

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Kristiina Keps**

# **Optiline noodituvastus**

**Bakalaureusetöö (9 EAP)**

Juhendaja: Sven Aller

Tartu 2020

## **Optiline noodituvastus**

### **Lühikokkuvõte:**

Bakalaureusetöö eesmärk on luua mobiilirakendus noodipildilt meloodia tuvastamiseks ning esitamiseks, lihtsustades seeläbi noodikirjast info lugemist. Töös kirjeldatakse optilise noodituvastuse olemust ja selle etappe ning uuritakse olemasolevaid rakendusi. Lisaks antakse ülevaade mobiilirakenduse loomise protsessist ning analüüsitakse valminud rakendust.

### **Võtmesõnad:**

Optiline muusikatuvastus, malltuvastus, mobiilirakendus

**CERCS:** P175 Informaatika, süsteemiteooria

## **Optical Music Recognition**

### **Abstract:**

The aim of this bachelor's thesis is to create a mobile application that recognizes the melody from a picture of sheet music and plays it, thus simplifying the task of extracting information from sheet music. An overview of optical music recognition, its steps and existing applications is given along with the description of the development process for the mobile application and an analysis of the result.

### **Keywords:**

Optical music recognition, template matching, mobile application

**CERCS:** P175 Informatics, systems theory

# Sisukord

Sissejuhatus .....	5
1 Optiline noodituvastus.....	6
1.1 TärGITUVASTUS .....	7
1.2 Mallituvastus.....	8
1.3 Optilise noodituvastuse etapid.....	8
1.4 Mobiilirakendused optiliseks noodituvastuseks .....	10
1.4.1 PlayScore.....	10
1.4.2 iSeeNotes.....	11
2 Noodikiri.....	12
3 Rakendus optiliseks noodituvastuseks .....	15
3.1 Androidi mobiilirakendus.....	15
3.2 Veebirakendus .....	20
3.3 Noodituvastuse algoritm .....	20
3.3.1 Mallide loomine.....	21
3.3.2 Eeltöötlus.....	22
3.3.3 Tuvastamine .....	23
4 Tulemused.....	26
4.1 Hinnang tuvastusele .....	26
4.2 Mobiilirakenduse testimine .....	29
4.3 Edasiarendamise võimalused.....	30
Kokkuvõte .....	32
Viidatud kirjandus.....	33
Lisad.....	36

I. Litsents.....36

## Sissejuhatus

Muusika on läbi ajaloo olnud inimeste jaoks olulisel kohal. See on tähtis osa kultuurist, millega paljud inimesed pea igapäevaselt kokku puutuvad. Lisaks salvestiste ja kontsertide kuulamisele kuuleme muusikat taustal isegi poes käies või ühistranspordiga sõites.

Hoolimata sellest, et muusika meid igapäevaselt ümbritseb, ei oska paljud inimesed kirja pandud muusikat lugeda. Noodikiri ehk spetsiaalsete märkide süsteem, mida muusika ülesmärkimiseks kasutatakse, on keeruline ja selle õppimine võib võtta aastaid. Selleks, et ka inimesed, kes ise noodikirja ei tunne, saaksid kirjapandud noodi põhjal ettekujutuse muusika kõlast, on võimalik noodikirjast info lugemist automatiseerida. Optiline noodituvastus eraldab noodipildilt vajaliku info, tuvastab seal leiduvad sümbolid ning paneb neist kokku muusikapala, mida on võimalik näiteks kuuldelselt esitada.

Bakalaureusetöö eesmärk on luua noodituvastuseks veebirakendus, mis kasutab malltuvastust, ning mobiilirakendus, mis on kasutajaliideseks veebirakendusega suhtlemisel. Mobiilirakendus võimaldab kasutajal teha telefoniga noodist pilti või kasutada juba olemasolevat pilti ning kuulata pildilt tuvastatud meloodiat.

Töö on jaotatud neljaks suureks peatükiks. Esimeses peatükis antakse ülevaade optilisest noodituvastusest, selle etappidest ning olemasolevatest rakendustest. Seejärel seletatakse teises peatükis lühidalt tähtsamaid noodikirja elemente, mida ka töö tulemusena loodud rakendus tuvastada üritab. Kolmandas peatükis on täpsem rakenduse kirjeldus, millele järgneb analüüs ning arutlus edasiarendamise võimaluste üle neljandas peatükis.

# 1 Optiline noodituvastus

Inimesed on läbi ajaloo muusikat tähtsaks pidanud. Muusika suudab edasi anda emotsioone ja mõtteid meloodiate ja harmoonia kaudu ning selle mõistmiseks ei pea oskama loomulikku keelt. See on oluline osa erinevatest kultuuridest ning paljud inimesed puutuvad sellega igapäevaselt kokku.

Muusikat saab edasi anda kuuldeliselt ning suurem osa muusika tarbimisest toimubki just kuulamise teel. Inimesed käivad kontsertidel, teatris, kuulavad raadiot ja salvestisi. Muusikat kasutatakse ka helindamiseks.

Teine viis muusikat edasi anda on visuaalselt. Muusika ülesmärkimiseks kasutatakse tavaliselt spetsiaalset märkide süsteemi ehk noodikirja. Kuna muusika võib olla tehniliselt üsna keeruline, siis on selle visuaalseks esitamiseks paratamatult vaja ka palju eritähenduslikke sümboleid, mis teeb noodikirja õppimise vaevaliseks ja aeganõudvaks. Paljudel inimestel pole noodilugemisoskust, seega nende jaoks on keeruline noodikirjast infot välja lugeda ja aru saada, kuidas see kirjapandud muusika võiks kõlada. Seetõttu üritatakse tänapäeval seda protsessi automatiseerida ning luua rakendusi, mis suudaksid ise noodikirja põhjal luua näiteks helifaili.

Tänapäeval leidub ka palju paberil noote, mida pole jõutud viia digitaalsele kujule ja mis võivad ajaga saada kahjustusi või kaduma minna. Digitaalsele kujule viimise ja digiteerimise all mõeldakse siin noodipildist vajaliku info eraldamist. Digiteerimine on tavalisest skannimisest parem, kuna võimaldab alles hoida vaid sisulist info. Käsitsi nootide digiteerimine on ajamahukas töö ning ka siin oleks abi rakendusest, mis suudaks suure osa sellest tööst automaatselt ära teha.

Optiline noodituvastus ehk OMR (ingl *optical music recognition*) on spetsiifiline tärgituvastuse variant, mis keskendub pildi kujul noodi digiteerimisele. Tavaliselt kasutatakse digitaalselt nootide hoiustamiseks näiteks MusicXML-faile [1], mis on mõeldud noodistusprogrammides kasutamiseks ning mida on lihtne jagada ja muuta. Variandiks on ka MIDI-failid [2], kuid need sisaldavad rohkem ettekandeks vajalikku tehnilist infot, mitte noodipildi küljendust.

## 1.1 TärGITUVASTUS

TärGITUVASTUS ehk OCR (ingl *optical character recognition*) tegeleb pildi kujul olevast tekstist tähtede ja sümbolite tuvastamise ning selle teksti teisendamisega arvutis töödeldavale kujule. TärGITUVASTUST kasutatakse erinevate trükitud tekstide digiteerimisel, et tekstide sisu oleks võimalik muuta, neist infot otsida ja neid kompaktsemalt säilitada [3].

TärGITUVASTUSE võib jagada kolme suurde etappi: eeltöötlus, tuvastamine ja järeltöötlus [3]. Eeltöötluse jooksul muudetakse pilt must-valgeks. Skannitud dokumentide puhul piisab tavaliselt fikseeritud lävendist ehk halltoonides pildipunkti ehk piksli väärtusest, millest suurema väärtusega pikslid muudetakse valgeks ja madalama väärtusega pikslid mustaks. Kaameraga tehtud pildidel kasutatakse ka kohanduvat lävendit, mis määratakse pildi heledust ja kontrasti analüüsis. Pildilt tuvastatakse ka teksti sisaldavad piirkonnad ning tavaliselt jagatakse tuvastatud tekst ka sõnadeks või üksikuteks tärkideks. Tavaliselt käib eeltöötluse juurde ka müra eemaldamine ning pildi pööramine nii, et tekstiread oleksid sirged. [4]

TärGI tuvastamiseks on mitmeid meetodeid. Üheks nendest on malltuvastus, mille puhul võrreldakse tuvastatavat täarki sisaldavat pildi tükki malliga ning leitakse nende sarnasus. Tuvastamiseks kasutatakse ka statistikat, kus funktsioonide ja kriteeriumite abil leitakse tõenäosus, et sümbol kuulub etteantud klassi. Rakendatakse ka närvivõrke, mis suudavad kohaneda ja leida sisendist klassifitseerimiseks vajalikke tunnusjooni [4].

Järeltöötluse etapis grupeeritakse tuvastatud sümbolid ja üritatakse tuvastada tekkinud vigu ning neid parandada. Sümbolid grupeeritakse ühte sõnesse, kui nad asuvad üksteisele piisavalt lähedal. Käitsi kirjutatud teksti puhul on see ülesanne keerulisem, sest siis võivad tärkide vahed olla ebahühtlase suurusega või üldse puududa. Erinevaid vigade tuvastamise meetodeid on palju. Kui teksti keel on teada, siis saab vigade tuvastamiseks kasutada ära ühe tähe teisele järgnemise tõenäosust antud keeles. Tuvastatud sõnu saab kontrollida ka sõnastiku abiga. See meetod leiab enim vigu ning aitab leida ka parimat sobivat vastet vigasele sõnale, aga on ka üsna ajamahukas [4].

## 1.2 Malltuvastus

Malltuvastus on masinnägemise tehnika, kus tuvastatakse objekte pildil etteantud mallidega. Kõige tavalisem viis malltuvastuse jaoks on kõikvõimalikud alad pildil üle vaadata, leides sarnasuse malli ja pildi tüki vahel. Sarnasust kirjeldavat numbrilist väärtust nimetatakse korrelatsiooniks. Selleks, et korrelatsiooni väärtus ei sõltuks nii palju näiteks pildi heledusest, kasutatakse normaliseerimist. Normaliseerimise üheks variandiks on iga piksli väärtusest pikslite keskmise väärtuse lahutamine. Tavaliselt teisendatakse korrelatsiooni väärtus ka mingisse kindlasse lõiku, näiteks  $[-1, 1]$ , kus 1 tähendab, et pildid on identsed ning -1, et pildid on vastandid [5].

Et leida pildilt tuvastatud objektid saab määrata korrelatsioonile lävendi ehk suuruse, mille ületamisel loetakse vastav koht pildil tuvastatavat objekti sisaldavaks. Niimoodi saab leida pildilt kohad, mis sisaldavad otsitavat objekti. Sellise lähenemise puhul peab objekt pildil olema pööratud samas suunas ning olema sama suur nagu mallil [5].

## 1.3 Optilise noodituvastuse etapid

Optilist noodituvastust on tänapäevaks juba päris palju uuritud ning selle probleemi lahendamiseks on katsetatud erinevaid meetodeid. Klassikalised optilise noodituvastuse etapid on eeltöötlus, noodijoonestiku eemaldamine, sümbolite eraldamine ja tuvastamine ning muusikapala kokkupanemine [6].

Esimeseks sammuks on teha pildile eeltöötlus. Kaameraga tehtud piltide puhul tuleb arvestada, et valgustingimused võivad olla väga erinevad, pilt võib olla tehtud mingi nurga all, mitte otse ülevalt alla, ning paberil muusikapala võib olla pildil pööratud. Eeltöötusega üritatakse liigset müra eemaldada ning pildi omadusi eduka tuvastamise jaoks parandada [6].

Paljud OMR rakendused alustavad pildi must-valgeks tegemisest. Selleks tuleb pilti analüüsida ning eraldada taust ja müra tähtsatest objektidest [6] [7] [8]. Moonutatud noodijoonestikuga toimetulekuks saab näiteks leida noodijoonestiku otspunktid ning nende põhjal arvutada transformatsioonimaatriks, mille abil pilti korrigeerida, et saada sirged noodijoonestiku read [7].



Sellele ülesandele võib ka läheneda ühe takti põhiselt, leides kõigepealt taktid ning arvutades iga takti jaoks eraldi transformatsiooni maatriks [9].

Sümbolite eraldamise lihtsustamiseks kasutatakse tihti noodijoonestiku eemaldamist [7] [8]. Noodijoonestiku eemaldamine pole lihtne ning selle käigus jääb tihtipeale tükikesi noodijoonestikust siiski alles või kustutatakse kogemata ka olulisi sümboleid, mis võib ka tuvastamist raskendada. Seetõttu jäetakse vahepeal noodijoonestiku eemaldamise etapp vahele [6] [9]. Samuti on noodijoonestik jäetud eemaldamata mitmes töös, mis kasutavad tuvastamiseks närvivõrke [10] [11] [12].

Järgmine samm on tavaliselt sümbolite eraldamine üksteisest ja nende tuvastamine. Üks variant on tuvastada noodipead, -varred, -lipud, punktid ja pausid eraldi, kuid kaustatakse ka noodipea, -varre ja -lipu tuvastamist korraga [6]. Tuvastamiseks saab kasutada näiteks malltuvastust, mille kasuks otsustasid ka Wallner [8] ning Vo jt [9] oma töödes.

Mõlemas töös tuvastati noodivarsi ja -lippe noodipeadest eraldi. Kuna Vo jt jagasid eelnevalt oma töös „Distorted Music Score Recognition without Staffline Removal“ noodijoonestiku taktideks, siis tehti noodituvastust ka ühe takti piires. Kõigepealt tuvastati noodivarred, seejärel mustad noodipead ning kui mõnele varrele ei leitud noodipead, siis prooviti ka poolnootide tuvastamist. Kui ei leitud musti noodipäid ega poolnoodipäid, siis tõenäoliselt oli taktis tervenoot. Kuna noodijoonestiku joonte asukohad pildil on teada, siis on ka lihtne määrata tuvastatud noodipeale õige kõrgus [9].

Wallner leidis oma töös „A System for Optical Music Recognition and Audio Synthesis“ kõigepealt x-telje projektsiooni abil tõenäolised nootide asukohad, et vähendada otsinguala. Seejärel tuvastas noodivarred ning nende läheduses kasutas malltuvastust ka noodipeade, alteratsioonimärkide ja punktide leidmiseks [8].

Rakendused, mis kasutavad tuvastamiseks närvivõrke, jätavad tihtipeale mõned klassikalise noodituvastuse etapid vahele ning üritavad leida viisi, kuidas mudel suudaks õppida ise erinevate omadustega piltidelt vajalikku infot eristama, sümboleid tuvastama ja muusikapala kokku panema. Tihti kasutatakse LSTM (ingl *Long Short-Term Memory*) närvivõrke, mis on sobivad järjestuste töötlemiseks, ning konvolutsioonilisi närvivõrke, mida kasutatakse pildilt tunnuste leidmiseks [12]. Juba eraldatud sümbolite tuvastamiseks võib kasutada ka ainult konvolutsioonilist võrku [7].

LSTM närvivõrku on kasutatud töös „Optical Music Recognition by Recurrent Neural Networks“, kus käsitleti ühe noodijoonestikuga pilte, mida saab vasakult paremale lugeda [11].

Viimane samm on tervikliku muusikapala konstrueerimine. Kasutades muusika kirjutamise reegleid ning muusikafailide struktuuri nõudeid luuakse tuvastatud nootidest ja rütmidest lõplik teos [6].

## **1.4 Mobiilirakendused optiliseks noodituvastuseks**

Nutitelefonidele on loodud juba mitmeid noodituvastuse rakendusi, mis võimaldavad kasutajal noodist pilt teha või see üles laadida ning tihtipeale ka esitada tuvastatud muusikat või seda alla laadida näiteks MIDI-faili kujul.

### **1.4.1 PlayScore**

PlayScore on rakendus nii Android kui iOS platvormidele. Androidi jaoks on sellel nii tasuta Lite versioon kui ka tasuline Pro versioon. Rakendus tuvastab masinkirjas noote eeldusel, et pilt on tehtud tasapinnal ja piisava valgusega. Tasuta versioon lubab kuni kahe noodijoonestikuga noote ning laseb esitada tuvastatud muusikat vaid läbi rakenduse. Pro versioon lubab MIDI- ja MusicXML-faile ka alla laadida ning tuvastamiseks mõeldud pilti üles laadida. Rakendus suudab tuvastada taktijooni, pause, alteratsioonimärke, kordusi, dünaamikat, noodivõtmeid, taktimõõte ning helistiku muutusi [13].

Autor katsetas rakendust ka ise. Tasuta rakendus oli minimalistlik, seal oli vaid kaamera nupp ning pärast pildi tegemist lisandus ka tuvastamise nupp. Pildi üleslaadimise võimalust ei olnud ning vaid üksikutel kordadel suutis rakendus pildilt noote tuvastada. Umbes pooled noodid jäid tavaliselt tuvastamata, kuid tuvastatud nootide helikõrgus tundus enamasti õige ning arvestati ka alteratsioonimärkidega.

Google Play poes on 720 kasutajat andnud rakendusele keskmiseks hindeks 2,5/5. Kasutajate jäetud tagasisides tuuakse välja, et kui rakendus ei suuda noote tuvastada, siis ei anta kasutajale

teada, milles täpsemalt probleem oli. Samuti mainitakse, et osa noote jäävad tuvastamata ning tihti ei suuda rakendus tuvastada ühtegi nooti ka hea kvaliteediga pildilt. [14].

### **1.4.2 iSeeNotes**

Rakendus iSeeNotes on sarnaselt PlayScore'ile nii Android kui iOS platvormidele. Rakenduses on võimalik tuvastatud muusika MIDI-failina alla laadida, kuid pilti pole võimalik üles laadida. Rakendus lubab kahe noodijoonestikuga noote ning suudab noodivõtmeid, helistikku, alteratsioonimärke, punktiga pikendatud noote, noodilippe, talasid ja pause [15].

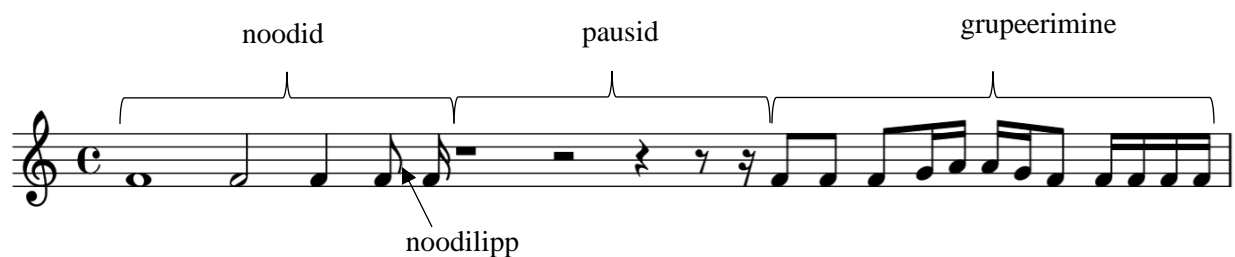
Autor katsetas rakenduse tasuta prooviversiooni. Rakendust sai kasutada vaid horisontaalses asendis, mistõttu sai selge pildi teha vaid paarist noodireast korraga. Tuvastamisel jäeti noote vahele ning helide kõrgused ei olnud alati õiged. Noote tuvastati ka seal, kus neid tegelikult ei olnud, näiteks viiulivõtme kohal. Lisaks nootidele suudab rakendus tuvastada ka akordimärke, mis on kasulik omadus. Tuvastatud nootidest ja akordidest kokkupandud muusikapala saab kuulata ning ka MIDI-failina alla laadida.

Google Play poes on sellele nii 72-tunnine tasuta prooviversioon, millele 192 kasutajat on andnud keskmiseks hinnanguks 1,9/5 [16] ning tasuline versioon, mis maksab 3,49 eurot ning mille keskmine hinnang on 2,7/5 (hinnanud on 57 kasutajat). Tekstilises tagasisides tuuakse välja, et rakendus jättis noote tuvastamata ja tuvastas müra nootidena. Kuna pilt peab olema tehtud väga lähedalt, et rakendus suudaks noote tuvastada, siis pole võimalik tervet lehekülge korraga pildile jätta. Lihtsate lugude tuvastamisega saab rakendus kasutajate sõnul hästi hakkama [17].

## 2 Noodikiri

Noodikiri on erinevatest sümbolitest koosnev süsteem, mida kasutatakse muusika ülesmärkimiseks. Selles peatükis käsitletavat elementide on vältused, grupeerimine, taktid, taktimõõt, noodijoonestik, noodid, võtmemärgid ja alteratsioonimärgid.

Heli suhtelist kestvust ehk vältust tähistatakse noodi kujuga. Põhivältuste hulka kuuluvad terve-, pool-, veerand-, kaheksandik- ja kuueteistkümnendiknoodid. Heli puudumist tähistavad pausid, mille vältusi tähistatakse samuti iseloomuliku kujuga. Lühikesi noodivältusi (kaheksandik- ja kuueteistkümnendiknoote) grupeeritakse ehk lipukesed asendatakse taladega. Talade arv on võrdne lipukeste arvuga [18]. Nootide ja pauside vältused kahanevas järjekorras ning grupeerimise näited on toodud joonisel 1.



Joonis 1. Vältused ja grupeerimine

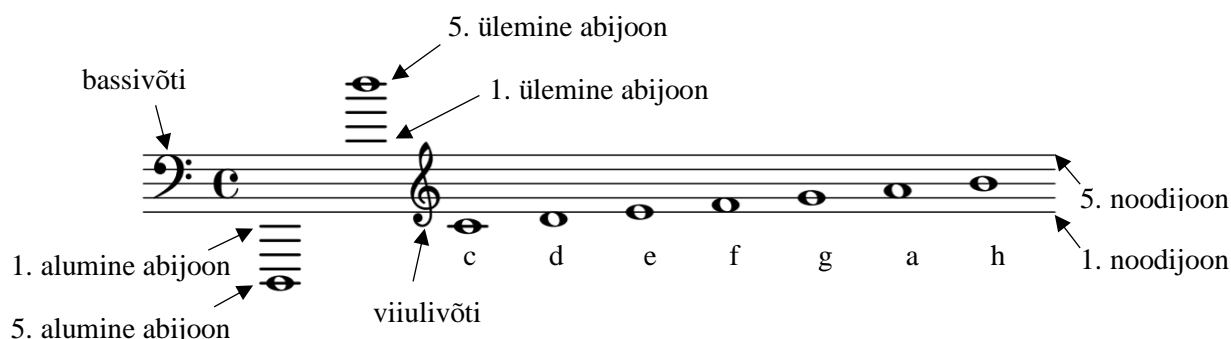
Taktid on üksused, mida eraldavad taktijooned. Löökide arv taktis ja löögi vältus määratakse taktimõõduga. Taktimõõt koosneb kahest kohakuti asetsevast numbrist, millest ülemine tähistab löökide arvu taktis ning alumine ühe löögi vältust. Taktimõõt kirjutatakse muusikapala esimese takti algusesse [18]. Mõned tavalisematest taktimõõtudest on 4/4, 3/4, 2/4, 2/2 ja 6/8. Näiteks 2/4 taktimõõdud tähendab, et taktis on kaks lööki ning ühele löögile vastab veerandnoot. Joonisel 2 on toodud taktimõõdud ja nendele vastavad ühelöögilised noodid. Taktimõõdu 4/4 tähistamiseks on ka eriline C-tähele sarnanev sümbol, mis on joonisel esimene [19].



Joonis 2. Taktimõõdud

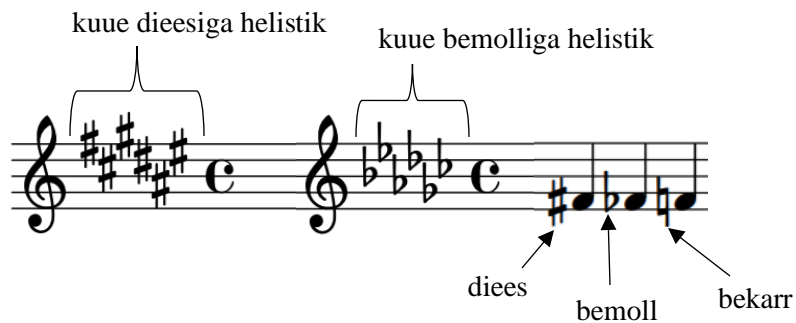
Helide kõrguse määramiseks paigutatakse noodid noodijoonestikule Noodijoonestik koosneb viiest joonest ning neljast joonevahest. Lisaks tavalisele viiele noodijoonele kasutatakse ka abijooni, mida lisatakse noodijoonestiku alla või kohale, et panna kirja madalamaid või kõrgemaid noote. Tavalisi noodijooni ja joonevahesid loendatakse alt üles. Sarnaselt loendatakse ka ülemisi abijooni, kuid alumisi loendatakse ülevalt alla (joonis 3) [18].

Helide kõrguste ja vältuste ülesmärkimiseks kasutatakse noote. Alushelidele on määratud ka tähtnimed: c, d, e, f, g, a, h (joonis 3). Kahe lähima c vahele jäävat vahemikku nimetatakse oktaviks. Iga noodirea algusesse kirjutatakse noodivõti, mis määrab kindlat helikõrgust tähistava noodi asukoha noodijoonestikul. Noodivõtmetest enimlevinud on viiulivõti ja bassivõti. Viiulivõti määrab ära esimese oktaavi g ning bassivõti määrab väikese oktaavi f [18]. Kõrgemad partiid on enamasti kirjutatud viiulivõtmes, madalamad bassivõtmes (joonis 3) [19].



Joonis 3. Noodivõtmed ja noodid

Alushelide kõrgendamiseks ja madaldamiseks ehk alteratsiooniks kasutatakse alteratsioonimärke. Diees kõrgendab nooti poole tooni võrra ning bemoll madaldab nooti poole tooni võrra. Bekarr muudab eelneva alteratsioonimärgi kehtetuks (joonis 4). Alteratsioonimärgid võivad esineda juhuslike märkidena teose vältel vahetult noodi ees. Nad kehtivad sama kõrgusega nootidele vaid selle takti piires, kus nad esinevad. Neid saab kasutada ka võtmemärkidena, mis määravad helistiku, ning sellisel juhul kirjutatakse nad iga rea algusesse noodivõtme järele. Võtmemärgid kirjutatakse vaid ühte oktavisse, kuid need kehtivad vastavatele helirea helidele kõikides oktavites kogu teose vältel. Võtmemärkidel on kindel järjekord (joonis 4) [18].



Joonis 4. Alteratsioonimärgid

Selles peatükis toodu pole kaugeltki kogu noodikirjas kasutatav tähiste hulk. Lihtsuse mõttes kirjutati vaid nendest noodikirja komponentidest, mille abil lihtsamaid teoseid kirja panna ning mida ka selle töö rakenduslikus osas kasutatakse.

### **3 Rakendus optiliseks noodituvastuseks**

Optilise noodituvastuse rakendus on jagatud kaheks osaks: eesrakenduseks ja tagarakenduseks. Eesrakenduseks on Androidi mobiilirakendus, mis võimaldab kasutajal lihtsa vaevaga teha või üles laadida pilt ning esitada ja alla laadida pildi põhjal tuvastatud muusikafail. Mobiilirakendus suhtleb päringute kaudu tagarakendusega, mis asub serveris ja mille tööks on pildilt ühehäälnelise meloodia tuvastada ning tagastada mobiilirakendusele MIDI-fail.

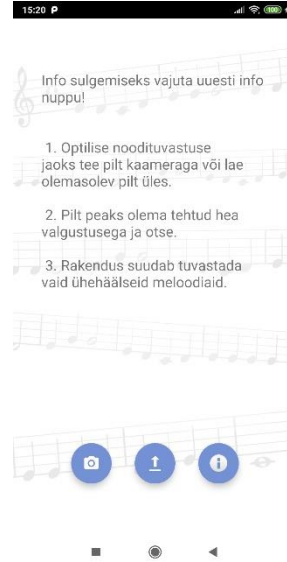
#### **3.1 Androidi mobiilirakendus**

Androidi mobiilirakendus on kasutajaliideseks noodituvastusrakendusele. Mobiilirakendus on minimalistlik ning koosneb vaid ühest vaatest, kus üksikud elemendid vahetuvad vastavalt kasutaja tegevusele. Rakenduse ülesanded on:

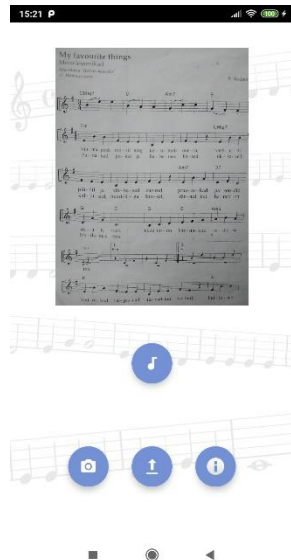
- 1) kasutajale rakenduse kasutamise ja noodituvastuse piirangutega seonduva info kuvamine;
- 2) pildi tegemise ja üleslaadimise võimaldamine;
- 3) pildi kodeerimine ja edastamine päringuga veebirakendusele;
- 4) päringust saadud MIDI-faili dekodeerimine ning selle esitamine ja allalaadimine.



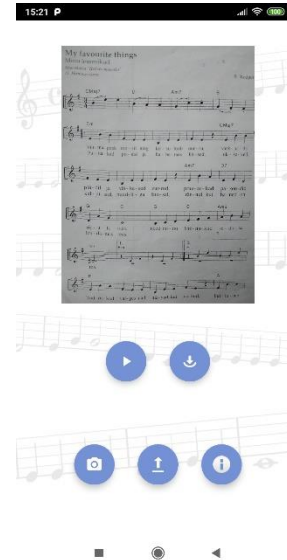
Joonis 5. Avaleht



Joonis 6. Info



Joonis 7. Pildi tegemine/valimine



Joonis 8. Tuvastatud muusikapala

Rakenduse avamisel suunatakse kasutaja pealehele (joonis 5). Kõikide tegevuste puhul kuvatakse kasutajale kolme põhinuppu: kaameraga pildi tegemine, pildi üleslaadimine ning info kuvamine. Nupud on tähistatud tegevustele vastavate ikoonidega. Infot on võimalik kuvada igal hetkel, vajutades info nupule ning sulgeda uuesti info nupule vajutades. Info kuvamine (joonis 6) ei



katkesta kasutaja hetketegevust ning selle sulgedes kuvatakse kasutajale sama vaadet, mis enne info lugemist.

Esmakordselt kaamera nupule vajutamisel küsib rakendus luba kaamera kasutamiseks ja salvestusruumi kirjutamiseks. Kui kasutaja vajalikke õigusi ei anna, siis kaamerat kasutada ei saa ning kasutajale kuvatakse ka teade ligipääsuõiguste puudumise kohta. Kui õigused on olemas, siis saab kasutaja kaameraga pildi teha ning tehtud pilti kuvatakse ka rakenduses koos tuvastamise nupuga, mis on tähistatud noodi ikooniga (joonis 7). Pildi üleslaadimisel küsitakse salvestusruumi lugemise õigust ning õiguste olemasolul ja pildi üleslaadimisel kuvatakse samuti valitud pilti ja tuvastamise nuppu.

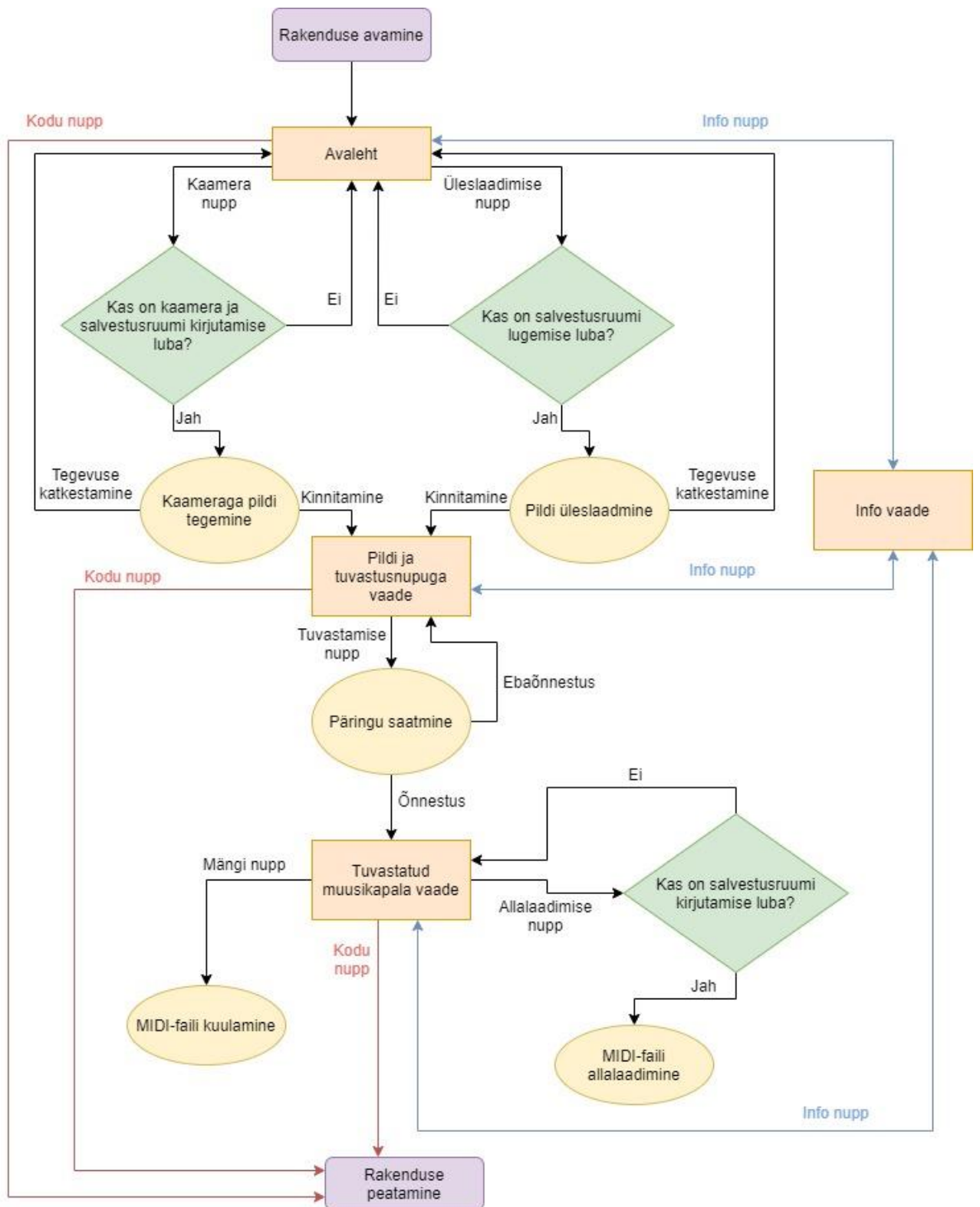
Vajutades tuvastamise nupule, käivitatakse tuvastamise protsess. Mobiilirakendus paneb kokku JSON päringu, mis saadetakse veebirakendusele. Tuvastamine võib võtta kaua ning tuvastamise ajal kuvatakse kasutajale tuvastamisnupu asemel laadimisikooni (ingl *spinner*). Tuvastamine katkestatakse, kui kasutaja laeb üles või teeb kaameraga uue pildi. Kui tuvastamise ajal tekib viga, siis kuvatakse kasutajale veateade ning laadimisikoon asendub jälle tuvastamisnupuga. Erinevaid veateateid on kolm:

- 1) „Ei suutnud tuvastada noodijooni. Kontrolli, kas noodijooned on pildil sirged ja pilt on selge!“
- 2) „Ei suutnud tuvastada ühtegi nooti! Kontrolli, et pilt oleks selge!“
- 3) „Tekkis viga! Palun proovi uuesti.“

Esimest veateadet kuvatakse, kui noodituvastuse rakendus ei suuda tuvastada ühtegi noodijoont – sellisel juhul on ilmselt tegemist kaldus pildiga, kus noodijooned on võrreldes pildi servaga liiga suure nurga all. Teist veateadet kuvatakse, kui noodijooned suudeti tuvastada, aga ühtegi teist sümbolit mitte. Sellisel juhul võib tegemist olla halva valgustuse või kehva kvaliteediga pildiga. Kolmandat veateadet kuvatakse, kui tekib mõni ootamatu viga.

Eduka tuvastamise korral kuvatakse kasutajale tuvastatud loo mängimiseks ning MIDI-faili allalaadimiseks vastavaid nuppe (joonis 8). Faili saab kuulata korduvalt ning vajadusel ka vahepeal pausile panna. Faili allalaadimiseks on jällegi vaja salvestusruumi kirjutamise õigusi ning kui neid veel antud pole, siis küsitakse neid esmakordsel allalaadimisel. Uue noodipildi tuvastamiseks saab kasutaja vajutada uuesti kaamera või üleslaadimise nuppu.

Mobiilirakenduse kasutamise skeem on toodud joonisel 9, kus erinevate elementidega vaated on oranžides ristkülikutes, tähtsamad tegevused on kollastes ellipsites, õiguste küsimised rohelistes rombides ning rakenduse avamine ja sulgemine lillades kastides. Rakenduse sulgemiseks saab igas vaates vajutada telefoni kodu nuppu (märgitud punase joonega) ning info kuvamiseks rakenduse info nuppu (märgitud sinise joonega).



Joonis 9. Mobiilirakenduse kasutamise skeem

Rakenduse minimaalne toetatud versioon on Android 8.0.0 (API tase 26). Mobiilirakenduse arendamisel kasutati arenduskeskkonda Android Studio ning programmeerimiskeelt Java.

### 3.2 Veebirakendus

Veebirakenduse ülesandeks on teostada päringuga saadud pildil noodituvastus, luua MIDI-fail ning saata see päringule vastuseks. Veebiraamistikuks valiti Flask [20], kuna see on mõeldud Pythoni jaoks ning ka noodituvastusrakendus on kirjutatud Pythonis. Flaski puhul on tegemist mikroraamistikuga, mis sobib hästi antud ülesande jaoks, kuna veebirakenduse ainus töö on ühele päringule vastus saata.

Tuvastamiseks tuleb veebirakendusele saata POST-päring, kus JSON vormingus on kaasas pilt. See dekodeeritakse ning salvestatakse ajutiselt rakenduse kausta. Pildi põhjal loob noodituvastusrakendus MIDI-faili, mis saadetakse samuti JSON formaadis päringule vastuseks. Pärast vastuse saatmist kustutatakse tuvastamise käigus loodud ajutised failid.

Veebirakenduse juurutamiseks (ingl *deploy*) ja majutamiseks valiti Heroku platvorm [21], sest seda on lihtne kasutada ning see on tasuta. Veebirakenduse lähtekood on vabalt kättesaadav autori hoidlas [22] ning töötav rakendus asub aadressil <http://musicrecognition.herokuapp.com/>.

Kuna noodituvastuse rakendus kasutab tuvastamisel lisaks Pythoni enda teekidele ka Lilypondi rakendust, siis konteineriseeriti rakendus koos vajalike teekide ja Lilypondiga, kasutades Dockerit [23]. Alusena kasutati Linuxit, kuna see lihtsustas vajalike komponentide paigaldamist. Dockeri konteiner sisaldab kõike, mida rakendusel töö jaoks vaja on, ning isoleerib ta ülejäänud keskkonnast.

### 3.3 Noodituvastuse algoritm

Noodituvastuses kasutatakse erinevate sümbolite tuvastamiseks eelnevalt loodud malle ning noodituvastus toimub kahes etapis: eeltöötlus ja tuvastamine. Eeltöötluse jooksul muudetakse pilt

mustvalgeks ning määratakse noodijoonte asukohad. Tuvastamise etapp on põimitud järeltöötlusega. Mallide abil leitakse erinevate sümbolite asukohad pildil ning eemaldatakse liigsed tuvastatud sümbolid. Jooksvalt muudetakse nootide vältuseid ja kõrgusi ning pannakse tuvastatud sümbolid muusikapalaks kokku.

### 3.3.1 Mallide loomine

Mallidel on noodituvastuse rakenduses tähtis roll – nende abil saab erinevaid sümboleid pildilt tuvastada. Mallide loomiseks kasutati tarkvara LilyPond, mille abil saab tekstifailist luua nii PDF-faile kui ka näiteks MIDI-faile. LilyPondi tekstifailidel on oma formaat ning seal on võimalik kirja panna kõikvõimalikke muusikalisi sümboleid. Näide LilyPondi tekstifaili formaadist on toodud joonisel 10.

```
\version "2.20.0"
\score {
  \new Staff {
    \clef treble
    \key c \major
    \time 4/4
    e'4 d2 c,1 d''1
  }
}
```

noodivõti  
helistik  
taktimõõt  
noodid (noodinimi, helistik, vältus)

Joonis 10. LilyPondi formaat

Noodituvastuse tarvis loodi mallid täis-, pool- ja veerandnoodi noodipeade ning kaheksandik- ja kuuteistkümnendiknootide eraldiseisvate ning taladega ühendatud lippude jaoks. Ühtlasi loodi mallid nii viuli- kui bassivõtme ning alteratsioonimärkide diees, bemoll ja bekarr jaoks. Lisati ka erineva pikkusega pauside (täis-, pool-, veerand-, kaheksandik- ja kuuteistkümnendikpaus) ning enamlevinud taktimõõtude (4/4, 3/4, 2/4, 2/2, 6/8) mallid.

Enne sellise mallide valikuni jõudmist katsetati ka varianti, kus tuvastati noodi kõrgus ja vältus korraga malli abil. See eeldas iga noodikõrguse iga vältuse jaoks malli loomist. Selliselt prooviti tuvastada noote ühe oktaavi piires, aga kuna vältuste ja helide kombinatsioone on palju, siis võttis mallituvastus väga kaua aega ning tulemus ei olnud väga täpne. Seetõttu otsustati hoopis teise

lähenemise kasuks – tuvastada noodipeasid, -lippe ja talasid eraldi ning kõrgused määrata hoopis tuvastatud noodijoonte abil.

### 3.3.2 Eeltöötlus

Kaameraga tehtud piltide puhul tuleb arvestada erineva valgustuse ja pildi suurusega, seega on eeltöötlus kindlasti vajalik. Eeltöötuse jaoks kasutati erinevaid OpenCV teegi meetodeid.

Esmalt teisendatakse pilt halltoonidesse ning seejärel tehakse pilt mustvalgeks. Mustvalgeks tegemisel kasutatakse kohanduvat lävendit (ingl *adaptive thresholding*), tänu millele ei ole tuvastusalgoritm pildi valgustuse suhtes nii tundlik. Iga piksli jaoks leitakse tema lähiümbruses asuvate pikslite väärtuste keskmine ja kui piksli väärtus on keskmisest suurem, siis muudetakse see valgeks, vastasel juhul mustaks. Autor katsetas ka teisi eeltöötuse tehnikaid, nagu pildi hägustamine ja müra eemaldamine, kuid tihti parandasid need pildi kvaliteeti vaid teatud pildidel ning teistel kaotasid hoopis osa tähtsast informatsioonist.

Eeltöötuse puhul on tähtis osa ka noodijoonte tuvastamisel. Ka selleks prooviti läbi mitmeid variante. Horisontaalse projektsiooni abil saab leida mustvalgel pildil mustade pikslite arvu igas reas ning sellega joonistuvad selgelt välja suurte mustade pikslite arvudega read, mis tähistavadki noodijoonte asukohti. See lahendus töötab väga hästi siis, kui noodijooned on pildi alumise äärega paralleelsed, aga juba väike kalle tähendab, et noodijoonte pikslid pole algusest lõpuni samas reas ning neid ei saa piisavalt usaldusväärselt tuvastada.

Seetõttu otsustati joonte tuvastamiseks kasutada hoopis Hough' transformatsiooni [24], mis suudab leida ka nurga alla olevaid jooni ning töötab ka mürase pildiga. Enne Hough' transformatsiooni kasutati halltoonides pildil ka Canny servatuvastust [25], mis leiab punktid, kus pildi heledus muutub järsult. Hough' transformatsiooni jaoks määrati punktide arv, mis peavad ühel joonel asuma, et seda jooneks loetaks. Algseks joone punktide arvuks määrati 400, mis on enamasti piisavalt suur arv, et tuvastada vaid noodijooni ja jätta välja näiteks jooned, mis võivad tekkida laulusõnade või pealkirjade tõttu. Kui aga sellise punktide arvuga tuvastatakse vähem kui viis noodijoont, mis on minimaalne arv, mis üherealisel noodil olla saab, siis vähendatakse punktide arvu 50 võrra ning proovitakse uuesti. Siiski ei lubata punktide arvul minna väiksemaks

kui 200, sest siis ei pruugi tuvastatud joonte puhul enam noodijoontega tegemist olla (joontena tuvastati ka näiteks laulusõnu).

Rakendus eeldab, et kasutaja saadab tuvastamiseks õiget pidi ning pildi alumise äärega paralleelsete noodijoontega pildi, kuna see peaks olema kasutaja poolt lihtsasti tehtav ning siis ei ole ka ohtu, et eeltöötluse käigus mõne valesti tuvastatud noodijoone tõttu pilti liialt moonutatakse ning tuvastamist hoopis raskendatakse. Rakendus jätab tuvastatud joontest alles vaid need, mis on horisontaalsed (kuni kahekraadise nurga all). Kui ühtegi joont alles ei jää, siis tuvastusega ei jätkata ja kasutajale saadetakse vastav teade.

Tuvastatud noodijoontest eemaldatakse liiga lähestikku asuvad jooned ning seejärel leitakse kõige sagedasem vahemaa kahe noodijoone vahel. Selle põhjal arvutatakse välja, kui palju peaks pilti vähendada või suurendama, et sellel oleks võimalik malltuvastust kasutada. Kõik mallid on tehtud ühtlase mõõtkavaga ning mallidel on kahe noodijoone vaheliseks kauguseks 14 pikslit, seega ka tuvastataval pildil peaks see kaugus olema sama.

### **3.3.3 Tuvastamine**

Noodijooni kasutatakse ka nootide kõrguste tuvastamisel. Noodijooned grupeeritakse nende asukohtade järgi, et leida read. Seejärel leitakse iga noodijoone jaoks vastava noodi nimetus ning oktavi tähis. Arvutatakse välja ka noodijoonte vahede keskkohad ning neile määratakse samuti noodi nimetused ja oktavid. Sarnaselt talitatakse ka kolme ülemise ja kolme alumise abijooone puhul. Rohkem abijooni ei tuvastata.

Kuna nootide nimetused ja oktavid olenevad noodivõtimest, siis proovitakse eelnevalt pildilt bassivõtit tuvastada ning kui see ei õnnestu, siis määratakse noodile vaikimisi viulivõti. Tuvastatakse ka taktimõõt ning kui ühegi malliga ei suudeta taktimõõtu tuvastada, siis määratakse vaikimisi taktimõõduks 4/4. Taktimõõt ei ole oluline edasise tuvastamise jaoks, aga see kirjutatakse järeltöötluse käigus MIDI-faili.

Noodipeade puhul tuvastatakse veerand-, pool- ja tervenoot. Tuvastatud noodipea puhul leitakse selle keskpunkti koordinaat ning sellele lähim noodijoon või noodijoonte vahe keskosa, mille järgi määratakse noodile kõrgus. Kui noot asub noodijoonestikust liiga kaugel, siis tõenäoliselt tuvastati

noodina hoopis müra ning seda nooti ei arvestata. Kõik tuvastatud sümbolid sorteeritakse ridades x-koordinaatide järgi, et saada õige järjestus.

Noodivarsi ei tuvastata, aga veerandnootide puhul tuleb kontrollida, kas on tegemist hoopis kaheksandik- või kuueteistkümnendiknoodiga. Selleks tuvastatakse nii õigetpidi kui ka pööratud lipud ja talad nii kaheksandik- kui kuueteistkümnendiknootide jaoks. Väiksema vältusega noote ei tuvastata, kuna neid esineb lihtsamates palades, mida antud rakendus tuvastada suudab, võrdlemisi harva. Iga tuvastatud lipu ja tala jaoks leitakse lähim veerandnoodipea ning muudetakse vastava noodi vältust.

Helistiku määramiseks proovitakse kõigepealt leida dieese, mis esinevad noodijoonestiku alguses, vähemalt kahe noodipea kaugusel esimesest tuvastatud noodist. Kui ühtegi dieesi ei leita, siis proovitakse tuvastada ka bemolle. Kui ei tuvastata ka neid, siis järelkult on tegemist märkideta helistikuga. Võtmemärkide jaoks ei leita kõrgust, kuna helistiku ja märkide kõrgused saab määrata lihtsalt tuvastatud märkide arvu järgi.

Juhuslike märkidena tuvastatakse dieese, bemolle ja bekarre. Lihtsuse mõttes jäeti harvem esinevad dubldieesid ja -bemollid alteratsioonimärkide seast välja. Nende puhul kontrollitakse, et nad asuksid mõne noodi ees ja maksimaalselt pooleteise noodipea kaugusel. Juhuslike märkide puhul ei leita samuti märgi kõrgust, vaid kõrgendatakse või madaldatakse järgmist nooti poole tooni võrra. Kui tuvastatavaks märgiks on bekar, siis eemaldatakse kõik varasemad alteratsioonimärgi tähised järgmise noodi küljest.

Hetkel lisatakse alteratsioonimärk vaid sellele noodile, mille ees ta esineb, mitte ka teistele sama kõrgusega nootidele samas taktis. Takti piires noodi kõrgendamine või madaldamine eeldab, et teatakse, kus asuvad taktijooned või loetakse tuvastatud nootide ja pauside vältuseid alates loo algusest ning peetakse järge, kus üks takt lõppeb ja teine algab. Taktijoonte tuvastamine osutus keeruliseks, kuna suuruselt on need sarnased noodivartele ning seega ei õnnestu hästi vaid taktijooni tuvastada. Kuigi igale noodivarrele peab vastama ka noodipea, ei saa olla kindel, et kõik noodipead on tuvastatud ning seega ei toimi hästi ka noodipeade järgi taktijoonte ja noodivarte eristamine. Vältuste lugemisega tekib sama probleem – kui mõni noot on vahelt puudu, siis pole ka taktid enam õigetes kohtades. Juba väike viga tekitab olukorra, kus alteratsioonimärk lisatakse noodile, mis tegelikult ei ole samas taktis ning kuna taktijoonte tuvastamine või vältuste lugemine



ei aita tihtipeale siiski õigetele nootidele alteratsioonimärke lisada, siis otsustati see samm üldse ära jätta.

Viimasena tuvastati ka pausid. Need jäeti viimaseks, et noodilippude ja alteratsioonimärkide tuvastamisel ei peaks kontrollima, kas lähim noot on ikka noot, mitte paus. Pausid on samade vältustega nagu tuvastatavad noodid: terve-, pool-, veerand-, kaheksandik- ja kuueteistkümnendikpaus.

Viimane samm oli tuvastatud nootidest ja pausidest MIDI-fail kokku panna. Selleks kasutati uuesti LilyPond tarkvara. Tuvastamise käigus anti nootidele ja pausidele nimetused ja vältused samas formaadis, nagu need LilyPondi failis olema peavad, seega nüüd tuli need vaid koos noodivõtme, helistiku ja taktimõõduga faili kirjutada. LilyPondi faili formaadi näide on toodud eelmises alapeatükis joonisel 10.

Kõikide mallide jaoks leidis autor sobivad lävendid erinevate piltidega katsetamise teel. Lävendid üritati määrata võimalikult madalad, et ka halva kvaliteediga piltide puhul oleks võimalik sümboleid tuvastada. Otsustati mitte lisada lävendite dünaamilist alandamist tuvastamise käigus, kuna juba natuke madalamate lävenditega aetakse sümboleid segamini ning tuvastatakse sümbolitena ka müra. Kui ühtegi sümbolit ei suudeta tuvastada, siis saadetakse kasutajale hoopis vastav teade ning palutakse teha uus pilt.

## 4 Tulemused

Noodituvastuse rakenduse täpsust hinnati erinevate piltidega katsetades. Mobiilirakenduse testimiseks paluti teistel kasutajatel seda kasutada ning avaldada oma arvamust. Kuigi tulemustega võib rahul olla, on rakendusel siiski palju edasiarendusvõimalusi.

### 4.1 Hinnang tuvastusele

Noodituvastusele on keeruline anda numbrilist hinnangut, kuna raske on määrata, milline on hea tuvastus ja milline halb – kas lugeda valeks kõik noodid, mis ei ole täiesti õigesti tuvastatud, või on poole tooni võrra valesti tuvastatud noot õigem kui kahe tooni võrra vale noot. Seetõttu otsustas autor tulemuse hindamiseks tuua näiteid erinevatest tuvastamise tulemustest ning neid analüüsida.

Kõige paremini saab rakendus hakkama LilyPondiga genereeritud piltidega. Sellisel juhul suudab rakendus peaaegu kõik tuvastatavad sümbolid koos helikõrgusega õigesti määrata. Joonisel 11 on toodud näide ühest lihtsast viisist „Meeri talleke”, mille noodituvastuse rakendus suutis õigesti tuvastada. Tuvastatud nootide nimetused ja vältused on lisatud nende alla.

The image shows two staves of musical notation for the piece 'Meeri talleke'. The top staff contains the original melody, and the bottom staff shows the same melody with pitch classes and accidentals identified below each note. The notes are: e'4, d'4, c'4, d'4, e'4, e'4, e'2, d'4, d'4, d'2, e'4, g'4, g'2, e'4, d'4, c'4, d'4, e'4, e'4, e'4, e'4, d'4, d'4, e'4, d'4. The bottom staff includes a c'1 note at the beginning and end.

Joonis 11. „Meeri talleke“ tuvastus

Kaameraga tehtud piltide puhul ei olnud rakendus tihti võimeline kõiki sümboleid tuvastama. Esiteks tuleb arvestada fontide erinevustega – need mõjutavad kõige rohkem just taktimõõtude ning teiste keerulisemate sümbolite nagu veerandpaus või diees tuvastamist. Samuti tekib probleeme, kui ei suudeta noodijooni õigesti tuvastada. Tihti juhtub, et mõni joon jääb tuvastamata ning sellisel juhul hakatakse noodijooni lugema kõige alumisest tuvastatud noodijoonest reas. Kui see aga tegelikult ei olnud kõige alumine joon, siis on kõikide nootide kõrgused selles reas nihkes.

Joonisel 12 on toodud näide kaameraga tehtud pildist, kus on lugu muusikalist "Helisev muusika", ning joonisel 13 on sama pildi tuvastamise tulemus.

muusikalist "Helisev muusika"  
 O. Hammerstein  
 R. Rodgers

CMaj7 D Am7 B

Em CMaj7

Vih-ma-piisk roo-sil ning kii-su lööb nur-ru, vask-pa-da  
 Pu-na-kad po-nid ja ha-be-mes kit-sed, uk-se-kell.

Am7 D7

plii-dil ja väi-ke-sed vur-rud, pruu-ni-kad pa-ken-did  
 kul-ju-sed, nuud-li-ga šnit-sel, rän-nul kui ha-ned on

G C G C Am6

rii-u-lil reas, need mi-nu lem-mi-kud se-da sa  
 len-da-mas reas,

Joonis 12. Kaameraga tehtud pilt loost "Lemmikasjad"

CMaj7 D Am7 B

Em CMaj7

Vih-ma-piisk roo-sil ning kii-su lööb nur-ru, vask-pa-da  
Pu-na-kad po-nid ja ha-be-mes kit-sed, uk-se-kell.

Am7 D7

plii-dil ja väi-ke-sed vur-rud, pruu-ni-kad pa-ken-did  
kul-ju-sed, nuud-li-ga šnit-sel, rän-nul kui ha-ned on

G C G C Am6

rii-u-lil reas, need mi-nu lem-mi-kud se-da sa  
len-da-mas reas,

Joonis 13. Tuvastuse tulemus loole "Lemmikasjad"

Noodijoontes on ridade alguses küll väiksed kõverused sees, kuid üldiselt on need piisavalt sirged, et noodituvastuse rakendus suudaks sealt üsna edukalt noodijooned ning sümbolid tuvastada. Siin on tuvastatud ka võtmemärk fis, seega f asemel on loos kasutatud kõrgendatud varianti. Esimese rea suudab rakendus tuvastada peaaegu veatult, vaid esimese ja viimase noodi kõrgustega on eksitud. Teises reas on pilt üsna udune ning noodipead on eeltötluse käigus muutunud hägusa pildi tõttu väiksemaks, seega paljud noodid jäävad siit tuvastamata. Esimese tuvastatud noodi kõrgus on tegelikust natuke kõrgem. Ka kolmandas ja neljandas reas on noote, mille kõrgus on natuke nihkes. See tuleneb sellest, et noodijoonestik pole päris sirge ning noodijoonte vahed on

võrdlemisi väiksed, seega noodipeale lähimat joont või vahet leides võib õige asukoht olla ühe piksli võrra kaugemal kui mõni teine.

## 4.2 Mobiilirakenduse testimine

Mobiilirakenduse testimiseks paluti kümnel kasutajal seda katsetada ning anda oma hinnang funktsionaalsusele ning disainile. Nii funktsionaalsuse kui disaini osas oli kasutajate seas erinevaid arvamusi. Kasutajatele anti rakenduse APK-fail installimiseks koos kahe näidispildiga (eelmises peatükis joonis 11 „Meeri talleke“ ja joonis 12 „Lemmikasjad“), mis on olemas ka autori koodihoidlas [22]. Kasutajatel paluti vastata vabas vormis järgmistele küsimustele:

- 1) Kas midagi on segast/arusaamatut? Kas disainis võiks midagi muuta?
- 2) Kas mingi funktsionaalsus võiks veel juures olla? Kas midagi on üleliigset?
- 3) Kas kasutaksite sellist rakendust (kui see tuvastaks täpselt)?
- 4) Kas on veel midagi tagasisideks lisada?

Rakenduse kujundus kasutajatele üldiselt meeldis. Toodi välja, et rakendusel on ilusad rahulikud värvid ning kasutajatele meeldis, et pärast pildi valimist kuvati pilti ka rakenduses. Icoonide osas jagunes kasutajate arvamus kaheks – suuremale osale meeldis ikoonide universaalsus ja arusaadavus, kuid mõned oleks soovinud ikoonidele lisaks selgitusi, mida erinevad nupud teevad. Disaini poolelt toodi välja ka see, et horisontaalses asendis rakendust kasutades on nupud liiga all ääres, kuid siiski nähtaval ja kasutatavad.

Funktsionaalsusega jäädigi ka üldjoontes rahule. Selgus aga asjaolu, et alates Androidi versioonist 10 ei toiminud MIDI-failide allalaadimine. Kasutajatel oli ka vastakaid arvamusi tuvastatud loo esitamisest. Nimelt jääb lugu taustal mängima, kui minna mõne teise rakenduse juurde ja jätta noodituvastuse rakendus käima, kui just lugu enne nupust pausile ei panda. Osad kasutajad leidsid, et see on häiriv, osad aga leidsid, et nii on just mugav tegeleda loo kuulamise ajal teiste asjadega.

Kasutajad tõid ka välja, et oleks mugav, kui rakendus salvestaks oma pildid ja MIDI-failid ühte kausta ning rakenduses oleks ka nupp, mis selle kausta avab. Hetkel salvestatakse tehtud pildid piltide alla ja alla laetud MIDI-failid rakenduse loodud noodituvastuse kausta. Lisaks pakuti välja

lisavõimalustega muusikamängija sisse ehitamist, et oleks võimalik lugusid ka kerida. Pooled kasutajatest kasutaksid sellist rakendust, kui see suudaks muusikat täpselt ja kiiresti tuvastada.

### **4.3 Edasiarendamise võimalused**

Nii noodituvastusrakendusel kui ka mobiilirakendusel on veel päris palju arenguruumi. Üheks suureks edasiseks etapiks on mitmehäälsete muusikapalade tuvastamine, aga enne seda on mõistlik parandada sümbolite ja noodijoonte tuvastamise täpsust ühehäälsel meloodial.

Noodituvastusrakenduse puhul on esimeseks sammuks eeltöötuse parandamine. Noodijoonte leidmisel saab katsetada veel teisi variante ning neid omavahel kombineerida, et saavutada võimalikult täpne tulemus. Hetkel leitakse rakenduses iga noodijoone jaoks tema y-koordinaat ning eeldatakse, et noodijoon on sirge. Täpsust aitaks ilmselt parandada noodijoonte hoopis joonevõrrandite leidmine. Kui iga rea puhul ei suudeta leida viit noodijoont, siis võib joonte ja ridade kaugusi võrrelda ning arvutuslikult leida, kust joon puudu on.

Tuvastuse puhul võib iga sümboli jaoks luua mitmeid malle, et oleks esindatud ka erinevad fondid. Samuti võib lahendada taktijoonte tuvastamise probleemi. Kui taktijoonte asukohad on teada, siis on ka teada, kui mõni löök on mõnest taktist puudu ning saab vajadusel lisada sellesse takti pause, pikendada mõnda nooti või proovida selle takti piires uuesti noote ja pause tuvastada. Alteratsioonimärke saab siis samuti kõikidele takti samal kõrgusel olevatele nootidele rakendada.

Klassifitseerimisel võib anda häid tulemusi närvivõrkude kasutamine. Sellisel juhul tuleb kõigepealt leida viis, kuidas erinevaid sümboleid pildil eraldada ning siis saab need eraldatud tükid anda närvivõrgule tuvastamiseks. Närvivõrgu jaoks saab kokku koguda erineva kvaliteedi ja fondiga pilte muusikasümbolitest.

Närvivõrkudega võib proovida ka algusest lõpuni kogu muusikapala tuvastamist, mitte ainult üksikute sümbolite klassifitseerimist. See ülesanne on aga üsna keeruline ning nõuab palju treeningandmeid ja treenimisaega. Autor katsetas ka ise sellist lähenemist ning genereeris 5000 lühikest meloodiat, kus noodid olid vaid ühe oktaavi piires ja ühe vältega, ning kirjutas skripti, mis moodustas nendest meloodiatest LilyPondi failid ja nendest omakorda PDF-failid ja pildid. Kõik pildid olid ühe suurusega ja noodijooned asusid täpselt samadel kohtadel. Närvivõrgu põhjaks

võeti piltide kirjeldamiseks mõeldud närvivõrk, kuna see ülesanne on oma olemuselt sarnane. Sellise väikse koguse andmetega ei saanud närvivõrk aga õppimisega hakkama ega suutnud ennustada pildi põhjal nootide järjestusi.

Kasutajaliidese edasiarenduseks saab viia sisse kasutajate pakutud muudatusi, mida on kirjeldatud ka eelmises alapeatükis. Rakenduses tehtud pildid ja tuvastatud MIDI-failid võib koondada ühte kausta ja teha see kaust kasutajale lihtsasti kättesaadavaks. Ühtlasi peab muutma failide allalaadimist, et see töötaks ka uuemate Androidi versioonidega. Täiendada võib ka muusikafailide esitamiseks mõeldud mängijat, lisades sinna loo kerimise võimaluse.

## Kokkuvõte

Bakalaureusetöö eesmärk oli luua noodituvastuseks veebirakendus, mis kasutab malltuvastust, ning mobiilirakendus, mis on kasutajaliideseks veebirakendusega suhtlemisel.

Töö raames loodi noodituvastuse veebirakendus, mis proovib tuvastada ühehäälsid meloodiaid kas viiuli- või bassivõtmes. Alustuseks tehti pildile eeltöötlus – pilt muudeti mustvalgeks, tuvastati noodijooned ja pildi suurust muudeti vastavalt vajadusele. Seejärel kasutati malltuvastust erinevate muusikaliste sümbolite tuvastamiseks ning lõpuks pandi nendest kokku terviklik muusikapala.

Loodi ka mobiilirakendus, mis võimaldab kasutajal telefoniga noodist pilti teha või see üles laadida, kuulata tuvastatud meloodiat ning muusikafaili MIDI-failina alla laadida.

Töö tulemusena valminud rakendus suudab edukalt tuvastada selgeid ja lihtsaid pilte, kuid kaameraga tehtud piltidega, millel on kehvem valgustus ja kus noodijooned ei ole täiesti sirged, võib tekkida raskusi. Töö käigus katsetati mitmeid erinevaid variante eeltöötluse ja tuvastamise jaoks ning valiti neist välja parimad. Rakendusel on palju edasiarendamise võimalusi, mis aitavad muuta tuvastust täpsemaks ja usaldusväärsemaks.



## Viidatud kirjandus

- [1] MusicXML, <https://www.musicxml.com/> (05.05.2020)
- [2] About MIDI-Part 1:Overview. The MIDI Association, 2015, <https://www.midi.org/articles-old/about-midi-part-1-overview> (05.05.2020)
- [3] Majumder M. R, Mahmud B. U., Jahan B., Alam M. Offline optical character recognition (OCR) method: An effective method for scanned documents. 22nd International Conference on Computer and Information Technology (ICCIT), 2019, [10.1109/ICCIT48885.2019.9038593](https://doi.org/10.1109/ICCIT48885.2019.9038593) (13.04.2020)
- [4] Chaudhuri A., Mandaviya K., Badelia P., Ghosh S. K. Optical Character Recognition Systems, Optical Character Recognition Systems for Different Languages with Soft Computing, p.9-41, 2017, [https://www.researchgate.net/publication/312273545\\_Optical\\_Character\\_Recognition\\_Systems](https://www.researchgate.net/publication/312273545_Optical_Character_Recognition_Systems) (04.04.2020)
- [5] Template Matching. Adaptive Vision, [https://docs.adaptive-vision.com/4.7/studio/machine\\_vision\\_guide/TemplateMatching.html](https://docs.adaptive-vision.com/4.7/studio/machine_vision_guide/TemplateMatching.html) (11.03.2020)
- [6] Rebelo A., Fujinaga I., Paszkiewicz F., Marcal A. R. S., Guedes C., Cardoso J. S. Optical music recognition: state-of-the-art and open issues. International Journal of Multimedia Information Retrieval volume 1, p. 173–190, 2012 <https://link.springer.com/article/10.1007/s13735-012-0004-6> (02.04.2020)
- [7] Rico A., Fornés A. Camera-Based Optical Music Recognition Using a Convolutional Neural Network. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017, [10.1109/ICDAR.2017.261](https://doi.org/10.1109/ICDAR.2017.261) (23.03.2020)

- [8] Wallner M. A System for Optical Music Recognition and Audio Synthesis. Faculty of Informatics, Vienna University of Technology Master's thesis, 2014, <https://www.semanticscholar.org/paper/A-System-for-Optical-Music-Recognition-and-Audio-Wallner-Thesis/f4f51cffaa1b6661e135aa3dedc26e5561e66578> (22.03.2020)
- [9] Quang Nhat Vo TNSHKHJYGSL. Distorted Music Score Recognition without Staffline Removal. 22nd International Conference on Pattern Recognition (ICPR), 2014, [10.1109/ICPR.2014.510](https://doi.org/10.1109/ICPR.2014.510) (03.04.2020)
- [10] Pacha A., Eidenberger H. Towards Self-Learning Optical Music Recognition. 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 2017, [1109/ICMLA.2017.00-60](https://doi.org/10.1109/ICMLA.2017.00-60) (03.04.2020)
- [11] Baró A., Riba P., Calvo-Zaragoza J., Fornés A. Optical Music Recognition by Recurrent Neural Networks. 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), [10.1109/ICDAR.2017.260](https://doi.org/10.1109/ICDAR.2017.260) (23.03.2020)
- [12] Baró A., Riba P., Calvo-Zaragoza J., Fornés A. From Optical Music Recognition to Handwritten Music Recognition: A baseline. PATTERN RECOGNITION LETTERS, 123 p1-p8, 8p, 2019, [10.1016/j.patrec.2019.02.029](https://doi.org/10.1016/j.patrec.2019.02.029) (03.04.2020)
- [13] PlayScore, <https://www.playscore.co/how-to-use/> (24.03.2020)
- [14] PlayScore Lite. Google Play, [https://play.google.com/store/apps/details?id=uk.co.dolphin\\_com.camrascorelite&hl=en\\_US](https://play.google.com/store/apps/details?id=uk.co.dolphin_com.camrascorelite&hl=en_US) (24.03.2020)
- [15] iSeeNotes, <http://www.iseenotes.com/> (05.05.2020)
- [16] iSeeNotes 72. Google Play,

- [https://play.google.com/store/apps/details?id=com.gearup.iseenotestimetrial&hl=en\\_US](https://play.google.com/store/apps/details?id=com.gearup.iseenotestimetrial&hl=en_US)  
(25.03.2020)
- [17] iSeeNotes, <https://play.google.com/store/apps/details?id=com.gearup.iseenotes&hl=et>  
(25.03.2020)
- [18] Muusikateooria õpik. Eesti Muusika- ja Teatriakadeemia, <http://mt.ema.edu.ee/> (05.05.2020)
- [19] Piispea M. Noodikirja põhielemendid, 2012,  
<http://eller.tmk.ee/~kaie.magimets/noodikiri1/index.html> (27.03.2020)
- [20] Flask, <https://flask.palletsprojects.com/en/1.1.x/> (05.05.2020)
- [21] Heroku, <https://www.heroku.com/> (05.05.2020)
- [22] Keps K. OMR. Github, <https://github.com/kristiinakeps/OMR> (05.05.2020)
- [23] Docker, <https://www.docker.com/> (06.05.2020)
- [24] Hough Line Transform. OpenCV, [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_houghlines/py\\_houghlines.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html)  
(05.05.2020)
- [25] Canny Edge Detection. OpenCV, [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_canny/py\\_canny.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html)  
(05.05.2020)

## Lisad

### I. Litsents

#### Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Kristiina Keps,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „**Optiline noodituvastus**“, 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.

*Kristiina Keps*

**07.05.2020**