15</sup> Protsent, mis näitab, mitu korda on õige märgend olnud mudeli kõige tõenäoslikuma pakkumise seas.



Joonis 11. Inception V3 arhitektuur [20].

Mudel osutus valituks, tuginedes eelnevalt välja toodud täpsustele klassifitseerimisülesannete puhul. Väiksemate objektide hea tuvastusvõime pildilt aitab kaasa linnu määramisel kasutatavate tunnuste tuvastamisel. Lisaks rääkis mudeli kasuks otsustamise poolt ka see, et mudelit on võimalik mugavalt sisse laadida, kasutades Keras rakendusliidest. Mudelit on võimalik sisse laadida nii eeltreenitud ImageNeti kaaludega kui ka ilma [21]. ImageNeti kaaludel mudeli kasutamine kiirendab mudeli treenimist ja hea täpsuse saavutamist, seega linnuklassifitseerija puhul ka seda võimalust kasutati.

## 2.3 Treenimine

Mudeli siirdeõppel (ingl *transfer learning*) on 4 põhilist etappi, mida tavaliselt läbitakse [22]:

1. Alusmudeli sisselaadimine
2. Alusmudeli kihtide külmutamine, et varasemalt õpitud teave treenimisel ei kaoks
3. Uute ennustavate kihtide lisamine baasmudeli külmutatud kihtide peale
4. Loodud uute kihtide treenimine

Selleks, et väljavalitud InceptionV3 ümber õpetada Eesti linde tuvastama, loodi mudelile uued ennustavad kihid. Kuna autoril puudus varasem teadmine, kuidas head ennustavat kihti luua, siis võeti koostamisel eeskuju Irla [23] poolt koostatud kassi ja koera klassifitseerijast ning kohandati kihid ümber mitmeklassilise mudeli jaoks (vt tabel 1). Mudeli optimeerijaks valiti mitmeklassilistel ülesannetel tavaliselt kasutatav RMSProp [24] õppimiskiirusega 0,0001. Mudeli kaofunktsiooniks valiti kategooriline ristentroopia, mida kasutatakse mitmeklassiliste ülesannete puhul kauguse mõõtjana mudeli pakutud tõenäosuste ja tegelike tõeväärtuste vahel. Hinnatavaks meetrikuks valiti täpsus (ingl *accuracy*).

Tabel 1. Mudeli ennustava kihi struktuur

Kiht	Ühikuid	Aktiveerimisfunktsioon
Silumiskiht ( <i>Flatten</i> )	-	-
Täissidus kiht ( <i>Dense</i> )	1024	ReLU
Väljajätku kiht ( <i>Dropout</i> )	0.2	-
Täissidus kiht ( <i>Dense</i> )	50	Sigmoid
Täissidus kiht ( <i>Dense</i> )	50	Softmax

Treenimiseks ja andmete ettevalmistamiseks kasutati Google'i poolt loodud keskkonda Google Colab<sup>16</sup>. Keskkonna kasutamise puhul kohaliku arvuti asemel on mõned eelised. Esiteks on võimalik hoiustada andmeid pilves, mistõttu on vähem ohtu andmete kaduma minemisel ning tööle saab ligi mitmest arvutist. Teiseks on võimalik mudeli treenimiseks kasutada Google'i enda GPU-sid (*Graphics Processing Unit*), mis kiirendavad märgatavalt mudeli treenimist. Töö tegemiseks kasutati Pythoni teegi Jupyter<sup>17</sup> märkmiku, mis oli üles laetud Colabi keskkonda.

Treenimiseks kasutatavate piltide ettevalmistamise jaoks kasutati Keras klassi *ImageDataGenerator*<sup>18</sup>, mis viib pildid mudeli jaoks sobivasse mõõtu ning ka normaliseerib<sup>19</sup> need. Lisaks on võimalik klassi puhul parameetritena määrata erinevaid andmete rikastamise funktsionaalsusi, mis on põhiliselt seotud pildi nihutamise ja pööramisega. Selle mudeli puhul said valitud treeningandmestiku rikastamiseks kõik funktsioonid peale ereduse muutmise, selleks, et mudel suudaks tuvastada paremini linnu värviga seotud tunnuseid. Valideerimis- ja testandmestike puhul rakendati ainult piltide normaliseerimist.

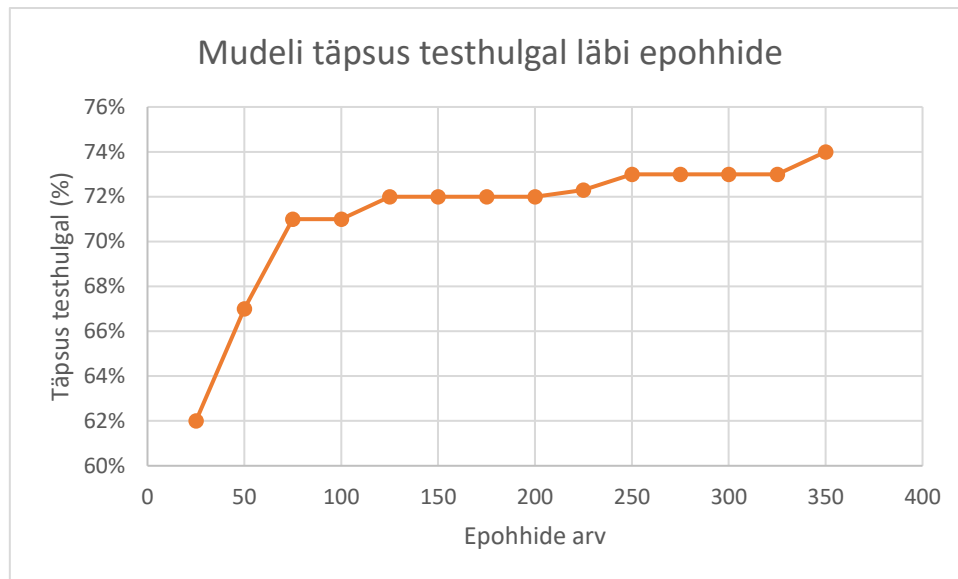
<sup>16</sup> <https://colab.research.google.com/> Vaadatud 08.05.2023

<sup>17</sup> <https://jupyter.org/> Vaadatud 08.05.2023

<sup>18</sup> [https://www.tensorflow.org/api\\_docs/python/tf/keras/preprocessing/image/ImageDataGenerator](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator) Vaadatud 08.05.2023

<sup>19</sup> Masinõppes andmete viimine ühele skaalale. Piltide puhul pikslite RGB väärtuste viimine 0 ja 1 vahemikku.





Joonis 12. Mudeli täpsus testhulgal iga 25 epohhi järel.

Mudelit treeniti 25 epohhi kaupa ning iga treenimistsükli järel muudeti ImageDataGenerator objektis funktsioonide väärtusi, selleks et mudel pidevalt uusi andmeid näeks. Treenimise ajal jälgiti meetrikuid nagu kadu ja valideerimise täpsust, et vaadata, kas mudel jätkab edasi arenemist või mitte. Lisaks jälgiti ka meetrikute põhjal seda, kas on tekkinud oht, et mudel on üle sobitama hakanud või mitte. Treenimine jätkus kuniks täpsus valideerimishulgal enam ei suurenenud, mis saavutati 350 epohhiga. Iga 25 epohhi järel vaadati ka, kuidas mudel hakkama saab testhulgal (vt joonis 12). Colabi keskkonnas treenis mudel tulemuse saavutamiseni kokku ligikaudu seitse tundi. Täpsemalt saab mudeli saavutatud tulemuste kohta lugeda järgmises peatükis.

## 2.4 Muud proovitud mudelid

Lisaks siirdeõppele sai ka proovitud ise luua mõni konvolutsiooniline närvivõrk ning panna see loodud andmestikul treenima. Närvivõrgud loodi tavalisema konvolutsioonilise närvivõrgu struktuuri baasil, kasutades iga kord erinev arv konvolutsioonilisi kihte, milles defineeriti erineva suurusega filtrite arvu ning kerneleid, erinev arv ahenduskihte ning ka plokknormaliseerimist (ingl *batch normalization*) ja ka väljajätku (ingl *dropout*) kihte. Proovitud närvivõrkude struktuuri lõpus oli ka erinev arv täissidusaid (ingl *dense*) kihte, millest viimane kasutas aktiveerimisfunktsioonina softmaxi.

Võrdluseks sai ka mõnele närvivõrgule lisatud algusesse andmete rikastamise kiht, mis kasutas erinevaid võtteid nagu roteerimine või suurendamine, et erinevaid andmeid luua. Mõlemal juhul loodud närvivõrgud loodetud tulemust ei saavutanud. Ilma andmete rikastamiseta hakkasid närvivõrgud kiirelt üle sobitama ning koos rikastamisega jäid nad kinni treenimishulgal umbes 2%

täpsuse juurde ehk täpsus ei läinud juhuslikust ennustamisest paremaks. Halbade tulemuste tõttu otsustati mudeli loomisel närvivõrkude ise koostamise asemel siirdeõppe variandi kasuks.

Ühe tõsisema kandidaadina sai proovitud ka siirdeõppeks kasutada Koopmani loodud UK bird classifier mudelit [25], mis on ise tehtud InceptionV3 baasil, kasutades siirdeõpet. Mudel suudab tuvastada pildi pealt 53 erinevat Inglismaa linnuliiki, mis suuremalt jaolt kattuvad Eestis kohatavate linnuliikidega. Seetõttu oleks mudel olnud idee poolest heaks baasiks Eesti linde tuvastava mudeli tegemiseks. Probleemiks muutus aga see, et mudel oli loodud vanema TensorFlow<sup>20</sup> versiooniga, mistõttu paljud funktsionaalsused mudeli käsitlemiseks olid aegunud ning uuemad funktsioonid sellega ei töötanud. Ajakulu ja liigsete kitsaskohtade tõttu, mis tulenesid mudeli vanusest, sai otsustatud selle mudeli ümber õpetamise asemel alustada otse ImageNet kaaludel eeltreenitud InceptionV3 mudelist.

Lisaks Irla [23] eeskujul loodud ennustavale kihile, mis jäi lõplikus mudelis kasutusse, prooviti ka Shao [26] poolt loodud puuvilja klassifitseerija eeskujul loodud struktuuri (vt tabel 2). Optimeerijaks oli RMSProp, kaofunktsiooniks kategooriline ristentroopia ja hindamise meetrikuks täpsus.

Tabel 2. Shao [26] eeskujul loodud ennustava kihi struktuur.

Kiht	Ühikuid	Aktiveerimisfunktsioon
Ahenduskiht ( <i>Pooling</i> )	-	-
Täissidus kiht ( <i>Dense</i> )	1024	ReLU
Täissidus kiht ( <i>Dense</i> )	50	Softmax

Kuigi selle kihiga mudel näitas treenimisel alguses lootustandvaid tulemusi, siis umbes 80-nda epohhi juures hakkas mudel üle sobitama. See väljendus valideerimishulga kao järsus suurenemises samal ajal kui treenimishulga täpsus suurenes. Parim tulemus testhulgal oli selle mudeli puhul 75 epohhi järel, kus täpsus oli 63%. Samas oli teise kihiga mudelil 75 epohhi järel täpsus testhulgal juba 71%. Sellest tulenevalt otsustati selle kihiga mudeli edasi treenimist mitte jätkata.

<sup>20</sup> Pythonis kasutatav masinõppe teek. <https://www.tensorflow.org/> Vaadatud 08.05.2023

### 3 Tulemused

Selles peatükis analüüsitakse valminud mudelit ning hinnatakse, kas seatud eesmärgid said saavutatud. Veel kirjutatakse peatükis mudeli koostamisel tehtud vigadest ning mida oleks saanud paremini teha.

#### 3.1 Valminud mudeli analüüs

Lõplik mudel saavutas testhulgal 74% täpsuse, mis on seatud eesmärgist ainult 1% võrra väiksem tulemus. Top-3 ja Top-5 täpsused tulid vastavalt 89% ja 92% (vt tabel 3), millest võib järeldada, et mudeli ennustuste seas on enamasti õigesti määratud liik.

Tabel 3. Valminud mudeli näitajad

Meetrik	Väärtus
Top-1 täpsus	74%
Top-3 täpsus	89%
Top-5 täpsus	92%
Kadu	1.119

Liigiti sai mudel kõige paremini hakkama liikidega, mis olid teistest kujult ja vormilt märgatavalt erinevad (vt lisa II, tabel 5) nagu merikotkas, lauk või valge-toonekurg. Väiksematest lindudest suutis mudel tuvastada kõige paremini punarinda ja linavästriku, kellel on hästi silmatorkavad tunnused, milleks punarinna puhul on punane rind ja linavästrikul pikk saba ja valge nägu.

Halvasti tundis mudel ära linde nagu rohevint, koduvarblane ning ka mõnevõrra üllatavalt kühmnokk-luik. Rohevindi ja koduvarblase puhul on emased linnud suhteliselt ilmetud ja pruuni värvi, mistõttu mudel pakkus neile sarnaseid ilmetuid liike nagu hall-kärbsenäpp, kane-pilind või hoopis teineteist (vt joonis 13). Kühmnokk-luige puhul mudel ei suutnud ära õppida talle omast oranži nokka tuvastama ja pakkus selle asemel laululuik, kellel on muidu kollast värvi noka. Lisaks kahele luigele, oli mudelil samamoodi raskusi põhjatihase ja salutihase omavahelise eristamisega, kuid see oli ootuspärane, sest need kaks liiki on omavahel hästi sarnased.



Joonis 13. Emale rohevint (vasakul) ja emale koduvarblane (paremal).

Mudeli vajalikkuse hindamiseks võrreldi seda Google'i poolt väljatöötatud linnu klassifikaatoriga, mis suudab tuvastada 964 linnuliiki [27]. Nendest 964-st liigist, kattub 45 liiki selle töö raames loodud mudeliga. Ausaks võrdluseks eemaldati testhulgast need 5 liiki, mida Google'i mudel ei oska tuvastada, mille järel lasti mõlemal mudelil ennustada testhulga peal õigeid linnuliike. Tulemusena saavutas Google'i loodud mudel testhulgal 71% täpsuse, samas kui Eesti liikidele kohandatud mudel saavutas 75% täpsuse. Seega regioonile spetsiifilisema mudeli vajalikkus on õigustatud, sest mudel ei ennusta kohalikele liikidele sarnaseid liike, mis mujal maailmas leiduvad.

Tuginedes eelnevale, võib tulemuslega rahule jääda. Valminud mudel suudab  $\frac{3}{4}$  juhtudest etteantud liiki täpselt ennustada ning 89% täpsusega on ennustatud liik mudeli 3 kõige suurema pakutud tõenäosuse seas. Lisaks saavutas mudel testhulgal parema tulemuse kui palju rohkem liike suuta tuvastav ning muidu üldisem mudel.

### 3.2 Mudeli avalikustamine ja väärtuste kaart

Lisaks mudelile sai ka koostatud väärtuste kaart, mida saab kasutada mudeli ennustuste dekodeerimiseks. Mudel tagastab ennustused viiekümne kohalises järjendis, kus iga tõenäosus asub oma klassile vastaval indeksil. Valminud väärtuste kaart on CSV (ingl *comma-separated values*) fail, kus igale indeksile vastab eestikeelne ja ladinakeelne nimi. Tõenäosuse indeksi kokku viimisel kaardis oleva indeksiga on võimalik seeläbi tõlkida mudeli ennustused arusaadavasse vormi.

Mudel avaldati leheküljel Hugging Face<sup>21</sup>, mis on suur masinõppe kogukonda koondav lehekülg, kus on võimalik teiste loodud mudeleid ja andmestike kasutada ning enda omi üles laadida. Töö raames valmis mudelit tutvustav lehekülg<sup>22</sup>, kus on olemas lühidalt mudeli kirjeldus

<sup>21</sup> <https://huggingface.co/> Vaadatud 08.05.2023

<sup>22</sup> [https://huggingface.co/Karl-Erik/Estonian\\_Bird\\_Classifier](https://huggingface.co/Karl-Erik/Estonian_Bird_Classifier) Vaadatud 08.05.2023

ja juhend mudeli kasutamiseks oma projektideks. Leheküljelt on võimalik alla laadida nii mudelit ennast, kui ka sellega kaasas olevat väärtuste kaarti.

### **3.3 Mida oleks võinud teha teisiti ja edasised arenguvõimalused**

Vaadates tagasi tehtud tööle, siis olid mõned asjad, mida uuesti peale hakkamisel saaks teha paremini. Esiteks oleks olnud võimalik loodud andmestikke kiiremini koostada. Nimelt on pildiliste andmestike puhul võimalik koostada struktuur, kus erinevast klassist pildid on jaotatud nendele vastavate nimedega alamkaustadesse, kus alamkausta nimi käituks märgendina. See oleks kõvasti aega kokku hoidnud märgendamise poolelt ning jätnud aega muudeks tegevusteks nagu rohkemate mudelite katsetamine. Kuna tegu oli autori esimese loodava andmestikuga, siis kahjuks jäi andmestike loomisel selline variant märkamata.

Veel oleks võinud leida võimaluse korruga suuremate piltide kogumiseks andmestiku jaoks, sest suurematelt piltidelt on paremini näha linnule iseloomulikud tunnused. Seega oleks tõenäoliselt suuremate piltidega treenimisel olnud osade klasside täpsused paremad kui nad olid. Näiteks eelnevalt analüüsi osas välja toodud juhul, kus mudel ei suutnud hästi kühmnokk-luik noka värvuse järgi eristada laululuigest, oleks luige nokk olnud suuremal pildil paremini näha ning tõenäoliselt oleks luikesid ka paremini eristatud.

Mudelite puhul jäi aega rohkemate ennustavate struktuuride ja alusmudelite proovimiseks liiga väheks. Kerase rakendusliides pakub lisaks InceptionV3 mudelitele ka palju muid mudeleid, mis suudavad pilte klassifitseerida [28]. Rohkemate treenitud mudelitega oleks saanud neid omavahel võrrelda ning valida välja neist kõige paremini toimiv variant avaldamiseks.

Tulevikus oleks kindlasti eesmärk koostada suurem mudel, mis tuvastaks rohkem liike kui praegune mudel. Töö käigus loodud mudel suudab tuvastada Eesti linnuliikidest ainult 50 liiki, mis moodustab Eesti lindude koguarvust vaid 12.5% [1]. Seega saaks sellise mudeli loomisel ära kasutada loodud andmestikke ning mudelit. Näiteks saaks kasutada loodud mudelit uue mudeli alusena ning kasutades siirdeõpet õpetada see ümber tuvastama palju rohkem liike.

## Kokkuvõte

Töö eesmärk oli luua mudel, mis võimaldaks algajal linnuvaatlejal tuvastada linde, keda ta üles pildistas. Mudel pidanuks 50 tavalisema Eesti linnu korral suutma õige linnu tuvastada vähemalt 75% täpsusega. Lisaks oli eesmärk luua piltidega andmestikud, mille abil saaks vastavat mudelit treenida.

Eesmärgi saavutamiseks koguti internetist kokku 6993 pilti, millest kõik märgendati eesti-keelse, ladinakeelse ja ingliskeelse nimega. Peale seda valiti mudeli aluseks eelnevalt treenitud ImageNet kaaludega mudel InceptionV3, mis kohandati ümber tuvastama Eesti linde. Mudelit treeniti andmete peal, kasutades erinevaid andmete rikastamise võtteid, kuniks täpsus enam ei paranenud. Selle järel analüüsiti mudelit ning avaldati see koos väärtuste kaardiga keskkonnas Hugging Face.

Mudel saavutas testhulgal lõpuks täpsuse 74%, mis oli seatud eesmärgist vaid 1% võrra halvem samas kui top-3 ja top-5 täpsus tulid vastavalt 89% ja 92%. Klassifitseerimisel sai mudel kõige paremini hakkama teistest selgelt eristuvate lindudega ning halvemini hakkama omavahel sarnaste liikidega. Üldjoones saab öelda, et eesmärgid said saavutatud ning valminud mudel ja andmestikud saavad olla tulevikus abiks vastava suurema liikide arvuga mudeli koostamisel.

## Viidatud kirjandus

- [1] Eesti Ornitoloogiaühingu koduleht, Eesti linnud vaheleht (2022). Vaadatud 04.05.2023 <https://www.eoy.ee/ET/16/28/eesti-linnud/>
- [2] Svensson, L, Mullarney, K, Zetterström D. Linnumääraja. Euroopa ja Vahemere maade lindude välimääraja. Teine trükk. Tallinn: Varrak. 2020.
- [3] Kuresoo, R. (02.03.2021). Nutis ja netis on lihtsam liike määrata. *Postimees*. Vaadatud 09.12.2022 <https://leht.postimees.ee/7191387/keskkond-rein-kuresoo-nutis-ja-netis-on-liike-lihtsam-maarata>
- [4] Lemley, J., Bazrafkan, S., & Corcoran, P. (2017). Smart Augmentation Learning an Optimal Data Augmentation Strategy. *Institute of Electrical and Electronics Engineers (IEEE)*, volume 5, 5858—5869. 1, <https://doi.org/10.48550/arXiv.1703.08383>
- [5] Shorten, C., Khoshgoftaar, T.M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* 6(1), 1-48. 4, 7-16, <https://doi.org/10.1186/s40537-019-0197-0>
- [6] Yang, Z., Sinnott, R.O., Bailey, J., Ke, Q. (2023). A Survey of Automated Data Augmentation Algorithms for Deep Learning-based Image Classification Tasks. *Knowledge and Information Systems*, 1-57. 6-8, <https://doi.org/10.48550/arXiv.2206.06544>
- [7] Powell, V. (2015) Image kernels explained visually. Setosa. Vaadatud 14.04.2023 <https://setosa.io/ev/image-kernels/>
- [8] Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y. (2017). Random Erasing Data Augmentation. *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 07, lk 13001-13008). 1, <https://doi.org/10.48550/arXiv.1708.04896>
- [9] Zhang, H., Cisse, M., Dauphin, Y. N., Lopez-Paz, D. (2018). mixup: Beyond Empirical Risk Minimization. 1, <https://doi.org/10.48550/arXiv.1710.09412>
- [10] Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., Yoo, Y. (2019). CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. *Proceedings of the IEEE/CVF international conference on computer vision* (lk 6023-6032). 1, 6-7, <https://doi.org/10.48550/arXiv.1905.04899>

- [11] Shah, K. (2019). Data Augmentation Tutorial: Basic, Cutout, Mixup. Kaggle. Vaadatud 15.12.2022 <https://www.kaggle.com/code/kaushal2896/data-augmentation-tutorial-basic-cutout-mixup>
- [12] Mandal, M. (2022). Introduction to Convolutional Neural Networks (CNN). Analytics Vidhya. Vaadatud 12.04.2023 <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>
- [13] Saha, S. (2018). A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. Towards Data Science. Vaadatud 12.04.2023 <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [14] Agarap, A. F. (2018). Deep learning using rectified linear units (relu). <https://doi.org/10.48550/arXiv.1803.08375>.
- [15] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2015) Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE conference on computer vision and pattern recognition* (lk 2818-2826). 7-9 <https://doi.org/10.48550/arXiv.1512.00567>
- [16] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015) Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (lk 1-9). <https://doi.org/10.48550/arXiv.1409.4842>
- [17] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning* (lk 448-456). pmlr. <https://doi.org/10.48550/arXiv.1502.03167>
- [18] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition* (lk 248-255). Ieee. doi:10.1109/CVPR.2009.5206848
- [19] He, K., Zhang, X., Ren, S, Sun, J. (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision* (lk 1026-1034). <https://doi.org/10.48550/arXiv.1502.01852>
- [20] Google (2023). Advanced Guide to Inception v3. Google Cloud. Vaadatud 06.04.2023 <https://cloud.google.com/tpu/docs/inception-v3-advanced>



- [21] Keras dokumentatsioon, InceptionV3 vaheleht (2023). Vaadatud 11.04.2023  
<https://keras.io/api/applications/inceptionv3/>
- [22] Chollet, F. (2020). Transfer learning & fine-tuning. Keras dokumentatsioon. Vaadatud 19.04.2023 [https://keras.io/guides/transfer\\_learning/](https://keras.io/guides/transfer_learning/)
- [23] Irla, T. (2019). Transfer Learning Using Inception-V3 for Image Classification. Analytics Vidhya. Vaadatud 26.04.2023 <https://medium.com/analytics-vidhya/transfer-learning-using-inception-v3-for-image-classification-86700411251b>
- [24] Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: *Neural networks for machine learning*, 4(2), 26-31.
- [25] Koopman, W. (2017) Uk-Bird-Classifier. GitHub. Vaadatud 26.04.2023  
<https://github.com/secretbatcave/Uk-Bird-Classifier>
- [26] Shao, C. (2019). Using a pre-trained model with transfer learning: InceptionV3. GitHub Gist. Vaadatud 26.04.2023  
<https://gist.github.com/ceceshao1/cb834ed819628244093a2b61408c86e0>
- [27] Google. (kuupäev puudub) ai/vision/classifier/birds\_V1/1. TensorFlow Hub. Vaadatud 21.04.2023 [https://tfhub.dev/google/aiy/vision/classifier/birds\\_V1/1](https://tfhub.dev/google/aiy/vision/classifier/birds_V1/1)
- [28] Keras dokumentatsioon, Keras Applications vaheleht (kuupäev puudub). Vaadatud 26.04.2023 <https://keras.io/api/applications/>

# Lisad

## I. Andmestike ülesehitus

Selles lisas on tabel, kus on välja toodud andmestike ülesehitus liikide kaupa (vt tabel 4). Välja on toodud liiginimi ning temast esinevate piltide arv mõlemas andmestikus.

Tabel 4. Piltide arv andmestikes liikide kaupa

Liik	Treeningandmestikus pilte	Testandmestikus pilte
Aed-lepalind	120	22
Hakk	106	23
Hall-kärbsenäpp	118	23
Hallpea-rähn	109	26
Hallrästas	114	20
Hallvares	110	20
Harakas	108	20
Hiireviu	108	23
Käblik	116	20
Kaelustuvi	114	20
Kanepilind	130	21
Kiivitaja	108	23
Kodutuvi	106	20
Koduvarblane	110	20
Kühmnokk-luik	108	22
Kuldnokk	118	20
Künnivares	118	23
Lauk	114	21
Laululuik	114	21
Laulurästas	121	21
Leevike	120	20
Linavästrik	124	20
Merikotkas	123	20
Metsvint	138	23
Must-kärbsenäpp	118	30
Musträhn	106	20

Musträstas	127	21
Ohakalind	120	20
Pasknäär	126	20
Põhjatihane	132	20
Pöialpoiss	126	22
Põldlööke	120	20
Põldvarblane	116	20
Punarind	120	20
Puukoristaja	126	20
Räästapääsuke	120	20
Rasvatihane	132	20
Raudkull	120	24
Rohevint	134	22
Ronk	106	20
Salutihane	128	20
Siisike	123	26
Sinikael-part	124	20
Sinitihane	130	20
Sõtkas	106	20
Suitsupääsuke	118	20
Suur-kirjurähn	114	27
Talvike	120	22
Urvalind	120	21
Valge-toonekurg	110	20

## II. Klassifitseerimise aruanne

Selles lisas on tabel, kus on mudeli analüüs klasside kaupa, kasutades scikit-learn'i *classification report* funktsiooni<sup>23</sup>. Andmetest on võimalik välja lugeda, kuidas mudel ühe või teise liigi klassifitseerimisega hakkama sai (vt tabel 5).

Tabel 5. Valminud mudeli täpsused liikide kaupa

	Täpsus ( <i>precision</i> )	Saagis ( <i>recall</i> )	F1-skoor	Esinemiste arv
Aed-lepalind	0,875	0,636363	0,736842	22
Hakk	0,875	0,608696	0,717949	23
Hall-kärbsenäpp	0,592593	0,695652	0,64	23
Hallpea-rähn	0,791667	0,730769	0,76	26
Hallrastas	0,625	0,75	0,681818	20
Hallvares	0,933333	0,7	0,8	20
Harakas	0,809524	0,85	0,829268	20
Hiireviu	0,7	0,608696	0,651163	23
Kaelustuvi	0,653846	0,85	0,73913	20
Kanepilind	0,8	0,571429	0,666667	21
Kiivitaja	0,75	0,652174	0,697674	23
Kodutuvi	0,833333	0,75	0,789474	20
Koduvarblane	0,428571	0,45	0,439024	20
Kuldnokk	0,857143	0,6	0,705882	20
Käblik	0,708333	0,85	0,772727	20
Kühmnokk-luik	0,588235	0,454545	0,512821	22
Künnivares	0,888889	0,695652	0,780488	23
Lauk	0,947368	0,857143	0,9	21
Laululuik	0,59375	0,904762	0,716981	21
Laulurastas	0,583333	1	0,736842	21
Leevike	0,76	0,95	0,844444	20
Linavästrik	0,9	0,9	0,9	20
Merikotkas	1	0,85	0,918919	20
Metsvint	0,708333	0,73913	0,723404	23

<sup>23</sup> [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification\\_report.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html) Vaadatud 04.05.2023

<b>Must-kärbsenäpp</b>	0,75	0,8	0,774193	30
<b>Musträhn</b>	0,619048	0,65	0,634146	20
<b>Musträstas</b>	0,633333	0,904762	0,745098	21
<b>Ohakalind</b>	0,866667	0,65	0,742857	20
<b>Pasknäär</b>	0,866667	0,65	0,742857	20
<b>Punarind</b>	0,9	0,9	0,9	20
<b>Puukoristaja</b>	0,875	0,7	0,777778	20
<b>Põhjatihane</b>	0,545455	0,6	0,571429	20
<b>Põldlõoke</b>	0,518519	0,7	0,595745	20
<b>Põldvarblane</b>	0,866667	0,65	0,742857	20
<b>Pöialpoiss</b>	0,857143	0,818182	0,837209	22
<b>Rasvatihane</b>	0,833333	0,75	0,789474	20
<b>Raudkull</b>	0,65625	0,875	0,75	24
<b>Rohevint</b>	0,481481	0,590909	0,530612	22
<b>Ronk</b>	0,62963	0,85	0,723404	20
<b>Räästapääsuke</b>	0,625	0,75	0,681818	20
<b>Salutihane</b>	0,666667	0,6	0,631579	20
<b>Siisike</b>	0,933333	0,538462	0,682927	26
<b>Sinikael-part</b>	0,85	0,85	0,85	20
<b>Sinitihane</b>	0,857143	0,9	0,878049	20
<b>Suitsupääsuke</b>	0,777778	0,7	0,736842	20
<b>Suur-kirjurähn</b>	0,956522	0,814815	0,88	27
<b>Sõtkas</b>	0,772727	0,85	0,809524	20
<b>Talvike</b>	0,941176	0,727272	0,820512	22
<b>Urvalind</b>	0,692308	0,857143	0,765957	21
<b>Valge-toonekurg</b>	1	0,8	0,888889	20
<b>Makro keskmine</b>	0,763306	0,742631	0,742963	1067
<b>Kaalutud keskmine</b>	0,764533	0,741330	0,742795	1067

### III. Litsents

#### **Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks**

Mina, **Karl-Erik Kanal**

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Eesti lindude pildiklassifitseerija koostamine eeltreenitud mudeli baasil**, mille juhendaja on **Markus Kängsepp**, 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.

*Karl-Erik Kanal*

**08.05.2023**