

TARTU ÜLIKOOL  
Loodus- ja täppisteaduste valdkond  
Arvutiteaduste Instituut  
Andmeteaduse õppekava

Gregor Nepste  
Absoluutne drooni positsioneerimine kasutades aerofotosid  
Magistritöö (15 ECTS)

Juhendaja: Tambet Matiisen, MSc

Tartu 2025

# Sisukord

<b>MÕISTED</b> .....	<b>6</b>
<b>SISSEJUHATUS</b> .....	<b>7</b>
<b>1. TAUST</b> .....	<b>8</b>
<b>1.1 DROONIDE KASUTUS LAHINGUVÄLJAL</b> .....	<b>8</b>
<b>1.2 DROONI LENNULOOGIKA NING PÕHIMÕTTED</b> .....	<b>9</b>
<b>1.3 KONVOLUTSIOONILISED NÄRVIVÕRGUD</b> .....	<b>11</b>
<b>1.3.1 Sisend ning konvolutsiooniline kiht</b> .....	<b>12</b>
<b>1.3.2 Ahenduskiht</b> .....	<b>13</b>
<b>1.3.3 Täissidus kiht</b> .....	<b>14</b>
<b>1.3.4 ResNet50</b> .....	<b>15</b>
<b>1.3.5 DINOv2</b> .....	<b>16</b>
<b>1.3.6 WALDO30</b> .....	<b>17</b>
<b>1.4 MALLI SOBITAMINE</b> .....	<b>18</b>
<b>1.5 KIRJANDUSE ÜLEVAADE</b> .....	<b>20</b>
<b>2 METOODIKA</b> .....	<b>25</b>
<b>2.1 ANDMETE KÄTTESAAMINE JA VALIM</b> .....	<b>25</b>
<b>2.2 REAALSE LAHENDUSE ÜLESEHITUS</b> .....	<b>26</b>
<b>2.3 PROTOTÜÜBI ÜLESEHITUSE LOOGIKA</b> .....	<b>27</b>
<b>2.4 DROONI ERI KÕRGUSTE MÕJU SAADUD TULEMUSTELE</b> .....	<b>31</b>
<b>2.5 DROONI ERI PÕÖRDENURKADE MÕJU SAADUD TULEMUSTELE</b> .....	<b>32</b>
<b>3 TULEMUSED</b> .....	<b>33</b>
<b>3.1 RESNET50</b> .....	<b>35</b>
<b>3.2 WALDO30 YOLOv8</b> .....	<b>36</b>
<b>3.3 DINOv2 MUDELID (BASE, SMALL, LARGE)</b> .....	<b>36</b>
<b>3.3.1 DINOv2-base</b> .....	<b>36</b>
<b>3.3.2 DINOv2-small</b> .....	<b>37</b>
<b>3.3.3 DINOv2-large</b> .....	<b>37</b>
<b>3.4 ERI KÕRGUSTE MÕJU ANALÜÜS</b> .....	<b>38</b>
<b>3.5 ERI PÕÖRDENURKADE MÕJU ANALÜÜS</b> .....	<b>40</b>
<b>4. ARUTELU</b> .....	<b>42</b>
<b>4.1 LAHENDUSE PIIRANGUD</b> .....	<b>42</b>
<b>4.2 IDEED EDASIARENDUSTEKS</b> .....	<b>43</b>

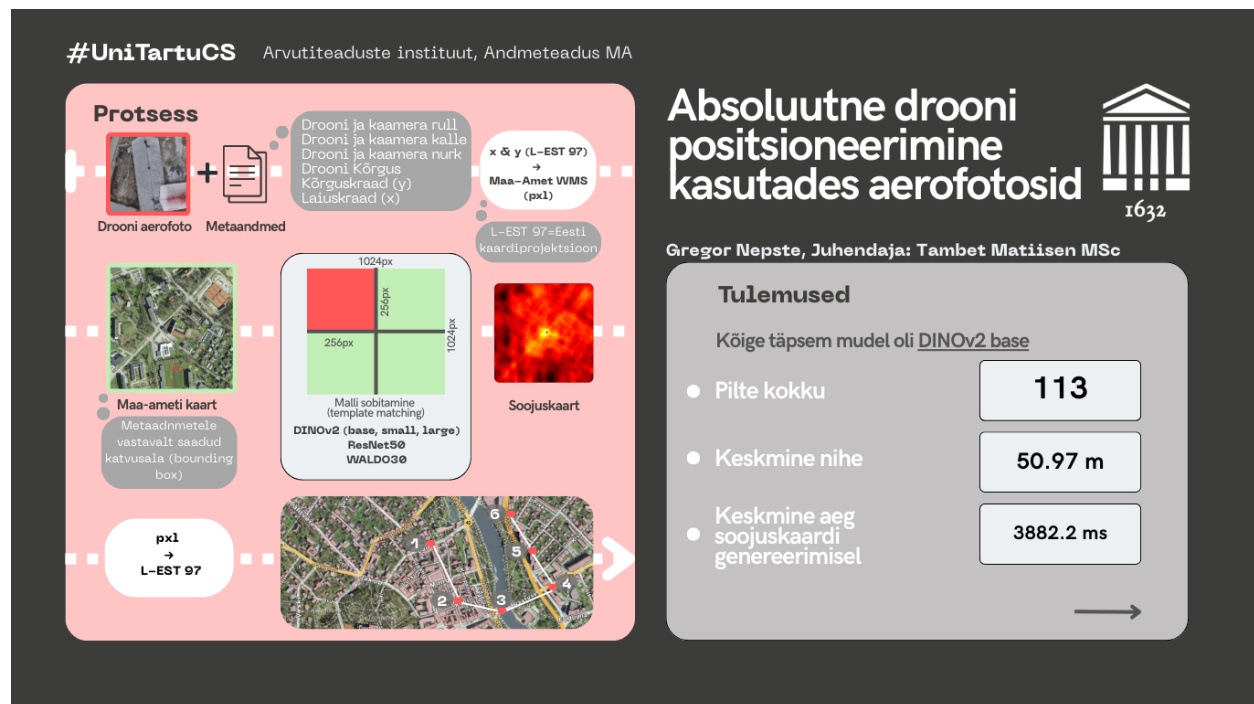
<b>KOKKUVÕTE</b> .....	<b>45</b>
<b>TUNNUSTUS</b> .....	<b>46</b>
<b>VIIDATUD KIRJANDUS</b> .....	<b>47</b>
<b>LISAD</b> .....	<b>51</b>
<b>LITSENTS</b> .....	<b>54</b>

## Absoluutne drooni positsioneerimine kasutades aerofotosid

### Lühikokkuvõte:

Droonide täpne ja usaldusväärne positsioneerimine on hädavajalik paljudes erinevates rakendustes, sealhulgas põllumajanduses, päästeoperatsioonides, militaarsektoris ja mujal. Kuna enamuse droone navigeerivad peamiselt GPSi abil on neil olulised puudused keerulistes keskkondades ning samuti häkkimis- või segamisohud militaarvaldkonnas. Et otsida alternatiivseid lahendusi probleemi lahendamiseks, uuritakse käesolevas töös droonide absoluutse positsioneerimise meetodeid, kasutades aerofotosid ja sügavõppe tehnikaid. Spetsiifiliselt keskendutakse konvolutsiooniliste närvivõrkude (CNN) arhitektuuride rakendamisele mallide sobitamiseks ja positsioneerimise täpsuse parandamiseks, mille eesmärgiks on saavutada drooni täpne asukoht ilma GPSi kasutamata. Töö sisaldab nii metoodilist kirjeldust kui ka eksperimentaalset valideerimist, tuginedes ettevõtte 3DI poolt tehtud Tartu linna ortofotodele. Saadud tulemused näitavad, et sügavõppe meetodid annavad täpseid tulemusi drooni absoluutseks positsioneerimiseks ning omavad potentsiaali luua veel täpsemaid tulemusi ka edasiarendustes. Antud magistritöö kood ning andmed on saadaval GitHub lingil: [https://github.com/gnepste/GregorN\\_AT-Magistritoo25](https://github.com/gnepste/GregorN_AT-Magistritoo25)

### Visuaalne abstrakt:



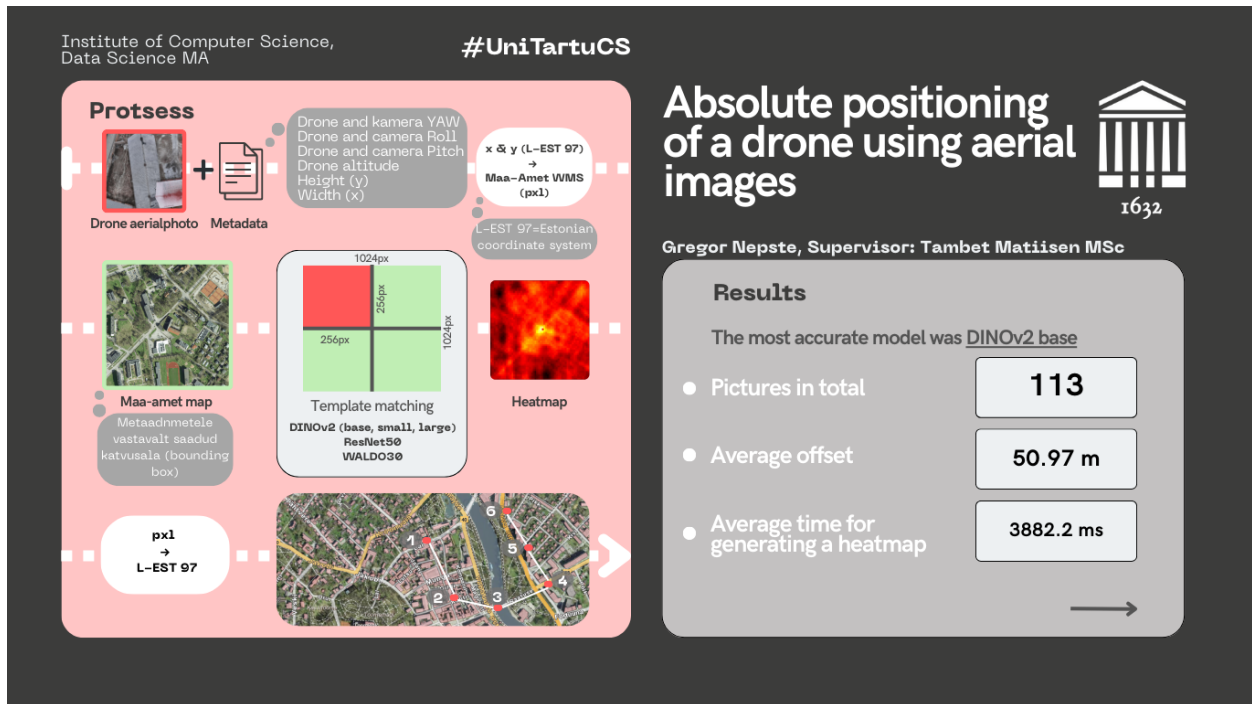
**Märksõnad:** Malli sobitamine, CNN, DINOV2, ResNet50, WALDO30, Koosinus sarnasus  
**CERCS:** T111 - Pilditehnika; P176 - Tehisintellekt

# Absolute positioning of a drone using aerial images

## Abstract:

Accurate and reliable drone positioning is critical in many domains, including agriculture, search-and-rescue, the military, and beyond. Most drones rely primarily on GPS, which can fail in complex environments and is vulnerable to spoofing or jamming in military scenarios. To explore alternative solutions, this project investigates absolute drone positioning methods that use aerial imagery and deep-learning techniques. Thesis focuses on applying convolutional neural-network (CNN) architectures for template matching in order to estimate the drone's position without GPS. The work combines a detailed methodology with experimental validation on city-scale orthophotos of Tartu provided by the company 3DI. Results demonstrate that deep-learning approaches achieve ~50-metre-level accuracy in absolute positioning and show potential for even higher precision in future developments. Source code and sample data are publicly available on GitHub: [https://github.com/gnepste/GregorN\\_AT-Magistritoo25](https://github.com/gnepste/GregorN_AT-Magistritoo25)

## Visual abstract:



**Keywords:** Template matching, CNN, DINOv2, ResNet50, WALDO30, Cosine similarity

**CERCS:** T111 - Imaging, image processing; P176 - Artificial intelligence

## Mõisted

- CAF** - *Cross Attention Fusion* (rist tähelepanu mestimine)
- CNN** - *Convolutional Neural Network* (konvolutsiooniline närvivõrk)
- CS** - *Cosine Similarity* (koosinus sarnasus)
- DRL** - *Drone Referring Localization* (droonipõhine lokaliseerimine)
- DSM** - *Digital Surface Models* (digitaalsed pinnamudelid)
- EW** - *Electronic warfare* (elektrooniline sõjapidamine)
- FCN** - *Fully Convolutional Network* (täiskonvolutsiooniline närvivõrk)
- FCNN** - *Fully-Connected Neural Network* (täissidus tehisnärvivõrk)
- FFB** - *Feature Fusion Block* (tunnuste mestimise plokk)
- FOV** - *Field Of View* (vaateväli)
- FPN** - *Feature Pyramid Network* (tunnuste püramiid vektor)
- FPV** - *First Person View* (esmaisikuline vaade)
- GC** - *Group Convolution* (grupi konvolutsioon)
- GNSS** - *Global Navigation Satellite System* (ülemaailmne navigatsiooni satelliidi süsteem)
- GPS** - *Global Positioning System* (globaalne positsioneerimissüsteem)
- GPU** - *Graphics Processing Unit* (graafika töötlemisüksus)
- IMU** - *Inertial Measurement Unit* (inertsiaalandur)
- PnP** - *Perspective-n-Point* (punkti perspektiiv)
- RANSAC** - *Random Sample Consensus* (juhusliku valimi konsensus)
- ReLU** - *Rectified Linear Unit* (parandatud lineaarne ühik)
- SAF** - *Self Attention Fusion* (enesetähelepanu mestimine)
- UAV** - *Unmanned Aerial Vehicle* (mehitamata õhusõiduk)
- ViT** - *Visual Transformer* (visuaalne transformer)
- VTX** - *Video Transmitter* (videopildi edastaja)
- WMS** - *Web Map Service* (veebikaardi teenus)
- 6-Dof** - *Six degrees of freedom* (6 vabadusastet)

# Sissejuhatus

Droonitehnoloogia kiire areng viimase kümnendi jooksul on toonud kaasa ulatuslikke muutusi erinevates sektorites: põllumajanduses, päästeoperatsioonides, militaarvaldkonnas ja mujal. Droonide autonoomne ja täpne positsioneerimine mängib kriitilist rolli nende süsteemide efektiivsuses ja usaldusväärsuses. Senini on peamiseks positsioneerimise meetodiks olnud globaalne positsioneerimissüsteem (GPS), mille peamised puudused militaarvaldkonnas on vastuvõtuvõimaluste katkemine keerulistes keskkondades ning segamis või häkkimisohud [1]. Probleemi aktuaalsust ilmestavad mitmed tegurid ning näited. Esiteks on GPS-vabade lahenduste väljatöötamine kriitilise tähtsusega militaarvaldkonnas, kus Ukraina sõja näitel saab vaenlase GPSi abil juhitavaid droone sihilikult häirida, lendu katkestada või kahjustada. Samuti on alternatiivsed meetodid olulised tsiviilkasutuses, kus Tartu lennujaama GPS seadmete segamine Venemaa poolt illustreerib otsest vajadust täiendavate tehnoloogiate järele, mis tagaksid usaldusväärse navigatsiooni olenemata olukorrast [2].

Üheks alternatiiviks militaarvaldkonnale keskendudes pakutakse antud magistritöös välja pildipõhist positsioneerimismeetodit, kus drooni kaamera kaadreid ning aerofotosid kasutatakse täpse asukoha määramiseks. Tänu suurenenud arvutusvõimsustele ja sügavõppe meetodite kiirele arengule on konvolutsioonilised närvivõrgud (CNN) muutunud efektiivseks vahendiks pilditötluse ja mustrituvastuse ülesannetes. Seetõttu keskendutakse töös konkreetselt CNN-põhise mallisobitamismeetodi (*template matching*) rakendamisele, et loodetavast parandada positsioneerimise täpsust sõltumata pildi valgustingimustest ja erinevates deformatsioonidest.

Magistritöö jaguneb sissejuhatust, kokkuvõtet ja tunnustust arvestamata neljaks sektsiooniks. Sektsioon 1 annab ülevaate töö taustast, milles selgitatakse peamiselt droonide kasutusala lahinguväljal ning lennuloogikaid ja antakse töös kasutatud mudelitest ja meetoditest. Sektsioon 2 kirjeldab töö metoodikat, milles kirjeldatakse valimit ning väljapakutud lahenduskäiku. Sektsioon 4 toob välja tulemused, milles võrreldakse töös kasutatud mudelite asuohatuvastus täpsus ning eksperimentaalsete testide tulemused. Sektsioon 4 on arutelu, kus analüüsitakse lahenduse piiranguid ning pakutakse välja ideid edasiarendusteks.

Käesoleva magistritöö sõnastuse sujuvamaks muutmisel on autor kasutanud OpenAI poolt arendatud suurt keelemudelit ChatGPT-04.<sup>1</sup>

---

<sup>1</sup> <https://chat.openai.com>

# 1. Taust

Selles peatükis antakse ülevaade mehitamata õhusõidukite (UAV) kasvavast rollist tänapäeva lahinguväljal ja selgitatakse, miks on GPS-vabad positsioneerimislahendused Euroopas ja sõjategevuses kriitilise tähtsusega. Seejärel tutvustatakse FPV-drooni riistvara põhikomponente ning lennuloogika termineid (*yaw, pitch, roll*), mis moodustavad aluse kogu edasisele metoodikale ja mudelite rakendamisele. Viimasena kirjeldatakse konvolutsiooniliste närvivõrkude arhitektuuri ning tutvustatakse töös kasutatavaid eeltreenitud sügavõppe mudeleid.

## 1.1 Droonide kasutus lahinguväljal

Droonid ehk mehitamata õhusõidukid (UAV-d) on muutnud sõjatööstust jäädavalt, olles üheks olulisemaks relvaks Ukraina sõjas. Täna on Ukraina sõjas kasutusel üle 100 erineva suuruse ja otstarbega drooni ning Ukraina suudab aastas lisaks sisseostetud droonidele toota ise ligi 4 miljonit drooni, olles sellega võrdne Venemaa tootmisvõimekusega [3]. Kõige tõhusamaks relvaks sõjas on kujunenud isepilootuva vaate (FPV) võimekusega droonid, mille maksumus koos lõhkepeaga on ligikaudu 500 eurot ning vaenlase tabamise täpsus ületab suurtüki oma [3]. Antud droonid on kujunenud niivõrd edukaks, et pea igasse Ukraina sõjaväe brigaadi on loodud eraldi droonikompanii, millesse kuuluvad erinevaks otstarbeks eri droonid: FPV droonid, vaatlusdroonid toetamaks FPV drooni sihtmärgi tuvastust ning pommikandja droonid [4]. Kuigi Ukraina on tõestanud sõja jooksul enda suurt mõjuvõimu droonide arendamises ning rajanud teed tulevastele standarditele selles vallas, siis on droonide kasutamisel sõjapidamises kriitilisi lahendusi vajavaid teemasid.

Esiteks toetub Ukraina suuresti Hiinale droonikomponentide tellimisel. Võttes arvesse geopoliitilist olukorda, kus Hiina ja Venemaa on sõlminud omavahel vankumatu bilateraalse suhte, tähendab koostöö Hiinaga suurt riski. Juba praegu on Hiina hakanud piirama drooni komponentide ning materjalide eksporti, mis juba avaldab mõju Ukraina droonide tootmisele [4].

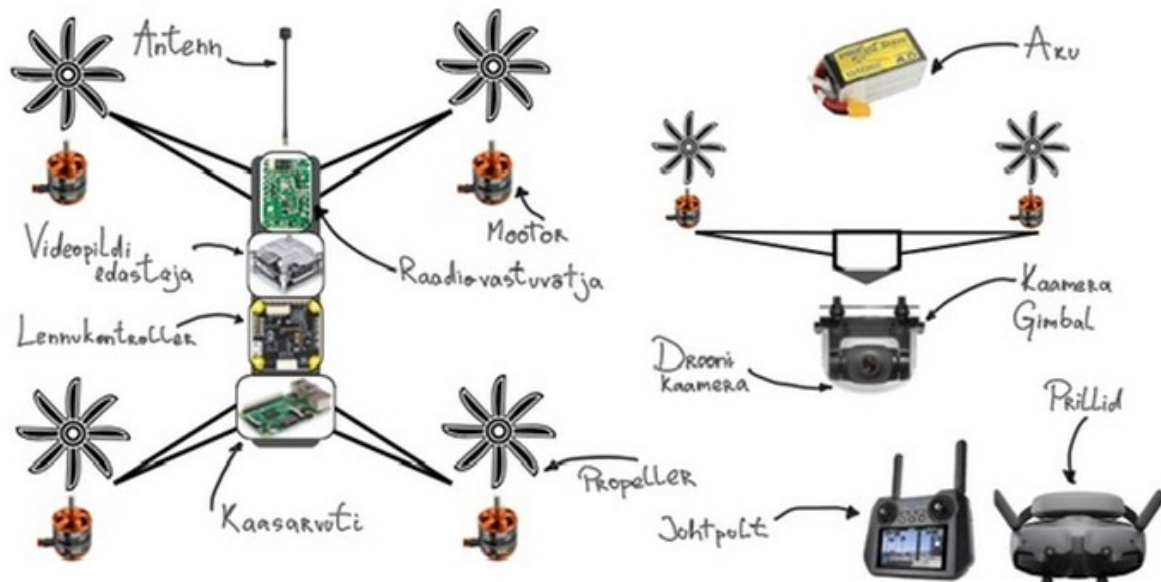
Ukraina ning Venemaa arendavad droonitehnoloogiaid tsükliliselt paremini segamiskindlaks, kuid iga uue droonitehnoloogia arendusega areneb ka elektroonilise segamise seadmete (EW) suutlikkus droonide juhtimiseks kasutatavat raadiosidet segada. Elektroonilised seadmed nagu segajad on kujunenud kõige efektiivsemaks mooduseks vaenlase droonide alla toomises. Kuna segajad suudavad kasutada limiteeritud hulka sagedusi ning drooni operaatorid üritavad toimetada vähe kasutatud sagedustel, toimub rindel kassi hiire mäng, kus elektrooniliste seadmete kasutajad üritavad leida õhus olevate droonide sagedusi ja neid alla tuua [4]. Lisaks sellele peavad mõlemad pooled võitlema ka elektrooniliste luuresüsteemidega, mis suudavad üles leida vaenlase droonipiloodi asukoha [4]. Kuigi rindel kasutusel olevad tehisintellektil põhinevad

tuvastussüsteemid aitavad ilma operaatorita rünnakut sooritada, on hetkel kõige töökindlamaks alternatiiviks füüsilise valguskaabli kasutamine rünnaku sooritamiseks. Seetõttu on Euroopas hetkel vajadus igasuguste GPS vabade alternatiivide arendamisele. Kuna Ukraina on näidanud, et rasketest oludest hoolimata on võimalik olla droonide innovatsioonis teerajaja, siis on see olnud ajendiks antud töö autorile panuse andmiseks. Ajakriitilisus alternatiivsete meetodite järele on hetkel olulisem kui kunagi varem.

## 1.2 Drooni lennuloogika ning põhimõtted

Drooni riistvaraline ülesehitus ning selle põhimõtted on määratud peamiselt missiooni ülesandepüstitusest ning vajadustest. Kuna erinevatest droonitüüpidest Ukraina sõjas on enim tulemuslikuks osutunud FPV droonid, siis keskendutakse antud peatükis vaid antud droonide ülesehitusele ning lennuloogikale. Siinkohal on ka oluline mainida, et kuna lahingus kasutatud FPV droonid võivad tavalistest kommertsdroonidest erineda lisakomponentide poolest, siis keskendutakse hetkel kõige standardsema FPV drooni riistvaralisele ülesehitusele. Sellegipoolest mainitakse antud peatükis ka lisakomponente, mida lahingus kasutatavate droonide puhul arendatakse ja kasutatakse eesmärgi paremaks täitmiseks.

FPV droonid koosnevad mitmetest erinevatest komponentidest (vaata Joonis 1) jagunedes kolme eri kategooriasse: elektrisüsteem, lennusteem, FPV süsteem [5]. Elektrisüsteem koosneb akust ning elektrijaotuskilbist [5]. Lennusteem koosneb propellerite jaoks vajalikest mootoritest, propelleritest, videopildi edastajast (VTX), kaas arvutist, lennukontrollerist ja raadiovastuvõtjast [5]. FPV osaks on prillid, raadiokontroller ehk juhtpult, kaamera, kaamera gimbal ehk stabiilsusraam ning signaali vastuvõtmiseks antenn. Lisaks asuvad droonis magnetomeeter pöörde tuvastamiseks, inertsiaalandur (IMU) mõõtmaks suhtelist liikumist nullpunktist ning kõrgusmõõdik, mis võib olla õhurõhu põhine (baromeeter) või laseril põhinev (*lidar*). Antud komponente joonisel ei märgitud.

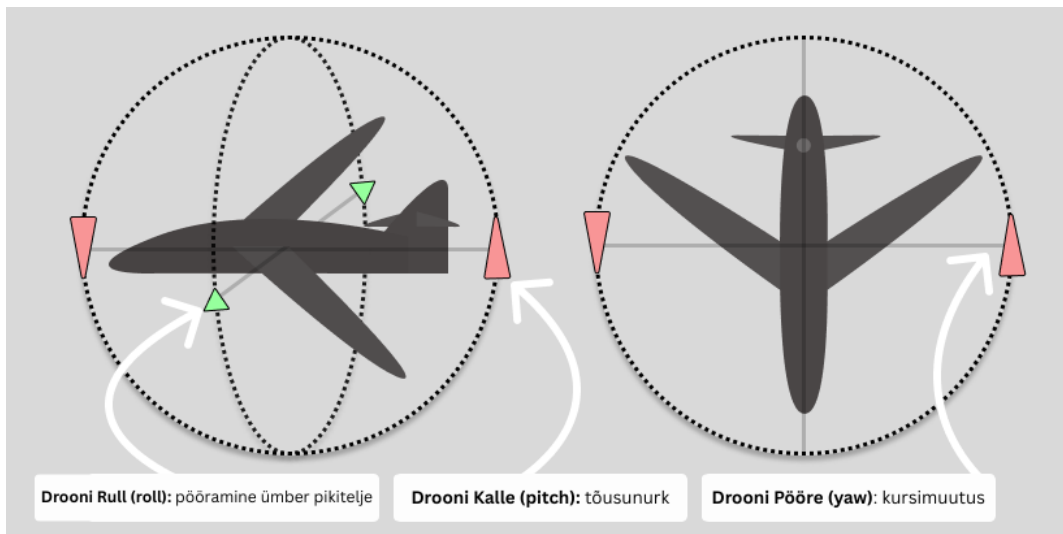


Joonis 1. Standardse FPV drooni riistvaraline ülesehitus [5].

FPV droon suudab liikuda kolmedimensioonilises ruumis, sarnanedes sellega lennuki lennuloogikale. Kuigi droon suudab teha manöövreid, mida lennuk ei suuda, siis kasutatakse drooni lennuloogika kirjeldamiseks sellegipoolest samu termineid ning lähenemist. See tähendab, et drooni ning lennuki erinevuseks on drooni suutlikus liikuda ka tagurpidi ning mõlemale poole külgsuunas, olles samal ajal kaameraga suunatud samasse punkti. Drooni lennuloogika kirjeldamiseks kasutatakse kolme terminit (vaata Joonis 2): pööre (*yaw*), kalle (*pitch*) ja rull (*roll*) [6].

- **Pööre** kirjeldab drooni või drooni kaamera pööramisnurka ehk aitab mõista, kuhu drooni nina on suunatud. Näiteks pööre =  $0^\circ$  puhul on drooni suund põhjakaarde ning pööre =  $90^\circ$  puhul on droon suunatud idasse. Ülejäänud ilmakaare nurgad on võimalik arvutada liites juurde kellaosuti suunas eelnevale võrratustele järgemööda  $90^\circ$ . Siinkohal on oluline eristada, et drooni pööre ja drooni kaamera pööre ei pruugi olla omavahel olla kooskõlas drooni kaamera gimballi tõttu. Ehk kui droon pöörab, võib kaamera endiselt jääda samasse suunda, eristudes sellega drooni pöördenurgast.
- **Rull** kirjeldab drooni või drooni kaamera rulli ehk defineerib, kui suure nurga all on droon või kaamera oma pikitelje suhtes kaldu. Drooni kaamera gimballi olemasolul on antud rulli väärtus alati  $0^\circ$  kuna kaameranurk on stabiliseeritud. Ka siin tuleb arvestada, et drooni kaamera rull ei pruugi olla sama, mis drooni rull kuna kaamera gimballi olemasolu võimaldab kaameralt jääda samasse asendisse isegi kui droon pöörab ümber oma pikitelje.

- **Kalle** kirjeldab drooni või drooni kaamera tõusunurka ehk defineerib, kui suure nurga all on drooni nina maapinnast. Kui kalle =  $0^\circ$ , siis see tähendab, et drooni kaamera vaatab täisnurkselt alla ning kui kalle =  $90^\circ$ , siis annab see mõista, et kaamera vaatab otse. Ka siinkohal võivad drooni kaamera kalle ja drooni kalle üksteisest drooni gimballi tõttu erineda.



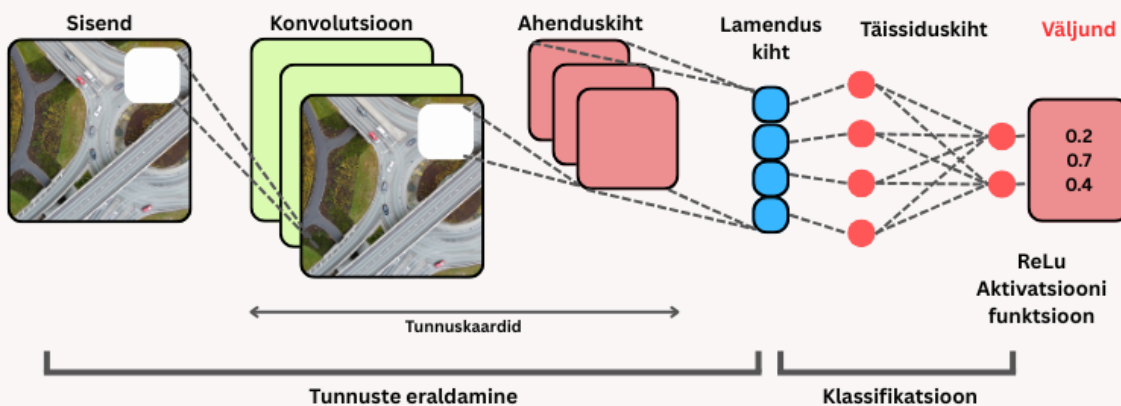
Joonis 2. Drooni lennuloogika terminoloogiline selgitus.

Lisaks eelmainitud terminitele on oluline ka välja tuua drooni operatiivne kõrgus merepinna suhtes. Kuna drooniliike on palju, siis tuuakse antud magistritöö raames välja ainult FPV drooni lennukõrguse vahemik milleks on lennu ajal 100-150m merepinnast. Olles sihtmärgi tuvastanud ja sooritades rünnet, laskub droon madalamale sihtmärgi tabamiseks.

### 1.3 Konvolutsioonilised närvivõrgud

Konvolutsioonilised tehisnärvivõrgud (CNN) on sügavõppe mudelid, mida kasutatakse näiteks isejuhtivate autode arendamisel, meditsiiniliste piltide analüüsis ning kõnetuvastuses [7]. Samuti kasutatakse CNN arhitektuurile toetuvaid mudeleid masinõppemises, tuvastamiseks erinevaid objekte, inimesi, hooneid, puid ning muid klasse. CNN-de arhitektuuri võib ülevaetlikult jagada viieks erinevaks komponendiks (vaata Joonis 3): tunnuste eraldamine konvolutsioonilise kihi kaudu, ahenduskihid, aktivatsioonikihid, lamenduskihid ja väljundkiht [7].

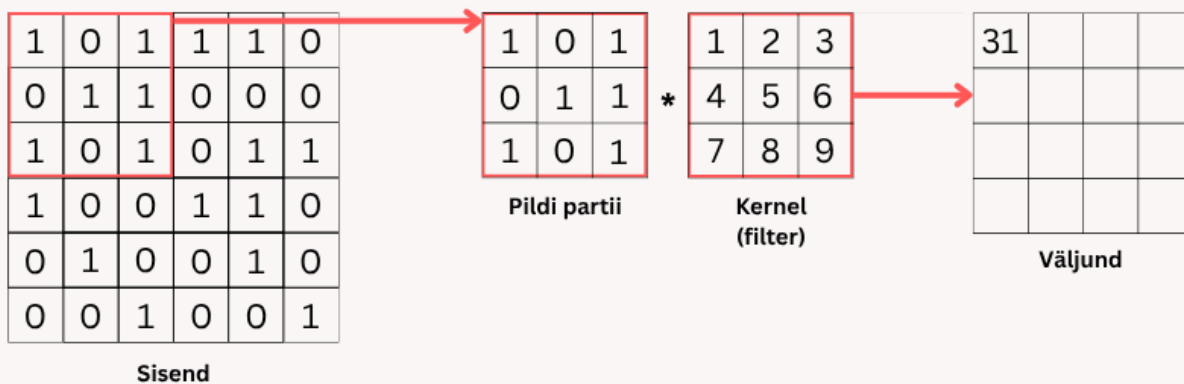
## Konvolutsioonilise närvivõrgu arhitektuur



Joonis 3. Konvolutsioonilise tehiskärvivõrgu ülevaatlik arhitektuur [8].

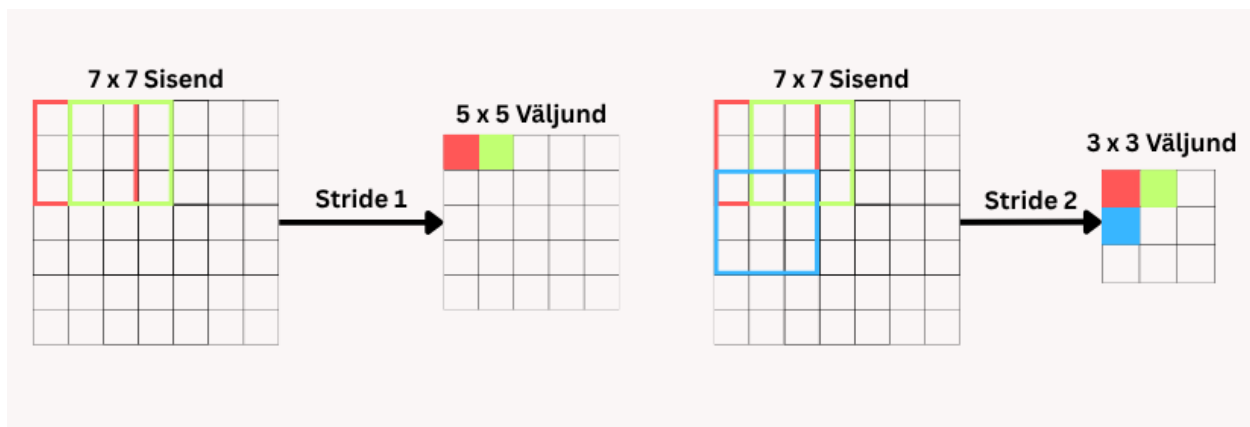
### 1.3.1 Sisend ning konvolutsiooniline kiht

Tehiskärvivõrgu esimeses kihis toimub andmete lugemine kasutades filtrit (*kernel*) (konvolutsiooni suurus), et tuvastada pildil oleva objekti kuju ning selle servasid [7]. Joonis 4 on kujutatud konvolutsioonilise kihi toimimisloogika.



Joonis 4. CNN-i konvolutsioonilise kihi toimimisloogika [7].

Konvolutsiooniline samm (*stride*) on parameeter CNN filtri sees, mis määrab, millise sammuga filter üle pildi liikuma peab (vaata Joonis 5) [9].



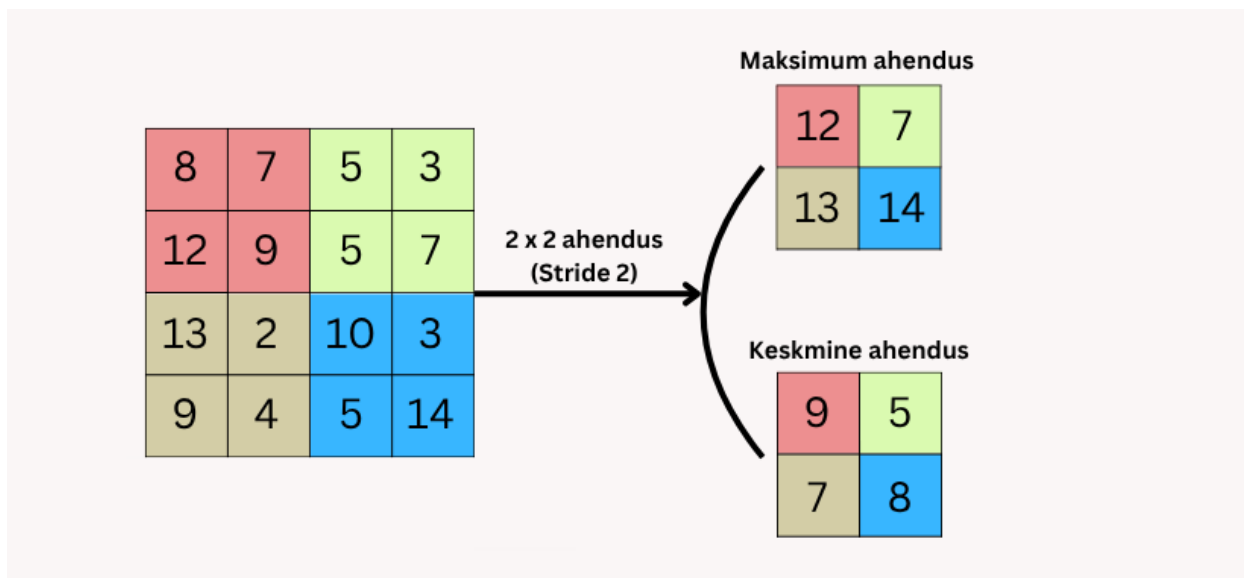
Joonis 5. Konvolutsioonilise sammu (*stride*) toimimise loogika [9].

Konvolutsioonikihid kahandavad ruumilisi mõõtmeid, mistõttu soovitakse enne ahendamisetappe tunnускаardi algset suurust säilitada. Selleks kasutatakse filtrite rakendamisel ääristamist (*padding*), kus pildi servadesse lisatakse täiendavaid piksliridu, et konvolutsioonijärgne mõõtmete arv vastaks algsele [9].

### 1.3.2 Ahenduskiht

Ahenduskihi ülesandeks on säilitada eraldatud väärtuste omadused, vähendades sellega samal ajal mudeli arvutuslikku keerukust [9]. Selle juures aitab ahenduskiht (vaata Joonis 6) lihtsustada tunnuseid üldistades nende olemasolu pisemas pildi piirkonnas. Ahenduskiht toimib vahelülina konvolutsioonilise kihi ja täissidusa kihi vahel. Antud kihi puhul on olulisteks komponentideks mitu ahenduse meetodit: maksimum väärtuse leidmine (*max pooling*), keskmise väärtuse leidmine (*average pooling*) ja summeeritud väärtuste leidmine (*sum pooling*) [7].

Maksimum väärtuse leidmise puhul võetakse tunnускаardi kõige suurem element, kus keskmise väärtuse leidmise abil arvutatakse mingi teatud pildi piirkonna keskmiste tunnuste väärtuse [7]. Summeeritud väärtuse leidmine aitab arvutada seejärel tunnuste summa eeldefineeritud piirkonnas [7].



Joonis 6. CNN ahenduskihti illustreeriv toimimisloogika [7].

Viimase etapina kasutatakse lamendamist, mille eesmärk on konverteerida saadud 3-dimensioonilised massiivid üheks lineaarseks vektoriks [7]. See on täissiduskihi ettevalmistav samm, tuvastamaks pilti.

### 1.3.3 Täissidus kiht

Täissidus kiht ühendab kõiki sisend neuroneid iga väljund neuroniga. Antud kihid on kasutusel CNN-ide viimases staadiumis, mille eesmärgiks on eelnevate kihtide baasil formuleerida terviklik ennustus või tulemus. Täissidusa kihi struktuuri defineerivad kaalude maatriksid ja nihete vektorid [10]. Ehk kui kihil on  $n$  sisendit ja  $m$  väljundit, siis kaalude maatriksi mõõtmed on  $n \times m$  ning nihete vektoril on  $m$  elementi [10]. Seejärel arvutab väljundneuron kõikide sisendite kaalutud summa, lisab sellele oma nihke ning rakendab aktivatsioonifunktsiooni nagu näiteks ReLu või sigmoid funktsiooni, mis viiakse ellu eraldi aktivatsioonikihis [10]. Aktivatsioonikihte kasutatakse sarnaselt ka konvolutsioonikihtide järel, et lisada mudelisse mitte-lineaarsust ja parandada õppimisvõimet [10]. Kuigi selline ülesehitus võimaldab modelleerida keerukaid seoseid erinevate tunnuste vahel, siis toob see kaasa suure hulga parameetrite tekke, mis võib muuta kogu närvivõrgu treenimise väga ressursimahukaks [10].

### Täiskonvolutsiooniline närvivõrk

Kuigi CNNi ülesehituse mõistmine on antud töö kontekstis oluline, siis tuleb ka esile tuua täiskonvolutsiooniline närvivõrk (FCN). Töös kasutatavad eeltreenitud mudelid on just antud tüüpi

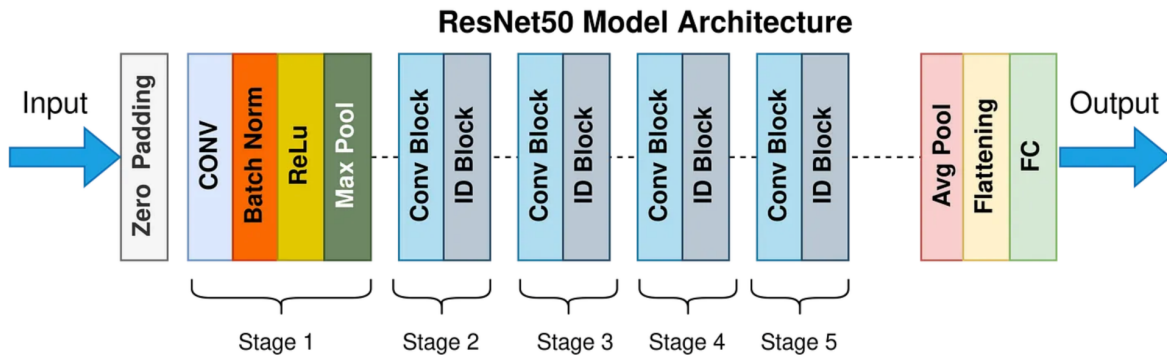
konvolutsioonilised närvivõrgud. FCN erineb konvolutsioonilisest närvivõrgust selle poolest, et puuduvad täissidusad kihid [11]. See tähendab, et kasutatakse vaid konvolutsioonilisi kihte.

Kui tavaline CNN-i saab rakendada ainult fikseeritud mõõtmetega pildile, siis FCN-i saab rakendada suvaliste mõõtmetega pildile. Sellegipoolest muutuvad ka sellisel juhul väljundi mõõtmed. Kuna enamus arvutusmahust tehakse konvolutsioonina kogu pildi ulatuses, teeb see FCN-i rakendamise mugavaks semantilise segmentatsiooni või kaartide mallisobituse jaoks.

### 1.3.4 ResNet50

ResNet50 on Microsofti poolt 2015 aastal avalikustatud juhendatud sügavõppe mudel, mis on treenitud ImageNeti peal ning mille eesmärk on visuaalne objekti tuvastus [12]. Antud mudelil on 50 erinevat kihti (vaata Joonis 7) ning selle arhitektuur koosneb neljast erinevast komponendist: **konvolutsioonikihid, konvolutsiooniplokid, jääkplokid** ning **täissidus kihid** [12].

- **Konvolutsioonikihid** on olulised tunnuste eraldamisel ning rakendavad filtreid sisendpiltidele, võimaldades mudelil tuvastada erinevaid mustreid, objekti servi ning tekstuure andmetes [13].
- **Konvolutsiooniplokid** hõlbustavad kõrgetasemeliste tunnuste eraldamist sisendandmetest koosnedes mitmetest konvolutsioonikihtidest, millele tavaliselt järgneb sageli väärtuste normaliseerimine ning aktivatsioonifunktsioonid [13].
- **Jääkplokkide** ülesanne on olla vahekihiks, mis võimaldab sisendil liikuda läbi erinevate konvolutsioonikihtide ja lisada sisendi tagasi väljundisse ning samuti vahele jätta erinevaid kihte, leevendamaks kaduva gradiendi probleemi treeningu ajal [13]. Lisaks sellele võimaldab antud kiht õppida jääkfunktsioone, mis kaardistavad sisendi soovitud väljundiks [13].
- **Täissiduskiht** on vastutav ennustuste ning klassifikatsiooni tegemisel eraldatud tunnuste pealt ehk ResNeti puhul kaardistab antud kiht õpitud tunnused lõpliku väljundi klassidele [13].



Joonis 7. Resnet50 mudeli arhitektuuri ülevaade [14].

Just jääkplokk on oluline lisand ResNet50 arhitektuuris, kuna võimaldab vajadusel üle hüpata teatud kihtidest, et gradient saaks liikuda ilma selle kahanemiseta [13]. Olulisteks osisteks jääkploki toimimisel on: **ReLU aktivisatsioon**, **vahela jätmise ühendus** (*skip connection*) ja **pudelikaela konvolutsioonikihid** (*bottleneck convolution layers*) [13].

**ReLU aktivisatsioon** on lineaarne funktsioon, mis väljastab sisendi otse juhul, kui see on positiivne, muutes sellega mudeli treenimise lihtsamaks ning hoides kokku arvutuslikku jõudlust [15].

**Pudelikaela konvolutsioonikihtide** eesmärgiks on õppida ning tabada tunnuste omadusi sisendandmetes madalama arvutusressursiga [16]. Just seetõttu kasutatakse pudelikaela konvolutsioonikihtide enamustes sügavõppe mudelites, kuna see aitab säästa aega. ResNet50 mudelis koosneb antud kihtidekogum kolmest konvolutsioonilisest kihist koos ploki normaliseerimise (*batch normalization*) ja ReLU aktivatsiooniga [17].

**Vahela jätmise ühenduse** ülesandeks on säilitada mudeli jõudlus ka siis kui kihtide aktiveerimise arv sisendi liikumisel suureneb [18].

ResNeti jääkploki olemasolu muudab mudeli efektiivseks, mille tõttu kasutati seda ka magistritöös ühe mudelina. Kuna tunnuseid on droonifotodel ja kaardil palju, aitab ResNet50 hõlpsamini tunnustevahelisi omadusi õppida ilma jõudlust kaotamata.

### 1.3.5 DINOv2

DINOv2 on iseseisvalt juhendatud masinõppimise mudel, mis on loodud ning avalikustatud ettevõtte Meta allharu Meta Research poolt 2023. aastal [19]. Iseseisvalt juhendatud tähendab, et antud mudel ei vaja märgendatud andmeid ning suudab õppida otse sisendi pealt. See muudab DINOv2 kasutusala laiaks kuna suudab tuvastada rohkem erinevaid klasse võimaldades sellega

kasutada suuremaid andmekogusid [20]. Lisaks sellele aitab DINOv2 lähenemine hoida palju aega kokku andmete märgistamise ning piltide eeltreenimise pealt muutes selle kasutamise kuluefektiivsemaks.

DINOv2 arhitektuur põhineb visuaalsetel transformetil (ViT), mille hierarhiline disain võimaldab tõhusalt haarata nii kohalikke kui ka globaalseid kujundijooni [21]. Nagu eelnevalt mainitud, on DINOv2 treenitud mitmekesise andmestikuga, mis hõlmab erinevatest domeenidest pärit pilte, tagades mudeli hea üldistusvõime. Treeningprotsessis kasutatakse mitmeid andmetäiustusi näiteks: kärpimist, peegeldamist ja värvide muutmist [21]. Seda selleks, et õpetada mudelile teisenduste suhtes erinevaid omadusi. Samuti on DINOv2-s täiustatud kadufunktsiooni (*loss function*), mille eesmärgiks oli ühtlustada kahe mudeli tulemusi ning vältida olukorda, kus mudel hakkab igas olukorras andma peaaegu täpselt ühesuguseid vastuseid. [21]. Lisaks kasutab mudel optimeerimis- ja reguleerimisstrateegiaid, mis aitavad stabiliseerida õppimist ja kiirendada koondumist [21].

DINOv2 puhul on võimalik rakendada ka eri suurusega eeltreenitud mudeleid vastavalt ülesande püstitusele ning keerukusele. Meta poolt arendatud DINOv2 eeltreenitud mudeleid on mitmeid ning need erinevad üksteisest peamiselt oma parameetrite arvu poolest.

### 1.3.6 WALDO30

WALDO30 on konvolutsionaalne närvivõrk, mis on disainitud tuvastamiseks aluspildil 12 erineva klassi objekte nagu näiteks sõidukeid, inimesi, hooneid ning palju muud [22]. Antud mudel on treenitud Ultralytics poolt arendatud YOLO-v8 selgroo ja sünteetiliste andmete baasil suutes tuvastada aluspildil olevaid objekte ligikaudu 10 meetri kuni sateliidifotode pealt [22].

#### YOLO-v8

YOLO-v8 on kõige värskem versioon samanimeliste mudelite perekonnast olles kiire ning täpne objektituvastusülesannetes. Antud versiooni muudab kiireks ja efektiivseks asjaolu, et tavalise libistava akna meetodi asemel, suudab mudel tuvastada objekti pildil vaid üheainsa pärilevisammuga (*forward pass*), muutes objektide tuvastuse sellega oluliselt kiiremaks. Kui jaotada YOLO-v8 komponentideks, koosneb see kolmest peamisest: **selgroost** (*backbone*), **kaelast** (*neck*) ja **peast** (*head*) [23].

**Selgroog:** Selgroog on konvolutsiooniline närvivõrk, mille ülesandeks on sisendpildist oluliste omaduste ehk tunnuste eraldamine [23]. Selle puhul kasutab YOLOv8 spetsiaalset CSPDarknet53

struktuuri, rakendamaks osalisi ristühendusi kihtide vahel, et parandada informatsiooni liikumist ning suurendada mudeli täpsust [23].

**Kael:** Kael ehk omaduste kombineerija, ühendab erinevates selgroo etappides saadud tunnускаardid, et haarata infot erinevatelt suuruskaaladelt [23]. YOLOv8 arhitektuur kasutab traditsioonilise omaduste püramiidvõrgu (FPN) asemel C2f-moodulit, mis suudab kombineerida kõrgtasemel semantilist infot ja madalatasemelist ruumilist infot, parandades sellega just eriti väikeste objektide tuvastamise täpsust [23].

**Pea:** Pea vastutab lõplike ennustuste tegemise eest, mille juures kasutab YOLOv8 mitmeid tuvastusmooduleid, mis ennustavad iga tunnускаardi ruudus lahtris piiravaid kaste, objekti tõenäosust ning klassi tõenäosust [23]. Lõpetuseks need ennustused kombineeritakse, et saada lõplikud tulemused.

Kuna WALDO30 on treenitud ülevalt nähtavate objektide tuvastamiseks, siis valiti antud põhjusel mudel ka magistritöö ülesandepüstituse lahendamiseks.

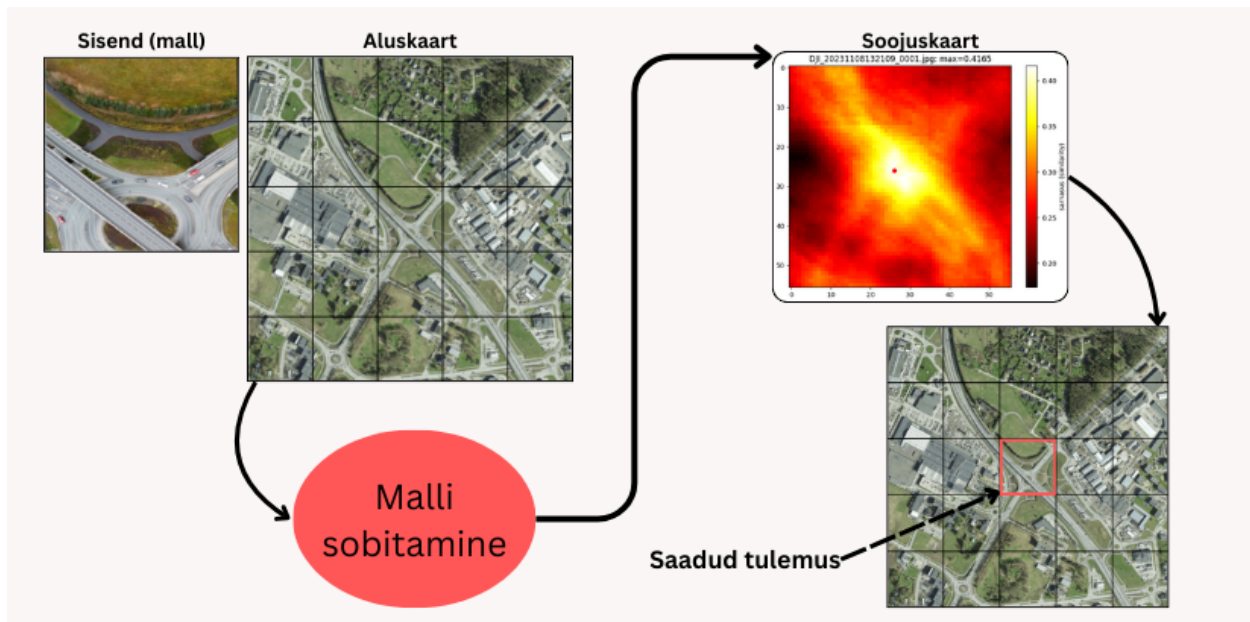
## 1.4 Malli sobitamine

Malli sobitamine (*template matching*) on meetod, kus algoritm proovib leida pildi või piltide vahelisi sarnasusi sõltumata pildi kontrastist, eredusest või värvist. Malli sobitamine on üks enim kasutatavaid meetodeid objekti tuvastamiseks pildilt kuna selle kasutuselevõtt on võrdlemisi lihtne ning paindlik [24].

Antud meetodi kasutamine eeldab kahe peamise komponendi olemasolu: sisend, mida hakatakse inspekteerima ning mall millega liigutatakse üle sisendi ülekattumusega sarnasuste tuvastamiseks (vaata Joonis 8). Samuti peab mall olema pööratud sisendi suhtes samasse suunda, sest vale nurga all olles ei pruugi mudel sama objekti sisendilt tuvastada.

Malli sobitamine koosneb erinevatest etappidest. Esiteks nagu eelnevalt mainitud tuleb valida välja objekt malli näol, mida pildilt on vaja leida, ning seejärel sisend, millel objekt on olemas ning mille asukohta tuleb tuvastada. Seejärel liigutab algoritm malli üle endast suurema sisendpildi piksel piksli haaval, paremalt vasakule ning ülevalt alla käies üle kogu sisendpildi. Samal ajal arvutatakse igal positsioonil olles kahe erineva pildi sarnasust, mis on malli sobitamise üks levinumaid arvutusmeetodeid [25]. See tähendab, et algoritm leiab piltide omavahelise sarnasuse, võrreldes vastavalt malli ja sisendpildi paiknevuse piksleid. Selle arvutuse jaoks on mitmeid võimalusi: erinevuste ruutude summa (SSD), normaliseeritud ruutude summa erinevus (NSSD), ristkorrelatsioon (CC), koosinus sarnasus (CS), normaliseeritud ristkorrelatsioon (NCC),

korrelatsioonikoefitsient (CCOEFF) ja normaliseeritud korrelatsioonikoefitsient (NCCOEFF) [25].



Joonis 8. Töös kasutatud malli sobitamise meetodi kokkuvõtlik toimimisloogika.

Antud töös rakendatakse sarnasuste leidmisel koosinus sarnasust, mille valem (1) on järgmine [26<sup>2</sup>]:

$$score(i, j) = \frac{M_{ij} \cdot D}{\|M_{ij}\| \|D\|} \quad (1)$$

- $M_{\{i,j\}}$  – tunnusmaatriksi akna tunnusvektor, mille vasak ülemine nurk asub pikslil  $(i, j)$ .
  - Tegelikult on  $M_{\{i,j\}}$  3-mõõtmeline tensor (laius x kõrgus x sügavus), kus sügavus = filtrite ehk tunnускаartide arv. Laius ja kõrgus on määratud droonifoto poolt.
- $D$  - droonifoto tunnusvektor (sama pikk kui  $M_{\{i,j\}}$ ).
- $\cdot$  - punktkorrutis.
- $\| \cdot \|$  - eukleidiline ( $\ell^2$ ) norm.

Antud meetodi kasuks räägib asjaolu, et see on piltide omavahelise sarnasuse leidmisel üks robustsemaid meetodeid mõistmaks semantilist sarnasust piltide vahel [27]. Koosinus-sarnasuse arvutamiseks tehakse esmalt kahe vektori skalaarkorrutis, mille väärtus näitab, kui hästi

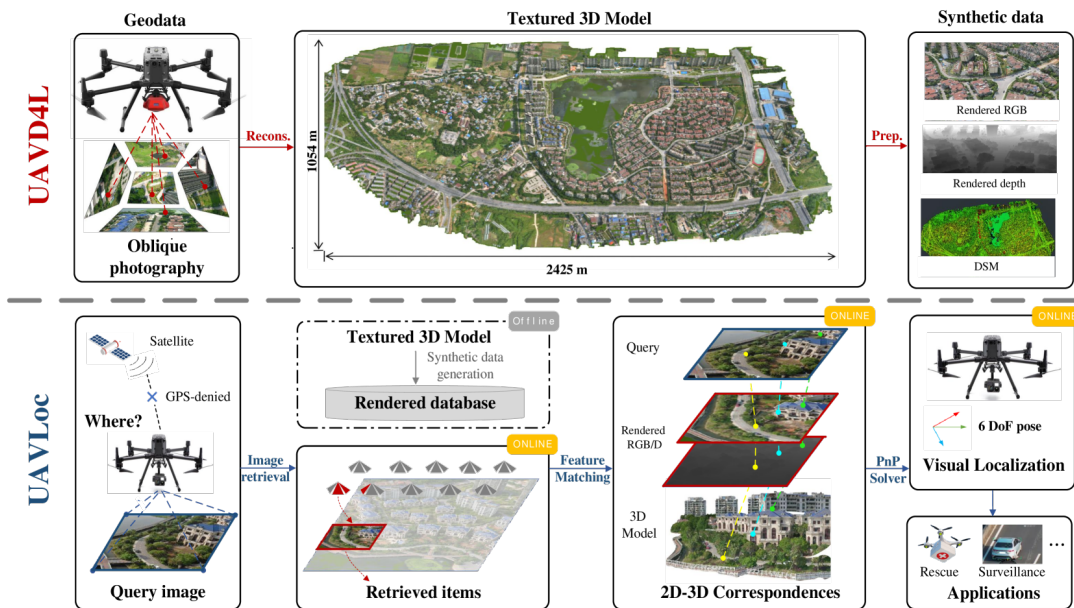
<sup>2</sup> <https://chat.openai.com>.

vektorid ühes suunas joonduvad [27] Seejärel leitakse mõlema vektori pikkused ning jagatakse skalaarkorrutisega, mille tulemusel saadakse normaliseeritud koosinus -sarnasus, mille väärtused on 0 ja 1 vahel, kus 0 on kõige vähem sarnasem ja 1 kõige rohkem sarnasem [27].

## 1.5 Kirjanduse ülevaade

Droonide positsioneerimiseks kasutatakse peamiselt GPS tehnoloogiat. Aastatega on proovitud leida erinevaid alternatiive GPS-i asemel ning seoses sügavõppe mudelite arenguga on üha enam esilekerkinud pildipõhiste asukohatuvastuste meetodid. Just pildipõhine asukohatuvastus on erinevates artiklites olnud autorite arvates läbimurdepunktiks positsioneerimaks UAVd seal, kus GPSi või GNSSi kasutamine ei ole võimalik. Kuigi tulemused on erinevates teadustöodes väga paljulubavad, esineb artiklites ka lahenduste edasiarendusi pärssivaid asjaolusid. Kaardistamiseks ja analüüsimeks hetke olukorda alternatiivsete UAVde positsioneerimise meetodites, tuuakse edasiselt esile antud hetke ühed asjakohasemad artiklid meetodi rakendamises.

Autorid Rouwan Wu, Xiaoya Cheng, Juelin Zhu, Xuxiang Liu, Maojun Zhang ja Shen Yan pakuvad enda artiklis [28] välja meetodi täpse 3D kaardi modelleerimiseks, et droon saaks oma positsiooni kaardil tuvastada. Ülevaade pakutavast lahendusest on kujutatud Joonis 9. Lisaks vertikaalsetele maapinna suhtes täisnurga all (*nadir*) fotodele tehakse ka pilte erinevate kallete alt, luues sellega 2,5 miljonit m<sup>2</sup> katva ruumilise kaardi. Iga punkt saab täpsed GPS-koordinaadid (x, y, z) ning võimaldab määrata drooni täpse orientatsiooni (*yaw, pitch, roll*). Kuna GPS võib erineda tegelikust asukohast 3–10 m, kaamerapilti võib mõjutada valgusmüra ja inertsiaalandur (IMU) võib sisaldada vigu, kontrollitakse kogu kaart visuaalse sobitamisega, et määrata puhas tõde (*ground truth*). Selle põhjal genereeritakse hulgaliselt sünteetilisi andmeid: RGB-pildid, sügavuskaardid ja digitaalsed pinnamudelid (DSM).



Joonis 9. Laiapõhjaline andmekogu UAV 6-DoF lokaliseerimiseks [6].

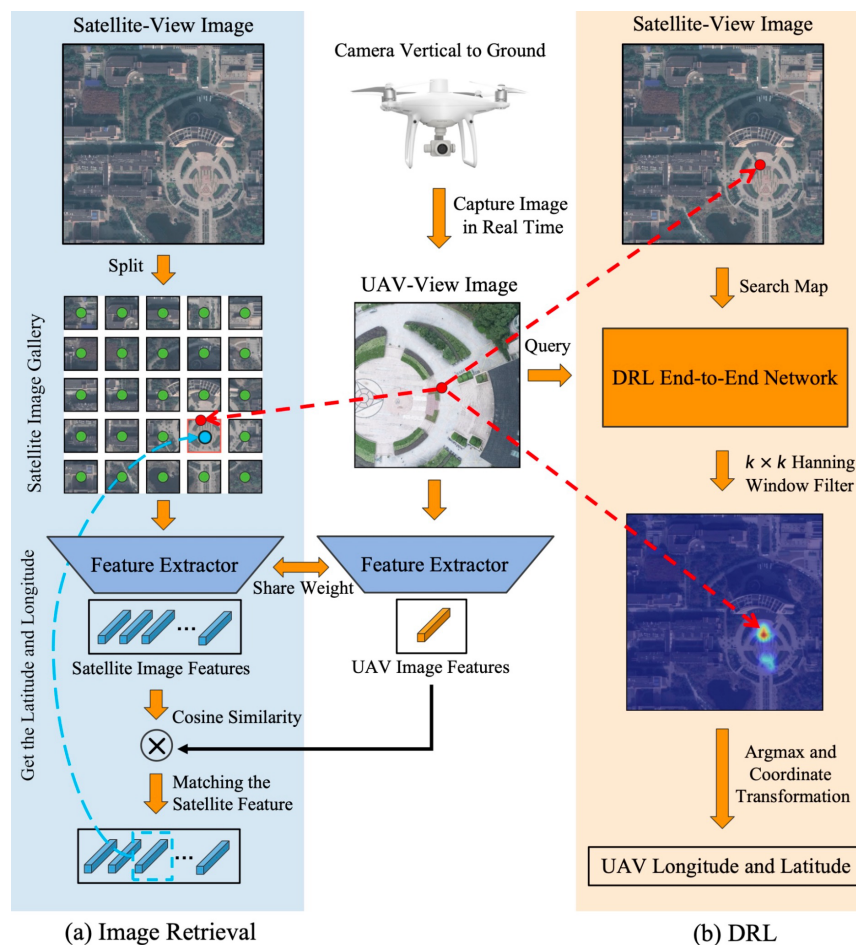
Drooni positsioneerimiseks rakendatakse mitmeastmelist lähenemist. Esmalt kasutab süsteem IMU sensorit, et kitsendada otsinguala. Seejärel toimub droonikaamera pildi ja 3D kaardi tunnuste sobitamine, otsides sünteetiliselt piltidelt drooni asukohta (*Top-k image retrieval*) [28]. Pärast seda teisendatakse kaamera tuvastatud 2D punktid kaardi 3D punktideks ning drooni positsiooni ja orientatsiooni määramiseks kasutatakse PnP + RANSAC meetodit. Kuna IMU annab teada gravitatsiooni suuna, stabiliseeritakse PnP lahendus selle abil, mis kiirendab asukoha leidmist, vähendab vigu ja tagab usaldusväärse drooni asendi lokaliseerimise ka kiirel või kaldunud lennul [6].

Autorite väljapakutud lahendus [28] saavutab suurepäraseid tulemusi, suutes tuvastada objekti meetri täpsusega. Kuigi tulemus on muljetavaldav, on käesoleva töö autori hinnangul lähenemisel ka mitmeid puudusi, mis pärsiks sellise lahenduse paindliku kasutamist lahingolukorras:

- Esiteks on antud lähenemise paindlikkus ääretult väike kuna suudab toimida vaid juhul, kui lennu piirkonnast on modelleeritud 3D kaart. Kuigi mõnel juhul ei pruugi vajadust drooni väljaspool teatud piirkonda lennutada olla, siis antud magistr töö kontekstis artiklis pakutud lahendus ei toimiks. Seda põhjusel, et 3D kaardi töötlemine nõuab võrreldes ortofotodoga rohkem arvutusressursi ning see tähendaks omakorda pikemat asukoha tuvastusaega.

- Teiseks vajab antud lähenemine suuremahulist kaardistustööd ning andmeid, mis võib magistritöö teema kontekstis saada lahenduse arendamisel saatuslikuks. Kuna Ukraina drooniarenduse näitel toimub tehniliste arenduste väljatöötamine vajadust arvestades kiirelt, siis oleks antud meetodi laialdasem väljatöötamine liiga ajakulukas.

Teiseks lahenduseks pakuvad autorid Ming Dai, Enhui Zheng, Jiahao Chen, Lei Qi, Zhenhua Feng ja Wankou Yang artiklis [1] droonipõhist lokaliseerimist (DRL), kus drooni asukoht määratakse ainult piltide abil. Selleks luuakse otsast lõpuni (*end-to-end*) sügavõppesüsteem, mille eesmärk on sobitada droonikaamera pilt satelliidipildiga ning anda ennustusena soojuskaart (*heatmap*) [1]. *End-to-end* tähendab siin, et mudel ühendab drooni- ja satelliidifunktsioonid võrgu sees ning loob ennustuse otse. Joonis 10 on kuvatud antud autorite poolt väljapakutud lahendus.



Joonis 10. Tõhus heterogeensete ruumiliste tunnuste interaktsiooni meetod UAV isekaliseerimiseks [1].

Autorite pakutud lahendus algab kahe sisendiga: UAV pilt (128x128 px) ja kolmekordse suurusega satelliidipilt (384x384 px). Mõlemad pildid jaotatakse väikesteks plokkideks ja seotakse järjestusse (sequence), vajadusel transformeri kaudu. Järgnevalt toimub tunnuste eraldamine, milleks kasutatakse kahte arhitektuurivarianti: järelmestimine (Post-Fusion) (kahe selgrooga) ja segumestimine (Mix-Fusion) (ühe selgrooga). Järelmestimises töödeldakse UAV ja satelliidipilte eraldi transformeritega, seevastu segumestimine ühendab mõlemad sisendid varakult samasse transformerisse, lubades tunnustel omavahel suhelda läbi tähelepanuplokkide. Autorid katsetavad mõlemat arhitektuuri, et hinnata, kas integreeritud või eraldi töötlemine annab paremaid tulemusi.

Järelmestimises kombineeritakse tunnuskaardid tunnuste mestimise plokkis (FFB), kasutades grupi konolutsiooni (GC), risttähelepanumestimise (CAF) või enesetähelepanu mestimise (SAF) meetodeid. Segumestimise puhul toimub tunnuste ühendamine juba varasemas etapis. Edasi viiakse läbi mitmetasandiline täiustamine multi-skalaarse mudeli (MSM) kaudu, kus ühendatakse erinevad skaalad ja sügavamad kihid, et tugevdada nii detaile kui ka suuremaid mustreid [1]. Ainult segumestimisse lisandub kanalite ühendamine (*channel fusion*), mis vähendab mälu kasutust. Viimases etapis toimub piltide suurendamine tagasi originaalsuurusesse (384x384 px), mille tulemusel saadakse soojuskaart, kus iga piksel näitab drooni paiknemise tõenäosust.

Autorite väljapakutud lahendus on oma olemuselt sarnane käesoleva magistr töö lahendusega. Sellegipoolest erineb artiklis esile toodu käesoleva magistr tööga selles osas, et ei tükeldata autori poolt pakutud lahenduses otsinguala. Lisaks leitakse antud artikli puhul puudujääke, mis võivad muuta lahenduse valmisoleku erinevates olukordades puudulikuks.

- Esiteks vajab antud artiklis pakutud lahenduse rakendamine tuhandeid drooni- ja satelliitfotode paare erinevates asukohtadest, et muuta lahendus paremini üldistavaks uute asukohtade puhul. Konkreetsete droonipiltide ja satelliitfotode paaride treenimine ei pruugi aga eeltreenitud mudelite võrdluses palju juurde anda.
- Teiseks on taaskord tegemist arendust silmas pidades ajakuluka lähenemisega kuna pildipaaride ettevalmistus treeninguks nõuab korraliku ajalist ressursi.

Viimasena uuriti ka kolmandat alternatiivi, milles autorid Alrich A. Cabrera-Ponce ja Jose Martinez-Carranza pakuvad enda artiklis [29] välja võimaluse, kus üheainsa kaamerakaadri põhjal hinnatakse drooni  $[x, y, z]$  asukoht sügavõppe abil. Lähtekoht on väike käsitsi kogutud pildikorpus, mille igale foto-le on lisatud vastav GPS-koordinaat ja kõrgus. Fotod normaliseeritakse ühtlaseks sisend-suuruseks ning koordinaadid teisendatakse meetermõõdustikku.

Pildipõhiseks positsioneerimiseks rakendatakse kergendatud PoseNeti varianti (*Compact PoseNet*), kus liigsed ja mahukad moodulid on eemaldatud, et saavutada parem kiirus, loobumata oluliselt täpsusest. Erinevaid mudeliversioone treenitakse Adam-optimeerijaga ja võrreldakse klassikaliste CNN-ide (nt PoseNet, VGG, ResNet) vastu mitme drooni-trajektoori peal.

Tulemused näitavad, et kergendatud võrgu täpsus jääb originaal-mudelile väga lähedale, kuid läbilaskevõime on enam kui kahekordne. Selline lahendus võimaldab reaalajas visuaalset drooni-geolokaliseerimist ilma kaardi, SLAM-i või mitme kaamera süsteemita – piirdudes üksnes tavalise RGB-kaadriga.

Kuigi saavutatud tulemused on artiklis väljatoodud lähenemisel head, siis on autori hinnangul magistritöö fookust arvesse võttes järgmised kitsaskohad:

- Lahendus vajab suurt treeningandmestikku konkreetsest piirkonnast, üldistamine teistes piirkondadesse võib olla ebatäpne ning seetõttu ei pruugi lahendus olla töökindel kaardistamata aladel.
- Kõrgema täpsuse saavutamiseks võivad vajalikud olla täiendavad andmed või keerukamad arhitektuurid.
- Väljapakutud lahendus vajab hoolikalt kalibreeritud andmestiku sarnase vaatenurga järgi, mille tõttu mudel ei pruugi toimida, kui droon lendab väga erineval kõrgusel või vaatenurgaga kui treeningandmestikus. See võib muuta lahenduse ebausaldusväärseks käesoleva magistritöö teema kontekstis.

## 2 Metoodika

Alljärgnevas seksioonis kirjeldatakse magistritöö metoodika põhipunkte: andmete valimist ja eeltöötlust, reaalse lahenduse ning prototüübi ülesehitust ning eri eksperimentidest mõjude analüüsimiseks. Esiteks selgitatakse, kuidas droonifotode metaandmeid ja ortofotosid ühtsesse koordinaatsüsteemi teisendatakse ning milliseid meetodeid kasutati visuaalse sobituse jaoks. Seejärel tutvustatakse samm-sammult reaalse lahenduse töövoogu, mille aluseks on IMU-põhine eelhindang ja koosinussarnasusel põhinev mallisobitus. Lõpuks käsitletakse prototüübi loogikat ning eksperimente, millega hinnatakse süsteemi täpsust nii kõrguse kui pöördnurkade suhtes.

### 2.1 Andmete kättesaamine ja valim

Antud magistritöö kasutatud andmeteks on 113 drooni aerofotot JPG formaadis. Andmed pärinevad ettevõtte 3DI poolt Tartu linna ortofotode jaoks teostatud droonilendudelt. Piltide dimensioonideks on 8192x5460 ning samuti sisaldavad fotod metaandmeid. Pilte ei ole eeltöödeldud ning valimi puhul võtsime arvesse mitu erinevat aspekti:

- **Fotode eriilmelisus ja asukoha keerukus:** Esiteks on fotode valimi puhul oluline nende eriilmelisus, et testida eeltreenitud mudeli suutlikust võrrelda omavahel erinevaid mustreid kaardi ja droonifotode vahel. Eriilmelisuse tagamiseks on valitud fotod üksteisest erinevad asustuse, kõrguse ning foto suuna osas. Näiteks on valitud andmetes fotosid metsatukkadest, raudteest, tihedatest hoonestustest ning ringtedest.
- **Kompaktse suurusega valim:** Kuna antud magistritöö eelduseks oli, et eeltreenitud mudelid suudavad piltide omavahelisi mustreid tuvastada, siis ei peetud antud magistritöös valimi suurust määravaks tulemuste olulisuse tagamisel. Seda põhjusel, et suurem valim oleks tulemuste kontrollimise teinud ajamahukamaks, kuid poleks tulemuste õigsusele väärtust juurde lisanud. Seetõttu keskenduti valimi koostamisel sellele, et kõik pildid testiks eeltreenitud mudeli võimekust eri nurkade alt.
- **Tartu keskne valim:** Viimaseks aspektiks valimi koostamisel oli piltide asukoha määramine. Kuna antud magistritöö raames oli võimalus kasutada Tartu ortofoto andmekogust saadud pildifaile, siis keskenduti töös antud piirkonnale. Samuti oli drooni lennutrajektoori Tartu väiksuse tõttu oluliselt lihtsam arvutada ning visuaalselt paremini kaardile jälgimiseks kuvada. Lisaks sellele oli magistritöö algusetappides tänu Tartu kompaktsusele algoritmi tulemusi oluliselt lihtsam käsitsi kõrvutada ette antud koordinaatide põhjal saadud vastetega.

## 2.2 Reaalse lahenduse ülesehitus

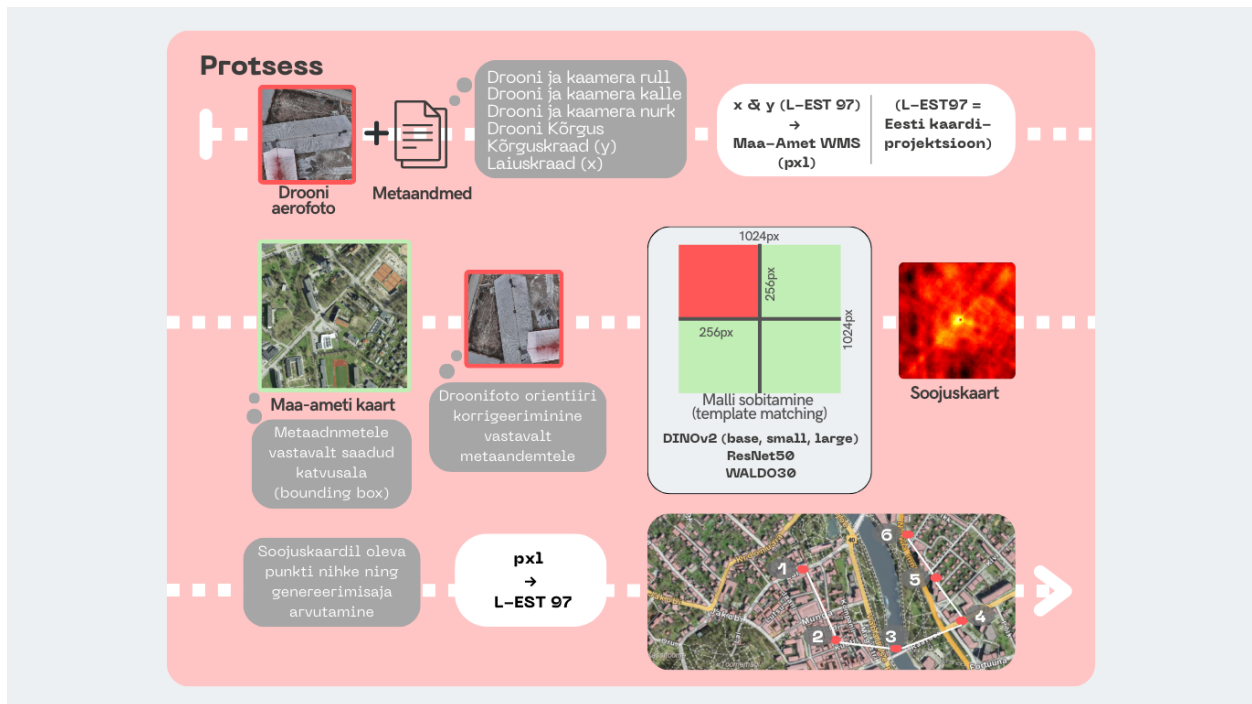
Antud magistritöös pakutakse välja visuaalipõhine drooni positsioneerimise süsteem, mis suudaks määrata droonifoto geograafilise asukoha ainult pildi ja selle metaandmete põhjal, ilma GNSSi (GPSi) või muude välistingimustest sõltuvate signaalideta. Antud töös on oluline eristada kahte väljapakutud lähenemist: reaalselt rakendatavat lahendust ning prototüüpi. Prototüüp on lahenduse idee testimine ilma riistvaraliste komponentide olemasoluta. See tähendab, et töös pakutakse välja üks GPSi vaba alternatiiv, mille rakendamist testitakse eelnevalt prototüübi abil.

### Esmalt tuuakse välja reaalse rakenduse töövoog:

- Drooni lähtekoordinaatide fikseerimine**  
Enne iga lennu määratakse droonile täpne stardipunkt WGS-84 GPS-koordinaatides. Need koordinaadid projitseeritakse süsteemi, mis ühtib drooni kaasarvutisse laaditud GeoTIFF-aluskaardi projektsiooniga, millega välditakse hilisemaid koordinaadi teisendusi. Stardikoordinaat toimib nullpunktina, millega võrreldakse nii IMU-põhist liikumist kui ka hiljem saadavat visuaalset koordinaati.
- IMU hinnang drooni hetke-asukohale**  
Lennu ajal integreerib IMU inertsimõõtmised ja väljastab näiteks iga teatud lennu intervalli tagant ligikaudse asukohavektori  $[x,y,z]$ . Praktikas võib drooni triiv olla 10-100 m iga minuti kohta [33], aga selline täpsus on piisav, et kitsendada otsinguala kaardil ja vähendada mallisobituse arvutusmahtu. IMU väljund toimib suhtelise hinnanguna drooni oletatavale asukohale ja suunale.
- Mallisobituse käivitamine absoluutse koha leidmiseks**  
Iga droonikaader vähendatakse ja pööratakse kõrgusmõõdiku-põhise kõrguse ja kaamera FOV järgi. Seejärel libistab algoritm lõike **tunnuskaarti**, mis on loodud eeltreenitud mudeliga, üle GeoTIFF-kaardi tunnuskaardi, arvutades igas asendis koosinussarnasuse. Kõrgeima sarnasusega asukoht kaardil teisendatakse pikselkoordinaatidest tagasi algseesse koordinaatsüsteemi. Nii kinnitatakse või korrigeeritakse IMU hinnang ning kaadrile omistatakse GPS-ist sõltumatu absoluutne koordinaat.

## 2.3 Prototüübi ülesehituse loogika

Prototüübi puhul imiteeritakse reaalse rakenduse kasutamist, et testida töös väljapakutud reaalse lahenduse tulemuslikkust (vaata Joonis 11). Prototüüp on ka antud töö keskne idee, tänu millele on võimalik selgitada välja väljapakutud reaalse lahenduse kitsaskohti ning on esmaseks eelduseks riistvaraliseks edasiarenduseks.



Joonis 11. Väljapakutud prototüüp lahenduse kokkuvõtlik ülevaade.

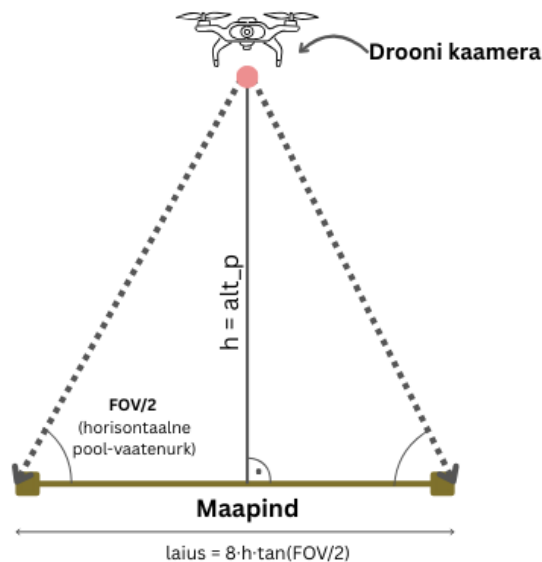
Lahenduse prototüübi töövoog on järgmine:

1. **Metaandmete lugemine EXIF-ist** - iga algne droonifoto avati ning sellest eraldati põhilised metaandmed: GPS-laius- ja pikkuskraad, lennukõrgus, kaamera vaateväli (FOV) ning gimballi ja drooni (kalle, nurk, roll) nurgad. Need väärtused moodustavad edasiste ruumiarvutuste sisendi. Kuna antud magistritöös puudub IMU riistvaralise rakendamise võimekus on antud etapp oluline imiteerimaks drooni reaalselt hinnangulise koordinaadi tuvastamist kasutades piltides olevaid metaandmeid. Reaalses lahenduses ei toimuks EXIF lugemist kuna sellisel juhul kasutataks ilma töötlemata drooni pilti. EXIF on vaid omane JPG vormingule.

2. **Koordinaatide teisendamine L-EST97 süsteemi** - WGS84 koordinaadid konverteeriti Eesti riiklikku tasapinnalisse süsteemi (EPSG 3301). See tagas, et hiljem veebist hangitav ortofoto ja kõik järgnevad punkt- ning kaugusarvutused toimuksid ühes ja samas koordinaatsüsteemis.
3. **Maa-ameti WMS-kaardi allalaadimine** - droonifoto asukoha ümber moodustati neljakordse katvusega ala, mille suurst skaleeriti dünaamiliselt foto tegeliku lennukõrguse ja kaamera FOV järgi. Selleks kasutati järgmist valemit (2), mis on visualiseeritud joonisel Joonis 12:

$$laius = 4 \times 2h \tan\left(\frac{FOV}{2}\right) = 8h \tan\left(\frac{FOV}{2}\right) \quad (2)$$

- $h = \text{alt\_p} = (\text{GPS-kõrgus} - \text{ground\_offset})$ .
- kui EXIF-kõrgus ( $h$ ) puudub, siis ignoreeritakse pilti.
- kui korrigeeritud kõrgus  $< 0$  m, siis ignoreeritakse pilti.
- $\text{fov\_deg}$  - kaamera horisontaalne FOV kraadides (puudumise korral pilti ignoreeritakse).

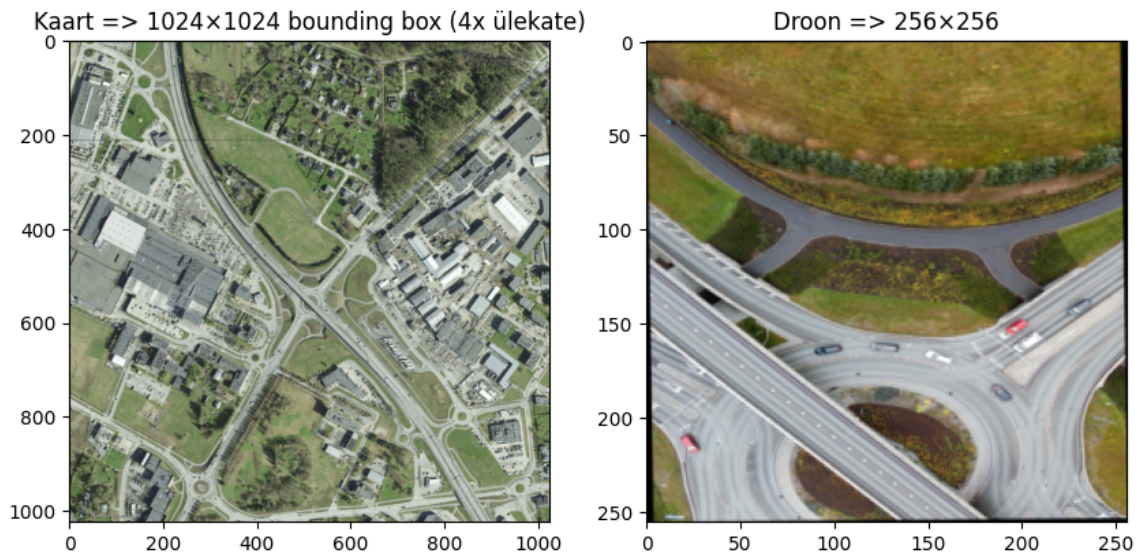


Joonis 12. Drooni katvusala arvutamine kasutades lennukõrgust ja kaamera FOV-i.

Selle alusel laaditi Maa-ameti WMS-teenusest 1024x1024 px ortofoto, et tagada piisav detailsus visuaalseks sobituseks. Allalaetud pildi suuruse juures on oluline, et see sisaldaks

piisavalt konteksti, mida oleks võimalik droonifotoga valesti sobitada leidmaks õiget asukohta.

4. **Droonipildi pööramine ja lõikepildi (*crop*) tegemine** - droonifotot korrigeeriti pöördenurga alusel ning seejärel lõigati keskkohast 256x256 px fragment, mis vastas lennukõrgusest tulenevale maapinna katvusele (vaata Joonis 13). See tagas mõõtkava ühtlustamise võrreldes allalaetud kaardiga.



Joonis 13. Maa-ameti WMS allalaetud kaart ning korrigeeritud droonifoto.

5. **Tunnuskaardi eraldamine kaardilt ja droonipildilt** - nii ortofoto kui ka kaart normaliseeriti ning söödeti sügavõppemudelitesse nagu DINOv2 *base*, *small*, *large* ning ResNet50 ja WALDO30. Mudeli viimase peidetud kihi aktivatsioonid vormiti ruumiliseks tunnuskaardiks, mis säilitas nii kõrgtasemelise semantilise info kui ka ruumilise paigutuse.

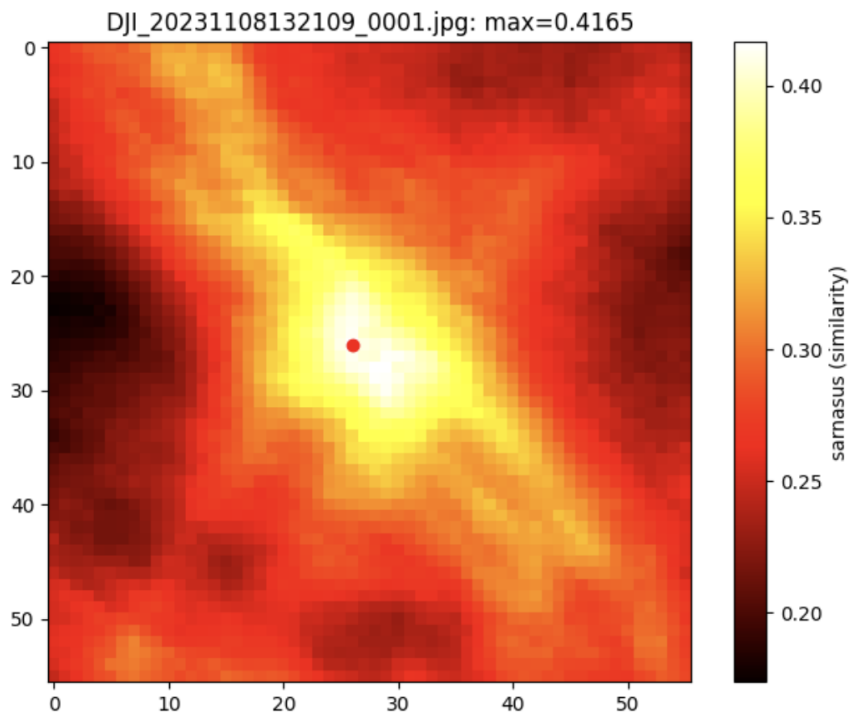
DINOv2 mudelid erinesid üksteisest oma parameetrite arvu poolest:

- *dinov2-base* - 86M parameetrit, keskne kompromiss täpsuse ja kiiruse vahel [30].
- *dinov2-small* - 22M parameetrit, kiirem aga madalama detailsusega [31].
- *dinov2-large* - 304M parameetrit, kõige täpsem aga aeglasem ja suurema GPU mälunõudlusega [32].

<sup>3</sup><https://huggingface.co/facebook/dinov2-base>

Kõigi antud DINOv2 mudelite rakendamise puhul kasutati sama koodistruktuuri ning rakendamiseks vahetati vaid koodi töövoos veebiaadressi nime (vaata Lisa II). Samuti leiab lisadest ResNet50 (vaata Lisa I) ja WALDO30 (vaata Lisa III) rakendamisest põhjalikuma ülevaate.

6. **Koosinussarnasuse (CS) põhjal sarnasuse soojuskaardi leidmine** - droonipildi tunnuskaarti libistati konvolutsioonina üle ortofoto tunnuskaardi. Iga positsiooni jaoks arvutati CS, millega kõrvaldati amplituudist sõltuvus ning saadi mõõdik, mis sobib nii valgustuse kui kontrasti kõikumise korral. Soojuskaardi maksimum tähistas punkti kaardil, mis visuaalselt kõige paremini vastas droonifotole (vaata Joonis 14).



Joonis 14. Koosinus sarnasuse põhjal genereeritud soojuskaart ning tuvastatud asukoht.

7. **Positsioneerimisvea arvutamine** - soojuskaardi maksimaalse koordinaadi projektsioon L-EST97 süsteemi võrreldi drooni tegeliku GPS-põhise asukohaga. Nende kahe punkti eukleidiline kaugus andis konkreetse foto geolokaliseerimisvea meetrites.

---

<sup>4</sup><https://huggingface.co/facebook/dinov2-small>

<sup>5</sup><https://huggingface.co/facebook/dinov2-large>

8. **Statistiliste näitajate arvutamine** - kõigi testfotode kohta koguti nihked ja soojuskaardi genereerimise ajad. Iga kõrguse või katsetingimuse kaupa arvutati keskmine, minimaalne ja maksimaalne viga ning standardhälve. Samu näitajaid hinnati ka töökiiruse (ms/pilt) osas.
9. **Üldkokkuvõtte koostamine** - koondtabelis esitati pildi-, vea- ja kiirusstatistika, mille põhjal hinnati valitud mudeli ning CS-põhise sarnasusmeetodi täpsust ja reaajas kasutatavust. Lõpuks visualiseeriti drooni trajektoori ortofoto taustal, koos iga pildipunkti järjestuse ja arvutatud nihkega, et anda selge ülevaade lahenduse ruumilisest täpsusest kogu lennu vältel.

## 2.4 Droni eri kõrguste mõju saadud tulemustele

Uurimaks väljapakutud kontseptsiooni kitsaskohti, hinnati ühe eksperimendina, kuidas simuleeritud lennukõrgus mõjutab mallisobitusel põhineva geolokaliseerimise täpsust. Ühe dronifoto EXIF-andmetele tuginedes lõigati samast pildist seitsme kõrguse (120, 80, 60, 40, 30, 20 ja 10 m) katvusele vastavad kärped. Lisaks arvutati iga kõrguse põhjal, kui palju meetreid kaamera vaatevälja ühe piksliga kaardil vastab (px\_per\_m), et kärpe mõõtmed pikslites täpselt õigeks muuta. Iga kärbe võeti pildi keskelt, skaleeriti 256x256 px suuruseks ning seejärel arvutati igale kärpele soojuskaart ja leiti argmaxi positsioon, mille alusel määrati geograafiline nihe.

Iga kõrguse korral eraldati tunnused nii kärpest kui ka Maa-ameti WMS-ist hangitud vastava ortofoto lõigust, arvutati koosinus sarnasuse soojuskaart ning määrati maksimaalse sarnasuse punkt. Vastava ortofoto hankimisel, jäi selle suurus kõrguse muutmisel samaks. Maksimaalse sarnasuse punkt ja EXIF-koordinaatide eukleidiline kaugus tähistas kõrguse-spetsiifilist nihet.

Kõigi kõrguste nihked koguti, neist arvutati keskmine, mediaan, minimaalne ja maksimaalne viga, standardhälve ning mõõtmiste arv. Katse viidi demonstriivselt läbi 10 dronifoto põhjal, pakkudes kiiret ja selget ülevaadet kõrguse mõjust positsioneerimise täpsusele. Tulemused esitati tabelina koos koondstatistikaga, mis näitab, milline lennukõrgus tagab parima täpsuse ja kuidas viga kõrguse muutudes varieerub.

## 2.5 Drooni eri pöördenurkade mõju saadud tulemustele

Teise eksperimendina uuriti, kuidas droonipildi pööramine eri pöördenurkadega mõjutab drooni absoluutse positsioneerimise täpsust. Selleks loodi spetsiaalne katsetusfunktsioon, mis võtab sisendiks droonifoto ja pöörab seda etteantud pöördenurkades ( $1^\circ, 2^\circ, 3^\circ, 4^\circ, 5^\circ, 10^\circ, 20^\circ, 30^\circ, 40^\circ$ ). Funktsiooni sees kärbiti esmalt pilt vaatevälja (originaalkõrgus ja FOV) alusel vastava suurusega plokiks ning seejärel rakendati iga test-pööre, et simuleerida drooni pöördenurka. Pööratud kärped skaleeriti  $256 \times 256$  px mõõtu ja võrreldi Maa-ameeti ortofotoga, genereerides igale nurga väärtusele sojukaardi ja leidmaks argmax-punktist vastava asukoha nihke.

Kõigi 10 pildi katsetatud pöördenurkade nihked kogutakse eraldi loenditesse. Seejärel arvutatakse iga nurga kohta statistilised näitajad: keskmine nihke väärtus, mediaan, minimaalne ja maksimaalne nihe, standardhälve ning mõõtmiste arv. Tulemused esitatakse tabelina, mis annab selge ülevaate, millised pöördenurgad põhjustavad positsioneerimisel suurima või vähima vea.

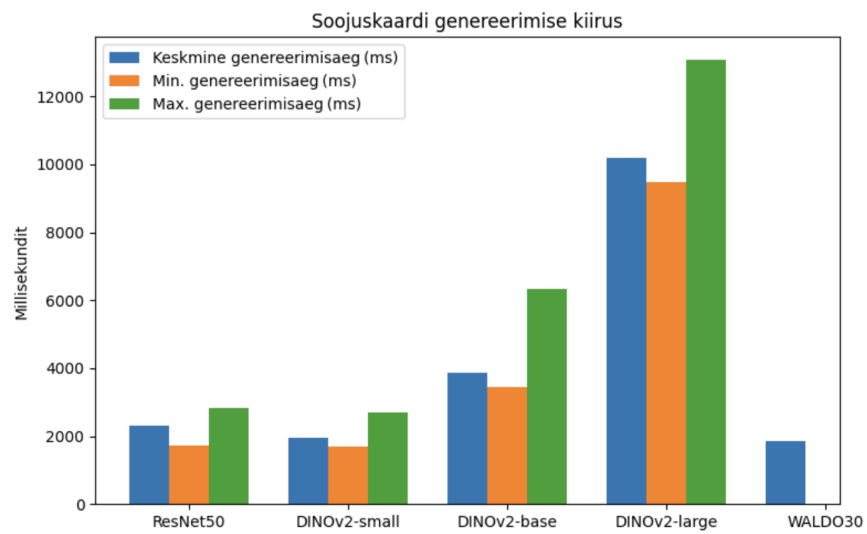
Lisaks koostatakse üldtabel, mis sisaldab koondstatistikat kõigi pöördenurkade lõikes, võimaldades hinnata meetoodika üldist täpsust sõltumata konkreetsest pöördenurgast. Selline lähenemine võimaldab kvantitatiivselt hinnata, kuidas droonipildi pööramine mõjutab mallisobitusel põhinevat positsioneerimist, ning pakub olulist sisendit süsteemi optimeerimiseks ja mudeli täpsuse parandamiseks.

### 3 Tulemused

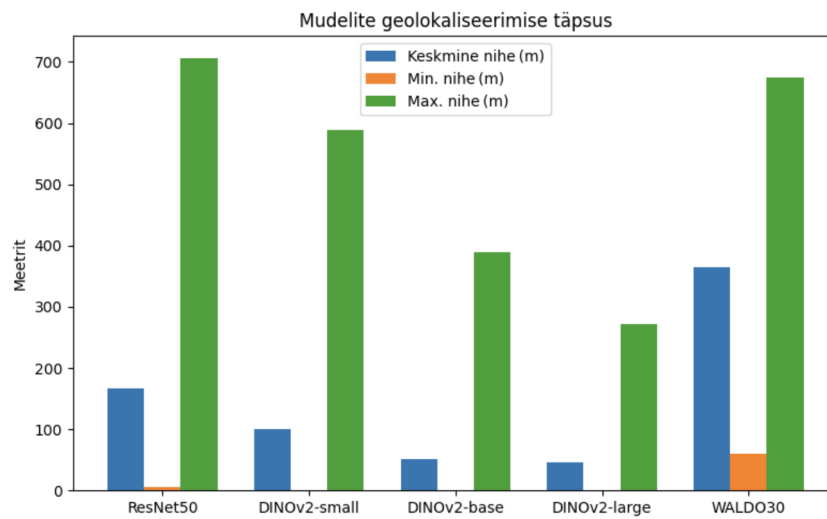
Antud peatüki eesmärk on võrrelda kolme eeltreenitud mudelite-ResNet50, DINOv2 (*base, small, large*) ja WALDO30 - drooni- ja ortofotode sobitamise ülesandes. Analüüsitakse, kuidas iga mudeli arhitektuur mõjutab positsioneerimise täpsust, töökiirust ja arvutusressurssi. Soojuskaarte genereeriti lokaalselt Macbook M2 Pro 10 tuumalise CPU peal, mille kiirus on 3.49 GHz. Saadud tulemused aitavad valida, milline lähenemine sobib kõige paremini antud magistritöö probleemipüstituse potentsiaalseks lahendamiseks. Tabel 1 on nähtavad mudeli sobitamise tulemused. Lisaks on visualiseeritud joonisel *Joonis 15* ja joonisel *Joonis 16* mudelite omavaheline erisus võimekuse osas.

Tabel 1. Mudeli sobitamise tulemused.

	<b>ResNet50</b>	<b>DINOv2- base</b>	<b>DINOv2- small</b>	<b>DINOv2- large</b>	<b>WALDO30</b>
<b>Piltide arv</b>	113.0	113.0	113.0	113.0	113.0
<b>Keskmine nihe (m)</b>	165.07	50.97	100.57	45.97	364.16
<b>Minimaalne nihe (m)</b>	5.90	0.0	0.0	0.0	60.80
<b>Maksimaalne nihe (m)</b>	706.79	388.83	588.53	271.42	673.76
<b>Standardhälve (m)</b>	193.51	71.67	127.30	55.05	148.05
<b>Keskmine soojuskaardi genereerimise aeg (ms)</b>	2299.2	3882.2	1955.1	10191.5	1860.1
<b>Minimaalne genereerimise aeg (ms)</b>	1713.6	3460.2	1707.9	9468.7	-
<b>Maksimaalne genereerimise aeg (ms)</b>	2824.6	6325.5	2707.1	13092.4	-



Joonis 15. Mudelite ülene soojuskaartide genereerimise kiiruse võrdlus.

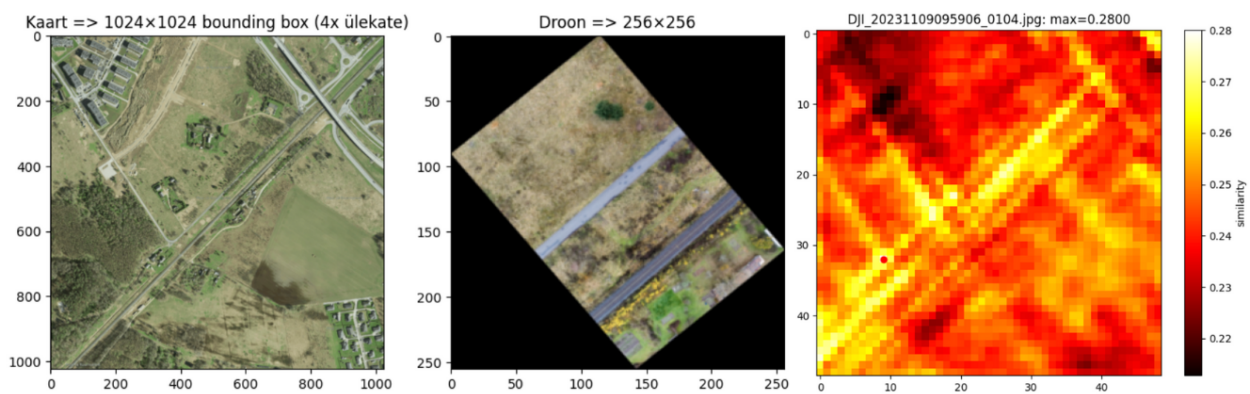


Joonis 16. Mudelite ülene geolokaliseerimise täpsuse võrdlus.

### 3.1 ResNet50

ResNet50 mudel tagastas kokkuvõttes keskpäraseid tulemusi tuvastades küll teatud mustreid ning sarnasusi kuid olles siiski ebatäpne. Mudeli poolt tuvastatud punkti keskmine nihe puhas tõest oli 165.07 m. Minimaalne nihe oli 5.90 m ning maksimaalne nihe 706.79 m.

Ebatäpsuse korral oli ResNet50 puhul tagastatud soojuskaart punkti valesti tuvastamise korral hägune. Isegi, kui kaardil oli tuvastatud teede, hoonete ja maastiku iseärasusi, ei suutnud antud mudeli siiski kokku viia droonifotot ning õiget punkti kaardil (vaata Joonis 17). Ehk siis võis tagastatud soojuskaardil näha olla kaardile omaseid mustreid, kuid mudeli ebakindlus õige punkti osas oli suur. Sellist ebakindlust võis peamisel antud mudeli kontekstis täheldada tiheda asustusega alal, kus katvusosalal oli palju erineva või sama suurusega hooneid. Samuti oli mudelil raskusi haljasalade ning metsarohkete piirkondade tuvastamisega kuna tihti puudusid antud fotodel või kaardi katvusosaladel selgelt iseloomulike objekte, mis võiks aidata kahe sisendi vahelist sarnasust leida.



Joonis 17. Mudeli ResNet50 ebatäpse asukohaga genereeritud soojuskaart.

Tunduvalt edukam oli mudel siis, kui droonifotol olev tee, hoone või objekt oli kaardi katvusala kontekstis unikaalne ning eristus teistest samas klassis olevatest. Selleks võis olla mõni keerdsus tee, iseloomuliku kujuga ristmik või mõne suurem maja.

Sellegipoolest oli ResNet50 mudeli keskmise soojuskaardi genereerimise aeg 2299.2 ms. Soojuskaarte genereeriti lokaalselt Macbook M2 Pro 10 tuumalise CPU peal, mille kiirus on 3.49 GHz. Kuigi asukoha tuvastamise kiirus on antud juhul arvestatav, siis on see tulnud ResNet50 mudeli puhul tuvastustäpsuse arvelt.

Kokkuvõttes sobib ResNet50 mudel pigem üldisemaks klassifikatsiooniks kui täpsete positsioonide leidmiseks, kuna selle tunnusruum ei ole nii spetsiifiline ega orienteeritud lokaalsele vastavusele.

## 3.2 WALDO30 YOLOv8

WALDO30 mudel, mis põhines YOLOv8 arhitektuuril ning oli spetsiaalselt treenitud objektide tuvastamiseks, ei andnud eriti konkurentsivõimelisi tulemusi. Antud mudeli kahe sisendi omavahelise sobitamise keskmine kiirus oli parim kõikidest mudelitest (1860.1 ms), kuna YOLOv8 mudel on optimeeritud reaajas vastete tuvastamise jaoks. Sellegipoolest oli WALDO30 mudeli puhul kahe sisendi vaheline tuvastusvea keskmine suurus 364.16 m. Antud mudel suutis hästi lokaliseerida teede ja hoonete läheduses, kuid kaotas täpsust suuremate avatud alade puhul, kus tunnuste eristusvõime YOLO vahekihis oli piiratud. Mudel võis küll hästi ära tunda erinevaid geomeetrilisi mustreid (nt parkimisplatsid, linnatänavad), kuid ei olnud nii tugev looduslike elementide (nt veekogud, põllud) lokaliseerimisel.

Kokkuvõttes oli WALDO30 mudel nõrk ning ei suutnud probleemipüstitusele head lahendust pakkuda. Kuigi WALDO30 mudeli puhul oli sarnasuse leidmine keskmisel kõige kiirem, siis täpsus oli antud mudeli puhul antud töö probleemi konteksti arvestades liiga madal.

## 3.3 DINOv2 mudelid (*base, small, large*)

DINOv2 mudelid andsid stabiilseid ja ühtlasi tulemusi olles sellega teiste mudelite võrdluses kõige täpsemad. Väiksem versioon pakkus kiiremat otsustusvõimet, kuid sarnasuse kaardil oli sageli rohkem müra, mis võis mõnikord viia vale maksimumini. *Base* ja *large* mudelid suutsid paremini eristada struktuurset sarnasust, kuna nende tunnuskaartidel oli kõrgem semantiline eristusvõime.

### 3.3.1 DINOv2-*base*

DINOv2-*base* oli teiste DINOv2 versioonide ning ka muude mudelite võrdluses kõige efektiivsem pakkudes head tasakaalu kiiruse ja täpsuse vahel. Antud mudel suutis tuvastada enamuse droonifotode ning kaardi sarnasustest ning teha seda mõistliku ajaga. DINOv2-*base* mudeli keskmine nihe oli 50.97 m suutes sisendite omavahelisi sarnasusi tuvastada keskmiselt 3882.2 ms juures. Antud mudeli maksimaalne nihe oli 388.83 m ning minimaalne 0.00 m. Kuigi soojuskaardi

genereerimise kiirus oleks reaalse rakenduse korral kiirem, siis on mudeliga saadud tulemus väga positiivne ning edasist arendust silmas pidades paljulubav.

### **3.3.2 DINOv2-small**

DINOv2-small näitas küll kõikide DINOv2 mudelite võrdluses kõige suuremat keskmist nihet (100.57 m), kuid oli ka pea poole kiirema soojuskaardi genereerimise ajaga kui DINOv2-base mudel (1955.1 ms). Kuigi ka sarnaselt DINOv2-base mudelile ning ka DINOv2-large mudelile oli minimaalse nihke suurus 0 m, siis ei paku antud mudel siiski piisavat täpsust antud probleemipüstitust silmas pidades.

DINOv2-small mudeli kehva tuvastusvea tulemusele võis saatuslikuks saada mudeli liiga madal representatsioonivõime, mis tulenes vähematest kihtidest ja parameetritest. Kuna kaardi ja droonifotode mustrite omavaheline võrdlemine nõuab mudelilt suuremat sügavust, siis tundub, et antud mudeli puhul jäi see väheseks.

### **3.3.3 DINOv2-large**

DINOv2-large näitas väikseimat keskmist nihet kõigist DINO mudelitest (45.87 m), kuid seevastu oli antud mudeli soojuskaardi genereerimise aeg märgatavalt pikem, kui kõigil muudel antud töös rakendatud mudelitel (10191.5 ms). Sellegipoolest oli antud mudeli maksimaalne nihe 271.42 m ning minimaalne taaskord 0 m. Suurem DINOv2 mudel oli parem olukordades, kus kaardil ja droonipildil oli vähe visuaalseid viiteid (nt põllumajanduslikud või metsaalad). Sellegipoolest võib probleemipüstitust arvesse võttes soojuskaardi genereerimisaeg saada määravaks.

Kuigi kahe sisendi vaheline tuvastatud nihe on antud mudeli puhul kõige madalam, siis üldjoontes on mudel antud probleemipüstituse jaoks üleliigse semantilise abstraktsiooni tasemega. Lisaks sellele on mudelil parameetrite rohkuse tõttu väga suur mälujalg.

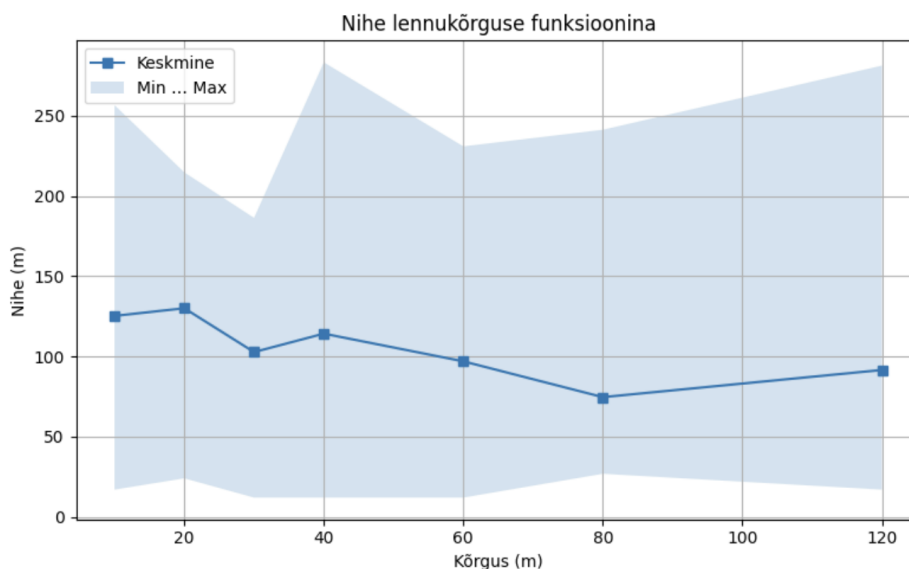
Kokkuvõttes pakkusid DINOv2 eri mudelid antud probleemi silmas pidades häid tulemusi ning ületasid teiste mudelite poolt saadud tulemusi. Seetõttu valiti DINOv2 antud töö kontekstis parimaks mudeliks ning loodi seda rakendades ka kaks eksperimenti.

### 3.4 Eri kõrguste mõju analüüs

Tehtud katsete põhjal ilmnes, et positsioneerimise viga ei kahane monotoonselt koos kõrguse vähenemisega, vaid moodustab pigem U-kujulise kõvera. Tulemused on nähtavad Tabel 2 ning lisaks visualiseeritud joonisel Joonis 18. Parim täpsus saavutati 80 m kõrgusel (vaata Joonis 19), kus keskmine nihe oli vaid 74.6 m (mediaan 60.6 m). Ka 120 m ja 60 m juures püsis viga suhteliselt madalal, vastavalt 91.5 m ja 96.9 m. Miinimumviga jäi kõigil kolmel tasandil 12-27 m vahemikku.

Tabel 2. Eri kõrguste mõju saadud tulemustele.

Kõrgus (m)	Keskmine nihe (m)	Mediaan nihe (m)	Minimaalne nihe (m)	Maksimaalne nihe (m)	Standardhälve
120	91.534464	63.348167	16.975594	281.252350	73.555521
80	74.603786	60.612082	26.840771	241.2683044	57.906488
60	96.917063	67.316653	12.003558	230.893044	70.412987
40	114.183159	80.334800	12.003558	283.294136	91.190254
30	102.690438	105.725975	12.003558	186.345328	59.064302
20	130.022074	155.990883	24.007116	214.726170	66.474802
10	125.244154	131.704668	16.975594	256.606757	80.730800



Joonis 18. Tulemuste nihe reaalsest asukohast sõltuvalt lennukõrguse muutusest.



Joonis 19. Näide drooni simuleeritud kõrguse mõjust saadud tulemusele.

Alla 60 m laskudes hakkas täpsus aga taas halvenema. Kõrgusel 40 m tõusis keskmine viga üle 114 m ning standardhälve kasvas 90 m-ni, mis näitab, et tulemused muutuvad ebahütlasemaks. 20 m ja 10 m puhul suurenes keskmine nihe vastavalt 130 m ja 125 m-ni, kuigi minimaalne viga jäi endiselt alla 25 m. See viitab, et väga madalal lennates võib süsteem mõnes kaadris endiselt hea vaste leida, kuid üldine töökindlus väheneb.

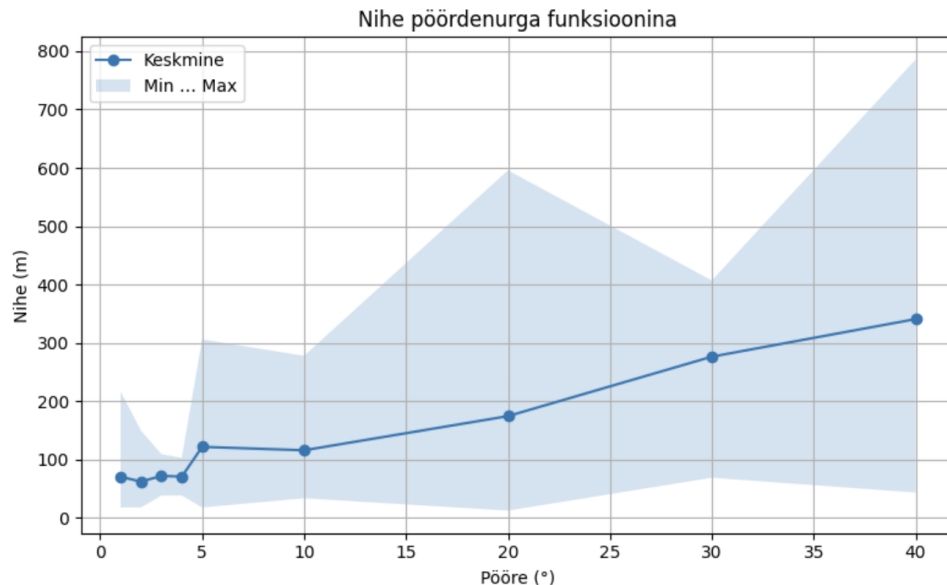
Kokkuvõttes pakub 80 m kõrgusel lendamine kompromissi väikese keskmise vea ja mõõduka dispersiooni vahel (standardhälve = 58 m), samas kui kõrgused alla 40 m toovad kaasa suurema hajuvuse ja kasvava maksimum nihke (kuni 283 m).

### 3.5 Eri pöördenurkade mõju analüüs

Ekspereiment kinnitas, et droonikaadri pöördenurk satelliidifoto suhtes määrab otseselt, kui hästi CNN-põhine mallisobitus drooni asukoha leiab. Tabel 3 ning jooniselJoonis 20 on väljatoodud eksperimendi tulemused. Kõige stabiilsemad tulemused saadi, kui pöördenurk jäi nelja kraadi sisse, mille juures püsis keskmine nihe 62-72 meetri vahemikus ja andmete hajuvus oli mõõdukas. Eriti hästi töötas süsteem 2° juures, kus keskmine viga oli umbes 62 m ning standardhälve 38 m. Ka minimaalsed vead olid seal väikesed (alla 20 m), mis näitab, et väike pöörde kõikumine ei sega tunnuste sobitamist.

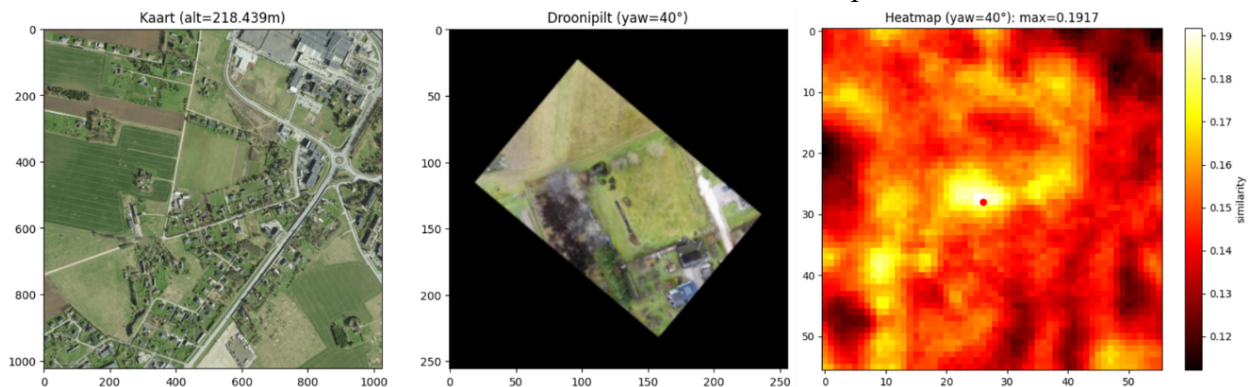
Tabel 3. Eri pöördenurkade mõju saadud tulemustele.

Pööre (°)	Keskmine nihe (m)	Mediaan nihe (m)	Minimaalne nihe (m)	Maksimaalne nihe (m)	Standardhälve
1	70.645838	51.509367	18.187652	216.383821	57.668786
2	62.355494	62.677438	18.187652	149.569232	38.264103
3	72.155993	66.742021	38.631650	109.293969	21.648371
4	70.907852	69.176736	38.631650	103.131526	18.824607
5	121.933212	89.606681	18.187652	306.012935	94.556281
10	116.064935	83.243272	33.908250	277.771430	71.852710
20	174.579895	111.150946	12.791891	595.606777	167.560135
30	276.438204	303.900678	69.096957	406.801977	110.950982
40	340.867343	294.191870	43.700753	785.656247	203.223162



Joonis 20. Tulemuste nihe reaalsest asukohast sõltuvalt drooni pöördenurgast.

Alates viiest kraadist muutus pilt kiiresti. 5° ja 10° nurkade korral kasvas keskmine positsioneerimisviga juba sajast meetrist ülespoole ning mõõtmiste hajuvus kahekordistus (vaata Joonis 21). See tähendab, et osa kaadreid sobitus endiselt talutavalt, ent teised eksisid sadades meetrites. Kui pöördenurk jõudis kahekümne kraadini, libises keskmine nihe 175 m peale ja üksikutes proovides tuli ette kuni 596-meetriseid vigu. Dispersioon oli sel tasemel niivõrd suur, et positsioneerimine ei ole enam töökindel. Kolmekümne ja neljakümne kraadi juures süvenes probleem veelgi, keskmised vead tõusid vastavalt ligikaudu 276 m-ni ja 341 m-ni, maksimaalne viga küündis 786 meetriteni ning standardhälbed ületasid 200 m piiri. Selliste nurkade puhul ei suuda mudel enam usaldusväärset tuvastada, milline satelliidifoto piirkond vastab droonikaadrile.



Joonis 21. Näide drooni simuleeritud pöördenurga mõjust saadud tulemusele.

Kokkuvõttes on antud eksperimendi tulemusena süsteem usaldusväärne kuni neljakraadiste pööranguteni. Suuremate nurkade puhul võivad vead kasvavad eksponentsiaalselt.

## 4. Arutelu

Antud peatükis arutletakse välja pakutud lahenduse tugevuste ja nõrkuste üle. Samuti tuuakse välja töö peamised piirangud ja pakutakse välja edasisi arendusi.

### 4.1 Lahenduse piirangud

#### Mudelite treeningu ja üldistamise piirangud

Kasutatud mudelid (sh DINOv2 erinevad versioonid, ResNet50, YOLOv8) on treenitud üldiste pildikogumite (nt ImageNet) või laiemate visuaalsete andmekogumite peal, mitte spetsiifiliselt aerofotode peal. See tähendab, et mudelitel võib puududa tundlikkus lokaalse maastiku või spetsiifiliste objektide suhtes (nt metsamaastikud, põllud, eripärased hoonestusstruktuurid).

Droonipildid võivad erineda märgatavalt maakaardipõhistest ortofotodest. Vaatenurk, kõrgus, kaamera kaldenurk ja valgusolud võivad muuta ühe ja sama ala visuaalselt erinevaks, põhjustades tunnuste mitesobivust ja CS meetodil arvatatud kaardil valesid tulemusi.

#### Tunnuskaardi eraldusvõime ja informatsioonikadu

Konvolutsioonivõrgud (nt ResNet50, YOLOv8) vähendavad järk-järgult pildi ruumilist eraldusvõimet, et genereerida kõrgema taseme tunnuseid. See viib väikese eraldusvõimega tunnuskaartideni, mis võivad kaotada väikseid, kuid olulisi detaile, eriti droonifotode puhul, kus visuaalsed erinevused võivad olla peened.

Antud töö koodi töövoos võeti tunnuskaardid ainult ühest kihist (nt YOLOv8 puhul `layer_index = 12`), jättes kasutamata võimaluse kombineerida mitme taseme tunnuseid (nt madala ja kõrge taseme tunnused). See piirab mudeli võimet haarata nii globaalset kui lokaalsemat konteksti samaaegselt.

#### Visuaalse sarnasuse sõltuvus

Kui maastikul puuduvad unikaalsed visuaalsed tunnused (nt ühtlased põllud, metsad), langeb sarnasuskaardi kontrastsus ja CS ei leia selget maksimumi. Sellistel juhtudel on isegi inimvaatlejal keeruline leida pildilt täpset vastavust. Lisaks võib valgus- ja varjuolude erinevus kaardipildi ja droonipildi vahel (nt hommiku vs õhtupäike, pilvisus) põhjustada tunnuste lahknevust, mida CNN-il põhinev mudel ei kompenseeri. Üheks lahenduseks oleks osakestefiltri (particle filter) kasutamine, kus süsteem säilitab iga sammuga drooni asukoha tõenäosusjaotuse eelmisest kaadrist ning uuendab seda iga uue kaadriga. Peale mitut iteratsiooni koonduvad tõenäosused ühte selgesse piirkonda, mis vastab tegelikule drooni positsioonile ja aitab parandada sellega geolokaliseerimist.

### **Mitme erineva tehnoloogia integreerimine**

Magistritöö käigus tuli integreerida mitmeid erinevaid teke ja raamistikke (nt PyTorch, torchvision, ultralytics YOLO, huggingface, geopandas, matplotlib, requests, PIL, pyproj). See tähendas erinevate API-de õppimist ja dokumentatsioonide läbitöötamist, mis võttis märkimisväärselt aega, eriti spetsiifiliste funktsioonide (nt YOLO vahekihist tunnuste püüdmine) juures.

### **Ressursside piirangud (riistvara)**

Suurte mudelite (DINOv2 *large*, YOLOv8) jooksutamine nõuab GPU olemasolu ning piisavat VRAM mahtu.

## **4.2 Ideed edasiarendusteks**

### **Mudeli täiustamine ja alternatiivide testimine**

Katsetada teisi sügavate närvivõrkude mudelid, mis võiksid paremini sobituda droonifotode ja satelliidipiltide sarnaste omaduste leidmiseks. Näiteks: D-FINE, ConvNeXt, Swin Transformer. Seejärel võrrelda tulemusi olemasoleva WALDO30, ResNet50 ja DINO mudelitega, et leida parima täpsuse ja kiiruse kompromiss. Kuna antud magistritööl on mahulised piirangud, siis ei mahtunud muude mudelite testimine tööraamistiku.

### **Mitmekihilise tunnuste eralduse lisamine**

Praegu kasutatakse vaid ühte vahekihti (YOLO 12. kiht). Edaspidi võiks integreerida mitme erineva kihi tunnuseid, mis võimaldaks süsteemil kasutada nii madalama taseme kui ka kõrgema taseme tunnuseid. See võiks viia täpsemate tulemusteni, võttes arvesse seda, et WALDO30 ei pakkunud sugugi täpset asukohatuvastust.

### **Mitme pildikaadri ja trajektoori liitmine**

Laiendada süsteemi nii, et see suudaks töödelda mitut järjestikust droonipilti korraga ja hinnata nende omavahelist asendit (ehk lennutrajektoori rekonstrueerimine). Võimaldaks paremat ajas püsivat positsioneerimist, kasutades liikumise konteksti (*motion model*).

### **Rahvusvahelise koordinaatsüsteemi kasutamine**

Kuigi antud prototüübis kasutati L-EST97 koordinaatsüsteemi, siis võiks edasiarendustest kasutada universaalsemat ning rahvusvahelisemat koordinaatsüsteemi UTM. Selle abil saaks lahenduse rakendamist laiendada ka muudele riikidele peale Eesti ning muudaks selle oluliselt paindlikumaks.

### **Konkreetsle objektile tuvastuse kitsendamine**

Väljapakutud lahendust saaks kitsendada veel ülesandespetsiifilisemaks ning kasutada näiteks raudtee taristu tuvastamises. Tegemist oleks militaarvaldkonna kontekstis olulise arendusega. Kuna raudteerööpad on aastaajast sõltuvalt nähtavana püsiv asimuut võimaldaks see magistritöös väljapakutud lahendusel nihkest hoolimata pidevalt veel suurema kindlusega absoluutset asukohta tuvastada. Lisaks oleks kuvatud trajektoori õigsust oluliselt lihtsam visuaalselt kontrollida.

# Kokkuvõte

Käesolevas magistritöös uuriti droonide absoluutset positsioneerimist aerofotode ja sügavõppe meetodite abil, keskendudes konvolutsiooniliste närvivõrkude rakendamisele visuaalse sobitamise kaudu. Töös rakendati ja katsetati mitmeid mudelipõhiseid lähenemisi, sealhulgas ResNet50, DINOv2 (*base, small, large*) ja WALDO30, eesmärgiga hinnata nende täpsust erinevates tingimustes. Lisaks mudelite võrdlusele viidi läbi eraldi eksperimendid, et analüüsida drooni lennukõrguse ja pöördenurga mõju positsioneerimise täpsusele.

Katsed näitasid selgelt, et mudeli valik ja lennuolud mõjutavad GPS-vaba positsioneerimise täpsust. Täpsuselt parimaks osutus DINOv2-*large* (keskmine nihe  $\approx 46$  m ning standardhälve  $\approx 55$  m), kuid see oli ühtlasi kõige aeglasem ( $\approx 10$  s / kaader). DINOv2-*base* pakkus veidi suuremat viga ( $\approx 51$  m) ja poole väiksemat arvutus-kulu ( $\approx 3.9$  s), samal ajal kui DINOv2-*small* oli kaks korda kiirem ( $\approx 2.0$  s), kuid täpsus langes keskmiselt 100 m-ni. Klassikaline ResNet50 oli veelgi kiirem ( $\approx 2.3$  s), ent 165-meetrine keskmine nihe piirab selle praktilist kasutust. WALDO30 jäi mõlemas mõõtmes kõige nõrgemaks ( $\approx 364$  m ja  $\approx 1.9$  s). Sellegipoolest oli kõigi DINOv2 mudelite keskmine nihe piisavalt väike, et reaalses elus lahendus absoluutse asukoha tuvastamisel toimiks.

Töös tehtud eksperimentide käigus selgus, et lennu-kõrgus  $\sim 80$  m annab parima keskmise nihke ( $\approx 75$  m); madalamale laskumine ei toonud täiendavat võitu ning väga madalalt ( $\leq 20$  m) täpsus hoopis halvenes. Kaamera pöördenurk mõjutab tulemust veelgi enam. Pöördenurkade  $1 - 4^\circ$  erinevused jäid  $\sim 70$  m täpsuse juurde, kuid alates  $20^\circ$  kasvab viga üle 170 m ning  $40^\circ$  juures üle 340 m.

Seega sobib DINOv2-*base* kõige paremini tasakaalustatud süsteemi ehitamiseks. Antud mudel saavutab  $< 60$  m täpsuse mõistliku arvutus-kuluga ning püsib stabiilsena kuni  $10^\circ$  pöördeni ja 80 m kõrguseni.

# Tunnustus

Avaldan siinkohal tänu oma juhendajale Tambet Matiisenile meeldiva koostöö eest. Samuti olen juhendajale tänulik andmete hankimise eest, mille tõttu oli võimalik esimesest päevast peale esmaseid tekkinud ideid kiirelt katsetada ning selletõttu magistritöö tegemise aega optimeerida.

# Viidatud kirjandus

1. Dai, M., Zheng, E., Chen, J., Qi, L., Feng, Z., & Yang, W. Drone referring localization: An efficient heterogeneous spatial feature interaction method for UAV self-localization. arXiv, 2024. <https://arxiv.org/abs/2208.06561>
2. Seema, K. “Balti välisministrid hoiatavad: Tartu lennujaam on venelaste hübriidrünnaku all”. Tartu Postimees, 2024: <https://tartu.postimees.ee/8010265/balti-valisministrid-hoiatavad-tartu-lennujaam-on-venelaste-hubriidrunnaku-all> (16.12.2024)
3. Franke, Ulrike. “Drones in Ukraine: Four lessons for the West”. European Council on Foreign Relations, 2025: <https://ecfr.eu/article/drones-in-ukraine-four-lessons-for-the-west/> (20.04.2025)
4. Zafra, Mariano; Hunder, Max; Rao, Anurag; Kiyada, Sudev. “How drone combat in Ukraine is changing warfare”. Reuters, 2024: <https://www.reuters.com/graphics/UKRAINE-CRISIS/DRONES/dwpkeyjwkpm/> (20.04.2025)
5. Lee, Roy. What are the Parts of FPV Drone?: MepsKing, 2025: <https://www.mepsking.shop/blog/what-are-the-parts-of-fpv-drone.html> (15.04.2025)
6. How Things Fly. Roll, Pitch, and Yaw. <https://howthingsfly.si.edu/flight-dynamics/roll-pitch-and-yaw> (15.04.2025)
7. Gurucharan, MK. Basic CNN Architecture: A Detailed Explanation of the 5 Layers in Convolutional Neural Networks: upGrad, 2025. <https://www.upgrad.com/blog/basic-cnn-architecture/> (20.04.2025)
8. Analytixlabs. Convolutional Neural Networks – Definition, Architecture, Types, Applications, and more. 2024. <https://www.analytixlabs.co.in/blog/convolutional-neural-network/> (20.04.2025)
9. Medium. 2022. Convolutional Neural Networks (CNN) — Architecture Explained. <https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243> (21.04.2025)
10. Milvus. What is a fully connected layer?. <https://blog.milvus.io/ai-quick-reference/what-is-a-fully-connected-layer> (21.04.2025)

11. Dive Into Deep Learning. 14.11. Fully Convolutional Networks. [https://www.d2l.ai/chapter\\_computer-vision/fcn.html](https://www.d2l.ai/chapter_computer-vision/fcn.html) (12.05.2025)
12. He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian. Deep Residual Learning for Image Recognition. Microsoft Research. 2015. <https://arxiv.org/pdf/1512.03385>
13. Ibrahim, Mostafa. The Basics of Resnet50. Weights & Biases, 2025. <https://wandb.ai/mostafaibrahim17/ml-articles/reports/The-Basics-of-ResNet50---Vmlldzo2NDkwNDE2> (19.04.2025)
14. Mukherjee, Suvaditya. The Annotated ResNet-50: Medium, 2022. <https://medium.com/data-science/the-annotated-resnet-50-a6c536034758> (12.05.2025)
15. Brownlee, Jason. A Gentle Introduction to the Rectified Linear Unit (ReLU): *Machine Learning Mastery* 2020. <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/> (19.04.2025)
16. Rehman, Yawar. How the bottleneck layers in the deep networks work and how do those layers reduce computational complexity: *Medium*, 2024. <https://medium.com/@reh.yawar2/how-the-bottleneck-layers-in-the-deep-networks-work-and-how-do-those-layers-reduce-computational-7bc99c0d1e96> (19.04.2025)
17. Potrimba, Petru. What is ResNet-50?: *Roboflow*, 2024. <https://blog.roboflow.com/what-is-resnet-50/> (19.04.2025)
18. Dutta, Bhumika. What are Skip Connections in Neural Networks?: *analyticssteps*, 2021. <https://www.analyticssteps.com/blogs/what-are-skip-connections-neural-networks> (19.04.2025)
19. Oquab, Maxime; Darcet, Timothée; Moutakanni, Théo; Vo, Huy V.; Szafraniec, Marc; Khalidov, Vasil; Fernandez, Pierre; Haziza, Daniel; Massa, Francisco; El-Nouby, Alaaeldin; Assran, Mahmoud; Ballas, Nicolas; Galuba, Wojciech; Howes, Russell; Huang, Po-Yao; Li, Shang-Wen; Misra, Ishan; Rabbat, Michael; Sharma, Vasu; Synnaeve, Gabriel; Xu, Hu; Jegou, Hervé; Mairal, Julien; Labatut, Patrick; Joulin, Armand; Bojanowski, Piotr. DINOv2: Learning Robust Visual Features without Supervision. Meta AI Research, 2024. <https://arxiv.org/pdf/2304.07193>

- 20.Mishra, Malyaj. DINOv2: A Complete Guide to Self-Supervised Learning and Vision Transformers: *Medium*, 2024. <https://medium.com/data-science-in-your-pocket/dinov2-a-complete-guide-to-self-supervised-learning-and-vision-transformers-d5c1fb75d93f> (20.04.2025)
- 21.Meta. DINOv2: State-of-the-art computer vision models with self-supervised learning. 2023. <https://ai.meta.com/blog/dino-v2-computer-vision-self-supervised-learning/> (20.04.2025)
- 22.Dataloop. WALDO30. 2024. [https://dataloop.ai/library/model/stephanst\\_waldo30/](https://dataloop.ai/library/model/stephanst_waldo30/) (15.04.2025)
- 23.Torres, Jane. YOLOv8 Architecture: A Deep Dive into its Architecture: YOLOv8, 2024. <https://yolov8.org/yolov8-architecture/> (15.04.2025)
- 24.Adaptive Vision. Template Matching. [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) (21.04.2025)
- 25.Timothy Malche. What is Template Matching? An Introduction: *Roboflow*, 2024. <https://blog.roboflow.com/template-matching/> (23.04.2025)
- 26.OpenAI (2025). ChatGPT o3: <https://chat.openai.com>.
- 27.Datastax. What is Cosine Similarity: A Comprehensive Guide. 2025. <https://www.datastax.com/guides/what-is-cosine-similarity> (23.04.2025)
- 28.Wu, Rouwan; Cheng, Xiaoya; Zhu, Juelin; Liu, Xuxiang; Zhang, Maojun; Yan, Shen. UAVD4L: A Large-Scale Dataset for UAV 6-DoF Localization. ChangSha, China: National University of Defence Technology, 2024, <https://arxiv.org/pdf/2401.05971>.
- 29.Cabrera-Ponce, A. Alrich; Martinez-Carranza, Jose. Convolutional Neural Networks for Geo-Localisation with a Single Aerial Image. *Journal of Real-Time Image Processing*, 2024, 19(3), 565–575. <https://doi.org/10.1007/s11554-022-01207-1>
- 30.Hugging Face. Vision Transformer (base-sized model) trained using DINOv2: AI at Meta, 2024. <https://huggingface.co/facebook/dinov2-base> (20.04.2025)
- 31.Hugging Face. Vision Transformer (small-sized model) trained using DINOv2: AI at Meta, 2024. <https://huggingface.co/facebook/dinov2-small> (20.04.2025)

32.Hugging Face. Vision Transformer (large-sized model) trained using DINOv2: AI at Meta, 2024. <https://huggingface.co/facebook/dinov2-large> (20.04.2025)

33.Ćwiakała, Paweł. Testing Procedure of Unmanned Aerial Vehicles (UAVs) Trajectory in Automatic Missions: AGH University of Krakow. 2019.  
[https://www.researchgate.net/publication/335399284\\_Testing\\_Procedure\\_of\\_Unmanned\\_Aerial\\_Vehicles\\_UAVs\\_Trajectory\\_in\\_Automatic\\_Missions](https://www.researchgate.net/publication/335399284_Testing_Procedure_of_Unmanned_Aerial_Vehicles_UAVs_Trajectory_in_Automatic_Missions)

# Lisad

## Lisa I: Eeltreenitud Resnet50 mudeli rakendamine

ResNet50 mudeli valiku taga oli selle sügav, kuid hästi tasakaalustatud konvolutsioonivõrk. Selle lühisühendused (*skip-connections*) aitavad hoida koos nii lihtsad kui ka keerukad tunnused, mistõttu sobib mudel hästi maastiku ja hoonete mustrite leidmiseks.

Põhi-ehitus:

- alguses üks suur 7x7 konvolutsioonifilter (pildi kiireks vähendamiseks).
- sellele järgneb 3x3 maksimum väärtuse leidmine, mis tõstab tähtsama info esile.
- edasi neli plokki järjest, kus iga plokk sisaldab mitut jääkplokki [26]:
  - 3 plokki 256 kanaliga
  - 4 plokki 512 kanaliga
  - 6 plokki 1024 kanaliga
  - 3 plokki 2048 kanaliga

Töös huvitati ainult viimasest konvolutsioonikihist. Eemaldati mudeli lõpetuseks mõeldud täissidusa kihi, et ResNet-50 toimiks puhta tunnuse-väljastajana. Nii saadi iga pildi kohta üks kompaktne tunnускаart mõõtmega 1x2048x7x7, mida kasutati edasiseks võrdlemiseks ja soojuskaardi loomiseks.

Piltide ette-valmistus enne mudelisse saatmist

1. Suuruse muutmine
  - droonifoto: 224x224 px (sobib ResNet-ile)
  - ortofoto löige: 1024x1024 px (säilitab vajalikud detailid)
2. Tensoriks muutmine - pilt teisendatakse formaati, mida PyTorch mõistab.
3. Värvide ühtlustamine - rakendatakse ImageNeti keskmisi ja hajuvusi
  - keskmine = [0.485, 0.456, 0.406]
  - standardhälve = [0.229, 0.224, 0.225]

Iga pilt läbib need sammud eraldi ja suunatakse seejärel ResNet-50-le, mis tagastab ülal mainitud tunnускаardi. Sellest kaardist arvutatakse hiljem drooni oletatava asukoha kaardil.

## Lisa II: Eeltreenitud DINOv2 mudeli rakendamine

DINOv2 mudel valiti, sest see on isejuhenduv visuaalne transformer. See suudab õppida nii üldisi kui ka kohalikke mustreid ilma märgendatud andmeteta. Seetõttu sobib mudel hästi olukordadesse, kus droonipilte on vähe ja maastikud on eriilmelised.

Kõik kolm versiooni (*small, base, large*) kasutavad täpselt sama arhitektuuri loogikat, andmetöötlust ja treening-skeemi. Antud mudelid erinevad vaid sisemiste mõõtmete poolest (peidetud kanaleid, peade arvu ja plokkide hulka on eri suurustega mudelites rohkem või vähem). Seega saab sama töövoogu rakendada kõigi kolme variandi puhul.

Põhiehitus:

- pilt jagatakse 14x14 px plokkideks, millest tehakse 768-kanaliga sümboliteks (*token*) [26].
- läbi käiakse 12 Transformer-plokki, kus liigub kogu pildi info [26].
- Klassifikatsiooni sümbol hoiab kokkuvõtlikku infot, kuid ruumilise kaardi jaoks kasutatakse ainult teisi plokkide [26].
- plokkide lõppu lisatakse klassi-normeerimine, mis hoiab gradienti stabiilsena ja parandab mudeli ülekantavust [26].

Töös võeti viimane peidetud olek enne väljundkihti. Klassifikatsiooni sümbol eemaldati ning ülejäänud sümbol kujundati tagasi  $14 \times 14$  ruudustikuks. Saadud tunnuskaart mõõtmega  $1 \times 768 \times 14 \times 14$  töötab nagu CNN-i tunnuskaart ja sobib slaidivaks konvolutsiooniks, et leida kaardil piirkond, millel on droonipildiga suurim koosinus sarnasus.

Piltide ettevalmistus enne mudelisse saatmist on sama mis Resnet50 ning WALDO30 mudeli puhul.

### Lisa III: Eeltreenitud WALDO30 mudeli rakendamine

WALDO30 mudel valiti, sest see põhineb YOLOv8 selgroot ja on eeltreenitud maastiku- ning satelliidifotodel.

- **Põhiehitus:**
- Sisend skaleeritakse 640x640 px-ni ja läbib CSPDarknet-selgroot. See loob kolmel konvolutsioonilisel sammul (8/16/32) tunnuskaardid: 80x80, 40x40 ja 20x20 px [26].
- Igas selgroot etapis on C2f-plokk, mis kasutab jääkühendusi ja osalisi kopeerimisi, et gradient liiguks sügavale ilma info kadumiseta [26].
- Kael (PAN-laadne C2f-võrk) liidab madala ja kõrge resolutsiooniga tunnused, parandades väikeste objektide nähtavust [26].
- Peas on kolm tuvastusoksast, millest võetakse 12. ploki väljund (40x40x192) tunnuskaardina [26];
- Plokkide lõpus on SiLU-aktivatsioon ja BatchNorm, mis hoiavad gradienti stabiilsena [30].

Lisaks kasutatakse valitud vahekihi väljundit (*forward-hook*). Kanaleid on 192 ja ruumiline maatriks 40x40. Selline tunnuskaart (1x192x40x40) töötab nagu CNN-i omaduskaart. Sellega saab teha slaidiva conv2d-võrdluse, et leida kaardilt ala, mille koosinus sarnasus on droonipildiga kõige suurem.

Piltide ettevalmistus enne mudelisse saatmist on sama mis DINOv2 mudelite ning WALDO30 mudeli puhul.

# Litsents

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

Mina, Gregor Nepste ,  
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Absoluutne drooni positsioneerimine kasutades aerofotosid  
(*lõputöö pealkiri*)

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

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;

2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commons 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.

Gregor Nepste  
15.05.2025