

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Kevin Telliskivi**

**Arvutinägemise Python OpenCV õppemooduli  
loomine gümnaasiumile**

**Bakalaureusetöö (9 EAP)**

Juhendaja: Tauno Palts, PhD

Tartu 2024

## **Infoleht**

### **Arvutinägemise Python OpenCV õppemooduli loomine gümnaasiumile**

#### **Lühikokkuvõte:**

Töö eesmärgiks on luua gümnaasiumi õpilastele suunatud „Tarkvaraarendus“ valikkursuse arvutinägemise lisamoodul. Valikkursus on osa Eesti „Informaatika“ valikõppekavast. Loodud lisamoodul õpetab põhilisi arvutinägemise tehnikaid kasutades Pythoni OpenCV teeki. Õppematerjalide koostamisel jälgiti Merrilli mudeli eeskirja, mis toetab probleemikeskset õpet. Loodud tööle andsid tagasisidet informaatika õpetajad ning töö autor parandas tööd vastavalt. Õppematerjalid on leitavad Tartu Ülikooli Courses keskkonnast.

**Võtmesõnad:** Programmeerimise õpetamine, pildituvastus, Python OpenCV, informaatika, gümnaasium

**CERCS:** P175 Informaatika, S270 Pedagoogika ja didaktika, S281 Arvuti õpiprogrammide kasutamise metoodika ja pedagoogika,

### **Computer vision programming module creation with Python OpenCV for secondary school**

#### **Abstract:**

The aim of this study is to create computer vision study materials for programming course “Software development” taught in Estonian high schools. The elective course is part of the Estonian "Informatics" curriculum. The created module teaches basic computer vision techniques using the Python OpenCV library. When compiling the study materials, the guidelines of the Merrill model were followed, which support problem-centered teaching. Feedback on the created work was provided by computer science teachers, and the author revised the work accordingly. The study materials can be found on the Tartu University Courses platform.

**Keywords:** Teaching programming, image recognition, Python OpenCV, informatics, secondary school

**CERCS:** P175 Informatics, S270 Pedagogy and didactics, S281 Computer-assisted education

## Sisukord

Sissejuhatus .....	4
1. Programmeerimise õpetamine ja arvutinägemine .....	5
1.1. Programmeerimise õpetamise olulisus .....	5
1.2. Arvutinägemine ja kasutusala .....	6
1.2.1. Arvutinägemise teek OpenCV .....	7
1.3. Gümnaasiumi informaatika õppekava .....	7
1.3.1. Arvutinägemine gümnaasiumi informaatika õppekavas .....	8
1.4. Õppematerjalide koostamise mudel .....	9
2. Metoodika .....	11
2.1. Õppematerjalide koostamise protsess .....	11
2.2. Tagasiside kogumine .....	11
3. Tulemused .....	12
3.1. Valminud õppematerjalid .....	12
3.2. Materjalide koostamine Merrilli mudeli põhjal .....	17
3.3. Õpetajate tagasiside materjalidele .....	19
3.4. Edasised võimalused .....	21
Kokkuvõte .....	22
Viidatud kirjandus .....	23
Lisad .....	25
I. Link valminud OpenCV õppematerjalidest .....	25
II. Litsents .....	26

## Sissejuhatus

Me elame üha enam digitaliseerivas maailmas, mis toob endaga kaasa ka muutused tööturul. Ennustatakse, et käimasoleval kümnendil kasvab informaatika valdkonna tööturg 22% [1]. Lisaks vajab juba praegu 92% töökohtadest digioskusi [2]. On oluline, et noored õpivad digioskusi ja saavad soovi korral juba enne ülikooli astumist alusteadmised informaatikast.

Eestis õpetatakse gümnaasiumis informaatikat valikõppeainena ja selle kontseptsioon kaasajastati 2017. aastal Haridus- ja Teadusministeeriumi tellimusel HITSA poolt [3]. Õppekava loomisel lähtuti metoodikast, et see arendaks nii raal- kui disainmõtlemist ja oleks projektipõhine [4]. Õppekava on jaotatud viieks valikkursuseks, mille hulgas on kursus „Tarkvaraarendus“. Töö eesmärk on täiendada gümnaasiumi programmeerimise „Tarkvaraarendus“ valikkursust õppemooduliga, mis käsitleb arvutinägemist, kasutades selleks Pythoni vabavaralist teeki OpenCV. Arvutinägemine on teadusharu, mis keskendub informatsiooni eraldamisele digitaalsetest piltidest või videotest ning seda kasutatakse tänapäeval väga laialdaselt erinevates eluvaldkondades [5].

Autor valis teema 2022. aasta sügisel Tartu Ülikooli Informaatika Didaktika töörühma pakutavatest teemadest. Autori valik osutus arvutinägemise temaatikale, sest tundis ise huvi selle vastu ning soovis panustada eestikeelsesse informaatika haridusse, õppides ise samuti pildituvastuse kohta.

Bakalaureusetöö koosneb kolmest osast: Programmeerimise õpetamine ja arvutinägemine, metoodika ja tulemused. Esimeses peatükis räägitakse, miks programmeerimise õpetamine on vajalik, kirjeldatakse, milline on hetkel Eesti gümnaasiumi informaatika ainekava ning millised valikkursused on läbimiseks saadaval. Metoodika peatükis kirjeldatakse, kuidas valmis „Tarkvaraarenduse“ lisaõppemoodul arvutinägemine Python OpenCV-ga. Lisaks kirjeldatakse tagasiside kogumise protsessi ja analüüsi. Tulemuste peatükis tutvustatakse valminud õppematerjale. Antakse ülevaade lisamooduli struktuurist ning kirjeldatakse igat peatükki. Tulemuste osas lähtutakse korrektsest õppematerjalide koostamisest, tuues näiteid, kuidas Merrilli mudel juhiseid on jälgitud. Lõpuks analüüsitakse tagasiside tulemusi ja kirjeldatakse, kuidas anti õppematerjalidele viimane lihv. Kõige lõpuks räägitakse edasiarenduse võimalustest. Lisade loetelus on link valminud materjalidele.

# 1. Programmeerimise õpetamine ja arvutinägemine

See peatükk annab ülevaate programmeerimise õpetamise olulisusest ja selle hetkeseisust Eesti õppekavades. Lisaks annab peatükk ülevaate arvutinägemisest ja selle rakendamiseks kasutatavast Pythoni teegist OpenCV. Lõpuks tutvustatakse õppematerjalide koostamiseks kasutatavat Merilli mudelit.

## 1.1. Programmeerimise õpetamise olulisus

Programmeerimist võib lugeda üheks 21. sajandi oskuseks, mis on paljudel töökohtadel nõutavate oskuste alustalaks. Nimelt 58% digitööstuse töökohtadest nõuavad programmeerimise oskust ning viimasel kümnendil on nõudlus selle oskuse järgi kasvanud 50% kiiremini kui üldine nõudlus töökohtadele [6]. Lisaks on programmeerimise oskus väga kasulik ja toetav ka mitte otseselt programmeerimisega seotud töökohtadel. Programmeerimise oskuse olulisust mõistavad ka noored - uuringu kohaselt üle poolte USA õpilastest peavad tulevaste karjäärivõimaluste jaoks programmeerimise oskust sama oluliseks kui võõrkeelte oskust [7].

Digitaalmaailm on tänapäeva ühiskonna paratamatu osa ja noored kasvavad üles olles pidevalt ümbritsetud tehnoloogiast ja kasutavad seda oma igapäeva elu toiminguteks. Kasvuaeg ollakse motiveeritud tundma õppima ümbritsevat maailma ja seejuures ollakse huvitatud tehnoloogiaseadmetest ja nende programmidest. Inimesed on juba varases eas teadlikud, et digitaalsete seadmeid kontrollivad spetsiaalsed programmid. Programmeerimise õppimine annab noortele ettekujutuse sellest, kuidas tehnoloogia töötab ja "suhtleb" inimesega ning aitab ühendada õpitu reaalse maailmaga ning võimaluse neid teadmisi päriselus rakendada [8].

Lisaks aitab programmeerimine arendada raalmõtlemist (inglise k. *computational thinking*) [9]. Raalmõtlemine kontseptsioonina ongi pärit arvutiteadusest ja see tähendab laiemalt probleemide lahendamise viisi, mis hõlmab endas ülesande oluliste aspektide eristamist, nende jagamist väiksemateks osaülesanneteks ja lahenduseni jõudmist, mida programmeerimise näitel täidab arvuti [9]. Probleemilahenduse oskuse arenemisega kaasneb oskus iseseisvalt otsuseid langetada ja raskustest üle saada, näiteks programmi testimisel võidakse saada valesid tulemusi või programm ei tööta ja vea kõrvaldamiseks peavad õpilased tihti mitmeid kordi proovima. [8]. Samas artiklis väideti, et programmeerimine aitab õpetada ka planeerimisoskust - seda nii programmi luues, kui otsustatakse koodijuppide järjekorda või planeerides üldist projektikulgu.

Programmeerimise õppimine toetab lisaks laiemalt eluks vajalikke oskusi. Tänapäeva esmavajalike oskuste hulka ei kuulu enam ainult teoreetilised teadmised, vaid esile on kerkinud uued oskused, mida õpilased peavad omandama, et tulevikus olla edukad ühiskonna liikmed. Neid teadmisi nimetatakse inglise keelest *Four C-s*, mille hulka kuuluvad kriitiline mõtlemine, suhtlusoskus, koostöö ja loomingulisus [8]. Programmeerimisega areneb loomingulisus, sest sellel on oluline osa probleemide lahendamisel, mis võivad programmeerides ette tulla. Lisaks õpetab programmeerimine koostööd ja suhtlusoskust, sest enamasti luuakse programme rühmades.

## **1.2. Arvutinägemine ja kasutusala**

Arvutinägemine on teadusharu, mis keskendub informatsiooni eraldamisele digitaalsetest piltidest või videotest [5]. Arvutinägemises treenitakse arvutit mõistma pildi- või videosisu, et seejärel oskaks arvuti selle infoga midagi edasi teha. Arvutinägemine sarnaneb inimese nägemisega, mõlemad eristavad ja iseloomustavad objekte, kuid erinevalt inimesest, arvutinägemine kasutab selleks kaamerat, andmeid ja algoritme. Arvutinägemise eelis inimese ees on just selle kiirus. Näiteks kindla protsessi jaoks treenitud süsteem võib ära tunda tuhandeid defektseid tooteid minutis.

Arvutinägemine on olnud teadusharuna olemas juba ligi 50 aastat, kuid siiani on objektide eristamine ja tõlgendamine, mida inimesed teevad alateadlikult, arvutitele väga keeruline [5]. Antud allikas väidetakse, et see on seetõttu, et esineb suur vahe pildi pikslite ja selle tähenduse vahel. Kui arvuti näeb 200 x 200 piksliga RGB pilti, siis see koosneb 120 000 väärtusest, millest peab arvuti korrektse informatsiooni eraldama.

Arvutinägemine on laialdaselt kasutusel erinevates valdkondades, sealhulgas tootmises, autotööstuses, tervishoius, turvateenustes ja meelelahutuses [10]. Antud allikas toodud näited on edaspidi refereeritud siin lõigus. Näiteks turvateenuste valdkonnas kasutatakse arvutinägemist valvekaamerate süsteemides. Lisaks kasutab Google arvutinägemist veebis olevate piltide ja videote puhul ning enamik õhu- ja tänavakaardi pilte (nagu Google Street View's) kasutavad intensiivselt erinevaid kaamerate kalibreerimise ja pildi õmblemise tehnikaid. Väga laialt on arvutinägemine levinud tootmises, kus praktiliselt kõik masstoodetud esemed on mingis etapis läbinud kvaliteedikontrolli kasutades arvutinägemist. Lisaks kasutatakse arvutinägemist mänguliidestest ning sellel on ka niširakendusi turvavaldkonnas, mehitamata lennumasinatel ja biomeditsiinilises analüüsis.

### 1.2.1. Arvutinägemise teek OpenCV

OpenCV (inglise k. *Open Source Computer Vision Library*) on avatud lähtekoodiga arvutinägemise ja masinõppe tarkvara teek, mis loodi eesmärgiga võimaldada head infrastruktuuri arvutinägemise rakendustele ning kiirendada masinnägemise kasutust laiatarbe projektides [11].

OpenCV kodulehelt [11] saab lugeda, et teegis on kättesaadaval üle 2500 optimeeritud algoritmi, mille hulgas on nii traditsioonilisi kui ka kaasaegseid arvutinägemise teadusharus levinud algoritme. OpenCV teek pakub vahendeid ja algoritme piltide ja videote töötlemiseks, et tuvastada objekte, jälgida liikumist ning teostada näo- ja žestituvastust. OpenCV on kirjutatud C++ keeles, kuid seda saab kasutada ka teistes programmeerimiskeeltes nagu Python ja Java tänu pakutavatele sidusliidestele.

OpenCV sisaldab mitmeid funktsionaalsusi, mille hulgas levinumad on [12]:

1. **Pilditöötlus:** filtrid, pikslite teisendamine, värviruumi muutmine;
2. **Objekti tuvastamine:** näotuvastus, objektide jälgimine;
3. **Masinõpe:** mudelite treenimine pildituvastuseks;

OpenCV on väga kasutajasõbralik olles üks populaarsemaid arvutinägemise teeke maailmas – tänaseks päevaks on sellel üle 18 miljoni allalaadimise ning see on kasutusel ka tuntud ettevõtetes nagu Google, Microsoft, Intel ja Yahoo [11].

### 1.3. Gümnaasiumi informaatika õppekava

Eestis riiklikus gümnaasiumi õppekavas on “Informaatika” valikõppeaine [13]. Õppekava ütleb, et õpetamise peamiseks eesmärgiks on anda informaatikast huvi tundvatele õpilastele erialased teadmised, et neil oleks võimalus juba gümnaasiumis planeerida oma tulevast karjääri selle teemaga seotud valdkondades. Lisaks on õppe-eesmärgiks äratada üldist huvi informaatika ja selle valdkonna töökohtade vastu.

Gümnaasiumiastme informaatikaõppe kontseptsioon kaasajastati 2017. aastal Haridus- ja Teadusministeeriumi tellimusel HITSA poolt [3]. Eelmises lauses viidatud kontseptsiooni dokumendis on kirjas, et seda tehti eesmärgil säilitada Eesti ühiskonna arengupotentsiaali läbi

õpilastele võimaluse andmise omandada info- ja kommunikatsioonitehnoloogiaga seonduvad 21. sajandi oskused. Dokumendi on kirjas, et visiooni kohaselt on tagatud 2030. aastaks kõikidele gümnaasiumilõpetajatele oskused toime tulla kiiresti muutuvast tehnoloogiamaailmas ja vajalikud digipädevused. See on oluline just seetõttu, et tänapäeva digitaliseerunud ühiskonnas puututakse digitehnoloogiaga kokku igal töökohal ja ka väljaspool tööd. Õppekava loomisel lähtuti metoodikast, et see arendaks nii raal- kui disainmõtlemist ja oleks projektipõhine [4].

Järgnev lõik on refereeritud riiklikust õppekavast [13]. “Informaatika” õppeaine keskmeks on rühmatööna loodav uurimuslik digilahenduse arendusprojekt ja see eeldab rühmaliikmetelt eelteadmisi tema vastava rolli sisu kohta. Eelteadmiste omandamiseks on gümnaasiumiõppekavas valikkursused: 1) „Programmeerimine“; 2) „Tarkvaraarendus“ (mille eelduseks on programmeerimise kursuse läbimine või vastavate oskuste omandatus juba põhikoolis); 3) "Tarkvaraanalüüs ja testimine;" 4) "Kasutajakeskne disain ja prototüüpimine;" 5) "Digiteenused." Lisaks nendele kursustele on valikus 2011. aastal ilmunud valikkursused. Projektõpe lähtub nüüdisaegsest õppemetoodikast ja toetab õpilastes enesejuhtimise, ettevõtlikkuse, loovuse ja koostöö oskuste arendamist.

### **1.3.1. Arvutinägemine gümnaasiumi informaatika õppekavas**

Riiklikus gümnaasiumi “Informaatika” valikõppeaines on kursus “Tarkvaraarendus” ning selle üks loodud lisamoodulitest on pildituvastusest Pythonis, mis käsitleb arvutinägemist. „Tarkvaraarendus“ kursuse ja lisamoodulite sisu on kirjeldatud õpikus [14]. Selle lisamooduli läbimiseks on vajalikud eelteadmised, mida saab omandada põhikursuse “Tarkvaraarendus” läbimisel. Tarkvaraarenduse kursus koosneb kahest osast. Esimeses osas keskendutakse klassikalistele programmeerimistemadele nagu kahemõõtmeline järjend, kahekordne tsükkel, erinevad teised andmestruktuurid ja rekursioon. Teises osas tegeletakse tarkvara arendamisega, milles saadakse ülevaade tarkvaraarenduse etappidest ja luuakse ise enda valitud mooduli abil tarkvararakenduse prototüüp. Tarkvaraarenduse projekti raames on vajadus lisamoodulile projekti tegemiseks.

Pythoni pildituvastuse lisamoodulit tutvustati 2023. aasta alguses, mis telliti Hariduse ja Noorteameti poolt koos teiste tarkvaraarenduse kursuse lisamoodulitega. Lisamoodulit tutvustaval veebiseminaril [15] räägiti, et mooduli pikkus on 10 õppetundi ja valminud lisamooduli õppematerjal kasutab koodi jooksumiseks Jupyter Notebook'i - töövahend,



milles saab kirjutada Pythoni koodi ja ka tavateksti. Selle eeliseks on võimalus kasutada seda läbi veebibrauseri ja jooksutada koodi nii serveri pilves kui ka oma arvutis. Järgnev õppemooduli sisu kirjeldus on kirjutatud „Tarkvaraarenduse“ õpiku põhjal [14]. Pythoni pildituvastuse lisamoodul koosneb kuuest peatükist:

- JupyterLab / Jupyter Notebook kasutamine Pythoni koodi jooksutamiseks;
- Konvolutsiooniline närvivõrk (inglise k. *Convolutional Neural Network*);
- Närvivõrgud ja PyTorch;
- Pildi kategoriseerimine masinõppega - protsessi ülevaade;
- Praktiline näide pildi klassifitseerimise kohta;
- Piltide klassifitseerimine.

Mooduli fookuses on teemadena närvivõrgud ja pildi kategoriseerimine. Teemade käsitlemisel on antud enamasti ajalooline ja sisuline ülevaade teemast, koos mitmete linkidega edasi lugemiseks. Lisaks on teemasid osati seletatud läbi matemaatika, kuid autori sõnul prooviti seda teha võimalikult vähesel määral [15]. Teemadest arusaamise kontrollimiseks on peatükkides lühikesed kontrollülesanded ja koodinäited. Viimased kaks peatükki on näited, kus õpilane saab pildituvastuse masinõppe mudelit treenida ja seda testida. Mooduli autor mainis õppematerjali esitledes, et mooduli keerukuseks osutus selle läbimise ajakulukus (nii mudelite treenimine kui ka keskkonna seadistus) ja gümnaasiumiõpilasele keeruline sisu [15]. See ajendas looma samal teemal gümnaasiumiõpilasele sobivamal tasemel lühema ajaga läbitavat lisamoodulit.

#### **1.4. Õppematerjalide koostamise mudel**

Õppematerjali koostamisel on oluline, et see sobituks kaasajastatud “Informaatika” õppekava kontseptsiooniga ja kasutaks nüüdisaegseid õppemeetodeid. Sellest lähtudes paistab silma haridusteadlase M. David Merrilli loodud "Õpetamise põhiprintsiibid" (inglise k. *First Principles of Instruction*), mis ühendab mitmed õppematerjalide koostamise teooriad ühtseks erinevatest osadest koosnevaks mudeliks [16]. Merrill väitis samas allikas, et õppematerjalide koostamise eeskiri on universaalne ning ei sõltu õpetava materjali sisust.

Merrill leidis, et kõige efektiivsem õpe on probleemikeskne ja õpe peaks toimuma läbi viie selgelt eristuva faasi: probleemikesksus, aktiveerimine, demonstreerimine, rakendamine ja integratsioon [16]. Allolevad punktid selgitavad sama allika põhjal neid viite etappi.

1. **Problem:** selles etapis esitatakse õpieesmärke läbi reaaleluliste probleemide. Seeläbi suudavad õpilased probleemiga suhestuda ja on tänu sellele õppetöösse kaasatumad. See tagab, et õpilased tajuvad õpitava väärtust ja saavad õpitut tulevikus potentsiaalselt rakendada.
2. **Aktivatsioon:** selles etapis aktiveeritakse eelnevad teadmised, mis on uute õpitavate teadmiste aluseks näiteks vahetestide abil. See aitab luua seoseid uue materjali ja juba olemasolevate teadmiste vahel, seeläbi vähendades õpilase jaoks õpitava abstraktsust ja see aitab neil hõlpsamini materjali omandada.
3. **Demonstreerimine:** selles etapis õpilased omandavad kontseptsioone paremini, kui neile tuuakse näide selle toimimisest. Seega on oluline demonstreerida õpetatavaid teadmisi või oskusi näidete abil.
4. **Rakendamine:** Õpitavast täielikult arusaamiseks on oluline, et õpilased rakendavad saadud uusi teadmisi probleemide lahendamisel. Seda saab teha läbi ülesannete, mis nõuavad uute teadmiste rakendamist.
5. **Integratsioon:** Viimane põhimõte keskendub sellele, et õpilastel oskaksid rakendada uusi teadmisi ka reaalses elus. Kasulik oleks luua õpilastele ülesanne, mis esineb ka tavaelus.

## **2. Metoodika**

See peatükk annab ülevaate õppematerjalide loomise metoodikast, lähtudes varasemalt mainitud Merrilli mudelist õppematerjalide koostamise eeskirjast. Lisaks kirjeldatakse, kuidas koguti tagasisidet.

### **2.1. Õppematerjalide koostamise protsess**

Arvutinägemise materjalidega hakati tutvuma 2022. aasta sügisel, mil töö autor võttis eesmärgiks tutvuda Pythoni OpenCV vabavaralise teegiga. Arutati juhendajaga, millistest osadest võiks õppematerjal koosneda. Jõuti järeldusele, et mõistlik oleks kasutada Thonny programmeerimise keskkonda ning õppematerjalide ülesehitusel peaks lähtuma Merrilli mudelist, sest „Tarkvaraarenduse“ teiste programmeerimise peatükid oli samuti ehitatud Merrilli mudeli struktuurile. Leiti, et loodav lisamoodul peab toetama „Tarkvaraarendus“ aine raames loovat projekti. Järk-järgult hakati Tartu Ülikooli Courses leheküljele lisama materjale, mis kõnetaksid gümnaasiumi õpilasi. Õppematerjalide koostamisel koguti inspiratsiooni OpenCV dokumentatsioonist [12] ja OpenCV veebilehe [17] sissejuhatavast kursusest. Valminud õppematerjal on kombinatsiooni inglise keelsete kursuste tõlkest ning autori enda loomingust.

### **2.2. Tagasiside kogumine**

Õppematerjalide autor kohtus juhendajaga ja sai õppematerjali kohta tagasisidet jooksvalt juba õppemooduli valmimise käigus. Tagasisidet anti töö valmimise käigus jooksvalt Google Docs'i tekstiredaktoris ning veebikoosolekul. Kui õppematerjalide esimene versioon sai valmis, koguti kvalitatiivset tagasisidet informaatika õpetajatelt. Õpetajatel paluti vabas vormis kirjutada õppematerjalide plusside, miinuste ja soovitude kohta. Tagasiside põhjal viidi materjalides sisse muudatused.

Tagasisidet andnud õpetajate taust:

1. Informaatika didaktika lektor ülikoolis ja gümnaasiumis õpetamise kogemus.
2. Arvutinägemise nooremteadur, ülikooli õppejõud.
3. Informaatika- ja matemaatikaõpetaja eriala lõpetaja, õpetaja põhikoolis

### 3. Tulemused

Selles peatükis antakse ülevaade töö autori koostatud arvutinägemist käsitlevast õppemoodulist „OpenCV“ (lisa 1) ja demonstreeritakse peatükkide ülesehitust Merrilli mudeli põhjal ühe peatüki näitel. Lisaks kirjeldatakse õppematerjalile saadud tagasisidet ja järeltegevusi. Viimasena pakutakse võimalusi töö edasiarenduseks.

#### 3.1. Valminud õppematerjalid

Õppemoodul koosneb 9 peatükist. Õppematerjalide arvestuslik läbimise maht on 8 tundi ning seda saab läbida õpetaja juhendamisel või iseseisvalt. Õppematerjale on soovituslik läbida järjekorras.

Järgnevalt on kirjeldatud igat õppemooduli peatükki:

##### 1) OpenCV sissejuhatus

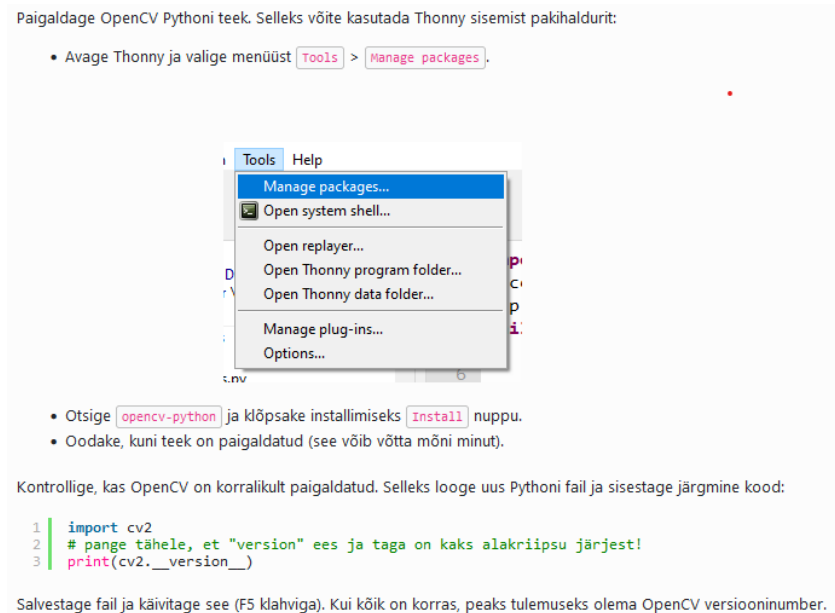
Esimene õppemooduli peatükk annab ülevaate OpenCV teegist. Peatükis kirjeldatakse terve lisamooduli ülesehitust, mida on näha joonisel 1. Lisaks kirjeldatakse õpitulemusi, mis peale selle lisamooduli läbimist saadakse.



Joonis 1. Õppemooduli ülesehitus.

## 2) OpenCV paigaldamine

Teises peatükis näidatakse, kuidas paigaldada OpenCV Pythoni teek Thonny'sse (joonis 2). Töö autor valis programmeerimise õpetamise keskkonnaks Thonny, sest sellega on õpilased varasemalt kursis. Peatüki lõpus saavad õpilased testida, kas teek sai õigesti paigaldatud.



Joonis 2. OpenCV paigaldus arenduskeskkonda Thonny.

## 3) Vajalike materjalide allalaadimine

Kolmas peatükk on õppemooduli kõige lühem peatükk sisaldades Google Drive linki. Lingi kaudu saavad õpilased alla laadida õppematerjali läbimiseks vajalikud failid.

## 4) Mis on pilt?

Peatükk algab sissejuhatava teooriaosaga, kus selgitatakse õpilastele, kuidas arvuti pilti näeb. Lisaks selgitatakse lahti olulised mõisted ning illustreeritakse teooriat näitega, kuidas näeb arvuti digitaalset pilti kahemõõtmelise järjendina (joonis 3).



```

import cv2

# Alusta videovoogu
kaamera = cv2.VideoCapture(0) # 0 on tavaliselt vaikimisi kaamera

# Kontrolli, kas kaamera on korrektselt avatud
if not kaamera.isOpened():
    print("Ei suuda kaamerat avada")
    exit()

while True:
    # Loe kaamerast uus pilt
    töötab, kaader = kaamera.read()

    # Kui pildi lugemine ebaõnnestub, lõpeta
    if not töötab:
        print("Ei suuda kaamerast pilti lugeda")
        break

    # Näita pilti aknas
    cv2.imshow('Kaameravoog', kaader)

    # Vajuta 'q', et väljuda tsüklist
    if cv2.waitKey(1) == ord('q'):
        break

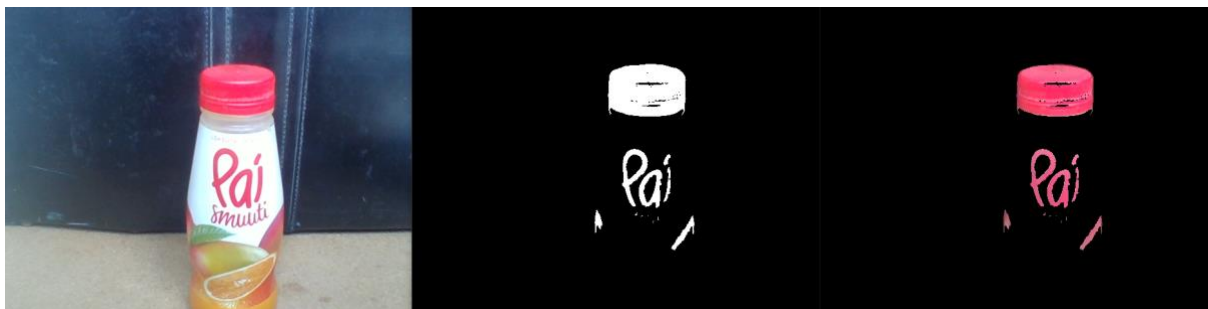
# Vabasta kaamera ja sulge kõik aknad
kaamera.release()
cv2.destroyAllWindows()

```

Joonis 4. Kaamera kasutamise koodi näide.

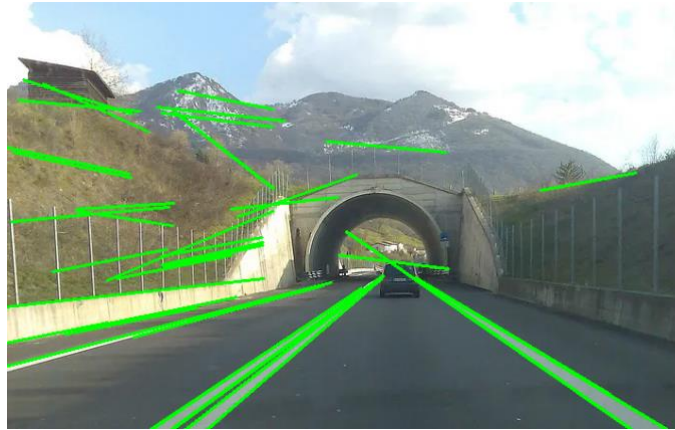
## 7) Pilditöötlus

Selles peatükis käsitletakse mõningaid piltide töötlemise võimalusi, mis võimaldavad saada pildilt kätte vajaliku informatsiooni. Peatüki saab jaotada kaheks temaatikaks. Esimene pool on värvide tuvastamine pildilt ja teine pool on joonte tuvastamine piltidelt. Peatükis on näidiskood, kuidas punast värvi läbi kaamera tuvastada (joonis 5). Koodi lõpus on antud ka infot, kuidas teisi värve ära tunda.



Joonis 5. Punase värvi kuvamine kaameraga.

Teises poole on näide äärte tuvastus algoritmist, kuidas kiirelt pildi pealt kõik ääred üles leida (joonis 6). Eesmärk on anda kiire ülevaade OpenCV võimalustest, mille programmeerimine algul võib tunduda keeruline.



Joonis 6. Joonte leidmine pildilt.

Viimaks jäeti õpilasele ülesandeks muuta pildil olevate joonte otsimispiirkonda, eesmärgiga leida neid ainult sõiduteelt.

## 8) Objektide tuvastus

Objektide tuvastuse peatükk on õppemooduli kõige mahukam peatükk. See peatükk võtab kokku ka varasemalt õpitud funktsioonid ning õpetab kuidas kasutada juba OpenCV's eeltreenitud objektide klassifikaatoreid. Õpilastele tutvustatakse olemasolevaid klassifikaatoreid ning järk-järgult hakatakse neid kasutama. Kõigepealt kasutatakse näotuvastust pildidel (joonis 7) ning see järel videotel. Õpilastel on vaheülesanne, kus nad peavad iseseisvalt kasutama mõnda teist saadaval olevat klassifikaatorit.

```
import cv2

näotuvastus = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
pilt = cv2.imread('inimesed.jpg')
gray = cv2.cvtColor(pilt, cv2.COLOR_BGR2GRAY)
näod = näotuvastus.detectMultiScale(gray, 1.3, 5)

for (x, y, w, h) in näod:
    pilt = cv2.rectangle(pilt, (x, y), (x+w, y+h), (255, 0, 0), 2)

cv2.imshow('Tuvastati inimesed!', pilt)
cv2.imwrite('./tuvastatud_inimesed.jpg', pilt)
cv2.waitKey(0)
```

Joonis 7. Näotuvastus pildilt.

Objektide tuvastuse teise ülesandega palutakse hakata tuvastama videotelt objekte. Lõpuks antakse tavaelus esinev ülesanne luua turvakaamera, mis tuvastab video pealt objekte ning hakkab videot salvestama, kui objekt on pildil on tuvastatud.



## 9) Lisa: Panoraam

Viimane õppepeatükk on lisapeatükk. Selles peatükis õpetatakse, kuidas siduda erinevaid pilte üheks panoraampildiks (joonis 8).



Joonis 8. Panoraam pildi tegemine.

### 3.2. Materjalide koostamine Merrilli mudeli põhjal

Õppematerjalide koostamisel jälgiti Merrilli mudeli viite printsiipi. Alljärgnevalt on toodi peatüki „Objektide tuvastus“ põhjal näide (peatükis 1.5 nimetatud) eeskirjade järgimisest.

**1) Probleem.** Esitatakse probleem, et arvuti treenimine objektide tuvastuse on ajamahukas töö. Mainitakse õppematerjalides, et alati ei pea asju nullist treenima ning võimalik on kasutada eeltreenitud mudeleid. Tihtipeale võtab see palju aega ja arvutusvõimsust. Soovime selle peatükiga näidata, et alati ei pea nullist mudeleid treenima. OpenCV’l on võimalust kasutada eeltreenitud mudeleid (joonis 9). Sõnastatakse probleem ja uue info saamise eesmärk.

#### Eesmärk

Arvutile on võimalik objektide tuvastust õpetada, kuid see on tihtipeale ajamahukas protsess, sest eeldab suures koguses näidispilete analüüsimist. Kui soovime, et arvuti suudaks piltidelt koerad ära tuvastada, siis see nõuab kõigepealt arvutile suurel hulgal koerte piltide näitamist. Arvuti loob piltide põhjal endale seosed ning alles siis on võimalik objektide tuvastamisega pihta hakata. Mida rohkem on näidisandmeid, seda paremini tuvastab arvuti objekte. OpenCV’l on olemas eeltreenitud mudeleid, mida saab kiiresti kasutada teatud liiki objektide tuvastamiseks, seega ei pea ise arvuti õpetamisele aega panustama.

Selles peatükis:

- Tutvume OpenCV moodulis olevate klassifikaatoritega.
- Kasutame Haari klassifikaatoreid objektide tuvastamiseks pildidel ja videotel.

Joonis 9. Probleemi sõnastus.

**2) Aktivatsioon.** Kasutatakse pildi avamise ja salvestamise teadmiseid, mida õpiti õppemooduli eelmistes peatükkides. Lisaks on ülesannetes vajalik *if* ja *while* laused, millega

on õpilased varem tutvunud. OpenCV õppemooduli läbimisel on vajalik läbida materjale õiges järjekorras, sest uutes peatükkides kasutatakse eelmistes alampeatükkides õpitud teadmisi.

**3) Demonstratsioon.** Näide koodijupist, kuidas haar klassifikaatorit kasutada. Samuti peale koodijupi näidet on selgitus koodi töötamisest (joonis 10). Õppimine on soodustatud, kui õpilastele tuuakse näiteid.

```
import cv2
# loeme sisse näotuvastuse faili
näotuvastus = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
# loeme sisse pildi, kus soovime nägusid tuvastada
pilt = cv2.imread('inimesed.jpg')
# Haar klassifikaator analüüsib ainult grayscale filtriga pilte
hall = cv2.cvtColor(pilt, cv2.COLOR_BGR2GRAY)
# näotuvastuse rakendamine
näod = näotuvastus.detectMultiScale(hall, 1.3, 5)

for (x, y, w, h) in näod:
    # joonistame ristkülikud ümber tuvastatud nägude
    pilt = cv2.rectangle(pilt, (x, y), (x+w, y+h), (255, 0, 0), 2)

cv2.imshow('Tuvastati inimesed!', pilt)
# salvestab tuvastatud näod uue failina
cv2.imwrite('./tuvastatud_inimesed.jpg', pilt)
cv2.waitKey(0)
```

Joonis 10. Koodi demonstratsioon.

**4) Rakendamine:** Õpilasel palutakse võtta teine eeltreenitud mudel ning proovida uue pildi peal iseseisvalt silmatuvastust läbi viia (joonis 11). Õpilased rakendavad uus omandatud teadmisi probleemi lahendamisel.

Ülesanne: Kirjuta programm, mis kasutab `haarcascade_eye.xml` klassifikaatorit, et tuvastada `inimene2.jpg` pildilt silmad.

Joonis 11. Materjali rakendamine.

**5) Integreerimine:** Materjalide läbimisel antakse ülesanne teha päriseluline turvakaamera, mis tuvastab õpilase enda valikult videost kindla objekti, kui objekt on videopildis, siis hakkab kaamera seda salvestama ning filmitakse senikaua, kuni objekt on vaateväljas (joonis 12). Tegu on mitme varasema teadmise põimimisega ning õpilane lahendab päriselu probleemi.

**Ülesanne:** Kirjuta programm, mis proovib kaamera videopildist tuvastada objekte. Lisaks peab hakkama kaamera salvestama videot, kui objekt tuvastatakse reaajas videolt. Kasutada võib kõiki saadaval olevaid OpenCV Haar klassifikaatoreid.

Joonis 12. Tavaelul põhinev ülesanne

### 3.3. Õpetajate tagasiside materjalidele

Õppematerjale tagasisidestati jooksvalt nende valmimisel Google Docsi või veebikohtumistel. Koguti kvalitatiivset tagasisidet ja õppemoodulile andsid omapoolse arvamuse kolm informaatika õpetajat.

Peamised kommentaarid (tabel 1) olid seotud tähelepanekuga teha iga õppepeatüki struktuur sarnaseks, et õpilastel oleks lihtsam materjali omandada. Autor sai tagasisidet, et osad terminite seletused võivad jääda gümnasistidele arusaamatuks. Need mõisted kirjutati selgemini lahti. Funktsiooni süntaksite kirjeldused parandati ühtlasemaks. Teksti struktuuri kohendati selliselt, et see oleks igal pool sarnase ülesehitusega. Õppematerjalide kasutajamugavuse kohta lisati, et piltidega töötades võiks need olla kättesaadavad kohe ülesandega samal lehel. Võeti soovitusi kuulda ja tehti materjalides vastavaid korrekture. Kokkuvõte õpetajate tagasisidest ning selgitused, kuidas parandati materjale on näha tabelis 1. Sarnased kommentaarid liigitati kokku üheks.

Tabel 1. Informaatika õpetajate tagasiside koostatud õppematerjalidele ja järeltegevused.

Õpetaja kommentaar	Tegevus
Programmeerimise koodinäiteid ühtlasemalt kommenteerida.	Lisati iga funktsiooni kasutamisele kommentaar.
Kohati keerulised selgitused peatükkides: „Mis on pilt?“ ja „Objekti tuvastus“. Toodi välja sõnastusi, mis tõenäoliselt tekitavad õpilastes segadust.	Autor selgitas lihtsamate sõnadega sisu lahti. Parandati välja toodud küsimused ja kirjutati paremini lahti.
Failid, mida kasutatakse võiksid olla kohe näite koodi juures kättesaadavad.	Lisati failide allalaadimise link näitekoodi juurde.
Näidiskoodis viga, segab programmi töötamist.	Parandati kopeerimisel sisse tulnud vead.
Osad muutujad koodis inglise keeles, osad eesti keeles.	Parandati võimalusel muutujanimed eesti keelde.
Funktsiooni süntaksite kirjelduses anti vormistamise ühtlustamise soovitusi.	Funktsioon süntaksi kirjeldust parandati arusaadavamaks.
Teksti struktuur ebahütlane. Osades peatükkides on eesmärk kohe alguses välja toodud, osades on teksti sees. Toodi tähelepanekuid, et igas peatükis oleks õppematerjalil sarnane ülesehitus.	Parandati struktuuri, et oleks igal pool sarnane ülesehitus.
Struktuuri koha pealt on positiivne, et peatükid on jaotatud ja igas peatükis ei ole ülemäära palju teksti.	Võeti tagasiside arvesse ja jälgiti vastavat soovitusi jooksvalt.
Toodi tähelepanekuid, et igas peatükis oleks õppematerjalil sarnane ülesehitus.	Ühtlustati õppematerjali ülesehitust.
Õpilastele igati jõukohane, ei ole üle jõu käiv. Saavad kohe midagi funktsionaalset. Hetkel kui eeldame, et õpilastel on varasem programmeerimise kogemus, siis peaks jõukohane olema.	Hoiti stiili, kus on põhirõhk funktsionaalsusel

### **3.4. Edasised võimalused**

Õppematerjali on plaanis testida 2024. aasta sügisel gümnaasiumi klassil kontaktõppes ning peale seda viiakse sisse uued muutused. Õppematerjalid saaks muuta interaktiivsemaks, lisades iga alampeatüki lõppu kontrollküsimustiku.

Eraldi saaks luua suurema ülesannete kogu, kuhu koondada arvutinägemisega seotud harjutused. Lisada materjalidesse rohkem ülesannete valikuvariante ja koostada õppevideod, et teha materjali iseseisev läbimine lihtsamaks.

Lisaks on valminud „Tarkvaraarendus“ lisamoodulina ka Raspberry Pi [14]. Tulevikus täiendada materjale nii, et saaks põimida arvutinägemise ja miniarvuti vahelise koostöö.

## Kokkuvõte

Arenevas digitaalmaailmas on aina suurenev vajadus informaatika valdkonna ja sealhulgas programmeerimise oskusega töötajate järgi, lisaks on üldised digioskused väga vajalikud peaaegu igal töökohal. Seetõttu on oluline võimaldada noortele õppida informaatikat juba enne ülikooli ja seeläbi tekitada või süvendada huvi antud valdkonna vastu. Eesti vaates aitab info- ja kommunikatsioonitehnoloogia oskuste arendamine ka säilitada meie ühiskonna arengupotentsiaali.

Eestis õpetatakse informaatikat gümnaasiumi astmes valikõppeainena ja see on loodud rakendades kaasaegseid õppemeetodeid – „Informaatika“ õppeaine arendab õpilastes nii raal- kui disainmõtlemist. Õppeaine koosneb viiest kursusest, mille hulgas on valikkursus „Tarkvaraarendus“. Antud bakalaureusetöö tulemusena loodi gümnaasiumi valikkursuse „Tarkvaraarendus“ uus arvutinägemise lisamoodul. Lisamoodul annab õpilastele lisateadmisi ja oskusi arvutinägemisest, mida nad saavad kasutada valikkursuse keskmes olevas projektitöös. Kokku loodi Python OpenCV arvutinägemise lisamoodulisse 9 alampeatükki, mille pikkuseks on 8 tundi. Materjalide koostamisel võeti eeskujuks Merrilli mudel, mis põhineb probleemikesksel õpetamisel ja sobib „Informaatika“ õppeaine uuendatud kontseptsiooniga.

Koostatud materjalides õpetatakse OpenCV mooduliga piltide ja videotega ümber käimist ning objektide tuvastamist kasutades eeltreenitud mudeleid. Õpilased peavad rakendama mooduli läbimisel varasemalt õpitut nagu näiteks *if* ja *while* lauseid. Õppematerjali peatükke on soovituslik läbida järjekorras, sest igas järgnevas peatükis kasutatakse eelmises peatükis õpitut.

Õppematerjalidele saadi tagasiside kvalitatiivse analüüsi tulemusel. Õppematerjalide koostamisel koguti tagasisidet kolmelt informaatika õpetajalt, kelle tagasiside tulemusel parendati õppematerjale. Kõige rohkem tagasisidestati mooduli ülesehitust, mille tulemusel muudeti peatükkide ülesehitus ühtlasemaks. Tagasiside hinnangul on valminud õppematerjal gümnaasiumis kasutatav. Loodud arvutinägemise lisamoodul laeti üles Tartu Ülikooli Courses keskkonda (lisa 1).

Antud töö võimalikud edasiarendused hõlmavad endast õppematerjalide muutmist interaktiivsemaks ja kontrollküsimuste koostamist peatükkide lõppu. Samuti koostada õppematerjalidele ülesannete kogu ja iseseisva õppimise parandamiseks luua õppevideod.

## Viidatud kirjandus

- [1] Jannik Lindner, „Computer Science Job Statistics“, Gitnux. Vaadatud: 15. mai 2024. <https://gitnux.org/computer-science-job-statistics/>
- [2] „New Report: 92% of Jobs Require Digital Skills, One-Third of Workers Have Low or No Digital Skills Due to Historic Underinvestment, Structural Inequities“, National Skills Coalition. Vaadatud: 15. mai 2024. <https://nationalskillscoalition.org/news/press-releases/new-report-92-of-jobs-require-digital-skills-one-third-of-workers-have-low-or-no-digital-skills-due-to-historic-underinvestment-structural-inequities/>
- [3] Gümnaasiumi informaatika tööühm, „Kontseptsioon ‘Gümnaasiumi informaatikaõpe’“. Haridus- ja Teadusministeerium, 2018. Vaadatud: 15. mai 2024. <https://oppekava.ee/wp-content/uploads/2021/08/Gumnaasiumi-informaatika-kontseptsioon.pdf>
- [4] K. Kasari, „Gümnaasiumi informaatika valikkursused“, ProgeTiiger. Vaadatud: 15. mai 2024. <https://projektid.edu.ee/pages/viewpage.action?pageId=81365502>
- [5] R. Krishna, „Computer Vision: Foundations and Applications“. Stanford University, 2017.
- [6] J. Lindner, „Must-Know Coding Statistics“, 23. detsember 2023. Vaadatud: 14. mai 2024. <https://gitnux.org/coding-statistics/#editorial-process>
- [7] K. Kuykendall, „Over Half of Students Surveyed See Coding Skills as Vital But Over a Third Lack Learning Access“, 3. november 2022. Vaadatud: 14. mai 2024. <https://thejournal.com/articles/2022/03/11/over-half-of-students-surveyed-see-coding-skills-as-vital-but-over-a-third-lack-learning-access.aspx>
- [8] The University of Georgia, Tbilisi, Georgia, S. Abesadze, ja D. Nozadze, „Make 21st Century Education: The Importance of Teaching Programming in Schools“, *jtle*, lk 158–163, 2020, doi: 10.18178/ijlt.6.3.158-163.
- [9] J. Bilbao *et al.*, „Introducing Computational Thinking and Algebraic Thinking in the European Educational Systems“, *International Journal of Education and Information Technologies*, kd 18, lk 11–19, märts 2024, doi: 10.46300/9109.2024.18.2.
- [10] R. Szeliski, *Computer vision: algorithms and applications*, Second edition. Texts in computer science. Cham: Springer, 2022.
- [11] „About“, OpenCV. Vaadatud: 15. mai 2024. <https://opencv.org/about/>
- [12] „OpenCV documentation“. Vaadatud: 15. mai 2024. <https://docs.opencv.org/4.x/>

- [13] „‘Gümnaasiumi riiklik õppekava’ Lisa 9 Valikõppeaine ‘Informaatika’“. Riigi Teataja I, 23. veebruar 2023. Vaadatud: 15. mai 2024. [https://www.riigiteataja.ee/aktilisa/1080/3202/3006/18m\\_gym\\_lisa9.pdf#](https://www.riigiteataja.ee/aktilisa/1080/3202/3006/18m_gym_lisa9.pdf#)
- [14] E. Tõnisson, T. Palts, K. Tõnisson, ja M. Säde, *Tarkvaraarendus*. Tartu Ülikool, 2023. <https://web.htk.tlu.ee/digitalu/tarkvara2/>
- [15] *Gümnaasiumi informaatika tarkvaraarenduse valikkursuse uued lisamoodulid*, (2. veebruar 2023). Vaadatud: 15. mai 2024. [Online Video]. [https://www.youtube.com/watch?v=ZCbUy39D0qc&list=PLwyEA\\_bBMcYHFV6xmTS0iIsZrWAjXxsAt&index=15](https://www.youtube.com/watch?v=ZCbUy39D0qc&list=PLwyEA_bBMcYHFV6xmTS0iIsZrWAjXxsAt&index=15)
- [16] M. D. Merrill, *First principles of instruction: assessing and designing effective, efficient, and engaging instruction*. San Francisco, CA: Pfeiffer, 2013.
- [17] „OpenCV University“. Vaadatud: 15. mai 2024. <https://courses.opencv.org>



## **Lisad**

### **I. Link valminud OpenCV õppematerjalidele**

<https://courses.cs.ut.ee/t/it/OpenCV>

## II. Litsents

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

Mina, Kevin Telliskivi,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „Arvutinägemise Python OpenCV õppemooduli loomine gümnaasiumile“, mille juhendaja on Tauno Palts, 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 4.0, 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.

Kevin Telliskivi

15.05.2024