



Norges miljø- og
biovitenskapelige
universitet

Masteroppgave 2023 30 stp

Fakultet for realfag og teknologi

Maskinlæring for klassifisering av batymetrisk LiDAR: Bruk av geometriske egenskaper og integrering av undervannsbilder

Machine Learning for Classification of Bathymetric
LiDAR:

Utilization of Geometric Features and Integration of
Underwater Imagery

Eira Sørsdal og Ingrid Kamilla Brynildsen

Geomatikk

1. FORORD

Vi avslutter vårt studium i Geomatikk ved Norges miljø- og biovitenskapelige universitet med denne masteroppgaven. De fem siste årene har vært preget av verdifulle erfaringer og den mest erfaringsrike av dem blir her presentert.

Gjennom arbeidet med denne oppgaven har vi opparbeidet oss omfattende kunnskap, og prosessen har vært både spennende og utfordrende. Vi ønsker å rette en spesiell takk til hovedveileder Ivar Maalen-Johansen for hans raske svar og dyktige veiledning.

En stor takk går også til *Field Group* og medveileder Floris Jan Groesz for deres støttende veiledning og assistanse gjennom hele prosessen, samt for den tilgjengelige kontorlassen.

Vi vil også rette en stor takk til Kristine Solheim for grundig korrekturlesing i innspurten av oppgaven.

Vi ønsker å takke våre medstudenter, samt venner og familie for den gode støtten vi har mottatt underveis i masteroppgaven og studietiden.

Til slutt vil vi takke hverandre for gode diskusjoner, et godt samarbeid og for at vi har spilt hverandre gode.

2. SAMMENDRAG

I denne masteroppgaven har det blitt undersøkt hvordan geometriske egenskaper påvirker klassifisering ved bruk av maskinlæring av batymetrisk LiDAR, med treningsdata basert på undervannsbilder som referansedata. Undersøkelsen tar utgangspunkt i undervannsbilder fra havforskningsinstituttet for å definere treningsområder. Geometriske egenskaper for punktskyen er deretter beregnet ved bruk av ulike radius med programvaren *CloudCompare*.

I klassifiseringsmodellen *Random Forest* har treningsområdene med beregnede geometriske egenskaper, blitt benyttet for å gjøre predikasjoner på den gjenværende punktskyen. Flere tilnærminger med *Random Forest* er gjennomført for å utforske ulike modeller og deres resultat.

Resultatene viser at beregnede geometriske egenskaper for punkter i en batymetrisk punktsky er relevante for å skille ulike bunntyper. Spesielt har geometriske egenskaper knyttet til punktetthet vist seg å ha en betydelig påvirkning på klassifiseringen. Dette gjenspeiles i visualiseringen av flystriper fra overlapp, som tydelig påvirker klassifiseringen, og gir verdifull innsikt i sammenhenger mellom geometriske egenskaper og bunntyper. Det er også blitt konkludert med at det må utforskes videre.

3. ABSTRACTS

In this master's thesis, the impact of geometric features on machine learning-based classification using bathymetric LiDAR has been utilized with training data derived from underwater images as reference data. The study leverages underwater images from the Institute of Marine Research to define training areas. Geometric features of the point cloud are then computed using various scales with the software *CloudCompare*.

In the classification model *Random Forest*, training areas with calculated geometric features have been employed to make predictions on the remaining point cloud. Multiple approaches with *Random Forest* have been conducted to explore different models and their results.

The results indicate that computed geometric features for points in a bathymetric point cloud are relevant for distinguishing various seafloor types. Particularly, geometric features related to point density have shown a significant impact on the classification. This is reflected in the visualization of flight strip overlaps, which clearly influence classification and provide valuable insights into the relationships between geometric features and different seafloor types. It has also been concluded that further exploration in this direction is needed.

4. BAKGRUNN

Bakgrunnen for denne masteroppgaven springer ut av en betydelig interesse for å utforske flybåren batymetrisk LiDAR, et felt som hittil har vært relativt underrepresentert i undervisningen. Etter å ha fått et innblikk i batymetrisk LiDAR under en forelesning, ble interessen vekket, og ønsket om å dykke dypere inn i dette fascinerende fagfeltet oppstod.

Med en allsidig bakgrunn innen maskinlæring, programmering, GIS og fjernmåling ønsket vi å formulere en problemstilling som kunne kombinere disse fagområdene. Etter å ha blitt introdusert for muligheten til å benytte undervannsbilder fra havforskningsinstituttet av *Field Group*, fant vi dette aspektet særlig tiltalende og bestemte oss raskt for å implementere dette i vår masteroppgave.

Et sentralt spørsmål som oppstod var hvordan vi kunne dra nytte av undervannsbildene. En umiddelbar tanke var å anvende dem som treningsdata for å definere grunnlaget for klassifisering. Vi utfordret også konseptet om hvordan en batymetrisk punktsky, som mangler multispektrale stråler, kunne klassifiseres effektivt. Mangelen på informasjon under vann ga oss et realistisk dilemma.

Vi begynte å vurdere om geometrien til punktene, deres form og plassering i forhold til andre punkter, kunne gi verdifull informasjon som kunne brukes til å tildele punktene egenskaper. Disse egenskapene kunne deretter være nøkkelen til å klassifisere punktene ved hjelp av treningsdata basert på undervannsbildene fra havforskningsinstituttet.

Vi definerte derfor vår problemstilling slik:
Hvordan vil geometriske egenskaper påvirke klassifisering ved bruk av maskinlæring med treningsdata basert på undervannsbilder som referansedata?

Denne problemstillingen åpner opp for spennende utfordringer og muligheter.

5. TIDLIGERE ARBEID

Tidligere har Løyning & Storsveen (2020) skrevet en masteroppgave om klassifisering ved luftbåren batymetrisk lidar. Oppgaven omhandlet bruk av tradisjonell, *unsupervised* klassifisering ved to ulike programvarer, *PCI Geomatica* og *ArcGIS Pro*, for å undersøke om intensitetsverdi kan benyttes for å skille mellom ulike bunnsedimenter. Det ble i denne oppgaven konkludert med at intensitetsverdiene kan gi informasjon om grenser mellom ulike bunnsedimenter, men at intensitetsverdien alene ikke er en optimal metode.

Gjennom samarbeidet med *Field Group* har det blitt gitt innsikt i et tidligere, upublisert arbeid med maskinlæring på punktsky fra batymetrisk laserskanning. Dette arbeidet ble utført før tilgjengeligheten av undervannsbilder, men viste lovende resultater for bruk av punktegenskaper og naborelasjoner inn i maskinlæring for å klassifisere ulike bunntyper.

I en studie publisert av Kogut & Slowik (2021) ble det utført klassifisering av data fra luftbåren batymetrisk laserskanning ved bruk av nevralt nettverk (*Artificial Neural Networks*). Studien gikk ut på å bruke egenskaper ved full bølgeform og geometriske egenskaper i detektering og klassifisering av objekter på sjøbunn. Modellen som ble brukt var *multilayer perceptron*. I studien ble resultatet at nesten 100% prosent av vannoverflate og sjøbunn ble korrekt klassifisert, hvor objekter på sjøbunn ble klassifisert med en nøyaktighet på 80%.

Studien publisert av Atik, Duran & Seker (2021) ble det brukt ulike maskinlæringsmodeller for å klassifisere bygninger, vegetasjon og bakke i fire ulike områder. Det ble benyttet geometriske egenskaper inn i klassifiseringen, og det var tre ulike datasett. Resultatet viste at det var forskjellige algoritmer som gjorde det best i de ulike områdene, og det ble konkludert med at riktig metode varierer med datasettet.

Det har også blitt utført en studie av Blomley et al. (2014), hvor målet for studien var å introdusere nye pålitelige geometriske egenskaper for klassifisering av punktsky som baserte seg på formedistribusjoner. I studien ble det konkludert med at det har potensialet til å kunne beskrive strukturer utenfor det som inngås i geometriske egenskaper basert på naborelasjoner.

Innhold

1	Forord	3
2	Sammendrag	4
3	Abstracts	4
4	Bakgrunn	5
5	Tidligere arbeid	6
6	Teori	10
6.1	Flybåren batymetrisk laskerskanning	10
6.1.1	Feilkilder batymetri	11
6.2	Geometriske egenskaper i CloudCompare	11
6.3	Maskinlæring	14
6.3.1	Random Forest Classifier	16
7	Programvarer og utstyr	16
8	Datagrunnlag	17
8.1	Punktsky	17
8.2	Referansedata	19
9	Metode	21
9.1	Treningsdata	21
9.1.1	Forarbeid av referansedata	21
9.1.2	Valg av klasser og innsamling av treningsdata	23
9.1.3	Registrering av treningsdata i Terrascan	30
9.1.4	Utfordringer ved utvalg av treningsområder	31
9.1.5	Resultater treningsdata	33
9.2	Punktegenskaper	34
9.2.1	CloudCompare	35
9.2.2	Python rutine	36
9.2.3	Radius	37
9.3	Maskinlæringsrutine	40
9.3.1	Forberedelse	40
9.3.2	Inspeksjon og rensing	40
9.3.3	Utvalg av egenskaper	42
9.3.4	Oppdeling av treningsdata	43
9.3.5	Oppbygning av Random Forest rutine	44
9.3.6	Ulike klassifiseringsrutiner	45
9.3.7	Forsøk med Support Vector Machine	51
10	Resultater	52
11	Utfordringer	66
12	Drøfting	67
12.0.1	Drøfting av metode	67
12.0.2	Drøfting av resultater	68
13	Konklusjon	70
14	Videre arbeid	71

Figurer

1	”Skjematisk fremstilling av grønn og infrarød laser ved flybåren batymetrisk laserskanning”. Figuren er hentet fra Mandlbürger, G (2022) og har blitt redigert for å fokusere på relevante detaljer.	10
2	<i>Octree</i> generert i <i>CloudCompare</i>	12
3	Prosessen i random forest algoritmen. Figuren er hentet fra Koehrsen, W. (2017)	16
4	Original punktsky med klassifisering	18
5	Utsnitt fra original punktsky.	18
6	Bunnsedimentkart Fjøløy	20
7	Oversikt over arbeidsflyt i oppgaven	21
8	Oversikt inndelte treningsområder	22
9	Ulike spor i treningsområde 16	22
10	Skjerm bilde ArcGIS: Viser markert punkt i grønt	23
11	Undervannsbilde ved markert punkt	23
12	Bilde hvor steinrøys er synlig	23
13	Eksempler på undervannsbilder fra klassen: Sand	26
14	Eksempler på undervannsbilder fra klassen: Spredt vegetasjon	26
15	Eksempler på undervannsbilder fra klassen: Tett vegetasjon	27
16	Eksempler på utvalg av land	28
17	Eksempler på utvalg av stein-berg klassen	28
18	Eksempler på undervannsbilder fra klassen: Slam	29
19	Eksempler på utvalg av land	29
20	Terrascan: Utvalg av sand i område 5 i spor 3	30
21	Eksempel 1: Definerer av punkter	31
22	Eksempel 2: Overgang fra land til vann	32
23	Eksempel 3: Spredt vannoverflate	32
24	Fordeling av treningsområder uten vannoverflate klasse	33
25	Fordeling av manuelt annoterte punkter i de ulike klassene	34
26	Fordeling av bunnklassene over ulike dybdeintervaller.	34
27	Oppdeling av punktsky	35
28	Radiuseffekt fra 1m til 8m på datamende i punktsky ved Cloudcompare egenskaper	37
29	Radiuseffekt fra 0.5m til 3m for treningsdata på fullstendig punktsky	38
30	Punktetthet til fullstendig punktsky i gitt i pkt/m^2	39
31	Manglende verdier i de ulike klassene	41
32	Manglende verdier i gjenværende bunnklasser	41
33	Prosentandelen av gjenværende manglende verdier etter fjerning av manglende data i klasse 4 og 7.	42
34	Eksempel: Klassefordeling mellom treningsdata og testdata	44
35	Rutine 0	46
36	Eksempel på slam som har blitt feilklassifisert i rutine 0	46
37	Rutine A	47
38	Eksempel på overlapp flystriper, samt blanding av spredt vegetasjon og sand i rutine A	47
39	Rutine B	48
40	Rutine C	49
41	Rutine D	50
42	Rutine E	50
43	Rutine for Support Vector Machine	51
44	Resultat av SVM på den ene predikerte punktskyblokken	51
45	Oversikt over de ulike fargekode på de ulike klassene i den visuelle inspeksjonen av klassifiserings resultatene	54
46	Treningspunkter i område 1, 2 og 3 til bruk som referanse. Gul illustrerer <i>sand</i> , brun viser <i>slam</i> og rød viser treningspunkter for <i>land</i>	54
47	Område: 1, 2 og 3	55
48	Treningspunkter i område 4, 5 og 6 til bruk som referanse. Gul illustrerer <i>sand</i> , lys grønn viser <i>spredt vegetasjon</i> , mørk grønn tilhører <i>tett vegetasjon</i> og rødt viser treningspunkter for <i>land</i>	56
49	Område: 4, 5 og 6	57

50	Treningspunkter i område 17 til bruk som referanse. Gul illustrerer <i>sand</i> , lys grønn viser <i>spredt vegetasjon</i> , mørk grønn tilhører <i>tett vegetasjon</i> og grått viser treningspunkter for <i>stein, berg</i>	58
51	Område: 17	59
52	Treningspunkter i område mellom 17 og 18 til bruk som referanse. Her illustrerer grått treningspunkter for <i>stein, berg</i>	60
53	Område: mellom 17 og 18	61
54	Treningspunkter i område 18 til bruk som referanse. Gul illustrerer <i>sand</i> , lys grønn viser <i>spredt vegetasjon</i> og mørk grønn viser treningspunkter for <i>tett vegetasjon</i>	62
55	Område: 18	63
56	Prosentfordeling av predikasjoner	64
57	Eksempel på flere punkter i treningsdata etter endring av punktsky	66

Tabeller

1	Formler for geometriske egenskaper fra Hackel et al. (2016)	13
2	Klassebeskrivelse for original punktsky	19
3	Klasseoversikt	25
4	Parametersøk	45

6. TEORI

Før arbeidet som ble utført i masteroppgaven blir presentert, vil dette kapittelet ta for seg bakgrunnsinformasjon som er viktig for forståelsen av deler av oppgaven. Denne teoridelen tar for seg konseptene for flybåren batymetrisk laserskanning og feilkilder for målemetoden. Det blir også gitt en beskrivelse av maskinlæring og hvordan programvaren *CloudCompare* beregner geometriske egenskaper.

6.1. Flybåren batymetrisk laserskanning

Flybåren batymetrisk laserskanning har mange likhetstrekk til tradisjonell topografisk laserskanning, hvor flygeprosessen og navigasjonssystemet er det samme. I kontrast til topografisk laserskanning på land, har batymetrisk laserskanning noen særegne faktorer som skiller disse to fjerntmålingsteknikkene fra hverandre. En stor del av disse faktorene er at laserpulsene som utsendes må gjennom to ulike medium, luft og vann.

I batymetrisk laserskanning anvendes kun to områder av det elektromagnetiske spekteret. En nærinfrarød laserpuls med bølgelengde på 1064 nm brukes til å registrere vannoverflaten, mens en laserpuls med grønt lys og bølgelengde på 532 nm brukes til å trenge gjennom vannsøylen. (Szafarczyk & Toś 2022). Dette skyldes at lengre bølgelengder blir absorbert raskere i vannkolonnen sammenlignet med bølgelengder nærmere synlig lys. Synlig lys inneholder mer energi og kan derfor nå dypere før det dempes (Webb (2023)).

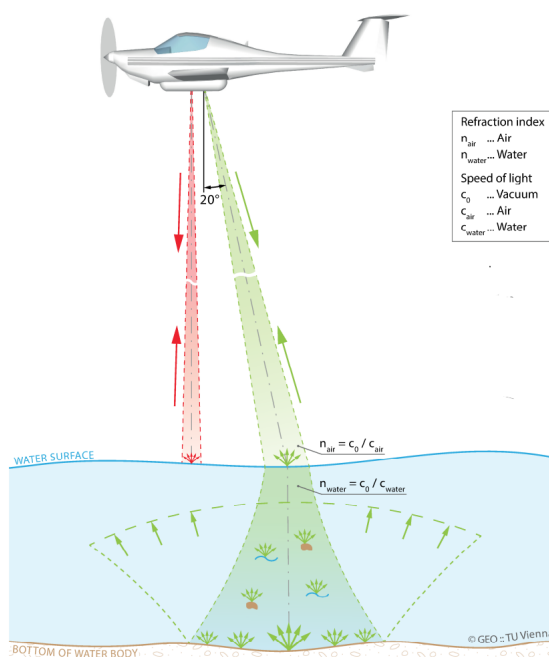


Fig. 1: "Skjematisk fremstilling av grønn og infrarød laser ved flybåren batymetrisk laserskanning". Figuren er hentet fra Mandlbürger, G (2022) og har blitt redigert for å fokusere på relevante detaljer.

Fremstillingen av disse laserpulsene er illustrert i figur 1. De utsendte pulsene av laserstrålene beveger seg gjennom atmosfæren og når laserpulsene treffer vannoverflaten absorberes det meste av den nærinfrarøde pulsen, men noe blir spredt og reflektert tilbake. Den kortere, grønne laserpulsene blir også noe spredt ved vannoverflaten, og resten av strålen trenger gjennom vannoverflaten. Når lyset går fra luft til vann oppstår en refraksjon, denne avbøyingen av lyset og endring i signalutbredelsen kan beskrives ved Snells brytningslov i likning 1 (Mandlbürger 2020).

$$\frac{\sin(\alpha_{luft})}{\sin(\alpha_{vann})} = \frac{n_{vann}}{n_{luft}} \quad (1)$$

hvor n_{luft} og n_{vann} er brytningsindeksen som bestemmes av henholdsvis lyshastighet i vakuum, luft og vann. α_{luft} er innfallvinkelen fra luft og α_{vann} er brytningsretning i vann. (Skaar & Holtmark 2020)

Etter å ha trengt gjennom vannsøylen, gjennomgår den grønne laserpulsens ytterligere påvirkning ved å bli videre spredt og absorbert på vei ned til sjøbunnen. Når pulsen reflekteres tilbake, varierer graden av refleksjon ut i fra bunntype og form. Mørke bunnsedimenter absorberer laserpulsens i større grad enn lysere sedimenter som reflekterer mer av strålen tilbake. Formen på sjøbunnen er også med på å avgjøre hva som reflekteres tilbake. Flate områder gir en annen spredning av signalet enn røffe overflater, slik som stein og klipper. Laserpulsens blir så reflektert tilbake, hvor den også møter på spredning og absorpsjon før den blir detektert av sensoren i flyet. Av disse grunnene trenger batymetrisk laserskanning høy energi på pulsen og sensitiv detektor (de Jongh, C 2021) (Vosselmann & Maas 2010).

6.1.1. Feilkilder batymetri

I tillegg til nevnte påvirkninger som spredning og absorpsjon, er det også andre faktorer som kan påvirke laserpulsens i form av avgrensinger eller feil. Disse er blant annet bevegelser i vannet og bølger, turbiditet og dybde.

Bølger og bevegelse i vannoverflaten er blant annet en av kildene til feil og støy i batymetrisk laserskanning. Avhengig av bølgenes størrelse kan dette påvirke laserpulsens ved vannoverflaten ikke lenger er horisontal. Dette resulterer i retningsendringer og spredning av strålen, eller linse-lignende effekter som fokuserer strålen. Slike påvirkninger kan føre til feil i avstandsmålingen, da de påvirker refraksjonsvinkelen og hvordan laserpulsens retning i vannsøylen beregnes (Philpot 2019).

Skum og luftbobler er også eksempler på tilfeller hvor laserpulsens kan spres eller skape støy. I henhold til Szafarczyk & Toś (2022) kan luftbobler i vannet fungere som små reflektorer som sprer energien fra laserpulsens. Dette kan føre til redusert signalstyrke og gjør tolkningen av den reflekterte strålen mer kompleks, da det blir utfordrende å skille mellom bunnrefleksjoner fra sjøbunn og refleksjoner fra luftbobler og andre partikler i vannet.

Dybde og turbiditet er også faktorer som har stor innvirkning på den utsendte laserpulsens. I vannsøylen reduseres lysintensiteten eksponentielt når dybden øker (Nababan et al. 2021). Kvaliteten til vannet har også en innvirkning. Turbiditet, som er et mål på vannets klarhet, er en samling av organisk materiale som alger og andre partikler som påvirker vannkvaliteten og gjennomsiktigheten til vannet (Grobelaar 2009). Ved økt turbiditet kan disse partiklene spre og absorbere innkommende lys, som beskrevet i Richter, Maas & Westfeld (2017) avgjør dette hvor dypt lyset kan trenge ned i vannsøylen. Ved høy turbiditet er den maksimale dybden rundt 1 til 2 meter, mens ved lav turbiditet med meget klart vann, kan laserpulsens nå ned til omtrent 50 meter.

6.2. Geometriske egenskaper i CloudCompare

CloudCompare er en gratis prosesserings- og redigeringsprogram for punkttskyer i 3D (Girardeau-Montaut 2017). Programvaren har mange ulike bruksmuligheter, som blant annet å kunne generere geometriske egenskaper til en punkttsky. Den har mulighet til å generere en rekke geometriske egenskaper, som er listet opp nedenfor.

- Curvature
 - Mean
 - Gaussian
 - Normal change rate
- Density
 - Number of neighbors
- Surface density
- Volume density
- Roughness
- 1st Order Moment
- Sum of eigenvalues
- Omnivariance
- Eigenentropy
- Anisotropy
- Planarity
- Linearity

- PCA1
- PCA2
- Surface variation
- Sphericity
- Verticality
- 1st eigenvalue
- 2nd eigenvalue
- 3rd eigenvalue

De geometriske egenskapene som beregnes for hvert punkt i programmet beseres på naboene innenfor en gitt radius. For å kunne definere nabopunkter benytter *CloudCompare* seg av **Octree**, mer definert et **DgmOctree**(Girardeau-Montaut 2015), for å lage strukturer mellom punkter.

Et *octree* er en trestruktur hvor hver node i treet har enten 0 eller 8 barn. Ved bruk av en punktsky deles punktskyen opp i 8 celler basert på totalt volum. Deretter deles hver enkelt celle opp igjen i 8 nye celler, og denne prosessen gjentas helt til det ikke er flere punkter igjen i cellene (Dybedal et al. 2019). Å benytte *octree* for å estimere naborelasjoner i en punktsky har store fordeler for lagring og indeksering av romlig data (Mosa et al. 2012). Kompleksiteten minskes, og tiden for å utføre operasjoner avtar. Oppbygningen av et *octree* sørger for at ikke all tilgjengelig data må undersøkes for hvert søk, men at søket følger grenene av et tre for å hente informasjon. Se figur 2 for visuell figur av de ulike nivåene i en slik struktur.

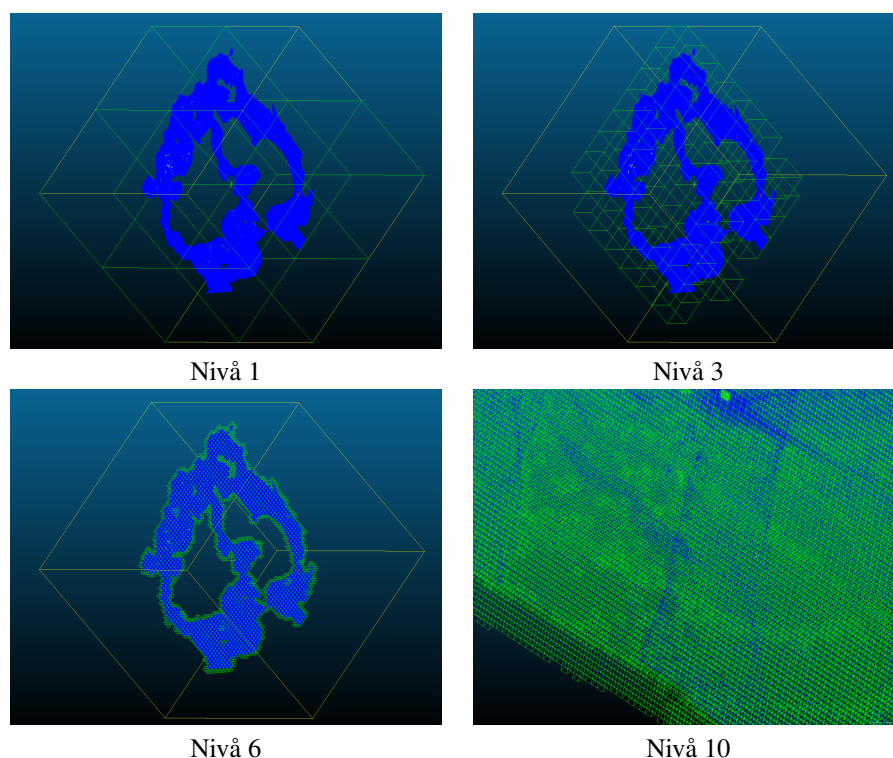


Fig. 2: *Octree* generert i *CloudCompare*

For å forstå nærmere hvordan *CloudCompare* håndterer octree-strukturen, er det viktig å se nærmere på den spesifikke implementasjonen de bruker, *Dgmoctree* (Girardeau-Montaut 2015). I *Dgmoctree* i *CloudCompare* defineres strukturen som en liste av numeriske verdier, der hvert tall korresponderer til den absolutte posisjonen til et punkt for alle nivåer av underinndeling. Dette avviker fra det klassiske treformatet ved å kode posisjonen til punktene på en metode som lettere egner seg for romlig indeksering.

Ved å koble disse kodene oppnås en unik representasjon av hver celle, der to punkter som ligger i samme celle på et gitt nivå av inndeling, deler samme delkode. Videre blir kodene sortert for å muliggjøre raskt binærsøk og romlig gjennomgang av punktskyen. Den sorterte listen sikrer at naboceller ligger ved siden av hverandre. Dette gir fordeler som effektiv romlig indeksering og optimal ytelse ved søk og analyse av punktskyer.

Ved hjelp av *Dgmoctree* gjøres et raskt søk etter hvert punkts nabopunkter innenfor en gitt radius. Ved hjelp av nabopunktene beregnes en kovarianstensor (likning 2), som videre benyttes ved beregningene av de geometriske egenskapene. De fleste matematiske beskrivelsene av egenskapene er beskrevet i (Hackel et al. 2016), som også *CloudCompare* sin egen *wiki* side (Girardeau-Montaut 2015) henviser til.

$$\Sigma_i = \frac{1}{N} \sum_{n \in P_i^N} (p_n - \bar{p})(p_n - \bar{p})^T \quad (2)$$

Likning 2 er beskrevet i Hackel et al. (2016), her representerer P_N settet av de N nærmeste naboene til p_i , p_n er individuelle punkter, \bar{p} er medoiden, \sum er summasjonssymbolet, og T indikerer transponering av vektoren $(p_n - \bar{p})(p_n - \bar{p})^T$.

De geometriske egenskapene fra *CloudCompare* bygger på ulike kombinasjoner av egenverdiene og egenvektorene til de beregnede kovarianstensorene. Disse er beskrevet i tabell 1 i Hackel et al. (2016), vist i tabell 1 nedenfor.

Sum of eigenvalues	$\lambda_1 + \lambda_2 + \lambda_3$
Omnivariance	$(\lambda_1 \cdot \lambda_2 \cdot \lambda_3)^{\frac{1}{3}}$
Eigenentropy	$-\sum_{i=1}^3 \lambda_i \cdot \ln(\lambda_i)$
Anisotropy	$\frac{\lambda_1 - \lambda_3}{\lambda_1}$
Planarity	$\frac{\lambda_2 - \lambda_3}{\lambda_1}$
Linearity	$\frac{\lambda_1 - \lambda_2}{\lambda_1}$
Surface Variation	$\frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$
Sphericity	$\frac{\lambda_3}{\lambda_1}$
Verticality	$1 - \langle [0 \ 0 \ 1], e_3 \rangle $

Table 1: Formler for geometriske egenskaper fra Hackel et al. (2016)

Hver egenverdi gir informasjon om størrelsen av spredningen for x, y og z-retning, og kan beskrive formen til punktet og dets nabopunkter. Summen av egenvektorene kan beskrive orientering til området, mens *Omnivarians* er den gjennomsnittlige spredningen for de tre retningene og beskriver skarpheten til overflaten (dos Santos et al. 2022).

Egenentropien gir verdier for hvor mye orden det er i området. Små verdier indikerer orden og lite støy, og store verdier indikerer mer uorden i området (Bell et al. 2023). Anisotropi av et punkt beskriver graden av variasjon og orientering i flere dimensjoner, større verdier indikerer sterk variasjon i ulike retninger. Disse egenskapene gir begge en indikasjon på grad av variasjon og orientering av punktene.

Planarity kvantifiserer i hvor stor grad et punkt og dets nærmeste nabopunkter ligger i et plan, mens linearitet måler graden av retthet i punktets omgivelser (Bell et al. 2023). Disse egenskapene er spesielt relevant for terrengformen. *Surface variation*, variasjon i overflaten, kan beskrive hvor godt et området danner en jevn overflate som defineres av lave verdier (Jia et al. 2019).

Sphericity, som definert av Guo et al. (2017), er et mål på forholdet mellom overflatearealet til en kule og overflatearealet til partikkelen. I denne konteksten beregnes *sphericity* ved å vurdere hvor mye et punkt og dets nabopunkter tilnærmer seg formen til en perfekt kule med samme radius som beregnet ved bruk av kovarianstensenoren.

Verticality beregnes ut ifra egenvektor tre fra kovarianstensenoren. Denne vektoren viser til retningen punktet har i z-retning og er punktets normalretning (Demantké et al. 2012). Vertikalmålet til punktet blir så estimert, og gir et mål på hvor horisontalt punktets normal er.

$$O = \frac{m^{\uparrow 2}}{m^{\uparrow}} \quad \text{med} \quad m^{\uparrow} = \sum_{n \in P_N} \langle p_n - p_i, e_{2i} \rangle, \quad m^{\uparrow} = \sum_{n \in P_N} \langle p_n - p_i, e_{2i} \rangle^2 \quad (3)$$

Likning 3 fra Hackel et al. (2016) beskriver beregningen av *1st order moment*, m^{\uparrow} . Denne verdien beregnes ved skalarproduktet mellom punktet, dets nabopunkter og den andre egenvektoren. Dette normaliseres ved andre ordens moment, m^{\uparrow} , som sikrer at resultatet er i riktig område mellom 0 og 1. Første ordens moment beskriver fordelingen av punkter rundt den minste egenvektoren, spesifikt i retning av den første egenvektoren.

PCA, *Principal Component Analysis*, er en teknikk for dimensjonsreduksjon (Basu Mallick 2023). Egenskapene baserer seg på egenvektorene og egenverdiene, som beskrevet av administrator for *cloudCompare*, Girardeau-Montaut (2019). *PCA1* er komprimert informasjon om relasjonen mellom de to første egenvektorene og den største egenverdien. Disse utgjør retningen i området som er mest utstrakt. *PCA2* er komprimert informasjon om relasjonen mellom den tredje egenvektoren knyttet til den minste egenverdien, som angir retningen med den minste utstrekningen.

Curvature, eller krumning, beskriver variasjonen til overflaten, hvor mye overflaten krummer lokalt for et område. Denne egenskapen gir informasjon om hvordan overflaten endrer seg for hvert punkt, og kan bidra til å skille mellom ulike terrengformer. Krumning kan estimeres på tre ulike måter, *Gjennomsnittlig*, *Gausisk* og *Normal endringshastighet*. *Normal endringshastighet* er beregnet ved å dele den minste egenverdien på summen av de totalt tre egenverdiene fra det lokale nabolagets kovariansmatrise (Girardeau-Montaut 2022a). Girardeau-Montaut (2022a) henviser også til Har'el (1995) for beregning av *Gausisk* og *Gjennomsnittlig* Krumning, disse beregningene bygger på en kvadratisk tilnærming.

Density kan også defineres som punkttetthet. Egenskapen gir en verdi på tilstedeværelsen av punkter i et område, og kan relateres til ulike objekter og strukturer. Punkttettheten beregnet i *CloudCompare* kan beskrives på ulike måter, enten ved *Antall nabopunkter*, *Overflate tetthet* eller *Volum tetthet*. *Antall naboer* gir punkttettheten ved å telle opp nabopunktene til et utgangspunkt innenfor en gitt radius (Girardeau-Montaut 2022b). *Overflate tetthet* bruker *Antall naboer* og deler antallet på overflaten til det estimerte nabolaget. Dette gjøres også ved estimering av *Volum tetthet*, men hvor *Antall naboer* deles på nabolagets volum.

En annen egenskap *CloudCompare* kan estimere er *Roughness*, oversatt til *ujevnhet*. Egenskapen gir en verdi for grad av ujevnhet på overflaten til et punkt, det vil si hvor mye overflaten avviker fra et ideelt plan i x- og y-koordinatene. Verdien som blir satt er avstanden fra punktet som undersøkes til det beste estimerte planet i x- og y-koordinatene, til de tilhørende nabopunktene (Girardeau-Montaut 2022c). *Ujevnhet* kan bidra til å karakterisere ujevnheter eller endringer på overflaten.

6.3. Maskinlæring

Maskinlæring går ut på å trene opp en modell til å finne mønster i et datasett. (Tidemann & Elster 2023) Bakgrunnen for videre beskrivelse av maskinlæring bygger på forelesningsnotater fra Tomic, O. 2022. Ch. 01 – Giving Computers the Ability to Learn from Data, Ch04 - Building Good Training Sets - Data Preprocessing. *DAT200*.

Det finnes mange ulike modeller som bygger på ulike algoritmer, disse brukes etter ulike formål. Felles for disse algoritmerne er at de tillater maskinen å lære i stedet for å bli direkte programmert.

Det finnes tre ulike typer maskinlæring:

- *Supervised Learning*

- *Unsupervised Learning*
- *Reinforcement Learning*

I denne oppgaven er det imidlertid kun blitt benyttet *supervised learning*. *Supervised learning* innebærer bruk av kategorisert data for å trene en maskinlæringsmodell. Basert på det modellen har lært fra treningsdata, kan den deretter utføre predikasjoner på et nytt datasett som maskinen ikke tidligere har blitt eksponert for.

Innenfor *supervised learning* finnes det flere underkategorier. Et eksempel på en slik underkategori er klassifisering, som også er blitt brukt i denne oppgaven. Klassifisering har som formål å predikere kategorier eller klasser i et datasett. Den gir predikasjoner om nye klasseetiketter for et ukjent datasett basert på tidligere observasjoner i et treningsdatasett.

Som tidligere nevnt, eksisterer det mange ulike modeller innenfor maskinlæring. Valg av modell samt bestemt tilnærming avhenger i stor grad av datagrunnlaget og formålet med maskinlæringsprosessen. De ulike modellene er designet for å takle varierte problemstillinger og krever derfor ulike tilnærminger.

En vesentlig komponent i maskinlæringsprosessen er forberedelsen av data. Data som går inn i modellene skal være rensset for manglende verdier og/eller utliggere (*outliers*). I mange tilfeller krever modellen at data skal være skalert. Det finnes flere tilnærminger å løse disse utfordringene på, og i maskinlæring finnes det ingen universell fasit. Valg av metode avhenger derfor av formålet en skal bruke maskinlæringen til.

I en typisk maskinlæringsprosess deles treningsdata vanligvis inn i et treningsett og et testsett. Dette gjøres blant annet for å kunne vurdere modellens evne til å generaliseres til ny, usett data. Videre kan det utføres et egenskapsutvalg av det tilgjengelige datasettet for å vurdere hvilke egenskaper som er mest relevante å inkludere i modellen.

Maskinlæringsmodeller har vanligvis flere parameter som påvirker algoritmen, og modellen må tilpasses problemstillingen og datasettet. En metode for å finne de mest optimale parameterne kalles *hyperparameter tuning*, der ulike kombinasjoner av parametre testes for å finne den beste tilpasningen til datasettet.

Videre må ytelsen til den endelige modellen evalueres. Det finnes ulike evalueringsmetoder som velges basert på problemstillingen, men generelt skal disse evalueringene gi en uavhengig måling av modellens ytelse på usett data.

Overtilpasning, kjent som *Overfitting*, er et begrep i maskinlæring der modellen presterer mye bedre på treningsdata enn på testdata. Dette kan skje hvis modellen er overtrent og ikke har så bra overføringsevne til ny data eller at modellen er for kompleks for det gitte datagrunnlaget. For å løse overtilpasning av en modell kan en samle mer treningsdata, velge en enklere modell med færre parametere eller redusere dimensjonen på datasettet gjennom et egenskapsutvalg.

6.3.1. Random Forest Classifier

Random Forest Classifier er en maskinlærings-algoritme som blir brukt til klassifiserings formål. Den baserer seg på ulike beslutningstrær og kombinerer disse for å predikere et resultat.

Hvert beslutningstre kan betraktes som et slags flytdiagram. I følge Liberman (2017) skal et beslutningstre besvare en rekke sekvensielle spørsmål som styrer retningen nedover i treet. De ulike betingelsene gir tilslutt et spesifikt resultat.

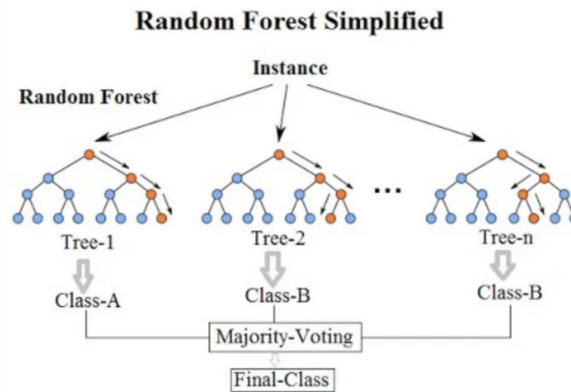


Fig. 3: Prosessen i random forest algoritmen. Figuren er hentet fra Koehrsen, W. (2017)

Figur 3 illustrer prosessen i *random forest* algoritmen. De ulike beslutningstrærne blir tilpasset med et tilfeldig utvalg av treningsdatasettet som gjør at det blir variasjon mellom beslutningstrærne. Når alle beslutningstrærne har kommet frem til et resultat, blir den endelig predikasjonen bestemt av flertallet fra beslutningstrærne (Pedregosa et al. 2011).

7. PROGRAMVARER OG UTSTYR

For gjennomføring av denne masteroppgaven er det benyttet flere ulike programvarer, disse er listet opp nedenfor og beskrevet kort. Videre gis det en kort beskrivelse av maskinvareutstyr som har blitt brukt.

Programvarer

- CloudCompare v2.12.4 (Kyiv) og v2.13.alpha
- Terrascan på Spatix plattform (versjon: 023_025)
- ArcGIS Pro 3.1.3
- Jupyter Notebook 6.5.4
- Anaconda Navigator 2.5.0
- Quick Terrain modeler 830_IX

CloudCompare er som nevnt tidligere en gratis prosesserings- og redigeringsprogramvare for punktskyer i 3D. Denne programvaren har blitt brukt i metoden til å generere geometriske egenskaper.

Terrascan er en applikasjon tilhørende *Terrasolid Software* og krever lisens (Terrasolid 2023). Programmet kjøres via plattformen *Spatix*. *Terrascan* kan håndtere og behandle alle typer punktskyer. I oppgaven har den vært svært nyttig for å klassifisere treningspunkter manuelt.

ArcGIS Pro er et lisensbasert program fra Esri (n.d.), som brukes for å utføre geografisk informasjonsbaserte (GIS) oppgaver. I denne oppgaven har ArcGIS Pro blitt brukt aktivt i prosessen med å velge ut treningsdata som manuelt skal annoteres.

Jupyter Notebook er, som beskrevet i Jupyter (2023), en gratis programvare med åpen standard. Det er en nettbasert applikasjon for programmering på tvers av programmeringsspråk. Programmet har i hovedsak blitt brukt til klassifiseringen av punktskyen. Denne applikasjonen brukes gjennom *Anaconda Navigator*, som er en plattform for å kjøre applikasjoner som *jupyter notebook*, håndtere eksterne tilleggspakker og programmeringsmiljøer (Navigator 2018). Dette er også en åpen og gratis applikasjon.

Quick Terrain Modeler, som leveres av Applied Imagery (2022), er en punktskyapplikasjon. Den har i hovedsak blitt brukt til visuell inspeksjon i oppgaven. Denne programvaren krever lisens, men *Applied Imagery* tilbyr også en gratis prøveperiode.

Maskinvare

- **MacBook Pro**
2.2 GHz firekjerners intel core i7 prosessor
16 GB minne
Windows med Parallels Desktop
- **MacBook Pro**
2.3 GHz firekjerners intel core i5 prosessor
8 GB minne
Bootcamp med Windows
- Ekstern harddisk 5TB
- Ekstern harddisk 2TB

Alle programvarene som har blitt brukt i oppgaven er ikke kompatible med *macOS*. Dette gjelder for *Terrascan*, *Quick Terrain Modeler* og *ArcGIS Pro*. For å løse dette har det blitt benyttet en løsning med *Windows med Parallels Desktop* og en løsning med *Bootcamp med Windows*. Disse løsningene tillater å kjøre *Windows* operativsystem på Mac. Programvaren *Terrascan* gjennom plattformen *Spatix* var derimot avhengig av å bli brukt med *Bootcamp* løsningen, på grunn av lisenstilgang gjennom Norges miljø- og biovitenskaplige universitets VPN-løsning. Løsningen med *Windows med Parallels Desktop* kjører *Windows* med virtuell maskin, noe som gjorde at det ikke fikk tilgang til lisensen.

For gjennomføring av denne oppgaven har det vært essensielt med ekstra lagringskapasitet. Selv om *laz*-filer er godt komprimert har det vært nødvendig med ekstra lagringskapasitet til lagring av flere ulike versjoner, og ved benyttelse av *csv*-filer. Det har vært viktig å frigjøre lagringsplass på maskinene for systemdata.

8. DATAGRUNNLAG

Det vil i dette kapitlet bli gitt en gjennomgang av datagrunnlaget for metoden i denne masteroppgaven. Det vil bli gjennomgått hvilke type data som er samlet inn, kildene til disse dataene og deres relevans for oppgaven. Datagrunnlaget er fra området rundt Fjøløy og Klosterøy, ikke så langt fra Stavanger. Arbeidet bygger på data fra *Havforskningsinstituttet* og fra *Field Group*.

8.1. Punktsky

Punktskyen som ble benyttet i denne oppgaven var en punktsky som ble laget i forbindelse med pilotprosjektet *Marine grunnkart i kystsonen Terratec AS (2021)*. Gjennom prosjektet har det blitt samlet inn data med flybåren batymetrisk laserskanning i Stavanger, med Statens Kartverk som oppdragsgiver. Prosjektet er gjennomført av *Field Group*, tidligere *Terratec* og *Bloom*.

Batymetrisk data har blitt samlet inn ved bruk av lasersensoren *CZMIL SuperNova (CZ10)*, hvor også tilhørende ortofoto har blitt fanget opp med kamera *Phase One IXM-RS 150F*. GNSS mottakeren *Trimble POS AV 610 med PPRTX* har blitt brukt til registrering av flyets posisjoner. Dato for flyingen var 25. april. 2021. Flere detaljer finnes i prosjektrapporten *Terratec AS*

(2021).

Punktskyen ble levert i koordinatsystemet *UTM sone 32, (ETRS89/UTM) (EPSG:25832)*, både med høydemodell definert i *NN2000(ortometrisk)* og ellipsoidisk høyde. For denne masteroppgaven ble ortometrisk brukt som vertikalt koordinatsystem. Punktettheten for punktskyen varierer med dybde, slik som det står beskrevet i Terratec AS (2021). Punktetthet på $5pkt/m^2$ ved sjøkart null, $3pkt/m^2$ ned til 5 meter under sjøkart null og $1pkt/m^2$ ned til 10 meter under sjøkart null.

Utsnitt fra punktskyen kan ses i figur 4.

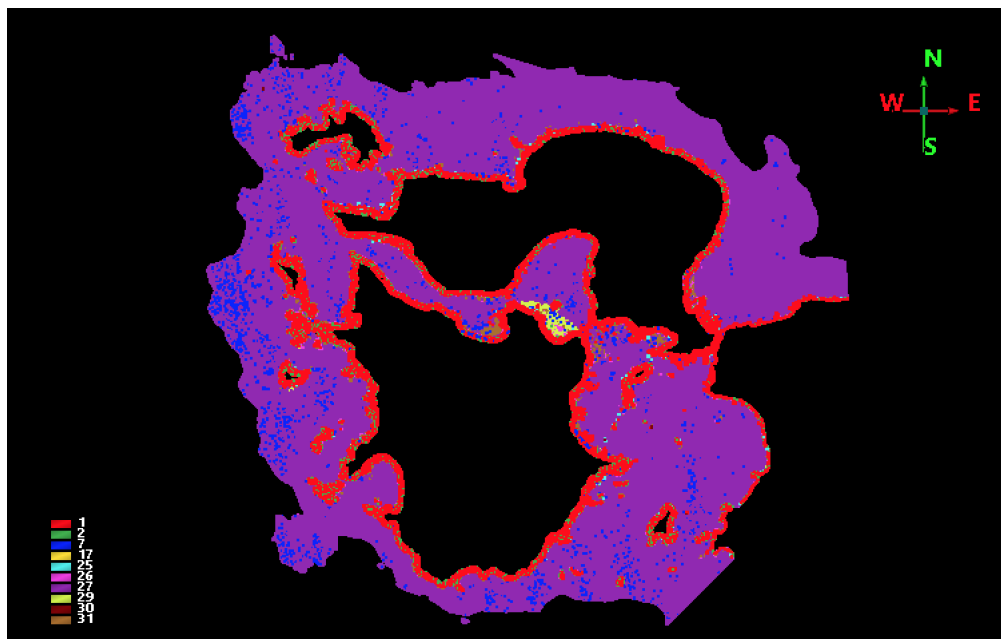


Fig. 4: Original punktsky med klassifisering

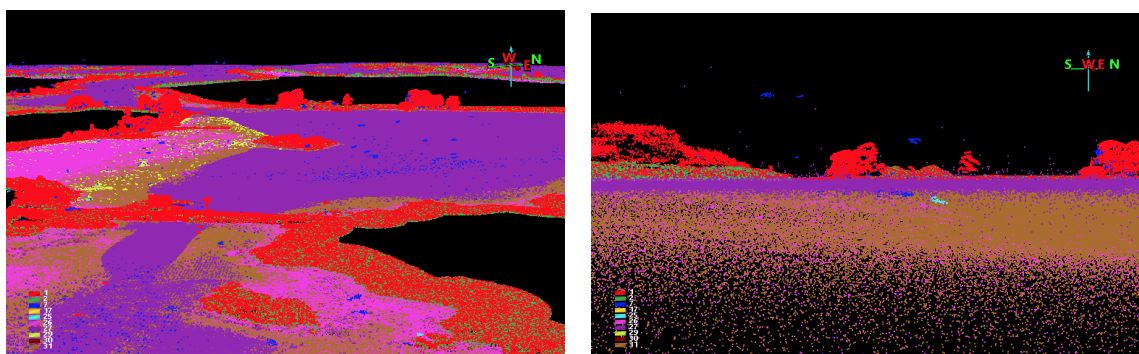


Fig. 5: Utsnitt fra original punktsky.

Table 2: Klassebeskrivelse for original punktsky

Klasse	Beskrivelse
1	Uklassifisert
2	Terreng
7	Støy
17	Bru
25	Enkeltstående stein i sjø eller elvebunn
26	Batymetriske punkt på hav og elvebunn
27	Vannoverflate (Batymetrisk)
28	Generert punkt på vannoverflaten for beregning av refraksjon
29	Marin vegetasjon
30	IHO S-57 Objekt (menneskeskapte objekter)
31	Ikke bunn (uklassifisert)

Figur 4 viser den opprinnelige klassifiseringen i punktskyen. Disse klassene er basert på klasser for standard laserdata-leveranse, og beskrivelsene av klassene er vist i tabell 2. Hva som inngår i de ulike klassene finnes mer detaljert beskrevet i prosjektrapporten.

Informasjonen som er tilgjengelig i punktskyen er:

- **X, Y, Z**
- **Intensitet**
- **Klassifisering**
- **PointSourceId**
- **UserData**
- **GpsTime**

8.2. Referansedata

I oppgaven har det blitt benyttet undervannsbilder som referansedatasett til innsamling av treningsdata. Undervannsbildene ble samlet inn av Havforskningsinstituttet i løpet av perioden 26. til 28.mai 2021. Disse bildene ble tatt med et undervannskamera som ble slept etter en båt.

Under opptak av bildene ble båtens posisjon registrert ved ulike tidspunkter. Hvert tidspunkt med tilhørende posisjon har så blitt tildelt en identifikasjon, kalt FID. Undervannsbildene ble tatt med ett-sekunds intervaller mellom disse posisjonssekvensene. De fleste posisjonssekvensene varer i omtrent 10 sekunder, men små avvik forekom i enkelte av sekvensene. Denne informasjonen har så blitt koblet til undervannsbildene i en Excel-fil hvor bildenavnet inneholder mappenavn, samt bildenavn.

Undervannsbildene ble fordelt i tre ulike mapper:

- **P1-100**
- **P2-100**
- **P2-101**

I tillegg til undervannsbildene så har det også blitt benyttet bunnsediment kart fra Norges geologiske undersøkelser (NGU). Dette datasettet ble hentet fra kartkatalogen på Geonorge.no, og det er kun lastet ned lokalt området over Fjøløy og Klosterøy. Bunnsedimentkartet er en av NGU sine maringeologiske kart, og viser ulike bunnsedimentklasser basert på kornstørrelse. Disse

klassene baserer seg på det øverste laget på sjøbunnen som strekker seg fra 0 til 10 cm. 'Datasettet er basert på tolkning av video av sjøbunnen, tolkning av digitale reflektivitetsdata, samt tolkning av analoge og digitale seismiske data. Akustiske data og video er verifisert med analyser av bunnprøver.' (Norges geologiske undersøkelse 2022)

Figur 6 viser bunn sedimentkartet over det aktuelle området. Klassene viser kun bunn sedimenter, og tar ikke hensyn til vegetasjon.

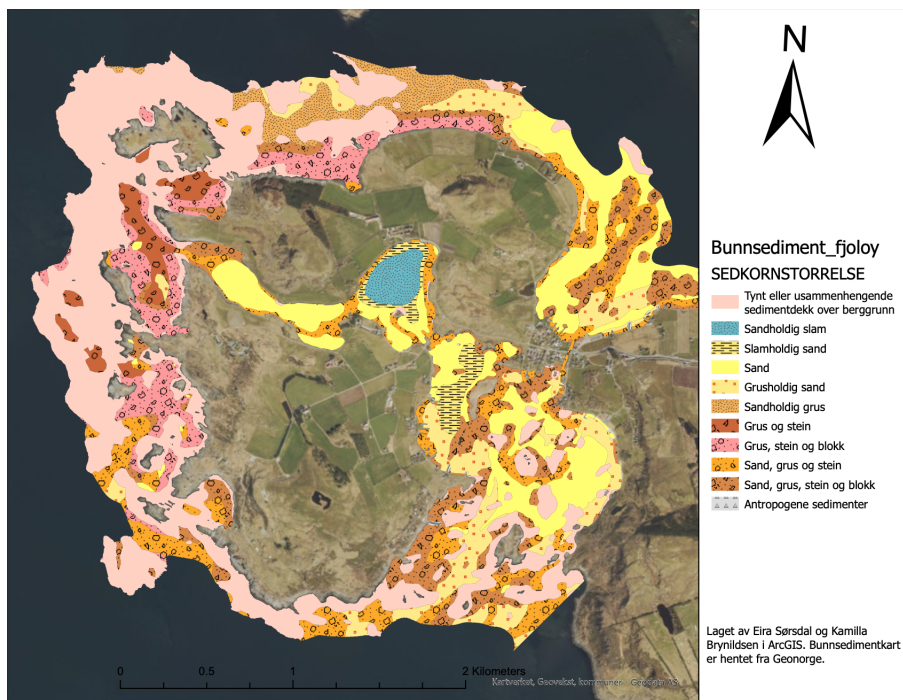


Fig. 6: Bunn sedimentkart Fjøløy

9. METODE

I dette kapitlet vil bli gitt en grundig innføring av de ulike metodene som har blitt benyttet i denne masteroppgaven. Kapitlet vil ta for seg arbeidet med innsamling av treningsdata, beregning av geometriske punktegenskaper og oppbygning av ulike tilnærminger for klassifiseringsrutiner. Metoden vil ha som hovedformål å undersøke hvordan geometriske egenskaper kan påvirke klassifisering ved bruk av maskinlæring med treningsdata basert på undervannsbilder som referansedata.

Figur 7 illustrerer et arbeidsflytdiagram som skisserer den generelle fremgangsmåten som ble fulgt i oppgaven, denne gir en oversikt over de ulike stegene for masteroppgaven.

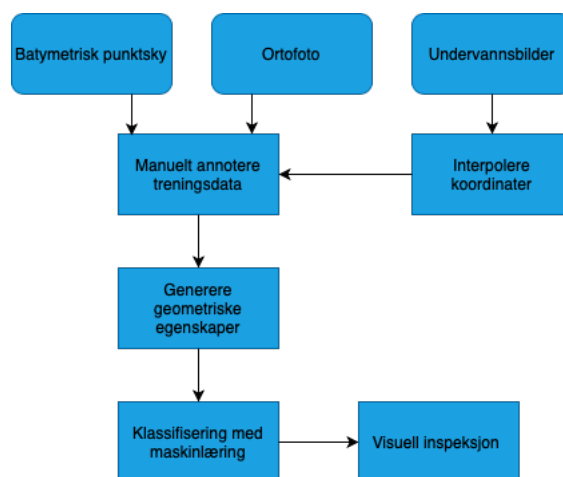


Fig. 7: Oversikt over arbeidsflyt i oppgaven

9.1. Treningsdata

Klassifisering ved hjelp av maskinlæring er avhengig av gode treningsdata med definerte etiketter. Det kreves definert data for å trene opp maskinlæringsmodellen, og definert data for å validere modellens ytelse. Derfor vil denne delen av metoden ta for seg definisjon av treningsdata, hvilke kategorier/klasseetiketter som defineres, utvalgelse av treningsområder og registrering av disse.

9.1.1. Forarbeid av referansedata

Dette kapitlet tar for seg forarbeidet av data fra havforskningsinstituttet, hvordan referansedata har blitt bearbeidet for videre bruk. Kapitlet tar også for seg valg av treningsområder, og hvordan dette arbeidet har foregått.

For å definere treningsdatasettet best mulig, ble det tatt i bruk ulike datasett som referansedata. Dette fordi det er vanskelig å definere punkter direkte fra en punktsky som er samlet inn fra batymetrisk laserskanning. I denne oppgaven ble det benyttet data fra Havforskningsinstituttet og NGU som referansegrunnlag til prosessen med manuell annotering av treningsdata.

Undervannsbildene fra Havforskningsinstituttet ble levert sammen med en tilhørende Excel-fil som inneholdt posisjoner, og informasjon om bildene som ble tatt mellom disse posisjonene. Et skript ble utviklet for å stedfeste bildene ved å interpolere koordinatene mellom disse posisjonene basert på tidsstempelen til bildene, slik at hvert bilde kunne tildeles en omtrentlig posisjon. Bildene som manglet start- eller slutt-koordinater, eller der slutt-koordinaten ble tatt på påfølgende dag, ble ikke inkludert i beregningen.

I *ArcGIS Pro* ble undervannsbildene koblet til ortofoto ved å representere bildene som punkter lagt over ortofoto. Hvert punkt inneholdt informasjon om hvilket undervannsbilde det representerte, samt tilhørende koordinater, og de ulike bildeskvensene utgjorde til sammen et spor.

De ulike sporene til undervannsbildene ble delt inn i ulike treningsområder som vist i figur 8. Hvert treningsområde bestod av ett til tre spor med bildesequenser som vist i figur 9.

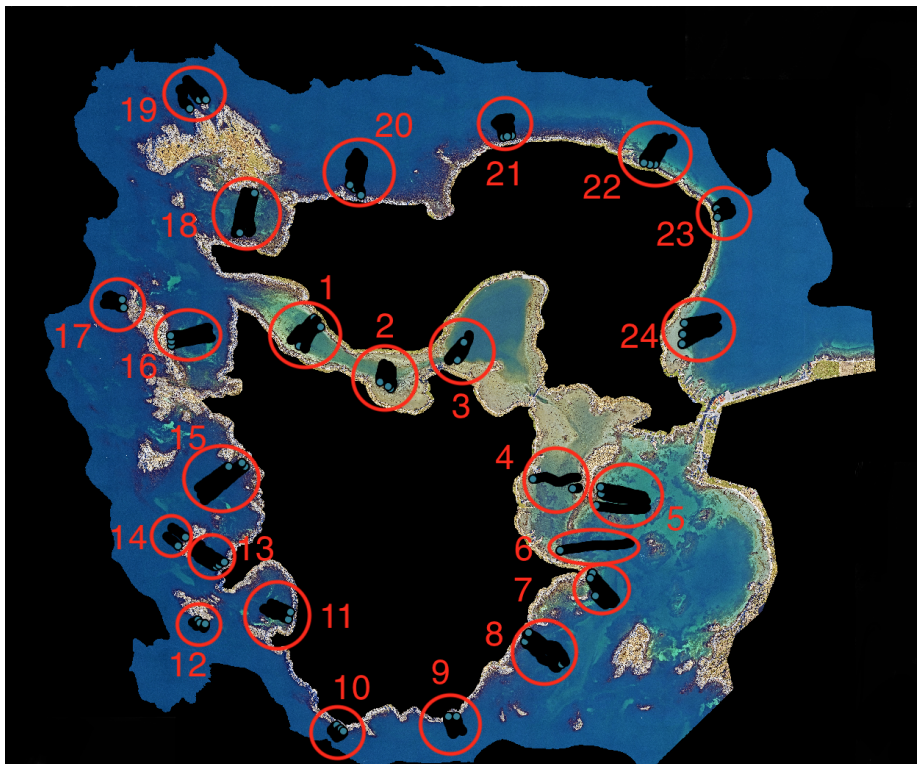


Fig. 8: Oversikt inndelte treningsområder

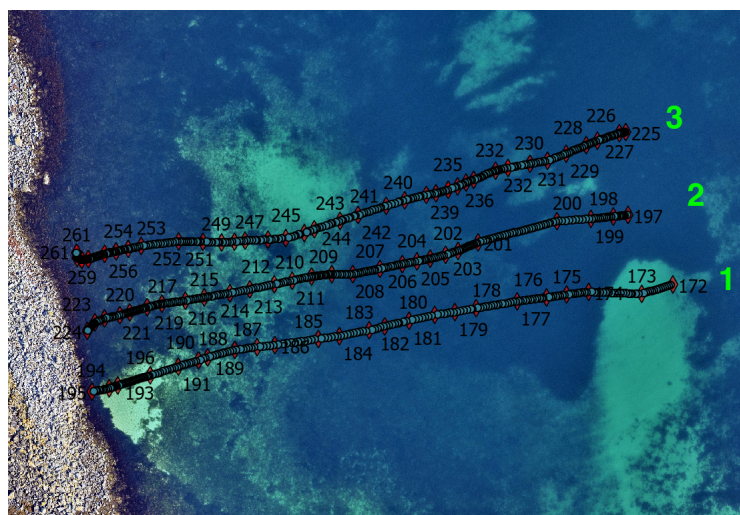


Fig. 9: Ulike spor i treningsområde 16

I figur 10 ble de interpolerte bildepunktene i de ulike sporene representert med blå punkter. De opprinnelige posisjonene som ble brukt til interpolasjon, ble vist i rødt sammen med bildesequensens ID (FID). Figur 11 viste det tilhørende undervannsbilde, markert som grønt punkt i skjermbildet. I den påfølgende figuren(12), har et undervannsbilde som var tatt litt senere i bildesequensen, fanget en steinrøys som er mulig å gjenkjenne i ortofoto i figur 10.

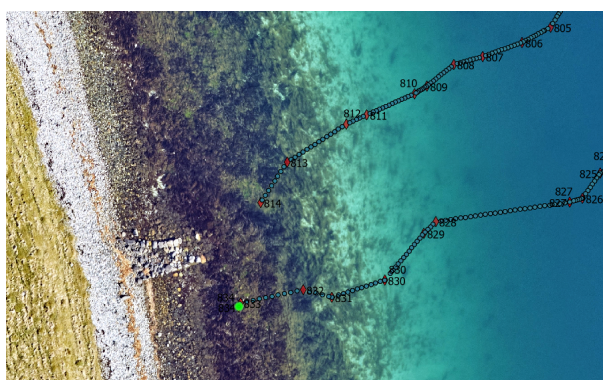


Fig. 10: Skjerm bilde ArcGIS: Viser markert punkt i grønt



Fig. 11: Undervannsbilde ved markert punkt



Fig. 12: Bilde hvor steinrøys er synlig

9.1.2. Valg av klasser og innsamling av treningsdata

Utvalget av klasser ble basert på en vurdering av flere faktorer. Dette inkluderte tolkning av undervannsbildene, hva som var mulig å detektere i punktskyen ved manuell annotering i *Terrascan*, samt en vurdering på hva en maskinlæringsmodell kunne skille mellom basert på punktettheten i den aktuelle punktskyen.

Det er verdt å nevne at da valgene om klassene ble tatt ble det brukt en annen punktsky enn den som egentlig skulle ha blitt brukt. Mer om dette kan leses i kappittel 11. Basert på større punkttetthet i den ” riktige” punktskyen, hadde det muligens blitt valgt andre klasser.








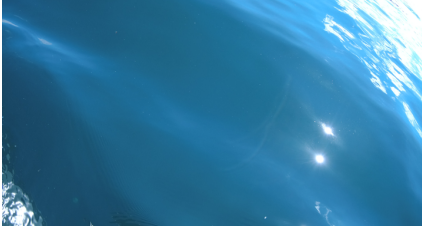



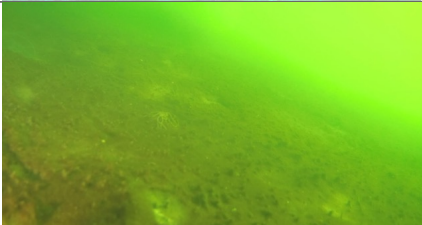


Valg av klasser falt på: *sand, spredt vegetasjon, tett vegetasjon, vannoverflate, stein og berg under vann, slam og land*. Disse klassene bestod av elementer som var gjentakende rundt øya, med unntak av slam, og som kunne representere området godt.

Slam var kun i et lokalt område på øya, men det ble valgt for at det ikke skulle bli forvekslet med vegetasjon. Vegetasjon ble kategorisert i to typer: *spredt- og tett vegetasjon*. Dette valget ble tatt med tanke på begrensingene med bruk av laser under vann. Siden det kun er grønn laser som trenger gjennom vannoverflaten er det ikke mulig å få skilt like godt mellom ulike typer vegetasjon, slik som en får med infrarød laser på land. Tett vegetasjon kan også skape skygger over bunn og annen vegetasjon, noe som resulterer i at laserstrålen ikke når det som er under. Derfor ble det inkludert en spredt vegetasjonsklasse for å muligens registrere bunnpunkter mellom vegetasjonen.

Treningsdata til klassene ble plukket ut ved å gå gjennom sporene og finne tilhørende undervannsbilder. Undervannsbildene, i kombinasjon med ortofoto og enkelte tilfeller bunnsedimentkartet, dannet grunnlaget for å avgjøre hvilken klasse ulike deler av sporet skulle tilhøre. Det ble lagt mest vekt på undervannsbildene, men i områder nærme land kunne ortofoto også benyttes. Bunnsedimentkartet ble brukt i tilfeller der det ikke var så tydelig fra undervannsbildene hvilken type bunnsediment som fantes i det aktuelle område, som for eksempel sand, stein – berg og slam.

Tabell 3 viser en oversikt over de ulike klassene med tilhørende eksempler fra undervannsbildene.

Table 3: Klasseoversikt

ID	Bunntype	Farge	Bilde
1	Sand		
2	Spredt vegetasjon		
3	Tett vegetasjon		
4	Vannoverflate		
5	Stein-berg		
6	Slam		
7	Land		

Sand

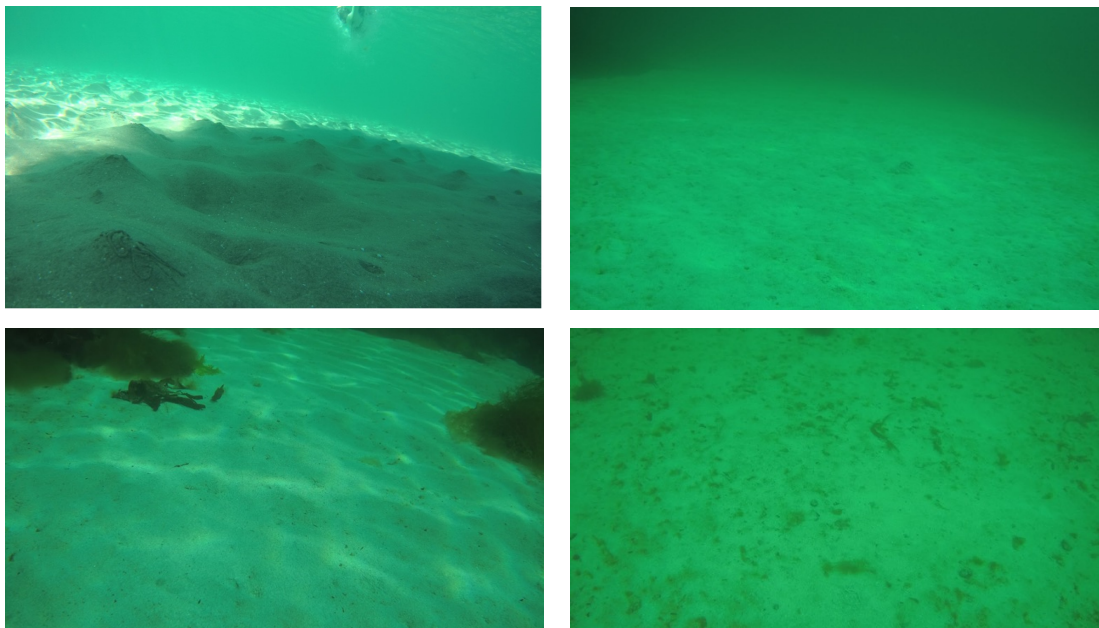


Fig. 13: Eksempler på undervannsbilder fra klassen: Sand

I figur 13 vises fire eksempler på treningsområder som ble manuelt annotert som sand. Det ble prioritert å velge områder med mest mulig ren sand, men det var flere tilfeller hvor sanden inneholdt litt vegetasjon, som fremkommer på undervannsbildet nede til høyre. For å gi klassen best mulig representasjon, ble også enkelte områder med sand som ikke var like rene valgt ut. Tilfellene hvor sandområde lå tett opp mot vegetasjon, slik som fremgår på bildet til venstre, ble de valgte områdene litt mindre. Dette for å forsikre at det ikke ble truffet punkter som bestod av vegetasjon.

Spredt vegetasjon

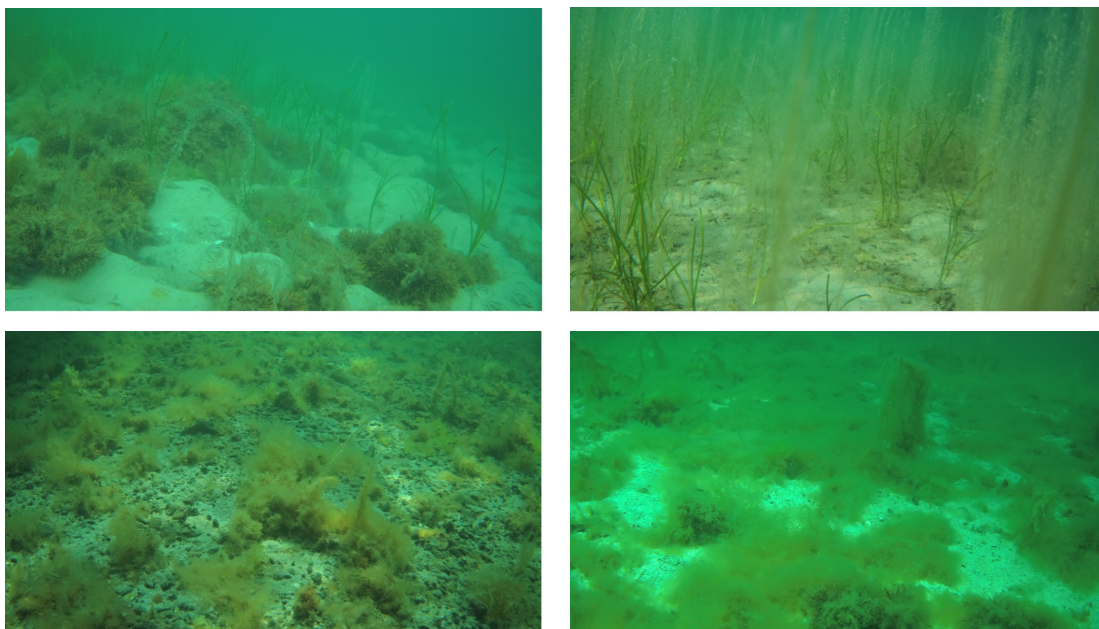


Fig. 14: Eksempler på undervannsbilder fra klassen: Spredt vegetasjon

Spredt vegetasjon ble valgt til å inneholde både vegetasjon og bunn, som i de fleste tilfeller var sand. Spredt vegetasjon strekker seg i høyde som vist i figur 14. Forekomsten av spredt vegetasjon var hovedsaklig ved grunne, flate områder eller på dyp etter et belte med tett vegetasjon.

Tett vegetasjon

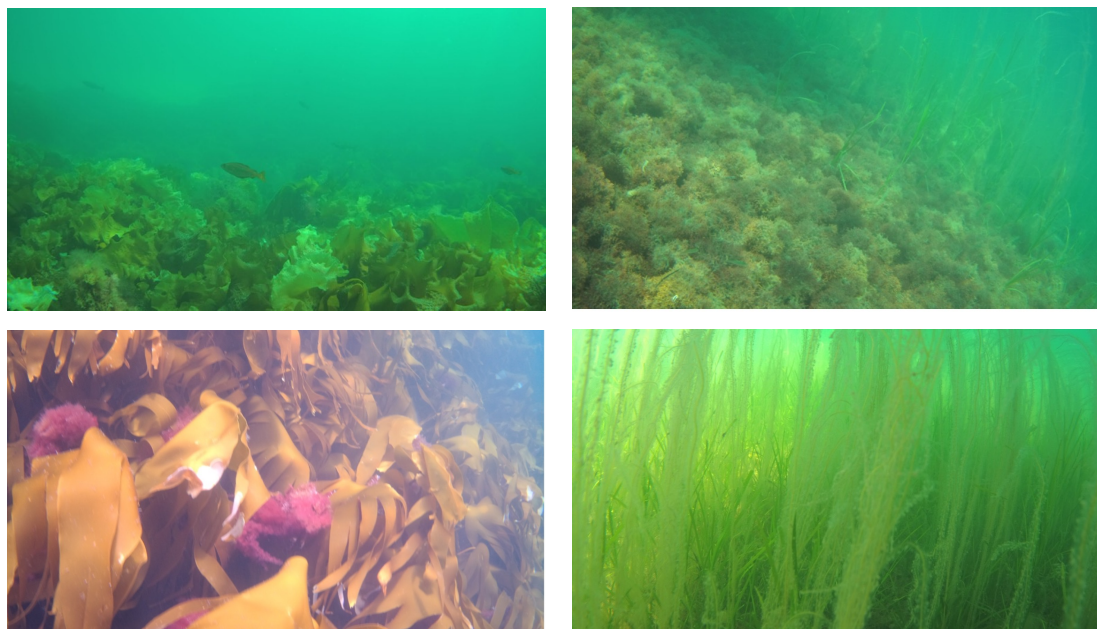


Fig. 15: Eksempler på undervannsbilder fra klassen: Tett vegetasjon

Tett vegetasjon er også en klasse som strekker seg i høyden, vist i figur 15. På vestsiden av øya er utbredelsen av tang og tare større enn på østsiden. Tett vegetasjon som tang og tare forekommer hovedsaklig nærme land, og har gjerne i slike tilfeller en helning. Ved grunnere områder på østsiden kan dette forekomme ved ulike dybder, men også områder hvor det er flatt.

Vannoverflate

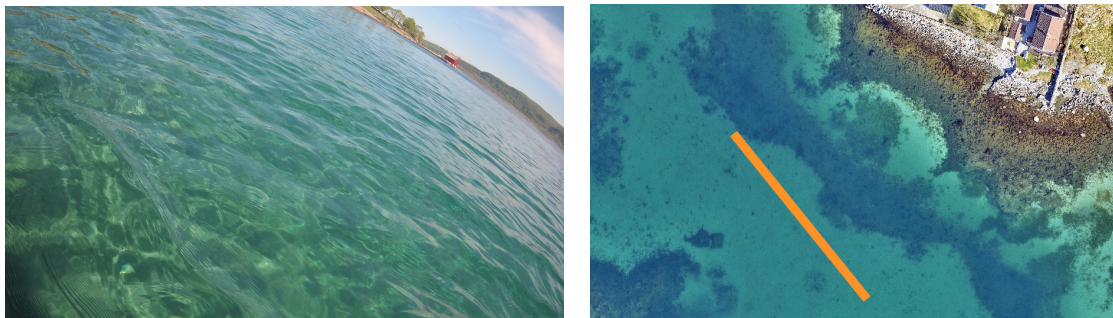


Fig. 16: Eksempler på utvalg av land

Vannoverflate ble valgt ut på forskjellige områder rundt øyene, for å kunne representere klassen best mulig med tanke på variasjoner i vannoverflaten. På vestsiden var det mer åpent vann og kunne derfor ha mer variasjoner på grunn av bølger, enn østsiden som lå mer skjermet mellom landområder. Det var også litt variasjon i forhold til vannoverflate nærmere land enn det var lenger fra land. Figur 16 viser hvordan et område for valg av vannoverflate til treningsdata ble plukket ut. Her ble koordinater til hver ende av den oransje streken plukket ut for å kunne bruke tversnittet senere i Terrascan.

Stein-berg

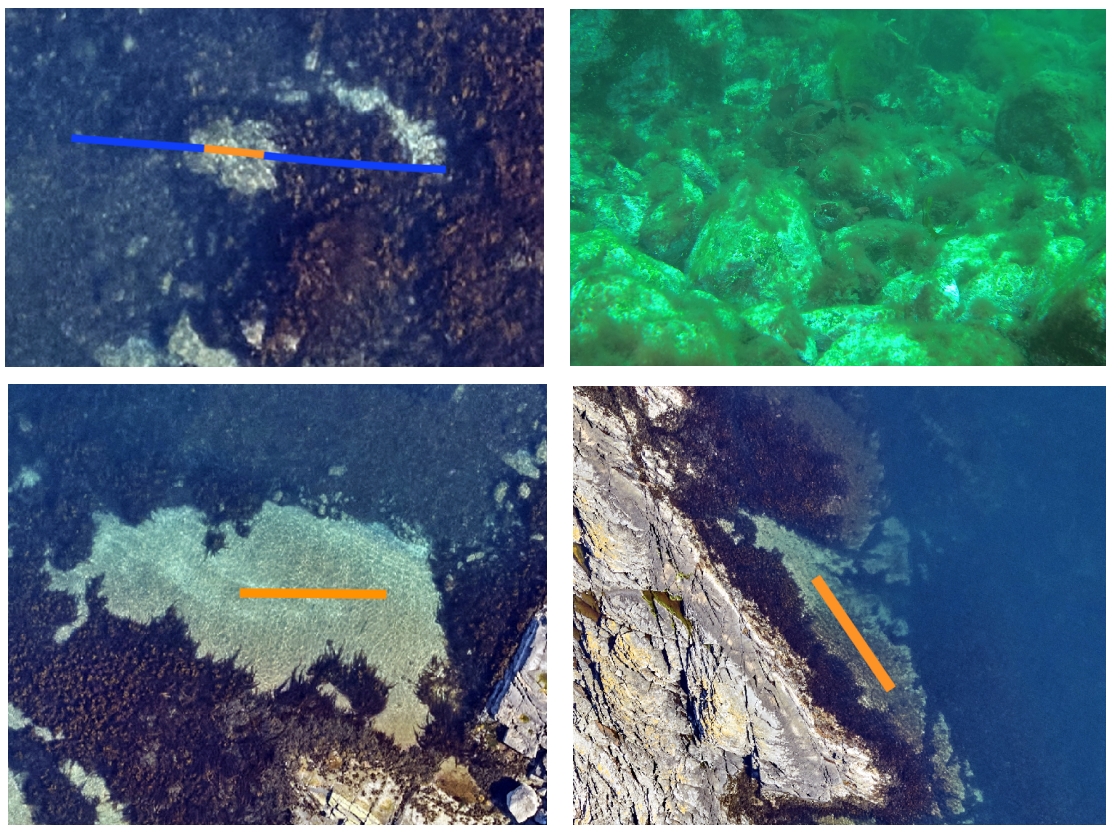


Fig. 17: Eksempler på utvalg av stein-berg klassen

Stein var den klassen det var vanskeligst å fremskaffe treningsdata til. Dette fordi det var svært få steder en kunne se stein på i undervannsbildene, samt at det er vanskelig å stedfeste steiner som enkelt objekter ut ifra bildene. Det ble her valgt inn

enkelte områder med berg for å øke punktmengden i denne klassen, som har ført til at dette har blitt hovedrepresentasjonen av klassen. Områdene med berg ble valgt ut ifra ortofoto kombinert med bunnsedimentkart. I figur 17, viser bildet øverst til høyre et steinområde som ble fanget opp av undervannsbildene. Dette området bestod av stein, som tilsammen utgjorde en større flate som det kunne velges treningsdata på.

Slam

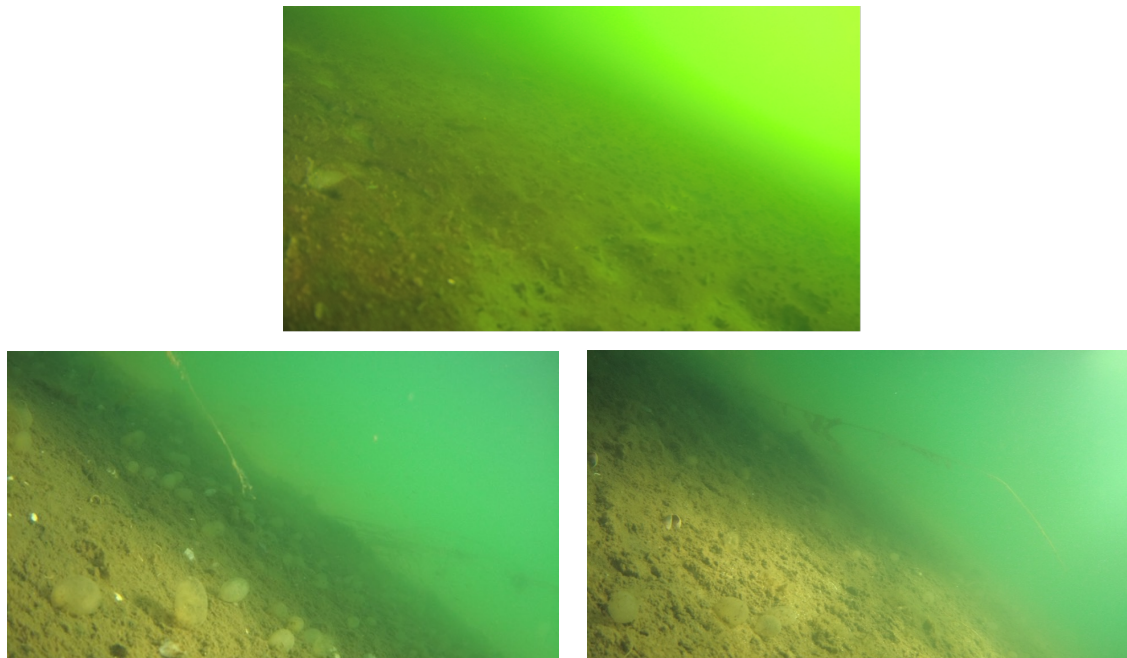


Fig. 18: Eksempler på undervannsbilder fra klassen: Slam

Forekomsten av slam var som nevnt tidligere begrenset til et lokalt område. Kartet over bunnsedimenter i figur 6 indikerer at det kun er tilstedeværelse av slam i treningsområde 3, hvor oversikten over treningsområder kan ses i figur 8. Derimot viste undervannsbildene at det også var forekomst av slam i nærliggende områder. Øverste bildet i figur 18 viser undervannsbildet i område 3, som hovedsaklig består av slam. Bildene under dette er undervannsbilder som viser slam i naboområde, definert som treningsområde 2.

Land



Fig. 19: Eksempler på utvalg av land

I landklassen ble det valgt ut ulike områder rundt øyene for å representere ulike typer av landskapet. Det ble blant annet valgt områder over små øyer, over bygg og trær for å få med ulike former og høyder slik som bildene i figur 19.

9.1.3. Registrering av treningsdata i Terrascan

Treningsområdene som er definert på bakgrunn av bilder fra havforskningsinstituttet har blitt registrert i *terrascan* ved bruk av egen dokumentasjon, **Innsamling av treningsdata i Terrascan**, et utdrag fra dette ligger vedlagt som **Vedlegg: Treningsdata 14**.

I Terrascan ble det opprettet et nytt prosjekt for å definere treningsområder. Dette gjøres ved å gå i fanen *File* og velge *new*. I Terrascan vinduet leses punktskyen inn gjennom *File* og *read collection* i menyen. Dette åpner et nytt vindu hvor punktskyen ble lagt til ved hjelp av *add folders*. Punktskyen som ble brukt for registrering er den samme punktskyen fra pilotprosjektet *Marine grunnkart i kystsonen (Terratec AS (2021))*.

Før treningsområder ble valgt, ble punktskyen rensket for eksisterende klasser. Det ble tegnet et polygon rundt hele punktskyen med verktøyet *place polygon*. Polygonen ble så valgt ved hjelp av verktøyet *choose element*, for å kunne bli brukt som det definerte området for å fjerne eksisterende klasser. Tilbake i Terrascan vinduet velges menyen *classify* og videre *inside fence*. Her ble det definert at alle tidligere klasser skal endres til klasse 0.

For å velge ut treningsområder i punktskyen ved hjelp av Terrascan, ble koordinatene for tverrsnitt fra dokumentet *Innsamling av treningsdata i Terrascan* lokalisert i punktskyen. I dette dokumentet defineres også bredden på tverrsnittet mellom koordinatene og hvilken bunntype som skal registreres. Koordinatene ble lokalisert med samsvarende koordinat i Terrascan på tredje desimal nivå, det vil si at punktene i Terrascan er registrert med millimeter nøyaktighet i forhold til de koordinatene som er gitt i dokumentet. På grunn av utfordringer med verktøyet *draw vertical section*, som ikke avgrensner tverrsnitt mellom koordinater, ble verktøyet *Place Line* brukt til å holde kontroll på avgrensningen.

Etter at en linje hadde blitt plassert kunne så verktøyet *draw vertical section* brukes for å lage tverrsnitt mellom koordinatene. Dette verktøyet lar en også definere bredden på tverrsnittet. Tverrsnittet ble satt til å bli vist i *view 2* av Terrascan vinduene, hvor verktøyet *classify using brush* ble brukt for å klassifisere punktene i **view 2**. I dette verktøyet må det defineres hvilken klasse som skal endres og hvilken klasse det skal endres til. Videre brukes et venstreklikk i *view 2* for å aktivere registreringen. Alle punkter som musepekeren dras over vil endres frem til et nytt venstreklikk som avslutter registreringen.

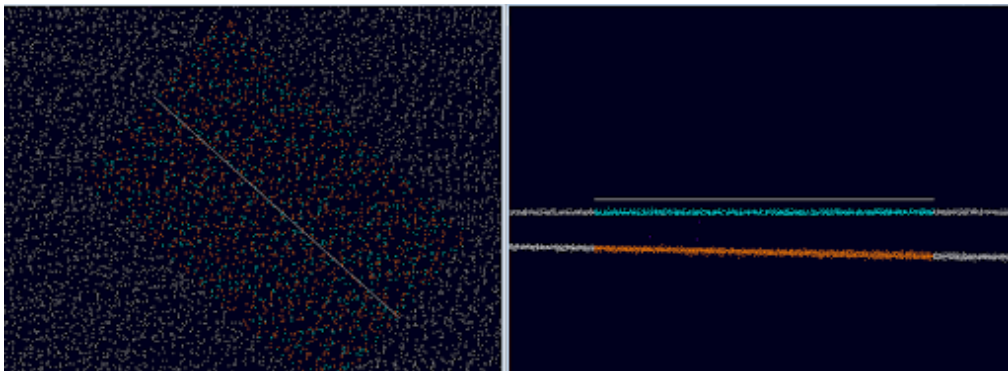


Fig. 20: Terrascan: Utvalg av sand i område 5 i spor 3

Etter Registrering av punkter for et tverrsnitt, ble dokumentet **Innsamling av treningsdata i Terrascan** oppdatert med at punktene var registrert. Fremgangsmetoden ble gjennomført gjennom hele listen med treningsområder i dokumentet. Figur 20 viser vinduene en og to i Terrascan. Her er sand og vann registrert i område 5, spor 3 mellom gitte koordinatene. Som det tydelig viser utvides tverrsnittet, linjen som har blitt satt har blitt brukt til visuell kontroll ved utvalg av punkter.

For hvert område fra figur 8 som ble fullført, ble det lagt til gruppeidentifikasjon til de klassifiserte punktene i området. Dette ble gjort ved å bruke verktøyet *place polygon* og definere et polygon rundt treningsområde. Deretter velges polygonet

ved hjelp av verktøyet *choose element*. I Terrascan vinduet velges så *Group* og *Assign groups*, og området grupperes etter det valgte området. Gruppene ble så ekstrahert som features tilhørende de klassifiserte punktene ved lagring og eksportering av punktsky. Punktskyen ble lagret i *laz*-format. Formålet med grupperingen er å kunne skille mellom de ulike treningsområdene for klassifiseringsrutinen. På grunn av variasjoner i områdene rundt øyene, skal gruppene kunne brukes senere til å få en bedre romlig fordeling av treningsdata og testdata for klassifiseringsrutinen.

9.1.4. Utdringer ved utvalg av treningsområder

Dette delkapitlet vil belyse utfordringer ved definisjon og utvalg av treningsdata. I utvelgelsen av treningsområdene måtte det tas hensyn til flere faktorer. En av disse faktorene var at undervannsbildene var tatt bak en båt som skapte forskyvninger i forhold til bildets interpolerte koordinater og faktisk posisjon da bildet ble tatt.

Denne forskyvningen var inkonsistent, og ble beregnet etter antall bilder fra en overgang som var synlig på ortofoto, til den samme overgangen dukket opp på undervannsbildene. Der det ikke var mulig å beregne seg frem til en forskyvning, for eksempel i områder med dypere vann der sjøbunn ikke var synlig på ortofoto, ble det forsikret om at den aktuelle klassen var tilstede på undervannsbildene både før og etter det utplukkede område. Dette ble gjort for å sikre at den aktuelle klassen ville havne innenfor det utvalgte området.

For å få representert de ulike klassene best mulig, ble det også forsøkt å fordele treningsdatasettet til de ulike klassene utover ulike dybder. Dette var ikke alltid mulig da det kun var tilgjengelige undervannsbilder på enkelte strekninger. Det var heller ikke alle klasser hvor det var mulig å annotere på ulike dybder, med tanke på representasjon og som nevnt tidligere med steinklassen, vanskelig å annotere stein som enkeltobjekter fra undervannsbilder.

I utvelgelsesprosessen av treningsdata var det utfordrende å tolke undervannsbildene og hvilken klasse det skulle tilhøre i flere av tilfellene. Klassen for spredt vegetasjon var den som var mest utfordrende da det ble bestemt at den skulle ha både bunnpunkter og vegetasjon, der utfordringen lå i forholdet mellom disse. Et eksempel på et dilemma er hvor lite bunn måtte til for å kunne kalle det spredt vegetasjon, mens hvor mye vegetasjon kan eksistere før det ble betegnet som tett vegetasjon. I tilfellene ved stor usikkerhet ble det besluttet å ikke ta med punkter.

Ved registrering av treningsdata i *Terrascan* var det utfordringer med å skille mulige støypunkter og faktiske punkter, og rundt det å skulle nøyaktig registrere overgangen fra land til vann. Registreringen foregikk manuelt og gjennom egen vurderings-evne. Ved stor usikkerhet har det ikke blitt registrert treningspunkter for å ikke bygge klassifiseringsrutinen på feilklassifisert data.

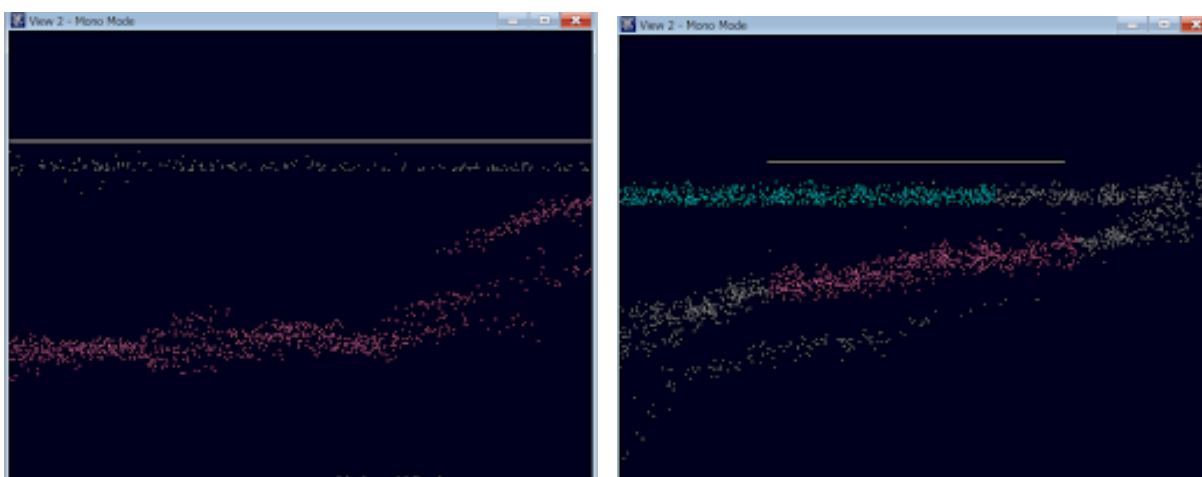


Fig. 21: Eksempel 1: Definerings av punkter

I figur 21 vises resultatet av registrering for to områder. I bildet til venstre er alle punkter innenfor tverrsnittet registrert. Her var det tydelig at det var formen til terrenget som skapte en slik spredning av punktene som en ser i bildet. For bildet til høyre

var ikke terrengformen like synlig og det var usikkert hva de underliggende punktene faktisk representerte. Disse ble dermed ikke registrert.

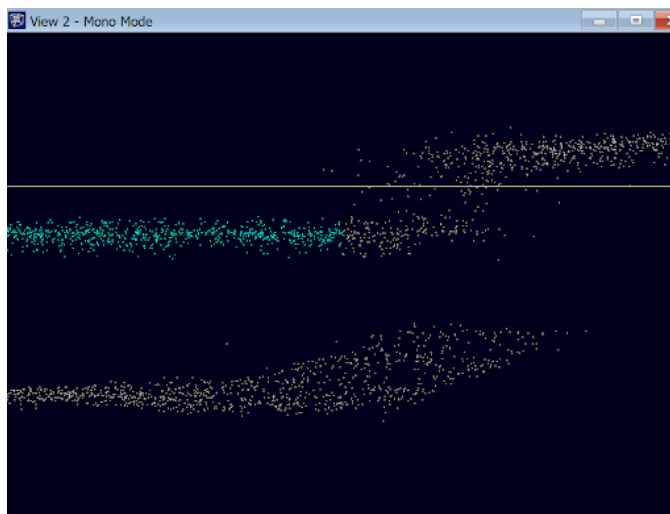


Fig. 22: Eksempel 2: Overgang fra land til vann

I figur 22 representeres en ferdigstilt registrering av vann. Her er tverrsnitt tatt i overgangen til land for å representere mer utfordrende tilfeller som kan påvirke klassifiseringen. Selv om en kan antyde en overgang i punktskyen, er det spesielt stor spredning av punkter akkurat for overgangen. Det ble her en usikkerhet i hva som skulle inkluderes som vann og ikke, som medførte at det ble tatt et valg om å ikke registrere forbi en tydelig og jevn vannoverflate.

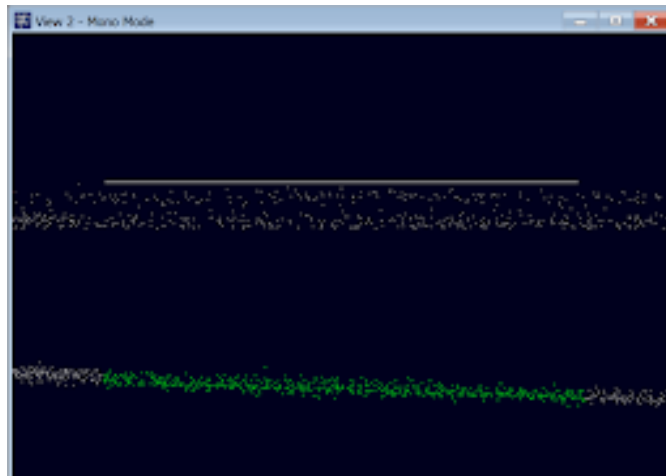


Fig. 23: Eksempel 3: Spredt vannoverflate

Et siste eksempel på utfordringer ved registrering av treningsområdene er spredning av punkter på vannoverflaten, se figur 23. Som bildet tydelig viser representerer ikke punktene på vannoverflaten en jevn og fin overflate. Det er tydelig stor variasjon i z-verdien, og en kan heller ikke se kontur av bølgeeffekter. Dette var en gjentakende observasjon på vestsiden av området, som også bli mer påvirket av vær og vind. Det er antatt at effekten kommer av at områdene er mer værutsatt. Det er likevel ikke registrert vannpunkter på slike tilfeller som i dette eksemplet. Det er heller forsøkt å finne nye områder som har en mer gjenkjennelig vannoverflate for å registrere.

9.1.5. Resultater treningsdata

Dette kapitlet gjennomgår resultatet fra treningdata som ble manuelt annotert i *Terrascan*. For videre dokumentasjon av arbeidet er utdraget fra *Innsamling av treningsdata i Terrascan* vedlagt, 14.

Figur 24 viser den romlige fordeling av treningspunktene. Punktene som ble definert som vann ble ikke tatt med i dette bildet da flere av disse ligger over punkter som er definert som bunnsklasser. Punktene ble som nevnt tidlige plukket ut etter hvor det var dekning av undervanns bilder, med unntak fra land og enkelte områder i steinklassen.

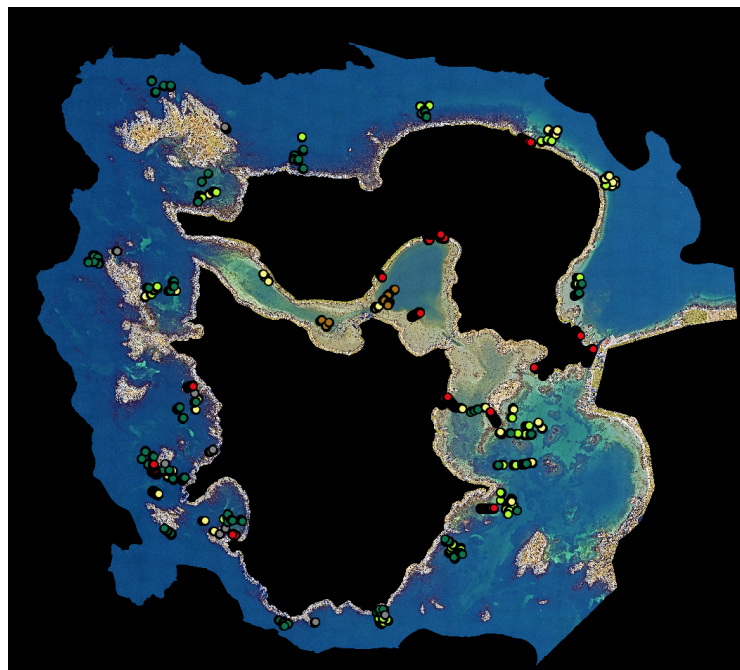


Fig. 24: Fordeling av treningsområder uten vannoverflate klasse

Fordeling av punkter over de ulike klassene fremstilles i figur 25. Det fremgår at land og vann har flere punkter enn bunnsklassene. Dette skyldes at treningsområdene for disse klassene har vært større, og derfor inneholder flere punkter. Det er også tydelig at stein, som det ikke var så mye dekning av i treningsområder, er den klassen som har fått minst punkter. Den totale mengden med treningspunkter er **449 899**.

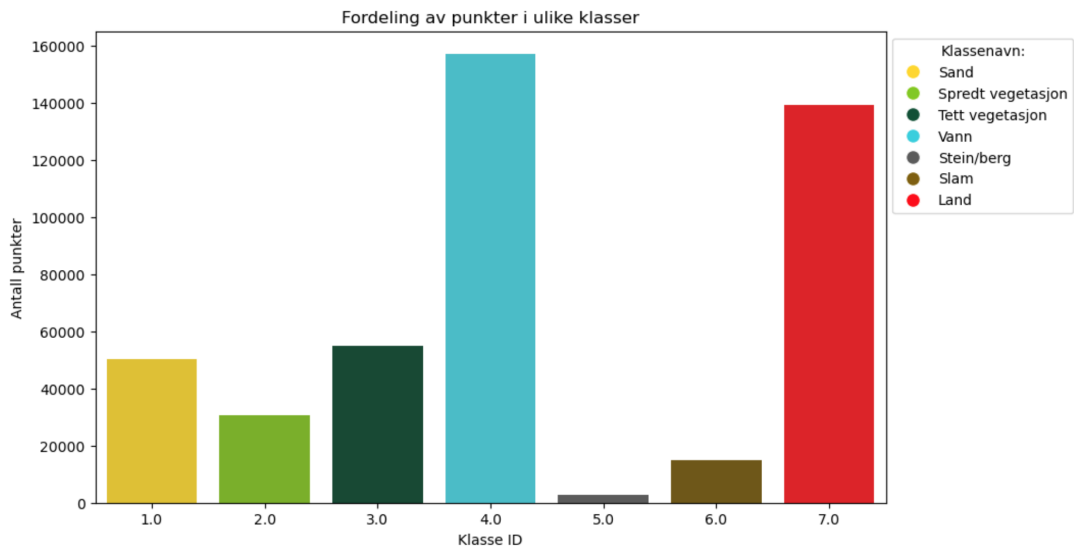


Fig. 25: Fordeling av manuelt annoterte punkter i de ulike klassene

Det ble forsøkt å fordele treningsdata over ulike dybder. Som nevnt tidligere var det ikke tilstrekkelig referansedata for alle dybder. Dybdefordeling for de ulike bunntypene illustreres i figur 26. Sand er den klassen som har blitt jevnt fordelt over ulike dybder. Spredt vegetasjon er den klassen som har fått punkter på det dypeste. Stein, slam og tett vegetasjon er klasser som har blitt samlet inn nærmere land.

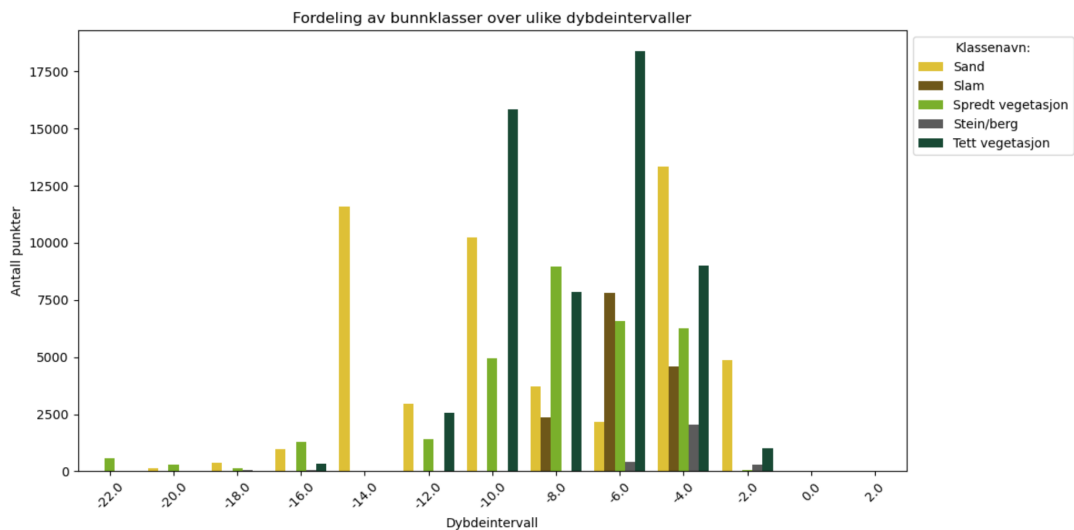


Fig. 26: Fordeling av bunntypene over ulike dybdeintervaller.

9.2. Punkttegnskaper

Punktskyen fra pilotprosjektet *Marine grunnkart i kystsonen (Terratec AS (2021))* består av x, y, z-koordinater, i tillegg til intensiteten fra retursignalet. For å kunne skille mellom punktene i en klassifisering, er det kritisk at punktene har flere egenskaper. Koordinater og intensitet alene er ikke tilstrekkelig for å differensiere punktene. Blomley et al. (2014) beskriver for en geometrisk punktskyanalyse at en enkelt observasjon kun kan bli tolket gjennom dets relasjoner til andre observasjoner.

Egenskapene som kun beskriver punktets plassering uten relasjoner til andre punkter er ikke godt nok for en klassifisering. Intensitetsverdien kan bidra noe inn mot en klassifisering, men denne verdien blir påvirket av mange ulike faktorer innen

flybåren batymetrisk laserskanning, hvor noen av disse er beskrevet i kapittel 6.1.1. Det er likevel mye informasjon som kan hentes ut kun ved hjelp av punktenes grunnleggende egenskaper, og sammenhenger mellom andre punkts observasjoner. Det kan beregnes geometriske egenskaper for punktene basert på observasjoner, disse egenskapene beskriver romlige relasjoner og kan også beskrive punktenes form.

9.2.1. CloudCompare

I CloudCompare er det generert geometriske egenskaper over hele punktskyen. Dette har blitt gjort ved hjelp av verktøyet *Compute geometric Features*. Egenskapene som har blitt generert er beskrevet i kapittel 6.2, som omhandler de geometriske egenskapene beregnet av *CloudCompare*.

Det har først blitt gjort et forsøk på å generere egenskaper for hele punktskyen samtidig. Fordelen med dette er at *octree* som blir generert, genereres for hele punktskyen, og naborelasjonene blir mer eller mindre beregnet sømløst. En stor utfordring ved denne metoden er at prosessen er svært tidkrevende og det fører også til store vanskeligheter for lagring og dermed for videre prosessering av data. Filene som lagres til videre bruk i metoden blir uhåndterlig i forhold til tilgjengelig kapasitet. Det var heller ikke mulig å lagre alle beregninger i en og samme fil.

Denne utfordringen ble løst ved å dele opp punktskyen i mindre deler, se figur 27. Ved åpning av en *laz*-fil i *CloudCompare* kan en velge å *tile* punktskyen. Dette deler opp punktskyen automatisk og lagrer delene i en egen spesifisert mappe. Størrelsen på *tiles* ble satt til 5×5 , slik at hver fil skulle bli lettere å håndtere. Siden områdene varierer en del ble det store forskjeller mellom antall punkt for hver fil. De aller største filene på 300 MB ble delt opp i to ved hjelp av samme funksjon for oppdeling. Den originale punktskyen ble totalt delt opp i 28 seksjoner.

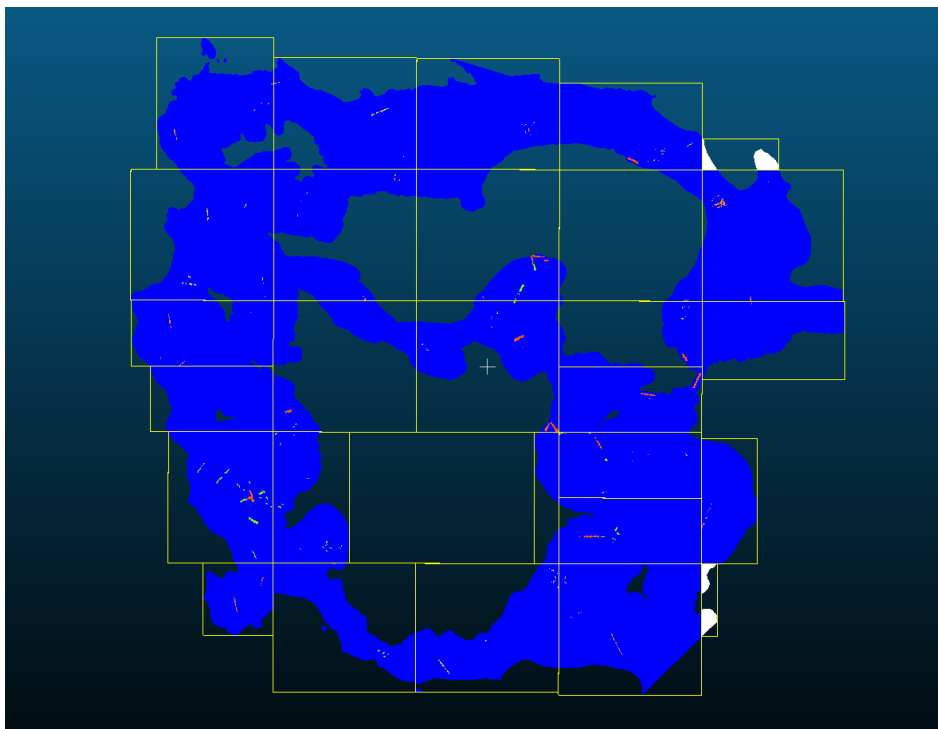


Fig. 27: Oppdeling av punktsky

En ulempe med denne oppdelingen er at den fører til at *octree* som blir beregnet ikke lenger lages for hele punktskyen sammen, men for hver av disse bolkene. Det vil si at relasjoner mellom punkter ved avgrensningene faller bort. Naborelasjoner mangler hvor avgrensningene skjer, og som kan føre til at de geometriske egenskapene får andre verdier enn det de ville ha fått

ved å bruke hele punktskyen.

Seksjonslinjene utgjør imidlertid en svært liten partisjon av punktskyen, men noen av treningsområdene splittes av linjene. Dette kan påvirke verdiene til egenskaper noe for de ulike klassene, selv om det er antatt at dette angår en svært liten del av treningsdata. Likevel var dette et nødvendig valg som var nødt til å tas, for at det skulle være kapasitet nok til å håndtere filene inn mot klassifiseringsrutinen.

Det ble lagret prosjektfiler, *.bin* filer, for hver enkelt seksjon. Noen av seksjonene ble også lagt til i samme prosjekt og kjørt samtidig, dersom filstørrelsen tillot det. Det ble generert geometriske egenskaper for ett prosjekt av gangen, med flere ulike radius. Det ble generert egenskaper for radius **0.5, 1, 1.5, 2, 2.5 og 3**, valget av disse er beskrevet i delkapittel 9.2.3. Etter at egenskapene var ferdig beregnet ble det også eksportert en *.csv* fil for hver av de ulike seksjonene.

9.2.2. Python rutine

Som et alternativ til egenskapene beregnet i *CloudCompare* ble det lagd en egen Python rutine for å hente ut punktegenskaper fra punktskyen. Denne metoden var ment som et eget tillegg av geometriske egenskaper inn mot klassifiseringen, mest for å kunne forstå sammenhengene bak egenskapene, kunne sammenlikne mot *CloudCompare* og generere andre egenskaper enn de som beregnes av *CloudCompare*.

På grunn av utfordringer i prosjektet er ikke *python* rutinen benyttet for klassifiseringen, dette beskrives mer i kapittel 11. Hovedårsaken er at kjøretiden til rutinen for den første punktskyen var lang. Den nye punktskyen hadde 120 millioner fler punkter, og det var derfor ikke fornuftig å gå videre med rutinen. Det vil likevel bli gitt en gjennomgang av den her, for å kunne beskrive andre muligheter for å generere geometriske egenskaper enn i *CloudCompare*. Rutinen bygger på naborelasjoner mellom punktene ved bruk av *kd-tree*. *Kd-tree* har flere likheter med *octree* som tidligere er beskrevet.

Et *kd-tree* har strukturen til et *binary-tree*. Det er bygd opp slik at strukturen deler opp en punktsky i mindre deler basert på punktenes romlige koordinater (Otto 2018). Hver node spesifiserer en akse og deler opp punktene basert på om punktet har en høyere eller lavere verdi for samme akse (The SciPy community 2023). Ved å strukturere punktskyen på denne måten sorteres den også basert på punktenes koordinater.

Python rutinen oppretter et *kd-tree* for hele punktskyen ved å bruke *scipy.spatial.cKDTree*, se The SciPy community (2023) for dokumentasjon. Videre leses *laz*-filer inn i deler ved hjelp av biblioteket **laspy**, (laspy 2022). Her hentes *x*, *y* og *z*-koordinatene ut, sammen med intensitet og klassifisering, og lagres til en dataframe gjennom biblioteket *Pandas* (NumFOCUS 2023). Deretter går hvert punkt i punktskyen inn i en funksjon for å beregne egenskaper.

Funksjonen tar inn indeks *i*, punkt og en liste med definerte radius. Den vil så kjøre gjennom en loop for de ulike radiusene og *scipy*'s funksjon *kdtree.query_ball_point*, The SciPy community (2023), blir brukt for å finne de nærmeste naboene til punktet gjennom det allerede definerte *kd-treet*. De nærmeeste punktene som funksjonen returnerer, basert på radius, danner et nabolag. Av dette nabolaget produseres en kovariansmatrise som benyttes til beregning av geometriske egenskaper. Punkter som ikke får nok naboer til å estimere en kovariansmatrise blir tildelt *NaN* verdier for de ulike egenskapene.

I likhet med *CloudCompare* lagres de tre egenverdiene, samt antall nabopunkter, som egenskaper i *dataframen*. I tillegg til dette inkluderer rutinen egenskaper om avstand mellom punktet og dets nabopunkter. Det er her inkludert korteste, lengste og gjennomsnittlig avstand fra punktet til dets nabopunkter. Formålet med dette var å få andre verdier som kunne beskrive fordelingen av punktene innenfor den definerte radiusen, og få en indikasjon på samlingen (*clustering*) rundt punktet.

Det er også telt opp antall punkter over og under punktet som undersøkes, disse er også inkludert i egenskaper. Hensikten her var å teste om *z*-posisjonens plassering i rommet i forhold til andre punkter kunne bidra inn mot klassifiseringsrutinen.

Rutinen returnerer til slutt to separate *csv* filer, en fil med punktene som har definert klasse, og en fil med de punktene som ikke har blitt definert. Det presiseres igjen at denne programmeringsrutinen ikke har blitt brukt videre på grunn av nevnte utfordringer.

9.2.3. Radius

Dette delkapittelet vil gjennomgå valg av radius for beregning av de geometriske egenskapene i *CloudCompare*. Det vil bli gitt en gjennomgang av tankeprosessen og utfordringer ved fastsettelse av denne.

Innledningsvis, til masteroppgaven, ble det arbeidet med en mye mindre punktsky. Et senere kapittel, kapittel 11, om utfordringer vil belyse mer av dette. På grunn av punktettheten i denne punktskyen baserte valg av radius seg på fullstendighet i datasettet. Det var en mye større spredning av punktene, slik at geometriske egenskaper ikke kunne beregnes for de minste radiusene. For å undersøke hvilke radius som kunne brukes uten å miste for mye data, ble det generert et plott for å undersøke nettopp dette, se figur 28.

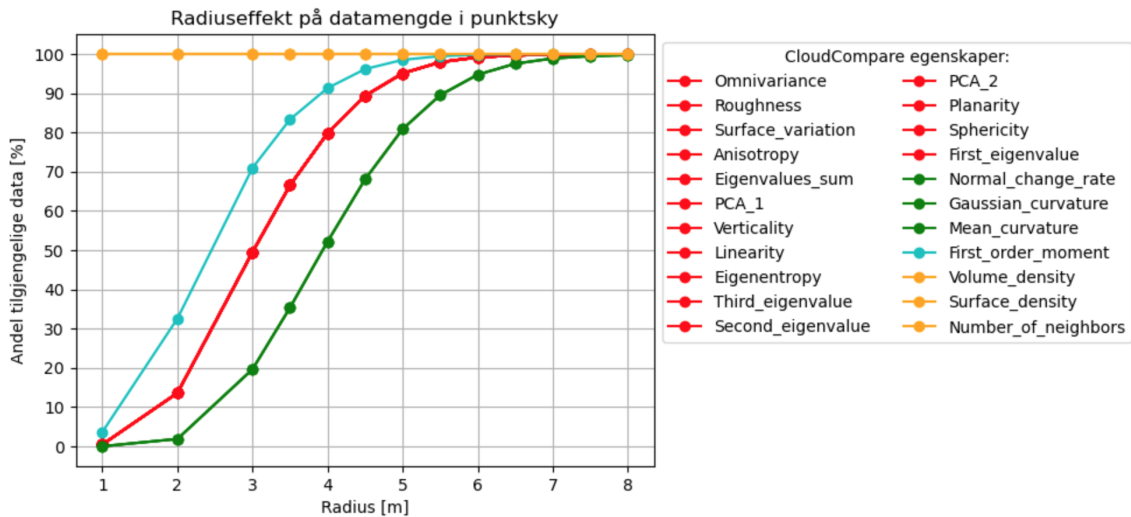


Fig. 28: Radiuseffekt fra 1m til 8m på datamende i punktsky ved Cloudcompare egenskaper

Figur 28 viser prosentdelen av punkter i hele datasettet som *CloudCompare* beregner egenskaper for, med ulike radius. Fra plottet er det tre av egenskapene som er uavhengig av radius, **Volume Density**, **Surface Density** og **Number of neighbors**. De resterende egenskapene er svært knyttet til valg av radius. Med en radius på 8 meter er det tydelig at *CloudCompare* klarer å beregne geometriske egenskaper for hele punktskyen, da alle de geometriske egenskapene dekker 100% av punktskyen og det ikke er mangel på data.

Valg av radius for bestemmelse av naborelasjoner avhenger av hvilke detaljer en ønsker å fange opp (Stella 2023). Egenskapene som representerer punktene er svært bundet til radius, og ulike radius kan gi ulike fremstillinger av et punkt eller området rundt punktet. De lave radiusene beskriver mindre detaljer og lokale variasjoner i punktskyen. Etterhvert som radiusen øker, øker også området som punktet beskriver og egenskapen blir mindre detaljert for selve punktet og gir en mer gjennomsnittlig beskrivelse.

Havbunnen rund Fjøløy og Klosterøy som benyttes har mange fine og små detaljer. Det er ønskelig å kunne skille mellom sand, spredt vegetasjon og tett vegetasjon. Grensene mellom disse ulike bunntypene er viktige å prøve å fange opp. For å unngå glatting av punktskyen, eller glatting av egenskapene til punktene, forsøkes det å unngå store radius som fører til at egenskapene blir mer gjennomsnittlig. Det er ønskelig at egenskapene er mest mulig på detaljnivå for å kunne oppdage de brå overgangene mellom bunntypene.

På bakgrunn av påvirkningene av størrelsen på radius som er beskrevet her, var det ønskelig å prøve å redusere radiusen slik at området hver egenskap beskrev ble redusert, men dette var problematisk fordi en reduksjon av radius ville medført tap av punkter inn mot klassifiseringen.

Ved et senere tidspunkt i masteroppgaven ble det oppdaget at arbeidet hadde foregått for feil punktsky. Den egentlige

punktskyen som skulle benyttes har en helt annen punktetthet. Siden denne punktettheten var mye høyere, ble det besluttet å justere ned radiusene og forsøke å undersøke radiuseffekten på nytt. Dette var for å komme på mer detaljnivå av områdene, og den nye punktettheten viste seg å håndtere lavere radius.

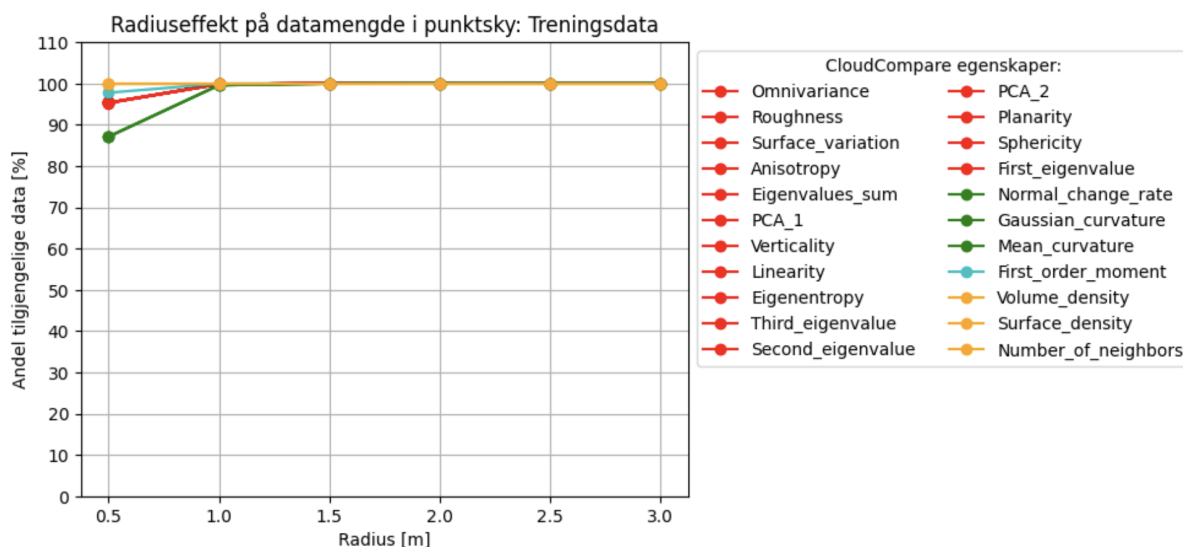


Fig. 29: Radiuseffekt fra 0.5m til 3m for treningsdata på fullstendig punktsky

Figur 29 viser prosentandel på punkter som *CloudCompare* klarer å beregne geometriske egenskaper for. Plottet representerer kun treningsdata, da denne er viktigst å beholde så mye data som mulig fra inn mot klassifiseringen. Det er tydelig stor forskjell mellom de ulike punktskyene. Den fullstendige punktskyen tillater å kunne benytte en radius på 0.5m og fortsatt beholde 85% av treningsdata.

Størrelsen på radius bestemmer hvor små og store detaljer som ønskes å beskrives av egenskapene. Likevel er det flere betraktninger som det videre må ta hensyn til. Selv om det viser seg at en radius på 0.5m kan benyttes på den nye punktskyen må punktetthet også tas i betraktning.

Punktettheten til punktskyen varierer ved dybde. Fra prosjektrapporten Terratec AS (2021) er punktettheten gitt til $5pkt/m^2$ ved sjøkartnull, $3pkt/m^2$ ned til 5 meter under sjøkartnull og $1pkt/m^2$ ned til 10 meter under sjøkartnull. I tillegg til punktetthet basert på dybde er det også overlapp i flystripene som gir flere områder overlapp og høyere punktetthet. Fordelingen av punktetthet er visualisert godt i figur 30.

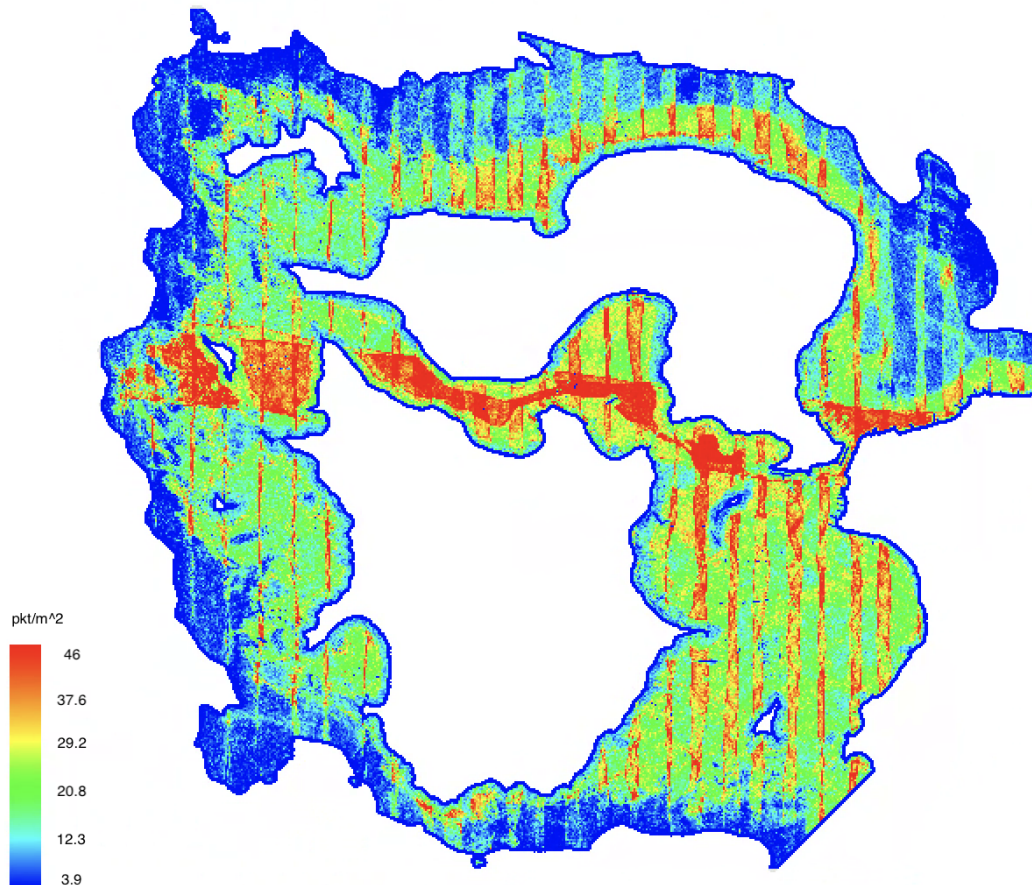


Fig. 30: Punktetthet til fullstendig punktsky i gitt i pkt/m^2

For områdene med høyere punktetthet kreves det ikke like høy radius for å beregne de nærmeste naborelasjonene, mens for områder med lav punktetthet kreves det høyere radius for å beregne nabopunkter. Dette innebærer at punkter på grunne områder eller punkter der det er mye overlapp fra flystripene ikke krever like stor radius for å kunne beregne geometriske egenskaper i *CloudCompare*, som det punkter på dypere områder gjør. For å få med nok punkter med geometriske egenskaper på dypere områder må radiusen økes noe. Dette igjen kan påvirke punktegenskapene på de grunneste områdene, hvor de blir mer områdebeskrivende enn detaljbeskrivende.

I likhet med disse betraktningene kan det også lønne seg å inkludere flere radius inn mot klassifiseringsrutinen. Slik som det beskrives i Blomley et al. (2014) er et punkts geometriske egenskaper avhengig av relasjonene til punktene rundt og sannsynligheten for at de tilhører en spesifikk klasse. De ulike klassene kan også vise karakteristisk sammenheng ved ulike romlige skala. Blomley et al. (2014) beskriver videre også at en tilnærming med flere ulike skala for beregning av geometriske egenskaper kan forbedre klassifiseringsresultatet.

Det er valgt å inkludere radius **1, 1.5, 2, 2.5 og 3** for å beholde omtrent 100% av treningsdata slik som plottet viser i figur 29. Flere radiusstørrelser kan være nyttig for en klassifisering. Størrelsene beskriver også ulike størrelser for et område rundt hvert punkt. I tillegg er det forsøkt å ta hensyn til endring av punktetthet med dybde.

9.3. Maskinlæringsrutine

Dette delkapitlet i metode vil ta for seg klassifiseringsrutinen ved hjelp av maskinlæring. Hensikten er å klassifisere punktskyen ved bruk av treningsområdene som bygger på referansedata fra havforskningsinstituttet. Det vil først bli gjennomgått hvordan forberedelse av data har foregått, deretter presenteres inspeksjon og rensing av data.

Videre presenteres metodikken bak utvalg av egenskaper og oppdeling av treningsdata som er en del av klassifiseringsrutinen. Avslutningsvis for delkapitlet presenteres selve klassifiseringen, som består av flere ulike tilnærminger.

I dette kapitlet vil det bli presentert flere figurer som inneholder de ulike klasse ID, og ikke klassens navn. For en oppfriskning av hvilke ID som representerer de ulike klassene kan ses i oversikten i tabell 3.

9.3.1. Forberedelse

Blokkene av punktskyen, med beregnede geometriske egenskaper, ble lagret i CSV-format. Disse blokkene inneholdt både treningsdata og resten av punktskyen. For å hente ut treningsdata og resterende punktsky som skulle predikeres, ble det utviklet et skript for å separere data inn i ulike filer. Skriptet gikk gjennom mappen med de ulike CSV-filene, leste dem inn som *pandas dataframe*, sorterte dem og lagret til nye filer basert på om de inneholdt klassifisering eller ikke. For hver blokk ble det opprettet en CSV-fil som kun inneholdt treningsdata, og en CSV-fil som skulle brukes som ny data inn i klassifiseringen. Etter at treningsdata var separert fra hver blokk, ble disse samlet i en CSV-fil som ble brukt inn i klassifiseringsrutinen.

9.3.2. Inspeksjon og rensing

Før treningsdatasettet inngikk i klassifiseringsrutinen ble det foretatt en inspeksjon for å undersøke omfanget av manglende data, og hvor disse var lokalisert. For å etablere best mulig grunnlag for klassifiseringsrutinen, ble målet for inspeksjonen å beholde mest mulig av treningsdatasettet.

Fra kapitlet 9.2.3 ble det besluttet å kun beholde radius **1, 1.5, 2, 2.5 og 3**. Etter at egenskapene med radius 0.5 ble fjernet, inneholdt treningsdatasettet totalt **19563** manglende verdier fordelt over **1975** rader. Treningsdatasettet totalt inneholdt **449899** rader. Det ble først undersøkt hvilke klasser de manglende verdiene tilhørte. Figur 31 illustrerer hvilke klasser som inneholder manglende verdier for egenskapene som var beregnet i *CloudCompare*. Flertallet av de manglende verdiene tilhørte klasse 4, vannoverflate og klasse 7, land.

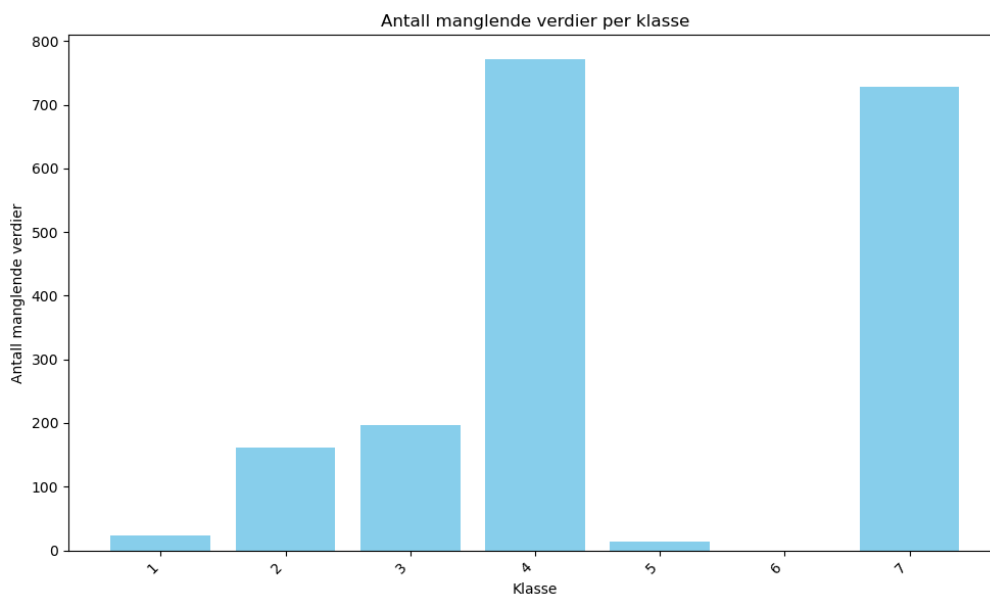


Fig. 31: Manglende verdier i de ulike klassene

Basert på tidligere undersøkelse i figur 25, fremgikk det at det var de aktuelle klassene som også inneholdt flest punkter. Basert på at klasse 4, vannoverflate og klasse 7, land inneholdt flest manglende verdier, hadde flest treningspunkter og at det var bunntypene som var av størst interesse, ble det i første omgang besluttet å fjerne rader med manglende verdier i disse klassene.

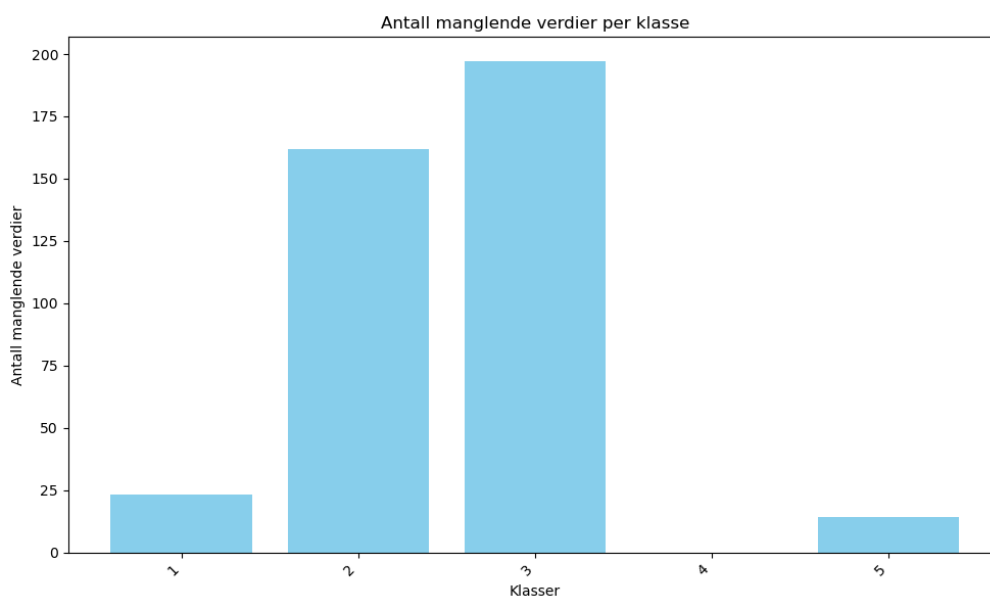


Fig. 32: Manglende verdier i gjenværende bunntyper

Figur 32 illustrerer manglende verdier for de resterende klasser etter at det ble fjernet manglende verdier for vannoverflate og land. Den totale mengden manglende verdier ble så redusert til **2834**, fordelt over **396** rader.

Videre ble det undersøkt hvor stor prosentandel av de ulike klassene disse manglende verdiene utgjorde. I figur 33 fremkommer det at de resterende verdiene som mangler utgjør mindre enn 1% av den totale datamengden i hver klasse. Basert på

denne informasjonen ble det besluttet at resterende **396** rader med manglende verdier kunne fjernes, uten at det skulle ha så stor innvirkning på datasettet.

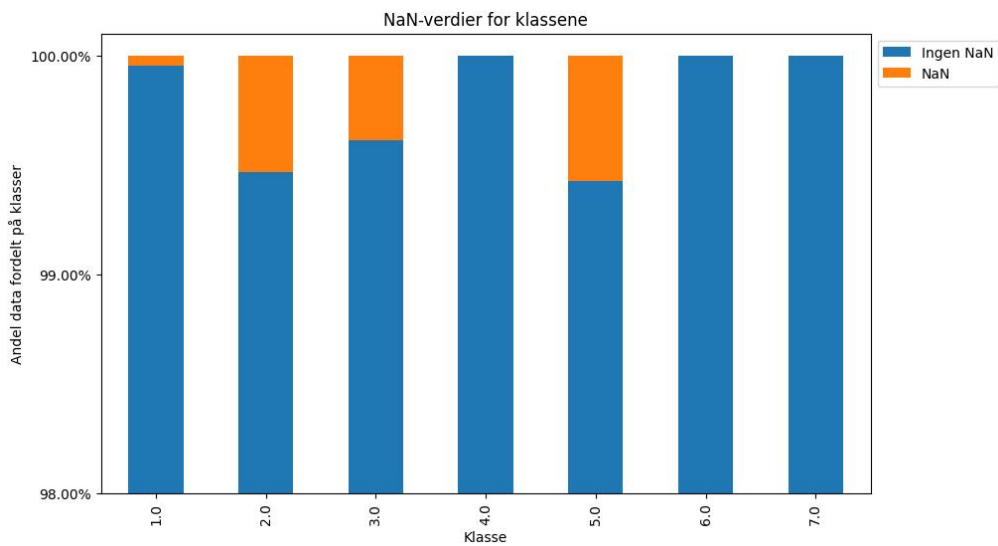


Fig. 33: Prosentandelen av gjenværende manglende verdier etter fjerning av manglende data i klasse 4 og 7.

9.3.3. Utvalg av egenskaper

I dette kapittelet blir det gjennomgått prosessen for utvalget av egenskaper. I *CloudCompare* ble det beregnet 22 egenskaper per radius, og med 5 ulike radius fører dette til at dimensjonen på datasettet blir ganske stort. Stor dimensjonalitet på datasettet kan påvirke tiden det tar å utøve klassifiseringen, men det kan også påvirke ytelsen til modellen. Ulike egenskaper har også ulik viktighet for en maskinlæringsmodell, det er derfor viktig å utføre et riktig utvalg basert problemstilling og datagrunnlaget som er tilgjengelig.

Det finnes flere ulike metoder for å velge egenskaper, og i oppgaven ble det benyttet en kombinasjon av *Feature Importance* som er innebygd funksjon i *Random Forest* algoritmen og funksjonen *SelectKbest*. Mer informasjon om *Feature Importance* og *SelectKbest* finnes ved [scikit-learn developers \(2023c\)](#) og [scikit-learn developers \(2023a\)](#).

Selve utvalget baserte seg på en vektet og ikke vektet versjon av *Random Forest* sin *Feature Importance* og *SelectKBest*. Funksjonen valgte hver for seg 40 egenskaper som hadde høyest viktighet, eller høyest verdi for modellen. Listen over disse egenskapene ble sammenlignet, og det var totalt **20** egenskaper metodene hadde til felles.

Felles egenskaper blant de ulike utvalgsmetodene:

- Z
- Intensity
- Surface density (r=1.5)
- Verticality (2.5)
- Number of neighbors (r=3)
- Surface density (r=2)
- Verticality (3)
- Volume density (r=2)
- Volume density (r=3)
- Number of neighbors (r=2.5)
- Verticality (1)
- Number of neighbors (r=1.5)
- Surface density (r=2.5)
- Surface density (r=3)
- Volume density (r=1)
- Number of neighbors (r=2)
- Verticality (1.5)
- Volume density (r=2.5)
- Verticality (2)
- Volume density (r=1.5)

Disse egenskapene ble så benyttet inn i enkelte av forsøkene som ble utprøvd klassifiseringsrutinen. Blant disse egenskapene er det mange som baserer seg på punkttetthet, samt at det er flere egenskaper som er gjentatt med ulik radius. Det har også blitt forsøkt med andre egenskaper, noe som beskrives senere i kapittel 9.3.5.

Det er verdt å nevne at intensitet har inngått som egenskap i klassifiseringen. Det har ikke i denne oppgaven blitt tatt hensyn til at intensitetsverdien påvirkes av dybde og andre faktorer som beskrevet i 6.1.1.

9.3.4. Oppdeling av treningsdata

I en maskinlæringsprosess deles treningsdata vanligvis opp i et treningssett og testsett, kapittel 6.3, for å kunne validere og evaluere maskinlæringsmodellen. I dette delkapitlet blir prosessen rundt oppdelingen av treningsdata gjennomgått.

I startfasen var det usikkert om det eksisterte tilstrekkelig treningsdata for å kunne sette av deler til et testsett. Det er derfor senere i metoden for klassifisering gjennomført en rutine uten en slik oppdeling. Resultatet fra denne blir også presentert. Likevel, uten et testsett, kan ikke prestasjonen til klassifiseringsmodellen vurderes. Derfor ble det viktig å inkludere et testsett for den videre tilnærmingen til oppgaven.

Et viktig aspekt ved oppdeling av treningsdata fra treningsområdene er romlig fordeling. Området som punktskyen beskriver har store variasjoner. Det er variasjon i forhold til om punktene befinner seg på vestsiden eller på østsiden av punktskyen. En klasse definert på østsiden ser ikke helt lik ut som samme klasse definert på vestsiden. I tillegg til dette er det variasjon i dybde og punkttetthet, slik at en klasse definert på et grunt område kan ha ulike geometriske egenskaper fra en klasse definert på et dypt område. Punkttettheten vil også bidra til forskjeller mellom de beregnede egenskapene.

For utførelsen av klassifiseringen har det dermed vært ønskelig å dele opp treningsområdene i treningsdata og testdata gjennom romlig fordeling. En fordeling som forsøker å inkludere alle scenarier i begge datasett. For å oppnå dette ble det ved definisjon av treningsdata, som tidligere nevnt, lagd grupperinger av treningsområdene. Oversikten over inndelingen er som vist i **figur 8**. Hensikten med dette var å kunne definere de ulike områdene som punktene befant seg i, og bruke disse grupperingene til romlig fordeling mellom treningssettet og testsettet.

Det ble opprettet en rutine som kunne dele opp treningsområdene i treningsdata og testdata basert på både gruppe og klasse. Først splittes treningsområdene opp med hensyn på gruppe, slik at treningsdata og testdata begge skal være representert av hver enkelt gruppe. Etter dette tas det hensyn til at klassene har ulik representasjon i hver enkelt gruppe, og punkter har blitt plukket ut slik at hver klasse innenfor hver gruppe blir representert. Størrelsen på fordelingen avhenger av hva en velger som teststørrelse for å sette av til testdata. De resterende punktene fra fordelingen forblir treningsdata.

Fastsettelsen av teststørrelsen er en viktig faktor. Som nevnt tidligere var det knyttet bekymring til om det eksisterte nok punkter for treningsområder til å benytte testdata. Det er kun treningsdata som trener opp klassifiseringsmodellen, og testdata benyttes kun til å evaluere og validere modellens ytelse etter at modellen er trent. Dette innebærer at treningsområder som settes av til testdata ikke kan brukes til å trene opp modellen. Det er derfor viktig å finne en balanse for teststørrelse, slik at det både er nok data med tanke på mengde og kvalitet for å validere modellen, men også nok data for å trene opp modellen.

Gruppering av treningsområder i *terrascan* var et godt utgangspunkt for romlig fordeling. Som et resultat av beregning av geometriske egenskaper i *CloudCompare* var ikke en slik oppdeling lenger mulig. Egenskapen for gruppering, knyttet til *laz*-filen, eksisterer i den fullstendige punktskyen, men for å estimere egenskaper for punktskyen ble den seksjonert opp i mindre deler. Som et resultat av dette eksisterer ikke lenger gruppe som en egenskap til punktene. Funksjonen *tiling* i *CloudCompare* har ikke tatt hensyn til den ekstra egenskapen i *laz*-formatet, til tross for at den ble valgt ut.

Det var likevel ønskelig med en god fordeling av klassene i treningsområdene mellom treningsdata og testdata. For å oppnå en god fordeling ble flere *python* bibliotek undersøkt for ulike funksjoner for oppdeling. Valget for funksjon for å dele opp treningsområdene falt på *StratifiedShuffleSplit* fra *scikit-learn* biblioteket, se *scikit-learn developers* (n.d.) for dokumentasjon.

StratifiedShuffleSplit deler treningsområdene inn i treningsdata og testdata, men passer samtidig på at den bevarer fordelingen av klassene fra treningsområdene (*scikit-learn developers* n.d.). Funksjonen omrokker rekkefølgen til dataen tilfeldig,

før den deler opp hele datasettet i biter. Dersom antall oppdelinger er mer enn 1, kan en risikere overlapp mellom punkter i treningsdata og testdata. Den er derfor satt til å være 1 for å unngå dette. Teststørrelsen kan også defineres, og verdien for denne presenteres i kapittel 9.3.5. Samt har `random_state` blitt satt til 42, denne funksjonen sørger for at dersom rutinen kjøres på nytt, vil samme resultat oppnås. Dette tillater sammenlikning mellom bruk av ulike parameter.

Et eksempel ved bruk av `StratifiedShuffleSplit`. Her brukes 1 oppdeling, `random_state` er satt til 42 og en teststørrelse er definert til 0.2. Dette vil si at 20% av treningsområdene skal fordeles til testdata og de resterende 80% forblir treningsdata. I dette eksemplet består treningsområdene av 447924 punkter totalt. Etter oppdeling av data ved hjelp av `StratifiedShuffleSplit` består treningsdata av 358339 punkter og testdata av 89585 punkter. Dette gir en fordeling på 79.99% og 20.00% for treningsdata og testdata.

Fra eksemplet er klasserepresentasjonen basert på andel for treningsdata og testdata representert i figur 34. Til tross for at mengde data i hver av de er ulike, er klassene mer eller mindre likt representert.

Eksempel: Oppdeling av treningsområder

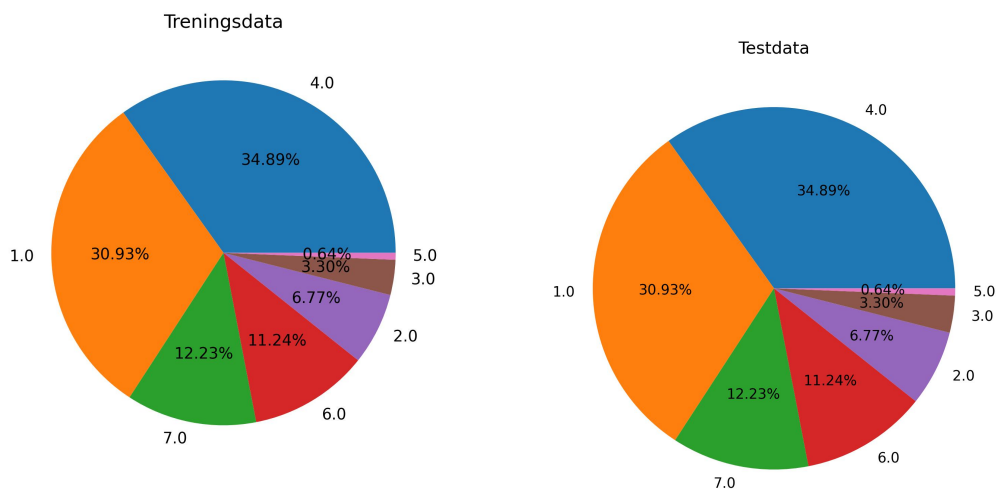


Fig. 34: Eksempel: Klassefordeling mellom treningsdata og testdata

9.3.5. Oppbygning av `Random Forest` rutine

Det vil her bli gitt en gjennomgang av klassifiseringsrutinen i `Python`. Klassifisering er gjort med flere ulike tilnærminger med `RandomForestClassifier`. Det vil først bli presentert hvordan den generelle rutinen er satt opp, uten spesifikasjoner av parametere og hvilke data som er brukt. Deretter presenteres de ulike klassifiseringsrutinene og deres spesifikasjoner. Disse vil videre bli gjennomgått, samt fremgangsmetoden for valg av de ulike rutinene.

Klassifiseringsrutinen leser først inn en samlet `csv` fil til en `dataframe` for treningsdata ved hjelp av `pandas`, (NumFOCUS 2023). Videre droppes `NaN` verdier fra `dataframen` og eventuelt annen data som ikke skal inkluderes. Treningsdata deles opp i `X` og `y`, dette er for å definere egenskaper og målverdi. `y` blir bestående av kolonnen `Classification` fra `dataframen` som er de definerte klassene, og `X` beholder de resterende kolonnene som består av egenskaper, utenom `Classification` og `x-` og `y-`koordinater.

Etter at `X` og `y` er definert deles så disse opp slik at deler av dataen blir avsatt til testdata for evaluering og validering, delkapittel 9.3.4. Teststørrelsen settes til 20%, og oppdelingen foregår likt for alle rutinene om ikke annet er presisert i senere beskrivelsen. Fordelingen av klassene mellom treningsdata og testdata blir kontrollert ved hjelp av figurer slik som figur 34 viser. Resultatet av oppdelingen er treningsdata definert som `X_train` med egenskaper og `y_train` med målverdier, og testdata

definert som **X_test** og **y_test**.

En *dictionary* brukes til å definere ønskede vektorer for klassene. Vektorer kan brukes til å håndtere ubalansert data og redusere støy. I denne treningsdataen er det store forskjeller mellom antall punkt som representerer de ulike klassene. Dette kan medføre at klassene som er overrepresentert dominerer beslutningene maskinlæringsmodellen foretar seg (Abhinav 2023). Det er derfor testet ulike tilnærminger for vektning av klassene.

RandomForestClassifier, se dokumentasjon scikit-learn developers (2023b), opprettes med vektene av klassene. Det gjøres et parametersøk og egenskaputvalg, om ikke annet er beskrevet for de ulike rutinene. *RandomizedSearchCV* funksjonen fra scikit-learn developers (2023j) *Python* biblioteket benyttes for å gjøre et søk etter egnede parametere. Funksjonen benytter treningsdata, **X_train** og **y_train**, og den definerte modellen for *RandomForestClassifier*. *RandomizedSearchCV* tester ut ulike kombinasjoner av parametrene fra tabell 4, og returnerer de parametrene som passer best for treningsdata og modell.

Table 4: Parametersøk

min_samples_leaf	[1, 2, 4, 5]
min_samples_split	[2, 5, 10]
max_depth	[None, 10, 20, 30, 35, 40]
n_estimators	tilfeldig verdi mellom 50 og 101

Slikt som beskrevet i delkapittel 9.3.3 er det flere ulike metoder for å velge ut egenskaper. Det er i dette delkapitlet definert et utvalg av egenskaper basert på en kombinasjon av ulike modeller, men for de ulike rutinene er det gjennomført flere ulike tilnærminger. Det er blant annet blitt brukt kun vektet *Feature Importance* fra *RandomForestClassifier*. Samt har egenbestemte utvalg av egenskaper blitt utforsket. Tilnærmingen til valg av egenskaper til klassifiseringsrutinene beskrives videre for hver av rutinene.

Etter at parametere for *RandomForestClassifier* modellen er bestemt defineres modellen på nytt med disse. Samtidig defineres **X_train** og **X_test** på nytt, slik at de kun inneholder egenskaper fra egenskaputvalget. Videre trenes modellen med ulike tilnærminger av treningsdata, parametere, vektorer og egenskapsutvalg som presenteres senere for hver enkel rutine.

Den trente modellen valideres med flere ulike funksjoner basert på den usette **X_test** med tilhørende *fasitdata* **y_test**. Valideringen gir verdier for *Precision*, *Recall*, *F1 Score*, *AUC-ROC Score*, *Log Loss* og en forvirringmatrise. Alle funksjonene er hentet fra *Sklearn.metrics*, og dokumentasjonen for de samtlige funksjonene er definert i scikit-learn developers (2023g,h,e,i,f,d). *Precision* gir en verdi på modellens evne til å ikke klassifisere et punkt som feil klasse, mens *Recall* gir en verdi for modellens evne til å predikere korrekt. *F1 score* er en beregning av middelerverdi for hver vektete klasse for de to første funksjonene. *AUC-ROC score* beregner en kurve som representerer ytelsen til modellen, og verdien den returnerer er arealet under kurven. Validering med *Log Loss score*, gir en presisjonsverdi for modellen basert på sannsynlighetene for predikasjon av riktig klasse. For den siste teknikken for validering brukes en forvirringmatrise. Denne viser direkte klassifiseringen til modellen av **X_test** og hvordan modellen har predikert i forhold til **y_test**. Den vertikale akse viser sann klasse, og den horisontale viser predikerte klasser, slik at diagonalen viser sanne predikasjoner for de ulike klassene.

Modellen brukes så til å predikere resten av punktskyen. CSV-filene for hver blokk, delt opp i forberedelsen av klassifiseringen, leses inn som *dataframe* en etter en. Først rekonstrueres *dataframen* med de samme kolonnenavnene som i *X_train* for å kunne brukes i modellen. Deretter ekstraheres rader med *NaN* verdier, disse skal ikke inngå i predikasjonene. For å ikke fjerne punktene helt defineres de som en ny klasse, klasse 8, som består av punkter der *CloudCompare* ikke har klart å beregne geometriske egenskaper for utvalget av egenskaper. Videre predikeres det klasseetiketter for *dataframen*, hvor de senere blir lagret til *laz*-format ved hjelp av *laspy*.

9.3.6. Ulike klassifiseringsrutiner

Det ble gjennomført 6 ulike klassifiseringer med bruk av *Random Forest* i denne oppgaven. Det vil under bli beskrevet hva som inngikk i de ulike klassifiseringsrutinene. Rutinene er uavhengig av rekkefølge.

Rutine 0 var det første av klassifiseringsrutinene. Dette var en relativ enkel rutine uten tilpasning av modellen, men med utvalg av felles egenskaper som tidligere beskrevet i kapittel 9.3.3. Det ble valgt å kjøre den første modellen uten tilpasning for å se hvordan modellens ytelse var på datasettet, slik at den kunne tilpasses bedre i senere rutiner. Det er også derfor den har blitt valgt å defineres som rutine null, da den kun ble brukt som et testforsøk.

I denne rutinen ble hele treningsdatasettet brukt, og det ble opprettet en vektning av de ulike klassene i modellen. I denne oppgaven var det mest fokus på de ulike bunnklassene, det var også disse som hadde minst andel punkter i treningsdatasettet. Land og vann har desidert flest definerte treningspunkter, men de er likevel ikke et hovedmål å skulle klare å klassifisere. Det ble derfor besluttet å vekte disse klassene ned i dette forsøket.

Rutine 0:

Egenskaper		Vekting av klasser		Parametere	
Egenskaper	Radius	Sand	1	random_state	42
Z		Spredt vegetasjon	1	class_weight	class_weight
Intensity		Tett vegetasjon	1	min_samples_leaf	
Verticality	1, 1.5, 2, 3	Vann	0.5	min_samples_split	
Number of neighbors	1.5, 2, 2.5	Stein	1	max_depth	
Surface density	2, 2.5, 3	Slam	1	n_estimators	100
Volume density	1, 1.5, 2, 2.5, 3	Land	0.5		

Fig. 35: Rutine 0

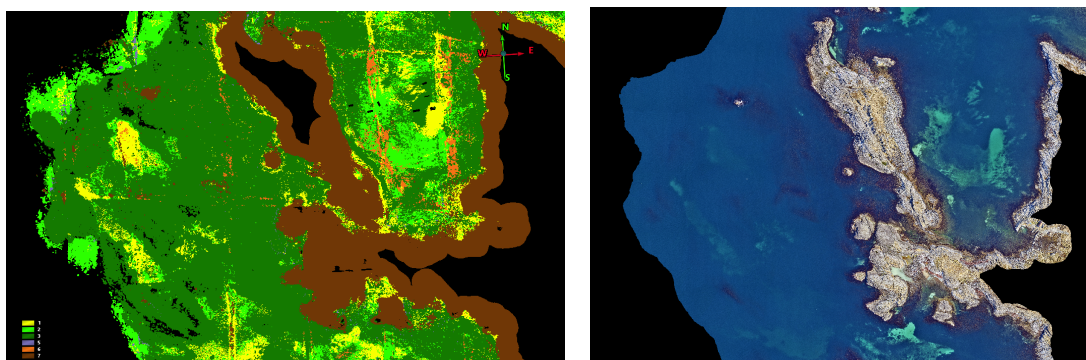


Fig. 36: Eksempel på slam som har blitt feilklassifisert i rutine 0

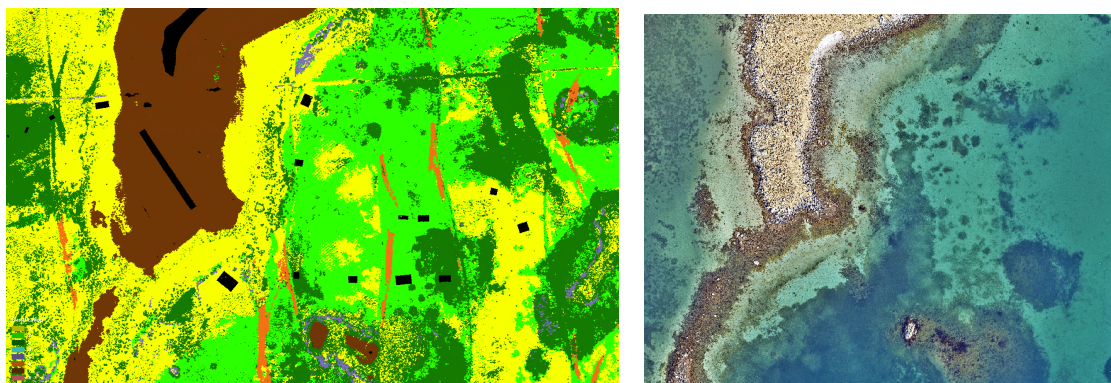
I rutine A ble treningsdatasettet delt opp i trening og test ved bruk av *StratifiedShuffleSplit* som forklart tidligere. Det ble satt av 20% som test sett til validering av modellen. Fra en visuell inspeksjon av **rutine 0** som illustrert i figur 36, var det tydelig at klassen slam hadde blitt klassifisert utenfor de respektive områdene. Det ble derfor implementert vektning av slam i rutine A. I denne rutinen ble det også foretatt en tilpasning av modellen med bruk av *RandomizedSearchCV*. Oversikt over de ulike parametrene kan ses i spesifikasjonene for rutine A nedenfor. Videre i rutinen ble det foretatt et utvalg av egenskaper med *Feature Importance* med den tilpassede modellen. Egenskapene som gikk inn i utvalget var alle egenskapene fra radius 1 til 3 meter.

Rutine A:

Egenskaper		Vekting av klasser		Parametere	
Egenskaper	Radius	Sand 2	1	random_state	42
Z		Spredt vegetasjon	1	class_weight	class_weight
Intensity		Tett vegetasjon	1	min_samples_leaf	1
Verticality	1, 1.5, 2, 2.5, 3	Vann	0.5	min_samples_split	5
Gaussian curvature	2.5, 3	Stein	1	max_depth	30
Sphericity	3	Slam	0.8	n_estimators	165
Anisotropy	2.5, 3	Land	0.5		
3rd eigenvalue	2.5, 3				
Surface variation	2.5, 3				
Normal change rate	3				
Omnivariance	3				
Number of neighbors	1.5, 2, 2.5, 3				
Surface density	1.5, 2, 2.5, 3				
Volume density	1.5, 2, 2.5, 3				

Validering 20% testdata	
Precision	0.998958
Recall	0.998961
F1 Score	0.998959
AUC-ROC Score	0.999992
Log Loss	0.015255

Forvirringsmatrise							
Confusion Matrix:							
[[10056	8	1	0	4	0	0]	
[7	6040	12	0	6	0	0]
[9	11	10938	0	2	0	0]
[0	0	0	31248	0	0	5]
[8	2	17	0	550	0	0]
[0	0	0	0	0	2953	0]
[0	0	0	1	0	0	27707]

Fig. 37: Rutine A**Fig. 38:** Eksempel på overlapp flystriper, samt blanding av spredt vegetasjon og sand i rutine A

Rutine **B** baseres på et tilfeldig utvalg av egenskaper. Den opprinnelige tanken var at det skulle undersøkes mønster i *CloudCompare*, som de ulike egenskapene ga, og ta et utvalg basert på om disse mønstrene var gjenkjennbare til enkelte bunntyper fra ortofoto. Dessverre ble det ikke tid til å gjennomføre dette da det var for tidskrevende å laste inn data. Det ble derfor forsøkt å kombinere enkelte egenskaper fra tidligere felles utvalg, og egenskaper som ikke var testet så mye. I denne rutinen ble det også forsøkt en annen tilnærming på vekting av klassene, slik at de ble mer stegvis representert. Tilpasning av modellen ble også utført med *RandomizedSearchCV*, og de ulike egenskapsutvalgene og parameterne er presentert nedenfor.

Rutine B:

Egenskaper		Vekting av klasser		Parametere	
Egenskaper	Radius	Sand	0.8	random_state	42
Z		Spredt vegetasjon	0.7	class_weight	class_weight
Intensity		Tett vegetasjon	0.75	min_samples_leaf	1
Verticality	1	Vann	0.5	min_samples_split	2
Anisotropy	2.5	Stein	1	max_depth	30
Mean Curvature	2	Slam	0.65	n_estimators	187
Normal change rate	2.5	Land	0.5		
Omnivariance	2.5				
PCA1	1				
Roughness	1.5				
1st order moment	1.5				
Volume density	2.5				
Linearity	1.5				
Planarity	1				
Eigenentropy	1.5				
PCA2	1				
2nd eigenvalues	1				

Validering 20% testdata	
Precision	0.994818
Recall	0.994831
F1 Score	0.994814
AUC-ROC Score	0.999903
Log Loss	0.031617

Forvirringsmatrise						
Confusion Matrix:						
[[9985	54	21	0	9	0	0]
[38	5941	73	0	12	1	0]
[46	69	10829	0	3	13	0]
[0	0	0	31248	0	0	5]
[22	7	39	0	509	0	0]
[1	0	32	0	0	2920	0]
[0	0	0	18	0	0	27690]

Fig. 39: Rutine B

I rutine C ble det forsøkt med alle egenskapene med radius 2 for å se hvordan kun et eksemplar av hver egenskap presterte sammen. Valget av radius var basert på at verdien var en midling av de tilgjengelige radiusene, og var hverken for lav eller for høy. Vektingen av spredt vegetasjon ble satt ned til 0.8, da det i figur 38, **rutine a** ble observert at spredt vegetasjon la seg i områder med ren sand. Oversikten nedenfor viser rutine C med alle de ulike egenskapene, samt parametere og valideringsresultater av brukt modell.

Rutine C:

Egenskaper		Vekting av klasser		Parametere	
Egenskaper	Radius	Sand	1	random_state	42
Z		Spredt vegetasjon	0.8	class_weight	class_weight
Intensity		Tett vegetasjon	1	min_samples_leaf	2
Verticality	2	Vann	0.5	min_samples_split	5
Gaussian curvature	2	Stein	1	max_depth	30
Sphericity	2	Slam	0.8	n_estimators	161
Anisotropy	2	Land	0.5		
Mean Curvature	2				
3rd eigenvalue	2				
Surface variation	2				
Normal change rate	2				
Omnivariance	2				
PCA1	2				
Roughness	2				
1st order moment	2				
Number of neighbors	2				
Surface density	2				
Volume density	2				
Eigenvalues sum	2				
Linearity	2				
Planarity	2				
Eigenentropy	2				
PCA2	2				
1st eigenvalue	2				
2nd eigenvalues	2				

Validering 20% testdata		Forvirringsmatrise	
Precision	0.993822	Confusion Matrix:	
Recall	0.993849	[[9985 54 21 0 9 0]	
F1 Score	0.993835	[38 5941 73 0 12 1]	
AUC-ROC Score	0.999875	[46 69 10829 0 3 13]	
Log Loss	0.041719	[0 0 0 31248 0 0 51]	
		[22 7 39 0 509 0 0]	
		[1 0 32 0 0 2920 0]	
		[0 0 0 18 0 0 27690]	

Fig. 40: Rutine C

For rutine **D** ble det besluttet å ekskludere egenskaper som var basert på punktetthet. Dette fordi at det i rutine **A** ble observert en effekt fra overlapp av flystripene, illustrert i figur 38. Spredt vegetasjon ble vektet på lik linje som land og vann for å se om det hadde en tydeligere effekt på skillet mellom klar sand og spredt vegetasjon. I oversikten nedenfor for modellen i rutine **D** er *Volume density*, *Surface density* og *Number of neighbors* ikke til stede, de er heller ikke inkludert i egenskaputvalget og parametersøket.

Rutine D:

Egenskaper		Vekting av klasser		Parametere	
Egenskaper	Radius	Sand	1	random_state	42
Z		Spredt vegetasjon	0.5	class_weight	class_weight
Intensity		Tett vegetasjon	1	min_samples_leaf	1
Verticality	1, 1.5, 2, 2.5, 3	Vann	0.5	min_samples_split	5
Gaussian curvature	2, 2.5, 3	Stein	1	max_depth	30
Sphericity	2.5, 3	Slam	0.8	n_estimators	176
Anisotropy	2.5, 3	Land	0.5		
Mean Curvature	3				
3rd eigenvalue	1, 1.5, 2.5, 3				
Surface variation	1, 1.5, 2.5, 3				
Normal change rate	1, 1.5, 2.5, 3				
Omnivariance	3				
PCA1	2.5, 3				

Validering 20% testdata	
Precision	0.997326
Recall	0.997332
F1 Score	0.997319
AUC-ROC Score	0.999967
Log Loss	0.030862

Forvirringsmatrise							
Confusion Matrix:							
[[10029	28	8	0	2	2	0]	
[39	5969	44	0	6	7	0]
[15	20	10922	0	2	1	0]
[0	0	0	31247	0	0	6]
[22	3	25	0	527	0	0]
[0	4	2	0	0	2947	0]
[0	0	0	3	0	0	27705]

Fig. 41: Rutine D

Spredt vegetasjon ble i rutine **E** fjernet for å se hvilken effekt dette hadde på hvordan modellen skilte mellom tett vegetasjon og sand. Egenskapene her ble som i rutine **D** valgt uten egenskaper for punktetthet.

Rutine E:

Egenskaper		Vekting av klasser		Parametere	
Egenskaper	Radius	Sand	1	random_state	42
Z		Spredt vegetasjon		class_weight	class_weight
Intensity		Tett vegetasjon	1	min_samples_leaf	1
Verticality	1, 1.5, 2, 2.5, 3	Vann	0.5	min_samples_split	2
Gaussian curvature	2, 2.5, 3	Stein	1	max_depth	30
Sphericity	1, 1.5, 3	Slam	0.8	n_estimators	155
Anisotropy	2.5, 3	Land	0.5		
Mean Curvature	3				
3rd eigenvalue	1, 1.5, 2.5, 3				
Surface variation	1, 1.5, 3				
Normal change rate	1, 1.5, 2, 2.5, 3				
Omnivariance	3				
PCA1	2.5				

Validering 20% testdata	
Precision	0.998677
Recall	0.998670
F1 Score	0.998650
AUC-ROC Score	0.999989
Log Loss	0.0153660

Forvirringsmatrise						
Confusion Matrix:						
[[10057	8	0	2	2	0]	
[18	10934	0	3	5	0]
[0	0	31251	0	0	2]
[34	29	0	514	0	0]
[4	1	0	0	2948	0]
[0	0	3	0	0	27705]

Fig. 42: Rutine E

9.3.7. Forsøk med Support Vector Machine

I tillegg til forsøkene med *Random Forest* ble det også forsøkt å bruke *Support Vector Machine (SVM)*. Dette er en mer kompleks modell enn *Random Forest*, og den brukte mye lengre tid på å utføre klassifiseringsrutinen med det samme datasettet. Med den datakapasiteten som var tilgjengelig i denne oppgaven brukte *SVM* rundt 3 timer på kun trening av modellen med et enkelt oppsett, og i løpet av en natt hadde den kun klart å predikere på 2 blokker av punkttskyen. På grunn av den lange tidsbruken, ble det derfor besluttet å ikke gå videre med modellen da utfordringene med bruk av feil punkttsky har tatt mye tid å rette opp i, kapittel 11 om utfordringer.

I forsøket med *SVM* ble det benyttet en mer standard klassifiseringsrutine med *train_test_split* funksjonen til *Scikit-learn*, og modellen ble kjørt med standard parametre inkludert *linear* som *kernel* parameter. Egenskapene som ble brukt var de felles utvalgte egenskapene. Det ble foretatt en skalering av trening og test data før trening av modellen, da *SVM* som modell krever dette, videre ble det ikke foretatt vektning av klasser.

Rutine for Support Vector Machine:

Egenskaper	
Egenskaper	Radius
Z	
Intensity	
Verticality	1, 1.5, 2, 2.5, 3
Number of neighbors	1.5, 2, 2.5, 3
Surface density	1.5, 2, 2.5, 3
Volume density	1, 1.5, 2, 2.5, 3
Definerte parametere	
random_state	1
kernel	linear

Forvirringsmatrise

	0	1	2	3	4	5	6
0	7817	1520	642	0	30	60	0
1	1577	2740	1479	0	5	264	0
2	280	435	9915	0	4	326	0
3	0	0	1	31208	0	0	44
4	280	60	221	0	16	0	0
5	167	62	126	0	0	2598	0
6	0	0	0	122	0	0	27586

Validering 20% testdata

average score	0.8929
training score	0.9124
accuracy score	0.9139

Klassifiseringsrapport

	precision	recall	f1-score	support
1.0	0.77	0.78	0.77	10069
2.0	0.57	0.45	0.50	6065
3.0	0.80	0.90	0.85	10960
4.0	1.00	1.00	1.00	31253
5.0	0.29	0.03	0.05	577
6.0	0.80	0.88	0.84	2953
7.0	1.00	1.00	1.00	27708
accuracy			0.91	89585
macro avg	0.75	0.72	0.72	89585
weighted avg	0.91	0.91	0.91	89585

Fig. 43: Rutine for Support Vector Machine

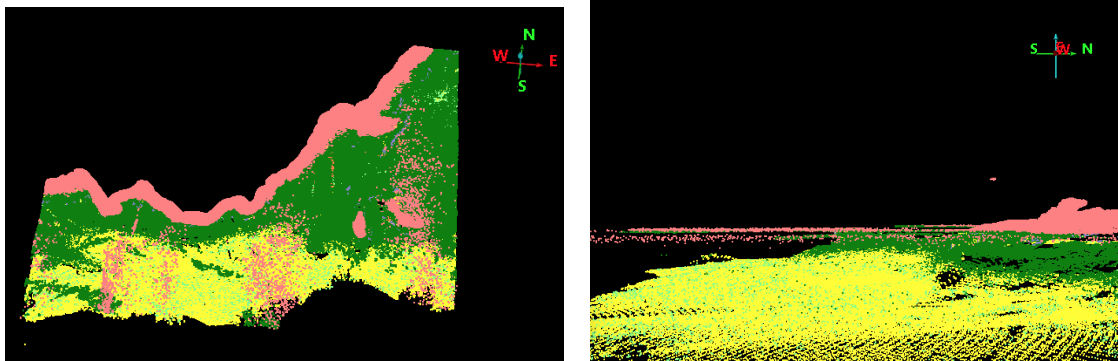


Fig. 44: Resultat av SVM på den ene predikerte punktskyblokken

Figur 44 viser det predikerte resultatet, på den ene predikerte blokken, fra forsøket med *SVM*. På bildene i figuren er vannklassen skrudd av for å kunne tyde hva som har blitt klassifisert som bunntyper. Det er enkelte punkter i overgangen av flystripene som har blitt klassifisert som land istedet for vannoverflate. Det er også blitt klassifisert en del sand i området. Undervannsbildene tyder derimot på at deler av dette området er spredt vegetasjon.

10. RESULTATER

Dette kapittelet tar for seg resultatene fra klassifiseringsrutine **A**, **B**, **C**, **D** og **E** som alle baserer seg på *Random Forest* modeller. Det vil bli gjennomgått en evaluering av ytelsene til de ulike rutinene, samt en visuell inspeksjon og sammenlikning med ortofoto.

Rutine A

Validering 20% testdata	
Precision	0.998958
Recall	0.998961
F1 Score	0.998959
AUC-ROC Score	0.999992
Log Loss	0.015255

Forvirringsmatrise						
Confusion Matrix:						
[[10056	8	1	0	4	0	0]
[7	6040	12	0	6	0	0]
[9	11	10938	0	2	0	0]
[0	0	0	31248	0	0	5]
[8	2	17	0	550	0	0]
[0	0	0	0	0	2953	0]
[0	0	0	1	0	0	27707]

Forvirringsmatrisen til modellen i rutine **A** illustrerer at det er et fåtall av observasjoner som har blitt feilklassifisert i dette forsøket. Hver rad i forvirringsmatrisen viser antall elementer som er klassifisert i den spesifikke klassen. Rad 1 tilhører klasse 1, *sand*, rad 2 tilhører klasse 2, *spredd vegetasjon* osv. Den horisontale akse viser predikasjonene av testdata, altså klassen modellen predikerer et punkt til å være. Det vil si at diagonalen, hvor predikasjon stemmer overens med definert måleverdi, er sanne predikasjoner som modellen har gjort på testsettet.

Fra forvirringsmatrisen virker det som at modellen har litt vanskeligheter med å skille sand og spredd vegetasjon, spredd vegetasjon blir i tillegg forvekslet med tett vegetasjon. Klassen hvor flest observasjoner har blitt feilklassifisert er klasse 5, *stein, berg*, som i flere tilfeller har falske positive observasjoner, blant annet tett vegetasjon og sand.

Valideringen med testdata viser også at modellen yter bra. *Precision*, *Recall*, *F1 score* og *AUC-ROC Score* ligger alle opp mot 1 som er et maksimalt mål og viser til perfekt resultat. *Log Loss* har en omvendt skala, hvor tall nærmere null viser et bra resultat.

Rutine B

Validering 20% testdata	
Precision	0.994818
Recall	0.994831
F1 Score	0.994814
AUC-ROC Score	0.999903
Log Loss	0.031617

Forvirringsmatrise						
Confusion Matrix:						
[[9985	54	21	0	9	0	0]
[38	5941	73	0	12	1	0]
[46	69	10829	0	3	13	0]
[0	0	0	31248	0	0	5]
[22	7	39	0	509	0	0]
[1	0	32	0	0	2920	0]
[0	0	0	18	0	0	27690]

Rutine **B** viser også lovende verdier for validering av modellen. Likevel presterer tidligere rutine **A** noe bedre på valideringresultatet.

Denne rutinen som bygger på selvvalgte tilfeldige egenskaper har flere ikke sanne predikasjoner enn det i rutine **A** for testdatasettet. Ut i fra forvirringsmatrisen er det flest feilklassifiseringer for klassene sand, spredd vegetasjon og tett vegetasjon. Sand blir forvekslet til å være spredd og tett vegetasjon, men den forveksles også med stein. Dette går også igjen for spredd vegetasjon, tett vegetasjon og stein. Klassen som blir mest feilklassifisert for rutine **B** ser ut til å være tett vegetasjon. Her er det flest punkter som er feilklassifisert totalt sett, hvor det er predikert til å være sand, spredd vegetasjon, stein og slam istedenfor den sanne verdien tett vegetasjon.

Forvirringmatrisen viser også at land og vann er knyttet til hverandre. De eneste feil predikasjonene av vann har sann klasse som land, og feil predikasjoner av land har kun sanne klasse som vann. De har i denne rutinen ingen feilklassifiseringer mot noen av de andre klassene for predikasjoner på testdata.

Rutine C

Validering 20% testdata	
Precision	0.993822
Recall	0.993849
F1 Score	0.993835
AUC-ROC Score	0.999875
Log Loss	0.041719

Forvirringsmatrise	
Confusion Matrix:	
[[9956 48 16 0 31 18 0]	
[[44 5914 77 0 26 4 0]	
[[34 87 10832 0 5 2 0]	
[[0 0 0 31247 0 0 6]	
[[43 7 39 0 488 0 0]	
[[43 2 16 0 0 2892 0]	
[[0 0 0 3 0 0 27705]	

Som en oppfriskning, bygger rutine **C** på alle egenskaper beregnet med radius 2. For validering av rutinen ser det ut til at den gjør det dårligere enn både rutine **A** og **B**, likevel er det gode validerings verdier.

Forvirringsmatrisen viser at vann og land har færrest feilklassifiseringer i rutine **C** på testdata. Det er tydelig at det lave antallet av feilklassifiseringer er mellom begge disse klassene, land og vann. Sand utgjør klassen som modellen har predikert mest falske verdier for, mens spredt vegetasjon har høyest antall av punkter som har sann verdi som spredt vegetasjon men som er predikert til andre klasser.

Det er også en økning i feil predikasjoner for både stein og slam i forhold til de forrige rutinene. Likevel er det få verdier sammenliknet med verdiene langs diagonalen for antall sanne predikasjoner.

Rutine D

Validering 20% testdata	
Precision	0.997326
Recall	0.997332
F1 Score	0.997319
AUC-ROC Score	0.999967
Log Loss	0.030862

Forvirringsmatrise	
Confusion Matrix:	
[[10029 28 8 0 2 2 0]	
[[39 5969 44 0 6 7 0]	
[[15 20 10922 0 2 1 0]	
[[0 0 0 31247 0 0 6]	
[[22 3 25 0 527 0 0]	
[[0 4 2 0 0 2947 0]	
[[0 0 0 3 0 0 27705]	

Rutine **D** baserte seg på å undersøke en rutine som ikke baserte seg på noen av egenskapene for punktetthet. Dette var for å se om disse egenskapene påvirket flystripene som gjenspeilte seg i de tidligere rutinene.

Valideringen av rutine **D** viser en sterkere evne til å klassifisere riktig og ikke klassifisere feil, generelt en sterkere predikasjonsnevne enn for rutinene **C** og **B**. Valideringen gir også en bedre verdi for presisjon enn for de tidligere modellene, med unntak av **A**.

Forvirringsmatrisen til rutine **D** viser færre feil for predikasjoner enn de tidligere rutinene, med unntak igjen av **A**. Relasjonen mellom land og vann går igjen for denne rutinen. Ingen av klassene forveksles av andre klasser, men de forveksles med hverandre. Modellens evne til å skille mellom sand, spredt vegetasjon og tett vegetasjon er også noe som tydelig går igjen for denne modellen, men med lavere antall predikasjonsfeil.

Rutine E

Validering 20% testdata	
Precision	0.998677
Recall	0.998670
F1 Score	0.998650
AUC-ROC Score	0.999989
Log Loss	0.0153660

Forvirringsmatrise	
Confusion Matrix:	
[[10057 8 0 2 2 0]	
[[18 10934 0 3 5 0]	
[[0 0 31251 0 0 2]	
[[34 29 0 514 0 0]	
[[4 1 0 0 2948 0]	
[[0 0 3 0 0 27705]	

Den siste rutinen for klassifisering, rutine **E**, ble definert uten egenskaper for punktetthet men også uten klassen for spredt vegetasjon. Valideringsresultatet for rutinen er svært lik som den for rutine **A**. Presisjonen og predikasjonsnevnen til modellen i **A** er likevel bedre med små marginer. Slik som alle de andre rutinene er valideringstestene gode.

Forvirringmatrisen representerer her ikke spredt vegetasjon, da den ikke er inkludert i modellen. I forhold til tidligere forklaringer har alle klasser etter rad to og kolonne to nå flyttet seg en plass til venstre. I denne modellen er det stein som blir klassifisert feil flest ganger i predikasjoner av testdata. Dette er en ny observasjon i forhold til de andre rutinene. Det er flest predikasjoner av sand som ikke er sanne, noe som har gått igjen. Vann og land er fortsatt de to klassene som har minst predikasjonsfeil og som isåfall predikeres feil seg imellom.

Valideringene er svært gode for alle rutinene og det er lite som skiller dem. For punktskyen og området som denne masteroppgaven tar for seg er det ikke mulig å evaluere modellenes ytelse kun på validering. Alle modellene bærer preg av overtilpasning, som har sin opprinnelse i at testdata ikke er uavhengig av treningsdata. I denne oppgaven er ikke dette mulig å unngå, da punktskyen er fra ett og samme område. Generelt sett kan det også være overtilpasning av modellene, men dette er ikke mulig å si noe om på grunn av overtilpasning mellom testdata og treningsdata.

Siden valideringstestene av de ulike modellene for de forskjellige rutinene ikke kan gi en robust beskrivelse av modellenes prestasjon er en videre tilnærming visuell inspeksjon. Utgangspunktet for inspeksjonen er ortofoto fra kapittel 8.1. Grunne områder, hvor deler av bunn er synlig i ortofoto, er utgangspunktet for visualiseringene. Det eksisterer samtidig ingen fasitdata for bunntyper i området, ikke som inkluderer vegetasjon. De tidligere treningsområdene som ble plukket ut fra undervannsbilder, vil derfor også bli brukt som en referanse til sann bunntype. Figur 45 viser de ulike fargekodene som er brukt på de ulike klassene i resultatene på punktskyen som blir videre presentert under.



Fig. 45: Oversikt over de ulike fargekode på de ulike klassene i den visuelle inspeksjonen av klassifiserings resultatene

Treningspunkter

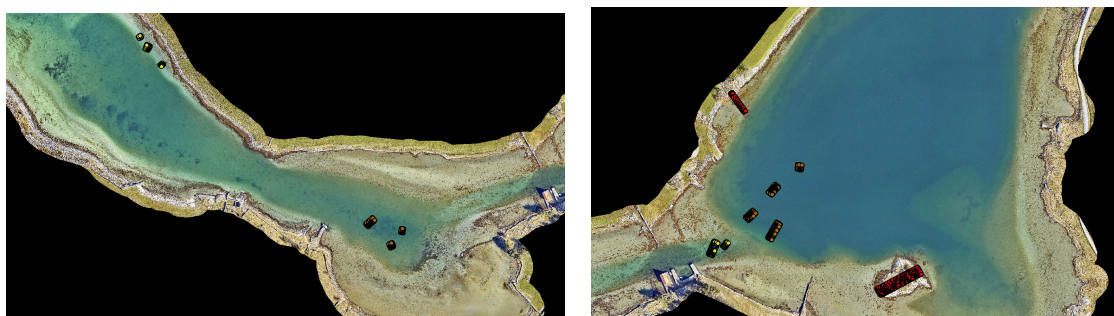
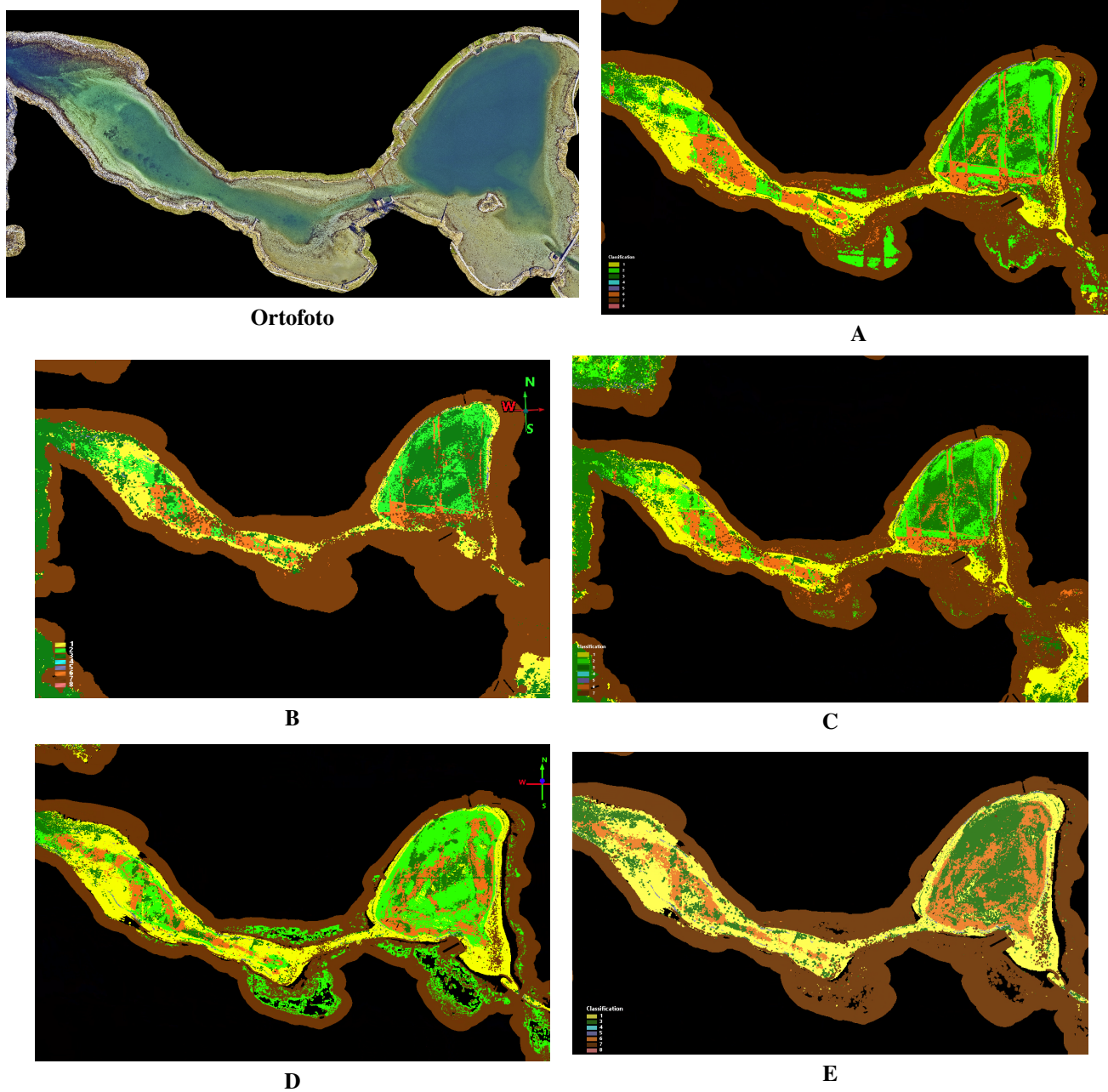


Fig. 46: Treningspunkter i område 1, 2 og 3 til bruk som referanse. Gul illustrerer sand, brun viser slam og rød viser treningspunkter for land.

Fig. 47: Område: 1, 2 og 3



Området 1, 2 og 3 er et område som i hovedsak består av slam og sand. Området 3 er det området som inneholder mest slam, som også inngår i bunnsedimentkartet i figur 6. Videre så har det også blitt definert treningsområder for slam i område 2 som vist i figur 46, hvor kantene av den smale korridoren har fått treningspunkter som sand.

De aller fleste modellene har klart å få til et omriss av sand med noe variasjoner. Modell i rutine **B** og modell fra **C** er de rutinene som har dette svake beltet av sand. Videre har de aller fleste modellene slitt litt med å skille på slam og begge vegetasjonsklassene. Modell **E** er den modellen som har klassifisert mest slam i område 3, mens modell **A** har klassifisert mest slam i område 1 og 2 i den smale korridoren. Det er også en tydelig effekt fra overlapp av flystriper som påvirker klassifiseringene i modellene **A**, **B** og **C**. I det smale området blir slam og spredt vegetasjon klassifisert litt annen hver gang i disse stripene.

For land har modellene litt ulik tolkning av det som fremgår i ortofoto som sandstrand. Modell **B**, **C** og **E** har definert dette som land, mens **A**, **D** og deler i **C** har fått land med variasjoner av spredt vegetasjon i disse områdene. Modell **D** og **E** har en glippe mellom land og bunnklassene. I disse illustrasjonene har vannklassen blitt slått av for å kunne se resultatet av de ulike bunnklassene. Disse glippene har blitt definert som vann og er derfor ikke med i illustrasjonen.

Treningspunkter

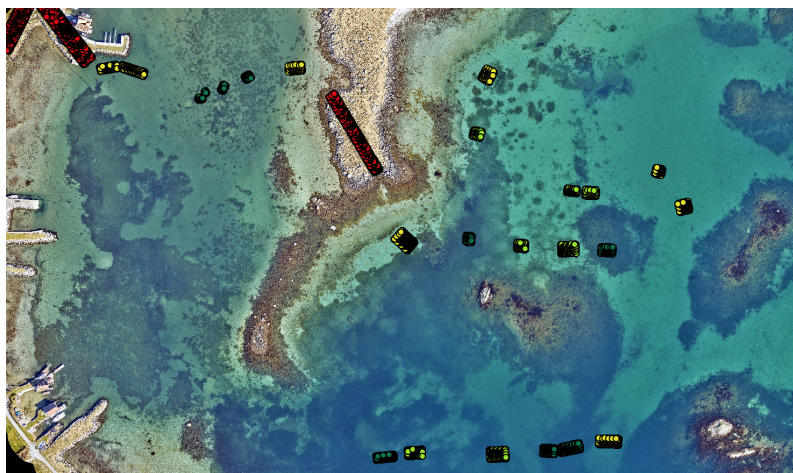
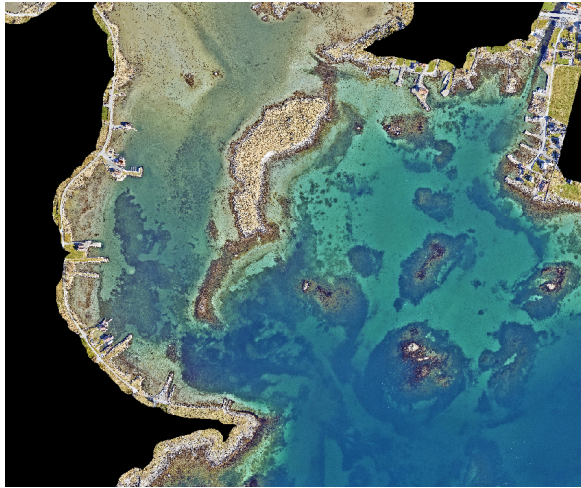
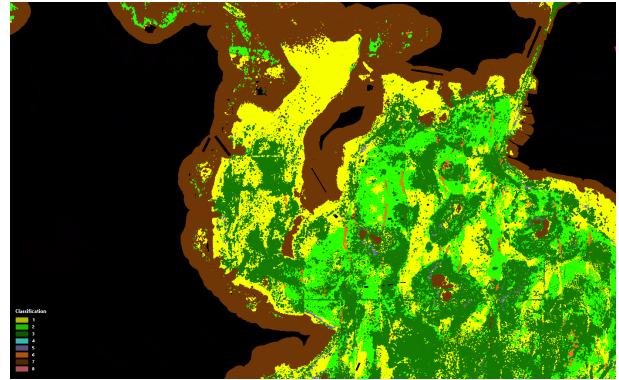


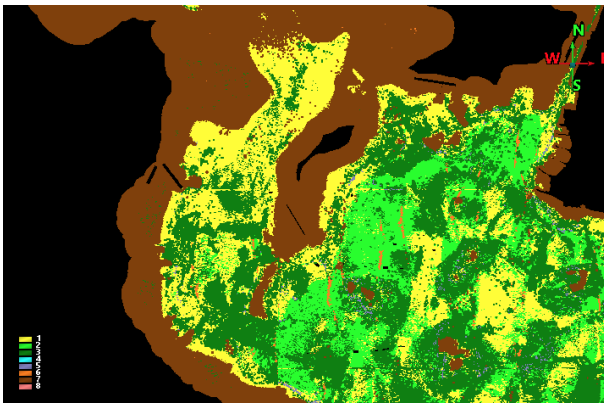
Fig. 48: Treningspunkter i område 4, 5 og 6 til bruk som referanse. Gul illustrerer sand, lys grønn viser *spredt vegetasjon*, mørk grønn tilhører *tett vegetasjon* og rødt viser treningspunkter for *land*.



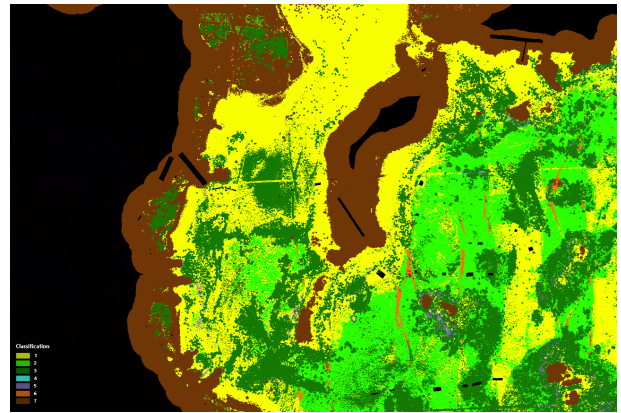
Ortofoto



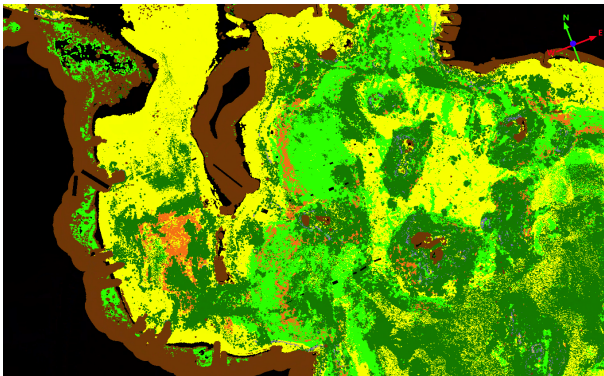
A



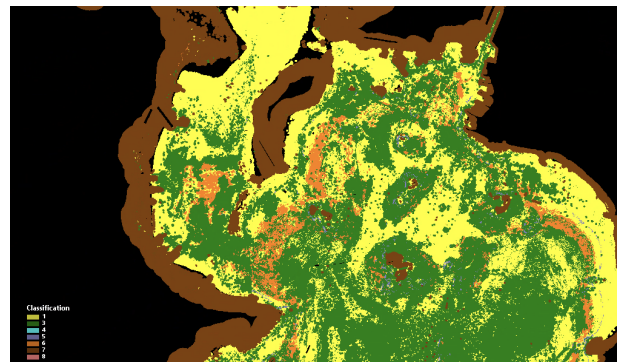
B



C



D



E

Fig. 49: Område: 4, 5 og 6

Område 4, 5 og 6 har i treningsdatasettet, som vist i figur 48, en variasjon av klassene *sand*, *spredd vegetasjon* og *tett vegetasjon*.

Mønstrene for tett vegetasjon på østsiden av halvøya ser ut til å være gjentakende i alle modellene, det er noe variasjon i bukta helt til venstre. Modell **D** og **E** er de som inneholder mest slam. **A**, **B** og **C** har fått slam definert på samme sted. Det har ikke blitt observert dette i områdene med dekning fra undervannsbildene. Videre så har modell **A**, **C** og **D** i dette området også fått vegetasjonsklasser på land.

Treningspunkter

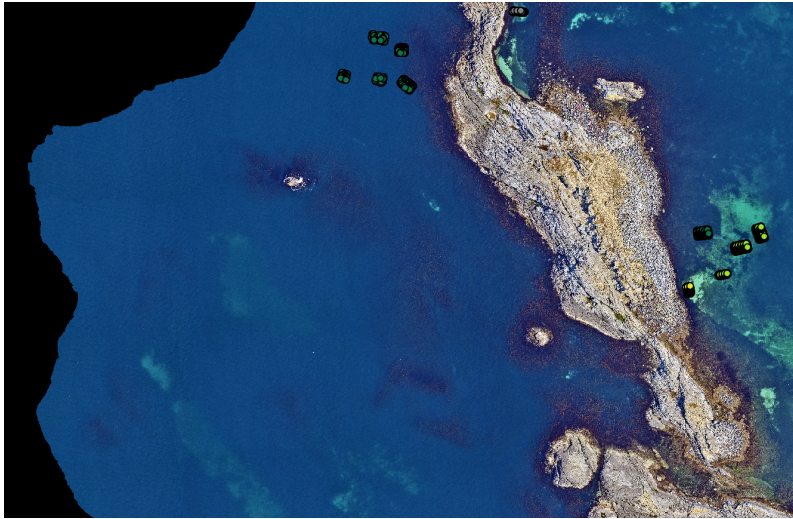
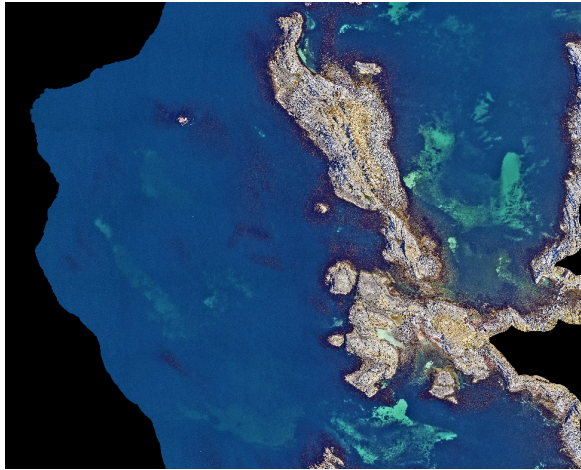
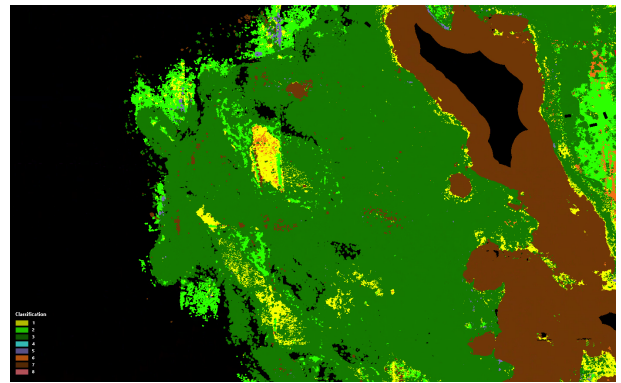


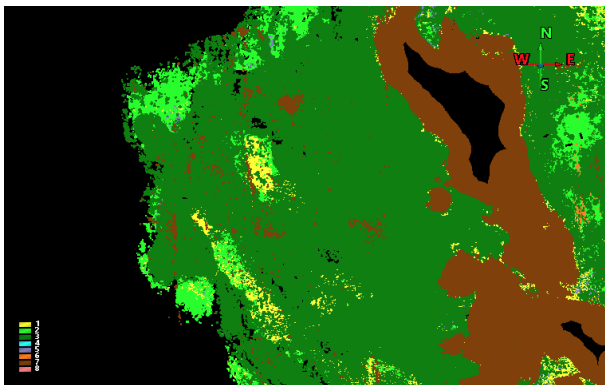
Fig. 50: Treningspunkter i område 17 til bruk som referanse. Gul illustrerer *sand*, lys grønn viser *spredd vegetasjon*, mørk grønn tilhører *tett vegetasjon* og grått viser treningspunkter for *stein, berg*.



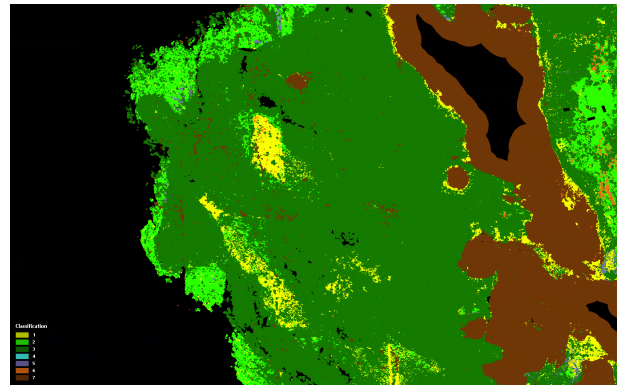
Ortofoto



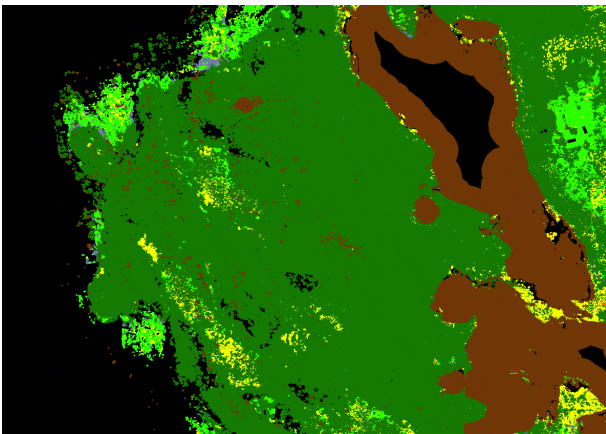
A



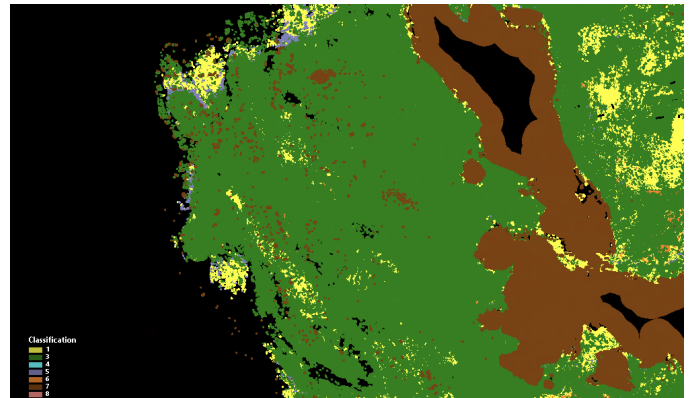
B



C



D



E

Fig. 51: Område: 17

I område 17 er det på vestsiden av øya, vist i 50, definert tett vegetasjon. Det ser også ut som i ortofoto at området på utsiden hovedsak består av tett vegetasjon med enkelte områder med sand. Modell C ser ut, basert på ortofoto å ha definert disse sandområdene klart av modellene.

Videre ser det ut til at modellene som inkluderer spredt vegetasjon har klassifisert dette på samme sted. Modell E, som ikke inneholder denne klassen, har definert samme områder som sand.

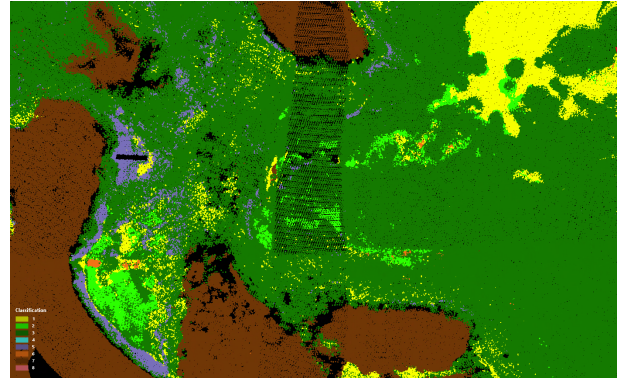
Treningspunkter



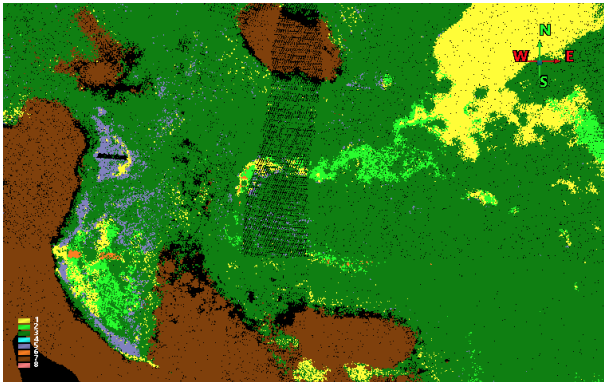
Fig. 52: Treningspunkter i område mellom 17 og 18 til bruk som referanse. Her illustrerer grått treningspunkter for *stein, berg*.



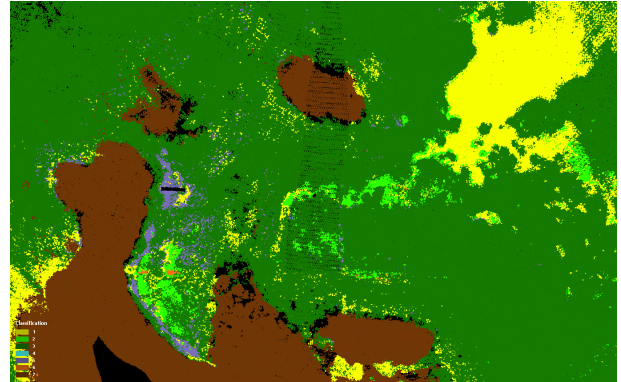
Ortofoto



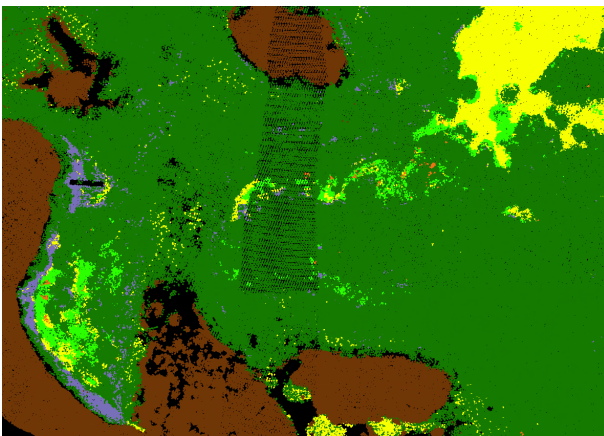
A



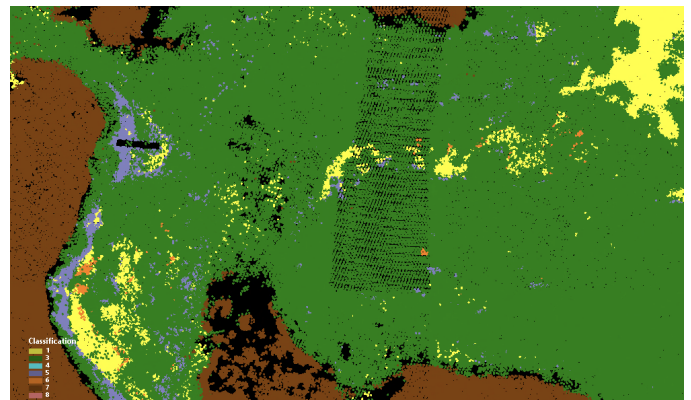
B



C



D



E

Fig. 53: Område: mellom 17 og 18

I området mellom 17 og 18 fra treningsområdene har det blitt plukket ut et treningsområde med stein, som vist ute på enden av landområdet i figur 52. I ortofoto ser det ut som det er et steinbelte som følger den lille viken inn, mens bunnen litt lenger fra land består av sand. Alle modellene har klart å fange deler av dette belte, med noe variasjoner mellom modellene. Det som på ortofoto syntes å være klar sand helt innerst i vika, har i modellene som inneholder spredt vegetasjons klassen blitt definert som spredt vegetasjon.

I modellene A, B og C ser det ut til å ha dukket opp en stripe med slam inne i denne vika. I modell D og E dukker det opp slam i samme område, men den ser ikke ut til å ha samme stripe effekt.

Videre har alle modellene klart å fange det litt større sandområdet som ligger litt lenger ut fra vika, øverst til høyre i bildene. Det er noe variasjoner i mønsteret mellom modellene og skillet mellom spredt vegetasjon og sand.

Treningspunkter

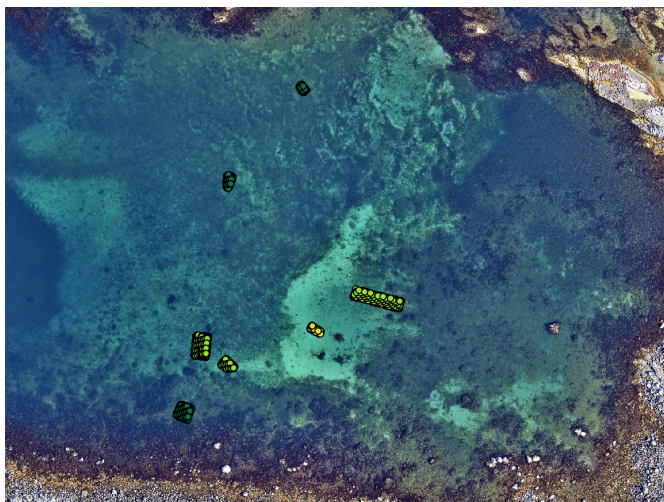


Fig. 54: Treningspunkter i område 18 til bruk som referanse. Gul illustrerer *sand*, lys grønn viser *spredt vegetasjon* og mørk grønn viser treningspunkter for *tett vegetasjon*.

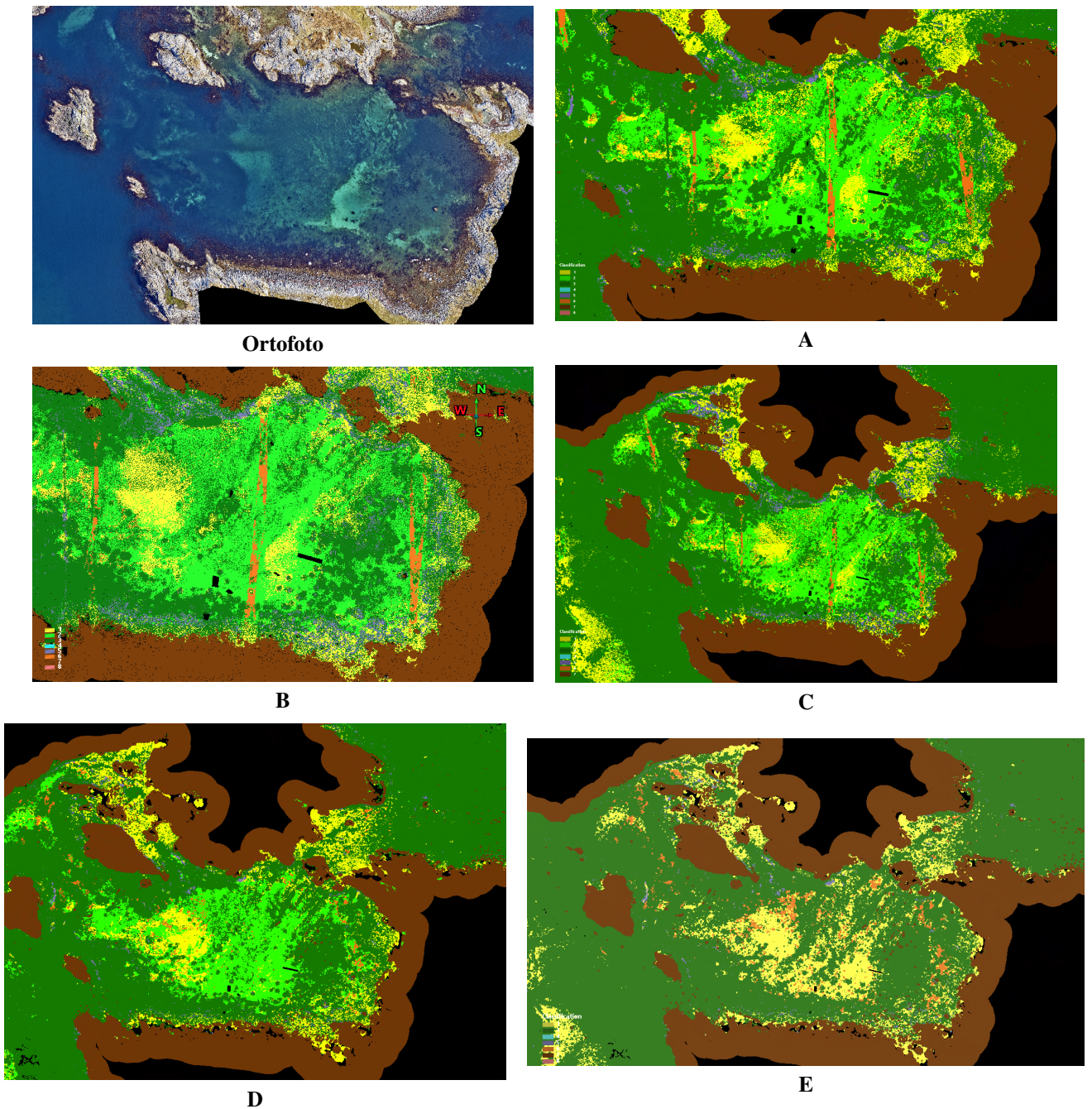


Fig. 55: Område: 18

Område 18 består av sand, spredt vegetasjon og tett vegetasjon som vist i figur 52. Alle modellene ser ut til å ha fått med ganske detaljerte små flekker eller samlinger av tett vegetasjon inne i bukta, hvor modell i **E** ikke har like stort skille da den ikke inneholder spredt vegetasjon. Modellene for **A**, **B** og **C** har i dette området fått klassifisert slam i overlapp av flystripene. Videre har sand blitt klassifisert forskjellig mellom modellene, hvor modellen i **D**, ikke har like mye sand inne i bukta som de andre.

I tillegg til visuelle klassifiseringsresultater for de ulike rutinene har fordelingen av predikasjonene blitt estimert. For hver lagrede *laz* fil har de ulike klassene blitt telt opp og plottet mot totalt antall punkt i punktsky. Resultatet av dette vises i figur 56.

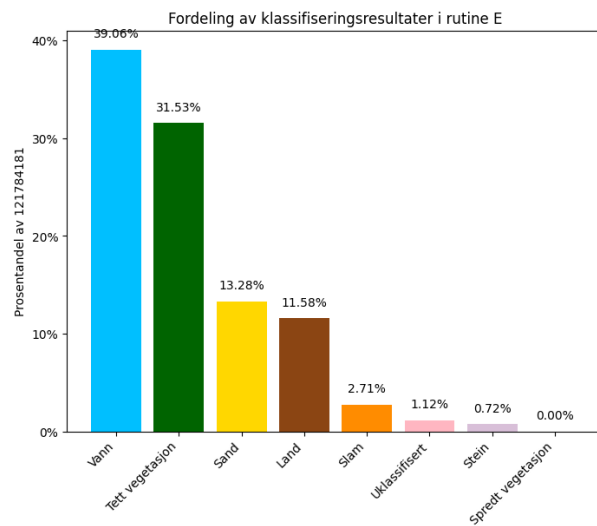
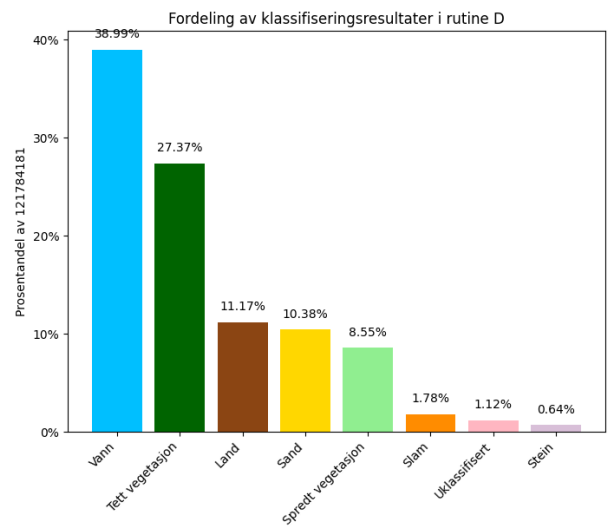
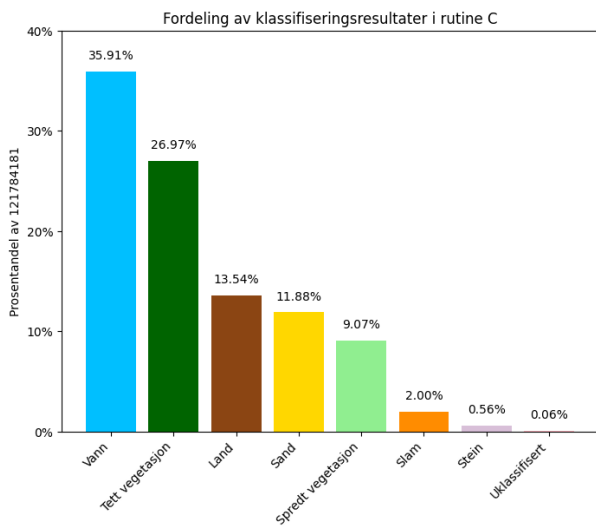
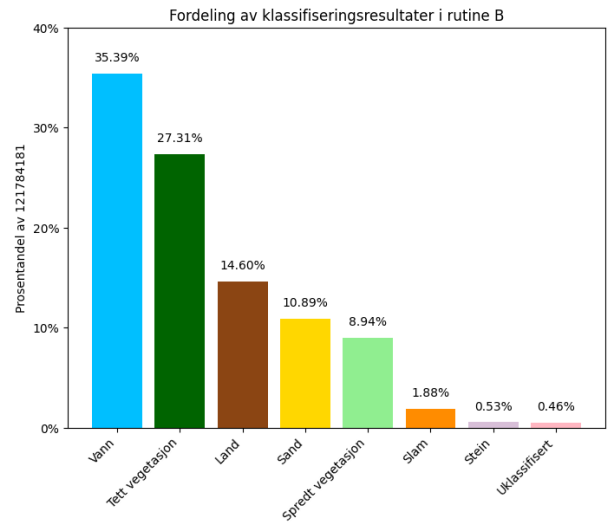
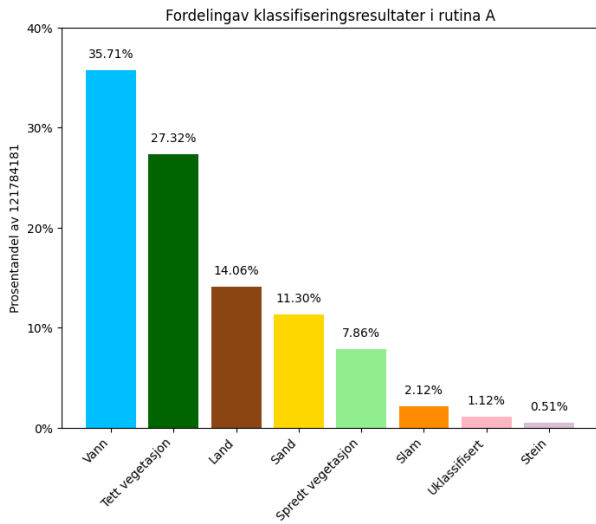


Fig. 56: Prosentfordeling av predikasjoner

Fordelingen av klasser i resultatene av de ulike rutinene er relativt like ved første øyekast. Likevel utgjør små forskjeller i prosentendring store forskjeller for antall punkt. For eksempel er det en prosentendring for tett vegetasjon mellom rutine **A** og **B** på 0.01%, men som utgjør en forskjell på over 16 000 punkter. Totalt sett for punktskyen utgjør ikke dette veldig mye, men det kan utgjøre større visuelle forskjeller. Fordelingen viser kun antall predikasjoner av punkter, men forteller ingenting om hvor disse punktene befinner seg. Tolkningen av resultatet bygger derfor på andelen av predikerte klasser for hver rutine.

Den uklassifiserte klassen som kun er avhengig av egenskaputvalget, utgjør svært lite for alle modellene. Rutine **C** sitt egenskaputvalg er den modellen som gir færrest uklassifiserte punkter. **A**, **D** og **E** sine egenskaputvalg har flest uklassifiserte punkter og er representert i like stor andel. Vann klassen har omtrent lik fordeling for **A**, **B** og **C**, men øker ganske mye for de to andre rutinene. For land er det en motsatt effekt hvor den er nokså lik for de tre første rutinene, men synker en del i de to siste.

Klassen for tett vegetasjon varierer en god del mellom de ulike rutinene. Det er en liten prosentforskjell mellom **A** og **B** som nevnt over, men de andre rutinene har større forskjeller som kan utgjøre store mengder punkt. For rutine **E** er økningen avhengig av at klassen for spredt vegetasjon er ekskludert fra rutinen. Av alle rutinene er det modellen i **C** som predikerer færrest punkter som tett vegetasjon.

Klassen for sand varierer også for alle de ulike modellene. Det er flest predikasjoner av sand i rutine **E** sammenliknet med de andre. Likevel er den som sagt avhengig av at spredt vegetasjon er ekskludert. For de andre rutinene er det ingen kritisk store endringer, men forskjellene kan utgjøre store visuelle forskjeller. Det samme tilfellet gjelder for predikasjoner av spredt vegetasjon for de fire første rutinene. Rutine **C** har predikert mest spredt vegetasjon.

Predikasjonene for Slam og Stein utgjør den minste andelen av punktskyen for alle modellene, figur 56. Likevel er det ikke forventet store områder bestående av disse bunntypene, og det er også disse to klassene som har minst treningsdata. Fra fordelingsplottene er det en del variasjon mellom modellene. Selv om det ikke er kritiske forskjeller er klassene så lite representert at det visuelt sett kan bli svært tydelig i flere områder. Mellom rutine **D** og **E** er det nesten 1% endring i fordeling av klassen for slam. Siden utgangspunktet er en liten del av punktskyen kan en slik økning vises godt visuelt. Det samme gjelder for klassen stein, hvor det er predikert mer stein i **D** og **E**, enn i de andre rutinene.

11. UTFORDRINGER

Før oppgaven videre tar for seg diskusjon og konklusjon av metoden vil utfordringer belyses. Det har blitt henvist til dette kapitlet tidligere i metoden for å kunne begrunne ulike valg og løsninger. For å bedre forstå disse valgene og løsningene må utfordringer belyses, men disse utfordringene vil også bli diskutert videre. Dette kapitlet tar for seg større utfordringer ved gjennomføring av denne masteroppgaven.

Ved arbeidet ved masteroppgaven har det blitt arbeidet med en punktsky i god tro om at denne var fullstendig. Punktskyen dekte hele det definerte området som er presentert flere ganger i oppgaven. Denne punktskyen bestod av nærmere 1 million punkter fordelt jevnt i hele området. Uten erfaring med batymetrisk data fra tidligere ble det ingen reaksjoner på dette. Punktskyen viste både bunnpunkter og punkter på vannoverflaten.

Dette har medført flere ulike beslutninger basert på feil grunnlagsdata. Bunntypene er definert på bakgrunn av referansedata fra havforskningsinstituttet, men også på bakgrunn av punktskyen. I tillegg til å se hva en kan definere ut i fra undervannsbilder, har det blitt vurdert hvilke detaljer en faktisk kan hente fra punktsky. I dette tilfellet var det svært lite. Punktene var svært spredt og punktskyen bestod av flere gap og hull. I figur 57 viser bildet til venstre treningsdata hentet ut fra den første antatt korrekte punktskyen og bildet til høyre viser samme utvalgsområde for treningsdata med den fullstendige punktskyen. Bestemmelsen av bunntyper tar utgangspunkt fra den antatt korrekte punktskyen.

Treningspunkter før og etter endring av punktsky

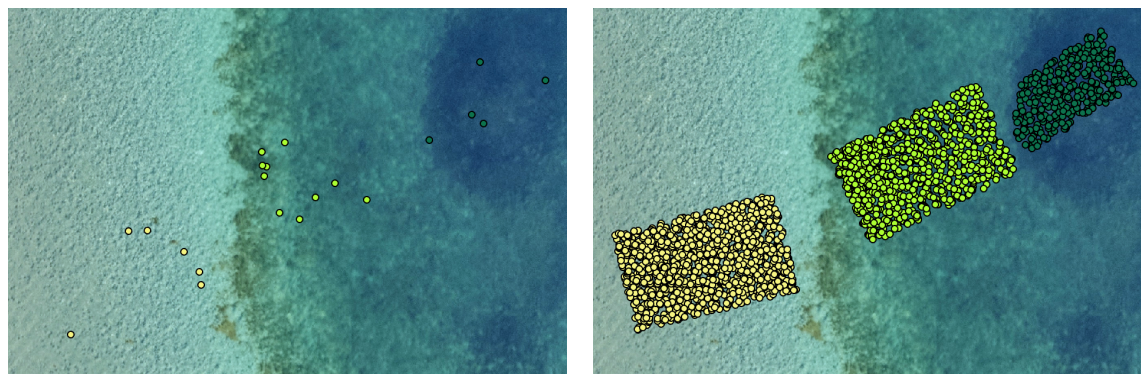


Fig. 57: Eksempel på flere punkter i treningsdata etter endring av punktsky

Med den antatt korrekte punktskyen var det også svært utfordrende å hente ut treningsdata. Det var svært få punkter i tverrsnittene, og det var flere områder det ikke var mulig å registrere noe som helst. I tillegg til dette ble det oppdaget at tverrsnittet i *Terrascan* ikke holder seg mellom de definerte start- og sluttpunktene som tegnes. Vinduet som viser tverrsnittet viste seg å vise litt tilfeldig forlenging av tverrsnittet. Dette ble oppdaget ved kontroll av treningsdata etter at registrering var fullført. Det var flere punkter langt forbi tverrsnitt koordinatene som hadde blitt registrert. Arbeidet i *Terrascan* var en tidkrevende jobb, og det var også en tidkrevende jobb å rette opp i disse punktene som var definert feil.

Oppdagelsen av at antatt fullstendig punktsky ikke var fullstendig var sent ut i masteroppgaven, det var allerede kjørt testrunder for maskinlæring. Forskjellen fra den fullstendige utgjorde over 120 millioner punkter. Dette introduserte et kristisk problem, om daværende grunnlagsdata var godt nok til å i det hele tatt kunne si noe om resultatene. Det ble rask besluttet at den antatt fullstendige punktskyen ikke kunne benyttes videre, da hele masteroppgaven ville bygd på et svakt utgangspunkt.

Det ble videre besluttet å implementere den fullstendige punktskyen. Grunnet godt arbeid og god dokumentasjon av treningsområdene, ble treningsdata plukket ut på nytt basert på dokumentet **Innsamling av treningsdata i Terrascan**, utdrag i vedlegg 14. Geometriske egenskaper ble generert på nytt, som viste seg naturlig nok å være en mer tidkrevende prosess. Datagrunnlaget for masteroppgaven baserer seg på den fullstendige punktskyen, men beslutninger og valg tatt før maskinlæringsprosessen bærer preg av den tidligere punktskyen. Det var ikke tilstrekkelig nok med tid til å kunne endre på alt av

beslutninger og tilnærminger.

Masteroppgaven bærer dessverre preg av denne utfordringen. Det er prøvd å belyse det så godt det lar seg gjøre. Det er viktig å få frem begrensninger som har oppstått i metode, men uten at det skal ta for mye plass i forhold til hva metoden bygger på. Datagrunnlaget bygger på en fullstendig punktsky, men det kan likevel bli misforståelser bak teorier og tilnærminger i metode basert på datagrunnlag som er forskjellig.

12. DRØFTING

I dette kapitlet vil det utforskes og drøftes mer rundt metoden og resultatene for å skape en helhetlig forståelse av arbeidet i masteroppgaven. Det vil først bli fokusert på en kritisk refleksjon over tilnærminger i metoden. Deretter vil de oppnådde resultatene drøftes og tolkes. Det vil forsøkes å avdekke styrker, svakheter og begrensninger i både tilnærminger og resultater.

Det vil forsøkes å gi et grunnlag for arbeidets bidrag til å undersøke hvordan geometriske egenskaper kan påvirke klassifisering ved bruk av maskinlæring.

12.0.1. Drøfting av metode

Metoden har som formål å klassifisere en batymetrisk punktsky ved bruk av maskinlæring, hvor treningsdata er basert på undervannsbilder som referansedata. Tilnærmingen som er presentert i metode vil her bli vurdert.

Referansedata fra havforskningsinstituttet er godt utnyttet for å bestemme treningsområdene til klassifisering. Innsamlingen har vært en tidkrevende prosess med nøye evalueringer, og det har vært et godt arbeid med å dokumentere treningsområdene. Det har vært viktig å få både nok mengde treningsdata, men også korrekt definisjon av punkter. Det er mer potensiale i referansedataen fra havforskningsinstituttet enn det som har blitt brukt og som kunne blitt brukt. Likevel var det flere ting som var ønskelig å utforske og arbeidet måtte gå videre.

De utvalgte treningsområdene kan heller ikke defineres som en fasit til virkeligheten. Undervannsbildene gir viktig innsikt i hva som eksisterer på havbunn, men det eksisterer ikke en registrert sann posisjon til det som utforskes i undervannsbildene. Arbeidet er basert på interpolerte koordinater og tolkning etter egen vurderingsevne. Det er i tillegg en forskyvning som er forsøkt løst etter beste evne, kapittel 9.1.4, ved hjelp fra synlige bunntype overganger i ortofoto. Det finnes ingen garanti for at absolutt alle treningspunkter er definert riktig i forhold til ønskede bunntyper. Likevel har det vært et solid arbeid med dette og det er gjort fornuftige og gode vurderinger.

Ved utvalg av treningsområder og manuell annotering i *Terrascan* baseres også arbeidet på egen vurderingsevne. Koordinatbestemmelser i tilnærmingene har ikke vært presise posisjonsbestemmelser, men det har vært forsøkt å være så nøyaktig som mulig ved utvelgning og registrering. Til tross for nøyaktig arbeid kan det eksistere ufrivillige forskyvninger, men selv uten forskyvning kan det eksistere ukorrekte definerte punkter på bakgrunn av manuell registrering i *Terrascan*. Slike tilfeller er isåfall antatt å være svært få, og treningsområdene har blitt kontrollert en ekstra gang i *ArcGIS*.

I gjennomgangen av utfordringer i kapittel 11, er det presisert at definisjonen av de ulike bunntypene for metoden bygde på feil punktsky. Ved videre tilnærming av metoden har arbeidet fortsatt med den fullstendige punktskyen, men valg av bunntyper forble uendret. Spredt vegetasjon er en vag bunntype, den hadde som formål å fungere som en mellomklasse for en punktsky med stor spredning. Det var bekymring for områder der det kunne eksistere både sand og vegetasjon, og at det ikke var nok punkter til å kunne fange opp alle detaljer. Spredt vegetasjon er definert som en kombinasjon av sand og tett vegetasjon, på grunn av dette ble det inkludert sandområder for sand klassen selv om det eksisterte litt vegetasjon i undervannsbildene. Dette ble gjort for å styrke sand klassen opp mot spredt vegetasjon ved å velge litt mindre "klar" sandbunn. I utgangspunktet var tilnærmingen svært god og rimelig for daværende punktsky. Det må likevel vurderes hvor godt de ulike bunntypene er definert, hvilken effekt definisjonene kan ha på klassifiseringsmodellene og om de skulle ha vært definert annerledes. Bunntypene, spesielt spredt vegetasjon, kan fremgå som uklare og føre til forvirring for modellene. For eksempel kunne det vært forsøkt å skille mellom vegetasjonstyper, slik som tang og tare.

Resultatet for utvalget av treningsområder, som presentert i kapittel 9.1.5, må også vurderes. Det er her beskrevet at det er forsøkt å fordele treningsdata utover ulike dybder, men av figur 25 er ikke dette oppnådd. Klassene for stein, slam og spredt vegetasjon har lite representasjon på dype områder. Samtidig viser figur 24 at treningsområdene generelt sett befinner seg nærme land. Likevel er det begrensninger for dette i forhold til hvilke områder referansedataen er fra. Resultatet for treningsområdene beskriver lite representasjon av stein og slam, og det fremgår at vann og land er overrepresentert. Dette kan påvirke maskinlæringsmodellen til å prioritere de mest representerte klassene ovenfor de mindre representerte.

Videre tar metoden for seg beregning av geometriske egenskaper. Disse har blitt beregnet for flere radius for å benytte forskjellig skala. De ulike radiusene beskriver forskjellige detaljnivåer i områdene rundt punktene. Her kunne flere radius blitt brukt, større radius for større detaljer, og mindre radius for mindre detaljer. Det kunne også blitt undersøkt flere trinn mellom de brukte radiusene. Valg av radius er imidlertid allerede definert i kapittel 9.2.3. For metoden kunne det være forsøkt med en tilnærming for andre metoder en programvaren *CloudCompare* for å beregne geometriske egenskaper. Det var forsøkt en egen metode, men denne måtte til slutt forkastes.

Et viktig aspekt ved tilnærmingen av maskinlæringsrutinen er oppdeling av data, som er beskrevet i kapittel 9.3.4. For å dele treningsområdene inn i treningsdata og testdata er det kun tatt hensyn til lik andel representasjon av klassene i oppdelingen. Det var ønskelig med oppdeling basert på definerte områder, som ville gitt en romlig fordeling. Dette ble ikke utforsket videre på grunn av ønske om fremdrift i oppgaven. Likevel burde det ha blitt utforsket andre programvarer for å dele opp punktskyen slik at grupperingene kunne ha blitt beholdt inn mot klassifiseringen. En annen tilnærming er en oppdelingsmetode basert på mer uavhengige treningsområder. Mer om dette drøftes videre i resultater for validering.

Metoden for utvalg av egenskaper, med *Random Forest* sin innebygde *Feature Importance*, tyder på at modellen foretrekker egenskaper som baserer seg på punktetthet, som er *Number of neighbors*, *Surface density* og *Volum density*. I tillegg til disse tre egenskapene ser det ut til at den også foretrekker *Verticality* som gjengår i flere forskjellige utvalg, og med flere ulike radius. I forsøkene hvor de punktetthetsbaserte egenskapene ekskluderes er *Verticality* fremdeles favorisert, det dukker også opp egenskaper som ikke har blitt like favorisert i forsøkene som inkluderer punktetthets egenskapene. I disse tilfellene, som blant annet i modell **D** og **E**, dukker egenskaper som *Normal change rate*, *Surface variation* og *3rd eigenvalue* med også ulike variasjoner i radius. Basert på resultatene fra egenskapsutvalget med eget utvalg, modell **B** og modell **C** med alle representasjoner av egenskaper med radius 2, hadde det vært av interesse å utføre et forsøk med de samme egenskapene men uten spredt vegetasjonsklassen for å se om de ulike modellene klarer å plukke ut flere detaljer.

12.0.2. Drøfting av resultater

Valideringsresultatene til de ulike klassifiseringsrutinene bar preg av overtilpasning. Det var ikke mye som skilte de ulike modellene i denne evalueringen, som igjen gjorde det vanskelig å bruke valideringen til å evaluere den faktiske ytelsen til modellene. Testdata var satt til 20% for å beholde mye av treningsdatasettet, men også for å holde av noe til vurdering av modellen. Til tross for dette ble det en overtilpasning, som muligens kunne blitt løst med å sette av mer til testdata. En kjent løsning på overtilpasning er å samle inn mer treningsdata, noe som det ikke var tilstrekkelig med tid til i denne oppgaven. Skulle forsøket ha blitt gjort igjen kunne det tenkes å opprette et eget sett med testdata på likhet med det innsamlede treningsdatasettet. Dette kan føre til større mengder data og at treningsdata og testdata ikke direkte blir korrelert. Med dette menes at punktene verken ville vært direkte eller nære naboer, og den romlige fordeling hadde blitt forbedret. Det ville likevel vært en form for overtilpasning av modellen fordi datasettene ikke er uavhengige av hverandre, og de er hentet fra ett og samme område.

Grunnet overtilpassede valideringsresultater ble det foretatt en visuell inspeksjon av klassifiseringsresultatene. Ortofoto, og treningsområdene som baserte seg på undervannsbilder, ble benyttet som referansedata til denne inspeksjonen. Det er derfor knyttet en viss usikkerhet til resultatene for områder uten referansedata.

I den visuelle inspeksjonen fremgikk det likhetstrekk og forskjeller mellom modellene. Slam var en klasse alle modellene slet med å klassifisere. I område 3 forvekslet modellene slam med tett vegetasjon, og i den smale korridoren blå sand forvekslet med spredt vegetasjon. En mulig forklaring på denne forvekslingen er at det finnes to ulike slam typer. Området hvor slam forveksles med tett vegetasjon, fremstilles som mørk i ortofoto og har tett overflate ved undervannsbilder. I den smale korridoren ser slam ut som sand i ortofoto. Slam som vises på undervannsbilder i dette området har litt grovere overflate og er muligens mer sandholdig, noe som er en mulig forklaring på hvorfor den har blitt forvekslet med spredt vegetasjon.

Rutinene **A**, **B** og **C** er de som ble mest påvirket av overlapp mellom flystripene i punktskyen, noe som har påvirket prediksjonene av de ulike klassene i disse stripene. Disse rutinene har modeller som bruker flere punktegenskaper som baserer seg på punktethet. I figur 30, som viser oversikt over punktethet, fremstår det en sammenheng mellom overlappende områder i kartet og i de predikerte modellene. Basert på denne observasjonen ble disse egenskapene ikke benyttet inn i modellene **D** og **E**. Flystripeeffekten er ikke like synlig i disse modellene som derimot ikke har punktethet baserte egenskaper.

Modell **E** var den klassifiseringsrutinen som ikke inneholdt spredt vegetasjon, dette ble prøvd for å undersøke hvordan modellen ville skille mellom sand og tett vegetasjon, uten mellomklassen spredt vegetasjon. Det ble observert i de tidligere rutinene at spredt vegetasjon overtok mye av sandklassen. Fra resultatene har modell **E** predikert sand godt ved store sandområder som er tydelig i ortofoto. Derimot har det i områder, som ikke har like klart skille mellom ren sand og vegetasjon, blitt derfinert tett vegetasjon. Det er observert flere tilfeller av slam rundt i områder der det mest sannsynlig ikke skal være slam. Dette tyder på at modellen har brukt slam som en slags mellomklasse i enkelte tilfeller hvor de andre modellen har predikert spredt vegetasjon. Dette forsøket i modell **E**, burde også blitt utført med en mer lik tilnærming som modell **A**, **B** og **C** for å kunne vurdere bedre spredt vegetasjon som klasse. Tilnærmingen som ble brukt er i dette tilfellet er ikke direkte sammenlignbart med rutinene **A**, **B** og **C**, på grunn av for ulikt utgangspunkt.

I figur 56 ble det vist at det var forskjeller i fordelingen av predikerte klasser i modellene. Modell **D** og **E** viste en økning i vannklassen, som mulig kommer fra reduksjon av land. I forvirringsmatrisene er det også tydelig at vann og land ble feilklassifisert mellom disse modellene. Gapet mellom land og bunnklassene som går igjen i modellene kan derfor antas å være at modellene predikerer vannoverflate istedet for land. Det som skiller disse to modellene fra de andre er ekskluderingen av punktethetsbaserte egenskaper.

13. KONKLUSJON

Problemstillingen for denne masteroppgaven ble formulert som følger: "Hvordan vil geometriske egenskaper påvirke klassifisering ved bruk av maskinlæring med treningsdata basert på undervannsbilder som referansedata?" Gjennom flere konklusjoner rundt trenings- og referansedata har det blitt forsøkt å belyse hvordan disse geometriske egenskapene påvirker klassifiseringen ved bruk av maskinlæring.

Referansedata har vært viktig for denne oppgaven, men har likevel mer potensiale. Undervannsbildene har blitt grundig utnyttet for å bestemme treningsområder til klassifiseringen og har samtidig blitt godt dokumentert. Dette gir muligheter til påbygging og forbedring i eventuelt fremtidig arbeid.

De valgte treningsområdene representerer ikke en absolutt fasit for virkeligheten, og det er en viss usikkerhet knyttet til definisjonen av punkter. Likevel kan grundig arbeid og gode vurderinger forsøke å håndtere disse usikkerhetene.

Valideringsresultatene var tydelig preget av overtilpasning i alle modellene. Det konkluderes derfor med at denne valideringen ikke er tilstrekkelig for å vurdere ytelsen av de ulike modellene. Økt mengde trenings- og testdata, romlig distribuert for å unngå korrelasjon, er nødvendig for å forbedre evalueringen.

I etterkant av arbeidet med fullstendig punkttsky er det konkludert med at bunntyper må redefineres. Spredt vegetasjon forstyrrer den visuelle inspeksjonen, slik at en videre sammenlikning av de ulike geometriske egenskapene bør foregå uten denne klassen.

Basert på resultatene fra denne oppgaven trengs det videre analyse av de ulike geometriske egenskapene for å avgjøre hvilke som er kritiske inn mot batymetrisk laserskanning og hvilke som er uviktige. Det fremgår at egenskapene *Number of neighbors*, *Surface density* og *Volum density*, som er punktetthetsbaserte egenskaper, påvirker klassifiseringen i den grad at overlapp mellom flystriper forstyrrer klassifiserings resultatet. Videre er det tydelig at det fremgår forskjeller blant de andre egenskapene, noe som ikke er grundig nok analysert i denne oppgaven. I tillegg til punktetthet, er *Verticality* en gjentakende geometrisk egenskap som også bør studeres nærmere.

I konklusjon vil det sies at *Number of neighbors*, *Surface density*, og *Volume density* påvirker klassifiseringen ved bruk av maskinlæring med treningsdata basert på undervannsbilder som referansedata. De øvrige geometriske egenskapene bidrar, etter visuell inspeksjon, positivt til de ulike klassifiseringsmodellene. Videre arbeid bør fokusere på å forstå detaljene i hvordan disse egenskapene påvirker klassifiseringen, og spesielt vurdere *Verticality* som en potensiell viktig faktor.

14. VIDERE ARBEID

Utforskningen av klassifisering av batymetrisk flybåren laser basert på referansedata fra havforskningsinstituttet har vist seg å ha flere kritiske aspekter. Basert på oppnådde funn og gode diskusjoner i løpet av masteroppgaven er det flere muligheter for utvikling. Dette kapitlet vil belyse ulike retninger for å ta arbeidet videre, i tillegg til hva som kan videreutvikles basert på funn.

Grunnlaget for treningsdata er nøye plukket ut og godt dokumentert. Dette er et godt utgangspunkt for mange ulike problemstillinger, hvor treningsdatasettet kan brukes videre. Det finnes også muligheter for flere definisjoner av ulike bunntyper som kan implementeres, og bunntypene kan også defineres mer konkret slik som tang og tare. Grunnlaget for treningsdata kan også brukes videre med andre typer punktegenskaper, eller mot klassifisering på objektnivå, slik det er beskrevet i tidligere arbeid.

De klassifiserte punktskyene har også flere muligheter for videre arbeid. Blant annet skulle masteroppgaven i hovedsak sammenlikne flybåren batymetrisk laserskanning og multistråle ekko-lodd, dette ble for omfattende for denne oppgaven. Mulige fremgangsmåter ved sammenlikning er å bruke den klassifiserte punktskyen fra batymetrisk laserskanning, og segmentere punktskyen etter bunntype. Deretter kan en gjøre flere undersøkelser mellom målemetodene for å se på likheter og forskjeller mellom punktskyene fra metodene innad i bunntypene. Her kan også segmentering av ulik homogenitet av terrengform undersøkes, og det kan også segmenteres som en kombinasjon av bunntype og terreng. Er det variasjon mellom målemetodenes punktskyer, endres variasjonen mellom bunntyper eller terrengformer, og endres variasjon mellom kombinasjoner av bunntyper og terrengformer?

I punktskyen er det ulik punktetthet for ulike dybdenivåer. Det hadde derfor vært av interesse å segmentere punktskyen basert på disse nivåene, slik at hvert nivå fikk egenskaper som passer til punktskyens lokale tetthet. Som nevnt i oppgaven, vil ulik radius påvirke den geometriske egenskapen og størrelsen på detaljer den beskriver, men ulik punktetthet gjør også at en må velge radius deretter for å kunne beregne egenskaper for alle punkter. For eksempel ett segment på et grunt område hvor det er større punktetthet, her kan en beregne geometriske egenskaper for en mindre radius og fange opp små detaljer. I ett annet segment på et dypere område, hvor det er lavere punktetthet, kan radius økes for å beregne nok geometriske egenskaper. Dette fører til at en ikke glatter ut egenskapsverdiene på grunne områder for at en skal ta hensyn til alle dybdenivåer i punktskyen. Videre kan de lokalt tilpassede geometriske egenskapene brukes inn i maskinlæringsrutinen, som mulig kan gi mer detaljerte predikasjoner av sjøbunn. Spesielt med tanke på dype områder.

Referanser

- Abhinav, R. (2023), 'Improving class imbalance with class weights in machine learning'. [https://medium.com/@ravi.abhinav4/improving-class-imbalance-with-class-weights-in-machine-learning-af072fdd4aa4\(H](https://medium.com/@ravi.abhinav4/improving-class-imbalance-with-class-weights-in-machine-learning-af072fdd4aa4(H) 13.12.2023).
- Applied Imagery (2022), 'Quick terrain modeler'. <https://appliedimagery.com> (lest: 09.12.2023).
- Atik, M., Duran, Z. & Seker, D. (2021), 'Machine learning-based supervised classification of point clouds using multiscale geometric features'. https://www.mdpi.com/2220-9964/10/3/187?fbclid=IwAR2k4oHhGSAjTA92StCNsrxrbvaFJz44zNA7yv4-SVAWvP_0PB43o1jtttLM (lest: 13.12.2023).
- BasuMallick, C. (2023), 'What is principal component analysis (pca)? meaning, working, and applications'. <https://www.spiceworks.com/tech/big-data/articles/what-is-principal-component-analysis/> (lest: 09.12.2023).
- Bell, A., Chambers, B., Butler, H. et al. (2023), 'Dimensions'. <https://pdal.io/en/2.6.0/dimensions.html> (lest: 06.12.2023).
- Blomley, R., Weinmann, M., Leitloff, J. & Jutzi, B. (2014), 'Shape distribution features for point cloud analysis - a geometric histogram approach on multiple scales'. <https://d-nb.info/1145337546/34> (lest: 11.12.2023).
- de Jongh, C (2021), The advantages of airborne lidar bathymetry, [video], The Hydrographic Society UK & Ireland. Hentet fra <https://www.youtube.com/watch?v=PQPIBraEHMM> den 06.12.23.
- Demantké, J., Vallet, B. & Paparoditis, N. (2012), 'Streamed vertical rectangle detection in terrestrial laser scans for facade database production'. <https://isprs-annals.copernicus.org/articles/I-3/99/2012/isprsannals-I-3-99-2012.pdf> (lest: 09.12.2023).
- dos Santos, R., Galo, M. & Habib, A. (2022), 'K-means clustering based on omnivariance attribute for building detection from airborne lidar data'. <https://www.proquest.com/docview/2665340653?pq-origsite=gscholar&fromopenview=true> (lest: 06.12.2023).
- Dybedal, J., Aalerud, A. & Hovland, G. (2019), 'Embedded processing and compression of 3d sensor data for large scale industrial environments'. https://www.researchgate.net/publication/330892083_Embedded_Processing_and_Compression_of_3D_Sensor_Data_for_Large_Scale_Industrial_Environments?tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6Ii19kaXJlY3QiLCJwYXVwIjoieX2RpcmVjdCJ9fQ (lest: 28.11.2023).
- Esri (n.d.), 'Arcgis pro'. <https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview> (lest: 10.12.2023).
- Girardeau-Montaut, D. (2015), 'Cloudcompare octree', (CloudCompareWiki). https://www.cloudcompare.org/doc/wiki/index.php/CloudCompare_octree(lest: 14.11.2023).

- Girardeau-Montaut, D. (2017), 'Introduction', (CloudCompareWiki). <https://www.cloudcompare.org/doc/wiki/index.php/Introduction>(lest: 20.11.2023).
- Girardeau-Montaut, D. (2019), 'Geometric feature indices and what they mean.'. I en diskusjon på CloudCompare sitt forum, ga Daniel, CloudCompareAdmin, følgende informasjon om PCA beregning i CloudCompare. Svaret ble gitt i tråden Geometric feature indices and what they mean.den [13.10.2019]. (<https://www.danielgm.net/cc/forum/viewtopic.php?t=3768> (lest: 09.12.2023)).
- Girardeau-Montaut, D. (2022a), 'Curvature', (CloudCompareWiki). <https://www.cloudcompare.org/doc/wiki/index.php/Curvature> (lest: 27.11.2023).
- Girardeau-Montaut, D. (2022b), 'Density', (CloudCompareWiki). <https://www.cloudcompare.org/doc/wiki/index.php/Density> (lest: 27.11.2023).
- Girardeau-Montaut, D. (2022c), 'Roughness', (CloudCompareWIKI). <https://www.cloudcompare.org/doc/wiki/index.php/Roughness> (lest: 27.11.2023).
- Grobbelaar, J. (2009), 'Turbidity, encyclopedia of inland waters'. <https://www.sciencedirect.com/science/article/pii/B9780123706263000752> (lest: 09.12.2023).
- Guo, B., Liu, X. & Tan, X. (2017), 'Chapter 14 - hydraulic fracturing'. <https://www.sciencedirect.com/science/article/pii/B9780128093740000143> (lest: 09.12.2023).
- Hackel, T., Wegner, J. & Schindler, K. (2016), 'Contour detection in unstructured 3d point clouds', (ETH Zürich). <https://ethz.ch/content/dam/ethz/special-interest/baug/igp/photogrammetry-remote-sensing-dam/documents/pdf/timo-jan-cvpr2016.pdf> (lest: 15.11.2023).
- Har'el, Z. (1995), 'Curvature of curves and surfaces - a parabolic approach', (Israel Institute of Technology). <http://harel.org.il/zvi/docs/parabola.pdf> (lest: 27.11.2023).
- Jia, C., Wang, C., Yang, T., Fan, B. & He, F. (2019), 'A 3d point cloud filtering algorithm based on surface variation research on the innovation of protecting intangible cultural factor classification factor classification'. **Forkortet link**(lest: 08.12.2023).
- Jupyter (2023), 'Jupyter'. <https://jupyter.org> (lest: 09.12.2023).
- Koehrsen, W. (2017), Random forest simplified, [bilde], Medium.com. <https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d> (lest: 11.12.2023).
- Kogut, T. & Slowik, A. (2021), 'Classification of airborne laser bathymetry data using artificial neural networks'. <https://ieeexplore.ieee.org/document/9320614?fbclid=IwAR2WvtfSsX6gLsCvC-24dxRFyofSaMJETTZc9qx6MtMNuIhoMzjGh7pynTY> (lest: 13.12.2023).
- Liberman, N. (2017), 'Decision trees and random forests'. <https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991> (lest: 11.12.2023).

- Løyning, S. & Storsveen, O. (2020), Klassifisering av sjøbunnen ved bruk av batymetrisk, flybåren lidar, [masteroppgave], Norges miljø- og biovitenskapelige universitet. https://nmbu.brage.unit.no/nmbu-xmlui/bitstream/handle/11250/2676962/Masteroppgave_Loyning_Storsveen.pdf?sequence=1&isAllowed=y (lest: 13.12.2023).
- Mandlbürger, G. (2020), 'A review of airborne laser bathymetry for mapping of inland and coastal waters.'. <https://henry.baw.de/server/api/core/bitstreams/4c3b5bd5-9ad5-4e71-b744-292ae52d956e/content> (lest: 03.12.2023).
- Mandlbürger, G (2022), schematic representation of airborne laser bathymetry fra a review of active and passive optical methods in hydrography, [bilde], The international Hydrographic review. <https://ihr.iho.int/wp-content/uploads/2022/12/IHR-22-A15.pdf> (lest: 03.12.2023).
- Mosa, A., Schön, B., Bertolotto, M. & F. Laefer, D. (2012), 'Evaluating the benefits of octree-based indexing for lidar data'. <https://www.ingentaconnect.com/content/asprs/pers/2012/00000078/00000009/art00001?crawler=true> (lest: 28.11.2023).
- Nababan, B., Ulfah, D. & Panjaitan, J. (2021), 'Light propagation, coefficient attenuation, and the depth of one optical depth in different water types'. <https://iopscience.iop.org/article/10.1088/1755-1315/944/1/012047/pdf> (lest: 12.12.2023).
- Navigator, A. (2018), 'Anaconda navigator'. <http://chat.openai.com/c/1eb44944-f4a2-4281-80bc-c73e314ad238> (lest: 09.12.2023).
- Norges geologiske undersøkelse (2022), Kartkatalog: Bunn sedimenter (kornstørrelse), detaljert, [datasett], Geonorge.no. <https://kartkatalog.geonorge.no/metadata/bunn-sedimenter-%kornstoerrelse-detaljert/79f0f17d-9f62-456d-b1a9-%2a8c754c51c4> (lest: 05.11.2023).
- Otto, Y. (2018), 'Exploring kd-tree in point cloud'. <https://medium.com/@OttoYu/exploring-kd-tree-in-point-cloud-c9c767095923> (lest: 06.09.2023).
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011), Scikit-learn: a random forest classifier, [dokumentasjon]. Hentet fra <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> 11.12.23.
- Philpot, W. (2019), 'Airborne laser hydrography ii'. <https://hdl.handle.net/1813/66666> (lest: 03.12.2023).
- Richter, K., Maas, H. & Westfeld, P. (2017), 'An approach to determining turbidity and correcting for signal attenuation in airborne lidar bathymetry'. <https://doi.org/10.1007/s41064-016-0001-0> (lest: 09.12.2023).
- scikit-learn developers (n.d.).

- Skaar, J. & Holtsmark, T. (2020), 'Snells lov i store norske leksikon på snl.no.'. Hentet 3. desember 2023 fra https://snl.no/Snells_lov.
- Stella, A. (2023), 'Estimating surface normals in a pointcloud', (PointCloudLibrary GitHub). https://pcl.readthedocs.io/projects/tutorials/en/master/normal_estimation.html(lest: 23.11.2023).
- Szafarczyk, A. & Toś, C. (2022), 'The use of green laser in lidar bathymetry: State of the art and recent advancements'. <https://www.mdpi.com/1424-8220/23/1/292> (lest: 04.12.2023).
- Terrasolid (2023), 'Managing and processing point clouds'. <https://terrasolid.com/products/terrascan/> (lest: 09.12.2023).
- Terratec AS (2021), 'Marine grunnkart i kystsonen - pilot område stavanger. prosjektrapport'. https://hoydedata.no/LaserServices/REST/DownloadPDF.ashx?filePath=\statkart.no_orig13597ÄylÄyy%20vÄër%20-MGK%20ALB%20Pilot%202021_prosjektrapport.pdf (lest: 11.12.2023).
- The SciPy community (2023), 'scipy.spatial.kdtree'. (SciPy Manual)<https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.KDTree.html> (lest: 05.09.2023).
- Tidemann, A. & Elster, A. (2023), 'Snells lov i store norske leksikon på snl.no.'. Hentet 11. desember 2023 fra <https://snl.no/maskinlÄering>.
- Vosselmann, G. & Maas, H.-G. (2010), Aireborne and Terrestrial Laser Scanning, Whittles Publishing.
- Webb, R. (2023), 'Introduction to oceanography, chapter 6.5 light'. <https://rwu.pressbooks.pub/webboceanography/chapter/6-5-light/> (lest: 04.12.2023).

Python Bibliotek

- laspy (2022), 'Python library for lidar las/laz io'. <https://laspy.readthedocs.io/en/latest/>(Hentet: 11.12.2023).
- NumFOCUS (2023), 'pandas documentation'. <https://pandas.pydata.org/docs/>(Hentet: 11.12.2023).
- scikit-learn developers (2023a), 'Feature importances with a forest of trees'. https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html(Hentet: 12.12.2023).
- scikit-learn developers (2023b), 'sklearn.ensemble.randomforestclassifier'. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn-ensemble-randomforestclassifier>(Hentet: 13.12.2023).
- scikit-learn developers (2023c), 'sklearn.feature_selection.selectkbest'. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html(Hentet: 12.12.2023).
- scikit-learn developers (2023d), 'sklearn.metrics.confusion_matrix'. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html(Hentet: 13.12.2023).
- scikit-learn developers (2023e), 'sklearn.metrics.f1_score'. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html#sklearn-metrics-f1-score(Hentet: 13.12.2023).
- scikit-learn developers (2023f), 'sklearn.metrics.log_loss'. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html(Hentet: 13.12.2023).
- scikit-learn developers (2023g), 'sklearn.metrics.precision_score'. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html(Hentet: 13.12.2023).
- scikit-learn developers (2023h), 'sklearn.metrics.recall_score'. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html(Hentet: 13.12.2023).
- scikit-learn developers (2023i), 'sklearn.metrics.roc_auc_score'. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html(Hentet: 13.12.2023).
- scikit-learn developers (2023j), 'sklearn.model_selection.randomizedsearchcv'. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html#sklearn-model-selection-randomizedsearchcv(Hentet: 13.12.2023).
- scikit-learn developers (2023k), 'sklearn.model_selection.stratifiedshufflesplit'. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html(Hentet: 12.12.2023).
- The SciPy community (2023), 'scipy.spatial.cKDtree'. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.cKDTree.html>(Hentet: 11.12.2023).

Vedlegg: Treningsdata

Treningsområder Masteroppgave Eira Sørsdal og Kamilla Brynildsen

Innsamling av treningsdata i Terrascan

Dette dokumentet er en dokumentasjon på arbeidet med innsamling av treningsdata i Terrascan, og har blitt brukt til å samle inn treningsdata som er manuelt annotert i Terrascan. Dokumentet er også en referanse til de valgte treningsområdene som er benyttet i masteroppgaven.

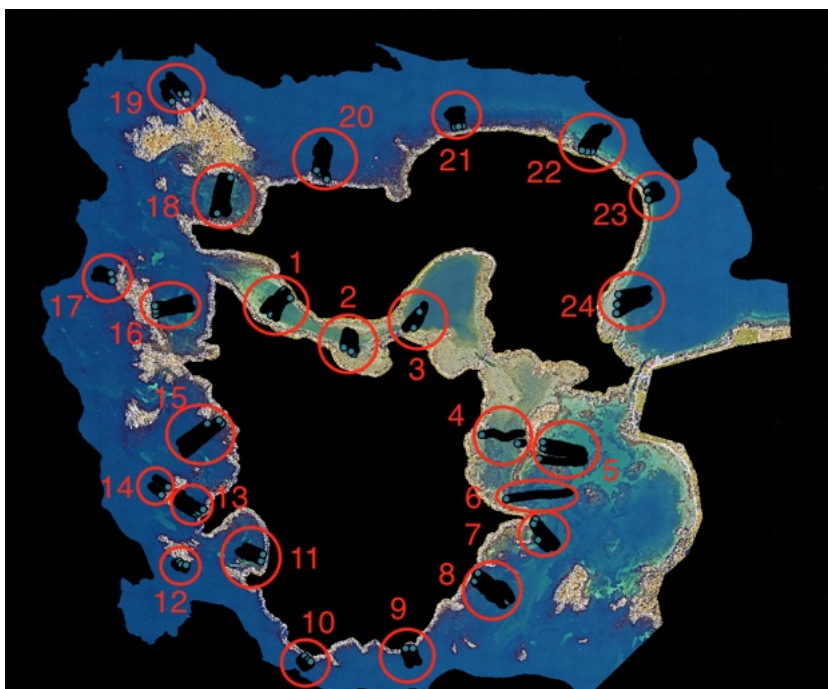
Koordinater og punkter er basert på interpoleringen av undervannsbilder fra Havforskningsinstituttet, mer om dette arbeidet kan leses om i oppgaven.

Alle områdene starter med et oversiktsbilde som viser alle sporene (tracks) som er tilstede i det tildelte område. Deretter så er det blitt gjennomgått spor for spor. Hvert annoterte området i det aktuelle sporet er dokumentert med tilhørende FID, klasse og bredde som skal settes på tverrsnitt. Det er også tatt med et bilde fra ArcGIS-prosjektet som viser strekning og tilhørende undervannsbilde, med mindre det er fanget treningsdata utenfor dekkningen av undervannsbilder. I tillegg er det inkludert koordinater som er brukt for å plukke ut tverrsnittet i Terrascan, samt bildenavn ved start og slutt koordinat på sporet. Enkelte steder er det også tatt med utsnitt fra Terrascan.

Undervannsbildene fra Havforskningsinstituttet er delt inn i tre mapper:

- P1_100
- P2_100
- P2_101

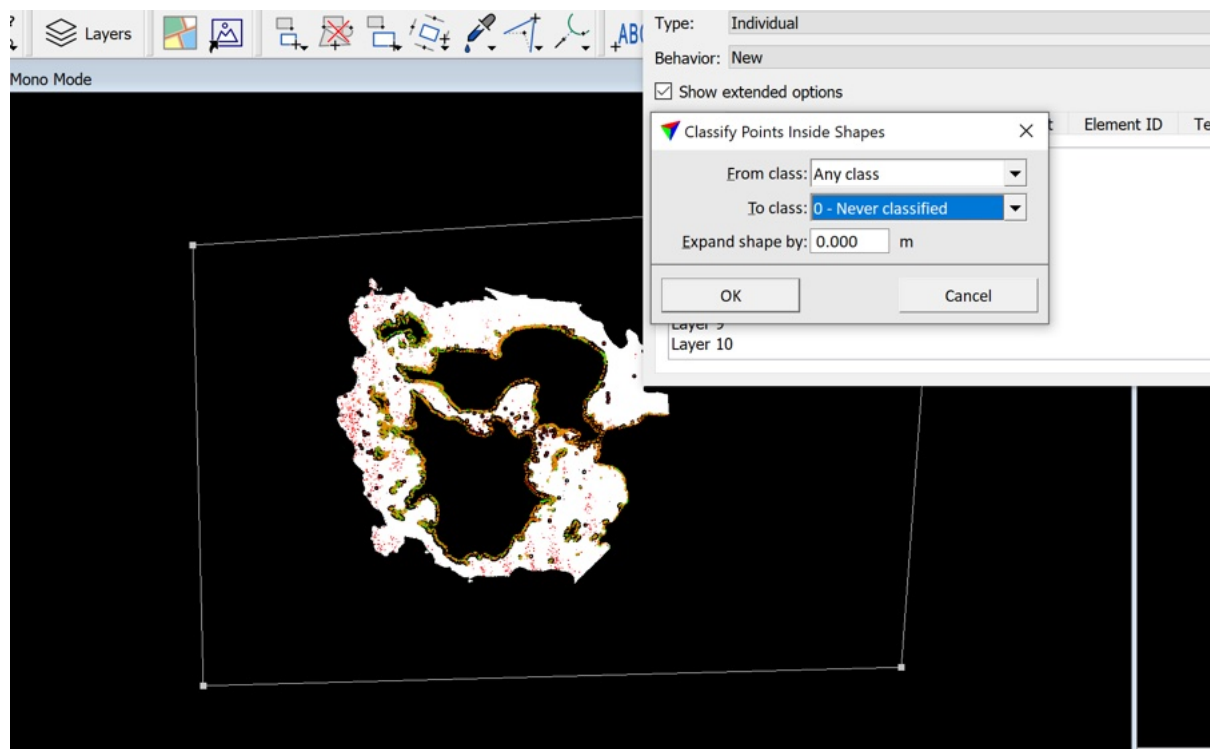
Oversiktsbilde treningsområder



Vedlegg: Treningsdata

Rensing av punktsky:

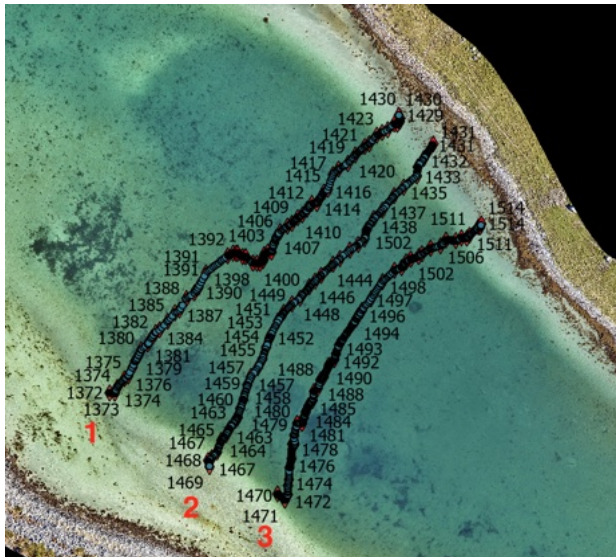
Før innsamling av treningsdata i Terrascan så ble alle klassene i punktskyen satt til null.



Vedlegg: Treningsdata

Innsamling av treningsdata

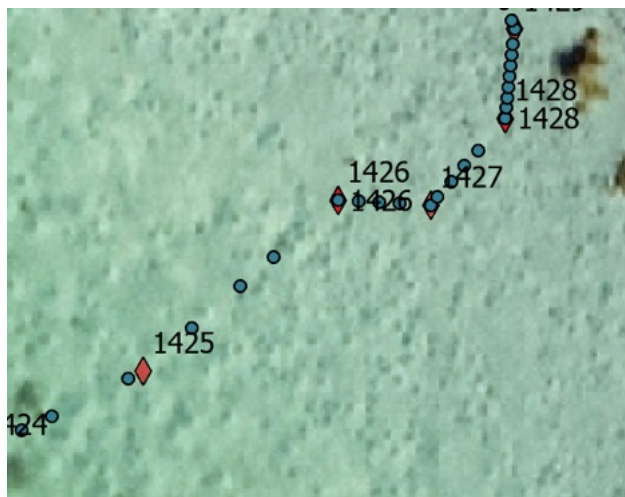
Område 1:



Bilder: P2-100

Track 1:

Sand FID 1425-1427: 2m



PARA7072

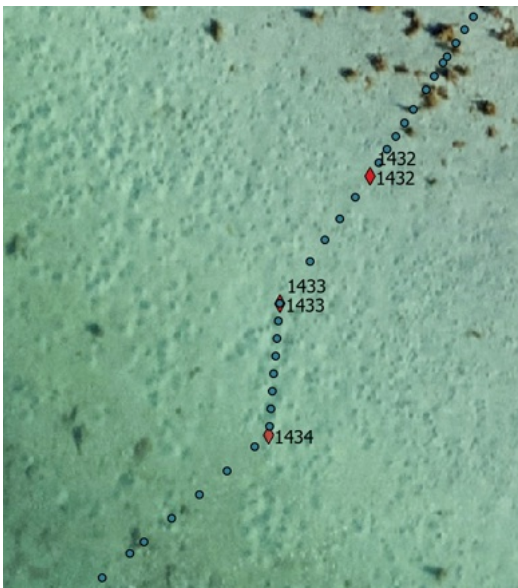
Vedlegg: Treningsdata



7067
303847.77 6556270.28
303851.51 6556272.42
7077

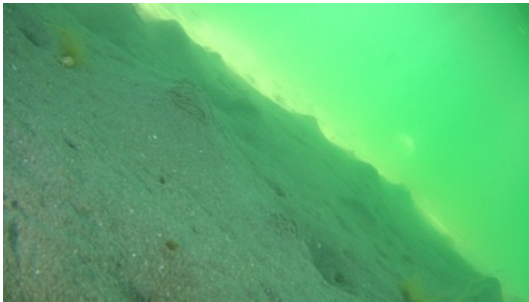
Track 2:

Sand FID 1432 - 1434: 3m



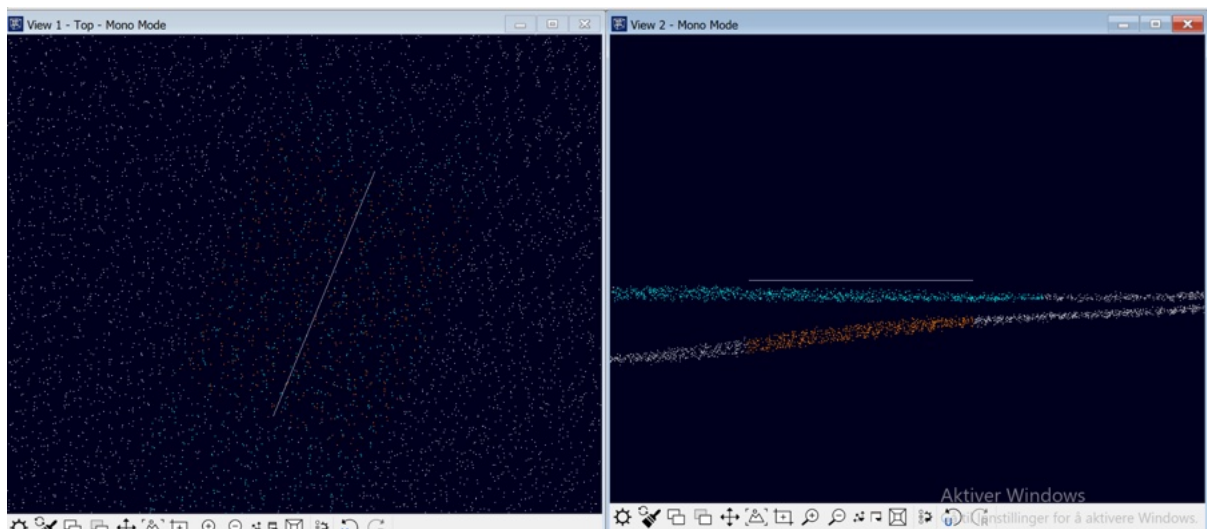
PARA7161

Vedlegg: Treningsdata



303861.57 6556259.98
303858.80 6556253.30

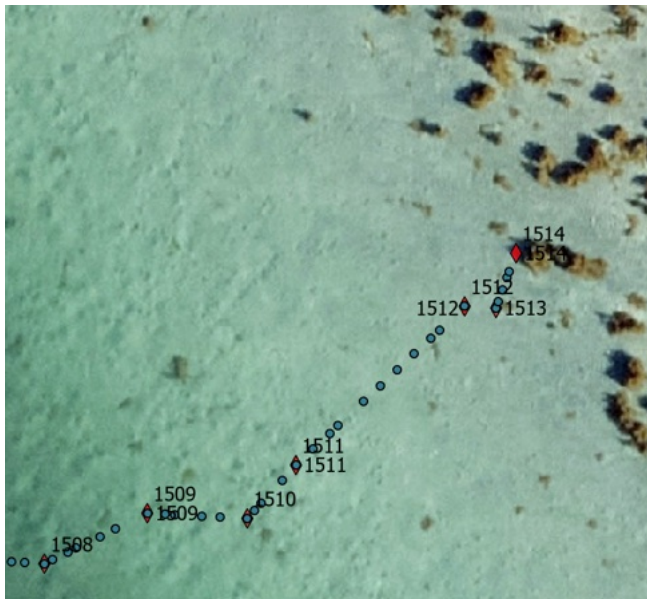
Skjerm bilde Terrascan:



Track 3:

Sand FID 1511-1512: 2m

Vedlegg: Treningsdata



PARA7851



7846

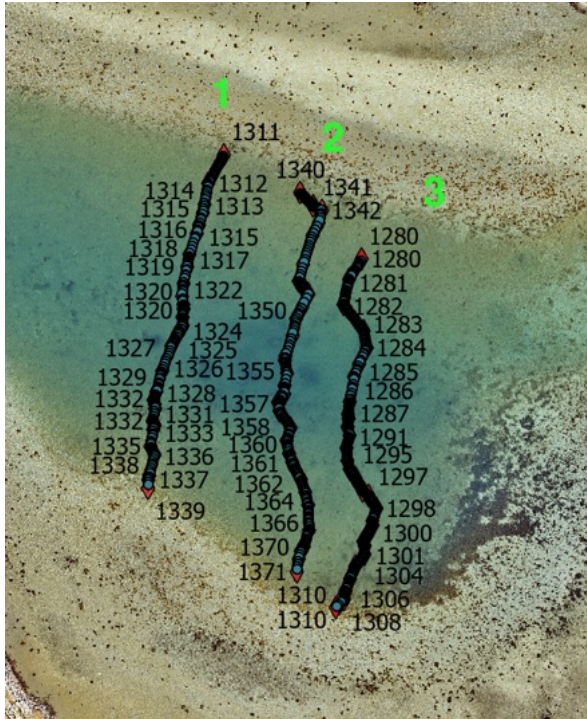
303878.24 6556230.89

303881.62 6556234.06

7856

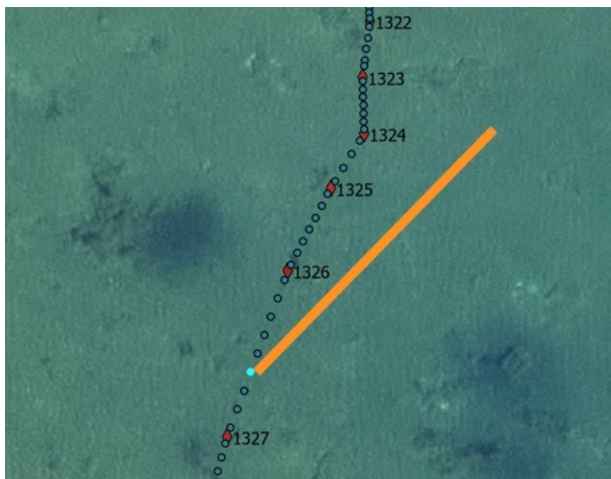
Vedlegg: Treningsdata

Område 2:



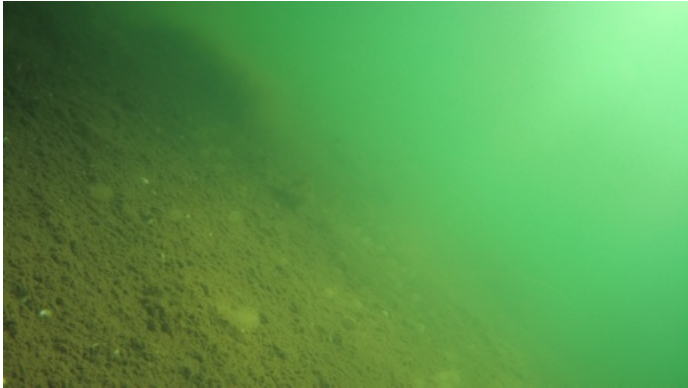
Track: 1

Slam (sandholdig slam) 3m

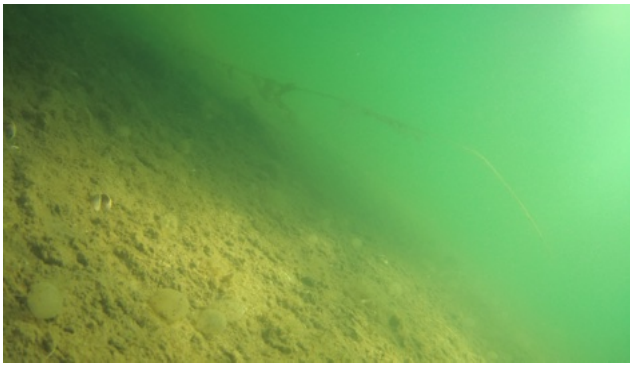


PARA6427

Vedlegg: Treningsdata



PARA6369



6424

304167.00 6556007.84

304176.65 6556017.06

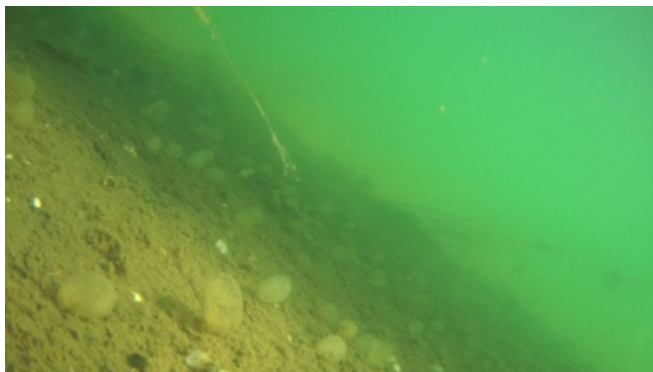
Track 2

Slam FID 1361- 1364: 3m

Vedlegg: Treningsdata



PARA6667



6665

304199.85 6555983.16

304202.65 6555977.06

6682

Track 3:

Slam FID 1286- 1287: 2m

Vedlegg: Treningsdata



PARA6083



6078

304217.01 6556003.79

304215.94 6555997.55

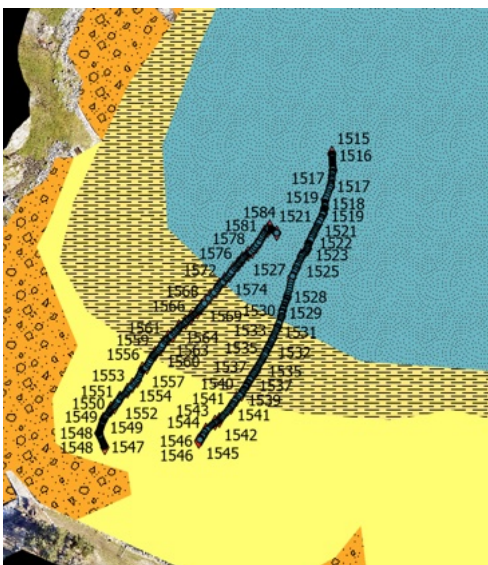
6087

Vedlegg: Treningsdata

Område 3:



Bunnsedimentkart:

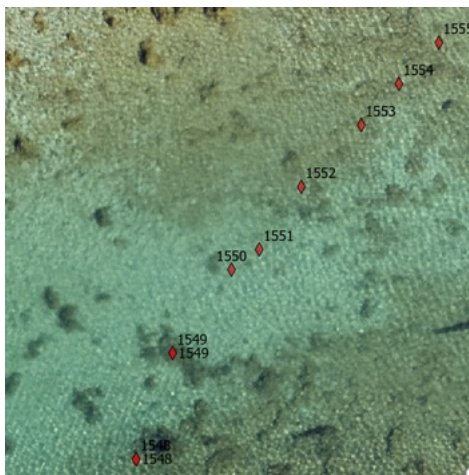


-  Sandholdig slam
-  Slamholdig sand
-  Sand
-  Sand, grus og stein

Track 1:

Sand nesten FID 1549-1551: 2m

Vedlegg: Treningsdata



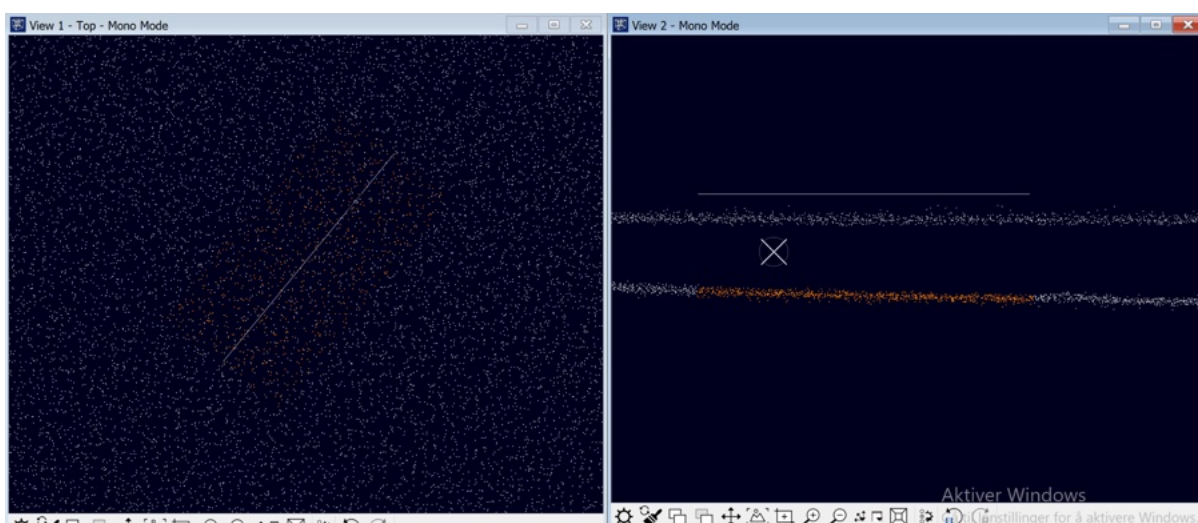
7177

304489.14 6556087.71

304493.78 6556093.36

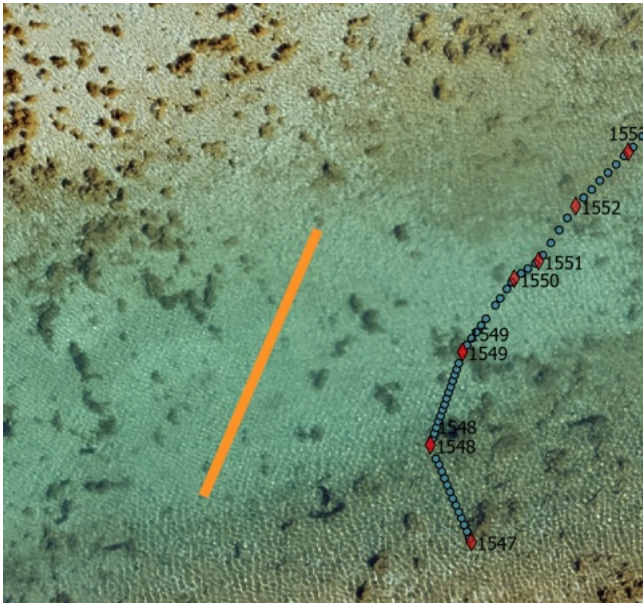
7190

Skjermbilde Terrascan:



Sand litt vest for track 1: 3m

Vedlegg: Treningsdata

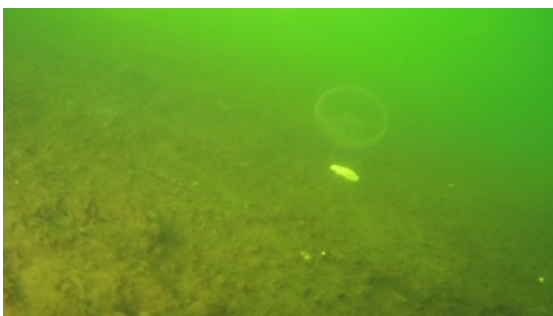


304480.03 6556094.23
304473.07 6556078.20

Slam FID 1582- 1577: 3m



PARA7298



Vedlegg: Treningsdata

7305

304,553.56E 6,556,162.33N

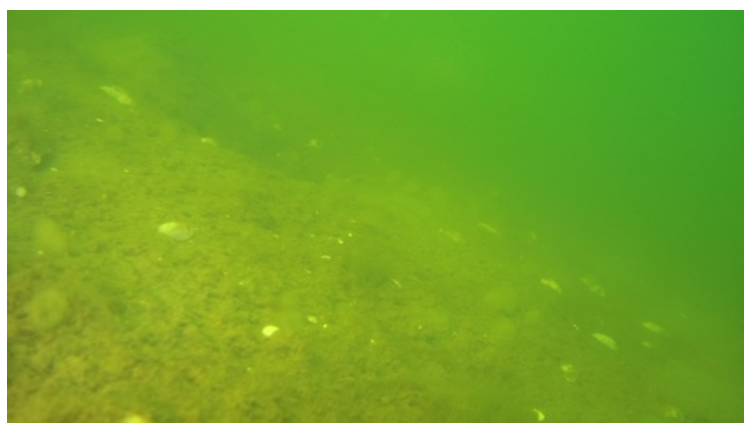
304,545.13E 6,556,152.36N

7289

Slam FID 1563-1567: 3m



PARA7241



7239

304,517.20E 6,556,121.90N

304,526.43E 6,556,130.54N

7257

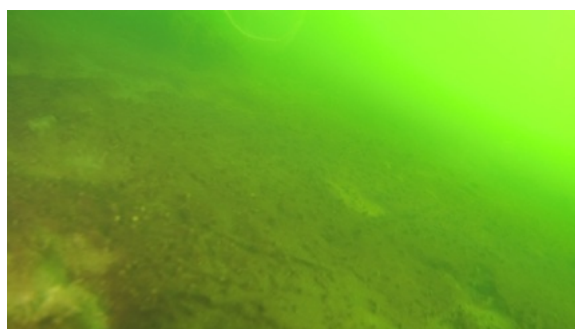
Track 2:

Slam FID 1516- 1517: 3m

Vedlegg: Treningsdata



PARA6921



6911

304581.430058 6556187.542054

304580.547166 6556181.502164

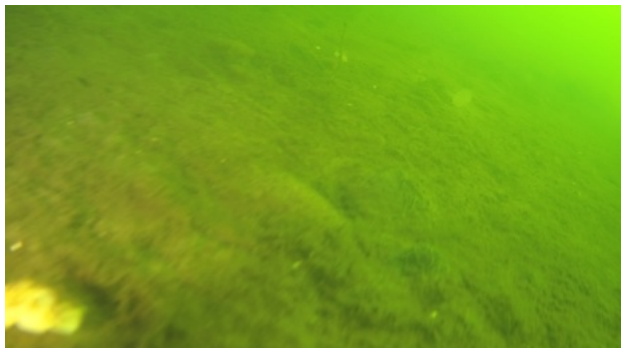
6929

Slam FID 1539- 1533: 3m



Vedlegg: Treningsdata

PARA7043



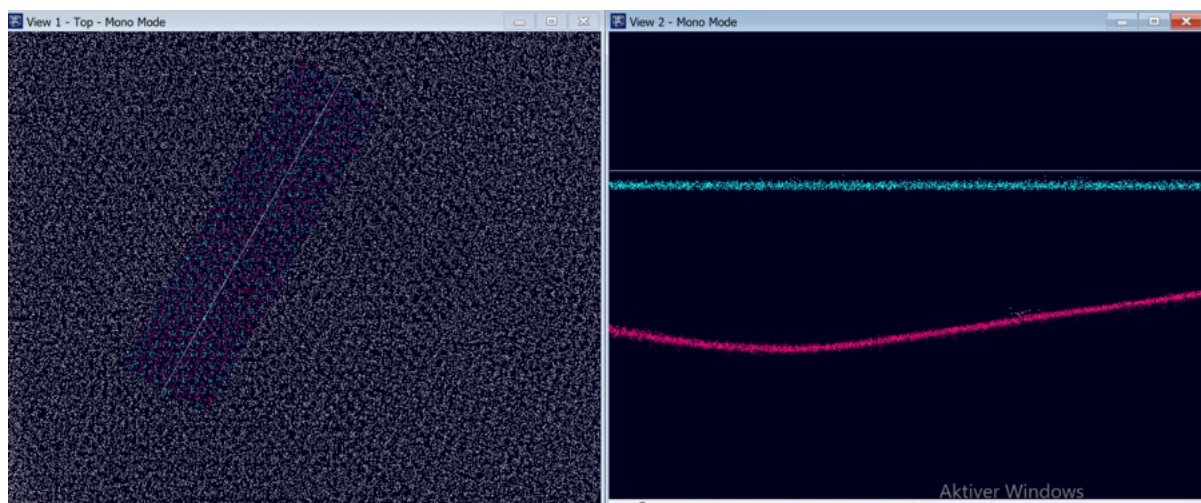
7051

304,544.67E 6,556,097.72N

304,555.52E 6,556,116.69N

7020

Skjermbilde Terrascan:



Lite eksempel: avvik på bunn, ikke registrert som slam.

Vedlegg: Treningsdata

Annet:

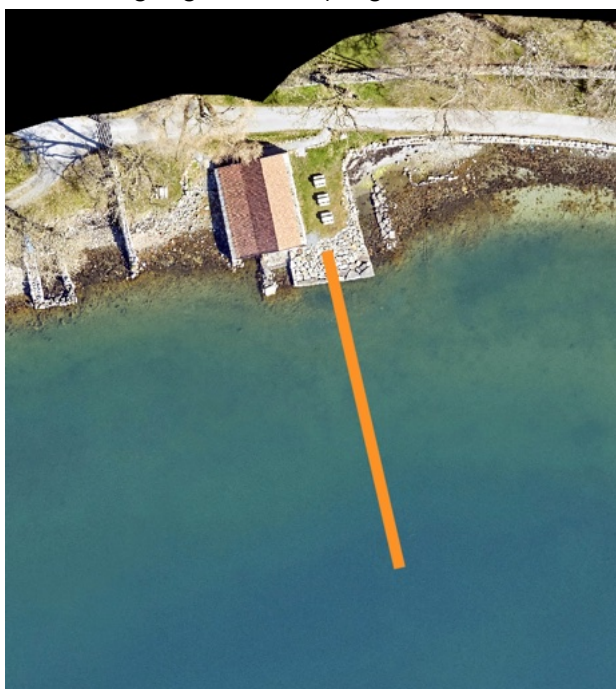
Vann 5m



304724.70 6556337.33

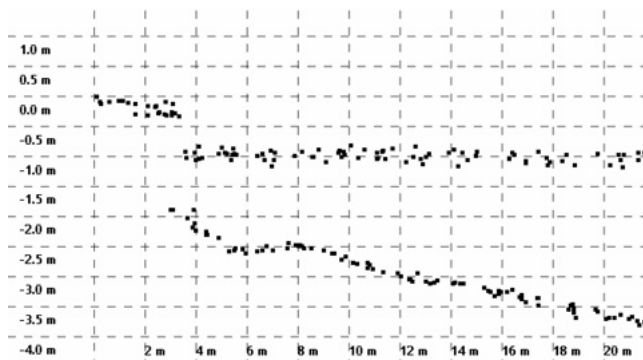
304670.49 6556236.20

Vann overgang fra land: (velg bredde under 5m)



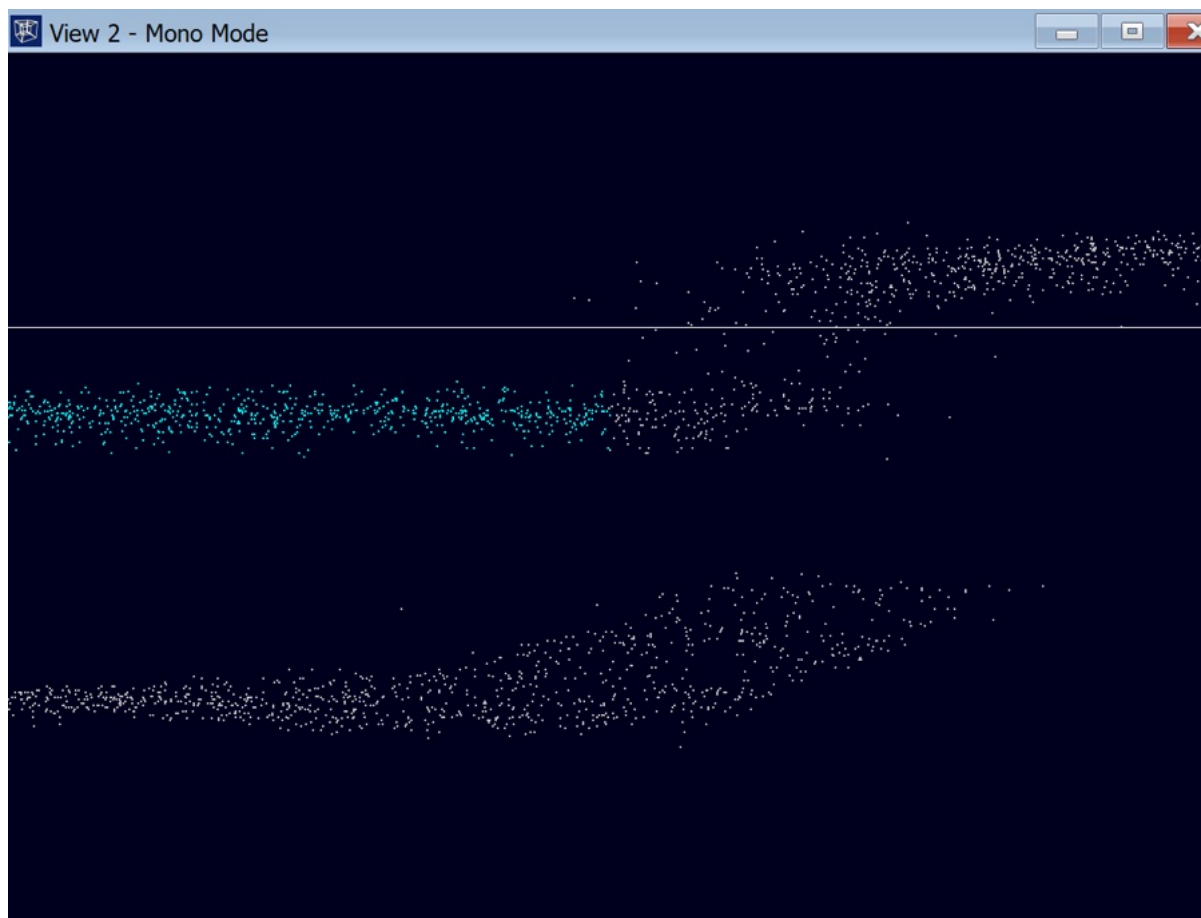
OBS! Vannet grenser brygga ca ved 304,775.05E 6,556,459.67N

Vedlegg: Treningsdata



304774.74 6556463.15
304788.19 6556416.49

Skjerm bilde Terrascan:



Vanskelig å si hvor bryggekannten treffer vannet. En del støy. Stoppet defineringen hvor vannpunktene ble mer spredt.

Land :1.5m

Vedlegg: Treningsdata



304515.02 6556251.22
304498.15 6556274.62

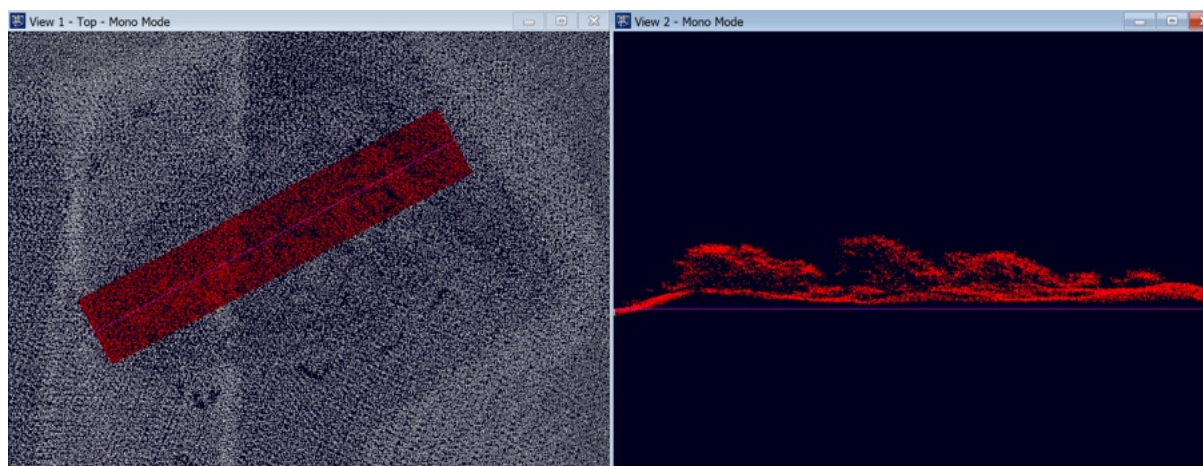
Land 5m

Vedlegg: Treningsdata



304676.75 6556033.02
304726.54 6556059.55

Skjerm bilde Terrascan:



Land: 3m

Vedlegg: Treningsdata



304781.66 6556494.08
304840.18 6556490.63

Land: 3m



304836.35 6556472.56
304865.48 6556470.31

Vedlegg: Treningsdata

Område 4:



Sand ved FID 2415- 2417: 2m



PARA9024



9022

304,939.11E 6,555,527.07N

304,924.93E 6,555,524.82N m

9038

Sand FID: 2414- 2411 2m



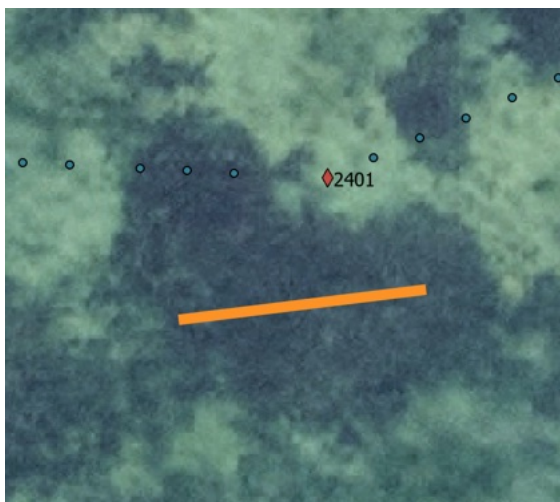
PARA9009

Vedlegg: Treningsdata



9018
304,942.86 6,555,527.08
304,961.76 6,555,520.74N
8996

Tett vegetasjon rundt FID 2401 1.5m



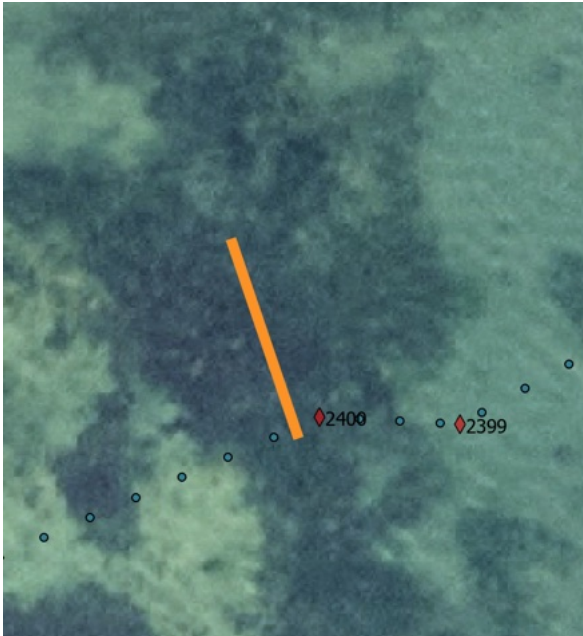
PARA8935 (punkt første til venstre for FID2401)



305,010.17E 6,555,501.03N
305,006.10E 6,555,500.16N

Vedlegg: Treningsdata

Tett vegetasjon rundt FID 2400 1 m



PARA8926

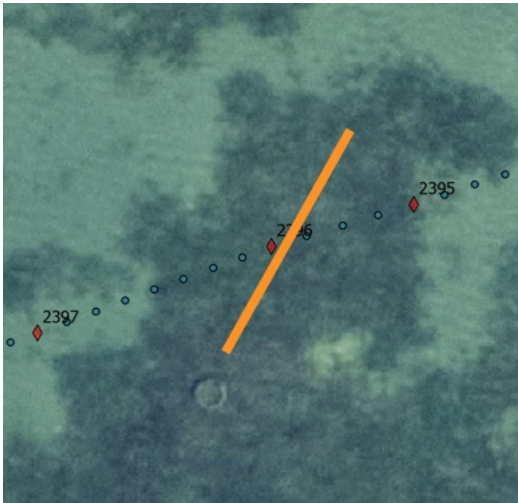


305,012.77E 6,555,504.35N

305,011.90E 6,555,506.94N

Tett vegetasjon ved FID 2396: 1.5m

Vedlegg: Treningsdata

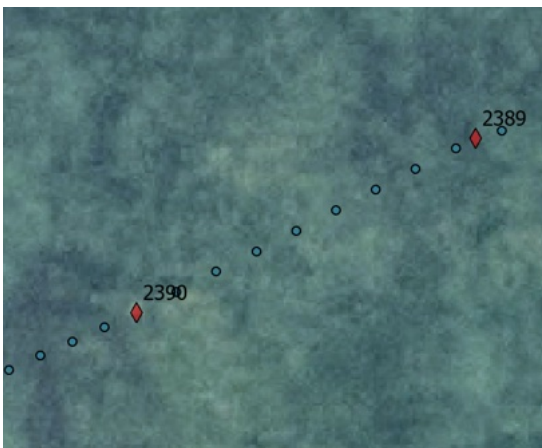


PARA8908



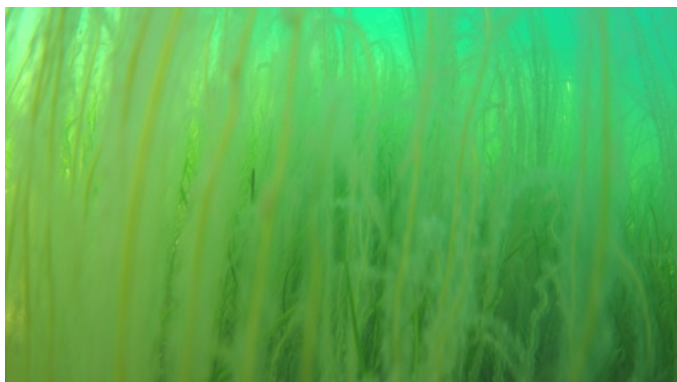
305,028.30E 6,555,511.99N
305,025.76E 6,555,506.95N

Tett vegetasjon FID 2390-2389: 2m



PARA8868

Vedlegg: Treningsdata



PARA8874



8873

305044.952797

655517.106645

305049.413671

655519.349823

8863

Sand mellom FID 2382-2380: 3m



PARA8811

Vedlegg: Treningsdata



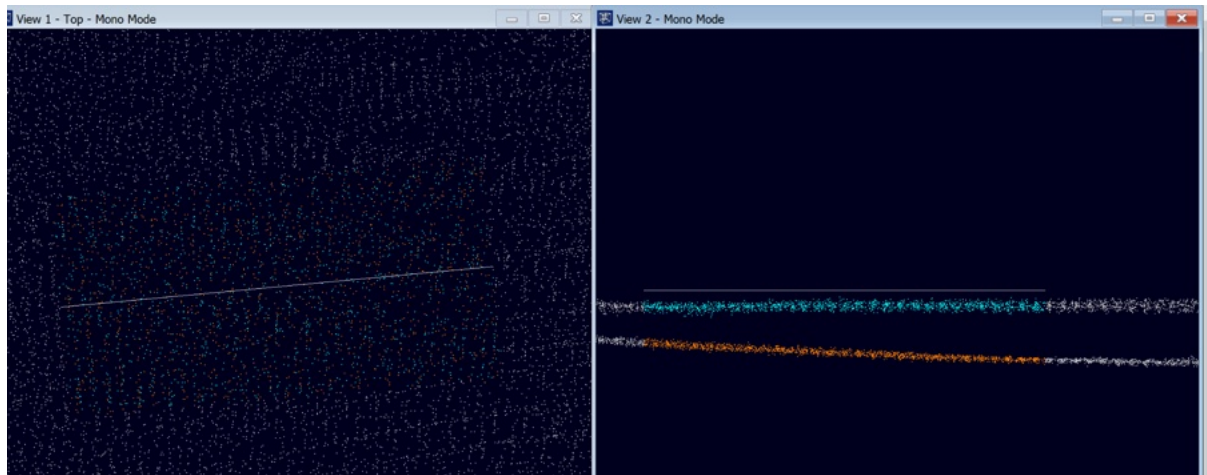
8820

305080.702744 6555525.07317

305092.395343 6555526.171058

8802

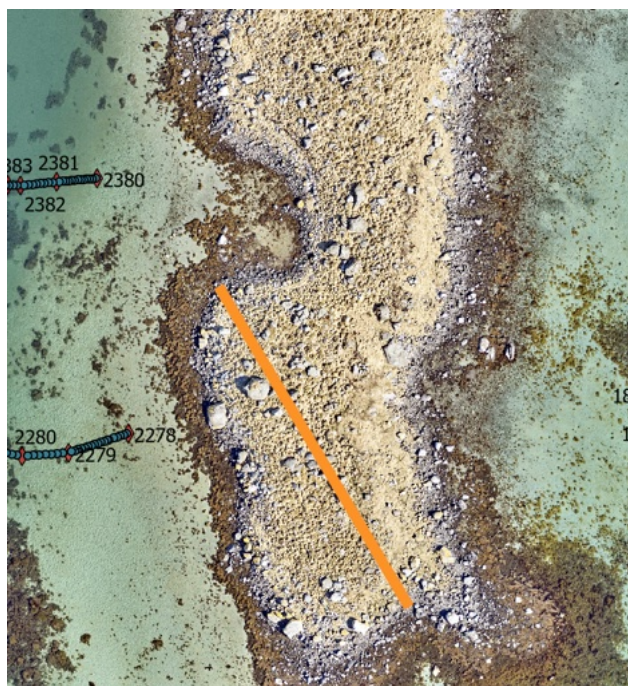
Skjermbilde Terrascan:



Annet:

Land 3m

Vedlegg: Treningsdata



305116.66 6555503.70

305154.44 6555440.36

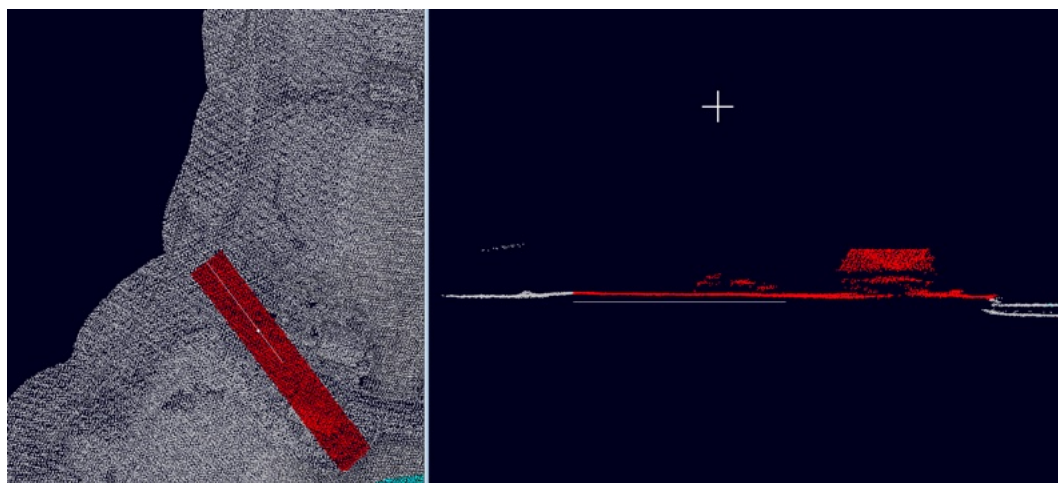
Land: 5m

Vedlegg: Treningsdata



304895.90 6555556.89
304876.07 6555583.01

Skjerm bilde Terrascan:



Tok med litt ekstra landområdet, fikk med bygning.

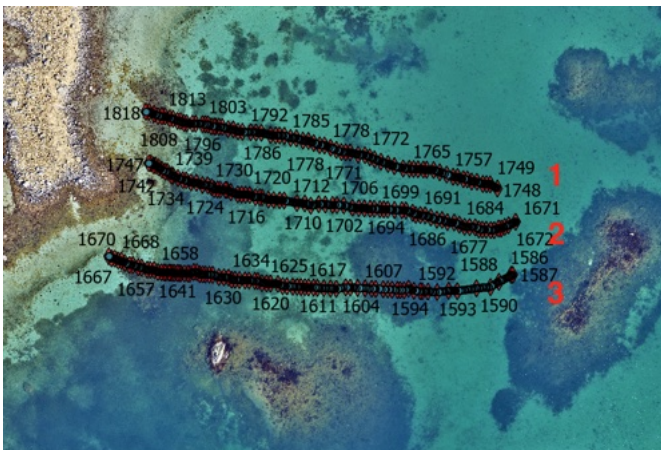
Land 5m

Vedlegg: Treningsdata



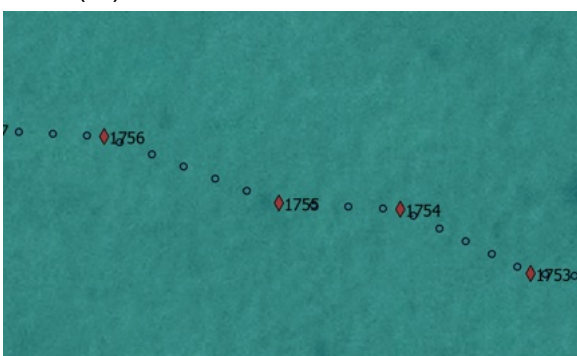
304845.59 6555541.51
304863.77 6555578.55

Område 5:



Track 1:

Sand (-5) FID 1754- 1756: 3m





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

Postboks 5003
NO-1432 Ås
Norway