



Norges miljø- og
biovitenskapelige
universitet

Masteroppgave 2024 30 stp
Fakultet for realfag og teknologi

Fotogrammetrisk modellering av steinoverflater i sideareal til veg fra 360-graders kamera: En undersøkelse av metode for detektering av steinsprang

Photogrammetric modelling of rock surfaces in
roadside areas using 360-degree cameras - A
survey on methods for detecting rockfalls

Isak Rakstang
Geomatikk

Forord

Denne masteroppgaven snurper sammen ranselen en siste gang og markerer slutten på fem gode år på Norges Miljø- og Biovitenskapelige universitet.

Først vil en takk gå til biveileder Jens Otto Dolva fra Nye Veier for utgangspunktet til oppgaven og bidrag som bindeledd fra meg til andre kontakter i oppgaven. Videre takk til alle som har bidratt med informasjon og veiledning til hjelp i oppgaven; Gunhild Mæhlum (Field), Univrses teamet, Mosaic teamet, Anette Wold Magnussen (Nye veier). En spesiell takk til hovedveileder Ivar Maalen-Johansen, bærebjelken til Geomatikk på NMBU. Takk for all hjelp på masteroppgaven, men enda viktigere den tilstedeværelsen du har vist til alle oss opp igjennom åra.

Så må det takkes hele hurven av en gjeng som har gjort åra på Ås til de fantastisk gode åra de har vært. Takker med hele mitt hjerte Mannskoret Over Rævne, UKEstyret 2022, geomatikkrommet, og lunsjgjengen fra vælas navle; Studentenes hus.

NMBU, 15.mai 2024

Isak Rakstang

Sammendrag

Denne masteroppgaven er skrevet for Nye Veier og utforsker en ny metode for registrering av steinspranghendelser, som kan redusere behovet for manuelle inspeksjoner og potensielt øke sikkerheten langs norske veier. Oppgaven undersøker anvendelsen av fotogrammetrisk modellering basert på 360-graders bilder, kombinert med maskinlæringsteknikker, for identifisering og klassifisering av steinsprang i sidearealer langs veien. Den presenterer en detaljert metodikk for behandling av 360-graders bildedata til fotogrammetri og utviklingen av en maskinlæringsmodell tilpasset klassifisering av steinoverflater. Oppgaven vurderer om metoden kan være en kostnadseffektiv og pålitelig tilnærming til detektering av steinsprang. Gjennom sammenligning av modellerte steinoverflater med laserdata, ble modellens nøyaktighet vurdert som utilstrekkelig basert på tilgjengelig grunnlagsdata. Det ble gjennomført grundig feilsøking for å finne årsaken til den unøyaktige modellen. En plausibel teori er at økt bildefrekvens ville ha betydelig innvirkning på modellens nøyaktighet. Oppgaven foreslår videre forskning for å forbedre og validere modellen, med mål om implementering i reelle veimiljøer.

Abstract

This master's thesis, written for Nye Veier, explores a new method for registering rockfall incidents, potentially reducing the need for manual inspections and increasing safety along Norwegian roads. The thesis investigates the application of photogrammetric modeling based on 360-degree images, combined with machine learning techniques, for identifying and classifying rockfalls in roadside areas. It presents a detailed methodology for processing 360-degree image data into photogrammetry and the development of a machine learning model tailored for classifying rock surfaces. The study assesses whether the method can be a cost-effective and reliable approach to detecting rockfalls. By comparing modeled rock surfaces with laser data, the model's accuracy was deemed insufficient based on the available baseline data. Extensive troubleshooting was conducted to identify the cause of the model's inaccuracy. One plausible theory is that increasing the image frequency would significantly impact the model's accuracy. The thesis suggests further research to improve and validate the model, with the goal of implementation in real road environments.

Innholdsfortegnelse

| | | |
|----------|---|-----------|
| 1 | Introduksjon | 1 |
| 1.1 | Bakgrunn | 1 |
| 1.1.1 | Definisjoner | 2 |
| 1.1.2 | Risikovurdering av steinsprangområder | 2 |
| 1.1.3 | Registrering av hendelser og vedlikehold av sidearealer | 4 |
| 1.1.4 | Problemstillingens utgangspunkt | 4 |
| 1.1.5 | Tidligere forskning og anvendelser | 4 |
| 1.2 | Problemstilling | 6 |
| 1.3 | Beskrivelse av studieområde | 7 |
| 2 | Teori | 10 |
| 2.1 | 360-graders kamera | 10 |
| 2.2 | Fotogrammetri | 11 |
| 2.2.1 | Bevegelsesparalakse | 11 |
| 2.2.2 | Egenskapsmatching | 12 |
| 2.3 | Klassifiseringsteknikker | 13 |
| 2.3.1 | Maskinlæring - En introduksjon | 13 |
| 2.3.2 | Instance-segmentering | 13 |
| 2.3.3 | YOLOv8: Anvendelse og prinsipper | 14 |
| 2.3.4 | Validering av klassifiseringsteknikker | 14 |
| 3 | Metode | 16 |
| 3.1 | Modellering via fotogrammetri | 16 |
| 3.1.1 | Utstyr | 16 |
| 3.1.2 | Prosessflyt for fotogrammetri | 18 |
| 3.1.3 | Programvare | 18 |

| | | |
|----------|---|-----------|
| 3.1.4 | Integrering av Kunstig Intelligens i forskningsmetodikk | 19 |
| 3.1.5 | Datainnsamling og -behandling | 20 |
| 3.1.6 | Overflate og tekstur-modellering | 21 |
| 3.1.7 | Sammenligningsmetode for modellen | 26 |
| 3.2 | Utvikling og validering av klassifiseringsmodell | 27 |
| 3.2.1 | Bildeannotering | 27 |
| 3.2.2 | Modelltreningsprosedyre | 27 |
| 4 | Resultater | 30 |
| 4.1 | Evaluering av fotogrammetrisk modell | 30 |
| 4.1.1 | Forbehandling av data | 30 |
| 4.1.2 | Modellgenerering og -analyse | 34 |
| 4.2 | Evaluering av klassifiseringsmodell | 42 |
| 5 | Diskusjon | 47 |
| 5.1 | Vurdering av fotogrammetriske teknikker | 47 |
| 5.2 | Effektivitet av klassifiseringsmetoder | 50 |
| 5.3 | Forslag til videre forskning | 52 |
| 6 | Konklusjon | 53 |
| 7 | Vedlegg | 59 |
| 7.0.1 | Kode for strukturering av GPX-fil til innkoding av EXIF-data til bildene | 59 |
| 7.0.2 | Rapport Agisoft Metashape | 61 |

Figuroversikt

| | | |
|------|--|----|
| 1.1 | Utredningsprosess steinsprangfare | 3 |
| 1.2 | Oversiktskart over studieområdet | 8 |
| 1.3 | Detaljert kart over studieområdet | 8 |
| 1.4 | Kart over skredpunkt i studieområdet | 8 |
| 1.5 | Statistikk over størrelse og type skred i studieområdet | 9 |
| 2.1 | Bevegelsesparallakse | 12 |
| 2.2 | Segmenteringsmetoder innen bildeprosessering | 13 |
| 3.1 | Illustrasjonsbilde Mosaic X | 17 |
| 3.2 | Flytdiagram over prosess for fotogrammetri | 18 |
| 3.3 | Bilde av rigg med 360-graders kamera og laserskanner | 20 |
| 3.4 | Eksempel på sammensydd 360-graders bilde | 21 |
| 3.5 | Eksempelbilder fra hvert kamera i kamerasystemet | 22 |
| 3.6 | Oversikt over kameraposisjoner brukt i overflatemodell | 23 |
| 3.7 | Råbilder og tilhørende masker | 24 |
| 3.8 | Målt avstand brukt til skalering av modell | 25 |
| 3.9 | Markerte punkter brukt til skalering av modell | 25 |
| 3.10 | Oversiktsbilde over tre utvalgte segmenter | 27 |
| 3.11 | Annotert bilde fra TrainYOLO | 27 |
| 3.12 | Diagram over senterpunkt i klassifiseringene | 29 |
| 3.13 | Kodesnutt fra TrainYOLO | 29 |
| 4.1 | Kameraposisjoner etter orientering | 31 |
| 4.2 | Kameraposisjoner med tilhørende feilellipser | 31 |
| 4.3 | Bildeorienteringer til kamerasystemet ved fire kameraposisjoner. | 31 |
| 4.4 | Visuell framstilling av overlapp i kamerasystemet | 32 |

| | | |
|------|--|----|
| 4.5 | Visell framstilling av objekter på tvers av kameraposisjon | 34 |
| 4.6 | Overflatemodell utsnitt 1 | 35 |
| 4.7 | Overflatemodell utsnitt 2 | 36 |
| 4.8 | Overflatemodell utsnitt 3 | 36 |
| 4.9 | Oversikt over segmenter til detaljert analyse | 37 |
| 4.10 | Punktfordeling i punktsky 1 | 37 |
| 4.11 | Punktfordeling i punktsky 2 | 37 |
| 4.12 | Punktfordeling i punktsky 3 | 38 |
| 4.13 | Konfidenskart segment 1 | 38 |
| 4.14 | Konfidenskart segment 2 | 39 |
| 4.15 | Konfidenskart segment 3 | 39 |
| 4.16 | Segmentvis avstand mellom overflatemodell og punktsky | 41 |
| 4.17 | mAP-graf for boks- og maskeklassifisering | 43 |
| 4.18 | Graf over loss i maskinlæringsmodell | 43 |
| 4.19 | Eksempel på riktig klassifisering av maskinlæringsmodellen | 44 |
| 4.20 | Eksempel på feilklassifisering av maskinlæringsmodellen | 45 |
| 4.21 | F1 - Confidence graf | 45 |
| 4.22 | Recall - Confidence graf | 46 |

Tabelloversikt

| | | |
|-----|--|----|
| 3.1 | Spesifikasjoner Mosaic X | 17 |
| 3.2 | Spesifikasjoner Velodyne Puck | 17 |
| 3.3 | Systeminformasjon PC til prosessering | 18 |
| 3.4 | Konfigurasjonsparametere maskinlæringsmodell | 28 |
| 4.1 | Bildekvalitet for ulike kameraposisjoner | 30 |
| 4.2 | Sammenbindingspunkt mellom enkeltkameraer i kamerasystemet | 33 |
| 4.3 | Sammenbindingspunkt i tre kameraposisjoner | 34 |
| 4.4 | Prosesseringstid for modelleringen | 35 |
| 4.5 | Antall punkter i punktskyene fra fotogrammetri og laserdata | 38 |
| 4.6 | Usikkerhet i sammenkobling mellom overflatemodell og punktsky | 40 |
| 4.7 | Statistikk for segmentvis avstand mellom overflatemodell og punktsky . . . | 42 |
| 4.8 | Evalueringsparametere for maskinlæringsmodell | 42 |

Forkortelser

CPU - Central Processing Unit

EXIF - Exchangeable Image File Format

FPS - Frames Per Second

GNSS - Global Navigation Satellite System

GPU - Graphics Processing Unit

GPS - Global Positioning System

LiDAR - Light Detection and Ranging

MP - Megapiksler

OS - Operating System

RAM - Random-Access Memory

SIFT - Scale-Invariant Feature Transform

SOTA - State-Of-The-Art

TLS - Terrestrisk Laserskanner

VR - Virtual Reality

YOLO - You Only Look Once

Kapittel 1

Introduksjon

1.1 Bakgrunn

Nesten halvparten av landarealet i Norge ligger i dag over skoggrensa (45 %). Dette er landskapet som vanligvis omtales som fjell. Store deler av Norges infrastruktur bygges i disse utfordrende områdene, som fører til en del problemer. (Austrheim et al., 2015)[p. 10]. Klimaendringen fører med seg økning i problemer med bygningsmasse og infrastruktur. Kombinasjon av dårlig vedlikehold og permafrost som tiner gir en risiko for mer ras i tiden som kommer. NGU, 2021. I følge Vegdirektoratet (2011)[p. 9] vil mest sannsynlig økningen i steinskred- og steinsprangaktivitet gjelde for hele landet. Det må antas at endringen i været de kommende 50 årene med mer nedbør, hyppigere situasjoner med kraftig nedbør, lengre og flere frys- og tineperioder er årsak til økningen.

Løsmasser fra fjell utgjør en risiko i veibildet med over 1000 menneskeliv tapt i skredulykker i Norge siden år 1900 (NGU, 2021). Regjeringens nullvisjon, som er beskrevet i Nasjonal tiltaksplan for trafikksikkerhet 2022-2025, jobber for at ingen liv skal gå tapt og at ingen blir hardt skadet i trafikken. (Statens Vegvesen et al., 2021)[p. 15]. Nullvisjonen legger grunnlaget for alt trafikksikkerhetsarbeid som gjøres i Norge. Nasjonal tiltaksplan beskriver en rekke tiltak som er rettet mot drift og vedlikehold av veg. Blant annet beskrives det at Nye Veier AS skal utvikle et registreringssystem for alle hendelser som skjer på sine veier, uavhengig av om det er relatert til skade eller ikke. Formålet er å avdekke feil og systematikk i hendelsene. ”Registreringssystemet vil bli videreutviklet til også å omfatte god stedfesting samt indikatorer for de uønskede hendelsene (tiltak nr. 118)” (Statens Vegvesen et al., 2021)[p. 122]. ”Statens Vegvesen vil på kort sikt gjennomføre dataanalyser

for å identifisere trafikkfarlige punkter og digitalt systematisere funn fra sine inspeksjoner og eksterne tilsyn/revisjoner (tiltak nr. 119)”(Statens Vegvesen et al., 2021)[p. 122]. ”I løpet av planperioden 2022-2025 skal Norge ligge langt framme innenfor intelligente transportsystemer, der vei- og kjøretøyteknologi inngår, og tilgjengelige systemer som bedrer trafikksikkerheten er tatt i bruk” (Statens Vegvesen et al., 2021)[p. 121].

Standarden for drift og vedlikehold, R610, omfatter krav til blant annet drift av vei og sideareal. Denne vil ifølge Nasjonal tiltaksplan revideres, og det settes lys på bruk av tilgjengelige kunnskap- og datakilder til nye løsninger som prioriterer trafikksikkerheten i driftsoppgavene til Statens Vegvesen. (Statens Vegvesen et al., 2021)[p. 126].

1.1.1 Definisjoner

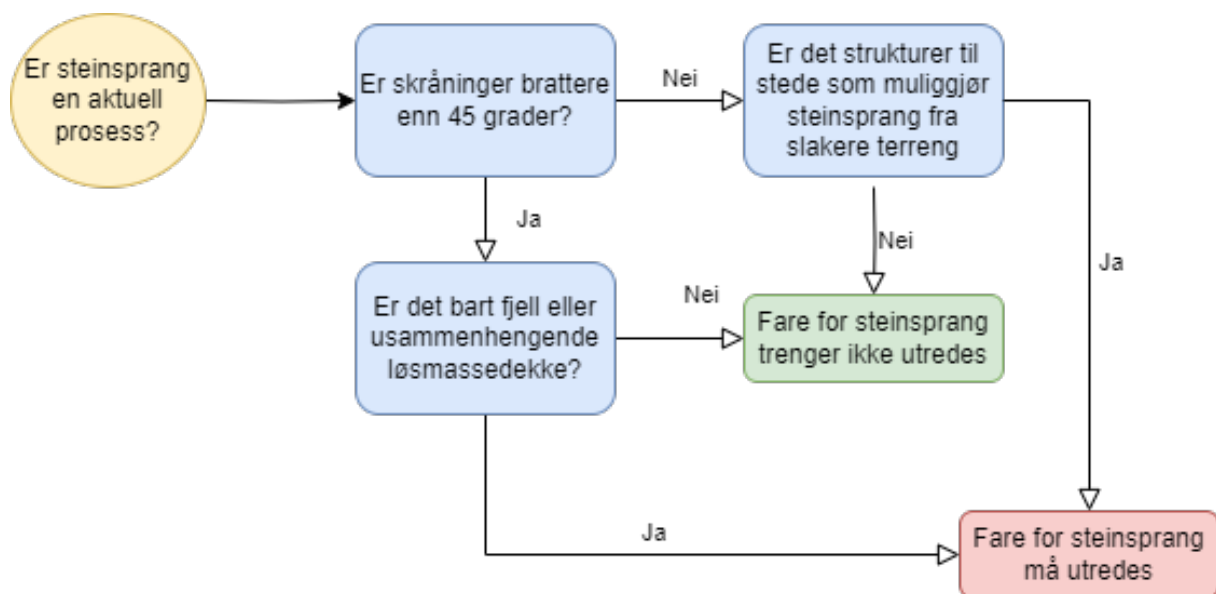
”Steinsprang brukes om hendelser der én eller et fåtall steinblokker løsner og faller, spretter, ruller eller sklir nedover en skråning. Steinblokkene beveger seg nedover stort sett uavhengig av hverandre, og vil tape energi når de spretter i bakken. I bratte fjellsider følger de terrenghelningen, men når terrenget falter ut kan blokkene bøye av og gå ut til sidene for fallretningen.” (NVE, 2018)[p. 2] Nedfall av stein er løse steinmasser og deles ofte inn i steinsprang, steinskred og bergskred avhengig av volumet på de ustabile massene: (Vegdirektoratet, 2011)[p. 5]

- Steinsprang - Volum under 100 m^3
- Steinskred - Volum fra 100 m^3 til $10\,000 \text{ m}^3$
- Bergskred - Volum over $10\,000 \text{ m}^3$

1.1.2 Risikovurdering av steinsprangområder

”Det er i de fleste tilfeller mulig å avgjøre om steinsprang er en aktuell prosess i et område ved bare å bruke tilgjengelig datagrunnlag som f.eks. helningskart, løsmassekart og flybilder. Utfordringen er kvaliteten på datagrunnlaget. Viser det seg å være feil, unøyaktig eller ufullstendig, kan det under feltarbeidet vise seg at steinsprang likevel kan være aktuelt” (NVE, 2021). Feltobservasjoner av løsneområdene er den viktigste kilden ved vurdering av løsnesannsynlighet, og disse må knyttes opp til resultat av undersøkelsene av grunnlagsdata. Skredavsetninger og skredhistorikk brukes for å vurdere historisk aktivitet og må også inngå i vurderingen av løsnesannsynligheten. I figur 1.1 vises NVE (Norges vassdrags- og energidirektorat) sitt flytdiagram for vurdering av steinsprangområder. NVE påpeker at en

av dagens utfordringer er kvaliteten på datagrunnlaget. Oppgaven vil se på muligheten for å styrke skred- og steinsprangshistorikken, som tidligere kan gi en indikasjon på risikobildet langs veien.



Figur 1.1: Flyttdiagram fra NVE for utredning av steinsprangfare. Illustrasjonen har hentet inspirasjon fra (NVE, 2021).

Under prosjektering og bygging av veien skal det skal gjøres nødvendige vurderinger slik at nedfall ikke skal treffe veien. Det skal derfor gjøres vurderinger av lokalstabilitet (mindre blokker) og totalstabilitet (større utglidninger). Lokalstabilitet skal ivaretas av bergsikring og fanggrøft og det gjøres ikke videre beregninger i disse områdene.

I sluttdokumentasjon for bergskjæringer skal inspeksjonsbehov beskrives. Bergskjæringer er gjenstand for hovedinspeksjon/tilstandsvurdering hvert 5. år etter ferdigstillelse. Da gjøres en tilstandsvurdering av geolog, med forslag til supplerende bolter, nett osv. Det er fokus på fare for steinsprang ned på vei og totalstabilitet. Det er bare i de tilfellene der det er tvil om totalstabilitet at det gjøres modellering/beregning etter inspeksjon. Det har tidligere blitt gjort vurderinger av usikre parti ved hjelp av 3D-modell basert på dronebilder.

Stabiliteten i bergskjæringer er gjenstand for tidens tann. Med tiden kan ulike prosesser føre til at nedfall skjer. Dette innebærer frysing/tinging, vegetasjon osv. Man ser også noen ganger nedfall fra nyere bergskjæringer. (A. Magnussen, Seniorrådgiver ingeniørgeologi, personlig kommunikasjon, 11. april 2024)

1.1.3 Registrering av hendelser og vedlikehold av sidearealer

Statens vegvesen har registrert skredhendelser på riks- og fylkesveger siden 1973 (Bjordal & Helle, 2011)[p. 4]. Det er driftsentreprenør for strekningen som registrerer nedfall fra bergskjæringene. R610 Standard for drift og vedlikehold forteller om ulike inspeksjoner på vegnettet (Vegdirektoratet, 2014). Entreprenøren registrerer nedfallene med riktig kode i registreringsprogrammet Collector, slik at det blir integrert i driftssystemet. Vegkart hos Statens vegvesen viser registreringer med kode ”skred”. Dersom det er bekymring for ytterligere nedfall blir geolog tilkalt for videre vurdering og eventuell supplerende stabilitetssikring. (A. Magnussen, Seniorrådgiver ingeniørgeologi, personlig kommunikasjon, 11. april 2024)

1.1.4 Problemstillingens utgangspunkt

Steinsprang registreres i dag ved ren manuell synfaring. Metoden har svakheter knyttet til menneskelige feil, som kan føre til sen eller mangelfull registrering. Visjonen til oppgaven, formidlet av Nye Veier, er å se etter nye måter å detektere steinsprang på som kan erstatte eller supplere den menneskelige synfaringen. Den manuelle registreringen erstattes med automatisk detektering fra et 360-graders kamera montert på en driftsbil. Visjonen er at detekteringen begrenses ned til et rimelig 360-graders kamera, som dermed vil kutte kostnaden og også øke hyppigheten på filming gjort på veiene Nye Veier drifter. Første steg ble en orientering i fagfeltet for å gjøre rede for hva som er gjort av andre aktører og hvilke erfaringer de har tilegnet seg.

1.1.5 Tidligere forskning og anvendelser

Automatisering innen drift og vedlikehold av veibildet er et fagfelt i utvikling, med innovative løsninger og anvendelse av ulike teknologi. Firmaet iSi insight har vært først ute i Norge med en løsning for detektering av feil på autovern. Løsningen baserer seg på masseinnsamling av bilder og skreddersydde maskinlæringsmodeller for bildegjenkjenning. Modellene detekterer automatisk type og feil på rekkverk, og gir nøyaktige posisjoner for hvor de befinner seg (iSi Insight, 2023).

En annen aktør som er i vinden innenfor hendelsesregistrering fra standard kamera er Univrses, som ble kontaktet via Nye Veier. Univrses sitt prosjekt 3D AI beskrives som et nytt paradigme innen urban og motorvei-styring. Ved et montert kamera i frontruten

genererer AI-programvaren egenskaper som veikvalitet, trafikk tetthet, feilparkerte kjøretøy og tilstanden til veiresurser direkte i sanntid. Univrses benytter seg av kameraene i en vanlig smarttelefon i kombinasjon med maskinlæringsalgoritmer til monitorering av veibildet. Univrses fokuserer i hovedsak på veibanen og ting i nær tilknytning til veibanen som skilt, sprekker og hull i asfalt (Univrses, n.d.).

Vår detektering vil strekke seg bredere enn veibanen, siden vårt interesseområde er steinoverflater i sidearealet til vegen. Stereokameraets synsfelt vil være for lite til vårt formål. Her vil et 360-graders kamera gi et komplett synsfelt som er nødvendig for å fange opp hele sidearealet.

Modellering ved hjelp av fotogrammetri basert på standardkameraer er et godt utforsket felt. Bruken av 360-graders kameraer er derimot mindre utforsket. Studien Janiszewski et al. (2022) har forsket på bruk av fotogrammetri ved hjelp av 360-graders kamera til tunnelmodellering. Det kan trekkes flere likhetstrekk mellom studien og oppgaven. Overflatene til tunnelen har samme tekstur som steinoverflatene som kartlegges langs veien og det er brukt like sammenligningsmetoder opp mot laserdata. Studien modellerte en strekning på 10 meter med kameraet Insta360 Pro. Kameraetsystemet til Insta360 Pro består av 6 kameraer, hvor bildene i studien ble brukt separat til modelleringen. Det ble brukt 27 posisjoner på strekningen på 10 meter. På samme strekning ble laserdata samlet inn av Riegl VZ-400i terrestrial laser scanner. Overflatemodellen basert på 360-graders bildene ga et avvik i avstand fra laserdataen på 4,6 mm.

Zoumpekas et al. (2021)[p. 4] sier i sine studier at det vanligste metoden for automatisk detektering av steinsprang er tidsseriebasert sammenligning av punktskyer målt med TLS (terrestrisk laserskanner). Denne teknikken krever en ekspertise og inspeksjon i etterkant, som er svært tid og datakrevende. Forskning på bruk av TLS til dette formålet har kommet i økende grad de siste årene. Studien av Farmakis et al. (2022) tar i bruk maskinlæring på LiDAR-data for å automatisere detekteringsprosessen. Det tas i bruk en enderingsdeteksjonsdatabase som inneholder 8000 punkter fra en steinskråning med markert endring, hentet fra en periode på 5 år. En maskinlæringsmodell er trent på disse punktene og deretter testet på et sett med 536 punkter. Modellen oppnår i følge artikkelen 89% nøyaktighet på forskjellige datasett. Til bruk i en annen steinskreddatabase, oppnår modellen 93% nøyaktighet. Dette er eksempler på hvordan monitorering av rasområder gjøres i dag. Det er ofte brukt stasjonære LiDAR-sensorer for å overvåke usikre steinskråninger. De gir ofte god nøyaktighet, men gir ikke mulighet for monitorering av en

hel veistrekning.

Til oppgaven har Field vært bidragsytende med innsikt i arbeidet gjort med 360-graders bilder og naturlig klassifisering langs veien. De har kjørt store deler av Rogaland med 360-graders kamera og laserskanner. De har derimot ikke gjennomført noe klassifisering kun basert på bildene. Field har også bidratt med datagrunnlaget som brukes videre i oppgaven.

1.2 Problemstilling

Mål

Oppgavens ser på å utforske følgende mål:

- Å styrke grunnlaget for registrering av steinsprang som en støtte for menneskelig registrering.
- Å gjøre metodene for registrering av steinsprang mindre kostbare og mer effektive.

Problemstilling

Oppgaven vil ta for seg følgende problemstilling:

Hvordan kan bilder fra 360-graders kamera i kombinasjon med maskinlæring gi et bedre grunnlag for detektering av steinsprang?

Utdypelse av problemstilling

Denne oppgaven står som et første steg i en helhetlig prosess hvor man undersøker om det er mulig å erstatte nåværende metoder for detektering av steinsprang. Oppgaven legger opp til videre arbeid av resterende punkter i prosessen dersom trinnene utforsket i oppgaven gir lovende resultater.

Erstatningen som undersøkes er om steinsprangene i veiens sideareal kan detekteres automatisk ved hjelp av en maskinlæringsalgoritme trent på et rimeligere 360-graders kamera. Det undersøkes om klassifiseringen av steinområdene, i kombinasjon med en endringsanalyse

basert på en fotogrammetrimodell laget av 360-graders bilder, kan fange opp steinsprang langs veien. Den fulle overordnede prosessen beskrives med fire forskningsspørsmål.

Forskningsspørsmål for full prosess

1. Er nøyaktigheten til en fotogrammetrimodell basert på 360-graders bilder fra eksisterende data tilstrekkelig nok til å detektere volumendringen i et steinsprang?
2. Hva er prestasjonen til maskinlæringsmodellen for klassifisering av steinoverflater i 360-graders bilder?
3. Hvordan er nøyaktigheten til en fotogrammetrimodell fra en rimeligere 360-graders kamera sammenlignet med fotogrammetrimodellen basert på eksisterende data?
4. Hvordan presterer klassifiseringsmetoden på et rimeligere 360-graders kamera?

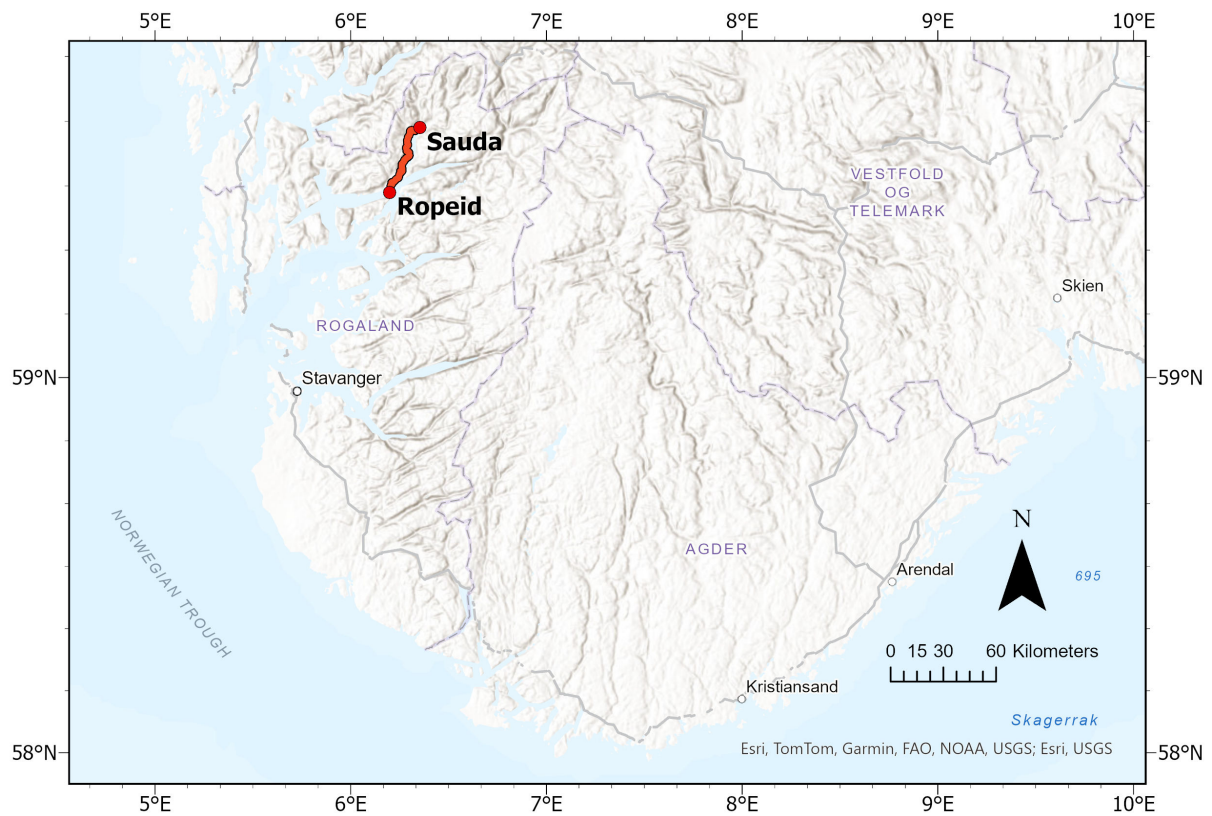
Forskningsspørsmål for oppgaven

For begrensninger i oppgavens omfang vil denne oppgaven fokusere på disse to forskningsspørsmålene som et grunnlag for videre analyse:

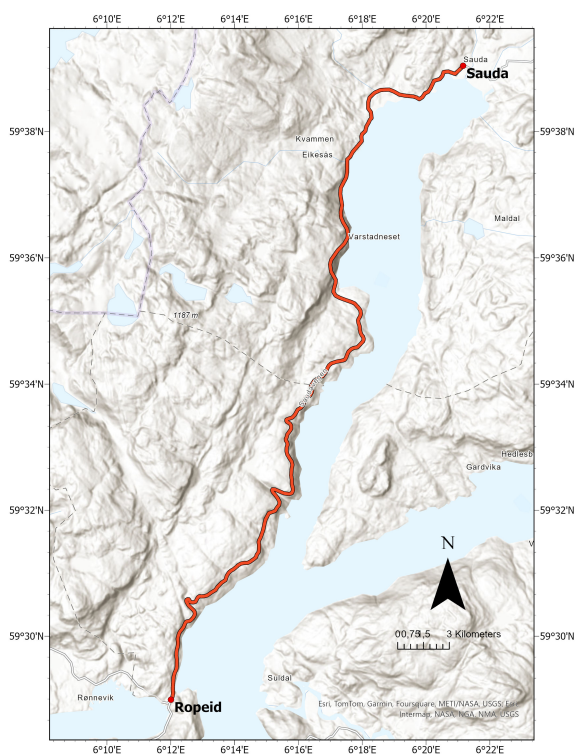
1. Er nøyaktigheten til en fotogrammetrimodell basert på 360-graders bilder fra eksisterende data tilstrekkelig nok til å detektere volumendringen i et steinsprang?
2. Hva er prestasjonen til maskinlæringsmodellen for klassifisering av steinoverflater i 360-graders bilder?

1.3 Beskrivelse av studieområde

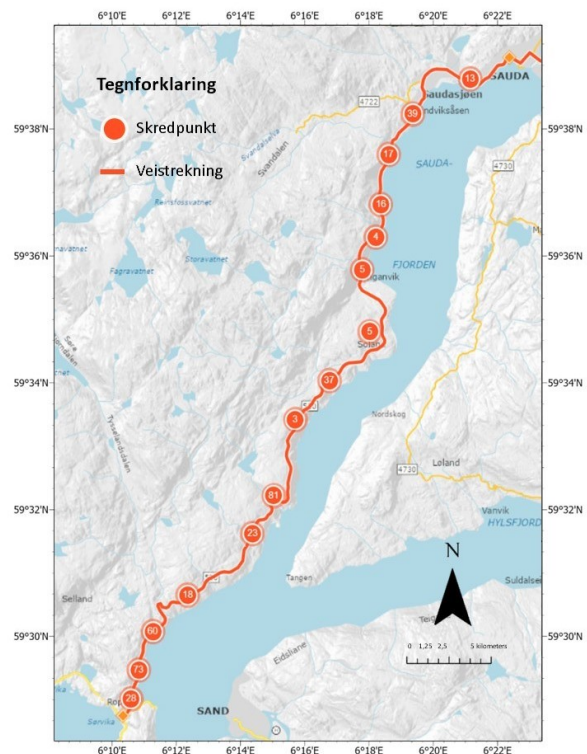
Fokusområdet i oppgaven omfatter veistrekningen Rv 520 mellom Ropeid og Sauda (Metrering: FV520 S1D1 m250 - S3D1 m1695). Denne strekningen er valgt i samråd med Rogaland Fylkeskommune på grunn av dens historie som en svært rasutsatt del av Rogaland. Området er hovedsakelig utsatt for steinsprang fra vegskjæring, men også noe fra naturlig sideterreng. 1.2 og 1.3 viser strekningen som er valgt.



Figur 1.2: Oversiktsbilde over utvalgt strekning fra Ropeid - Sauda, i Rogaland fylkeskommune.

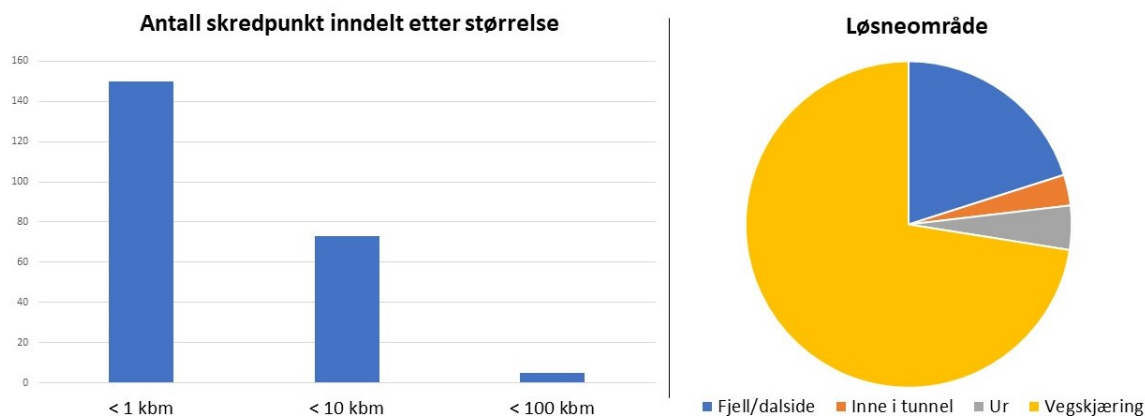


Figur 1.3: Et mer detaljert kartutsnitt over utvalgt strekning fra Ropeid - Sauda.



Figur 1.4: Framstilling av skredpunkt på strekningen fra Ropeid - Sauda.

Figur 1.4 viser antall skredpunkt langs strekningen. Av disse er det tatt ut de 228 tilfellene som er i kategorien ”steinskred”. Steinskredene forekom i perioden mellom 2003-2023. I figur 1.5 er det tatt uttak av kategori steinskred langs gitte strekning. Stolpediagrammet til venstre viser størrelsen på de registrerte steinskredene. Diagrammet gir en indikator på hvilke størrelsesforhold steinsprangene normalt er på. I kakediagrammet til høyre vises fordelingen av løsneområdene i de registrerte tilfellene. Et klart overtall forekommer fra vegskjæringer.



Figur 1.5: Antall skredpunkt per størrelse fremstilt med søylediagram og fordeling av løsneområdene fremstilt med sektordiagram.

Kapittel 2

Teori

Dette kapitlet viser til teori som gir grunnleggende forståelse av temaene 360-graders kameraer, fotogrammetri og maskinlæring.

2.1 360-graders kamera

Standardkameraer har vanligvis et synsfelt, som er det området som er synlig gjennom kameraet, på omtrent $60\text{-}90^\circ$. Selv et vidvinkel-objektiv dekker generelt bare rundt 100° . 360-graders kameraer er laget for å fange opp bilder eller video i hele perspektivet rundt kamerasytemet. Disse kameraene blir i økende grad brukt til forskjellige applikasjoner som VR (Virtual reality).

Kameraene kan være tolinsekameraer eller multilinsekameraer hvor enkeltkameraene er vidvinkellinser. Tolinsekameraene har to ultravidvinkelkameraer som typisk dekker 200° hver. Ulempen med tolinsekameraer er at de ikke kan brukes til 3D rekonstruksjon av en scene i mangel av parallaxse. Parallaxse refererer til hvordan et objekt synes å bevege seg eller endre posisjon når det observeres fra to forskjellige punkter, noe som er viktig i fotogrammetri for måling av avstander (Ringnes, 2023).

Etter bildetakningen gjennomgår bildene ofte en sammensyning som gjør at man får et sammenhengende landskapsbilde som er sømløst der bildene møtes. For å få en tilstrekkelig sammensyning har de fleste 360-graders kameraer $10\text{-}30^\circ$ overlapp. Etter dette blir de også stabilisert som kompenserer for rotasjon av kameraet og stabiliserer horisonten. For å vise sfærisk geometri må innholdet projiseres til et standard flatt bildeformat. Den vanligste metoden å gjøre det på, er ved en ekvirektangulær projisering som er tilsvarende

merkatorprojeksjonen, som er kjent fra vanlige kart. Dette medfører de samme ulempene, som er at toppen og bunnen til kula er svært fortegnet. I oppgaven blir det brukt både ekvirektangulære bilder og bilder som ikke er sammensydd (Avatour, 2024).

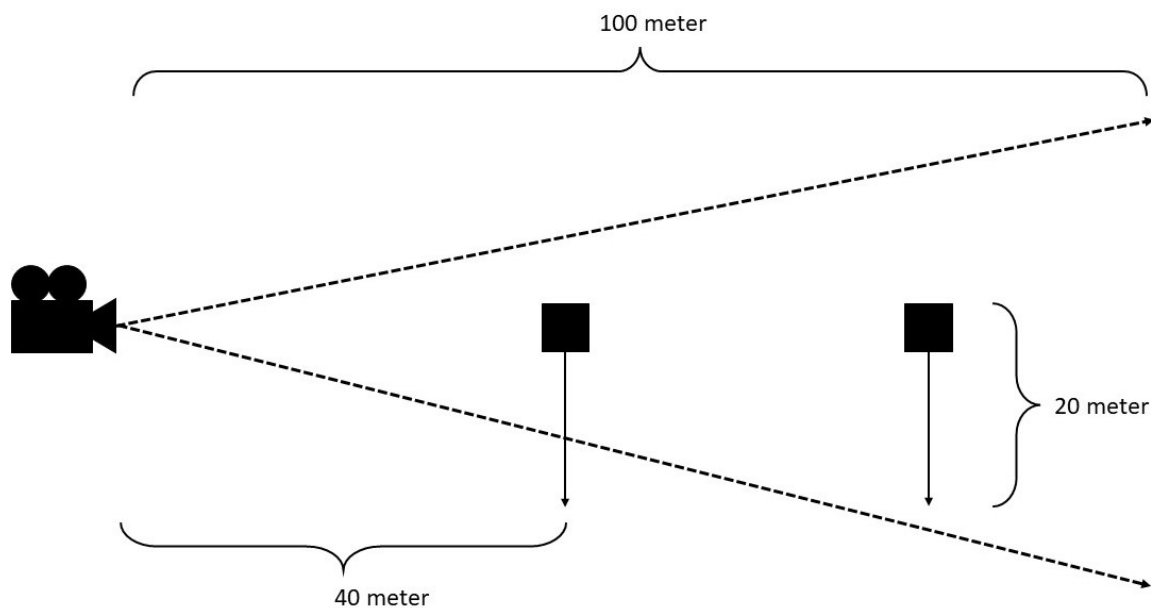
2.2 Fotogrammetri

I denne oppgaven er standard prosedyre for fotogrammetri benyttet. ”Fotogrammetri er vitenskapen bak å tilegne seg informasjon om objekter og overflater uten fysisk kontakt med objektet, og måling og tolkning av denne informasjonen” (Schenk, 2005).

Grunnleggende prinsipper om fotogrammetri som videre ikke diskuteres i oppgaven, men som ligger til grunn for analysen, antas kjent og vil ikke bli beskrevet i teoridelen.

2.2.1 Bevegelsesparalakse

Bevegelsesparallakse er et fenomen der objekter som beveger seg med en konstant hastighet over et bilde, ser ut til å bevege seg mer hvis de er nærmere en observatør eller et kamera, sammenlignet med om de var lengre unna. Dette gjelder uavhengig av om det er objektet som beveger seg, eller om det er observatøren eller kameraet som beveger seg i forhold til objektet. Effekten oppstår på grunn av forholdet mellom hvor langt objektet beveger seg og hvor stor del av kameraets synsfelt det dekker. For eksempel, et objekt som er 100 meter unna kan bevege seg 20 meter i en retning og dekke bare 25% av synsfeltet, mens et objekt som kun er 40 meter unna kan bevege seg ut av bildet ved samme forflytning på 20 meter (Brinkmann, 2008).



Figur 2.1: Figuren viser bevegelsesparallaksen til to objekter i synsfeltet til kameraet. Den viser at objekter lenger unna beveger seg relativt mindre i forhold til synsfeltet til kameraet. Figur er inspirert av kilde (Brinkmann, 2008).

2.2.2 Egenskapsmatching

Egenskapsmatching (Feature matching) er en teknikk som brukes i fotogrammetri for å finne samsvarende objekter (features) mellom forskjellige sett av bilder. Det brukes algoritmer som finner punkter i bildene som er stabile for endringer i skala, synsvinkel, fortegning og belysning. Programvaren brukt i denne oppgaven, benytter en variant av SIFT (Scale-Invariant Feature Transform) til sammenbindingen. (Semyonov, 2011) SIFT-prosessen kan deles inn i fire steg. Første steg er å bruke uskarphetsfiltere, som danner et sett av bilder i forskjellige grader av uskarphet. Uskarphetsfilterene får frem distinkte punkter i bildene. Av disse bildene blir det laget et sett med bilder i forskjellig skala for å sikre at sammenbindingen skjer uavhengig av skala. Deretter brukes teknikken Difference of Gaussians (DoG). Her trekkes forskjellige uskarphetsnivåer fra hverandre. Resultatet er kun objektene som skiller seg ut i bildet som kanter og kontrastfulle områder. SIFT plukker ut lokale minimums og maksimumspunkter i bildet. De punktene med lav kontrast og som er helt i kanten av bildet blir så fjernet. Det gir oss settet med punkter som testes opp mot de andre bildene i sammenbindingen. Disse punktene er i Agisoft Metashape referert til som keypoints. Punktene får en tilhørende beskrivelse av posisjonen sin i forhold til de andre punktene. Siste steg i prosessen er sammenkoblingen basert på punktene. (Singh, 2024)

2.3 Klassifiseringsteknikker

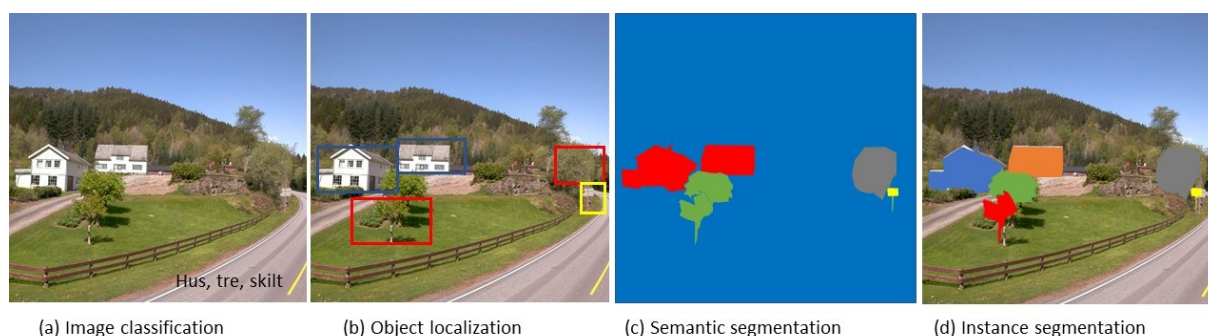
2.3.1 Maskinl ring - En introduksjon

En maskinl ringsmodell deles ofte inn i veiledet l ring og ikke-veilet l ring, hvor f rstnevnte er benyttet i oppgaven. ”Hovedm let med veiledet l ring er   l re opp en modell fra merkede treningsdata som tillater oss   gj re prediksjoner om usett eller fremtidig data. Veiledet til et sett med treningseksempler (Inndata) der  nskede resultater (merker) allerede er kjent” (Raschka & Mirjalili, 2019, p. 3).

Bildeprosessering omhandler analysen av digitale bilder ved hjelp av diverse algoritmer og teknikker. Dette feltet benytter maskinl ringsmodeller for blant annet   utf re tre oppgaver som er n rt beslektet: image classification, object localization, semantic segmentation, og instance segmentation.

Image classification identifiserer og klassifiserer objekter i et bilde ved   tilordne hele bildet til en eller flere kategorier. Object detection gir mer detaljert informasjon om kategorien ved   bruke avgrensingsbokser til   bestemme posisjonene til kategoriene.

Semantic segmentation g r enda dypere ved   tildele hver piksel i bildet til en bestemt kategori, noe som gir en detaljert piksel-for-piksel representasjon av ulike objekter. Instance segmentation gir i tillegg til en piksel-for-piksel representasjon av kategorien p  individniv . (Hafiz & Bhat, 2020)[p. 171]



Figur 2.2: Visualisering av ulike segmenteringsmetoder innen bildeprosessering.

2.3.2 Instance-segentering

Instance-segentering er en viktig teknikk innen maskinl ring som fokuserer p    forutsi b de klassen og den eksakte konturen av hvert enkelt objekt p  pikselniv  i et bilde. Denne metoden g r et skritt videre enn semantisk segentering ved ikke bare   klassifisere hver

piksel basert på objektkategori, men også ved å skille mellom individuelle enheter innenfor samme kategori. Dette gjør instance-segmentering særlig verdifull for anvendelser som krever detaljert bildeforståelse, som i robotikk, autonom kjøring og overvåkningssystemer. (Hafiz & Bhat, 2020)[p. 171]

2.3.3 YOLOv8: Anvendelse og prinsipper

YOLOv8 er et objekt-deteksjonsalgoritme utviklet av Ultralytics som regnes som state-of-the-art (SOTA). SOTA betyr at det er det nevralt nettverket som presterer best på en gitt oppgave. SOTA bemerkelsen er vurdert ut i fra prestasjonen på nøyaktighet og hastighet på modellen. Det er ofte et kompromiss mellom nøyaktighet og hastighet. Å forbedre prestasjonen på en av de, kan gå på bekostning av en annen (Shahab, 2023).

Algoritmen til YOLOv8 deler først opp opp bildet i et grid. For hver celle predikeres et antall avgrensingsbokser. Avgrensingsboksene har en tilhørende sannsynlighet for boksens gyldighet og hvilken klasse den tilhører. YOLO filtrerer ut overlappende avgrensingsbokser og finner hvilke avgrensingsbokser som er mest sannsynlig at stemmer. (Shahab, 2023) YOLOv8 er et CNN (convolutional neural network) som betyr at modellen har et lag-for-lag hierarki hvor det benyttes konvolusjonslag og samlelag. Konvolusjonslagene er grunnlaget for mange CNN-modeller og tar i bruk ulike filter for å få ut kanter og teksturer i bildet. Samlelagene reduserer dimensjonene i laget for å gjøre modellen mer sikker mot små variasjoner i bildet. (IBM, n.d.)

YOLOv8 skiller seg fra tidligere YOLO versjoner ved at den presterer bedre på flere parametere. Versjonens redefinerte modellarkitektur gjør den bedre på nøyaktighet, hurtighet og robusthet. Dette gjør modellen godt egnet til oppgaver som krever rask, men nøyaktig objekt-detektering. YOLO er tilgjengelig som åpen kildekode (open source). (Ali, 2023)

2.3.4 Validering av klassifiseringsteknikker

I objekt-deteksjonsmodeller er valideringsparametere avgjørende for å vurdere nøyaktigheten og effektiviteten av modellene. Disse verdiene gir innsikt i hvordan modellen identifiserer og lokaliserer objekter i bilder, samt håndterer falske positive og negative. Forståelsen av disse indikatorene er sentral for evaluering og forbedring av modellens prestasjon. Det brukes forskjellige verdier i denne oppgaven til å vurdere av modellens ytelse:

Precision gir et mål på prosentandelen av prediksjoner gjort av modellen som er korrekte. Gitt alle predikerte tilfeller, hvilken andel av disse er riktig predikert. (Huilgol, 2024)

Recall gir et mål på prosentandelen av relevante datapunkter som ble korrekt identifisert av modellen. Gitt de markerte tilfellene, hvilken andel av disse er riktig predikert. (Huilgol, 2024)

F1-score er det harmoniske gjennomsnittet av Precision og Recall. Det gir en balansert vurdering av modellens ytelse. (Sharma, 2023)

mAP (mean Average Precision) tar gjennomsnittet av AP, som beregner arealet under presisjon-gjenkallingskurven, over flere objektklasser. Dette gir en helhetlig evaluering i scenarier med flere klasser. I oppgavens tilfelle er det ingen forskjell på mAP og AP siden man kun opererer med en klasse. mAP gir en helhetlig vurdering av modellens ytelse.

Formler for valideringsparametere (Hui, 2018)(Sharma, 2023):

EP = Ekte positiv

FP = Falsk positiv

EN = Ekte negativ

FN = Falsk negativ

$$\text{Precision} = \frac{EP}{EP + FP} \quad (2.1)$$

$$\text{Recall} = \frac{EP}{EP + FN} \quad (2.2)$$

$$\text{Accuracy} = \frac{EP + EN}{EP + EN + FP + FN} \quad (2.3)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.4)$$

Kapittel 3

Metode

Dette kapitlet beskriver metodene som er brukt for å lage fotogrammetrimodellen og maskinlæringsalgoritmen til klassifisering.

3.1 Modellering via fotogrammetri

3.1.1 Utstyr

Mosaic X

Mosaic X er et 360-graders kamera produsert av Mosaic. Kameraetsystemet består av fem kameraer arrangert i en sirkel og et sjette kamera som peker oppover. Hvert av disse har en oppløsning på 12.32 megapiksler og en bildefrekvens på 10 bilder per sekund. Mosaic fremlegger et kamerasystem som gir skarpe bilder fra alle vinkler uten tap av oppløsning i kantene. Systemet har tilstrekkelig med overlapp mellom kameraene som gir en god kombinasjon mellom overlapp mot oppløsning.

Hovedprosessoren kombinerer en kraftig CPU og GPU, noe som gjør den helt mobil uten behov for eksterne lagringsenheter eller datamaskiner i kjøretøyet. Systemets kabinett og kjøleribber gir stabil temperatur i systemet og linsene har beskyttelse som skal tåle hard daglig bruk. Kameraet er også integrert med fire globale navigasjonssatellittsystemer (GNSS) til geolokalisering og sporing under bruk. Dette gjør systemet godt egnet til omfattende bildeinnsamling til 3D-modellering og fotogrammetri. (Mosaic, 2023)

Tabell 3.1: Tabellen viser kameraspesifikasjoner til 360-graders kameraet Mosaic X. (Mosaic, 2023)

| | |
|-------------------|--|
| Sensor | Sony 1-inch Global Shutter |
| Brennvidde | 5.5mm |
| Pikselstørrelse | $2.74 \mu m$ (H) \times $2.74 \mu m$ (V) |
| Blenderåpning | f/2.4 |
| Fokuseringsområde | 1m - ∞ |



Figur 3.1: Illustrasjonsbilde av kamerasystemet Mosaic X (Mosaic, 2023).

Velodyne Puck

Velodyne Puck, tidligere kjent som VLP-16, er en kompakt lidarsensor kjent for sin robusthet og effektivitet. Den tilbyr en rekkevidde på 100 meter fra et kompakt system. Systemet bruker godt testet 905 nm teknologi som tilbyr et godt synsfelt, førsteklasses nøyaktighet og kalibrert intensitet. Den er spesielt designet for lavhastighetsautonomi og førerassistansesystemer. Med dobbel returfunksjon forbedrer det systemets evne til å fange detaljerte 3D-data fra komplekse miljøer. Dette gjør Puck egnet for en rekke applikasjoner inkludert kartlegging og robotikk. (Hexagon, 2024)

Tabell 3.2: Laserskanneren Velodyne Puck sine spesifikasjoner. Hentet fra kilde (Hexagon, 2024).

| | |
|----------------------|----------------------------|
| Sensor | VLP-16, med Dual Returns |
| Kanaler | 16 |
| Rekkevidde | 100m |
| Punktetthet | 300,000 Punkter per sekund |
| Synsfelt horisontalt | 360° |
| Synsfelt vertikalt | $\pm 15^\circ$ |

PC til prosessering

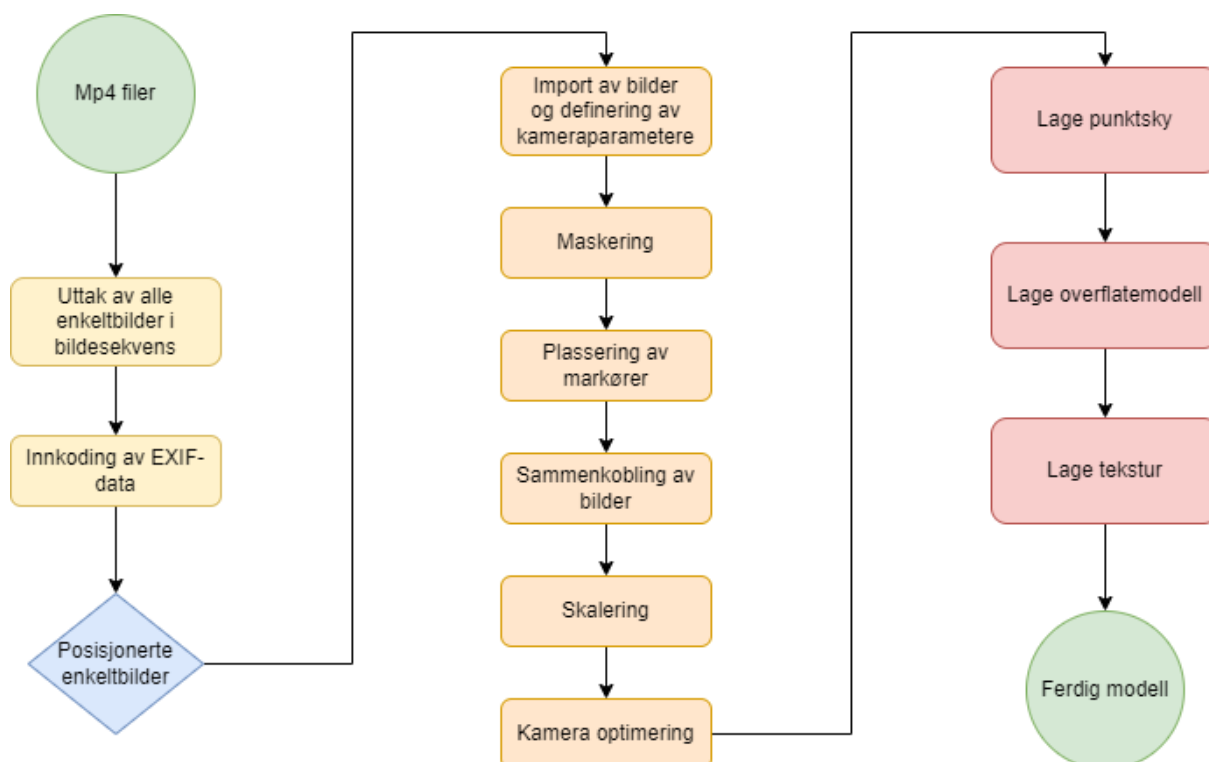
Tabell 3.3 viser informasjon om PC og system som er brukt til modelleringen.

Tabell 3.3: Informasjon om PC brukt til prosessering i Agisoft Metashape.

| Systeminformasjon | |
|-------------------|--|
| OS | Windows 64 bit |
| RAM | 7.88 GB |
| CPU | Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz |
| GPU | Intel(R) UHD Graphics 620 |

3.1.2 Prosessflyt for fotogrammetri

Figur 3.2 viser flytdiagrammet fra rådata til ferdig modeller brukt videre i analysen. Til venstre er preprocessing gjort på rådata før det tas inn i Agisoft Metashape. I midten er det beskrevet prosesser gjort i Agisoft Metashape for å klargjøre for modellering. Til høyre lages punktskyen, overflatemodellen og tekstur til modellen. Sluttproduktet er en overflatemodell med tekstur.



Figur 3.2: Flytdiagrammet viser hvert steg fra rådata i form av Mp4-filer til den ferdige overflatemodellen med tekstur.

3.1.3 Programvare

Python

Python er et populært programmeringsspråk som er laget i 1991. Språkets popularitet kommer av en allsidighet og lesbarhet. Python er et språk som er veldig nært det engelske

språk og gjør at brukerne kan skrive programmer med færre linjer enn hvis man skulle brukt andre språk. (W3schools, n.d.) (Python Software Foundation, 2023)

MistikaVR

MistikaVR er programvare fra SGO. Programvaren brukes til sammensying av 360-graders video og er integrerbar med Mosaic sine kameraer. Tjenesten er mye brukt i VR-miljøet. (SGO, 2023)

Agisoft metashape

Programvare for fotografisk behandling av digitale bilder og generering av 3D-romlig data. Dataen som behandles kan være sfæriske bilder, ortofoto og punktskyer. (Agisoft LLC, 2023)

CloudCompare

Programvare for bearbeiding av 3D-punktskyer. Det er designet for sammenligning mellom flere punktskyer, eller mellom punktskyer og triangulære nett. (CloudCompare, 2023)

VideoProc Converter AI

Programvare for sammensying av forskjellige kameraposisjoner i et kamerasystem. Det kan være 360-graders bilder og stereobilder. (Digiarty Software, 2023)

TrainYOLO

TrainYOLO er en platform for trening av YOLO-modeller. Det består av ulike annoteringsverktøy for bilder og gir mulighet for anvendelser til image classification, object classification, semantic segmentation og instance segmentation-prosjekter. (TrainYOLO Developers, 2023)

Google Colab

Colab er en vertsbasert Jupyter Notebook-tjeneste som ikke krever noen oppsett for å bruke og gir gratis tilgang til databehandlingsressurser, inkludert GPUer og TPUer. Colab er spesielt godt egnet for maskinlæring, dataanalyse og utdanning. (Colab, n.d.)

3.1.4 Integrering av Kunstig Intelligens i forskningsmetodikk

I denne oppgaven har kunstig intelligens, spesifikt ChatGPT-4 fra OpenAI, vært benyttet for å forbedre og effektivisere flere deler av forskningsprosessen. AI-verktøyet har bidratt til

korreksjon av skrivefeil og grammatikk, og til formatering og strukturering av LaTeX-kode. Videre har ChatGPT-4 assistert med feilsøking og forståelse av Python-kode og annen programvare som er anvendt i metoden. ChatGPT har forbedret både nøyaktigheten og effektiviteten igjennom hele prosessen.

3.1.5 Datainnsamling og -behandling

Datagrunnlaget til oppgaven ble levert av Field og innsamlingen ble utført den 20.05.2023. Datainnsamlingen brukte utstyret vist i 3.3, som inkluderer 360-graders kameraet Mosaic X og laserskanneren VLP-16. Disse var fast montert på en av Fields biler.



Figur 3.3: Bilde av rigg med kameraet Mosaic X og laserskanneren VLP-16 (Field, personlig kommunikasjon, 27. februar 2024).

Hos Field blir 360-graders bildene i hovedsak brukt til fargelegging av punktsky fra LiDAR. Bransjestandarden for 360-graders bilder ligger på fotografering med et intervall på 20 meter. Bildene er dermed sydd sammen i programmet Mistika VR til et sammenhengende bilde. Sammensyningen blir gjort i Mistika VR ved hjelp av en styringsfil fra Mosaic. Styringsfilen definerer parametere og posisjoner for sammensyningen og etterlater et sammenhengende bilde fra de 6 posisjonene uten tegn til sømmene i overlappen. De 6 kameraposisjonene i sirkel sammensyes i horisontal retning mens kameraet som peker oppover strekkes ut og dekker hele bredden til bildet i øvre del. Sluttresultatet er et sammenhengende bilde på 8000 x 4000 piksler vist i figur 3.4. Metadataen til bildene er så strukturert i .json-filer.

Alt dette ble gjort av Field og inngikk ikke i prosesseringen i oppgaven.

Opprinnelig datagrunnlag gitt til bruk i analysen besto av sammensydde 360-graders bilder og laserdata innhentet og prosessert av Field fra strekningen Rv 520 mellom Ropeid og Sauda (Metrering: FV520 S1D1 m250 - S3D1 m1695).



Figur 3.4: Bilde fra 360-graders kamera etter sammensying i Mistika VR.

3.1.6 Overflate og tekstur-modellering

Modellen i studien baserte seg opprinnelig på sammensydde 360-graders bilder. Av manualen til Agisoft Metashape (Agisoft, 2021) skal kildedata fra sfæriske kamera (Ekvirektangulær projeksjon) kun trenge definert kameratype til orientering av kameraene. Ingen andre input enn bildene skal være nødvendig. Bildenes posisjoner ble lagt inn i Agisoft Metashape under referanser. Gitt manualens instruksjoner ble kameratype satt til "Spherical" og kameraorientering ble kjørt uten manuell markering av referansepunkter. Denne initielle modelleringsmetoden resulterte ikke i en tolkbar modell, noe som førte til utforskning av alternative teknikker for å forbedre modellens nøyaktighet og kvalitet.

Mosaic, leverandøren av 360-graders kameraet, ble kontaktet for å få en vurdering av det originale datagrunnlaget. De påpekte at et intervall på 20 meter var for langt for å oppnå en tilstrekkelig fotogrammetrimodell. Anbefaling fra Mosaic var i stedet et intervall med øvre grense på 1 eller 2 meter for optimal nøyaktighet. Mosaic har uttalt at et intervall over dette nivået ville resultere i for svake resultater til formålet. Vurderingen baserer seg på deres dybdekunnskap om teknologien og erfaringer fra tidligere prosjekter innen liknende

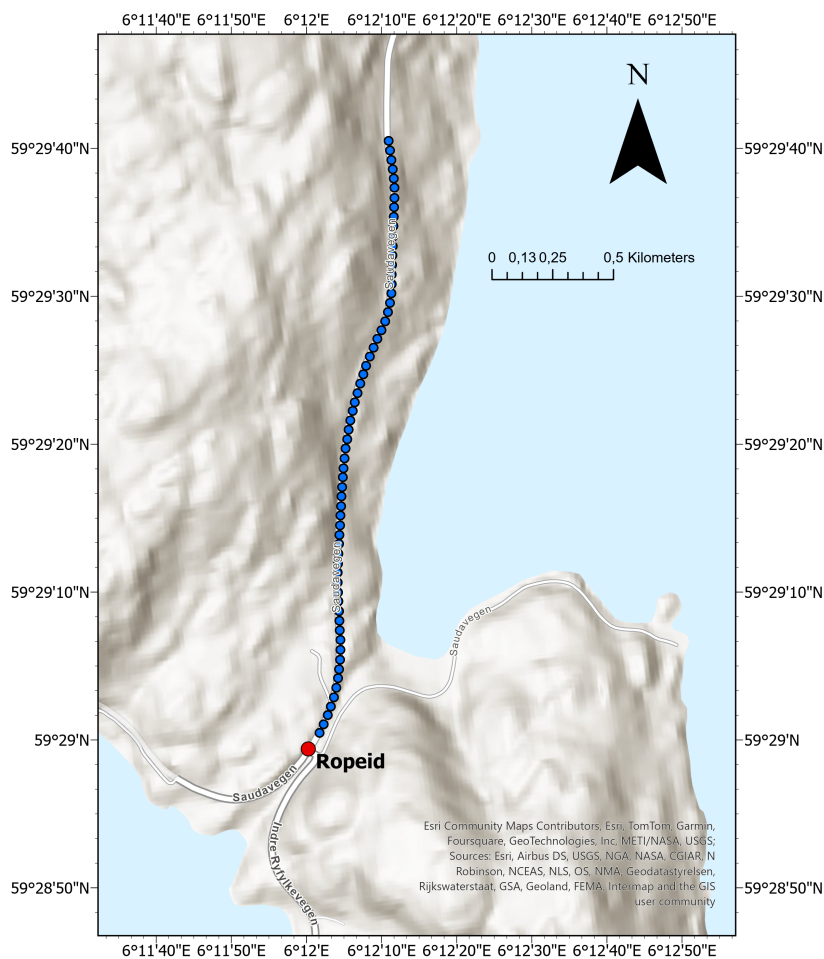
modellering. Etter denne tilbakemeldingen ble Field kontaktet for å styrke datagrunnlaget. Field ga tilgang til rådata fra filmingen som inneholder større tetthet av bilder. Videre modellering ble utført med rådataen levert av Field.

Rådataene besto av enkeltstående MP4-filer fra hvert av de 6 kameraposisjonene til kamerasystemet. 3.5 viser de ulike kameraposisjonene hvor fem av kameraposisjonene er i ring og et kamera peker oppover.



Figur 3.5: Viser eksempelbilder fra hver kameraposisjon i kamerasystemet Mosaic X. I resten av oppgaven vil kameraene følge betegningen kamera 1 - 6 illustrert her med 1-3 i første rekke og 4-6 i andre rekke.

Videoene fra kamerasystemet er inndelt i segmenter langs hele strekningen. Et segment ble valgt for å minke prosesseringstiden. Segmentet som er valgt ut er visualisert i 3.6. Det valgte segmentet inneholder nå en MP4-video fra hver kameraposisjon, hvor hver video består av 64 bilder. Intervallet mellom bildene er redusert til 4 meter, noe som markerer en signifikant forbedring sammenlignet med det tidligere intervallet på 20 meter.



Figur 3.6: Kartet viser utvalgt segment brukt til overflatemodell. Blå punkter viser de 64 kameraposisjonene langs strekningen.

Oppdeling av rådata

Rådataen direkte fra filmingen ga datagrunnlag nede på fire meters mellomrom mellom bildene. Oppdeling av MP4-filene ble gjort i VideoProc Converter AI. Her var det viktig å ikke miste noe kvalitet i prosesseringen. Hvert bilde i bildesekvensen ble tatt ut og organisert i en mappestruktur for hver kameraposisjon. Til innlasting av dataen ble hele mappestrukturen importert til Agisoft Metashape.

EXIF-koding

En bildefil har metadata knyttet til seg, som kalles EXIF-filen til bildefilen. Rådataen fra Mosaic-kameraet er strukturert slik at bildefilene ikke har GPS-koordinater knyttet til

seg, men heller en separat gpx-fil hvor GPS-koordinatene er synket opp mot hvert bilde i bildefrekvensen. Til sammenkoblingen av bildene i Agisoft Metashape er det ønskelig å ha GPS-koordinatene (Lengdegrad, breddegrad og høyde) i EXIF-dataen til bildene. Det ble da brukt EXIF-tool by Phil Harvey (Harvey, n.d.) til å kode inn koordinatene til hvert bilde. Script til koden ligger vedlagt.

Maskering

Det er anbefalt å maskere ut alle irrelevante elementer i bildene fordi disse kan være forvirrende i sammenkobling og konstruksjon av modellen. (Agisoft, n.d.). I gitte tilfellet er det manuelt laget masker til alle 384 bildene i serien ved hjelp av Agisoft Metashape sine maskeringsverktøy. I hovedsak dreier det seg om tre ulike objekter; himmelen, riggen som brukes til bildetakningen og andre objekter som er i bevegelse relativt til bilen. Det ble testet med og uten masker til modelleringen. Ut i fra resultatene ble det ble vurdert som nødvendig med masker for at sammenkoblingen av bildene skulle fungere. Figur 3.7 viser to bilder med tilhørende masker. Maskene er binære bilder hvor det hvite representerer det som beholdes i bildet og det svarte representerer det som er maskert ut. Maskeparet til venstre viser himmelen og riggen til bildetakning. Høyre maskepar viser himmel og en forbipasserende bil.



Figur 3.7: To bilder med tilhørende binær maske, hvor hvitt beholdes i bildet.

Markering av markører

For sammenkobling av bildene ble det lagt inn 51 markører langs strekningen. Markørene ble manuelt registrert, både mellom forskjellige kameraposisjoner, men også mellom de forskjellige kameraene i kamerasystemet. Etter sammenkoblingen ble åtte av markørene deaktivert etter vurdering gjort basert på estimatet av usikkerhet i punktene og visuell kontroll.

Sammenkobling

I valg av parametere til sammenkoblingen ble nøyaktighet satt til høy og kildedataen satt til source. Nøyaktigheten er valgt for å holde prosesseringstiden til et rimelig nivå. Kildedataen er valgt for å ta hensyn til gps-posisjonene til kameraene. Guided image matching er valgt fordi det øker antall keypoints per bilde uten økning i prosesseringstiden. Manualen for Agisoft Metashape sier at det gir bedre resultater for bilder med vegetasjon, eksempelvis skogterreng og gress (Agisoft, 2021). Key point limit ble satt til 80,000 og tie point limit til 0 som tilsvarer ubegrenset antall tiepoints. Standardverdien til key point limit er 40,000. Det argumenteres for at det ikke utgjør store forskjellen på dense cloud dersom denne økes. (Erium, 2019) Mask tie points er valgt for å ta hensyn til maskene laget i Agisoft Metashape.

Skalering

Skalering av modellen kan gjøres ved å markere en kjent avstand i bildene. Til skaleringen ble det hentet ut avstand mellom to hushjørner fra ortofoto i Norge i Bilder. I Agisoft Metashape ble de samme punktene markert og det ble etablert en målestokk på seks meter mellom disse to punktene. Figur 3.8 viser målingen gjort i Norge i Bilder, og figur 3.9 illustrerer hvordan disse punktene er markert i bildene i Agisoft Metashape.



Figur 3.8: Viser utklipp fra uthentet avstand fra Norge i bilder mellom to hushjørner (Norge i Bilder, 2024).



Figur 3.9: To markerte hushjørnet i Agisoft Metashape som er brukt til skalering av modellen.

Proessen "Optimize Camera Alignment" ble brukt til å optimalisere den indre orienteringen basert på den kjente størrelsen mellom hushjørnene. Kameraparameterne ble her kalibrert med en selvkalibrering som ga en nøyaktighet til målestokken på 1 mm.

Punktsky

Etter visuell kontroll av kameraposisjonene ble punktskyen laget i "build point cloud". Dette inkluderer prosessen med å lage depth maps. Kvaliteten ble her satt til "høy" og filtering mode satt til "mild". Datakilden ble satt til dybdekartene. Fullstendig oversikt over parametere brukt under modellering kan leses ut i fra vedlegg "Agisoft Rapport" 7.0.2

Overflatemodell med tekstur

Det ble laget overflatemodell av strekningen i verktøyet "Build Model". Kildedataen som ble valgt var dybdekart og kvalitet ble satt til "høy".

Verktøyet "Build texture" ble brukt til å forbedre overflateteksturen til modellen. Her ble standardverdiene brukt med teksturtype diffuse maps og kildedataen satt til bildene.

3.1.7 Sammenligningsmetode for modellen

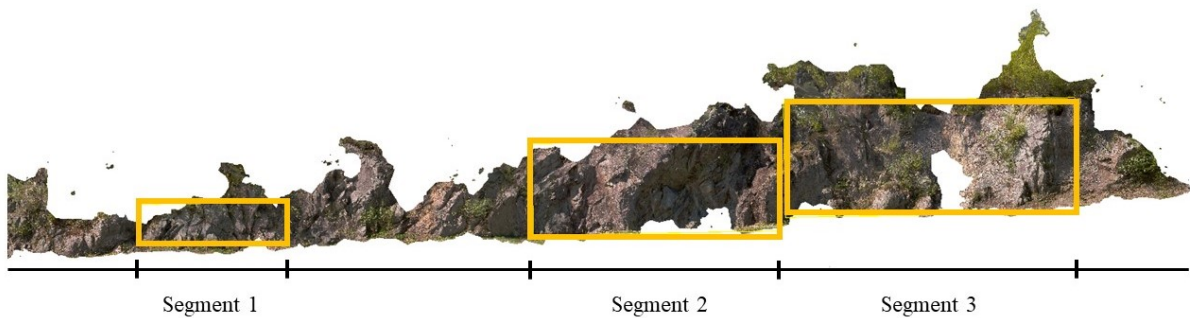
For å få et mål på nøyaktigheten til modellen ble det valgt å sammenligne modellen opp mot punktskyen fra LiDAR. Cloud Compare har verktøy for sammenligning av overflatemodell og punktsky kalt "Cloud-to-mesh Distance". Overflatemodellen ble eksportert fra Agisoft metashape som obj-fil. Det ble brukt lokalt datum for eksporteringen. Dette ble gjort fordi Cloud Compare ikke håndterer høye verdier i koordinatene. Overflatemodellen og punktsky fra laserskann ble importert inn i Cloud Compare. For å kompensere for de høye verdiene i koordinatsystemet til laserskannet ble det lagt til et skift fra det originale koordinatet til et lokalt koordinatsystem. For å justere koordinatene, er forskyvningen satt til -341000 i både x-, y- og z-retningen, som posisjonerer nullpunktet til avgrensingsboksen for laserdataene i nullpunktet til det lokale koordinatsystemet.

Sammenkobling av overflatemodell og laserdata

For å sammenkoble overflatemodellen med laserpunktskyen, ble sju punkter som var synlige i begge modellene valgt ut. Punktene er valgt ut fra karakteristiske steder i fjellside og i veimarkeringene. Det var viktig å velge punkter som ikke er på linje slik at rotasjonen ble korrekt. Cloudcompare gjennomfører en analyse av RMS-verdier for sammenkoblingen.

Segmentering

Tre segmenter ble klipt ut fra punktskyen og overflatemodellen for sammenligning. De utklipte segmentene vises i figur 3.10. For hvert segment ble det brukt verktøyet "Cloud-to-mesh distance". Punktskyen ble satt som referanse og overflatemodellen til sammenligning.



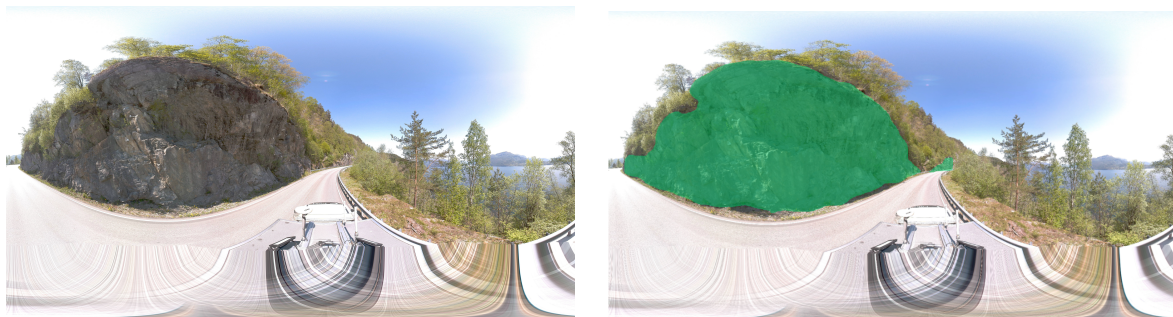
Figur 3.10: De tre gule boksene viser segmenter som er utvalgt til videre sammenligning opp mot laserdata.

3.2 Utvikling og validering av klassifiseringsmodell

Denne delen beskriver metoden som brukes for å lage treningsdata og trene opp en maskinlæringsalgoritme til klassifisering av steinområder.

3.2.1 Bildeannotering

500 bilder ble lastet inn i annoteringsverktøyet til TrainYOLO. Annoteringsverktøyene i TrainYOLO ble brukt til markering av områdene som klassifiseres som steinoverflater. Det ble brukt brush-tool til manuell markering av de første bildene. Et lite sett med bilder ble brukt til opptrening av første YOLO-modell. 3.11 viser hvordan et bilde ser ut før og etter annotering.



Figur 3.11: Bilde før og etter annotering av steinområder gjort i annoteringsverktøyet til TrainYOLO.

3.2.2 Modelltreningsprosedyre

TrainYOLO gir mulighet for oppretting av et Google Colab script for opptrening av maskinlæringsmodellen. Scriptet importerer inn de annoterte bildene og man kan definere

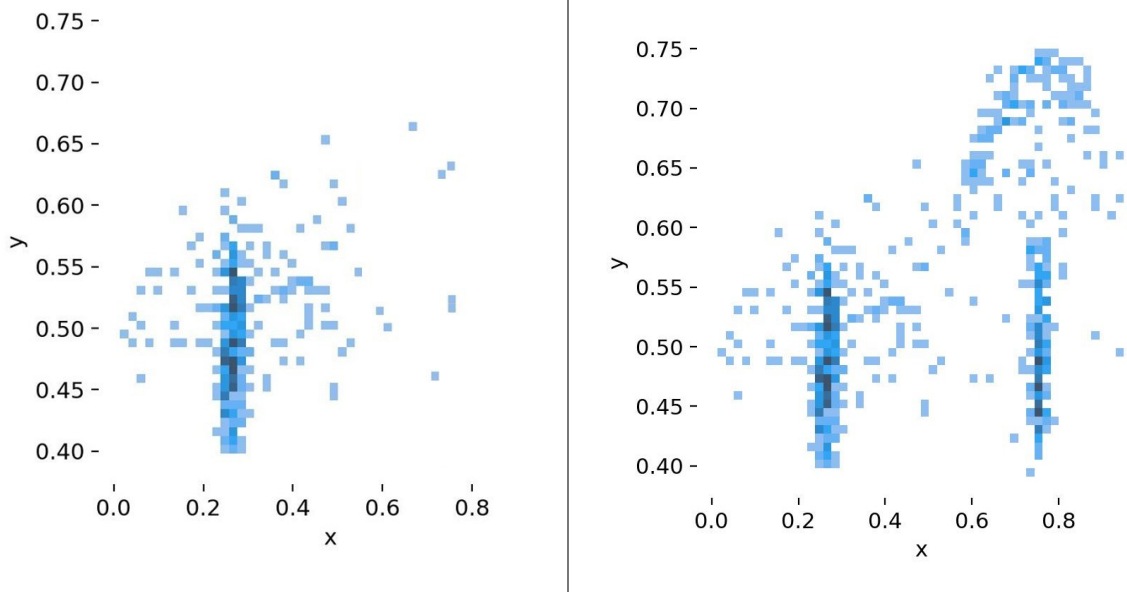
parameterne til treningen. Tjenesten definerer automatisk et trenings- og et valideringsdatasett med en 80/20 splitt. Tabell 3.4 under viser parameterene som er valgt for treningen. Prestasjonen til maskinlæringsmodellen kan øke dersom Image Size er høyere. Det ble dermed testet ut hvilken størrelse som PCen kunne håndtere. I dette tilfellet tilsvarte det 1850 piksler, et tall som kunne vært høyere med annen PC. Number of epochs ble valgt til et antall som brukte rimelig med tid. Det ble brukt standard verdier for paramterene utover dette. I og med at fokuset for oppgaven er nøyaktigheten i overflatemodellen er det ikke videre testet og sammenlignet prestasjon basert på andre parametere i maskinlæringsmodellen.

Tabell 3.4: Viser konfigurasjonsparametere brukt i maskinlæringsmodellen. "Image size" definerer hvor store bilder som går inn i modellen. Dersom bildene er større blir de komprimert ned til størrelsen. "Number og epochs" definerer hvor stor andel data som brukes til en frem og tilbake passering gjennom nettverket. "Model File" definerer hvilken modell som brukes.

| Parameter | Verdi |
|-----------------------------|------------------|
| Image Size (IM_SIZE) | 1850 |
| Number of Epochs (N_EPOCHS) | 30 |
| Batch Size (BATCH_SIZE) | 8 |
| Model File (MODEL) | 'yolov8n-seg.pt' |

Maskinlæringsmodellen blir så brukt videre til annotering av de neste bildene og ble korrigert der den gjorde feil. Annoteringsprosessen akselererer ettersom man trener opp en bedre modell så det ble trent opp nye modeller med jevne mellomrom.

I scriptet ble Comet importert og brukt for monitorering av modellens ytelse. Fra Comet kan det hentes ut et plott over hvor senterpunktene til klassifiseringene fra treningsdataen ligger i bildet. Da Comet viste at mesteparten av klassifiseringene skjedde i venstre halvdel av bildene, ble det supplert med flere annoteringer. I siste modell ble 657 bilder brukt hvor 157 er hentet fra et datasett bestående av bilder tatt i andre kjøreretningen. Det førte til et treningsdatasett som bedre representerte hvordan situasjonen er og en maskinlæringsmodell som er mer robust. Figur 3.12 viser plottet før (venstre) og etter (høyre) det ble supplert med nye bilder. Figuren plottet x- og y-posisjon til senterpunktene og mørkere farge indikerer høyere tetthet av punkter. I og med at stenoverflatene befinner seg på siden av veien gir det mening at flesteparten av senterpunktene fordeler seg på venstre og høyre side av bildet.



Figur 3.12: Diagram over fordeling av senterpunkt i hver klassifisering før og etter supplering med bilder. Høyre bilde viser en høy forbedring i antall klassifiseringer i den manglende delen av bildet.

Figur 3.13 viser scriptet generert av TrainYOLO og som er brukt til siste klassifisering. Scriptet viser oppkobling mot TrainYOLO-tjenesten, uthenting av treningsdata og initiering av modellen. Det kobles også på Comet for monitorering av ytelsen til modellen.

```

Trene YOLOv8 segmenteringsmodell.
Først fylles "apikey" og "project name" inn. Så startes treningen ved å trykke "Runtime -> Run all"
For mer avanserte valg som antall epochs eller endring av basemodell, se i dokumentasjonen.
Når treningen er ferdig, blir den nye modellen lagt til i trainYOLO platformen under prosjekter. Du
kan sammenligne prestasjon med andre versjoner og bruke modellen videre til annotering.
Comet er her lagt til for å gjøre ekstra analyse av resultatene.

# DATASET PARAMETERS

API_KEY = "ed62e84df34f0968355390b0646cbadb97e90569"
PROJECT_NAME = "Steinoverflater"

# YOLO parametere
IM_SIZE = 1850
N_EPOCHS = 30 # you can lower this if you have a large dataset
BATCH_SIZE = 8 # you can increase this if you have a large dataset, > 50 images, you can
MODEL = 'yolov8n-seg.pt'

# installere yolov8
%pip install -q ultralytics
%pip install ultralytics comet_ml torch torchvision
%pip install -q trainyolo-py

import comet_ml
import torch

torch.cuda.empty_cache()

comet_ml.init(project_name="Steinoverflater")

Vis skjulte resultatdata

#kobler til google drive for lagring av resultater

from google.colab import drive
drive.mount('/content/drive')

from trainyolo.client import Client, Project

# oppretter kobling til TrainYOLO

client = Client(API_KEY)

# Laster inn prosjekt

project = Project.get_by_name(client, PROJECT_NAME)
project_location = project.pull(location='./data', format='yolov8')

Vis skjulte resultatdata

# Trene YOLO modell

!yolo task=segment mode=train model=$MODEL data="{project_location}/dataset.yaml"
epochs=$N_EPOCHS imgsz=$IM_SIZE batch=$BATCH_SIZE project=/content/drive/MyDrive/klassifis

```

Figur 3.13: Kodesnutt fra TrainYOLO, med oppkobling til Comet for monitorering av ytelse.

Kapittel 4

Resultater

Dette kapitlet presenterer resultatene for fotogrammetrimodellen og for klassifiseringsalgoritmen. Det vises resultater som har vært viktig i prosessen underveis og de endelige resultatene.

4.1 Evaluering av fotogrammetrisk modell

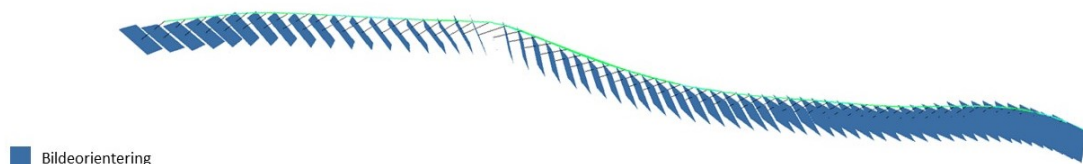
4.1.1 Forbehandling av data

Tabell 4.1 viser Agisoft Metashape sin analyse av bildekvalitet til alle bildene som er brukt til modelleringen. Verdien av parameteren beregnes basert på skarphetsnivået i den delen av bilde som er mest fokusert og ligger mellom spennet på 0-1 (Agisoft, 2021) Den gjennomsnittlige verdien i oppgavens tilfelle ligger på 0.810.

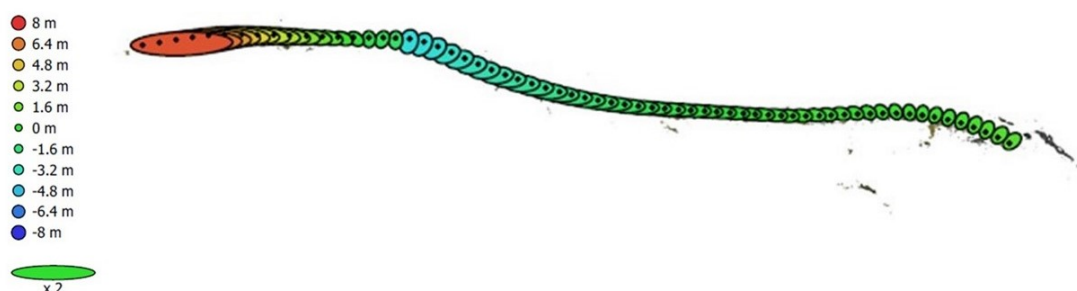
Tabell 4.1: Bildekvalitet for ulike kameraposisjoner

| Kameraposisjon | Bildekvalitet |
|----------------|---------------|
| 1.000 | 0.840 |
| 2.000 | 0.827 |
| 3.000 | 0.808 |
| 4.000 | 0.813 |
| | ⋮ |
| 61.000 | 0.809 |
| 62.000 | 0.816 |
| 63.000 | 0.808 |
| 64.000 | 0.756 |
| Gjennomsnitt | 0.810 |

I figur 4.1 vises alle kameraposisjonene etter kameraorienteringen er gjort. Det gir et overblikk over hvordan kameraene ligger i forhold til hverandre langs strekningen. I figur 4.2 vises kameraposisjonene med feilellipser. Feilellipsene er basert på posisjon innhentet av GPS sammenlignet med estimert posisjon etter kameraorienteringen.

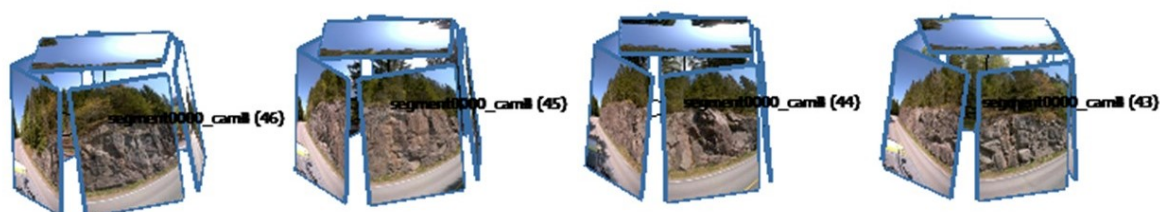


Figur 4.1: Kameraposisjoner og orienteringer etter prosess for kameraorientering er gjennomført.



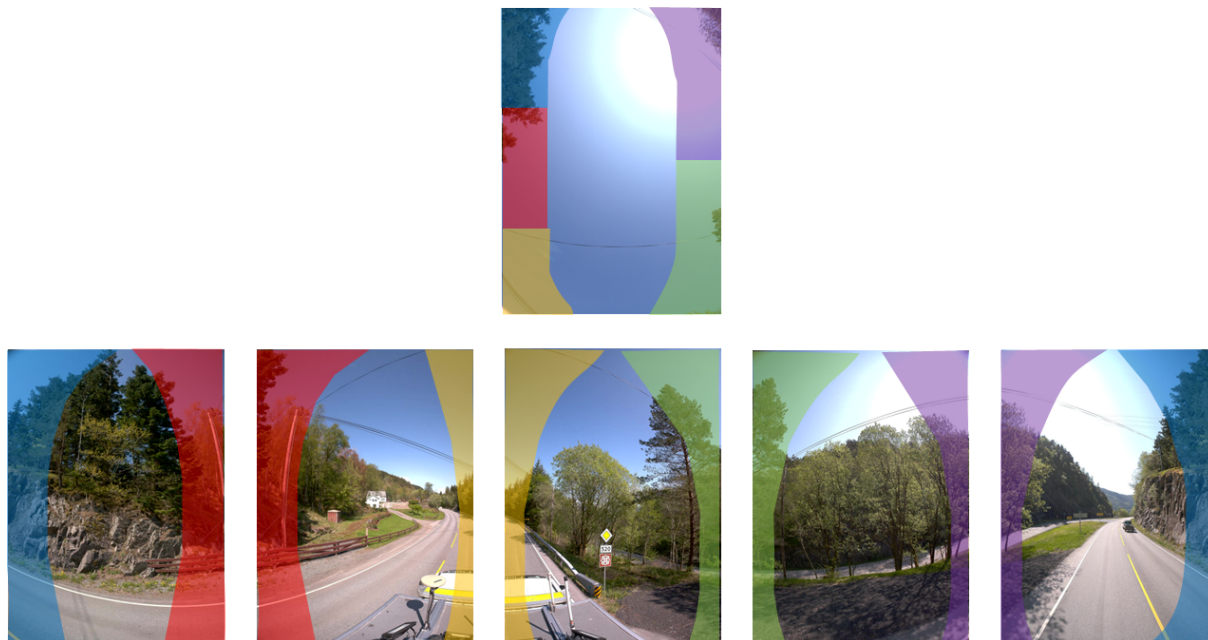
Figur 4.2: Kameraposisjoner med tilhørende feilellipser

Figur 4.3 viser et utsnitt av fire kameraposisjoner for å se hvordan de fire kameraposisjonene er i forhold til hverandre.



Figur 4.3: Bildeorienteringer til kamerasystemet ved fire kameraposisjoner.

Kameraoverlapp er viktig for å gi tilstrekkelig med sammenbinding mellom bildene. Figur 4.4 gir en visuell fremstilling av hvor stor overlappen er mellom bildene i en kameraposisjon. Bildene er posisjonert slik de fremstår i kamerasystemet med ett kamera som peker oppover og fem i sirkel.



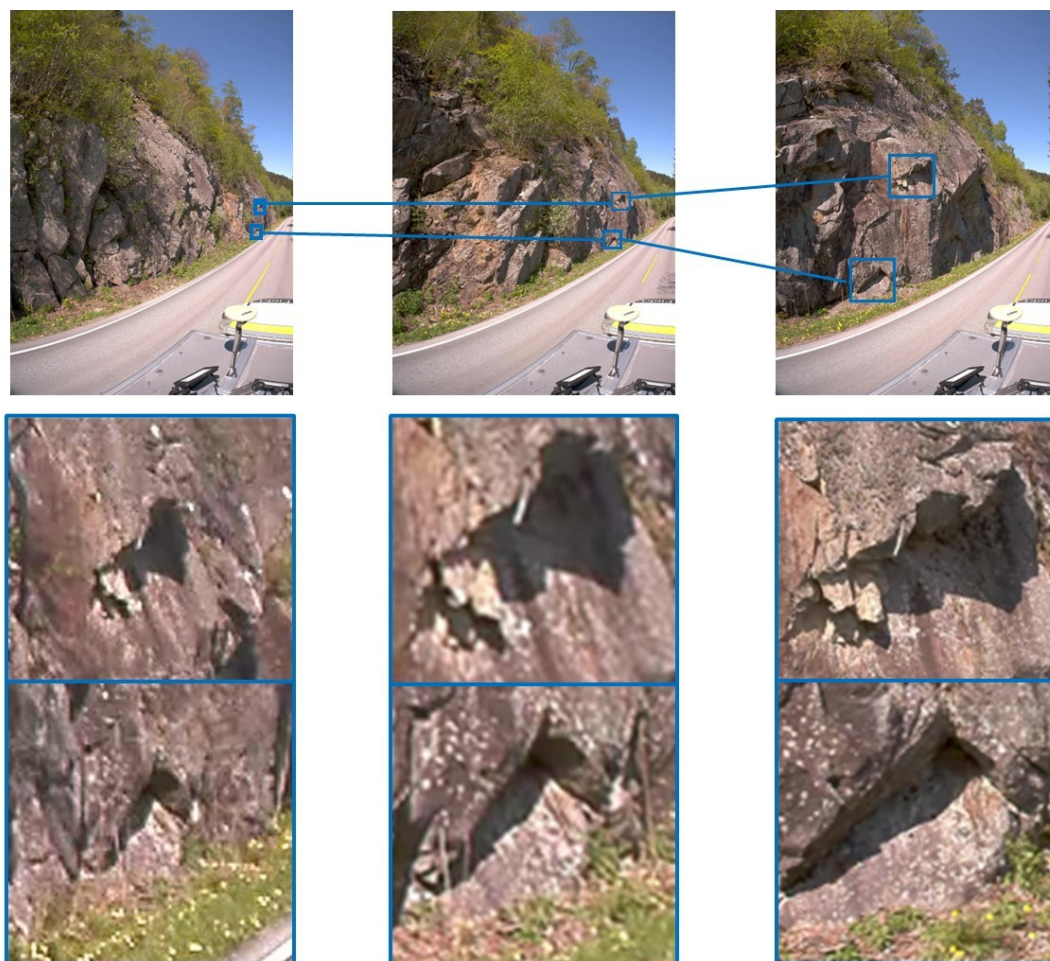
Figur 4.4: Visuell framstilling av overlapp mellom kameraene i kamerasystemet.

I tabell 4.2 er det listet antallet sammenbindingspunkt mellom kameraposisjonene i kameraposisjon 25. Antallet gir et inntrykk av om hver kameraretning har tilstrekkelig sammenbinding med de andre kameraretningene. Tabellen viser at hver kameraretning har tilstrekkelig antall sammenbindingspunkter med minst 3 andre kamera. 5/15 kamerakombinasjoner har ingen sammenbindingspunkt. Antallet er varierende i de ulike kombinasjonene, men skal det være tilstrekkelig nok til at sammenkoblingen mellom bildene i skal kunne gjennomføres.

Tabell 4.2: Sammenbindingspunkter mellom kamerakombinasjoner i kameraposisjon 25

| Kamera kombinasjoner | Sammenbindingspunkt |
|-----------------------------|----------------------------|
| Kamera 1 - Kamera 5 | 125 |
| Kamera 1 - Kamera 6 | 561 |
| Kamera 2 - Kamera 1 | 1313 |
| Kamera 2 - Kamera 5 | 0 |
| Kamera 2 - Kamera 6 | 1949 |
| Kamera 3 - Kamera 1 | 2016 |
| Kamera 3 - Kamera 2 | 5031 |
| Kamera 3 - Kamera 4 | 335 |
| Kamera 3 - Kamera 5 | 0 |
| Kamera 3 - Kamera 6 | 0 |
| Kamera 4 - Kamera 1 | 261 |
| Kamera 4 - Kamera 2 | 0 |
| Kamera 4 - Kamera 5 | 788 |
| Kamera 4 - Kamera 6 | 0 |
| Kamera 5 - Kamera 6 | 611 |

Det er tatt ut tre bilder fra første kameraposisjon for å se på endringen i sammenbindingspunkter mellom de tre posisjonene. De tre bildene vises i figur 4.5 og inneholder blå bokser som tilsvarer to områder som typisk ville blitt brukt i sammenkoblingen. Del to av figuren viser endringen i disse to områdene ettersom kameraposisjonen endres.



Figur 4.5: Visuell framstilling av objekter på tvers av kameraposisjon i posisjon 3.

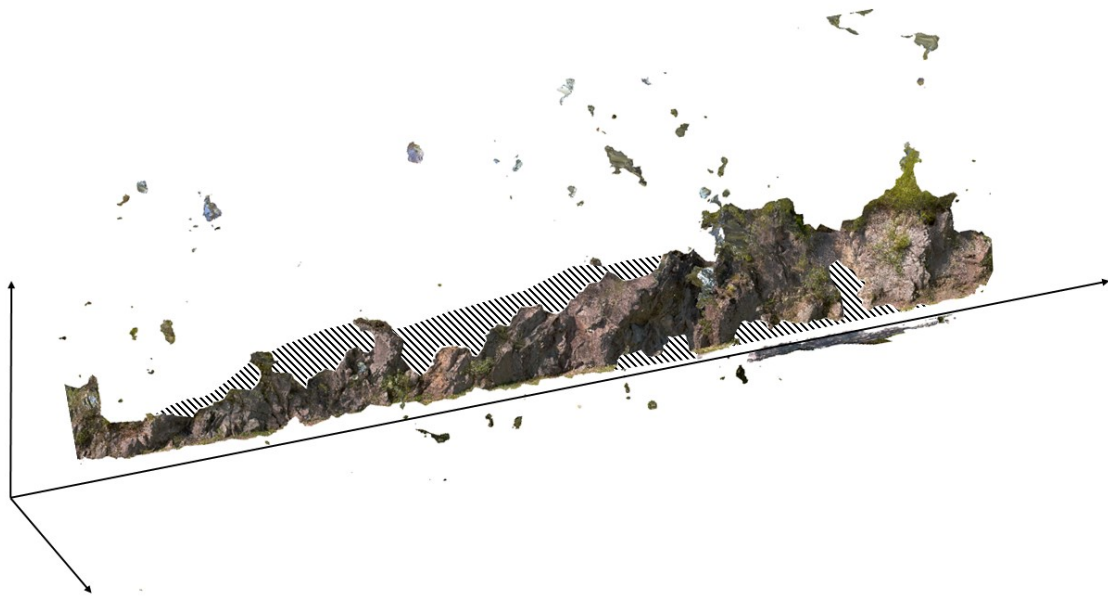
Tabell 4.3 viser antall sammenbindingspunkter som er funnet mellom de tre kameraposisjonene. Den viser det totale antallet og hvor mange av punktene som blir brukt i sammenkoblingen.

Tabell 4.3: Sammenbindingspunkt i tre kameraposisjoner

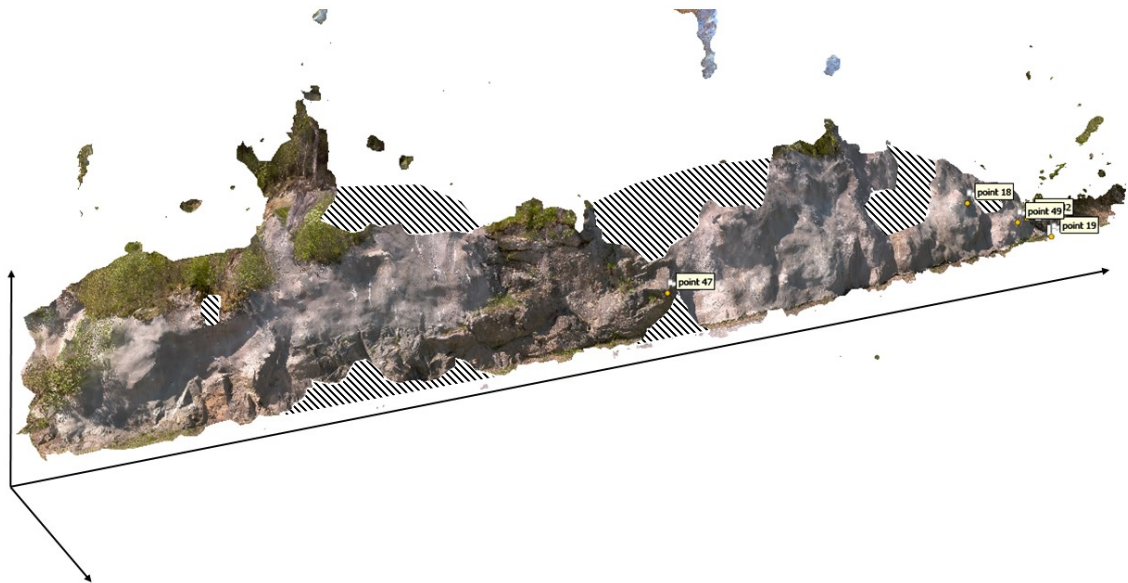
| Bilde | Totalt | Gyldig | Ugyldig |
|-------|--------|--------|---------|
| 25-24 | 655 | 450 | 205 |
| 25-23 | 366 | 159 | 207 |
| 24-23 | 640 | 323 | 317 |

4.1.2 Modellgenerering og -analyse

Tabell 4.4 viser tiden brukt for de forskjellige prosesseringene. Total prosesseringstid for analysen er 10 timer og 4 minutt, men dette avhenger mye av prosessorkraft på datamaskinen.

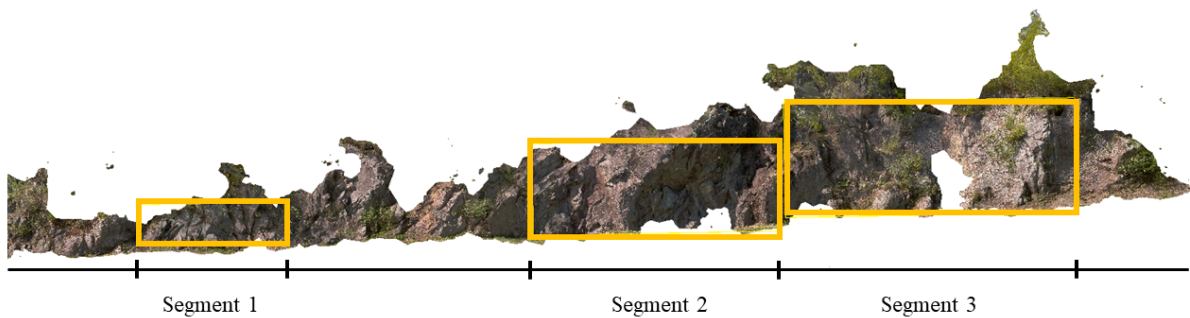


Figur 4.7: Overflatemodell utsnitt 2



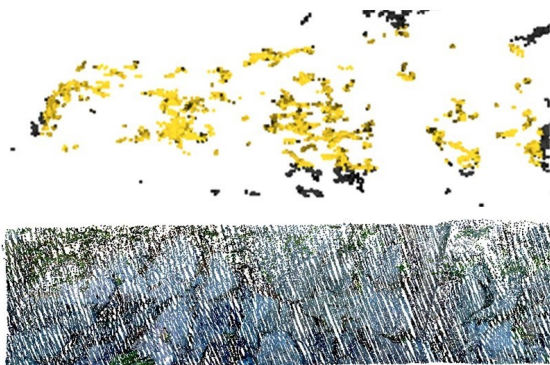
Figur 4.8: Overflatemodell utsnitt 3

Tre segmenter er plukket ut fra områdene der modellen er fullstendig og visuelt ser ut til å ha modellert godt. Segmentene brukes videre til mer detaljert vurdering av nøyaktigheten. De tre segmentene er vist i Figur 4.9

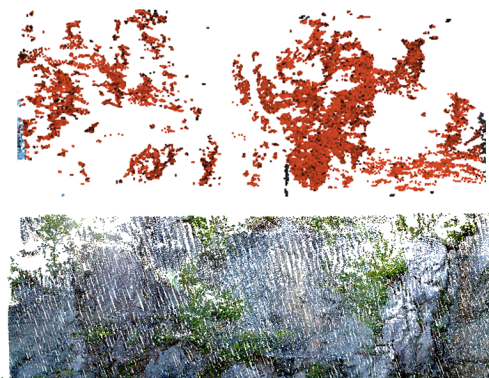


Figur 4.9: Oversiktsbilde over utvalgte segmenter til videre detaljert analyse.

I figur 4.10, 4.11 og 4.12 sammenlignes punktfordelingen i punktskyen fra fotogrammetri (øverst) opp mot punktskyen fra av LiDAR (nederst) til de tre segmentene. Punktskyen fra fotogrammetri har områder som er veldig tette og noen områder som ikke har punkter i det hele tatt. Punktsky fra laserdata har generelt god tetthet og god fordeling. I tabell 4.5 vises antall punkter i de tre segmentene i punktsky fra fotogrammetri og fra laserdata. I høyre kolonne vises forskjellen i antall punkter. Tabellen viser at punktskyen fra laserdata er betydelig tettere enn punktskyen fra fotogrammetri.

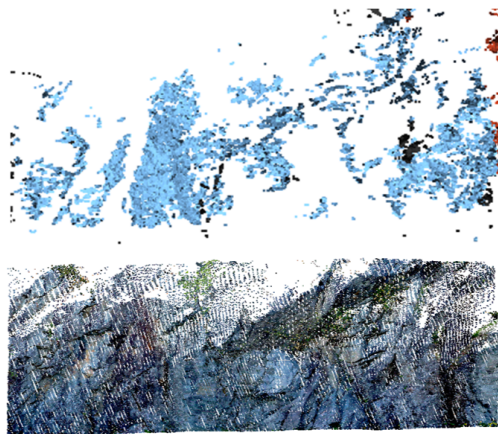


6



10

Figur 4.10: Punktfordeling i punktsky 1 **Figur 4.11:** Punktfordeling i punktsky 2



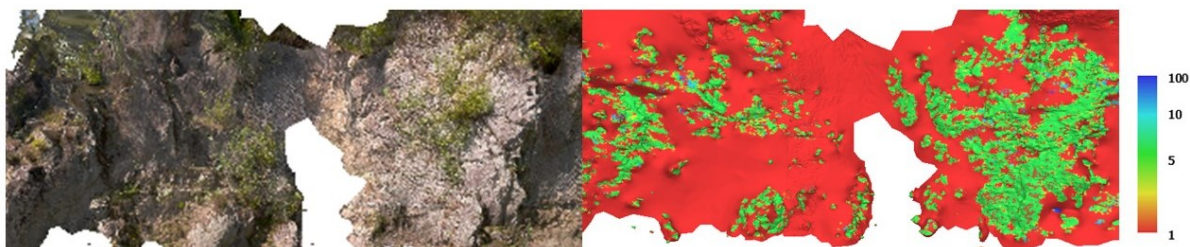
ic

Figur 4.12: Punktfordeling i punkttsky 3

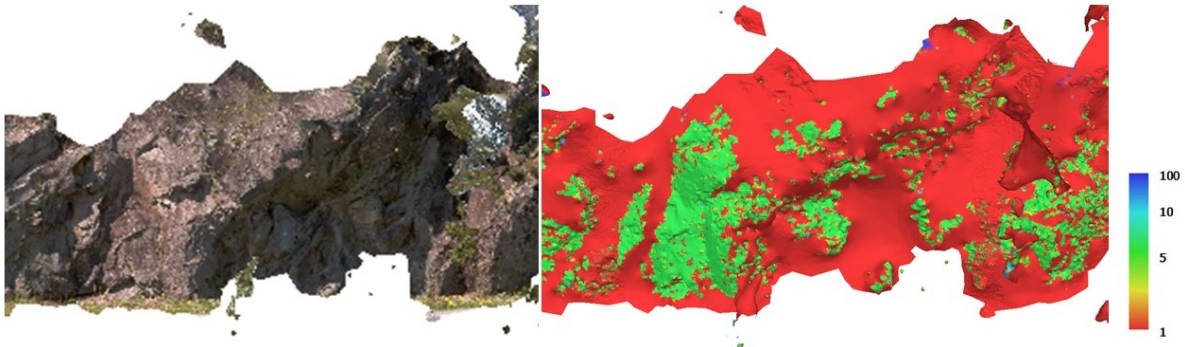
Tabell 4.5: Antall punkter i punkttskyene fra fotogrammetri og laserdata

| Segment | Punker punkttsky fra fotogrammetri | Punker punkttsky laserdata | Δ |
|---------|------------------------------------|----------------------------|----------|
| 1 | 12230 | 134183 | -121953 |
| 2 | 112749 | 345818 | -233069 |
| 3 | 184141 | 290266 | -106125 |

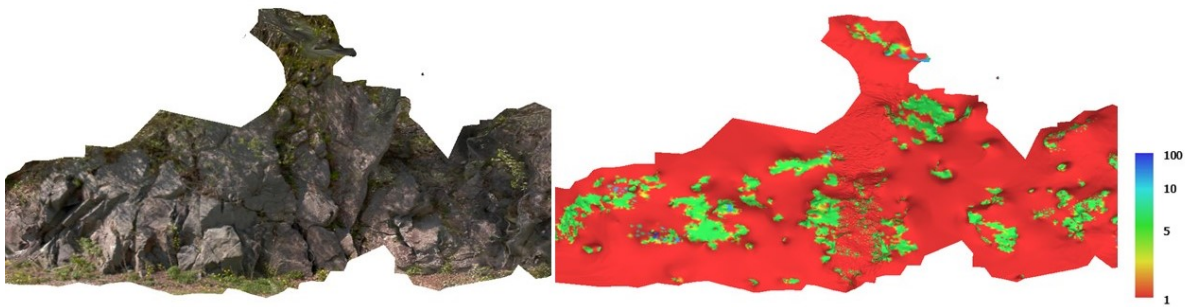
Figur 4.13, 4.14, 4.15 viser konfidenskartene til de 3 segmentene. Konfidenskartene gir et overblikk over av antall bilder punktene i overflatemodellen er synlig i. Flesteparten av områdene har rød/oransje farge som betyr at de observeres i underkant av 3 bilder. Segmentene har noen områder som er oppe i 5 bilder og svært få som er oppe i 10-100 bilder. Det er gjennomgående lave verdier på alle segmentene.



Figur 4.13: Konfidenskart segment 1



Figur 4.14: Konfidenskart segment 2



Figur 4.15: Konfidenskart segment 3

RMS-verdiene i tabell 4.6 og 4.7 er regnet ut med formelen for RMS:

$d = \text{avstand}$

$n = \text{verdier i datasettet}$

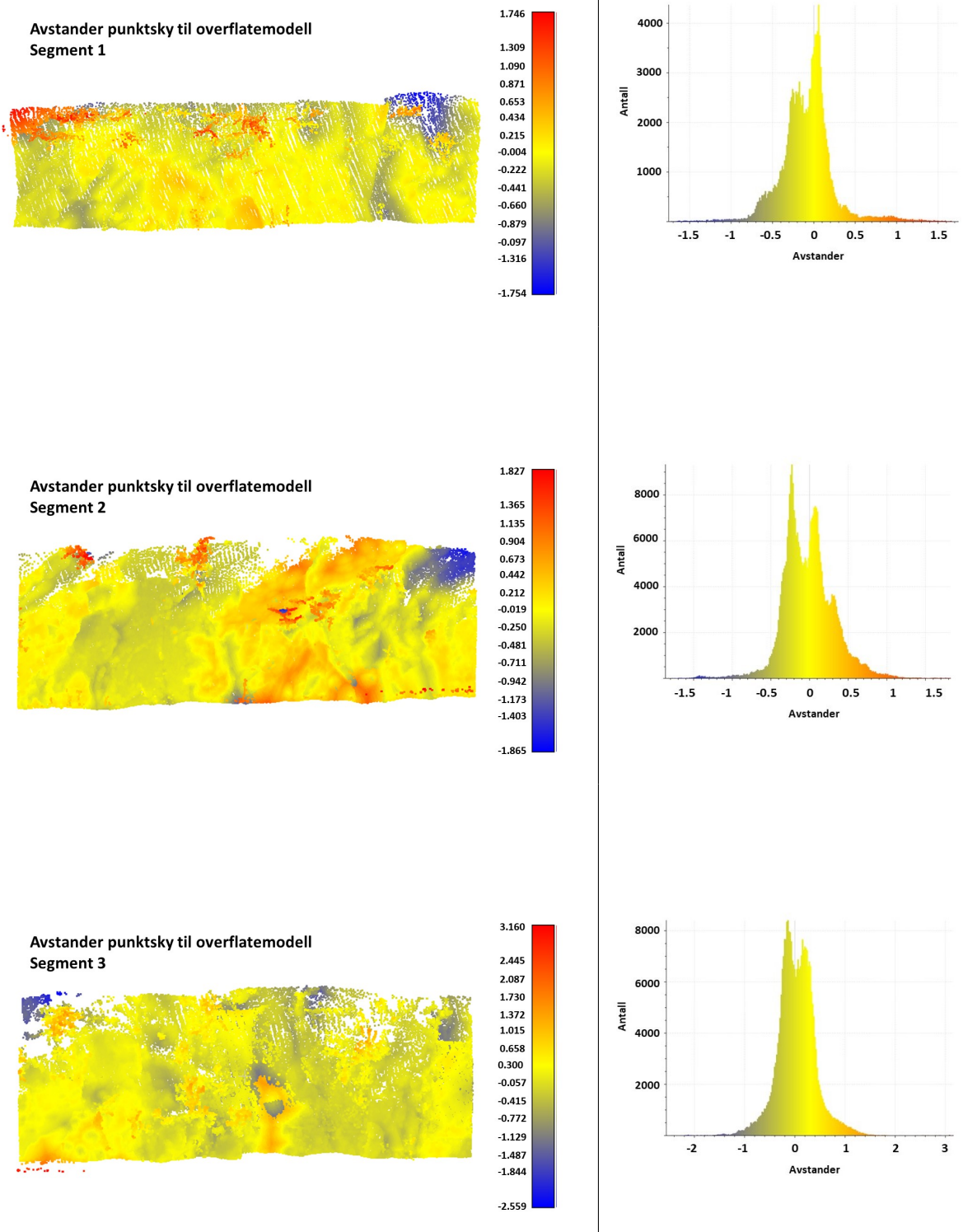
$$\text{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i)^2} \quad (4.1)$$

Tabell 4.6 viser RMS-verdiene til de sju punktene som er brukt til sammenkobling mellom punktsky og overflatemodell. Gjennomsnittlig RMS-verdi er 0.368 som tilsvarer en gjennomsnittlig feil på 36.8 cm.

Tabell 4.6: Usikkerhet i sammenkobling mellom overflatemodell og punktsky

| Punkt | Error |
|-----------|-------|
| A0 | 0.136 |
| A1 | 0.230 |
| A2 | 0.388 |
| A3 | 0.288 |
| A4 | 0.112 |
| A5 | 0.471 |
| A6 | 0.659 |
| A7 | 0.315 |
| R0 | 0.165 |
| R1 | 0.230 |
| R2 | 0.388 |
| R3 | 0.288 |
| R4 | 0.112 |
| R5 | 0.471 |
| R6 | 0.659 |
| R7 | 0.315 |
| RMS | 0.368 |
| Skalering | 1.340 |

Figur 4.16 viser til venstre plot av ”punktsky-til-overflatemodell-avstand”. Fargeskalaen er sentrert rundt fargen gul der modellene er like, fargen rød der punktskyen er foran overflatemodellen og fargen blå der punktskyen er bak modellen. Avstandene er regnet fra hvert punkt i punktskyen til nærmeste triangel i overflatemodellen. 4.7 viser gjennomsnittlig avstand mellom de ulike segmentene og RMS verdi. Venstre segment har en gjennomsnittlig avstand på -0.085 som indikerer at den i gjennomsnitt ligger 8.51 cm innenfor punktskyen. Midtre segment ligger i gjennomsnitt 1.56 cm innenfor punktskyen og høyre segment ligger 3.31 cm på utsiden. 95% av observasjonene ligger respektivt innenfor $\{-72.4 \text{ cm}, 55.3 \text{ cm}\}$, $\{-67.4 \text{ cm}, 64.2 \text{ cm}\}$ og $\{-72.5, 79.1 \text{ cm}\}$.



Figur 4.16: Avstand mellom overflatemodell og punktsky fra laserdata for de tre segmentene utregnet i Cloudcompare. Avstandene er visuelt framstilt med fargeskala fra blå (laserdata innenfor overflatemodell) til rød (laserdata utenfor overflatemodell). Histogrammet viser antall per avstand.

Tabell 4.7: Statistikk for segmentvis avstand mellom overflatemodell og punktsky

| Segment | Gj.snitt avstand (m) | Std. Avvik (m) | RMS |
|---------|----------------------|----------------|-------|
| 1 | -0,085 | 0,319 | 0,331 |
| 2 | -0,016 | 0,329 | 0,329 |
| 3 | 0,033 | 0,379 | 0,381 |

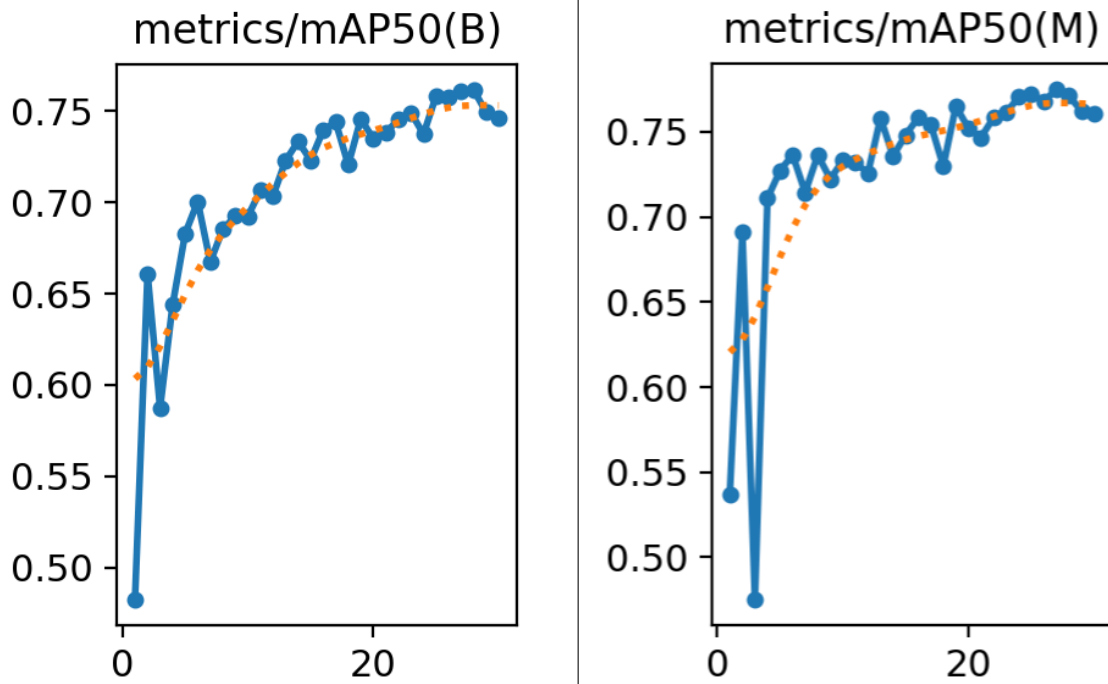
4.2 Evaluering av klassifiseringsmodell

Figur 4.8 framstiller de beste verdiene for evalueringsparameterene for maskinlæringsmodellen. I tabellen er mAP, precision og recall representert med den høyeste verdien under treningen, mens loss viser de laveste verdiene. (B) og (M) betegner om det er boks- eller maskeklassifisering. mAP og accuracy gir generell prestasjon på modellen, mens de andre verdiene går dypere inn på spesifikke deler av modellens ytelse.

Tabell 4.8: Evalueringsparametere for maskinlæringsmodell

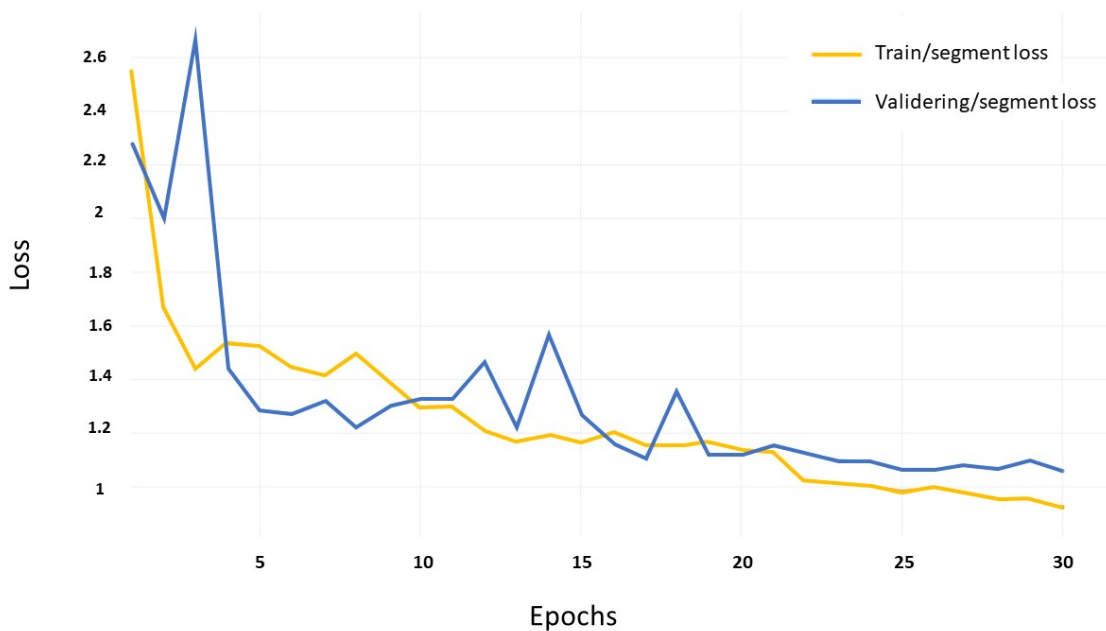
| Evalueringsparameter | Verdi |
|----------------------|-------|
| mAP 50 (B) | 0.761 |
| mAP 50 (M) | 0.774 |
| Accuracy (M) | 0.682 |
| Precision (B) | 0.948 |
| Precision (M) | 0.975 |
| Recall (B) | 0.693 |
| Recall (M) | 0.739 |
| Train Loss (B) | 0.714 |
| Train Loss (M) | 0.924 |
| Test Loss (B) | 0.842 |
| Test Loss (M) | 1.059 |

4.17 viser mAP scoren for boksklassifiseringen og maskeklassifiseringen. mAP-verdien for boksklassifisering starter på omtrent 0,60 og viser en rask forbedring i de første epokene, og når nesten 0.75 rundt epoke 5. Deretter øker mAP gradvis og men stabiliserer seg rundt 0.75.



Figur 4.17: mAP-graf for boks- og maskeklassifisering

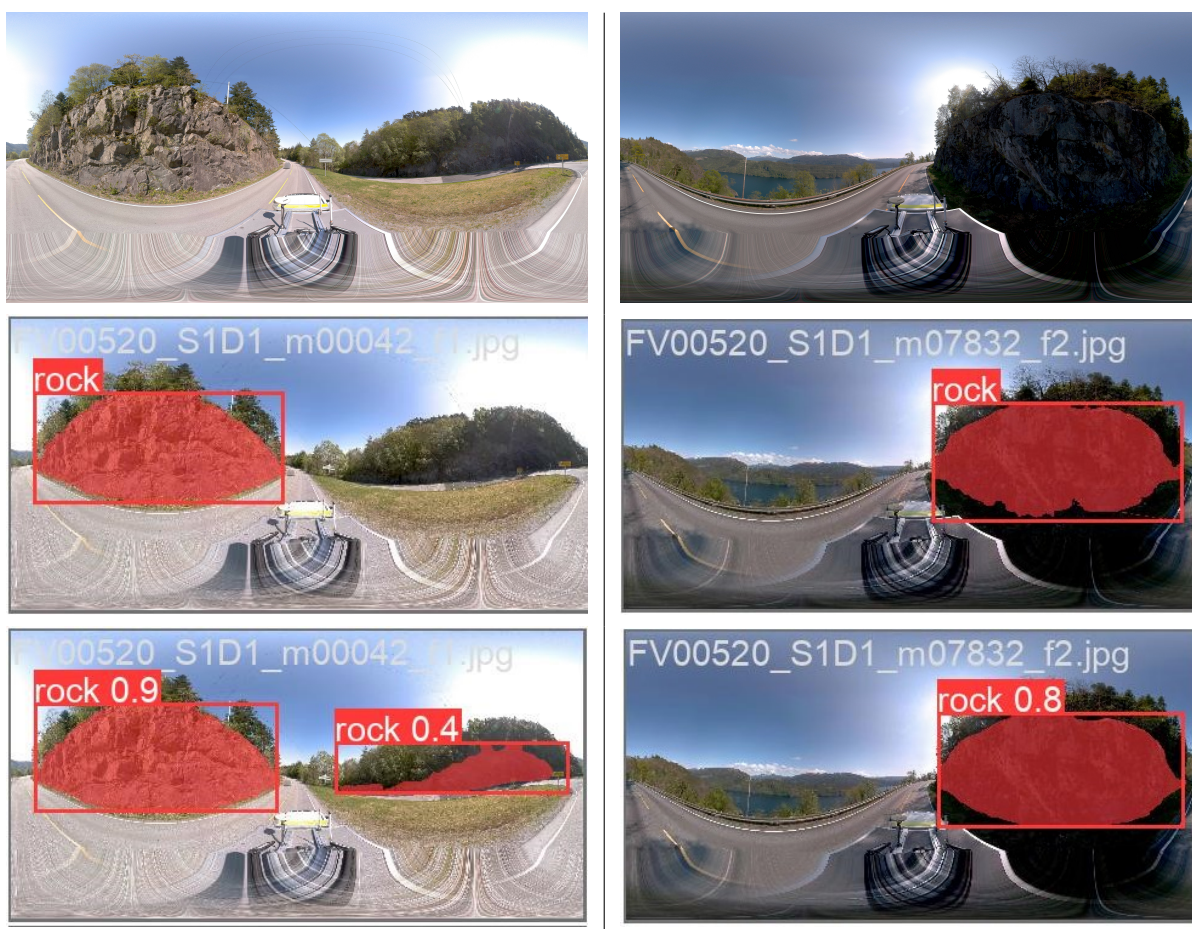
Grafene i 4.18 viser utviklingen i loss i trening og validering for modellen på de ulike epokene. Begge grafene har en naturlig nedgang, med en stabilisering mot slutten. Begge grafene følger hverandre tett og gir en indikasjon på å fortsette noe nedover på siste epoke.



Figur 4.18: Graf over loss for trening og validering til maskinlæringsmodellen per epoke

Figur 4.19 viser to bilder tatt ut fra resultatene fra TrainYOLO-scriptet som tilsvarer

en god klassifisering av steinområdene til sidearealet. I første rad vises råbildene før annotering. I andre rad vises de annoterte bildene som er brukt til treningen. I tredje rad vises prediksjonene maskinlæringsalgoritmen gjør. Det vises avgrensningsboksen som algoritmen predikerer og også pikselvis predikasjon. Tallet i kanten av avgrensningsboksen indikerer hvor sikker maskinlæringsalgoritmen er i prediksjonen sin. Venstre bilde predikerer samme vegskjæring som i annoteringen med en sikkerhet på 90%. Den predikerer også et steinområde som ikke er annotert i treningsdataen, men som med visuell sjekk tydelig er til stede. Bilde til høyre predikerer det naturlige steinområdet som er annotert med en sikkerhet på 80%.



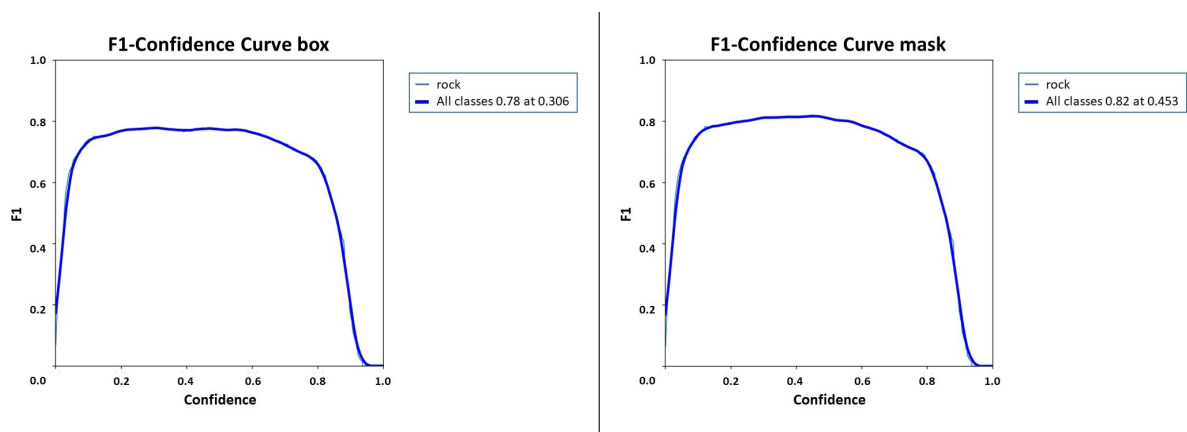
Figur 4.19: Eksempel på riktig klassifisering av maskinlæringsmodellen

Figur 4.20 viser to bilder med mangelfull klassifisering. De annoterte bildene viser markerte steinflater som opptar en liten del av bildet. Det er to eksempler hvor steinområdene ligger langt unna observatøren eller er veldig små. I disse tilfellene viser prediksjonen ingen funn av steinområder.



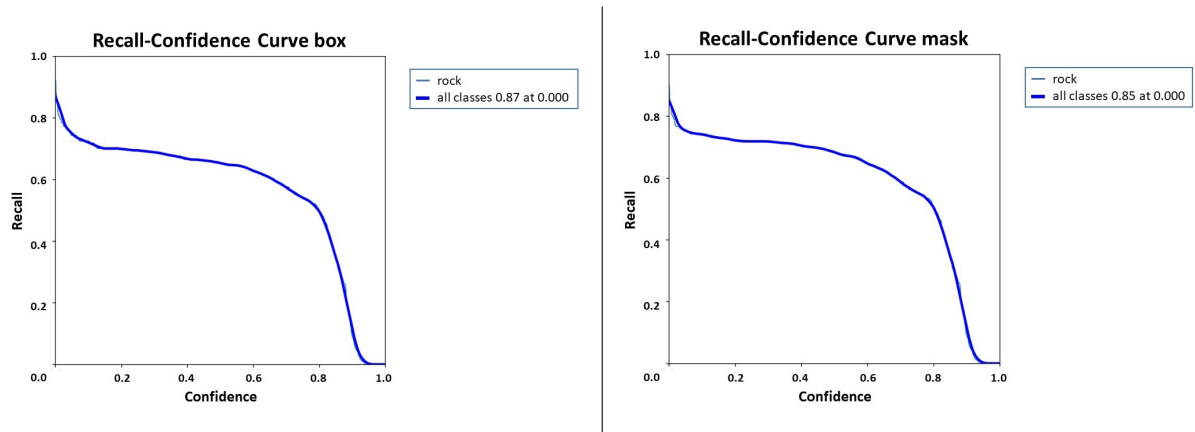
Figur 4.20: Eksempel på feilklassifisering av maskinlæringsmodellen

F1-Confidence kurven 4.21 viser F1 scoren på forskjellige konfidensgrenser. En høy F1-score ved en lavere konfidensgrense er ønskelig, da dette viser at modellen kan forutsi korrekte resultater med selvtillit uten å være for konservativ. F1-kurven til boks- og maskeprediksjonen starter generelt høyt, noe som indikerer god balanse ved lavere terskler, men de begynner å falle bratt ettersom konfidensgrensen øker.



Figur 4.21: F1 - Confidence graf for boksklassifisering (V) og maskeklassifisering (H).

Figur 4.21 viser relasjonen mellom recall- og confidence-terskelen som er brukt for å akseptere prediksjonene. Høye recallverdier betyr at modellen finner mesteparten av sanne positiver. Grafene for både boks og maske har høy recall helt til et stort dropp på 0.8 confidence.



Figur 4.22: Recall - Confidence graf for boksklassifisering (V) og maskeklassifisering (H).

Kapittel 5

Diskusjon

Dette avsnittet forsøker å gi bakgrunn for hvorfor resultatene opptrer som de gjør. Her diskuteres metodene som ble benyttet i undersøkelsen, hvilket bidrar til å forstå både styrkene og begrensningene ved resultatene.

5.1 Vurdering av fotogrammetriske teknikker

Visuell validering av modell

En tidlig indikasjon på modellens nøyaktighet vil være det totale inntrykket av modellen. Utsnittene fra modellen i figur 4.6, 4.7 og 4.8 viser et oversiktsbilde over de modellerte flatene. Et gjennomgående problem med modellen er dens ufullstendighet. Modellen har store hull, som gjør at det ikke trengs videre statistikk for å vurdere at man ikke kan stole helt på resultatet. I tillegg til hullene i modellen er det tydelig at det er områder som er modellert men som fremstår detaljløse.

Sammenligning av modellerte flater

For å se på nøyaktigheten til de modellerte steinoverflatene, ble det tatt ut tre fokusområder vist i 4.9. Fokusområdene ble sammenlignet i cloudcompare opp mot laserdata. Det ble brukt 7 punkter fra overflatemodellen og punktskyen for å sammenstille modellene før sammenligningen. Tabell 4.6 viser til en gjennomsnittlig RMS-verdi mellom punktene som er på 0.368 som tilsvarer en gjennomsnittlig feil på 37 cm. Det viser usikkerheten i sammenstillingen og må ligge til grunn når man sammenligner overflatemodellen og punktskyen.

Sammenligningen av de tre segmentene ble vist ved et Scalefield i figur 4.6, med tilhørende histogram over fordelingen og tabell 4.7 til oppsummering av gjennomsnittlige verdier og standardavvik. Til vurdering av avstandene i modellen skal det ligge til grunn at punktsky fra laserdata har plukket opp vegetasjon i første og andre segment. Dette finner man i de røde områdene i toppen av segmentene. De blå områdene i samme segment har to hull i høyre hjørne som forklarer den lave negative avstanden. Dermed vil segment tre gi den mest troverdige sammenligningen av avstand fra punktsky til overflatemodell. Gjennomsnittlig avstand på 0,033 er ganske nære å være sentrert rundt 0. Det kan tyde på at feilen ikke er systematisk og man kan se litt bort fra RMS-feilen i sammenkoblingen av modellene. Fra gjennomsnittet og standardavviket ble det regnet ut at 95% av avstandene ligger mellom -72.5 cm og 79.1 cm. Sammenlignet med at de fleste steinsprangene som ønskes detektert ligger på under 1 m³ vil en usikkerhet i modellen på 70 cm være kritisk dårlig. Det vil være nødvendig med ytterligere forbedringer for å oppnå pålitelige resultater.

Vurderinger i grunnlagsdata

Med et suboptimalt resultat vil det være nødvendig å avdekke mulige feilkilder og problemer med metoden.

En mulig feilkilde kan være bildene som brukes. Bilder med kvalitetsverdi på mindre enn 0.5 enheter anbefales å deaktiveres og dermed utelukkes fra fotogrammetrisk prosessering, forutsatt at resten av bildene dekker hele scenen som skal rekonstrueres. (Agisoft, 2021). Ut i fra tabell 4.1 var alle bildenes kvalitet godt over kvalitetsverdien på 0.5 enheter så ingen bilder ble utelukket. Bildekvaliteten kan sees bort ifra som potensiell feilkilde til at modellen er ufullstendig.

Antallet bilder utgjør en viktig rolle for modellens nøyaktighet. Mosaic har selv laget en 3D modell av gaten Lužického Semináře i Praha med Mosaic X. Modellen er laget med bildeintervall på ca.1 meter. (Anastasia, 2023). Resultatet gir en indikasjon på at økning i avstanden vil spille en stor rolle i kvaliteten man får ut i fra modellen. Til grunn ligger også samtaler med Mosaic-teamet som ga innsikt i at fire meter mellom bildene var for langt.

I bakgrunn for antagelsen om at økt antall bilder gir økt kvalitet skal man også ha at landskapet som modelleres i de to situasjonene er svært ulikt. I modellen for gaten i Praha er det karakteristiske objekter og harde linjer. Naturlig landskap er vanskeligere å modellere, med mindre karakteristiske objekter og mye vegetasjon. I kapittel 1.1.5

vises det til en studie av Janiszewski et al. (2022) hvor det brukes 360-graders kamera til modellering av innsiden av en tunnel. Her er det brukt 27 kameraposisjoner på en 10 meters strekning, som er en betydelig høyere tetthet av bilder enn det er brukt i dette forsøket. Sammenlignet med laserdata fra samme område ga det en nøyaktighet på 4.6 mm, som gir en indikasjon på effekten av å øke bildetettheten.

Det ble også testet med og uten masker på bildene brukt til modelleringen. Modelleringen uten masker førte til at perspektivet i bildet ble tolket helt feil. Det er blant annet fordi kamerariggen alltid er synlig i bildet, selvom kameraet beveger seg. Det ble vurdert som en nødvendighet å maskere ut elementene beskrevet i kapittel 3.1.6.

Vurderinger i sammenkobling av bildene

For å vurdere om sammenkoblingen av kameraposisjonene har betydning for kvaliteten på modellen, ble det tatt i bruk flere resultater. Det første er feilellipsene til kameraposisjonene i figur 4.2. Disse viser et godt sammenfall mellom estimert posisjon og posisjon fra GPS i nedre halvpart av posisjonene. I øvre del ser man feil på opp til 8 meter. Dette kan være en påvirkning fra den kjente avstanden som ble markert i figur 3.9. Da denne ble brukt til å skalere modellen, kan feilen ha forplantet seg ettersom avstanden ble større til dette punktet.

Tabell 4.2 gir oss en indikasjon på antall sammenbindingspunkter som brukes mellom hver kameraposisjon i kamerasystemet. Alle posisjonene har tilstrekkelig med punkter til å sikre en solid integrasjon. Selv om tabellen kun viser en kameraposisjon, er antallet såpass høyt at det kan antas at et varierende antall punkter vil gi nok dekning i de andre posisjonene også. Det ble også sett på punktene mellom ulike posisjoner til kamerasystemet i tabell 4.3. Fra kameraposisjon 3 har vi i dette tilfellet et relativt høyt antall punkter mellom posisjonene, noe som gir inntrykk av god integrasjon.

Markering av markører var en avgjørende faktor for kameraorienteringen. Det ble som beskrevet i Kapittel 3.1.6 markert 51 markører i bildene. Prosessen med markering av markører er tidkrevende og manuell. Dersom ønsket er et helautomatisk system for steinspranganalyse, vil dette trenge utbedring. At det var nødvendig å markere så mange markører for at kameraorienteringen skulle være riktig kan tyde på to ting. Det ene er at de forskjellige orienteringene til kamerasystemet er vanskelig å tolke for Agisoft Metashape siden de er vinklet forskjellig. Det andre er at det er for få Keypoints som er detektert automatisk mellom kameraposisjonene slik at det trengs manuelt registrerte punkter.

Vurderinger i modellering

Tabell 4.4 viser den totale prosesseringstiden til modellen. Med en prosesseringstid på 10t på et relativt kort strekke er det noe å ta med i vurdering av den praktiske anvendelsen av modellen. Prosesseringstiden er sterkt knyttet til prosessorkraften til maskinen som gjør at det kan kortes ned en del.

For modellering med fotogrammetri lages det en punktsky. Punktskyen basert på fotogrammetri sammenlignes i 4.10, 4.11 og 4.12 med tilhørende data fra LiDAR. Tettheten til punktskyen fra LiDAR er svært mye høyere enn fra fotogrammetri. Hvert segment har klynger med høy tetthet av punkter, og store områder uten noen punkter. Dette er sammensvarende med konfidenskartene som er vist i figur 4.13, 4.14 og 4.15. Her ser vi at klyngene er sett i fem ulike bilder, mens områdene uten punkt er sett i under tre bilder. Disse to plottene viser tydelig problemet med overflatemodellen. Sikkerheten til punktene brukt til å lage overflatemodellen er for lav til å få et godt resultat. En mulig løsning på dette vil være å øke antall bilder og dermed øke sikkerheten til modellen.

5.2 Effektivitet av klassifiseringsmetoder

Det første som må diskuteres er valget av grunnlagsdata for opptrening av maskinlæringsmodellen. Maskinlæringsmodellen er trent på sammensydde 360-graders bilder, mens fotogrammetrimodellen baserer seg på råbildene. Grunnen til dette er en tidsbegrensning siden annotering av bildene ble gjort i forkant av å lage fotogrammetrimodellen. Med lenger tid skulle maskinlæringsmodellen helst vært trent på råbildene også for å gi et mer riktig resultat. Det kan argumenteres for at prestasjonen som sees på de sammensydde bildene også ville blitt oppnådd med opptrening på råbildene ettersom disse har mindre fortegnings og er enklere å tolke for en maskinlæringsmodell.

Annoteringen av bildene er delvis gjort med ren manuell markering. Den manuelle markeringen gir mulighet for menneskelig feil i annotering av bildene. Etter hvert som deler av annoteringen ble gjort av maskinlæringsalgoritmen begrenses den menneskelige feilen. Samtidig får man en feilkilde på at man lettere overser feilklassifiseringer. Ved lite treningsdata er modellen lettere påvirket av slike feilklassifiseringer.

TrainYOLO og Comet ga ut en rekke evalueringsparametere for vurdering av modellens prestasjon. mAP-verdien gir samlet ytelse til modellen og er ifølge tabell 4.8 lik 0.774 for

maskeklassifisering. mAP-verdien kan i dette tilfellet vurderes til god, men med mulighet for en del forbedring. Verdien må vurderes opp mot oppgaven den skal gjennomføre. I oppgavens tilfelle dreier det seg om steinoverflater som i mange bilder er tydelige, men som i en del andre bilder er dekket av vegetasjon og vanskelige å se. I vurdering av resultatet må man også ta hensyn til at mange av de annoterte bildene har en fjellskjæring som opptar store deler av bildet, mens fokusområdet til modellen burde vært på å få med alle små stener også.

Recall og F1-kurvene gir større innsikt i om dette blir oppnådd. Recall-confidencekurven sitt dropp på 0.8 tyder på at modellen har få objekter den er helt sikker på. En mulig løsning for å gjøre modellen mer sikker kan være en økning i datamengden.

En høyeste recall på 0.731 fra tabell 4.8 indikerer at modellen er relativt god til å oppdage positive klasser, men det er fortsatt rom for forbedring. Dette ser man i et praktisk tilfelle i figur 4.20, hvor det er to bilder der modellen ikke oppdager positive klasser. Karakteristisk for disse to bildene er at steinområdene er små eller langt unna.

En grunn til at modellen ikke finner små steinoverflater kan være hvilke parametere som er satt til opptrening av modellen. I tabell 3.4 er `img_size` satt til 1850 x 1850 piksler, hvorav original bildestørrelse er 8000 x 4000 piksler. Som en del av opptreningen gjennomgår bildene en komprimering ned til bestemte `img_size`. En slik komprimering kan være noe av grunnen til at små objekter ikke blir klassifisert.

F1-confidencekurven har et kraftig fall ved 0.8 i confidence. Dette fallet antyder at en økning i terskelen kan utelukke mange korrekte prediksjoner fordi de ikke møter det høye konfidenskriteriet. Ut ifra dette kan vi si at mange av de predikerte objektene er gjort med under 0.8 i confidence. Det kan dermed trekkes en slutning om at mange objekter blir oversett.

Grafen i 4.18 viser loss i treningen av modellen. At både trenings- og valideringstapet stabiliserer seg og følger hverandre tett med lite avstand i mellom seg etter de innledende epokene er et godt tegn. Det tyder på at modellen verken overtilpasser seg eller underpresterer betydelig. Modellen ser ut til å generalisere godt til ny data den ikke har sett under treningen. Både trenings og valideringskurven går fortsatt nedover i siste epoke, som kan tyde på at flere epoker kunne gjort utslag på prestasjonen.

5.3 Forslag til videre forskning

Oppgaven la opp til videre arbeid dersom metoden basert på dagens grunnlagsdata ga lovende resultater. Fotogrammetrimodellens unøyaktighet krever imidlertid at man først styrker den nåværende modellen før videre forskning på området kan utføres. Det vil være nødvendig å primært forbedre grunnlagsdataen til modellen.

Dette vil gjøres ved å redusere bildefrekvensen til maksimalt en meter og teste resultatene ved hjelp av lignende metoder som presentert i oppgaven. Andre studier, som beskrevet i oppgaven, gir en indikasjon på modellens nøyaktighet med bedre datagrunnlag. Det er imidlertid relevant å teste dette i praksis for å bekrefte funnene til dette formålet.

En viktig del av prosessen er at det skal begrense, og i beste fall erstatte den manuelle registreringen som gjøres i dag. For at det skal være en forbedring fra dagens metode er det viktig at metoden har lite manuell analyse som krever kompetanse. Prosessen i oppgaven beskriver spesielt to steg som er tidkrevende og manuelle. Det blir videre viktig å se på å automatisere disse. Det var nødvendig å manuelt markere markører for at sammenkoblingen av kameraene ble riktig. Det var også nødvendig å maskere bildene for riktig sammenkobling. Disse trinnene må sees på å automatiseres.

For at det skal være en brukbar anvendelse til drift og vedlikehold av veien må det settes opp en arbeidsflyt for prosessen. Prosessene beskrevet i denne oppgaven må gjennomføres på flere tidspunkt. Når man har gjennomført flere modelleringer kan det sees på volumendringen mellom modellene. Etter modellene fra ulike tidspunkt er sammenstilt, kan en endringsanalyse gi indikasjon på et steinsprang. Posisjonene der det indikeres et steinsprang integreres så inn i GIS-systemet for registrering av avvik for Drift og vedlikehold.

Når metoden ansees som god nok basert på Mosaic X blir videre arbeid å se på resterende punkter i fullstendig prosess beskrevet i 1.2:

1. Hvordan er nøyaktigheten til en fotogrammetrimodell fra en rimeligere 360-graders kamera sammenlignet med fotogrammetrimodellen basert på eksisterende data?
2. Hvordan presterer klassifiseringsmetoden på et rimeligere 360-graders kamera?

Kapittel 6

Konklusjon

Formålet med analysen var et første steg i en større prosess med visjon om tidligere å detektere steinsprang langs veien. Analysen skulle fokusere på disse to spørsmålene:

1. Er nøyaktigheten til en fotogrammetrimodell basert på 360-graders bilder fra eksisterende data tilstrekkelig nok til å detektere volumendringen i et steinsprang?
2. Hva er prestasjonen til maskinlæringsmodellen for klassifisering av steinoverflater i 360-graders bilder?

Til å svare på første problemstilling ble det tatt i bruk bilder tatt fra 360-graders kameraet Mosaic X. Kameraet ansees som et høykvalitetskamera og er laget for modellerer av denne typen. Bildene fra kamera ble brukt til å lage en fotogrammetrimodell langs strekningen Ropeid - Sauda. Strekningen består av områder som historisk er utsatt for steinsprang. Det overordnede inntrykket av modellen ble først analysert. Modellen fremstår som ufullstendig med store områder som ikke har blitt modellert. De områdene som ble modellert ble videre analysert ved å sammenligne de med laserdata fra samme område. Her ble det regnet ut avstanden mellom overflatemodellen og punktskyen fra laserdata. Usikkerheten i modellens nøyaktighet er for stor sammenlignet med størrelsen på endringene som oppstår når et steinsprang forekommer.

Videre ble det forsøkt å inspisere hva som var grunnlaget for den unøyaktige modellen. Datagrunnlaget og metoden ble gått igjennom og vurdert. Det ble utelukket av bildekvaliteten og selve posisjoneringen av kameraene ikke var kilden til problemet. En plausibel kilde til problemet som forstatt ikke er utelukket er antallet bilder langs strekningen. Om en høyere frekvens av bilder er avgjørende faktor kan støttes opp mot tidligere modellering

Mosaic har gjort i Praha. Med bildefrekvens på en meter ble nøyaktigheten betydelig høyere enn det som ble oppnådd i oppgavens modell.

Til å svare på andre problemstilling ble det trent opp en maskinlæringsmodell på 360-graders bilder med hensikt å klassifisere steinområder langs veien. Modellens nøyaktighet ble vurdert ved hjelp av TrainYOLO og Comet sitt analyseverktøy. Maskinlæringsmodellens evalueringparametere indikerer at modellen presterer godt til å være trent på et relativt lite datasett. Resultatene tyder også på at prestasjonen og selvsikkerheten vil øke med et større datasett. Ved videre vurdering av resultatene ser man at modellen presterer veldig godt på store vegskjæringer som opptar en stor del av bildet. Steinoverflater som er lenger unna og opptar en mindre del av bildet blir ofte ikke klassifisert av modellen. Resultatet er ikke helt representativt, ettersom klassifiseringen av de små steinoverflatene er det viktigste i denne sammenhengen. Når modellen gjør det dårlig på små steinoverflater, påvirker det totalprestasjonen mindre enn klassifiseringen av de store vegskjæringene. Dette skyldes at de små steinoverflatene utgjør en liten del av det totale antallet piksler som blir klassifisert. Til tross for dette er klassifiseringen av små steinoverflater svært viktig, ettersom de er vanskeligst å detektere manuelt.

Referanser

- Agisoft. (n.d.). Working with masks [Hentet: 26-04-2024]. <https://agisoft.freshdesk.com/support/solutions/articles/31000153479-working-with-masks>
- Agisoft. (2021). Agisoft metashape user manual professional edition, version 1.7 [Hentet: 26-04-2024]. https://www.agisoft.com/pdf/metashape-pro_1_7_en.pdf
- Agisoft LLC. (2023). Agisoft metashape: Professional photogrammetry software [Hentet: 08-05-2023]. <https://www.agisoft.com/>
- Ali, S. Z. (2023). Principles of yolov8 [Accessed: 2024-05-15]. <https://medium.com/@syedzahidali969/principles-of-yolov8-6a90564e16c3>
- Anastasia. (2023). Mosaic x: Capturing high-resolution images of infrastructure, roadside assets, and utility faster and cheaper [Hentet: 06-05-2024]. <https://www.mosaic51.com/industry/mosaic-x-capturing-high-resolution-images-of-infrastructure-roadside-assets-and-utility-faster-and-cheaper/>
- Austrheim, G., Hjelle, K., og Kathrine Stene, P. S., & Tretvik, A. M. (2015). Fjelleets kulturlandskap [Tilgjengelig fra NTNU, Hentet: 13-05-2024].
- Avatour. (2024). How do 360-degree cameras work? [Hentet: 12-05-2024]. *Avatour*. <https://avatour.com/article/how-do-360-degree-cameras-work>
- Bjordal, H., & Helle, T. E. (2011). Skred og flom på veg [VD rapport, Hentet: 13-05-2024]. <https://vegvesen.brage.unit.no/vegvesen-xmlui/bitstream/handle/11250/2577089/VD%20%20Skred%20og%20flom%20p%C3%A5%20veg.pdf?sequence=1&isAllowed=y>
- Brinkmann, R. (2008). *The art and science of digital compositing: Techniques for visual effects, animation and motion graphics* [Hentet: 13-05-2024]. Morgan Kaufmann. <https://doi.org/10.1016/B978-0-12-370638-6.00002-X>
- CloudCompare. (2023). Cloudcompare: 3d point cloud and mesh processing software [Hentet: 08-05-2023]. <https://www.cloudcompare.org/>

- Colab, G. (n.d.). Google colabory [Hentet: 01-05-2024]. <https://colab.research.google.com/>
- Digiarty Software. (2023). Videoproc converter ai: Video processing software [Hentet: 08-05-2023]. <https://www.videoproc.com/>
- Erium, V. (2019). Alignment experiments (from agisoft forum. by marcel) [Hentet: 12-05-2024]. <https://madpro.medium.com/alignment-experiments-from-agisoft-forum-by-marcel-19b2c7727d7a>
- Farmakis, I., DiFrancesco, P.-M., Hutchinson, D. J., & Vlachopoulos, N. (2022). Rockfall detection using lidar and deep learning [Hentet: 14-05-2024]. <https://doi.org/10.1016/j.enggeo.2022.106836>
- Hafiz, A. M., & Bhat, G. M. (2020). A survey on instance segmentation: State of the art [Hentet: 13-05-2024]. *International Journal of Multimedia Information Retrieval*, 9(3), 171–189. <https://doi.org/10.48550/arXiv.2007.00047>
- Harvey, P. (n.d.). Exiftool by phil harvey [Hentet: 13-05-2024]. <https://exiftool.org/>
- Hexagon. (2024). *Velodyne puck* [Hentet: 12-05-2024]. <https://autonomoustuff.com/products/velodyne-puck-ulp-16>
- Hui, J. (2018). *Map (mean average precision) for object detection* [Hentet: 12-05-2024]. <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>
- Huilgol, P. (2024). *Precision and recall - essential metrics for machine learning* [Hentet: 12-05-2024]. <https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/>
- IBM. (n.d.). What are convolutional neural networks? [Accessed: 2024-05-15]. <https://www.ibm.com/topics/convolutional-neural-networks>
- iSi Insight. (2023). *Isi insight digital rekkvervskontroll - første leveranser gjennomført* [Hentet: 06-05-2024]. <https://isi.no/project/isiinsight.leveranser>
- Janiszewski, M., Torkan, M., Uotinen, L., & Rinne, M. (2022). Rapid photogrammetry with a 360-degree camera for tunnel mapping [Hentet: 14-05-2024]. <https://doi.org/10.3390/rs14215494>
- Mosaic. (2023). *High precision 360 camera for panoramic street view, photogrammetry, og 3d scanning* [Specifications for Mosaic X, provided by Mosaic, Hentet: 13-05-2024]. <https://www.mosaic51.com/cameras/mosaic-x/?playlist=72dd746&video=1cf8024>

- NGU. (2021). Skred og ras [Hentet: 13-05-2024]. <https://www.ngu.no/geologi-og-risiko/skred-og-ras>
- Norge i Bilder. (2024). Norge i bilder [Accessed: 2024-05-15]. <https://www.norgebilder.no>
- NVE. (2018). Vedlegg 1 - steinsprang og steinskred: Bransjestandard for kartlegging av fare for steinsprang og steinskred [Høringsdokument NVE, Hentet: 13-05-2024]. <https://www.nve.no/media/8357/vedlegg-1-steinsprang-og-steinskred.pdf>
- NVE. (2021). Steg 1: Er steinsprang aktuell prosess i området [Hentet: 13.05.2024]. <https://veileder-skredfareutredning-bratt-terreng.nve.no/hvordan-utfore-en-skredfareutredning/fase-2-utfore-oppdrag/prosedyre-steinsprang/steg-1-er-steinsprang-aktuell-prosess-i-området/>
- Python Software Foundation. (2023). Python: A dynamic, open source programming language [Hentet: 08-05-2023]. <https://www.python.org/>
- Raschka, S., & Mirjalili, V. (2019). *Python machine learning: Machine learning and deep learning with python, scikit-learn, and tensorflow 2, 3rd edition* [Hentet: 13-05-2024]. Packt Publishing. <https://books.google.no/books?id=n1cJyAEACAAJ>
- Ringnes, T. (2023). Parallaxe [Hentet: 06-05-2024]. *SNL*. <https://snl.no/parallaxe>
- Schenk, T. (2005). Introduction to photogrammetry [Hentet: 13-05-2024]. *The Ohio State University, Columbus, 106*(1). <https://www.mat.uc.pt/~gil/downloads/IntroPhoto.pdf>
- Semyonov, D. (2011). Algorithms used in photoscan [Hentet: 30-04-2024]. <https://www.agisoft.com/forum/index.php?topic=89.msg323#msg323>
- SGO. (2023). Mistika vr: Virtual reality stitching software [Hentet: 08-05-2023]. <https://www.sgo.es/mistika-vr/>
- Shahab, M. S. (2023, October). YOLO v8 [Hentet: 12-05-2024]. <https://medium.com/@muhammadshabrozshahab/yolo-v8-104f1375242c>
- Sharma, N. (2023). Understanding and applying f1 score: Ai evaluation essentials with hands-on coding example [Hentet: 12-05-2024]. <https://arize.com/blog-course/f1-score/#:~:text=F1%20score%20is%20a%20measure,can%20be%20modified%20into%20F0.>
- Singh, A. (2024). *Sift algorithm: How to use sift for image matching in python* [Hentet: 12-05-2024]. <https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/>
- Statens Vegvesen, Politiet, Helsedirektoratet, Utdanningsdirektoratet, Trygg Trafikk, fylkeskommunene, & storbykommunene. (2021). Nasjonal tiltaksplan for trafikksikker-

- het på vei 2022–2025 - på vei mot nullvisjonen [Hentet: 13-05-2024]. <https://www.regjeringen.no/contentassets/c91632e1e2a84454b72072c5d51bf517/nasjonal-tiltaksplan-for-ts-pa-vei-2022-2025-endelig.pdf>
- TrainYOLO Developers. (2023). Trainyolo: Object detection training framework [Hentet: 08-05-2023]. <https://trainyolo.com/>
- Univrse. (n.d.). Univrse 3dai city [Hentet: 01-05-2024]. <https://univrse.com/city>
- Vegdirektoratet. (2011). Sikring av veger mot steinskred [VD rapport, Hentet: 13-05-2024]. <http://hdl.handle.net/11250/2582754>
- Vegdirektoratet. (2014). *Standard for drift og vedlikehold av riksveger, håndbok r610* (tech. rep.) (Hentet: 13-05-2024). <https://www.vegvesen.no/globalassets/fag/handboker/hb-r610.pdf>
- W3schools. (n.d.). Python introduction [Hentet: 26-04-2024]. https://www.w3schools.com/python/python_intro.asp
- Zoumpikas, T., Puig, A., Salamó, M., García-Sellés, D., Blanco Nuñez, L., & Guinau, M. (2021). An intelligent framework for end-to-end rockfall detection [Hentet: 06-05-2024]. *International Journal of Intelligent Systems*, 36(11), 6471–6502. <https://doi.org/10.1002/int.22557>

Kapittel 7

Vedlegg

7.0.1 Kode for strukturering av GPX-fil til innkoding av EXIF-data til bildene

```
In [2]: # %pip install pillow
# %pip install exif
# %pip install piexif
```

kode for strukturering av GPX-fil i en lesbar txt-fil til innkoding av EXIF-data til bildene

```
In [1]: from exif import Image
import os, json
import pandas as pd
import csv
import glob

import os
# Lager liste med alle filer i mappen
path = r"C:\Users\isak\bilder_master\Nye_bilder_field\segment0000_cam0"

dir_list = os.listdir(path)
leng= len(dir_list)
#print(len(dir_list))

with open('file_route6.txt', 'w', newline='') as file:
    writer = csv.writer(file, delimiter=',')
    field = ["SourceFile", "GPSLatitude", "GPSLongitude", "GPSAltitude"]
    writer.writerow(field)
    with open('route_frame_synced.gpx') as route_file:
        counter=0
        for line in route_file:
            if (line.startswith('<trkpt')) and counter < leng:
                y=line.split(" ")
                z=line.split("<")
                #print(z)
                #print(counter)
                Lat= z[1].split(" ")[1].split("\")[1]
                #print(Lat)
                Long= z[1].split(" ")[2].split("\")[1]
                ele=z[16][4:]
                #print(ele)
                frame = "".join([ele for ele in (z[19]) if ele.isdigit()])
                writer.writerow([dir_list[counter],Lat, Long,ele])
                counter = counter+1
```

Åpner cmd.exe og navigerer til mappen med txt-filen med filnavn og tilknyttet koordinat

Bruker exiftool til å lese txt-fila og kode inn hvert enkelt bilde med riktig exif-data

```
In [ ]: exiftool -csv=file_route1.txt -gpslatituderef=N -gpslongituderef=W -gpsaltituderef=
```

7.0.2 Rapport Agisoft Metashape

Rapport mesh3

Agisoft rapport

27 March 2024



Survey Data

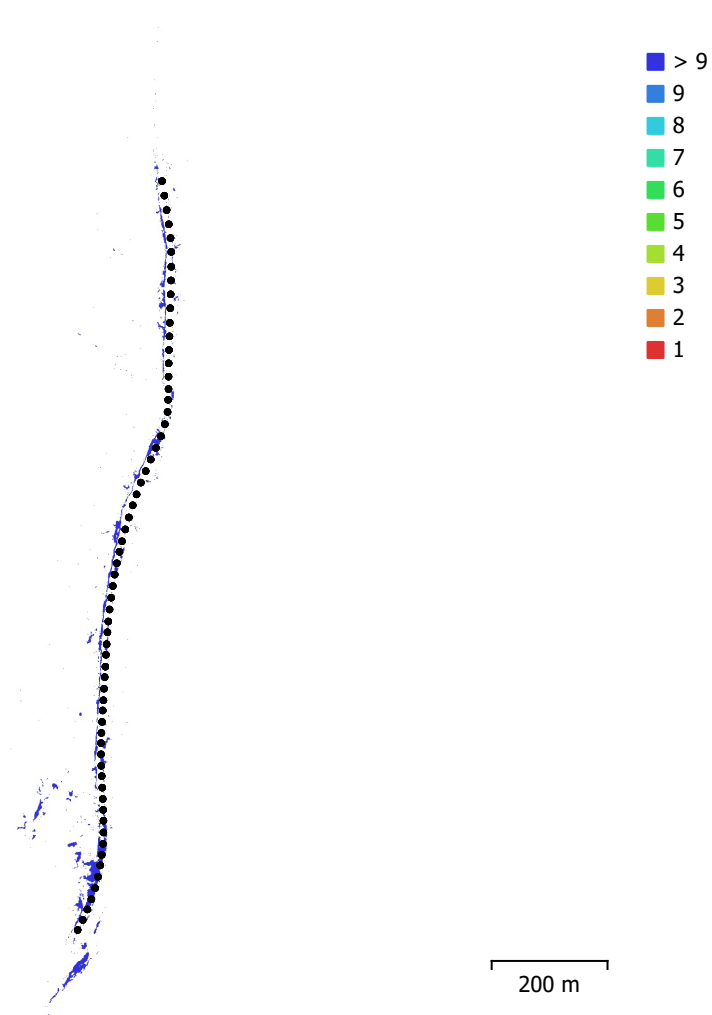


Fig. 1. Camera locations and image overlap.

| | | | |
|--------------------|------------------------|---------------------|-----------|
| Number of images: | 384 | Camera stations: | 384 |
| Flying altitude: | 49.1 m | Tie points: | 855,242 |
| Ground resolution: | 1.64 cm/pix | Projections: | 1,769,775 |
| Coverage area: | 8.4e+03 m ² | Reprojection error: | 3.02 pix |

| Camera Model | Resolution | Focal Length | Pixel Size | Precalibrated |
|--------------|-------------|--------------|---------------------------|---------------|
| Kamera 1 | 4096 x 3008 | 5.5 mm | 2.74 x 2.74 μm | No |
| Kamera 2 | 4096 x 3008 | 5.5 mm | 2.74 x 2.74 μm | No |
| Kamera 3 | 4096 x 3008 | 5.5 mm | 2.74 x 2.74 μm | No |
| Kamera 4 | 4096 x 3008 | 5.5 mm | 2.74 x 2.74 μm | No |
| Kamera 5 | 4096 x 3008 | 5.5 mm | 2.74 x 2.74 μm | No |

| Camera Model | Resolution | Focal Length | Pixel Size | Precalibrated |
|---------------------|-------------------|---------------------|---------------------------|----------------------|
| Kamera 6 | 4096 x 3008 | 5.5 mm | 2.74 x 2.74 μm | No |

Table 1. Cameras.

Camera Calibration

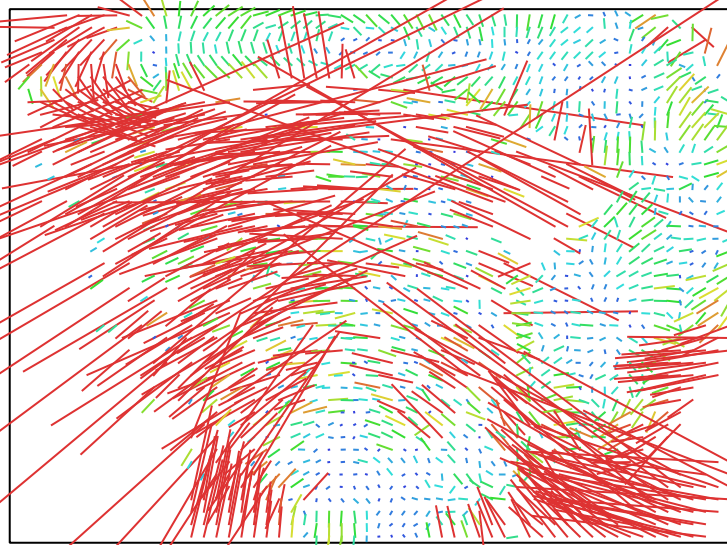


Fig. 2. Image residuals for Kamera 1.

Kamera 1

64 images

Type

Resolution

Focal Length

Pixel Size

Frame

4096 x 3008

5.5 mm

2.74 x 2.74 μm

| | Value | Error | F | Cx | Cy | B1 | B2 | K1 | K2 | K3 | K4 | P1 | P2 |
|-----------|--------------------|---------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| F | 2141.3 | 0.25 | 1.00 | 0.14 | -0.22 | -0.29 | 0.18 | -0.71 | 0.67 | -0.63 | 0.60 | -0.30 | 0.46 |
| Cx | 7.90976 | 0.28 | | 1.00 | -0.14 | -0.10 | 0.15 | -0.11 | 0.12 | -0.12 | 0.13 | -0.21 | 0.08 |
| Cy | 156.23 | 0.42 | | | 1.00 | 0.08 | -0.14 | 0.24 | -0.27 | 0.28 | -0.28 | 0.27 | -0.59 |
| B1 | 21.6855 | 0.15 | | | | 1.00 | 0.14 | -0.01 | -0.02 | 0.02 | -0.03 | 0.07 | -0.29 |
| B2 | -3.63391 | 0.083 | | | | | 1.00 | -0.16 | 0.14 | -0.12 | 0.11 | 0.06 | 0.08 |
| K1 | -0.289626 | 0.00016 | | | | | | 1.00 | -0.98 | 0.96 | -0.92 | 0.12 | -0.18 |
| K2 | 0.0998913 | 0.00016 | | | | | | | 1.00 | -0.99 | 0.97 | -0.14 | 0.23 |
| K3 | -0.0214275 | 6.3e-05 | | | | | | | | 1.00 | -0.99 | 0.14 | -0.24 |
| K4 | 0.00193592 | 8.5e-06 | | | | | | | | | 1.00 | -0.14 | 0.24 |
| P1 | -0.00118356 | 3.6e-06 | | | | | | | | | | 1.00 | -0.35 |
| P2 | 0.000289805 | 1.1e-05 | | | | | | | | | | | 1.00 |

Table 2. Calibration coefficients and correlation matrix.

Camera Calibration

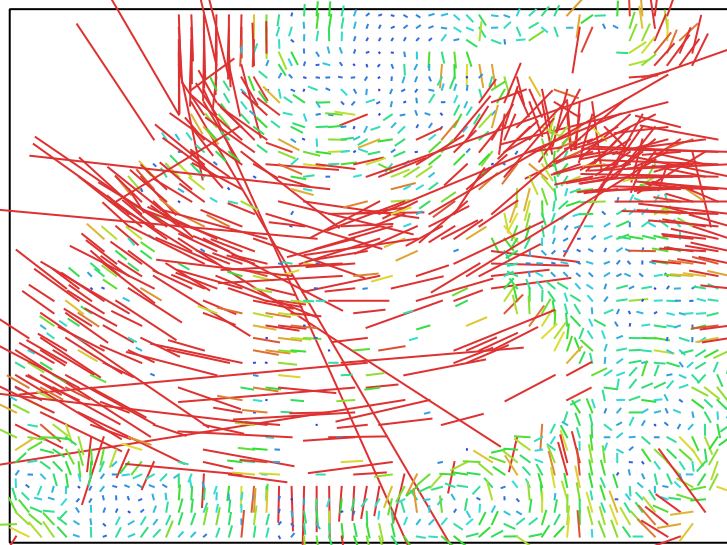


Fig. 3. Image residuals for Kamera 2.

Kamera 2

64 images

| | | | |
|--------------|--------------------|---------------|---|
| Type | Resolution | Focal Length | Pixel Size |
| Frame | 4096 x 3008 | 5.5 mm | 2.74 x 2.74 μm |

| | Value | Error | F | Cx | Cy | B1 | B2 | K1 | K2 | K3 | K4 | P1 | P2 |
|-----------|---------------------|---------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| F | 2114.01 | 0.28 | 1.00 | 0.29 | 0.16 | -0.24 | -0.13 | -0.71 | 0.64 | -0.59 | 0.55 | -0.21 | -0.29 |
| Cx | -101.494 | 0.34 | | 1.00 | 0.10 | -0.12 | 0.03 | -0.13 | 0.10 | -0.08 | 0.07 | -0.33 | -0.14 |
| Cy | 43.0553 | 0.41 | | | 1.00 | 0.02 | -0.07 | -0.10 | 0.09 | -0.07 | 0.06 | -0.16 | -0.55 |
| B1 | 24.8843 | 0.14 | | | | 1.00 | -0.05 | -0.01 | -0.01 | 0.01 | -0.01 | 0.08 | 0.20 |
| B2 | 0.733501 | 0.1 | | | | | 1.00 | 0.04 | -0.04 | 0.03 | -0.03 | -0.24 | 0.01 |
| K1 | -0.271055 | 0.00021 | | | | | | 1.00 | -0.98 | 0.95 | -0.91 | 0.01 | 0.03 |
| K2 | 0.0847842 | 0.00022 | | | | | | | 1.00 | -0.99 | 0.97 | -0.00 | -0.04 |
| K3 | -0.0165546 | 8.7e-05 | | | | | | | | 1.00 | -0.99 | -0.01 | 0.03 |
| K4 | 0.00137481 | 1.2e-05 | | | | | | | | | 1.00 | 0.01 | -0.02 |
| P1 | -0.000826239 | 4.5e-06 | | | | | | | | | | 1.00 | 0.21 |
| P2 | -0.000704217 | 1.1e-05 | | | | | | | | | | | 1.00 |

Table 3. Calibration coefficients and correlation matrix.

Camera Calibration

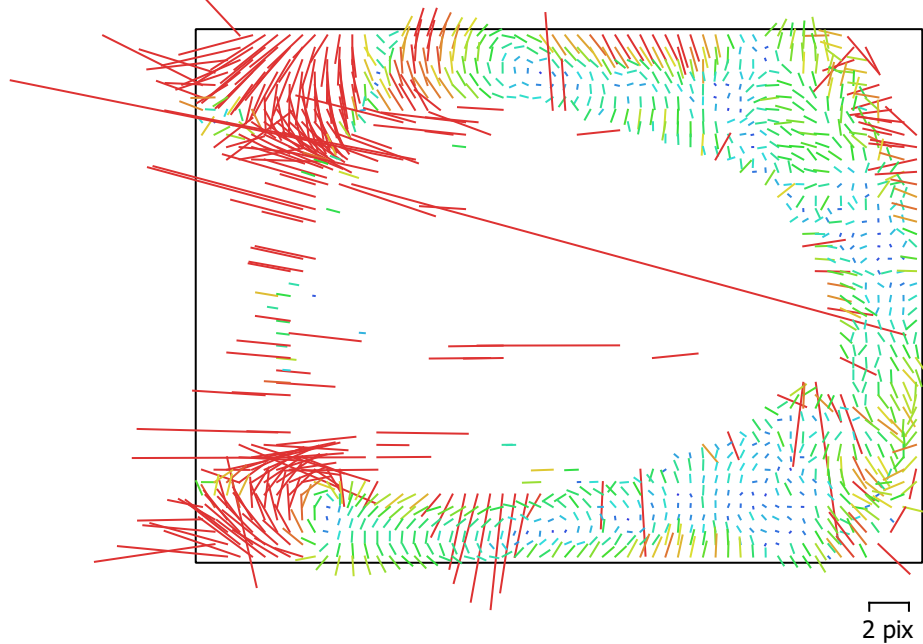


Fig. 4. Image residuals for Kamera 3.

Kamera 3

64 images

| | | | |
|--------------|--------------------|---------------|--------------------------------------|
| Type | Resolution | Focal Length | Pixel Size |
| Frame | 4096 x 3008 | 5.5 mm | 2.74 x 2.74 μm |

| | Value | Error | F | Cx | Cy | B1 | B2 | K1 | K2 | K3 | K4 | P1 | P2 |
|-----------|---------------------|---------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| F | 2145.99 | 0.2 | 1.00 | 0.14 | 0.18 | -0.20 | -0.20 | -0.73 | 0.67 | -0.63 | 0.60 | -0.18 | -0.11 |
| Cx | -31.3299 | 0.22 | | 1.00 | 0.04 | -0.06 | 0.01 | -0.11 | 0.10 | -0.09 | 0.09 | -0.22 | -0.01 |
| Cy | 51.7673 | 0.37 | | | 1.00 | -0.03 | -0.26 | -0.09 | 0.10 | -0.11 | 0.11 | -0.09 | -0.47 |
| B1 | 39.8183 | 0.17 | | | | 1.00 | -0.03 | -0.06 | 0.04 | -0.04 | 0.05 | 0.04 | 0.24 |
| B2 | 1.19414 | 0.075 | | | | | 1.00 | 0.07 | -0.06 | 0.06 | -0.06 | -0.04 | 0.08 |
| K1 | -0.311853 | 0.00016 | | | | | | 1.00 | -0.98 | 0.96 | -0.92 | 0.02 | -0.00 |
| K2 | 0.12584 | 0.00019 | | | | | | | 1.00 | -0.99 | 0.97 | -0.00 | -0.01 |
| K3 | -0.0326621 | 8.8e-05 | | | | | | | | 1.00 | -0.99 | -0.01 | 0.02 |
| K4 | 0.00358523 | 1.4e-05 | | | | | | | | | 1.00 | 0.03 | -0.03 |
| P1 | -0.00136706 | 3.5e-06 | | | | | | | | | | 1.00 | 0.04 |
| P2 | -0.000104722 | 6.2e-06 | | | | | | | | | | | 1.00 |

Table 4. Calibration coefficients and correlation matrix.

Camera Calibration

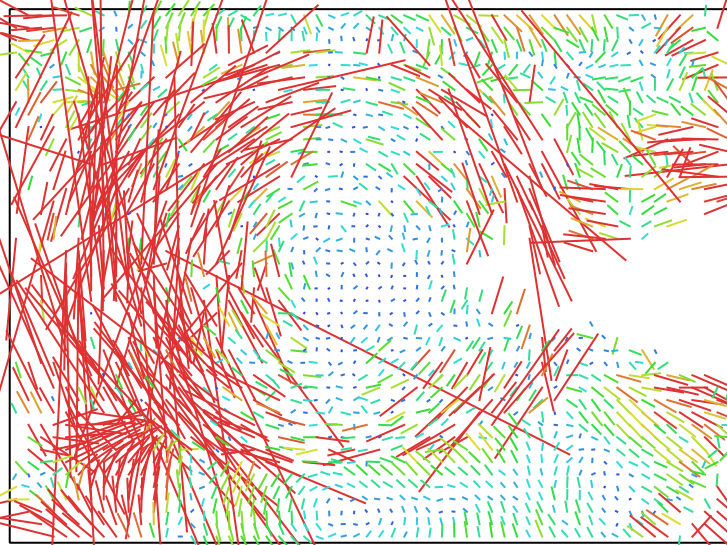


Fig. 5. Image residuals for Kamera 4.

Kamera 4

64 images

Type

Resolution

Focal Length

Pixel Size

Frame

4096 x 3008

5.5 mm

2.74 x 2.74 μ m

| | Value | Error | F | Cx | Cy | B1 | B2 | K1 | K2 | K3 | K4 | P1 | P2 |
|-----------|---------------------|---------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| F | 2135.04 | 0.27 | 1.00 | 0.33 | 0.01 | -0.45 | -0.03 | -0.71 | 0.68 | -0.66 | 0.65 | -0.27 | -0.06 |
| Cx | -28.7961 | 0.33 | | 1.00 | -0.01 | -0.28 | -0.12 | -0.27 | 0.30 | -0.32 | 0.35 | -0.38 | 0.05 |
| Cy | -70.046 | 0.33 | | | 1.00 | -0.01 | 0.07 | -0.15 | 0.17 | -0.19 | 0.21 | 0.08 | -0.60 |
| B1 | 29.2256 | 0.19 | | | | 1.00 | -0.13 | 0.07 | -0.09 | 0.11 | -0.11 | -0.04 | 0.07 |
| B2 | -1.15034 | 0.096 | | | | | 1.00 | 0.08 | -0.07 | 0.05 | -0.05 | 0.10 | -0.26 |
| K1 | -0.288928 | 0.00016 | | | | | | 1.00 | -0.99 | 0.96 | -0.92 | 0.17 | 0.06 |
| K2 | 0.101522 | 0.00016 | | | | | | | 1.00 | -0.99 | 0.97 | -0.18 | -0.09 |
| K3 | -0.0225988 | 6.3e-05 | | | | | | | | 1.00 | -0.99 | 0.19 | 0.11 |
| K4 | 0.00212999 | 8.5e-06 | | | | | | | | | 1.00 | -0.20 | -0.13 |
| P1 | -0.00125827 | 3.3e-06 | | | | | | | | | | 1.00 | -0.11 |
| P2 | -0.000145697 | 5.1e-06 | | | | | | | | | | | 1.00 |

Table 5. Calibration coefficients and correlation matrix.

Camera Calibration

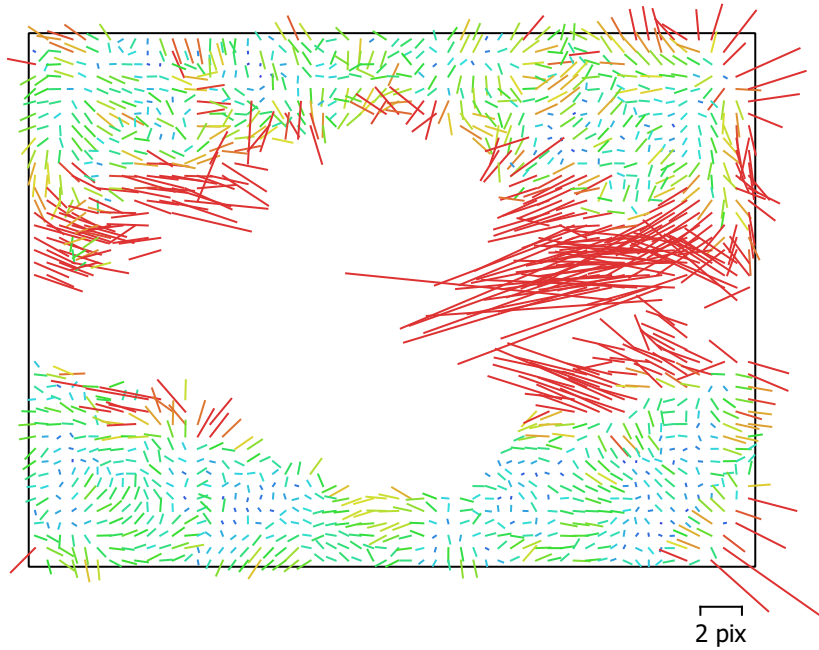


Fig. 6. Image residuals for Kamera 5.

Kamera 5

64 images

| | | | |
|--------------|--------------------|---------------|--------------------------------------|
| Type | Resolution | Focal Length | Pixel Size |
| Frame | 4096 x 3008 | 5.5 mm | 2.74 x 2.74 μm |

| | Value | Error | F | Cx | Cy | B1 | B2 | K1 | K2 | K3 | K4 | P1 | P2 |
|-----------|--------------------|---------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| F | 2088.78 | 0.18 | 1.00 | 0.23 | -0.03 | -0.34 | 0.07 | -0.74 | 0.67 | -0.63 | 0.60 | -0.01 | -0.12 |
| Cx | -112.927 | 0.45 | | 1.00 | -0.01 | 0.09 | 0.06 | -0.30 | 0.32 | -0.35 | 0.39 | -0.48 | 0.07 |
| Cy | -1.53708 | 0.23 | | | 1.00 | 0.06 | -0.03 | -0.05 | 0.07 | -0.07 | 0.07 | 0.00 | -0.43 |
| B1 | -14.9855 | 0.11 | | | | 1.00 | 0.08 | -0.10 | 0.07 | -0.06 | 0.06 | -0.21 | 0.05 |
| B2 | -6.5768 | 0.098 | | | | | 1.00 | -0.08 | 0.07 | -0.07 | 0.07 | 0.01 | 0.10 |
| K1 | -0.265165 | 0.00011 | | | | | | 1.00 | -0.98 | 0.95 | -0.92 | 0.10 | 0.06 |
| K2 | 0.08184 | 9.8e-05 | | | | | | | 1.00 | -0.99 | 0.97 | -0.11 | -0.05 |
| K3 | -0.0156313 | 3.3e-05 | | | | | | | | 1.00 | -0.99 | 0.12 | 0.04 |
| K4 | 0.00125437 | 3.9e-06 | | | | | | | | | 1.00 | -0.15 | -0.03 |
| P1 | 0.000128574 | 4.1e-06 | | | | | | | | | | 1.00 | -0.03 |
| P2 | -6.6989e-05 | 4.5e-06 | | | | | | | | | | | 1.00 |

Table 6. Calibration coefficients and correlation matrix.

Camera Calibration

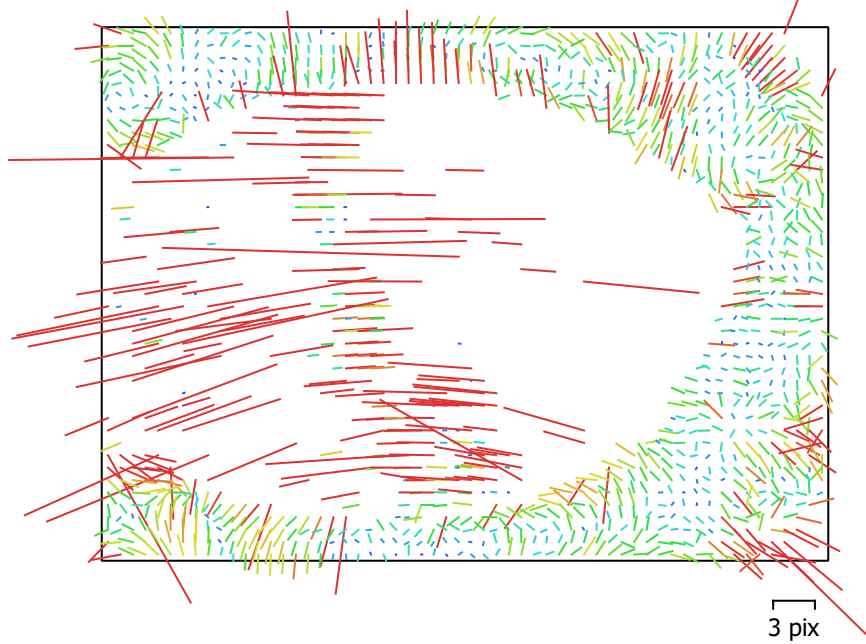


Fig. 7. Image residuals for Kamera 6.

Kamera 6

64 images

| | | | |
|--------------|--------------------|---------------|--------------------------------------|
| Type | Resolution | Focal Length | Pixel Size |
| Frame | 4096 x 3008 | 5.5 mm | 2.74 x 2.74 μm |

| | Value | Error | F | Cx | Cy | B1 | B2 | K1 | K2 | K3 | K4 | P1 | P2 |
|-----------|---------------------|---------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| F | 2105.31 | 0.25 | 1.00 | 0.20 | -0.15 | -0.24 | 0.18 | -0.79 | 0.75 | -0.71 | 0.67 | -0.12 | 0.11 |
| Cx | -81.0186 | 0.28 | | 1.00 | -0.07 | -0.21 | 0.01 | -0.05 | 0.04 | -0.04 | 0.04 | -0.33 | 0.04 |
| Cy | 16.3427 | 0.39 | | | 1.00 | 0.09 | -0.24 | 0.07 | -0.07 | 0.07 | -0.08 | 0.03 | -0.57 |
| B1 | 28.5763 | 0.15 | | | | 1.00 | -0.01 | -0.02 | 0.00 | 0.01 | -0.01 | 0.09 | -0.33 |
| B2 | 4.86915 | 0.09 | | | | | 1.00 | -0.08 | 0.06 | -0.06 | 0.05 | 0.19 | 0.11 |
| K1 | -0.271469 | 0.00019 | | | | | | 1.00 | -0.99 | 0.96 | -0.93 | -0.02 | -0.00 |
| K2 | 0.0845609 | 0.00017 | | | | | | | 1.00 | -0.99 | 0.97 | 0.02 | 0.01 |
| K3 | -0.0163109 | 6.2e-05 | | | | | | | | 1.00 | -0.99 | -0.01 | -0.01 |
| K4 | 0.00133077 | 7.7e-06 | | | | | | | | | 1.00 | -0.00 | 0.01 |
| P1 | -0.000758384 | 3.9e-06 | | | | | | | | | | 1.00 | 0.05 |
| P2 | 0.000530721 | 6.6e-06 | | | | | | | | | | | 1.00 |

Table 7. Calibration coefficients and correlation matrix.

Camera Locations

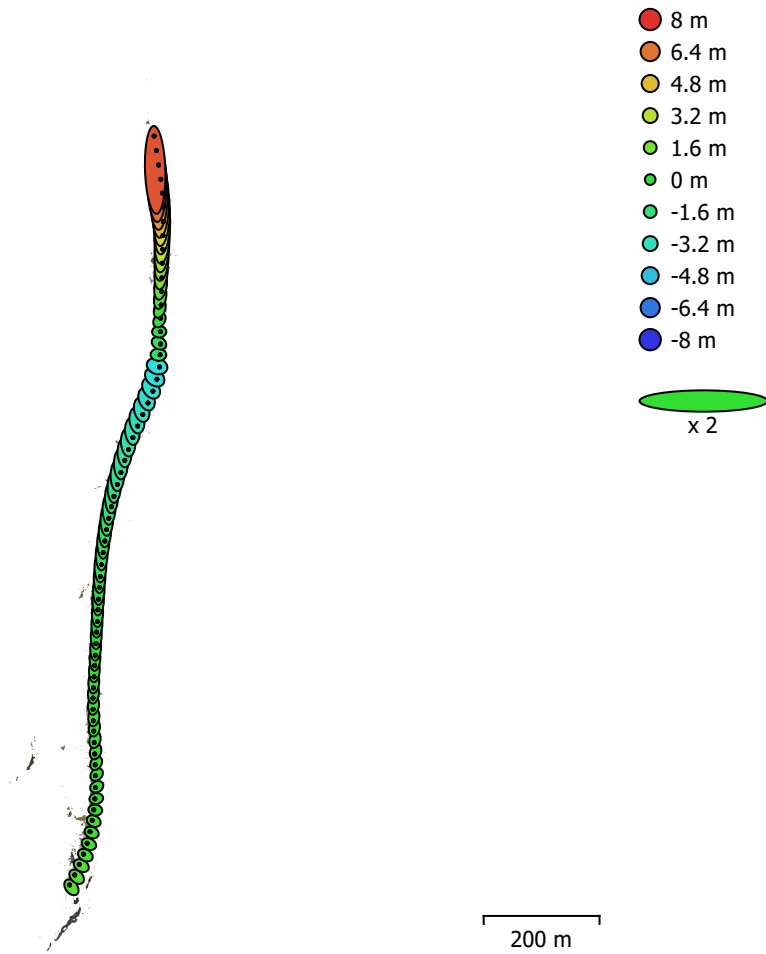


Fig. 8. Camera locations and error estimates.

Z error is represented by ellipse color. X,Y errors are represented by ellipse shape.

Estimated camera locations are marked with a black dot.

| X error (m) | Y error (m) | Z error (m) | XY error (m) | Total error (m) |
|--------------------|--------------------|--------------------|---------------------|------------------------|
| 1.86737 | 18.2595 | 2.334 | 18.3547 | 18.5025 |

Table 8. Average camera location error.

X - Longitude, Y - Latitude, Z - Altitude.

Ground Control Points



Fig. 9. GCP locations and error estimates.

Z error is represented by ellipse color. X,Y errors are represented by ellipse shape.

Estimated GCP locations are marked with a dot or crossing.

| Count | X error (m) | Y error (m) | Z error (m) | XY error (m) | Total (m) |
|-------|-------------|-------------|-------------|--------------|-----------|
| 2 | 2.85616 | 8.54791 | 1.25163 | 9.01246 | 9.09895 |

Table 9. Check points RMSE.

X - Longitude, Y - Latitude, Z - Altitude.

| Label | X error (m) | Y error (m) | Z error (m) | Total (m) | Image (pix) |
|--------------|--------------------|--------------------|--------------------|------------------|--------------------|
| point 1 | | | | | 1.101 (8) |
| point 2 | | | | | 1.923 (6) |
| point 3 | | | | | 10.899 (5) |
| point 4 | | | | | 2.841 (7) |
| point 5 | | | | | 1.262 (5) |
| point 7 | | | | | 0.994 (7) |
| point 8 | | | | | 1.769 (6) |
| point 9 | | | | | 1.230 (6) |
| point 10 | | | | | 1.453 (4) |
| point 11 | | | | | 1.539 (3) |
| point 12 | | | | | 0.852 (4) |
| point 13 | | | | | 3.221 (4) |
| point 14 | | | | | 0.409 (2) |
| point 18 | | | | | 0.479 (2) |
| point 19 | | | | | 1.938 (3) |
| point 20 | -3.38279 | 0.880389 | 1.70373 | 3.88858 | 4.465 (64) |
| point 23 | | | | | 2.156 (5) |
| point 24 | | | | | 1.550 (4) |
| point 25 | | | | | 1.417 (6) |
| point 32 | | | | | 1.366 (3) |
| point 39 | | | | | 0.888 (4) |
| point 40 | | | | | 1.223 (5) |
| point 41 | | | | | 2.171 (5) |
| point 42 | | | | | 1.091 (4) |
| point 43 | | | | | 2.424 (5) |
| point 44 | | | | | 2.698 (5) |
| point 46 | | | | | 1.171 (2) |
| point 47 | | | | | 0.796 (2) |
| point 49 | | | | | 1.855 (4) |
| point 50 | | | | | 5.443 (6) |
| point 51 | -2.20726 | -12.0565 | 0.480088 | 12.2662 | 1.692 (49) |

| Label | X error (m) | Y error (m) | Z error (m) | Total (m) | Image (pix) |
|--------------|--------------------|--------------------|--------------------|------------------|--------------------|
| Total | 2.85616 | 8.54791 | 1.25163 | 9.09895 | 3.540 |

Table 10. Check points.
X - Longitude, Y - Latitude, Z - Altitude.

Scale Bars

| Label | Distance (m) | Error (m) |
|-------------------|---------------------|------------------|
| point 50_point 51 | 5.95227 | -0.0477276 |
| Total | | 0.0477276 |

Table 11. Control scale bars.

Digital Elevation Model

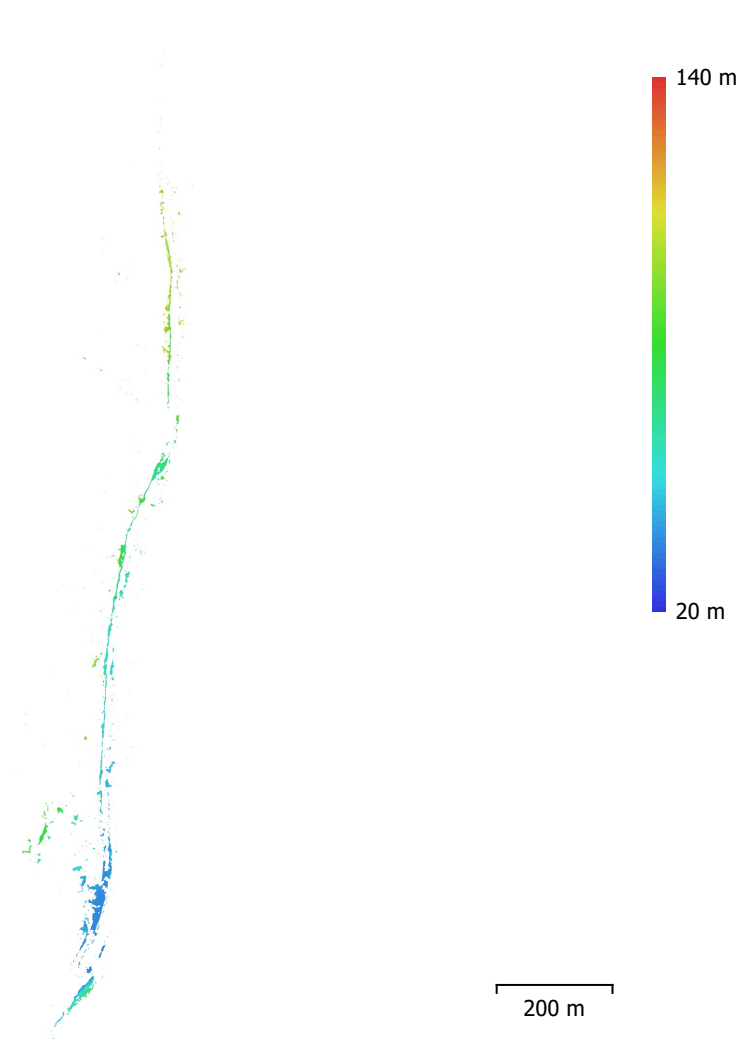


Fig. 10. Reconstructed digital elevation model.

Resolution: 2.4 cm/pix
Point density: 0.174 points/cm²

Processing Parameters

General

| | |
|-------------------|---------------------|
| Cameras | 384 |
| Aligned cameras | 384 |
| Markers | 43 |
| Scale bars | 1 |
| Coordinate system | WGS 84 (EPSG::4326) |
| Rotation angles | Yaw, Pitch, Roll |

Tie Points

| | |
|--------------------------------|------------------------|
| Points | 855,242 of 1,007,636 |
| RMS reprojection error | 0.895427 (3.02169 pix) |
| Max reprojection error | 10.8853 (233.128 pix) |
| Mean key point size | 3.8964 pix |
| Point colors | 3 bands, uint8 |
| Key points | No |
| Average tie point multiplicity | 2.09719 |

Alignment parameters

| | |
|-------------------------------|-----------------------|
| Accuracy | High |
| Generic preselection | Yes |
| Reference preselection | Source |
| Key point limit | 80,000 |
| Key point limit per Mpx | 80,000 |
| Tie point limit | 0 |
| Filter points by mask | Yes |
| Mask tie points | No |
| Exclude stationary tie points | No |
| Guided image matching | Yes |
| Adaptive camera model fitting | No |
| Matching time | 50 minutes 8 seconds |
| Matching memory usage | 1.12 GB |
| Alignment time | 15 minutes 39 seconds |
| Alignment memory usage | 1.23 GB |

Optimization parameters

| | |
|-------------------------------|----------------------------------|
| Parameters | f, b1, b2, cx, cy, k1-k4, p1, p2 |
| Adaptive camera model fitting | No |
| Optimization time | 12 seconds |
| Date created | 2024:03:27 00:00:00 |
| Software version | 2.1.0.17532 |
| File size | 54.75 MB |

Depth Maps

| | |
|-------|-----|
| Count | 193 |
|-------|-----|

Depth maps generation parameters

| | |
|------------------|---------------------|
| Quality | High |
| Filtering mode | Mild |
| Max neighbors | 16 |
| Processing time | 7 hours 25 minutes |
| Memory usage | 1.09 GB |
| Date created | 2024:03:27 00:00:00 |
| Software version | 2.1.0.17532 |
| File size | 25.36 MB |

Point Cloud

| | |
|--|-------------------------------|
| Points | 3,803,420 |
| Point attributes | |
| Color | 3 bands, uint8 |
| Normal | |
| Confidence | 1 - 4 |
| Point classes | |
| Created (never classified) | 3,803,420 |
| Depth maps generation parameters | |
| Quality | High |
| Filtering mode | Mild |
| Max neighbors | 16 |
| Processing time | 7 hours 25 minutes |
| Memory usage | 1.09 GB |
| Point cloud generation parameters | |
| Processing time | 7 minutes 9 seconds |
| Memory usage | 4.30 GB |
| Date created | 2024:03:27 00:00:00 |
| Software version | 2.1.0.17532 |
| File size | 57.06 MB |
| Model | |
| Faces | 13,895,559 |
| Vertices | 9,562,558 |
| Vertex colors | 3 bands, uint8 |
| Texture | 8,192 x 8,192, 4 bands, uint8 |
| Texturing parameters | |
| Mapping mode | Generic |
| Blending mode | Mosaic |
| Texture size | 8,192 |
| Enable hole filling | Yes |
| Enable ghosting filter | Yes |
| UV mapping time | 12 minutes 47 seconds |
| UV mapping memory usage | 1.74 GB |
| Blending time | 1 hours 15 minutes |
| Blending memory usage | 3.18 GB |
| File size | 793.81 MB |
| Tiled Model | |
| Texture | 3 bands, uint8 |
| Depth maps generation parameters | |
| Quality | High |
| Filtering mode | Mild |
| Max neighbors | 16 |
| Processing time | 7 hours 25 minutes |
| Memory usage | 1.09 GB |
| Reconstruction parameters | |
| Source data | Depth maps |
| Tile size | 256 |
| Face count | High |
| Enable ghosting filter | No |
| Processing time | 2 hours 28 minutes |
| Memory usage | 2.58 GB |
| Date created | 2024:03:27 04:06:46 |
| Software version | 2.1.0.17532 |
| File size | 396.01 MB |
| Orthomosaic | |
| Size | 20,997 x 83,150 |
| Coordinate system | WGS 84 (EPSG::4326) |

Colors 3 bands, uint8

Reconstruction parameters

Blending mode Mosaic
Surface Model
Enable hole filling Yes
Enable ghosting filter No
Processing time 9 minutes 6 seconds
Memory usage 1.57 GB
Date created 2024:03:27 00:00:00
Software version 2.1.0.17532
File size 1.18 GB

System

Software name Agisoft Metashape Professional
Software version 2.1.0 build 17532
OS Windows 64 bit
RAM 7.88 GB
CPU Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz
GPU(s) Intel(R) UHD Graphics 620



Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitenskapelige universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway