

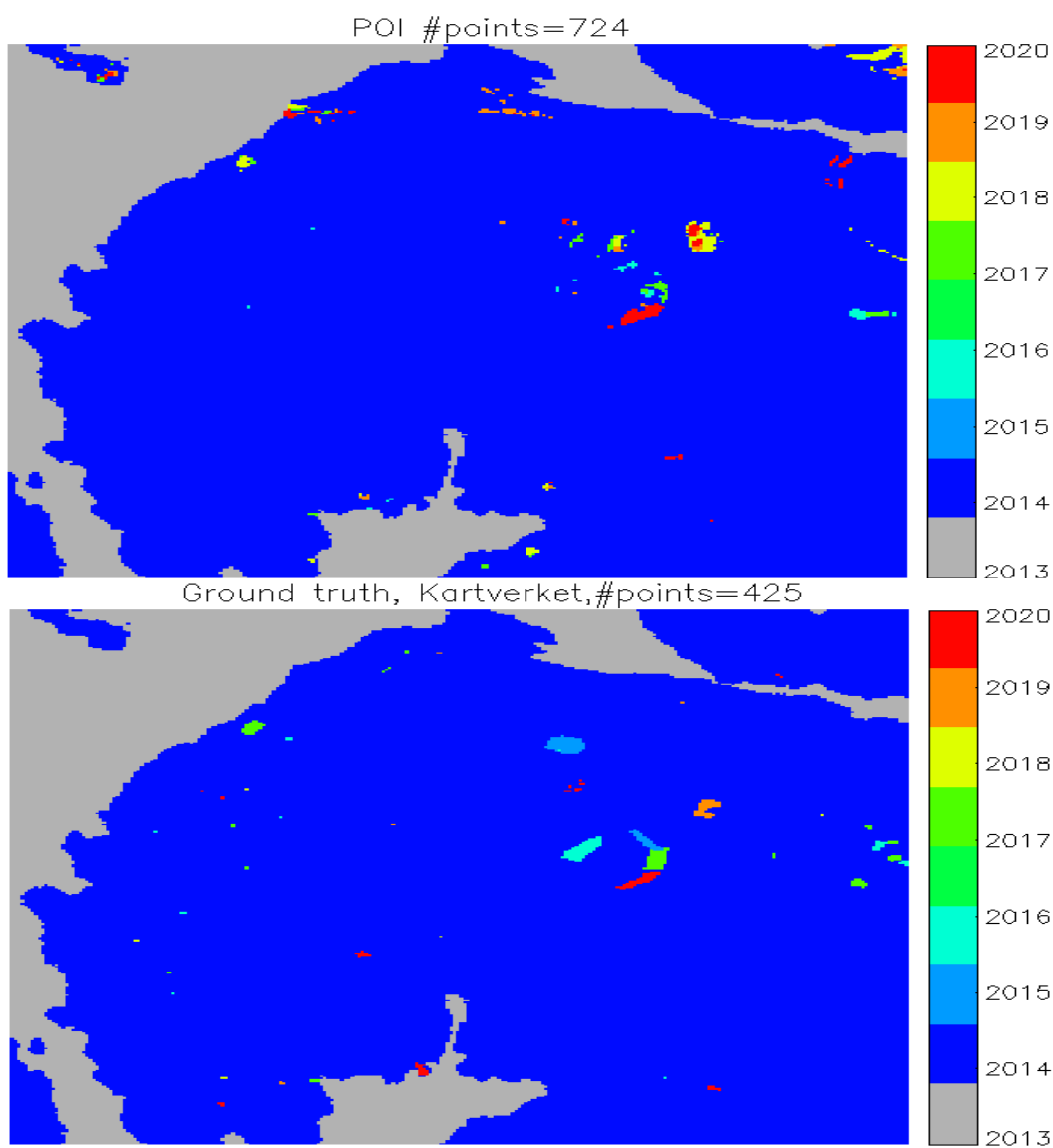
SENBYGG

Sluttrapport

Forfattere:

Eirik Malnes, Stian Normann Anfinsen, Tom Rune Lauknes, Jörg Haarpaintner, Lotte Wendt og Jakob Grahn

Rapport 1-2023, NORCE Teknologi



Rapporttittel	SENBYGG Sluttrapport
Prosjektnummer	105108
Institusjon	NORCE Teknologi
Oppdragsgiver	Statens kartverk
Gradering	Åpen
Rapportnr	1-2023
ISBN	978-82-8408-275-2
Antall sider	49
Publiseringsdato	Februar, 2023
CC-lisens	CC BY 4.0
Bildekreditering	Noen bilder i rapporten er hentet fra
Geografisk område	Norgebilder.no Kristiansand (Søm og Vågsbygd)
Stikkord	Bygningsdeteksjon, SAR, Endringsdeteksjon, AI
Sammendrag	

Rapporten dokumenterer resultater fra prosjektet SENBYGG der Statens kartverk er oppdragsgiver. Prosjektet har som formål å detektere bygningsendringer (nybygg, tilbygg eller revet bygg) ved hjelp av radarsatellittene Sentinel-1 A og B. I prosjektet har vi studert flere mulige metoder for endringsdeteksjon. Radarsatellitter (SAR) har relativt god oppløsning (10m) men betydelig med støy. For å redusere støyen midler vi bilder over samme område for hvert kalenderår. Endringer detekteres ved å sammenligne tilbakespredning fra to etterfølgende år. I prosjektet har vi påvist at det er fullt mulig å detektere bygningsendringer i SAR bilder basert på årlige middelbilder. Man kan også redusere tidsintervallene noe, men antall feil vil øke siden støyene i bildene da blir mer merkbar.

Revisjoner

Rev.	Dato	Forfatter	Kontrollert av	Godkjent av	Årsak til revisjon/
1	28.1.23	EM,SA,JH,TRL,JG,LW	KAH	KAH	Første versjon
2	28.02.23	EM	KAH	KAH	Endelig versjon

Ansvarsavgrensning

NORCE er ikke i noen henseende ansvarlig for den aktuelle bruk av dokumenter, programvare eller andre verktøy eller prosjektresultater og påtar seg derved intet ansvar eller garanti for den helhetlige funksjonalitet ved bruk av informasjonen om dette ikke er spesifikt angitt i tilbudsdokumentet og den etterfølgende kontrakt.

Sammendrag

Rapporten dokumenterer resultater fra prosjektet SENBYGG der Statens kartverk er oppdragsgiver. Prosjektet har som formål å detektere bygningsendringer (nybygg, tilbygg eller revet bygg) ved hjelp av radarsatellittene Sentinel-1 A og B. I prosjektet har vi studert flere mulige metoder for endringsdeteksjon. Radarsatellitter (SAR) har relativt god oppløsning (10m) men betydelig med støy. For å redusere støyen midler vi bilder over samme område for hvert kalenderår. Endringer detekteres ved å sammenligne tilbakespredning fra to etterfølgende år. I prosjektet har vi påvist at det er fullt mulig å detektere bygningsendringer i SAR bilder basert på årlige middelbilder. Man kan også redusere tidsintervallene noe, men antall feil vil øke siden støyene i bildene da blir mer merkbar.

Innhold

1.	Innledning	5
1.1.	Arbeidsbeskrivelse	5
1.1.1.	WP 1: Tilrettelegging av data	5
1.1.2.	WP 2: Metoder for endringsdeteksjon.....	6
1.1.3.	WP 3: Avslutning av prosjektet	6
2.	Teori.....	7
2.1.	Spredning fra objekter	7
2.2.	Kort oversikt over litteratur på fagfeltet.....	8
3.	Preprosessering	10
3.1.	Sentinel-1-data	10
3.2.	Geokodet tilbakespredning.....	12
3.3.	Koherens	12
4.	Endringsdeteksjon	15
4.1.	Endring i tilbakespredning (enkeltgeometrier).....	15
4.2.	Temporale endringer.....	19
4.3.	Endring i tilbakespredning (innfallsvinkelkorrigerte bilder).....	20
4.4.	Endring i koherens (en geometri).....	21
4.5.	Kombinert endringsdeteksjon (koherens og tilbakespredning).....	23
4.6.	Metoder for endringsdeteksjon basert på maskinlæring	23
4.6.1.	Inndata til maskinlæringsalgoritmene	24
4.6.2.	Referansedata og usikre merkelapper	25
4.6.3.	Klassifisering av tidsrekker	26
4.6.4.	Kontekstuell klassifisering	27
4.6.5.	Valgte metoder for tidsrekkeklassifisering	28
4.7.	Falsk alarmrate	30
4.8.	Klassifisering av endringer.....	32
4.9.	Følsomhet for innstillinger for deteksjonsalgoritmen	33
4.10.	Visualisering i NLIVE	34
5.	Resultat og diskusjon	35
5.1.	Resultat fra område Vest (Vågsbygd).....	35
5.2.	Resultat fra område Øst (Sem).....	36
5.3.	Diskusjon	36
6.	Programvare	38
6.1.	Endringsdeteksjon basert på terskling	38
6.2.	Endringsdeteksjon basert på maskinlæring	38
6.3.	Eksport av endringsdeteksjonsbilde til GML.....	38
7.	Oppsummering	40
8.	Referanser	41
9.	Appendiks: SNAP	42

1. Innledning

Rapporten gir en oversikt over arbeidet utført i prosjektet SENBYGG «Anskaffelse av konsulentbistand-Muligheter for å detektere bygg og bygningsendringer med Sentinel-1» finansiert av Statens kartverk med kontrakt-nummer 166801057. Prosjektets målsetning var å avdekke potensialet som er mulig å oppnå med Sentinel-1. I tillegg er det et krav om at metoden bør kunne kategorisere detekterte bygg som uendret, nytt, tilbygg eller revet bygg.

NORCE har utført arbeidet som spesifisert i kontrakten med Statens kartverk. Vi finner at det er mulig i mange tilfeller å detektere bygningsendringer med Sentinel-1 satellittene. Det er ofte dessuten mulig å angi om endringene skyldes nybygg, revet bygg eller bygningsendringer, men nøyaktigheten i slike kategorier vil avhenge sterkt av en rekke faktorer (geometrier, størrelsen på endringene, naturlige variasjoner som snø, jordbruk, fuktighet m.fl.) og det har ikke vært mulig i prosjektet å lage et system som kan løse alle slike problemer.

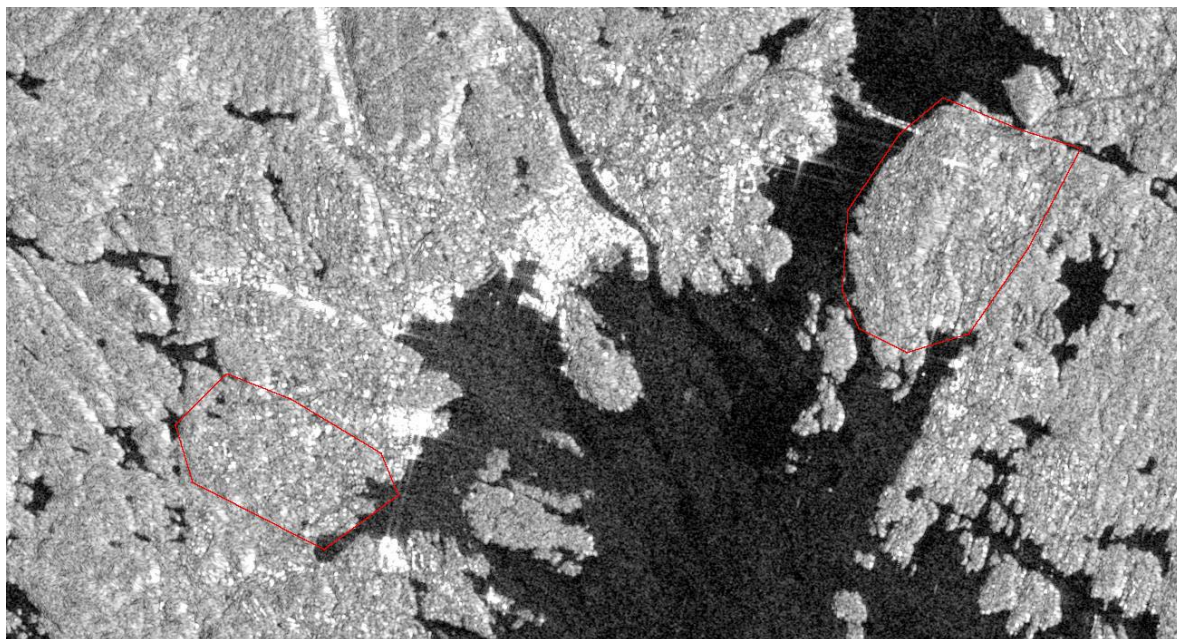
1.1. Arbeidsbeskrivelse

Prosjektet var organisert i 3 hovedarbeidspakker

1.1.1. WP 1: Tilrettelegging av data

Prosjektet startet opp med å samle inn all tilgjengelig Sentinel-1 data over de to aktuelle studieområdene. Data ble deretter prosessert i data-stacker for de to områdene. Første steg i pre-prosessering er å geokode alle scener til de aktuelle area of interest (AOI). Geokoding er prosessering som transformerer SAR bilder fra GRD-formatet som leveres fra ESA (SciHub eller ColHub) til et griddet format f eks geotif i en veldefinert kartprojeksjon. I vårt tilfelle bruker vi WGS-84 UTM sone 33N (som er den projeksjonen vi har lagret vår felles høydemodell for Norge, 10m). En viktig del av geokodingen består i å korrigere for terrenget ved hjelp av høydemodellen. Siden SAR avbilder med skrått innfall er det viktig å korrigere for høyden til hvert piksel. I tillegg til terrengekorrigerings brukes også de medfølgende kalibreringsinformasjon i SAFE-filene til å transformere fra det komprimerte formatet i SAFE-filene til tilbakespredning på float-format.

To varianter av geokoding ble anvendt, en der man ikke prøver å korrigere for innfallsvinkel og en der innfallsvinkel variasjoner blir korrigert for. I tillegg ble endringsmålet koherens, som er et biprodukt av en såkalt intererometrisk prosessering, beregnet. Til sammen er det snakk om ca. 1350 satellittbilder. I tillegg til satellittdata ble det også samlet inn en del andre datatyper som masker for ulike tema som innsjø, hav, skog mm der endringsdeteksjon vil være vanskelig. I tillegg ble det samlet inn ulike versjoner av kartdata fra de to områdene for å skaffe tilveie bakkesannhet om bygningsendringene fra kartverkets egne databaser. Disse kartene skulle seinere vise seg å være vanskelig å bruke til direkte trening av endringsdeteksjonsmetoder. For de to områdene hadde vi også tilgjengelig flyfoto fra Norge i Bilder.



Figur 1. SAR bilde over Kristiansand med inntegnet avgrensning for områdene Vågsbygd (vest) og Søm (øst).

1.1.2. WP 2: Metoder for endringsdeteksjon

WP2 utgjør hoveddelen av prosjektet. Her testet vi ut en rekke metoder for midling av data og endringsdeteksjon basert på terskler eller dynamisk basert på maskinlæring (AI).

1.1.3. WP 3: Avslutning av prosjektet

I denne arbeidspakken ble prosjektet oppsummert gjennom denne rapporten samt at programvaren som ble utviklet i prosjektet ble tilrettelagt på GitHub.

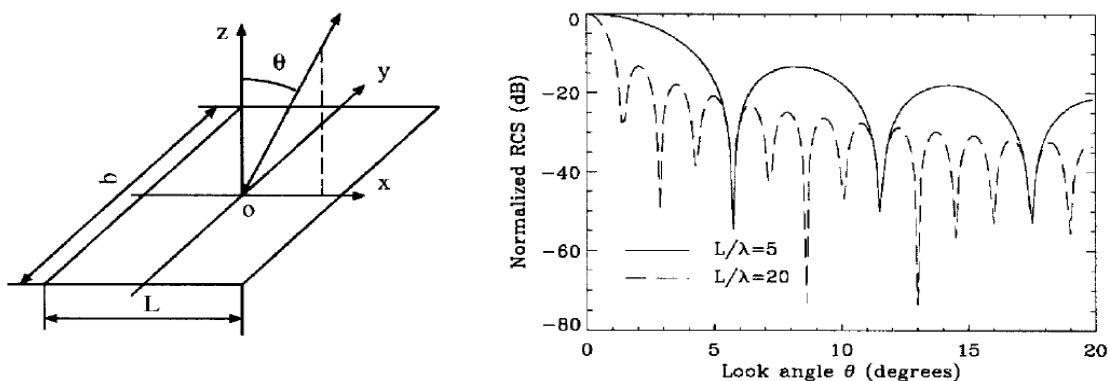
2. Teori

Radarsatellitter sender ut radarpulser og måler tilbakespredt energi fra bakken. Ved å benytte såkalt Syntetisk Apertur prosessering av dataene kan man lage bilder av bakken med relativt høy oppløsning (typisk fra 50 cm opp til 100 m) avhengig av radarenes egenskaper (puls-kompresjon/chirp båndbredde, pulsrepetisjonsfrekvens, innfallsvinkel og sporbredde). For Sentinel-1 er det Interferometric Wide moden som er aktuell. Denne operasjonsmoden har en oppløsning på 10 m og det er to aktuelle SAR produkter som er brukt i dette prosjektet. Disse er på formatet single look complex (SLC) og ground range detected (GRD) som blir brukt til hhv. interferometrisk prosessering (koherens) og til å hente ut målinger av tilbakespredning. Endringer på 10 m skala er dermed den minste endringen man i praksis kan måle med denne satellitten. Dette setter noen begrensninger på hvor små endringer man kan detektere.

SAR bilder er beheftet med multiplikativ støy (spekkelstøy). Dette er støy som er forholdsvis betydelig i opprinnelig oppløsning (10m). For å fjerne denne støyen kan man enten filtrere romlig (med vektet eller uvektet middelværdi over nabopikser) eller temporalt (midle bilder med samme geometri). Jo mer man midler, desto mindre støy får man i bildene, men dette går jo utover den romlige og/eller temporale oppløsninga.

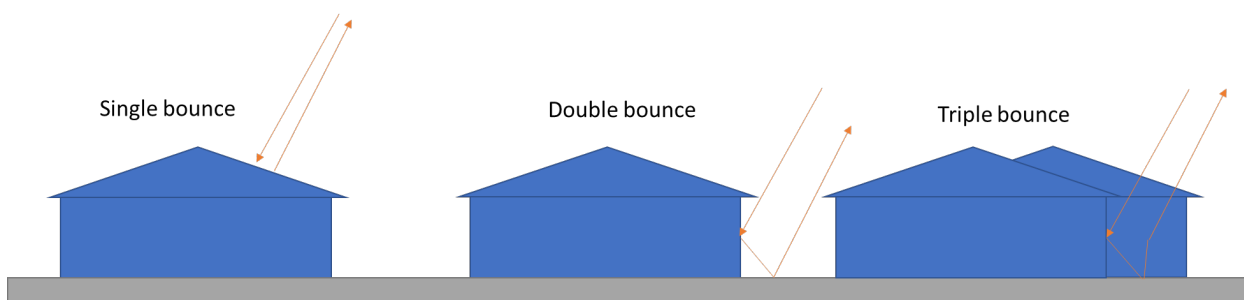
2.1. Spredning fra objekter

Tilbakespredning fra bygninger er relativt komplisert å modellere. Figur 2 under viser spredning fra en idealisert flat plate. Når platens utstrekning L er stor i forhold til bølglengden ($\lambda = 5\text{cm}$ for Sentinel-1) dannes et komplisert strålingsdiagram med mange side-lober. Bygninger vil i praksis ha mye mer kompliserte strålingsdiagram enn idealiserte plater.



Figur 2. Spredning fra flat plate

For bygninger vil mange faktorer spille inn: Oppløsning, innfallsvinkel, retning, bygningens form og orientering samt byggematerialer. I tillegg vil enkel, dobbel og trippel-bounce effekter spille inn der VV og VH polarisasjon kan oppføre seg ulik.



Figur 3. Spredning fra bygninger

For en radar som Sentinel-1 med moderat oppløsning (10m) vil husklynger og omkringliggende strukturer (plen, skog, vei etc.) også kunne spille inn. Andre faktorer som biler, anleggsmaskiner/heiser, containere etc. som er i bevegelser på byggeplasser vil også kunne forårsake endringer fra bilde til bilde. Til sammen utgjør mulighetsrommet for alle endringer som kan forekomme i en by et altfor komplisert bilde til at det er mulig å modellere disse effektene i detalj. Vår tilnærming må bli statistisk, der vi midler så mange satellittbilder som mulig. Håpet er da at de dominerende endringene som er forholdsvis stabile over tid vil framtre tydelig.

2.2. Kort oversikt over litteratur på fagfeltet

Det er skrevet mange artikler om bruk av SAR til urban kartlegging og bygningsdeteksjon. I dette avsnittet nevner vi noen artikler, men det anbefales å søke på Google scholar med søkeord som «Sentinel-1», «building», «change detection» osv.

Chini et al. 2018 viser at automatisk byggdeteksjon kan gjøres på global skala med 20 m oppløsning med akseptabel nøyaktighet (92% sammenlignet med data fra TerraSAR-X med høyere oppløsning, antatt som bakkesannhet). For å oppnå dette har forfatterne brukt intensitet (tilbakespredning) i to polarimetrisk kanaler til deteksjon og interferometrisk kohrens til å fjerne falske deteksjoner.

Koppel et al. (2017) har studert kvantitativt hvordan ulike bygningsparametere (høyde, orientering, materiale, tetthet og form) påvirker tilbakespredning fra S1 Høyde er generelt den viktigste parameteren, men innfallsvinkel fra satellitten er også viktig. I tillegg har bygninger av glass eller metall sterkere signal enn tre/steinbygg. VV-kanalen er generelt mer robust i klassifisering enn VH. Klimatiske forhold som snø, samt vegetasjonsmessige forhold (skog eller landbruk) kan være med på påvirke bygningsdeteksjon. Dette er delvis belyst i Bartch et al. (2020).

Gruenhagen og Juergens (2022) har studert bruk av Sentinel-1 til å detektere nye bygg eller riving av bygg i Ruhr-området. Metoden bruker bare 3 bilder fra hhv 2015, 2017 og 2020, men finner delvis overenstemmelse sammenlignet med kartdata.

Hlaváčová et al. (2022) beskriver en automatisk metode for å detektere nybygg basert på Sentinel-1. Metoden har likheter med vår metode siden den beregner detekterer endringer i midlede datastacker for enkelt-geometrier. Petrushevsky et al. (2022) har også en lignende metodikk med 1-2 måneders datastacker for å detektere endringer.

Aimaiti et al. (2022) demonstrer hvordan Sentinel-1 og Sentinel-2 kan brukes til å vurdere skader på bygninger i Ukraina etter krigen. Metoden som brukes på SAR er relativt lik det som gjøres i denne rapporten, der man forventer redusert tilbakespredning når bygg har blitt bombet. De

finner at Sentinel-1 er bedre enn Sentinel-2 til å detektere endringer (skader) på små og mellomstore bygninger. Ferrentino et al. (2016) har brukt Sentinel-1 for å detektere skadde bygninger etter jordskjelv, mens Meneses og Bianco (2022) har brukt samme sensor for skadeanalyse etter en tyfon.

Interferometri kan også brukes til å overvåke byggeaktivitet (Letsios et al., 2023).

3. Preprosessering

I dette kapitlet beskrives pre-prosessering av Sentinel-1 data.

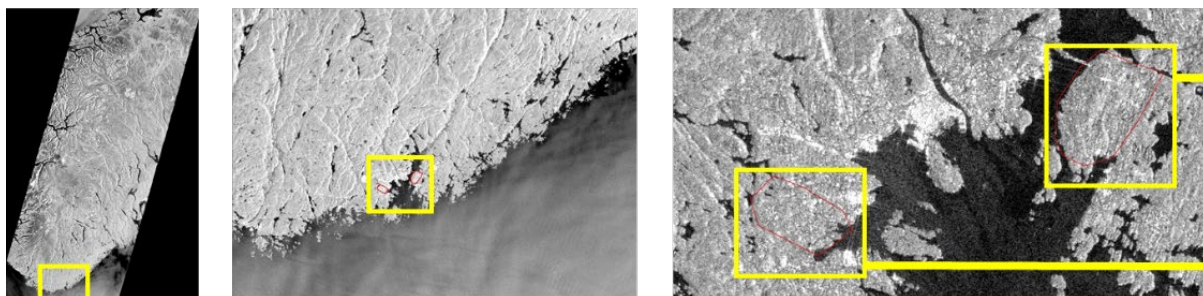
3.1. Sentinel-1-data

NORCE har lastet ned alle Sentinel-1 scener for Kristiansand-området som GRD-filer i Sentinel sitt SAFE-format til disk. Til sammen er det 1672 scener som dekker hele eller deler av området. Disse fordeler seg på 6 geometrier (se tabell 1 under) men for det østlige området er det bare 4 geometrier som dekker. Den temporale dekninga er heller ikke helt uniform. Sentinel-1B ble skutt opp i starten av 2016 slik at full kapasitet ble oppnådd omkring midten av året. I tillegg har det i perioder vært litt utfall av data. Uansett anser vi datamengden som tilfredsstillende til å kunne gjennomføre prosjektet.

For å forenkle prosesseringssteg i senere deler av prosjektet delte vi området i to testområder (E og V) som gir bilder av relativt liten størrelse. Dette har mange fordeler ved senere prosessering. De to områdene angis som AOI (area of interest) under i koordinatsystemet WGS-84 UTMz33N:

- AOI_V= [83 880,86 330,6 463 000,6 465 000] -245x200 piksel
- AOI_E= [91 220, 93850, 6 465 100,6 468 000] - 263x290 piksel

Størrelsen på områdene er av størrelsesorden 250x250 piksler slik at en datakube på typisk ~1400 bilder blir moderat (~650 Mbyte) og enkelt kan håndteres i minnet på regnemaskiner.



Figur 4. Inndeling av studieområdet i to utsnitt (venstre: S1 scene over Sør-Norge, midten: utsnitt over Agder, høyre: Kristiansand med de to testområdene).

I tabell 1 vises de ulike satellittgeometrier som dekker de to områdene. Banenumrene svarer til de faste banene satellittene beveger seg i og gir dermed faste innfallsvinkler fra henholdsvis østlig (stidende bane) og vestlig (synkende bane) siktretning¹. Vi ser at område V er dekt av 6 geometrier, mens E bare har 4. De to tilleggsgemetriene i V dekker imidlertid ikke hele området eller har relativt få scener, så vi har valgt å kutte disse ut i det videre.

¹ Satellittbanene karakteriseres som stigende (engelsk: *ascending orbit*) dersom satellitten beveger går mot høyere breddegrader og synkende (*descending*) dersom den går i retning av lavere breddegrader.

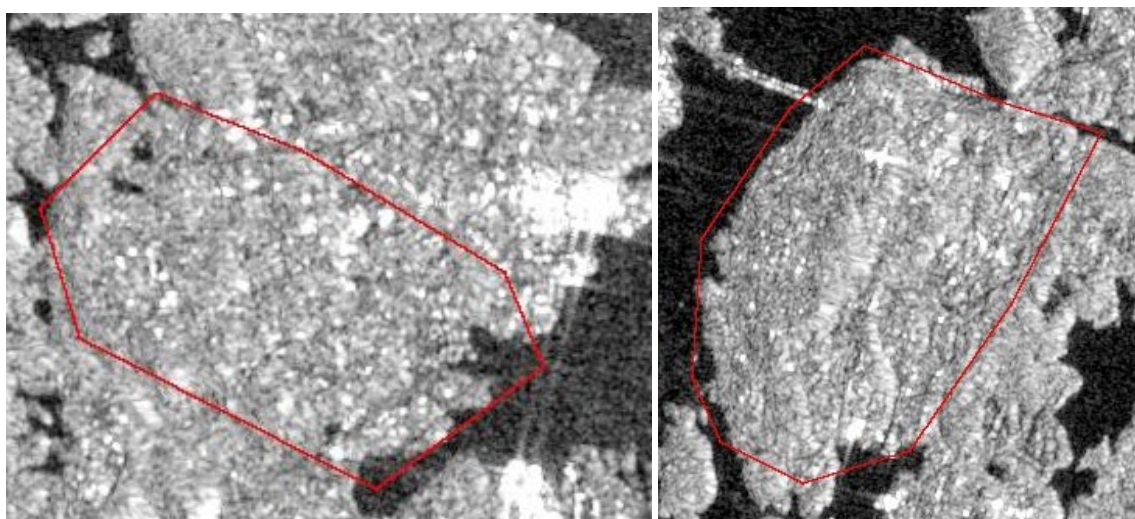
Geom/Year	2016	2017	2018	2019	2020	2021	Totalt	Innfallsvinkel Vågsbygd/Søm
015A (V)	29	45	33	23	48	53	223	45.9°/-
037D	28	59	56	56	60	55	316	38.2°/38.0°
044A	24	57	57	53	59	57	309	32.6°/31.6°
110D (V)*	25	60	59	56	61	55	318	-/-
117D	2	41	34	29	29	45	184*	39.5°/39.3°
139D	28	57	56	57	61	53	314	44.9°/44.7°
<i>Totalt</i>	<i>136</i>	<i>319</i>	<i>295</i>	<i>274</i>	<i>318</i>	<i>318</i>	<i>1672</i>	

Tabell 1: Oversikt over antall satellittgeometrier som dekker de to områdene per år. D angir synkende satellittbane (descending orbit) og A angir stigende satellittbane (ascending orbit), som svarer til observasjoner fra henholdsvis vest og øst siden radaren observerer sideveis.

3.2. Geokodet tilbakespredning

NORCE bruker egenutviklet programvare, GDAR, til geokoding av Sentinel-1 GRD-filene. Programmet geoposisjonerer hver piksel i GRD-fila til et raster som er projisert i en ønsket kartprojeksjon. I tillegg korrigerer programmet for terrenget ved å bruke en høydemodell (Kartverkets 10 m terrenngmodell). Dette er viktig, siden sideveis-seende radarsatellitter ellers vil forvrenge og feilplassere data i kartet. Endelig kalibreres data til tilbakespredning (dB-skala) i henhold til oppslagstabeller som følger med i produktet.

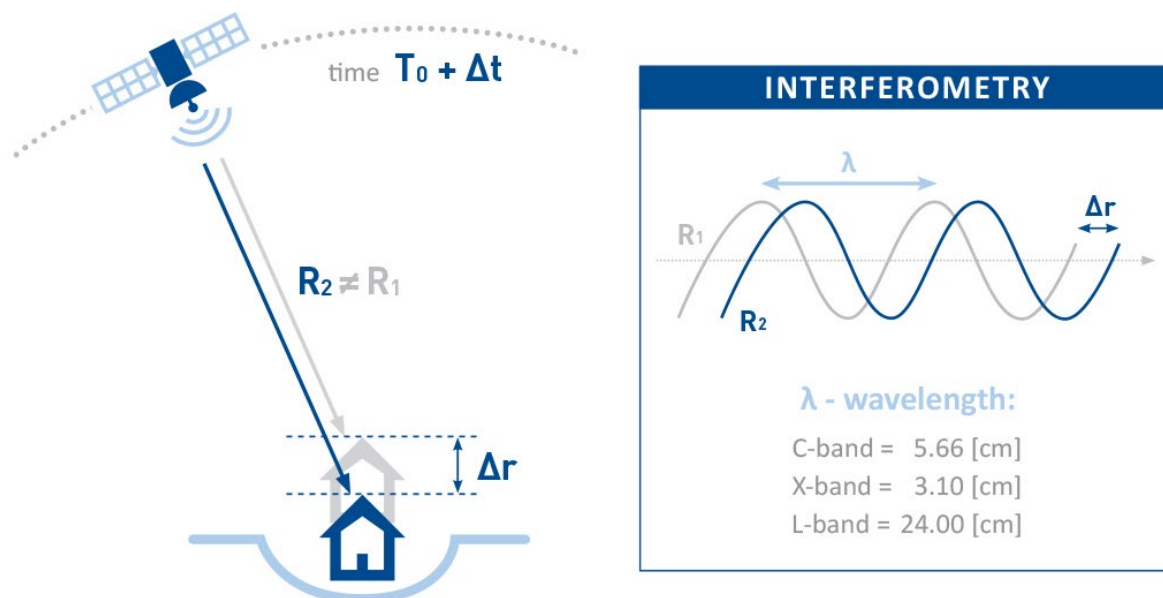
Geokoding gir som output et geotiff-bilde per satellittscene. Disse geokodete bildene angir tilbakespredning i dB per piksel. Som man kan se i figur 5 under har disse bildene temmelig mye spekkelstøy (interferens) og det er ikke enkelt å identifisere enkeltbygg. For å kunne identifisere enkeltbygg og eventuelle bygningsendringer er man avhengig av å midle for å bli kvitt noe av støyen. Dette vil bli undersøkt i avsnitt 4.9 For hver scene leverer Sentinel-1 bilder i hhv VV-polarisasjon og VH-polarisasjon. Begge disse tidsrekkene tar vi vare på og kan brukes til endringsdeteksjon enkeltvis eller i sammen.



Figur 5. De to studieområdene (øst og vest) avbildet i VV-polarisasjon.

3.3. Koherens

Koherens er et mål for stabilitet i interferometrisk prosessering. Koherens er et statistisk mål for fase-stabilitet. Range for koherens er (0,1). Interferometrisk fase beregnes fra bildepar i dataformatet single look complex (SLC) – som oftest fra to etterfølgende bilder i tid. Dersom avstanden til målet på bakken endres over tid, vil den interferometriske fasen $\Delta\phi \sim \Delta r/\lambda$ endre seg (se figur 6).



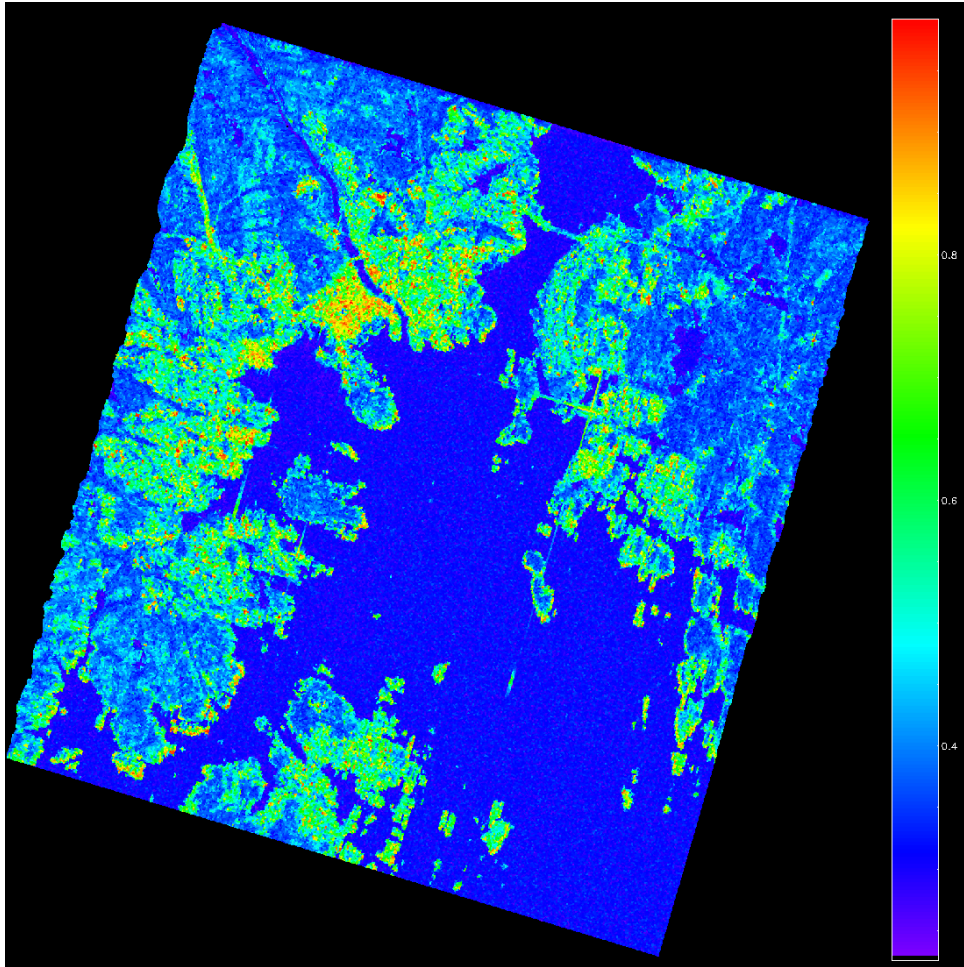
Figur 6. Interferometrisk måling av nedsynking

Den komplekse korrelasjonskoeffisienten γ mellom to komplekse SAR-bilder S_1 og S_2 representerer den normaliserte korrelasjonen mellom S_1 og S_2 :

$$\gamma = |\gamma| e^{i\phi} = \frac{E \{S_1 S_2^*\}}{\sqrt{E \{|S_1|^2\} E \{|S_2|^2\}}}$$

der $E\{\cdot\}$ representerer den matematiske forventningsoperatoren og $*$ indikerer det komplekse konjugatet. Vanligvis tolkes den komplekse fasen ved hjelp av amplituden $|\gamma|$, eller koherens, og fasen ϕ som tilsvarer den interferometriske fasen. Mens fasen stort sett er relatert til de geometriske aspektene ved scenen, beskrives koherensen vanligvis som kombinasjonen av multiplikativt sett med dekorrelerende faktorer.

For Kristiansand har vi estimert koherensen for en satellittgeometri (037 DES). Koherensen kan midles over lengre eller kortere perioder. Som oftest vil det være mindre støy desto lengre perioder man midler, og i vårt tilfelle har vi midlet 8×2 pixler (asimut x range) over henholdsvis hele år og per bildepar. Vi har geokodet koherensbilder til samme utsnitt og oppløsning som tilbakespredningsbildene. Figur 7 viser koherensbildet over Kristiansand, og man ser at de bebygde områdene har generelt høyere koherens en ubebygde områder.



Figur 7. Midlet koherens for hele studieområdet 2019-2020.

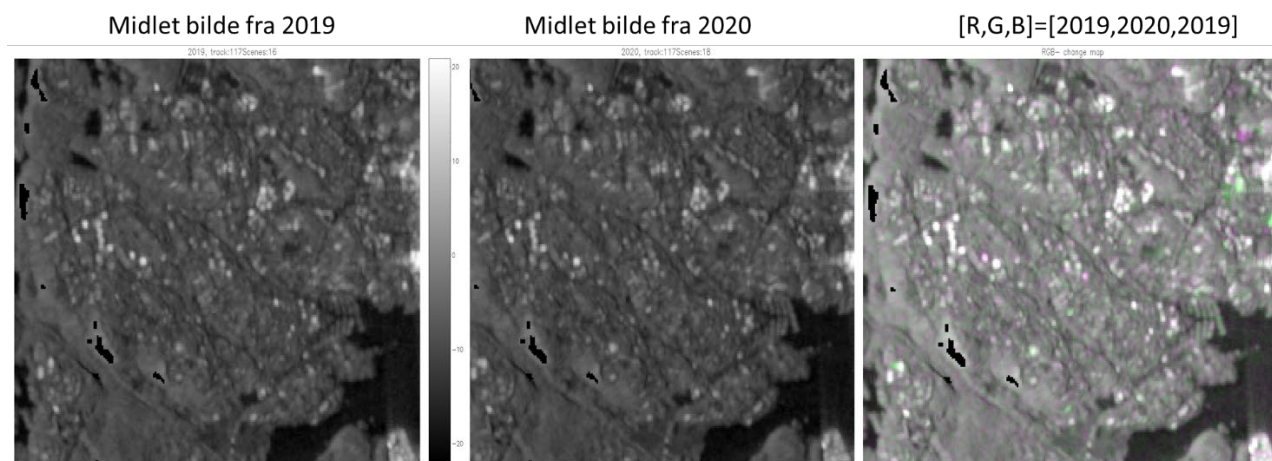
4. Endringsdeteksjon

Endringer i SAR-bilder kan skyldes en rekke ulike faktorer som naturlige svingninger på grunn av temperaturendringer i bakken, snø (våt og tørr), skog med og uten blader og hogstflater, jordbruksarealer med pløying og plantevekst, varierende jordfuktighet samt en rekke andre faktorer. Konstruksjonsarbeid av boliger vil også påvirke radardata og generelt vil man forvente økt radar-tilbakespredning fra bygg sammenlignet med uberørt natur (når man sammenligner før- og etterbilder). I vår studie har vi sett på bydeler i Kristiansand der mange av faktorene ikke er veldig dominerende. Det er f.eks. få dager med snø i byen per år, og disse unntakene blir i stor grad midlet bort når vi midler i størrelsesorden 300 bilder per år. Noen endringer som hogst, forandringer i sandtak, biltrafikk mm kan imidlertid dukke opp og skape false alarmer (deteksjoner som framkommer som bygningsendringer uten å være det i virkeligheten).

NORCE har i hovedsak testet ut to hoved-metoder for endringsdeteksjon: - En tradisjonell metode basert på terskling av endringer mellom to midlede perioder (årlige gjennomsnitt) og en metode som er basert på moderne maskinlæring (kunstig intelligens) der metodene trenes opp på kvalitetssikrede treningsdata.

4.1. Endring i tilbakespredning (enkeltgeometrier)

For å finne signifikante endringer i tilbakespredning har vi først midlet lengre perioder for de ulike geometriene (4 stk), polarisasjonene (VV og VH) og testområdene (E/V). I utgangspunktet midlet vi hele år med data (2016-2021) slik at vi kan finne signifikante endringer ved terskelmetoder. I figuren under vises område V der vi ser på middelbildet for 2019 og 2020. Ved å sammenstille disse bildene i et 3 kanals RGB-bilde ser vi tydelig hvilke områder der det er endringer som lilla (lavere tilbakespredning) og grønn (økt tilbakespredning).



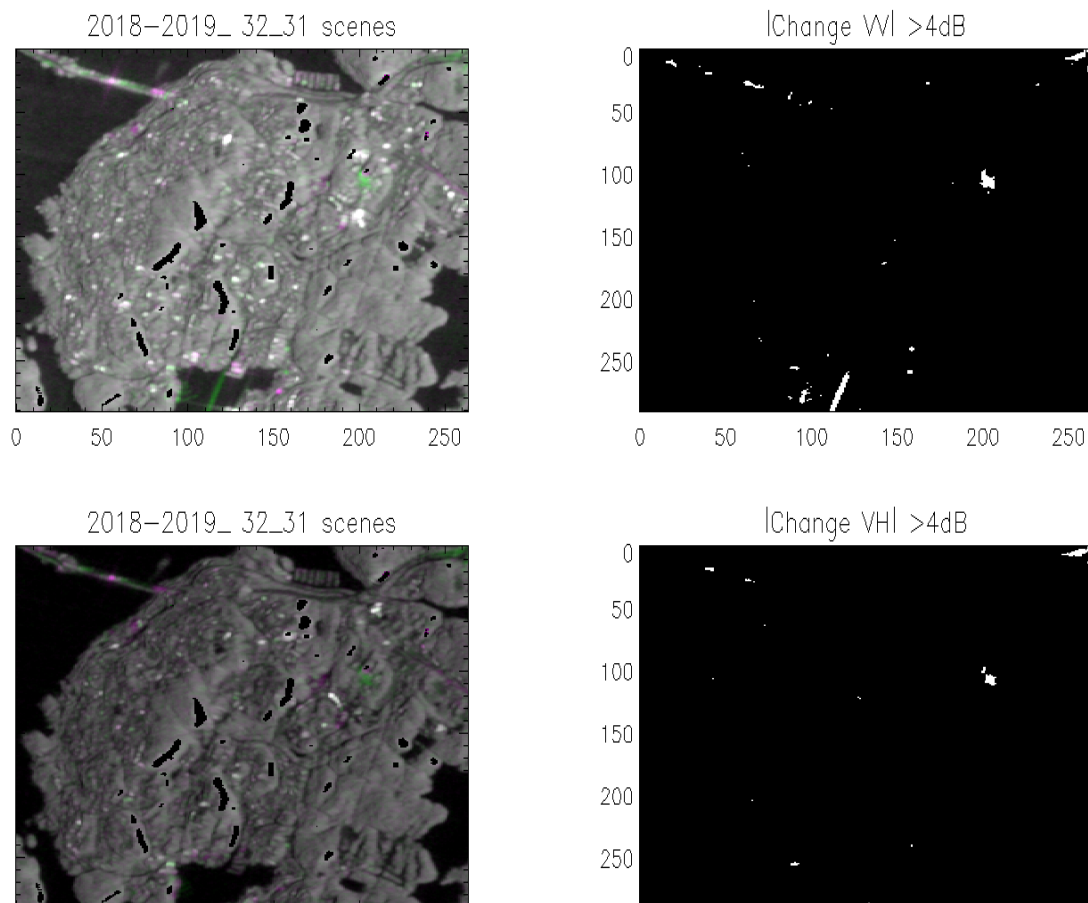
Figur 8. Endringsdeteksjon på årlige middelbilder.

Ved å terskle disse endringene finner vi binære områder der endringen er signifikant. I vårt tilfelle bruker vi kriteriet

$$|\sigma^{y+1}(x,y) - \sigma^y(x,y)| > \text{Thr, der Thr}=4 \text{ dB}$$

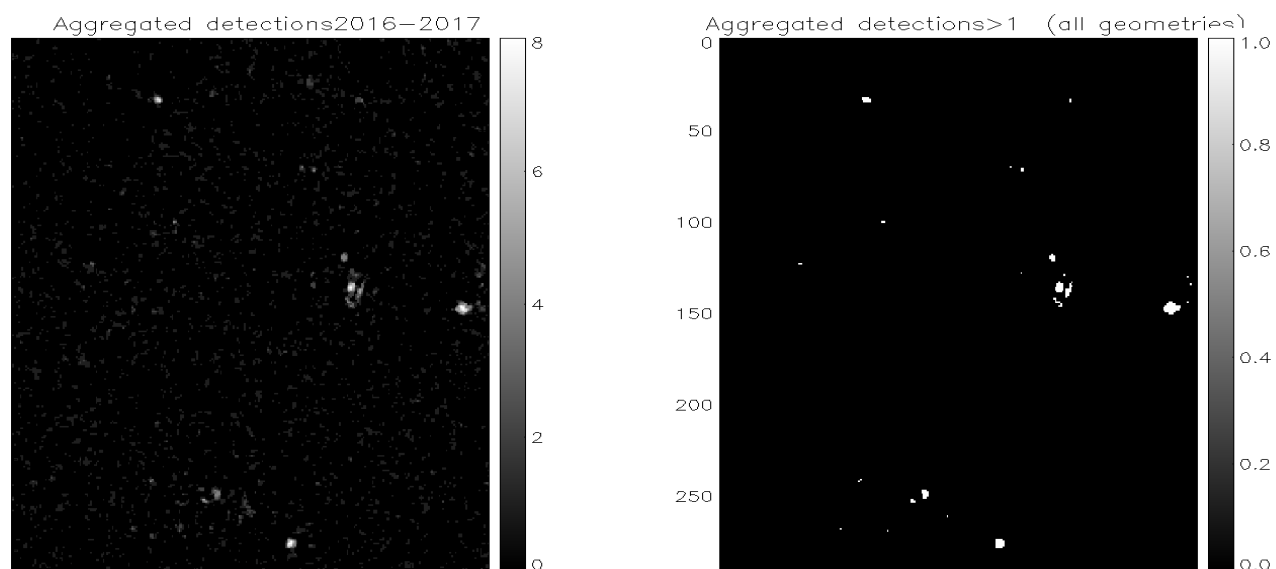
Der er $\sigma^y(x,y)$ tilbakespredning for pikselen (x,y) for året y , og Thr er terskelen. I de fleste tilfeller har vi valgt å bruke terskler i området 3-4 dB for å filtrere bort svake endringer.

I figuren under vises endringsdeteksjon for område E mellom 2018 og 2019. Her vises også likheter og forskjeller mellom de to polarisasjonene for to ulike år. De er tydelig at VV har flere detekterte endringer enn VH, men i noen tilfeller (f eks striper på havet) kan det være gunstig å bruke VH da det muligens gir færre feildeteksjoner.



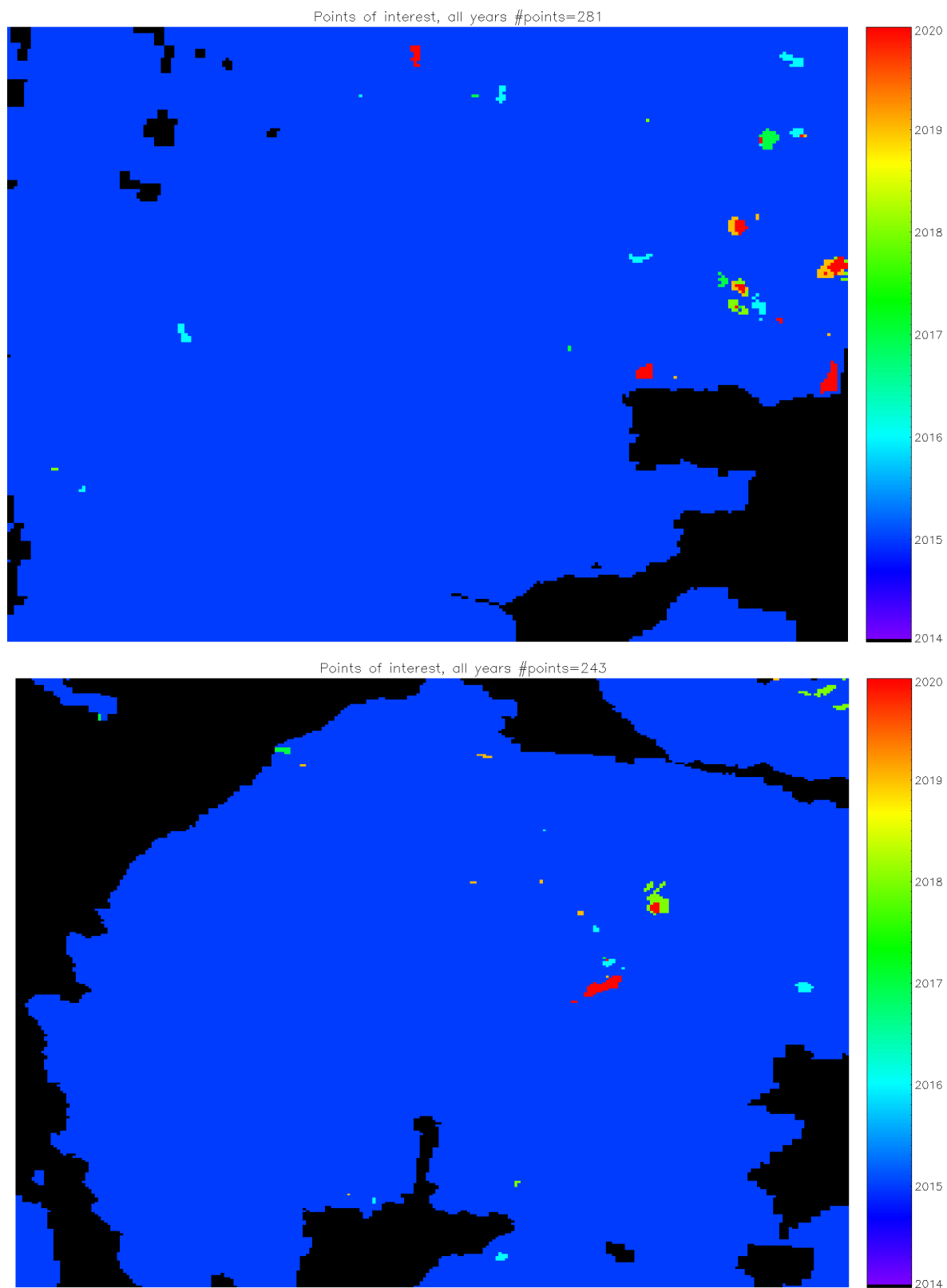
Figur 9. Endringsdeteksjon på årlige middelbilder. Øverst vises RGB og binær endringsdeteksjon for VV polarisasjonen. Nederst vises VH polarisasjon. I stor grad er endingene sammenfallende, men det er også forskjeller.

Siden vi har 4 ulike geometrier og 2 polarisasjoner for begge områdene, og endringene til en viss grad er forskjellig fra geometri til geometri og polarisasjon til polarisasjon har vi valgt å kombinere dette til en felles endringsdeteksjon ved å summere alle endringene. Altså, områder der det er en endring i ett av de 8 endringsbildene får verdi 1 mens områder uten endring har verdi 0. Når alle 8 bilder summeres ender vi med et aggregert endringsbilde som maksimalt kan ha verdien 8. Dette bildet kan deretter terskles. Vi har i de fleste tilfeller valgt verdien 2 som terskel, slik at ikke en enkeltendring skal være nok til å trigge en deteksjon.



Figur 10. Aggregert endringsdeteksjon (venstre). Tersklet aggregert endring (høyre). I bildet til venstre sees en del falske alarmer som bare detekteres i en av geometriene. Disse forsvinner i det tersklede bildet til høyre.

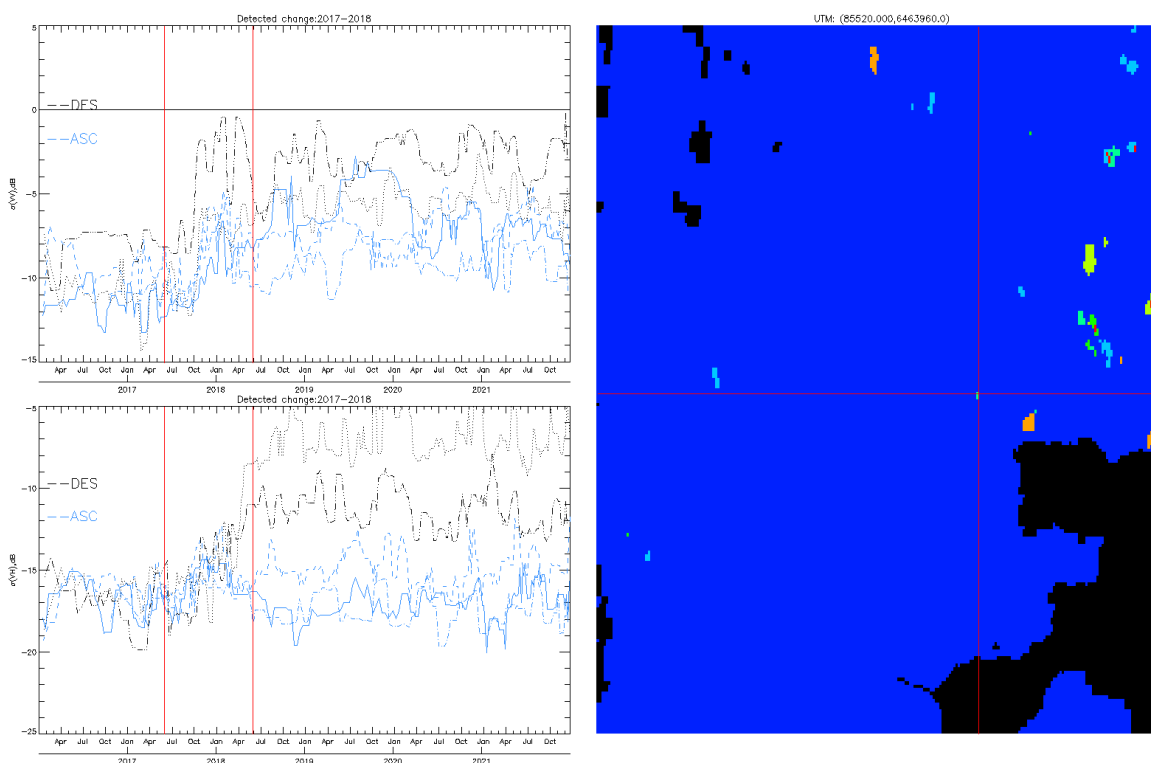
Som et sluttprodukt i denne seksjonen kan man lage endringskart som oppsummerer endringene for alle årene ved å kode inn tidspunkt for endringen. I dette tilfellet er det hensiktsmessig å velge midten av tidsintervallet (1.7) man midler over som tidspunkt. Output bli da kart som vist i figur 7.



Figur 11. Tidspunkt for endring (topp område V, bunn område E). Hav er kodet med sort og land uten endring i marineblått.

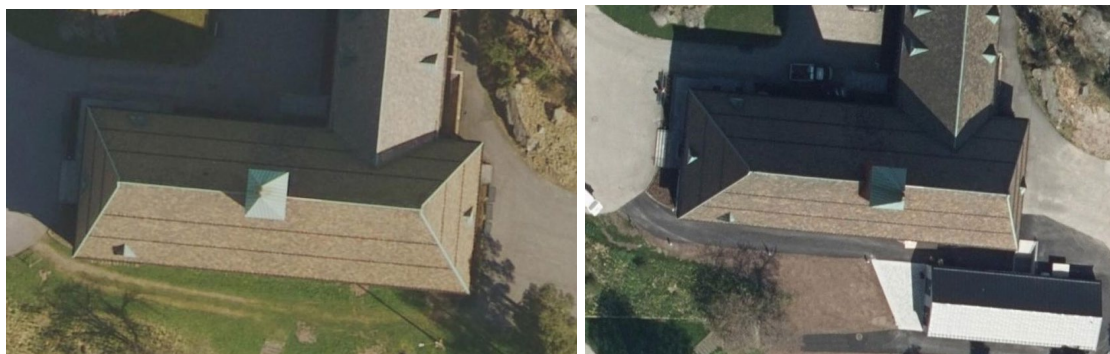
4.2. Temporale endringer

Ved å studere enkelt piksler der man finner signifikante endringer fra ett år til det neste, er det mulig å finne mer eksakte tidspunkter for når endringene inntreffer ved å gå tilbake til de opprinnelige SAR bildene og plote tilbakespredning som funksjon av tid for disse enkeltpikslene. I figur 12 sees tilbakespredning for alle geometrier og alle polarisasjoner for en piksel der vi fant tydelig endring mellom 2017 og 2018 (se rødt kryss i bildet til høyre). I tidsrekkene ser vi at det er en kraftig økning i tilbakespredning for de fleste geometriene mellom sommeren 2017 og sommeren 2018, økningen er noe varierende, og sannsynligvis sterkest for de synkende satellittbanene. Dette skyldes sannsynligvis at bygningens orientering eller takvinkel er mer fordelaktig for å se endringer i denne geometrien.



Figur 12. Temporal endring i tilbakespredning for en piksel (i det røde krysset i bildet til høyre).

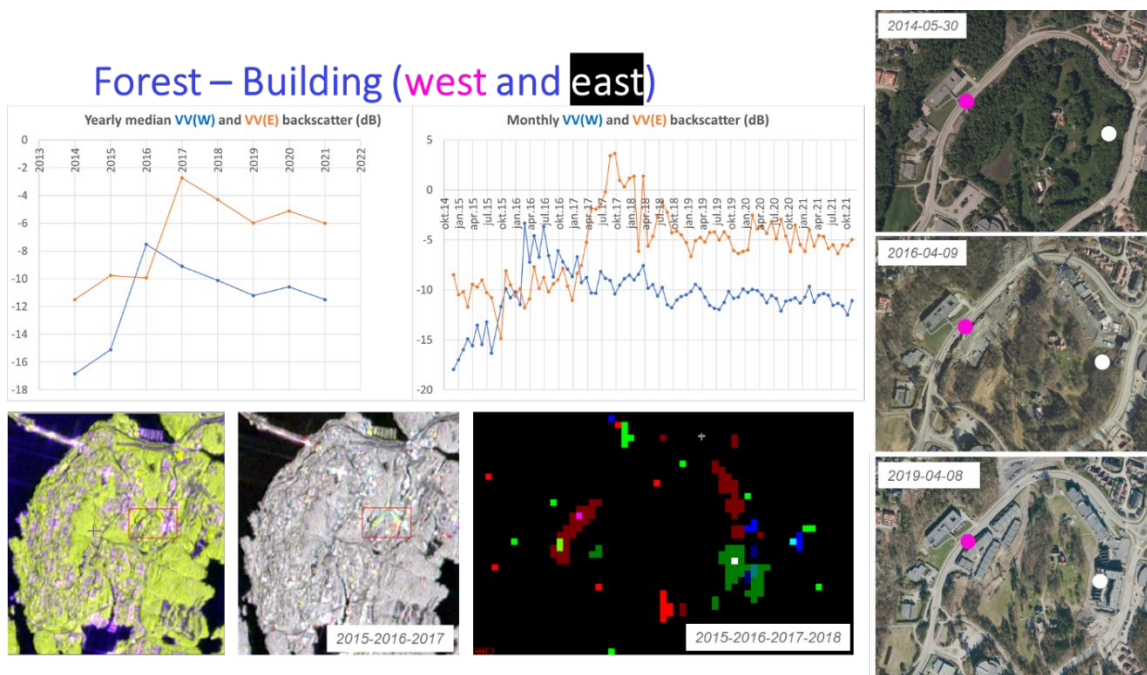
For det aktuelle bygget fant vi at det var kommet et tilbygg (eller annek) i denne perioden (se fig 13).



Figur 13. Utsnitt av flyfoto fra «Norge i bilder» for aktuell villa fra figur 12 for 8.4.2017 og 8.5.2018. Tilbygget i nedre høyre hjørne er mest sannsynlig årsaken til endringene i tilbakespredning.

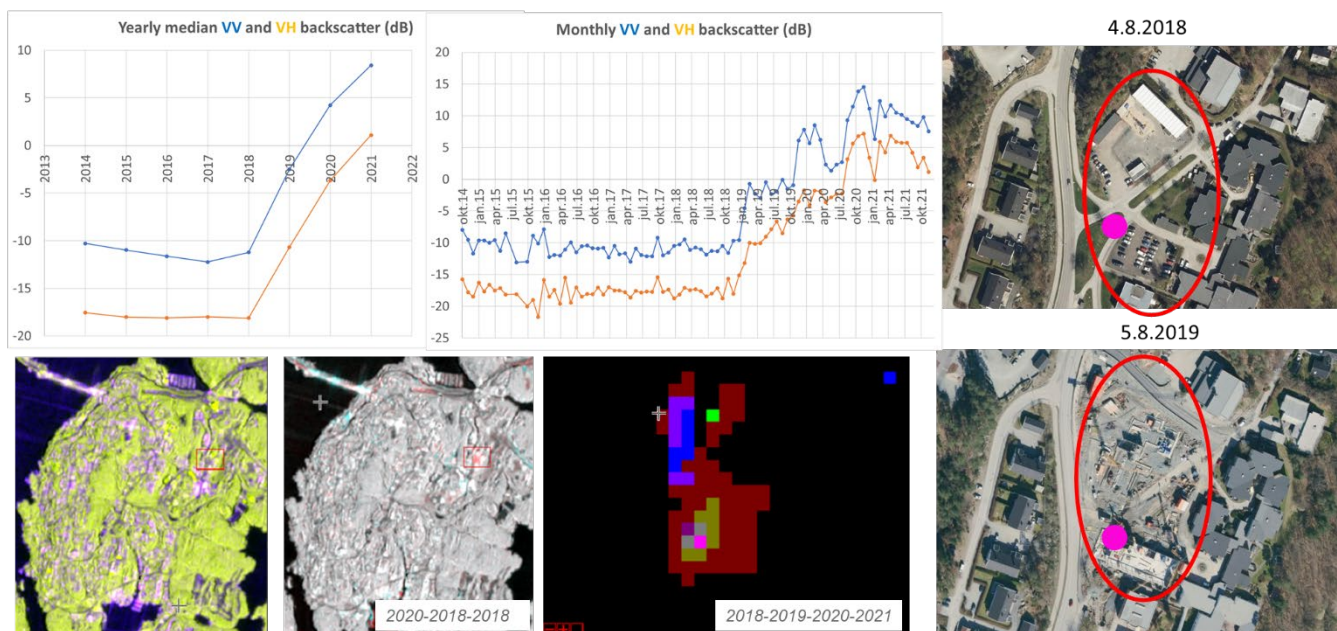
4.3. Endring i tilbakespredning (innfallsvinkelkorrigerte bilder)

Ved å korrigere tilbakespredning for innfallsvinkelen (hovedsakelig ved å dividere tilbakespredninga σ^0 med $\sin(\theta)$, der θ er innfallsvinkelen), oppnår man en første ordens korleksjon av de ulike tidsrekkene. I figuren under vises årlige og månedlige middelerdiene av samtlige geometrier (synkende og stigende satellittbaner) for en piksel der vi observerer endring. I dette tilfellet er det et skogsområde der det kommer opp bygninger (rekkehus/småblokker) på to forskjellige tidspunkter. Fordelen av en slik prosessering er at alt prosesseres i ett og at man trenger ikke å sammenligne resultatet fra forskjellige satellitt geometrier, men ulempen er at midlingen kan minke deteksjonsmuligheten fra et sterkt forandringssignal i en spesiell geometri.



Figur 14. Endringsdeteksjon med innfallsvinkel-korrigert tilbakespredning.

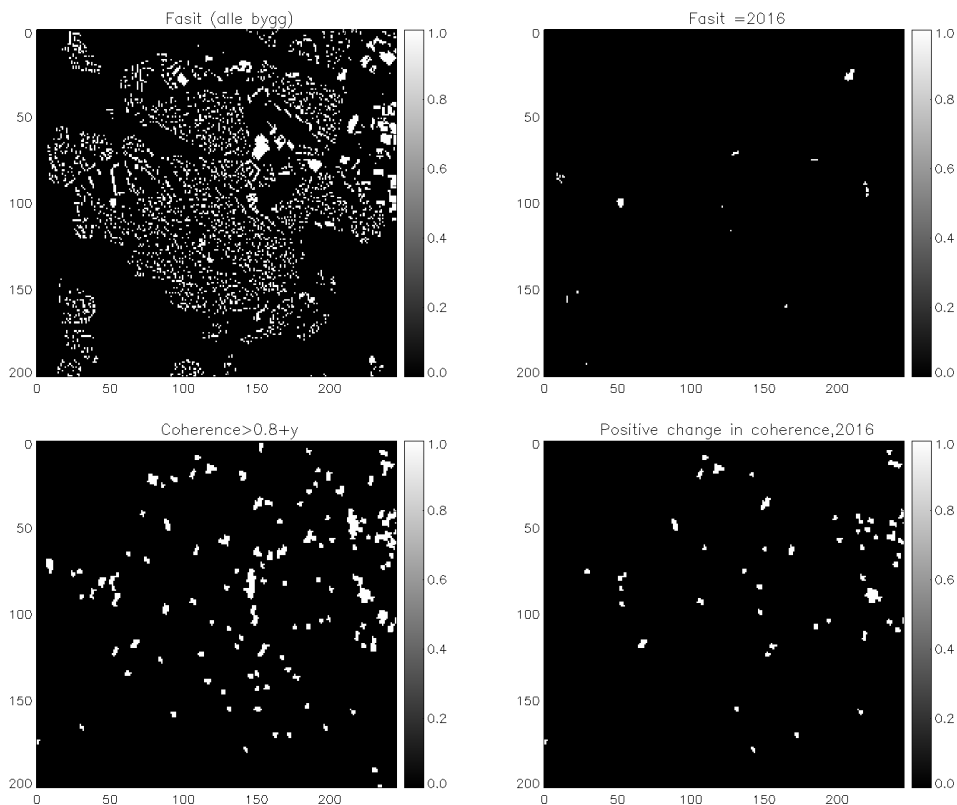
Revet i 2019 – byggeplass (93235E, 9466916N)



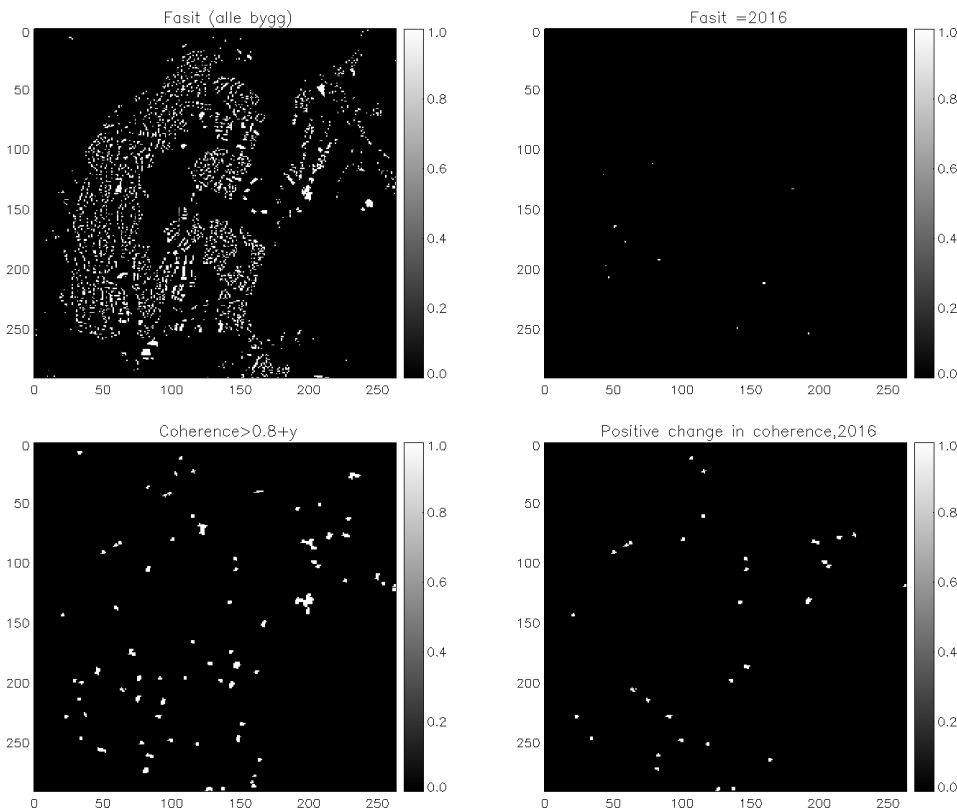
Figur 15. Endringsdeteksjon med tilbakespredning korrigert for innfallsvinkel.

4.4. Endring i koherens (beregnet i én geometri)

Endringer i koherens kan studeres på tilnærmet samme måte som vi tidligere har studert endringer i tilbakespredning. Vi har tatt utgangspunkt i midlet årlig koherens og prøvd å detektere tilfeller der et område endres fra lav koherens til høy koherens. I vårt tilfelle har vi satt en forholdsvis høy terskel på 0.8. Hypotesen er da at hvis koherensen endres fra å være lavere enn terskel et år til høyere enn terskel neste år, så indikerer dette et nybygg. Resultatene i figur 16 og 17 viser en del deteksjoner for begge testområdene.



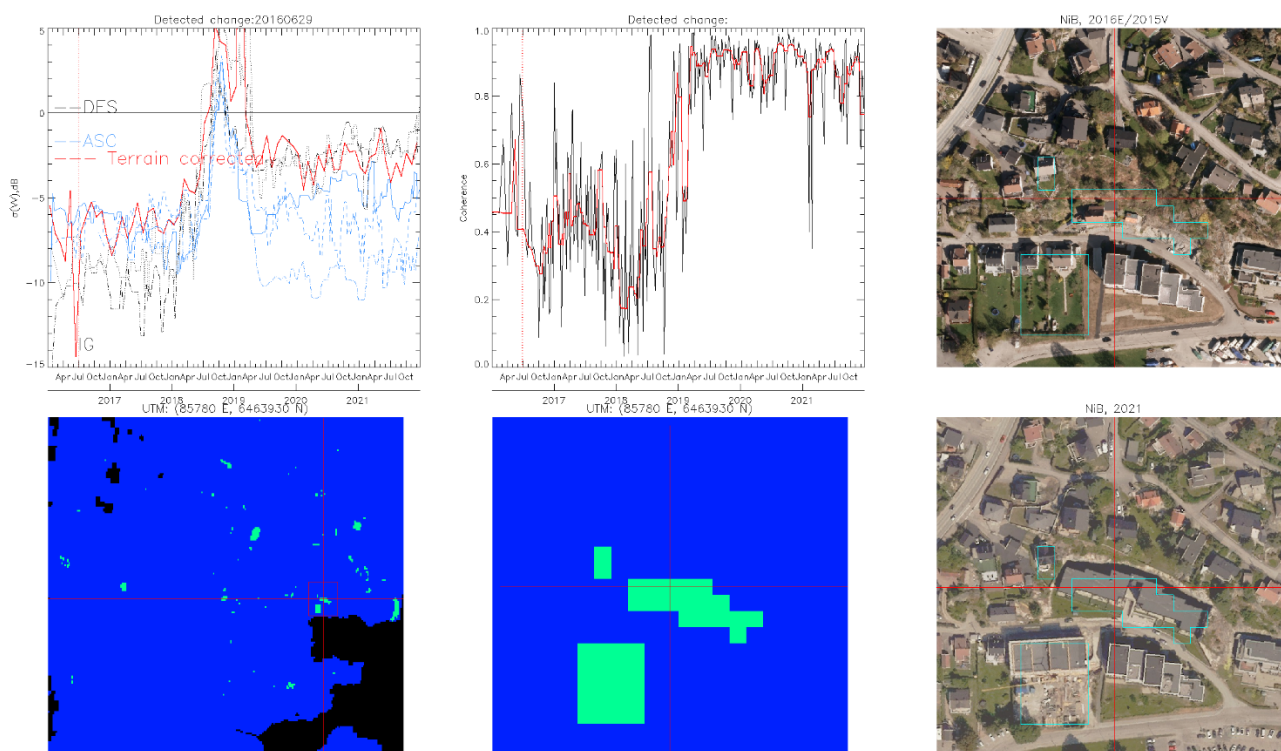
Figur 16. Endringsdeteksjon basert på endring i koherens (testområde vest)



Figur 17. Endringsdeteksjon basert på endring i koherens (testområde øst)

4.5. Kombinert endringsdeteksjon (koherens og tilbakespredning)

For å studere kombinert endringsdeteksjon ved bruk av både tilbakespredning og koherens har vi sett på alle enkeltspredere der vi har signifikant endring i tilbakespredning. I mange av disse kan vi også se at koherensen endres. I figur 18 vises et eksempel der et bygg kommer opp en gang mellom 2016 og 2021 (jf. Flyfoto fra NorgeiBilder). I eksemplet er også tidspunkt for igangsettelse (IG) inntegnet til 2016. Vi ser imidlertid at de mest markante endringene forekommer fra ca. juli 2018. Dette er tydelig både i tilbakespredning (øverst til venstre) og koherens (øverst midten).



Figur 18. Endringsdeteksjon basert på endring i koherens (Vågsbygd).

4.6. Metoder for endringsdeteksjon basert på maskinlæring

I dette avsnittet beskriver vi problemet i et maskinlæringsperspektiv. Vi innfører terminologi fra fagfeltet maskinlæring som vil være relevant for litteratursøk og videre utvikling av metodikken, og forklarer vårt forslag til løsningsmetode. Vi diskuterer også de særskilte utfordringene med den konkrete problemstillinga som krever prosjektspesifikke løsninger. Resultatene gjenspeiler at prosjektets rammer ikke har gitt rom for omfattende eksperimentering med ulike kjente metoder eller utvikling av nye algoritmer som er tilpasset datasettet. Vi har i stor grad vært nødt til å benytte eksisterende standardmetoder og beskriver hvilke valg vi har gjort ut fra dette.

Endringsdeteksjonen i prosjektet kan som maskinlæringsoppgave defineres som binær klassifisering av hele tidsrekker i klassene “endring” og “ikke endring”. Klassifisering er en oppgave som tilhører

området *styrt læring*², ettersom treningsdatasettet inneholder både inputdata og tilhørende *merkelapp*³ (eller *klassemerking*) som angir klassetilhørighet. Inputdata består av ei tidsrekke for hver piksel, mens merkelappene er informasjon på pikselnivå om hvorvidt det har foregått endring eller ikke innafør tidsrommet som tidsrekka representerer. Merkelappene omtales innafør jordobservasjon også som *bakkesannhet*⁴.

4.6.1. Inndata til maskinlæringsalgoritmene

Inndata består av flervariabel tidsrekker som inneholder ulike fjernmålte variabler og eventuelle tilleggsdata. De fjernmålte variablene er en kombinasjon av tilbakespredning (kalibrert radarintensitet) i ulike polarisjoner, ettersom Sentinel-1 gjør samtidige målinger i VV-kanalen og VH-kanalen⁵, og interferometrisk koherens. Tilleggsdata kan for eksempel være indekser eller indikatorer som angir hvilken type inputdata som er registrert for et gitt tidspunkt. Dette henger sammen med at vi i inputdata har heterogene målinger, og må ta standpunkt til om vi skal gjøre endringsdeteksjonen ved å ta ei beslutning på et *heterogent datasett* eller ved å slå sammen flere beslutninger på oppsplitta, homogene datasett. Tidsrekkene er heterogene fordi målingene består av både tilbakespredning og koherens, som er målinger med ulikt fysisk opphav, og fordi målingene er gjort med ulike satellittgeometrier, som betyr at satellitten måler fra ulike vinkler og dermed potensielt ser ulike aspekter av bakkemålet.

La oss forklare de ulike tilnærmingene med følgende eksempel: Dersom vi har måleserier for ulike geometrier, kan vi velge å skille dem ut i ulike tidsrekker – ei tidsrekke per geometri – slik at målingene i samme tidsrekke er lett sammenliknbare. Deretter kan vi gjøre endringsdeteksjon på tidsrekkene hver for seg, ut fra en antakelse av at noen endringer kan være lettere å oppdage med en spesifikk geometri. Den endelige avgjørelsen får vi ved å kombinere de geometri-spesifikke deteksjonene etter en valgt regel gjennom såkalt *beslutningsfusjon*⁶. Alternativt kan vi blande målinger fra ulike geometrier i samme tidsrekke, men legge til en ekstra variabel i tidsrekka som indikerer hvilken geometri målinga stammer fra. Denne indikatoren vil være det vi ovenfor refererte til som tilleggsdata. Den gjør at maskinlæringsalgoritmen potensielt kan lære seg å behandle informasjonen fra ulike geometrier ulikt og ta hensyn til dette i beslutningsprosessen. Da utfører vi *egenskapsfusjon*⁷ i stedet for beslutningsfusjon. Dette skiller seg fra klassisk mønstergjenkjenning, der en typisk vil holde datasettet homogent for at hypotesetestene i deteksjonsalgoritmen (beslutningene) skal være gjennomsiktlige og tolkbare, for eksempel gjennom terskeltester lik den vi har brukt i den *tradisjonelle metoden* vi har implementert. Ulempen med egenskapsfusjon og å gjøre deteksjonen på heterogene data er at det krever mer treningsdata for å lære å kombinere informasjonen, sammenlikna med å splitte data eksplisitt. Til

² Engelsk: *supervised learning*; Motsatsen til styrt læring er *ikke-styrt læring*, som betyr at inputdata ikke har tilhørende merkelapper eller klassemerking, på engelsk kalt *unsupervised learning*.

³ Engelsk: *label*

⁴ Engelsk: *ground truth*

⁵ Sentinel-1 sender ut mikrobølger som er vertikalt polarisert og måler både den vertikale og horisontale komponenten av de tilbakespredte bølgeene. Disse to målingene refereres til som VV- og VH-kanalen.

⁶ Engelsk: *decision fusion*; Eksempel på regel i beslutningsfusjon: Dersom vi har n ulike satellittgeometrier, kan vi teste hver geometri hver for seg og bruke en beslutningsregel som sier at pikselen er endret dersom vi detekterer endring i m av n geometrier.

⁷ Engelsk: *feature fusion*; Fagfeltet datafusjon skiller mellom egenskapsfusjon og beslutningsfusjon ut fra hvilket nivå i prosesseringskjeden en slår sammen data på.

gjengjeld kan metoden oppnå bedre ytelse ved å lære fra data hvordan de kombineres optimalt framfor å bruke en forhåndsdefinert fusjonsregel for beslutningene.

På samme måte kan vi i prinsippet velge å gjøre endringsdeteksjon på tidsrekker som inneholder både tilbakespredning og koherens, ved å legge til en tilleggsvariabel som indikerer hvilken type måling datapunktet i tidsrekka representerer. Den alternative metoden er å holde tidsrekkeene av tilbakespredning og koherens hver for seg, utføre deteksjonen separat på hver tidsrekke, og til slutt slå sammen deteksjonsresultatene ved hjelp av en beslutningsregel.

Vi må altså bestemme hvilke variabler som skal inngå i tidsrekkeene som utgjør inndata. Vi må også bestemme oss for hvilken tidsoppløsning de skal ha og hvilket tidsrom de skal dekke. Vi har valgt å dele måleseriene opp i årlige tidsrekker som representerer kalenderår. Dette er gjort ut fra tanken om at endringskartene kan oppdateres på årlig basis ved at en kjører deteksjonsalgoritmen på ei tidsrekke som representerer satellittmålinger fra siste år. Her kan en velge å la tidsrekka bestå av samtlige målinger som er gjort. Dette har visse konsekvenser:

- Tidsrekkeene vil få ulik lengde for hvert år, ettersom antall satellittoverflygninger og dataopptak vil variere. Det begrenser utvalget over anvendbare metoder for tidsrekkeklassifisering til de algoritmene som kan trenes og kjøres på tidsrekker med varierende lengde.
- Tidsrekkeene vil potensielt inneholde mye data, noe som kan ha konsekvenser for hvilke algoritmer som lar seg trene opp, siden komplette tidsrekker vil kreve mye beregningskraft.
- Tidsrekkeene vil ha høyt støynivå, siden radarmålinger er påvirket av såkalt spekkel, et interferensfenomen som gir en støyliknende variasjon i signalet. Dette kompenseres mest effektivt gjennom å midle målinger i tid eller rom, kjent som spekkelfiltrering.

Et attraktivt alternativ er å midle målingene til månedlige verdier av hver variabel. Fordelene er at dette reduserer datamengden betraktelig, tidsrekkeene få en fast lengde på 12 datapunkter og de blir samlet uniformt i tid. I tillegg er en måned et såpass langt intervall at det med stor sannsynlighet finnes minst et datapunkt, slik at tidsrekkeene ikke blir manglende data i tidsrekkeene, et kjent problem fra tidsrekkeanalyse som må behandles med særskilte metoder som for eksempel imputasjon av data. Dette gjør at vi anser månedlig midling som et passende for preprosessering av tidsrekkeene.

4.6.2. Referansedata og usikre merkelapper

På lik linje med inputdata er det verdt å diskutere egenskapene til merkelappene i datasettet. Merkelappene inneholder endringsstatus som er hentet fra polygoner som definerer omrisset av bygninger hvor omsøkte byggeprosjekter eller bygningsendringer skal skje ifølge Kartverkets databaser. Disse polygonene gjøres om til *rasteriserte pikselmasker*⁸ som representerer et gitt tidsrom, hvor uendrede piksler får verdien 0 og piksler med endring får verdien 1.

⁸ Ved *rasterisering* konverterer vi polynomer, som er vektordata hvor punktene eller nodene som avgrensner polynomet kan ta alle flyttallsverdier langs koordinataksene, til pikselverdier som er lokalisert på et definert pikselgrid med diskrete posisjonsverdier. Denne konversjonen innfører unøyaktigheter.

Vi har erfart at datasettet vi bruker har *usikre merkelapper*⁹ og usikkerheten omfatter både tid og rom. Usikkerheten i tid skyldes at informasjonen om bygningsendringer i Kartverkets database ikke inneholder sikre tidspunkt for endringene. Dette skyldes delvis at innmeldte tidspunkt for igangsettelse av byggeaktivitet ikke nødvendigvis er helt nøyaktige, men også at tidspunktet for når bygningsaktivitet er målbart fra satellitt er vanskelig å fastslå. I verste fall kan usikkerheten medføre at endringstidspunktet blir satt til feil år, som vil få uheldige konsekvenser for treningen av deteksjonsalgoritmen. Usikkerheten i rom kan skyldes både unøyaktige bygningspolygoner og at rasterisering av endringspolygoner kan introdusere feil som følge av at kontinuerlig informasjon blir diskretisert, men i enda større grad at endringene ikke alltid skjer innfor bygningspolygonene, men kanskje et annet sted på samme eiendom. Denne typen merkelapper kan i litteraturen også være referert til som *svake merkelapper*¹⁰. Dette begrepet brukes om merkelapper der objektet i den angitte klassen ikke oppgis i sin helhet, men med en indikasjon av posisjon eller tidspunkt, eller gjennom merking av et eller flere datapunkter (for eksempler piksler eller tidspunkt) av et objekt som har utstrekning i rom og tid.

Det finnes ulike metoder å håndtere usikre merkelapper i litteraturen. Dette kan for eksempel være å gå bort fra å bruke binære endringsmasker som merkelapper, og i stedet innfører flytende logikk¹¹ i form av endringsmasker tar verdier mellom 0 og 1. Her vil en typisk tilordne verdier større enn null til et område utenfor endringspolygonene og til tidspunkt utenfor byggeperioden for å ta høyde for den usikre informasjonen. Dette er en metode som kunne vært forsøkt brukt på dette datasettet i en fortsettelse av prosjektet.

4.6.3. Klassifisering av tidsrekker

La oss betrakte endringsdeteksjonsoppgaven som et klassifiseringsproblem. Klassifisering behandles i maskinlæring som en to-trinns prosess, der en først gjør *egenskapsutvelgelse*¹² fra det originale inndatasettet og deretter utfører *klassifisering* av den utvalgte egenskapsvektoren (eller en annen datastruktur, for eksempel en tensor). I egenskapsutvelgelsen transformerer en inndatasettet til en representasjon av data som egner seg bedre for å utføre *kategorisk prediksjon*¹³ for det gitte formålet. Deretter vil klassifikatoren predikere en kategori eller klasse.

Tidsrekker er en egenartet type data som krever tilpassede metoder for egenskapsutvelgelse. Tidsrekker kan være lange, informasjonen kan være sterkt korrelert for nærliggende tidssteg, og det gir sjelden mening å stappe originale tidsrekker av høy dimensjon rett inn i en klassifikator. I stedet vil en typisk beregne nye egenskaper som fanger opp relevante karakteristika ved tidsrekkene og bruke disse som inndata til en klassifikator. Tradisjonelt har dette vært gjort

⁹ Engelsk: *noisy labels*; Uttrykket usikker merkelapp angir en merkelapp som er inneholder usikker informasjon. Usikkerheten er ofte så betydelig at dette må tas hensyn til og behandles med tilpassede virkemidler av algoritmen som skal bruke treningsdata.

¹⁰ Engelsk: *weak labels*

¹¹ Engelsk: *fuzzy logic*; På norsk brukes begrepene uller, lodden, uskarp eller flytende logikk for å angi logikk hvor en erstatter logiske variabler med de binære verdiene 0 eller 1 med variabler som kan ta et kontinuum av verdier mellom 0 og 1.

¹² Engelsk: *feature extraction*

¹³ Engelsk: *categorical prediction*; Kategorisk prediksjon er et annet begrep for problemer som klassifisering og segmentering, der inndata tilordnes en diskret kategori, i vårt tilfelle «endring» eller «ikke endring». Motstykket er kontinuerlig prediksjon, der en predikerer en kontinuerlig variabel, som i regresjon.

gjennom egenskapsproduksjon¹⁴, det vil si beregning av egenskaper med en formel som er forhåndsbestemt av en ekspert ut fra ei oppfatning av at den resulterende egenskapen er relevant for oppgaven. Dette kan for eksempel være frekvenser, autokorrelasjoner, eller andre variabler som karakteriserer tidsrekka. Motstykket til dette finner vi i moderne maskinlæring, der egenskapene er lært og beregna gjennom en adaptiv prosess som leter etter de egenskapene som gir best resultat for oppgaven, i vårt tilfelle binær klassifisering av tidsrekkene. I dette tilfellet tilpasses både egenskapsutvelgelsen og den kategoriske prediksjon ved såkalt gjennomgående trening¹⁵, der alle parametre i hele algoritmen tilpasses ut fra data og hva som viser seg å gi best resultat. *Dype nevrale nettverk*¹⁶ utmerker seg for dette formålet, siden de har effektive metoder for å lære fra store mengder data og oppnår oppsiktsvekkende gode resultater.

4.6.4. Kontekstuell klassifisering

Det er kjent fra klassisk mønstergjenkjenning og maskinlæring at en drar nytte av å se på naboskap av piksler når en skal klassifisere bilder. Det betyr at en kan utnytte informasjonen i både egenskapene og de predikerte merkelappene til nabopiksler når en skal klassifisere en piksel. Tilleggsinformasjonen fra romlige naboskap kalles for *kontekstuell informasjon*¹⁷ og metodene som utnytter denne gjør såkalt *kontekstuell klassifisering*. Dette ønsker vi ideelt sett å benytte oss av i klassifiseringa av tidsrekker også, men antall kontekstuelle metoder for tidsrekker er begrensa og disse krever mye treningsdata og beregningskraft. Innen dyp læring er *nevrale konvolusjonsnettverk*¹⁸ en særdeles effektiv måte å utnytte kontekstuell informasjon, mens tidsrekker ofte modelleres gjennom *rekursive nevrale nettverk*¹⁹, og særlig den populære nettverksarkitekturen kjent som *LKTM-nettverk*²⁰. For å modellere både det romlige og temporale aspektet ved datasettet må vi kombinere disse arkitekturene til et konvolusjonelt LKTM-nettverk, som kan gjøre klassifisering av tidsrekker av bilder.

¹⁴ Engelsk: *Feature engineering*; Egenskapsproduksjon skiller seg fra egenskapsutvelgelse ved at en i førstnevnte prosess beregner egenskaper ved hjelp av forhåndsdefinert prosedyre. Egenskapsutvelgelse kan utføres som egenskapsproduksjon, men kan også bruke adaptive metoder der beregninga av egenskaper for eksempel tilpasses iterativt for å oppnå best mulig resultat i den påfølgende oppgaven, som for eksempel endringsdeteksjon.

¹⁵ Engelsk: *end-to-end training*; Kjennetegnes ved at en trener hele algoritmen og alle lagene i det nevrale nettverket samtidig. Det gjør en typisk ved å tilpasse parametrene i algoritmen til den konfigurasjonen som gir best mulig sluttresultat ut fra et gitt kriterium. I vårt tilfelle betyr det at en bruker datasettet til å tilpasse både egenskapsutvelgelsen og den kategoriske prediksjonen slik at endringsdeteksjonen blir mest nøyaktig.

¹⁶ Engelsk: *Deep neural networks*; Læring gjennom kunstige nevrale nettverk med stor dybde (mange etterfølgende lag av nevroner i nettverket) er også kjent som dyp læring, eller *deep learning* på engelsk.

¹⁷ Engelsk: *Contextual information*; En klassisk metode å utnytte den kontekstuelle informasjonen som finnes i omgivelsene eller det romlige naboskapet til en piksel, er å bruke tilfeldige Markov-felt (*Markov random fields* på engelsk) til å modellere naborelasjonene og gjøre kontekstuell klassifisering (*contextual classification* på engelsk).

¹⁸ Engelsk: *Convolutional neural networks*, ofte forkorta som *conv-nets* eller *ConvNets*.

¹⁹ Engelsk: *Recurrent neural networks (RNNs)*

²⁰ Engelsk: *Long short-term memory (LSTM) network*; Kan på norsk oversettes til *langt korttidsminne-nettverk* eller *LKTM-nettverk*.

Vi har så langt ansett datasettet vårt som for lite til å kunne trene et konvolusjonelt LKTM-nettverk eller alternative romlig-temporale²¹ modeller fra dyp læring. Vi har derfor valgt å klassifisere tidsrekkene hver for seg eller piksel for piksel. En eventuell utvidelse til kontekstuell klassifisering og bruk av romlig-temporale konvolusjonsnettverk vil måtte komme på et senere tidspunkt, når vi ser at det finnes mer treningsdata og merkelapper av tilstrekkelig kvalitet. Vi har ikke på det nåværende tidspunkt forutsetning for å tallfeste hvor mye og hvor gode treningsdata dette vil kreve. I mellomtida foreslår vi å ta i bruk metoder som er mindre dataintensive og beregningskrevende. De valgte metodene beskrives i neste avsnitt.

4.6.5. Valgte metoder for tidsrekkeklassifisering

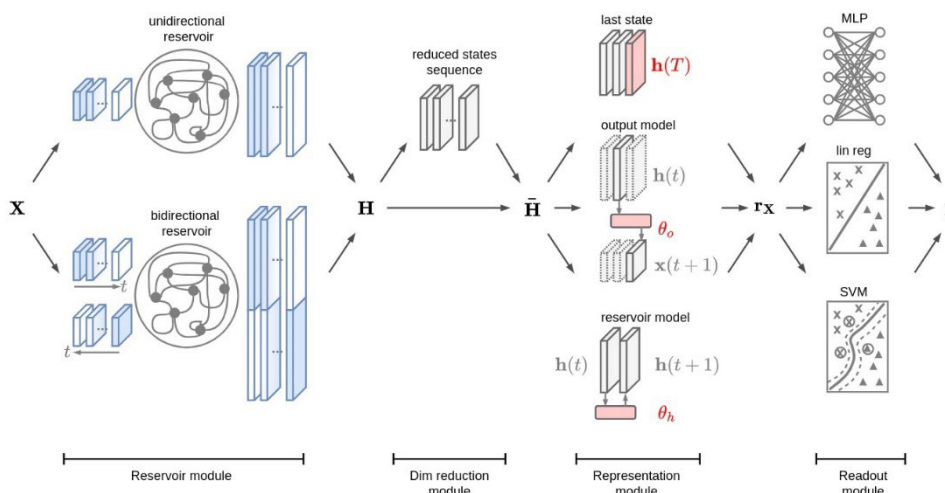
I prosjektet har vi først valgt å teste en type beregningseffektive nevrale nettverk for tidsrekkeklassifisering og -klynging som går under navnet reservoarberegning. Reservoarberegning er en såkalt ensemblemetode der ensemblet består av et utvalg forhåndsgenererte og fastlåste rekursive nevrale nettverk. At de nevrale nettverkene er fastlåst betyr at de motsetning til vanlig ikke trenes eller tilpasses oppgaven ved hjelp av data, men brukes slik de er generert. Mangfoldet i de ulike nettverkene som genereres sørger for de fanger opp ulike mønster i data og trekker ut gode nok egenskaper fra data til å kunne utføre oppgaven, som å klassifisere eller klynge data. Ved å hoppe over treninga i egenskapsutvelgelsen sparer en mye tidskrevende beregninger sammenlikna med vanlige dype nevrale nettverk. Reservoarberegning har like fullt vist seg å produsere gode representasjoner av data, som brukes sammen med tidsrekkenes merkelapper i trening av klassifikatordelen til nettverket (den fjerde og siste modulen i Figur 19), og klassifiseringsresultatene kan i mange tilfeller konkurrere med langt mer beregningskrevende metoder. Den konkrete modellen for reservoarberegning som er brukt i prosjektet er vist i Figur x. Den er også beskrevet i [Bianchi et al., 2021] og på [førsteforfatterens Github-side](#), hvor åpen kildekode er lagt ut.

I tillegg til reservoarberegninger har vi eksperimentert med metodene ROCKET (Random convolutional kernel transform) [Dempster et al., 2022] og MiniROCKET (Minimally random convolutional kernel transform) [Dempster et al., 2021] fra programvarebiblioteket sktime²², som tilbyr åpen kildekode for ulike maskinlæringsoppgaver knytta til tidsrekker. Dette inkluderer tidsrekkeklassifisering, og dermed også endringsdeteksjon. ROCKET er i utgangspunktet å anse som en transformasjon – ikke en klassifikator. Den består av tilfeldige konvolusjonskjerner som anvendes på tidsrekka som analyseres. Dette kan minne om reservoarberegning, bortsett fra at sistnevnte bruker rekursive nevrale nettverk i stedet for konvolusjonskjerner til å prosessere tidsrekka og trekke ut egenskaper. MiniROCKET er en videreutvikling som gjør konvolusjonskjernene nær deterministiske og oppnår nesten samme nøyaktighet med opptil 75 ganger høyere hastighet. Den regnes derfor nå som standardutgaven av ROCKET-algoritmen og som en av de beste tilgjengelige metodene for egenskapsutvelgelse og klassifisering av tidsrekker. Når ROCKET og MiniROCKET omtales som tidsrekkeklassifikatorer, er det fordi sktime tilbyr en pakkeløsning som kobler (Mini)ROCKET-transformasjonen med en lineær klassifikator (RidgeClassifierCV) fra maskinlæringsbiblioteket scikit-learn²³. Denne kalles RocketClassifier.

²¹ Engelsk: *spatio-temporal*; Beskriver noe som omfatter både romlig dimensjon og tidsdimensjon.

²² www.sktime.net/en/latest/examples/02_classification.html

²³ <https://scikit-learn.org/stable/>



Figur 19: Reservoarberegningssnettverket består av flere moduler: Tidsrekkene fores som inndata x til en reservoarmodul som utfører et mangfold av ulike beregninger som transformerer data. De neste modulene reduserer datamengden og trekker ut mønstre som karakteriserer og representerer tidsrekka. Den siste modulen klassifiserer datarepresentasjonen r_x og gir den en merkelapp y som angir *endring* eller *ikke endring*. Kilde: <https://github.com/FilippoMB/Time-series-classification-and-clustering-with-Reservoir-Computing>

Vi har så langt ikke oppnådd ønskede resultater med maskinlæringsmetodikk. De første eksperimentene med reservoarberegninger ga lave deteksjonsrater som viser at klassifikatoren ikke har lært å skille mellom klassene. For å undersøke årsaken til de dårlige resultatene, har vi studert representasjonene fra reservoarberegninga. Vi har vi klynget dem ved å utføre prinsipalkomponentanalyse og plottet de to første prinsipalkomponentene mot hverandre. Plottet viser at det i dette to-dimensjonale rommet ikke er en tydelig klynging av piksler som i referansedatasettet (polygoner som angir bygningsendringer) er merket som *endret* og *ikke endret*. Vi har gjentatt dette med ulike verdier av hyperparametrene²⁴ som kan varieres i reservoarberegningssalgoritmen. Inndatasettet er endret fra å bruke komplette tidsrekker for et helt år til å bruke tidsrekker bestående av månedlige middelerverdier, det vil si tolv datapunkter per tidsrekke. I tillegg er merkelappene endret etter at Kartverket gjorde tilgjengelig manuelt vaskede referansedata. Ingenting av dette har ført til vesentlige endringer i algoritmens ytelse. Eksperimentering med ROCKET-klassifikatorene fra sktime har heller ikke gitt noen gjennombrudd.

Som videre steg i uttestinga av maskinlæringsmetodikk foreslår vi følgende:

- En måte å videreutvikle den klassiske deteksjonsalgoritmen beskrevet i avsnittene 4.1 – 4.5 vil være å bytte ut den enkle tersklingsalgoritmen med klassifikatorer fra moderne maskinlærings, som for eksempel tilfeldig skog-klassifikator²⁵. Denne kan ta multivariate tidsrekker som inndata, og vil da kunne kombinere de ulike regressorene:

²⁴ Hyperparametre er overordnede parametre i maskinlæringsalgoritmer som må bestemmes enten gjennom kvalifiserte valg på bakgrunn av kunnskap om algoritmen og datasettet eller gjennom å søke over mulige verdier og studere påvirkninga av algoritmens ytelse.

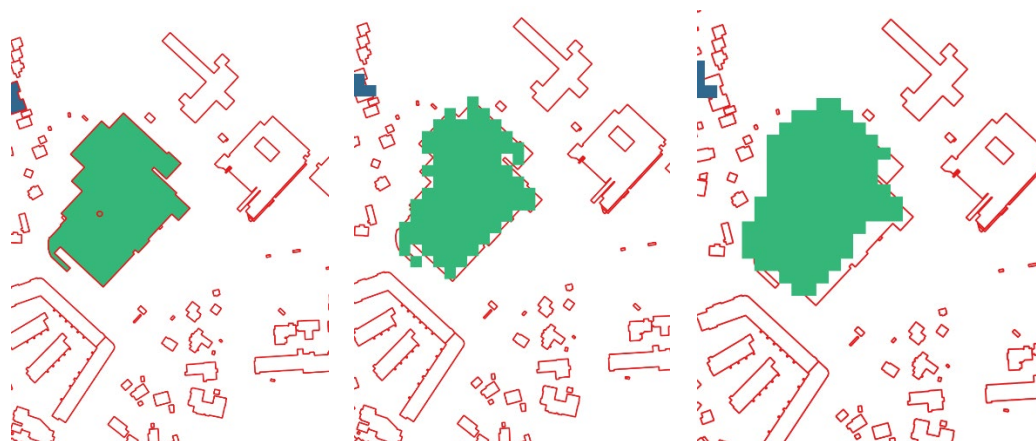
²⁵ Engelsk: *random forest classifier*; Dette er en lettanvendelig og mye brukt klassifiseringsalgoritme med generelt god ytelse på høyst forskjellige datasett.

tilbakespredning målt i ulike geometrier og polarisasjoner samt koherens. Dette vil fjerne behovet for å bestemme optimale terskler. Det vil riktignok innføre noen algoritmespesifikke hyperparametre som må bestemmes, men deteksjonsresultatet vil sannsynligvis være langt mindre sensitive for disse enn for tersklene i den klassiske detektoren som er implementert.

- Inndata kan forbedres ved å gå fra differansen av årlige middel (ett datapunkt per år) til differansen av månedlige middel (tolv datapunkter per år). For å kunne produsere støyreduerte tidsrekker med et spesifisert antall datapunkter per år, bør en implementere en metode som samtidig spekkelfiltrerer og interpolerer/resamplere de årlige tidsrekkene og deretter beregner differansen for etterfølgende år. Bruk av årlige tidsrekker med flere datapunkter eller flerårige tidsrekker vil gjøre det mulig å detektere endringstidspunkter med høyere oppløsning enn årlig.
- Det ser ut til å være behov for en algoritme som gjør automatisk forbedring av merkelappene i treningsdatasettet. Denne vil for eksempel kunne endre og flytte merkelappenes posisjon og tidspunkt for endring gjennom iterativ analyse og ikke-styrt læring som for eksempel klynging av tidsrekke-representasjoner. Her kan en la seg inspirere av litteratur på emner som usikre merkelapper (*noisy labels*) og flytting av merkelapper (*label propagation*). Dette vil være avgjørende for å kunne hente ut merkelapper av tilstrekkelig kvalitet for større områder, da det vil være svært arbeidskrevende å gå gjennom og vaske disse manuelt.

4.7. Falsk alarmrate

Falsk alarmrate (FAR) er definert som raten mellom antall falske deteksjoner og totalt antall deteksjoner. Denne oppgis ofte i prosent. I de fleste deteksjonssystemer er det ønskelig med en lav FAR (noen få %) siden dette ellers vil belaste de som bruker systemet med manuelle operasjoner samt svekke tiltroen til systemet. Falsk alarmrate kan beregnes ved å sammenligne mot bakkesannhet (dvs. eksakte kart over nybygg, revet bygg, tilbygg). For rasteriserte bilder som vi arbeider med er det en kompliserende faktor at rasterisering av høyoppløste polygoner (med nøyaktighet på cm-nivå) fra kartdatabasen ikke er en entydig operasjon gitt et koordinatsystem med 10 m oppløsning. Om en piksel på 10 m x 10 m inneholder et bygg eller ikke vil avhenge av hvor stor del av pikselen som fylles av bygget. En rimelig definisjon er at minst 50% av pikselen bør være dekket av bygget. I Figur 18 vises to eksempler på rasteriseringer, og det er tydelig at denne operasjonen ikke er entydig.

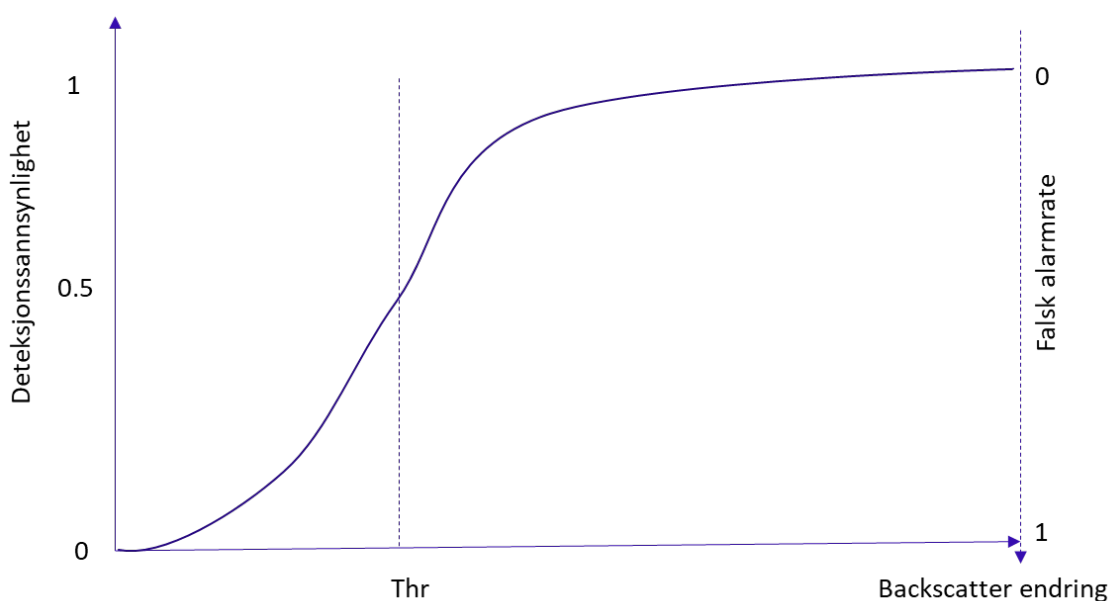


Figur 19. Eksempel på rasterisering av bygg. Til venstre: høy oppløsning. I midten og til høyre: Rasterisering i 10 m oppløsning med små forskyvninger av koordinatsystemets origo.

Konsekvensen av problemene med å rasterisere kartdatabasen til en entydig bakkesannhet er at det også vil bli introdusert feil i beregningen av falske alarmrater. Når man tar dette i betraktning, samt at man heller ikke kan forvente at deteksjon av bygg vil være perfekt (siden deler av bygg ofte vil forårsake sterkere endringer i tilbakespredning enn andre deler av bygget), må man betrakte de estimerte falsk alarmratene/deteksjonssannsynlighetene mer som et kvalitetsestimat av klassifiseringen, enn som en fullgod deteksjonssannsynlighet. Tatt dette i betraktning har vi gjort forsøk på å på å estimere FAR.

For å beregne løpende falsk alarmrate for hver enkelt piksel som detekteres er det ikke uten videre mulig å gjøre valideringer som over. En forenklet metode er da å bruke avstand til deteksjonsterskel som en proxy for falsk alarmraten. Endringer nær deteksjonsterskelen har da høy FAR mens endringer høyt over deteksjonsterskelen har tilsvarende lav FAR.

Deteksjonssannsynligheten kan modelleres med funksjonen: $P_d = \text{sig}(|\sigma^{y+1}(x,y) - \sigma^y(x,y)| - \text{Thr})/k$, der vi bruker sigmoid-funksjonen $\text{sig}(x) = 1/(1 + e^{-x})$ og konstanten k som bestemmer hvor bratt kurven skal være. Falsk alarmraten blir da: $\text{FAR} = 1 - P_d$.

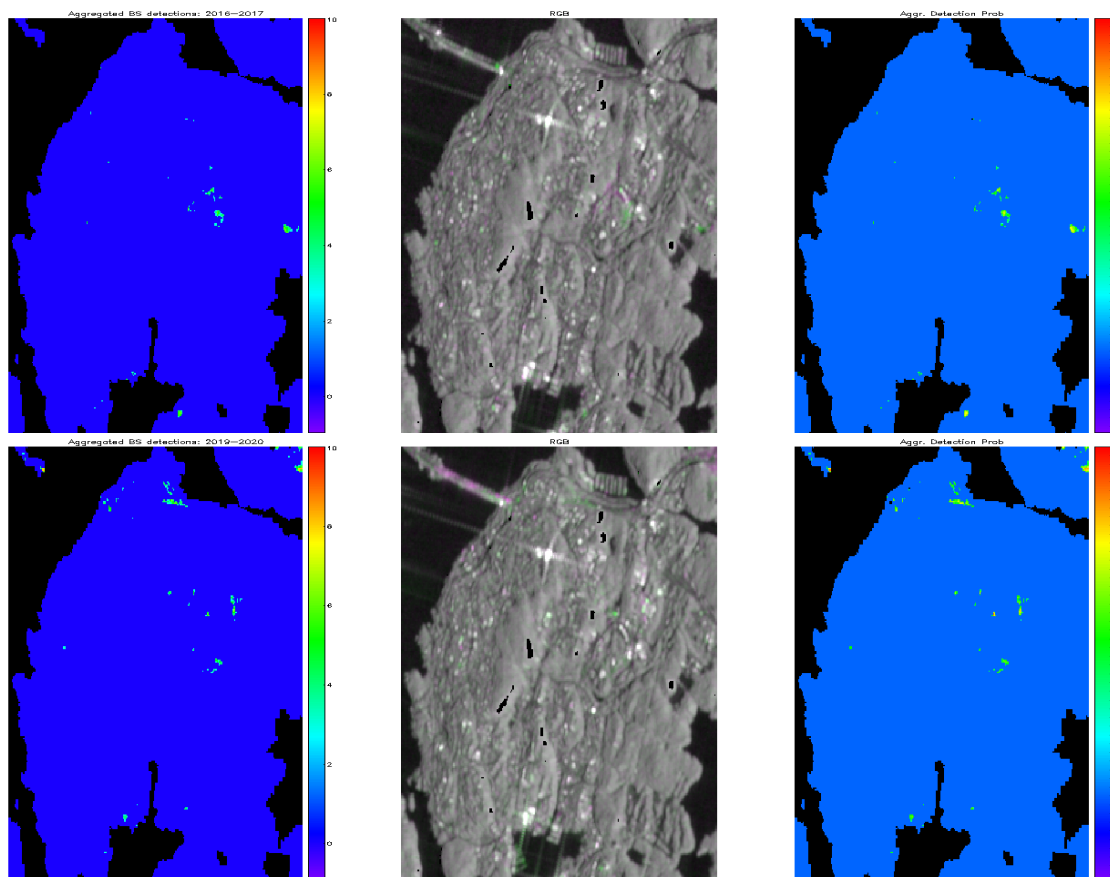


Figur 20. Deteksjonssannsynlighet og falsk alarmrate beregnet med utgangspunkt i endring i tilbakespredning.

Siden vi har flere satellittgeometrier og to polarisasjoner (til sammen N geometrier og polarisasjoner) vil den samlede deteksjonssannsynlighet (eller FAR) bli en vektet sum over de enkelte sannsynlighetene:

$$P_d = \frac{1}{N} \sum_{N_{track}} \sum_{VV, VH} \text{sig}(|\sigma^{y+1}(x,y) - \sigma^y(x,y)| - \text{Thr})/k$$

I figur 20 vises to eksempler for endringer fra hhv 2016-2017 og 2019-2020 der vi har beregnet deteksjonssannsynligheten på denne måten.



Figur 21. Deteksjonssannsynlighet for Sem-området for de to periodene 2016-2017 og 2019-2020.

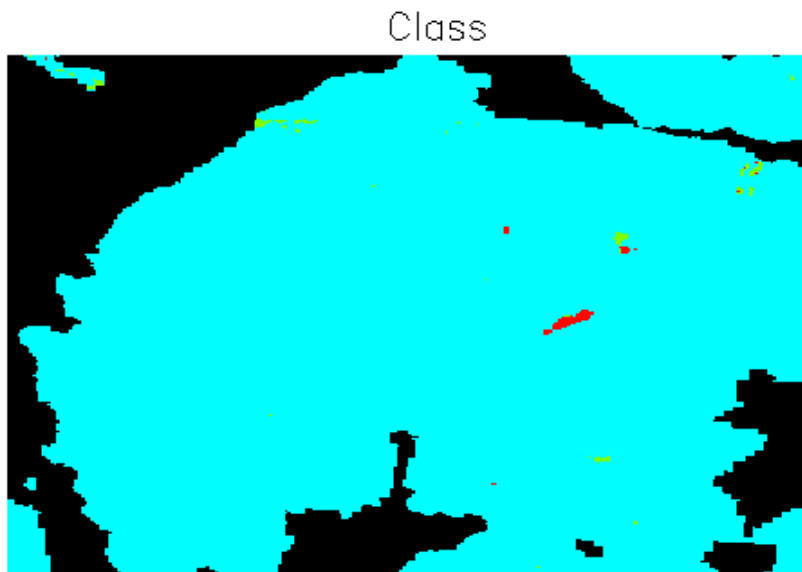
4.8. Klassifisering av endringer

Å klassifisere endringer som nybygg, revet bygg eller tilbygg er ikke en entydig operasjon. I vår tilnærming er det nærliggende å anta at et nybygg i en piksel fører til økt tilbakespredning, mens et revet bygg tilsvarende fører til redusert tilbakespredning. Ved å bruke terskelmetoden fra avsnitt xx kan man skissemessig ha følgende algoritme for piksler (x,y) med signifikante endringer fra ett år til det neste:

- $\sigma^{y+1}(x,y) - \sigma^y(x,y) > Thr$, indikerer nybygg
- $\sigma^{y+1}(x,y) - \sigma^y(x,y) < - Thr$, indikerer revet bygg
- $\sigma^{y+1}(x,y) - \sigma^y(x,y) > Thr$ OG forbundet med eksisterende bygg (fra kartdatabase) -tilbygg

Den siste klassen er ikke implementert, og blir dermed klassifisert som nybygg. For å gjøre dette må man koble til database for eksisterende bygg. Et eksempel på tilbygg ble vist i avsnitt 4.2 figur 12

og 13. I figur 21 vises et eksempel på klassifisering med denne metoden. Nybygg og revet bygg er klassifisert med ulik farge.



Figur 22. Klassifisering av bygningsendring (2020-2021). Rød indikerer nybygg, grønn indikerer revet bygg.

4.9. Følsomhet for innstillinger for deteksjonsalgoritmen

Metoden for deteksjon basert på årlige middelværdier har to parametere som kan tunes:

- Thr1 er deteksjonsterskel for endring i tilbakespredning fra et år til det neste.
- Thr2 er en terskel for hvor mange av de 8 geometriene/polarisasjonene som kreves før deteksjon

Tabell 2 under viser hvordan antall deteksjoner varierer med innstillingen til disse parameterene. Man må forvente at deteksjonssannsynlighet og falsk alarmrate vil variere tilsvarende.

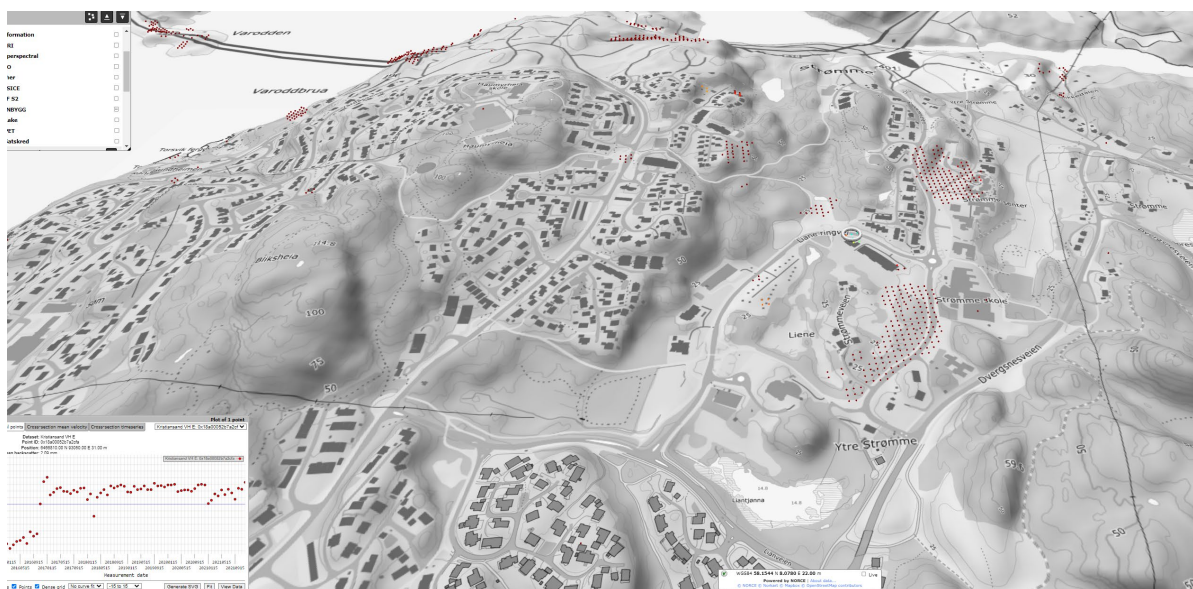
Table 2. Sensitivitet for tuning parametere i deteksjonsalgoritmen.

Område	Thr1	Thr2	#Deteksjoner
V	2.0	3	799
V	3.0	3	470
V	4.0	3	281
V	4.0	2	406
V	4.0	4	195
E	2.0	3	871

E	3.0	3	471
E	4.0	3	243

4.10. Visualisering i NLIVE

I prosjektet fant vi at det kunne være nyttig å visualisere tidsrekkene av tilbakespredning i visualiseringsprogrammet NLIVE, som NORCE bruker utstrakt til visualisering av ulike jordobservasjonsdata (bl.a. i InsarNorge). Tidsrekkene i prosjektet kan finnes i (<https://nlive.norceresearch.no/>). I figur 21 under vises ei tidsrekke for et bygg der vi brukte endringsdeteksjon for å finne signifikante endringer. I tidsrekka ser vi en betydelig økning av tilbakespredning i perioden



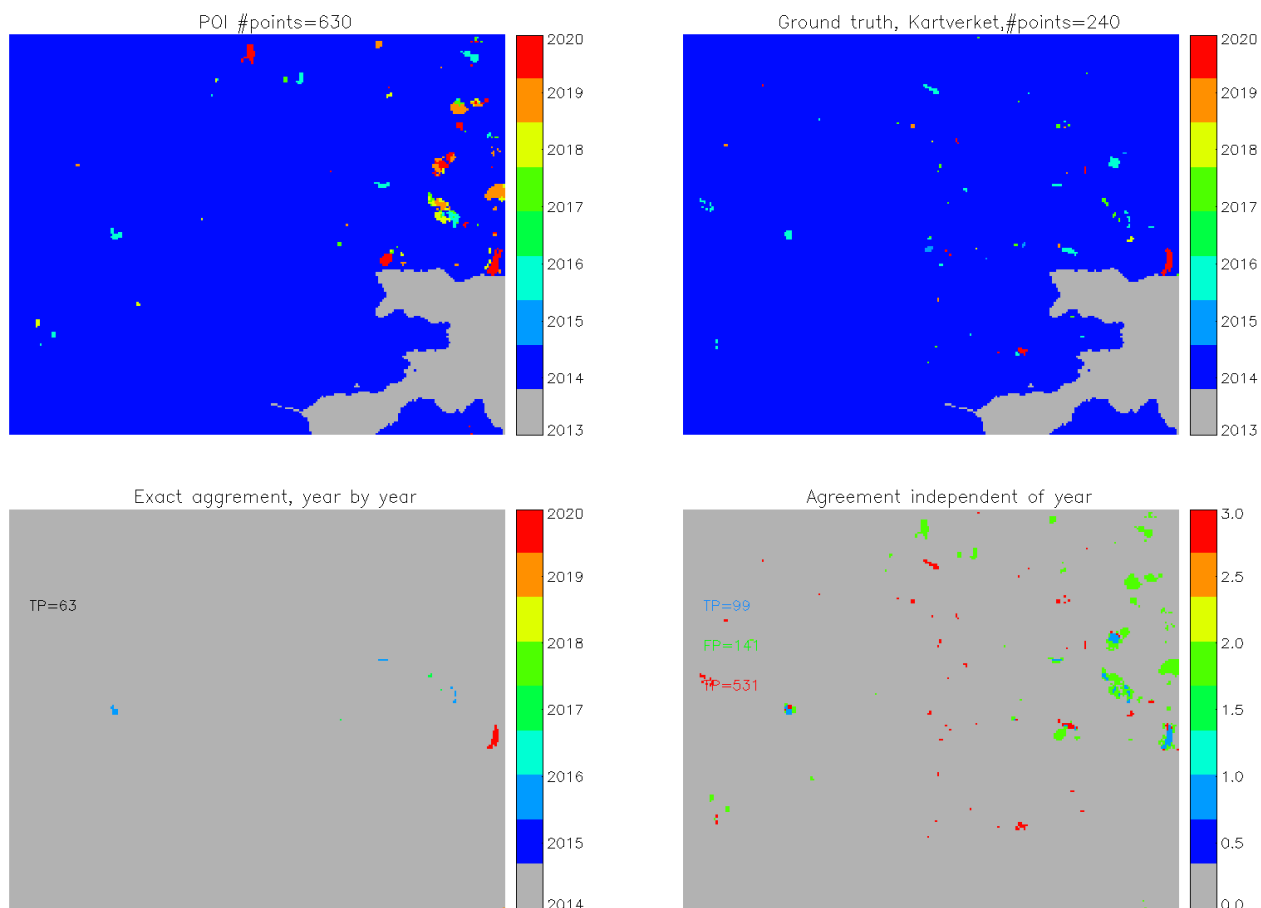
Figur 23. Visualisering av tidsrekker av tilbakespredning i tre dimensjoner. Hver prikk viser en piksel der det er detektert signifikant endring. Ved å velge en bestemt piksel, kan hele tidsrekka mellom 2016 og 2021 visualiseres nede til venstre, og man kan finne perioden der endringer skjer.

5. Resultat og diskusjon

I dette avsnittet gjennomgås resultatene fra endringsdeteksjon sammenlignet mot manuelt kvalitetssikret endringskart fra Kartverket

5.1. Resultat fra testområde vest (Vågsbygd)

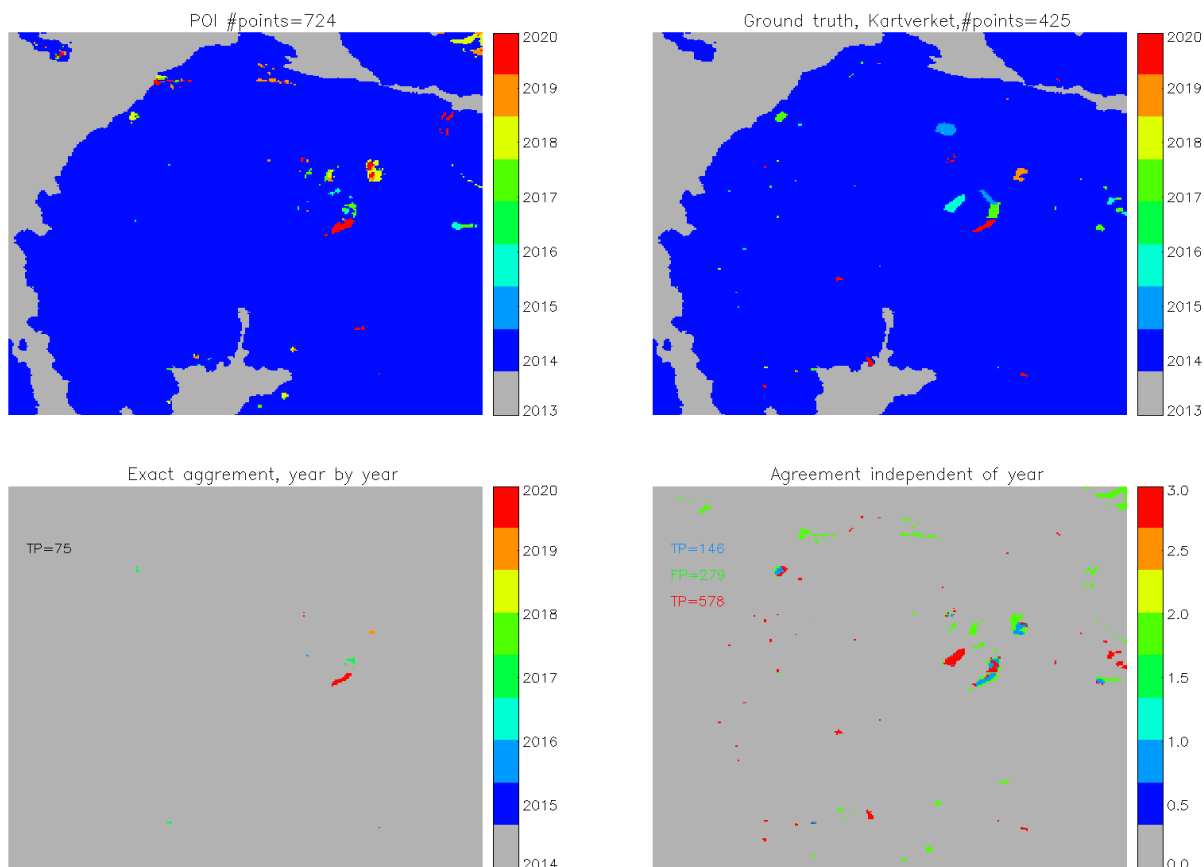
Figur 24 viser resultatet fra endringsdeteksjoner basert på midlet tilbakespredning for hvert år (avsnitt 4.1). Dette kan vi sammenligne mot bakkesannhet. Vi beregner deretter hvordan de to kartene stemmer overens ved å finne eksakt overenstemmelse der både årstall for endring og posisjonen skal stemme (nederst venstre). I dette tilfellet er det relativt få punkter som stemmer overens (63 pixel av totalt 240 punkter i ground truth). Dette svarer til en deteksjonssannsynlighet på 26%. Hvis vi ikke tar hensyn til årstallet for endringer (nederst til høyre) finner vi 99 punkt med overenstemmelse som svarer til en deteksjonssannsynlighet på 41%. Falsk alarmrate (uavhengig av år) blir $141/240=59\%$.



Figur 24. Sammenligning av detekterte endringer i Vågsbygd (øverst til venstre) og bakkesannhet (øverst til høyre) fra Kartverket. Nederst venstre vises eksakt overenstemmelse med deteksjon og bakkesannhet (ikke bare posisjon, men også årstallet skal stemme). Nederst til høyre vises overenstemmelse mellom deteksjon og bakkesannhet uavhengig av årstall. Blått viser områder med full overenstemmelse, rødt viser falske positive (deteksjoner uten bakkesannhet) mens grønt viser sanne negative (bakkesannhet uten deteksjon).

5.2. Resultat fra testområde øst (Sem)

Figur 23 viser resultatet fra Sem. I dette tilfellet er det relativt få punkter som stemmer overens (75 piksel av totalt 425 punkter i Ground truth). Dette svarer til en deteksjonssannsynlighet på 18%. Hvis vi ikke tar hensyn til årstallet for endringer (nederst til høyre) finner vi 146 punkt med overenstemmelse som svarer til en deteksjonssannsynlighet på 34%. Falsk alarmrate (uavhengig av år) blir $272/425 = 64\%$.



Figur 25. Sammenligning av detekterte endringer i Søm (øverst til venstre) og bakkesannhet (øverst til høyre) fra Kartverket. Nederst venstre vises eksakt overenstemmelse med deteksjon og bakkesannhet (ikke bare posisjon, men også årstallet skal stemme). Nederst til høyre vises overenstemmelse mellom deteksjon og bakkesannhet uavhengig av årstall. Blått viser områder med full overenstemmelse, rødt viser falske positive (deteksjoner uten bakkesannhet) mens grønt viser sanne negative (bakkesannhet uten deteksjon).

5.3. Diskusjon

Resultatene fra avsnitt 5.1 og 5.2 for de to områdene Vågsbygd og Søm viser at vi har relativt liten dekteksjonssannsynlighet hhv 26%/42% (Vågsbygd) og 18%/34% for Søm. Dette er betydelig lavere enn man skulle ønske seg av en deteksjonsmetode. Etter vår oppfatning er det flere faktorer som gjør at deteksjonssannsynligheten/falsk alarmrate blir lav/høy:

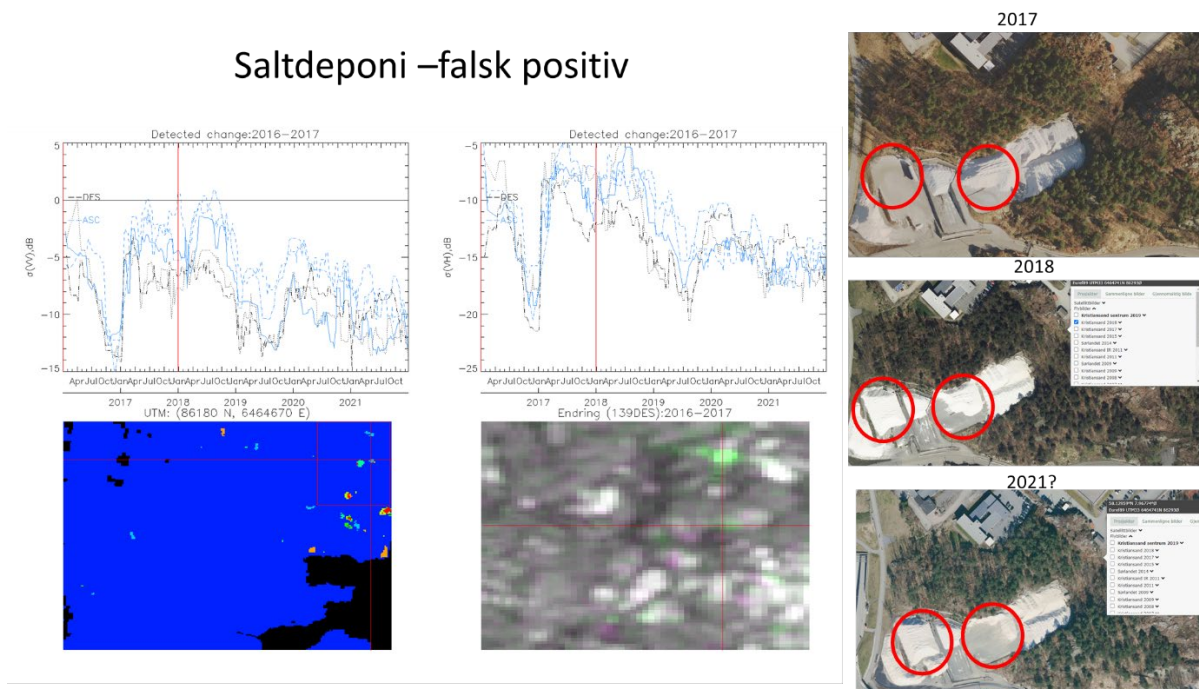
Unøyaktig bakkesannhet: Kartene som angir årstall for igangsettelse av bygg er unøyaktig av flere grunner:

- Rasterisering fra vektordata i Kartverkets database til 10 m grid (se avsnitt 4.7)
- Validering basert på NorgeIBilder er avhengig av i beste fall et bilde per år. Dermed kan man ikke si sikkert i de fleste tilfeller nøyaktig når byggeaktivitet starter.

Unøyaktig deteksjon: SAR deteksjon har alltid en del usikkerhet som behefter dataene:

- Tidspunkt for byggestart er unøyaktig siden vi midler over to år. Byggestart settes til det første året, mens den i prinsippet kan være nærmere slutten av det andre året.
- Det tas ikke hensyn til byggenes høyde i geokoding og videre prosessering. Siden SAR har skrått innfall, vil dermed refleksjoner fra over ca 5 m høyde bli feilplassert geometrisk. Dette vil føre til at bygninger sprer seg utover (både mot øst og vest) når ulike geometrier brukes. I figur 22 kan man se denne effekten ganske tydelig
- Deteksjoner kan forekomme uten at det er byggeaktivitet. Skoghogst, bygging av kai/steinfylling, flytting av sandtak samt mange andre inngrep kan feilaktig bli klassifisert som byggeaktivitet. Vi vil anta at objekter på byggetomter (utover posisjonene til bygget) som bygningsmaskiner, heisekraner, containere også kan dukke opp som deteksjoner (falske alarmer).
- I sammenligningene i 5.1 og 5.2 skilte vi ikke mellom bygning og rivning. Det er uklart om dette er hensyntatt i bakkesannheten. Revet bygg kan dermed opptre som falske alarmer.
- Støyreduksjon og terskling av middelerverdi bildene vil alltid være følsom for variasjoner. Vi har valgt terskler som gir en tydelig endring i tilbakespredning. I noen tilfeller er det kanskje ikke tilstrekkelig målbar endring mellom referanse verdi før rivning og endring etter nybygg til at man får deteksjon.

Saltdeponi –falsk positiv



Figur 26. Eksempel på falsk positiv. Salt- eller sanddeponi.

6. Programvare

NORCE har utviklet flere kodebiter i prosjektet som leveres til oppdragsgiver som åpen kildekode.

6.1. Endringsdeteksjon basert på terskling

Python-skriptet «senbygg_classical_detector.py» utfører deteksjon av bygginger basert på en klassisk tersklingsalgoritme anvendt på differanser av årlige middelveier av målt tilbakespredning. Programmet forutsetter tilgang til ei tidsrekke med geotiff-bilder over et gitt område og leverer som output en geotiff der alle piksler hvor man detekterer endring blir kodet med aktuelt årstall for endring. Det vil si, hvis man sammenligner midlere bilde for 2015 og 2016, så blir en piksel med endring kodet med verdien 2015.

6.2. Endringsdeteksjon basert på maskinlæring

Python-skriptet «senbygg_machine_learning.py» inneholder eksperimenter med maskinlæringmetoder anvendt på tidsrekker av tilbakespredning. Disse eksperimentene har ikke ført til vellykkede resultater i form av endringskart eller andre produkter, men er grunnlag for videre forskning og utvikling på denne type metodikk.

6.3. Eksport av endringsdeteksjonsbilde til GML

Endringsdeteksjonsalgoritmene returnerer raster. Et skript (senbygg_vectorize.py) har blitt laget for vektorisering av slike raster. Skriptet tar en rasterfil som input og skriver en GML-fil som output (shapefile eller geojson er også mulig). Skriptet kjøres fra terminal:

Programmet leser rasterfilen “change_detection.tif” og lager en ny fil «change_detection.gml». Kartprojeksjon for outputfilen kan settes med flagg --epsg, som f eks:

der GML-filen vil være i EPSG:4326. Hvis epsg-koden ikke settes eksplisitt, vil den være EPSG:25833. Navnet på outputfilen kan også settes eksplisitt med flagg --target:

For hjelp, tast i terminal:

Et eksempel på vektorisering av et raster vises i figur 27.



Figur 27. Eksempel på vektorisering.

Årstallet for endring i det aktuelle pikselen, eller klyngen av piksel, som rasteriseres lagres i GML-fila som attributt for det aktuelle polygonet. Her kunne det vært aktuelt å også lagre flere attributter som deteksjonssannsynlighet/kvalitet (se avsnitt 4.7) og klassifisering (avsnitt 4.8), men dette er ikke implementert siden det ville fordre flere input-filer i API-et.

7. Oppsummering

Prosjektet SENBYGG har demonstrert bruk av data fra radarsatellittene Sentinel-1 A og B til deteksjon av bygningsendringer i to bydeler av Kristiansand. Vi har brukt data fra 2015-2021 og til sammen 1672 bilder. Vi har undersøkt flere metoder for å gjøre deteksjon av både revet bygg samt nybygg og tilbygg basert på endringsdeteksjon i målinger av tilbakespredning med ulike satellittgeometrier, i geometrinormaliserte målinger av tilbakespredning, og i målinger av interferometrisk koherens, samt mulig bruk av flere av disse målingene samtidig. Bruk av metoder fra maskinlæring (kunstig intelligens) har også vært studert. Siden prosjektet er begrenset i omfang, har det ikke vært mulig å gå i full dybde på alle disse problemstillingene, men de er behandlet i respektive avsnitt.

Metoden vi har hatt mest fokus på er endringsdeteksjon i radarmålinger av tilbakespredning midlet over hele år for separate geometrier og polarisasjoner. Ved å midle over hele år fjernes mye av spekkelstøyen som vanligvis er assosiert med SAR-bilder og man kan bevare den opprinnelige oppløsninga til satellittproduktene (10m). Som vist i avsnitt 5 finner vi moderate deteksjonssannsynligheter og falsk alarm-rater med denne metoden. Disse sannsynlighetene er alle beregnet utfra en bakkesannhet som vi ikke har full kontroll over. Vi mistenker at det er en del feil knyttet til både tidspunkt og posisjon for de rapporterte bygningsendringene vi bruker som referanse.

En mulig måte å finne tidspunkt for byggestart vil være å bruke de årlige deteksjonene som utgangspunkt, og deretter studere tidsrekker av tilbakespredning (evt. i kombinasjon med koherens) slik vi har illustrert i avsnitt 4.5. Man kan da ofte finne tydelige økninger/senkninger i tilbakespredning og muligens assosiere starten på disse økningene med igangsettelse. For mange av eksemplene vi har studert vil det være mulig å oppnå nøyaktigheter som er i størrelsesorden én måned eller mindre.

NORCE har også arbeidet med å bruke ulike maskinlæringsmetoder for å detektere bygningsendringer. I dette arbeidet har vi kartlagt hvilke særlige utfordringer problemstillinga bringer opp og hvilke konsekvenser dette får for valg av metodikk, samt behov for tilpasning av eksisterende algoritmer. Dette arbeidet har foreløpig ikke ført til metoder som gir gode resultater, noe våre foreløpige analyser tilskriver kvaliteten på referansedata og merkelapper som angir piksler med bygningsendring og til dels også mengden av treningsdata. Forslag til forbedringer og endringer ved videre eksperimentering er gitt i avsnittene som beskriver endringsdeteksjon med maskinlæring.

8. Referanser

- Aimaiti, Y., Sanon, C., Koch, M., Baise, L. G., & Moaveni, B. (2022). War Related Building Damage Assessment in Kyiv, Ukraine, Using Sentinel-1 Radar and Sentinel-2 Optical Images. *Remote Sensing*, 14(24), 6239.
- Bartsch, A., Pointner, G., Ingeman-Nielsen, T., & Lu, W. (2020). Towards Circumpolar Mapping of Arctic Settlements and Infrastructure Based on Sentinel-1 and Sentinel-2. *Remote Sensing*, 12(15), 2368.
- Bianchi, F. B., Scardapane, S., Løkse, S. & Jenssen, R. (2021). Reservoir computing approaches for classification and representation of multivariate time series. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5):2169-2179.
- Chini, M., Pelich, R., Hostache, R., Matgen, P., & López-Martínez, C. (2018). Towards a 20 m global building map from sentinel-1 sar data. *Remote Sensing*, 10(11), 1833.
- Dempster, A., Petitjean, F., & Webb, G. I. (2020). ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454-1495.
- Dempster, A., Schmidt, D. F., & Webb, G. I. (2021). Minirocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 248-257).
- Ferrentino, E., Nunziata, F., Bignami, C., Graziani, L., Maramai, A., Migliaccio, M., & Stramondo, S. (2022). On the Combination of Dual-Polarization Sentinel-1 Ascending/Descending Orbiting Passes to Estimate Damage Due to the 2016 Central Italy Earthquake. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15, 9509-9518.
- Hlaváčová, I., Kačmařík, M., Lazecký, M., Struhár, J., & Rapant, P. (2022) Automatic detection of new building construction from sentinel-1 multi-temporal imagery. <https://doi.org/10.31490/9788024846026-4>
- Koppel, K., Zalite, K., Voormansik, K., & Jagdhuber, T. (2017). Sensitivity of Sentinel-1 backscatter to characteristics of buildings. *International Journal of Remote Sensing*, 38(22), 6298-6318.
- Letsios, V., Faraslis, I., & Stathakis, D. (2023). Monitoring Building Activity by Persistent Scatterer Interferometry. *Remote Sensing*, 15(4), 950.
- Meneses III, S. F., & Blanco, A. C. (2022). Rapid mapping and assessment of damages due to typhoon rai using sentinel-1 synthetic aperture radar data. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.
- Petrushevsky, N., Manzoni, M., & Monti-Guarnieri, A. (2022). Fast Urban Land Cover Mapping Exploiting Sentinel-1 and Sentinel-2 Data. *Remote Sensing*, 14(1), 36.

9. Appendiks: SNAP

SNAP manual for backscatter/interferogram/coherence product generation

Preparation

For this manual, the Sentinel Application Platform (SNAP) from ESA is required. The SNAP software can be downloaded from <https://step.esa.int/main/download/snap-download/>.

The Sentinel-1 data to be analysed should be of the types GRDH (backscatter analysis) and SLC (coherence analysis). Sentinel-1 imagery can for example be downloaded from the Copernicus Open Access Hub (<https://scihub.copernicus.eu/>), where an Area Of Interest (AOI) can be specified.

The processing in SNAP can utilize certain automatic-download DEM's (e.g. SRTM-1sec, Copernicus 30m etc). If an external DEM should be used, then the file should cover all bursts processed. SNAP can read many different raster formats, including GeoTIFF, ENVI, NetCDF etc. The file needs to have a geographic reference and can be projected.

In SNAP, processing can be performed step-by-step or via the Graph builder. This manual will utilize the Graph builder, which allows to afterwards batch-process image stacks. More advanced processing via the python-SNAP interface module *snappy* or command-line processing is also possible, but not described in this manual.

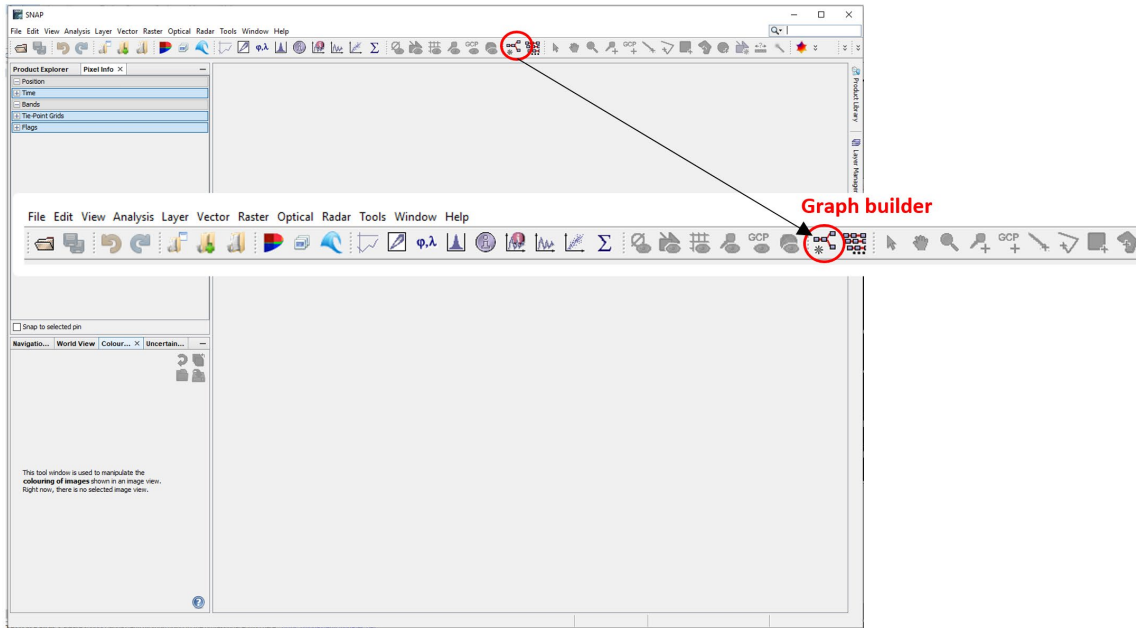
Recipe to generate a stack of backscatter products from SNAP

Preparation:

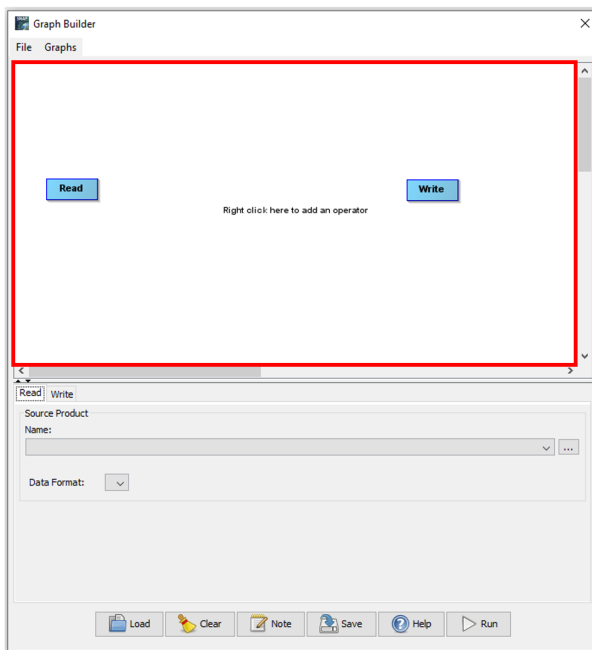
- 1) Collect GRDH data
- 2) If external DEM to be used, prepare DEM (Convert to accepted file format)

Processing in SNAP:

The following steps can be performed either individually or via the Graph processing tool (GPT) as batch-processing. For the GPT processing, open the Graph builder in SNAP (red circle).



A window called graph builder will pop up. Here, several processing steps can be linked together and run at once or saved as “recipe” for batch-processing. In the work space (red rectangle), processing steps can be added by right click. The steps have to be connected by pulling an arrow from one processing step to the next one. In the lower part of the window, the settings for each processing step can be specified.

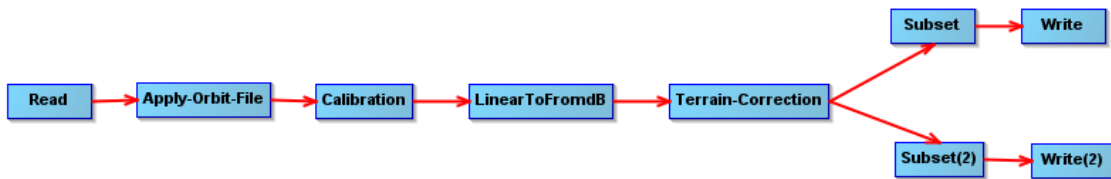


In the Graph Builder, add the following from right to left by right-click and choosing from the processing options:

1. Add > Input-Output > Read
2. Add > Radar > Apply-Orbit-File
3. Add > Radar > Radiometric > Calibration
4. Add > Raster > Data conversion > LinearToFromdB
5. Add > Radar > Geometric > Terrain correction > Terrain correction

6. 2x: Add > Raster > Subset (one per AOI and per source band)
7. 2x: Add > Input-Output > Write (one per AOI and per source band)

The different steps then have to be connected, and the resulting graph should look like this:



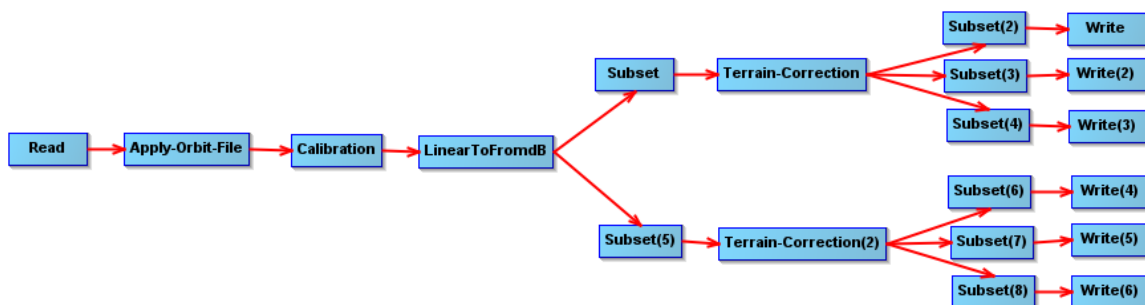
Now the settings have to be specified underneath the graph:

- Read: do not specify any input yet
- Apply-Orbit-File: default settings
- Calibration: specify polarization (all or specific type)
- LinearToFromdB: default settings
- Terrain correction: Source bands = all; Digital Elevation Model = pick DEM wanted (auto-download or manual); Pikel spacing (m) = 10m; Map projection = WGS84 UTM 33, unselect "mask out areas with no elevation" (optional)
- Subset: One subset per AIO (e.g. one West and one East). Press "geographic coordinates" and specify AOI under the worldview. Add here a polygon statement in WKT format, e.g. POLYGON((7.930165640451276 58.111809, 7.930165640451276 58.131931, 7.967902136589363 58.131931, 7.967902136589363 58.111809, 7.930165640451276 58.111809)). The coordinates must be in order "longitude latitude" and in decimal degrees.
- Write: Specify output path to respective folder (for each subset region). Output format can be specified under "save as", and formats include GeoTIFF (would lose metadata) and SNAP's own format .dim (includes the metadata).

If also incidence angle and layover-shadow masks are needed, the graph can be extended by adding additional operators:

1. 1x Add > Radar > Geometric > Terrain-Correction
2. 6x Add > Raster > Subset
3. 6x Add > Input-Output > Write

The additional processing steps should be connected as follows:

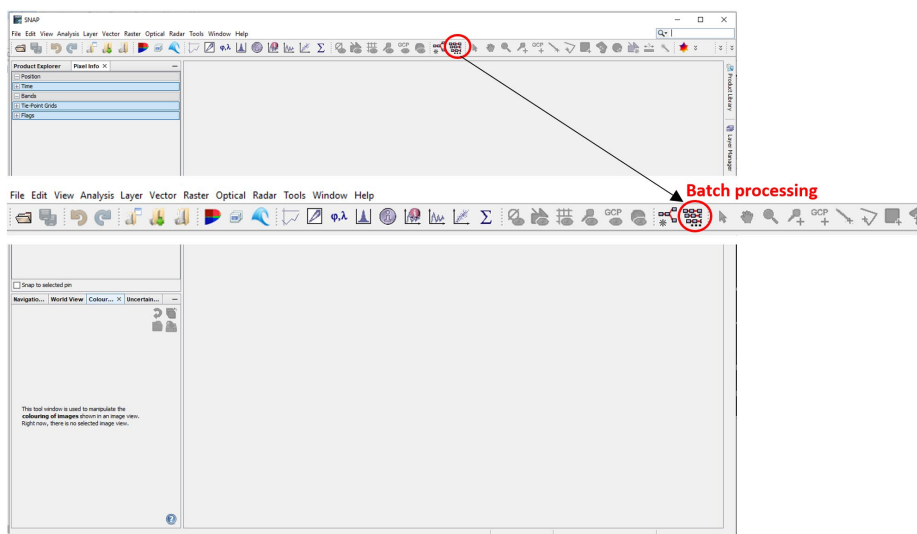


The settings should be updated:

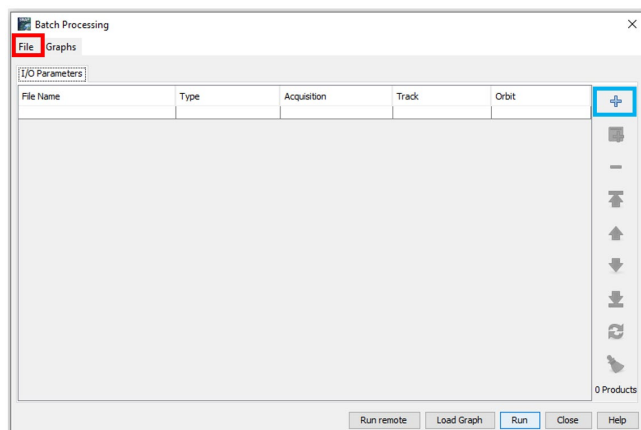
- Additional subsets following LinearToFromDB: One subset per AOI. Press “geographic coordinates” and specify AOI under the worldview. Add here a polygon statement in WKT format, e.g. POLYGON((7.930165640451276 58.111809, 7.930165640451276 58.131931, 7.967902136589363 58.131931, 7.967902136589363 58.111809, 7.930165640451276 58.111809)). The coordinates must be in order “longitude latitude” and in decimal degrees.
- Terrain-Correction: Source bands = sigma0; Digital Elevation Model = pick DEM wanted (auto-download or manual); Pixel Spacing (m) = 10m; Map projection = WGS84 UTM 32, unselect “mask out areas with no elevation” (optional). Select Output bands for: “Selected source band”, “Local Incidence Angle”, “Shadow-Layover-Mask”. Note: Update also the other Terrain-Correction operator.
- Subsets following Terrain-Correction: One subset per output product. Select source band to be exported (e.g. sigma0, incidence angle or shadow-layover-mask).
- Write: Specify output path to respective folder (for each subset region and for each product type). Output format can be specified under “save as”, and formats include GeoTIFF (would lose metadata) or SNAP’s own format (.dim) which includes the metadata.

Afterwards, the graph can be saved by clicking on File > save (upper left corner). The graph can be run from the Graph builder for single images.

For image stacks it is wise to use the batch-processing option. Close the Graph builder and open the batch-processing short cut:



The following window will open, where the previously created graph can be loaded from File > Open graph (red rectangle).



All files to be processed can be added by clicking on the “+” (blue rectangle). Then, the batch processing can be started by clicking on “Run”. The resulting files can then be found in the previously specified output folder. It is recommended to use a specific folder for area/incidence angle, area/layover-shadow, area/backscatter because during batch-processing SNAP saves output files using the filename of the import product. Thus, user-defined output filenames are lost when batch-processing.

Recipe to provide a stack of interferogram and coherence products

Preparation:

1. Collect SLC data
2. If external DEM to be used, prepare DEM (Convert to accepted file format)

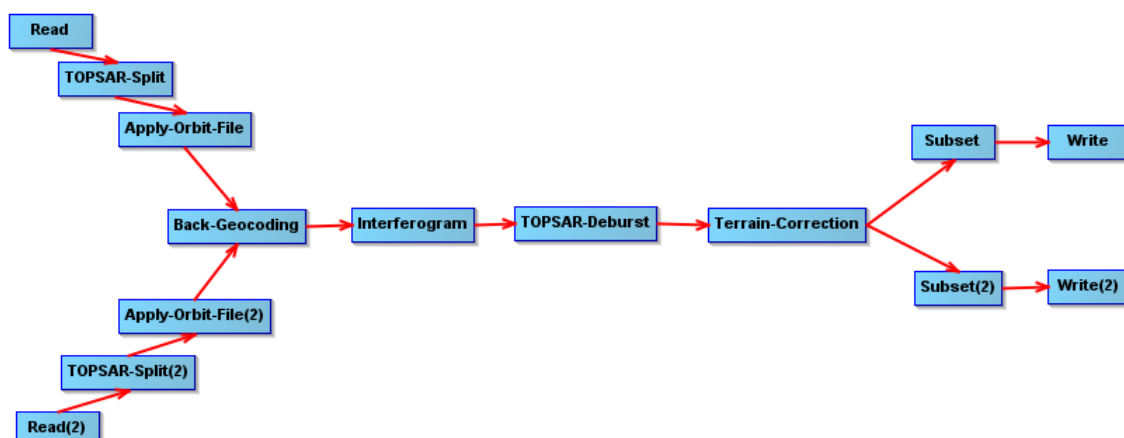
Processing in SNAP:

The following steps can be performed either individually or via the Graph builder. Batch-processing is only limited possible, as the interferogram master cannot automatically change and has to be specified in the used graph.

In the following, the steps will be described for building a processing graph. For this, open the Graph builder in SNAP as previously done for the Backscatter processing. In the Graph Builder, add the following from right to left by right-click and choosing from the processing options:

1. 2x Add > Input-Output > Read
2. 2x Add > Radar > Sentinel-1-TOPS > TOPSAR-Split
3. 2x Add > Radar > Apply-Orbit-File
4. Add > Radar > Coregistration > S-1 TOPS Coregistration > Back-geocoding
5. Add > Radar > Interferometric > Products > Interferogram
6. Add > Radar > Sentinel-1-TOPS > TOPSAR-Deburst
7. Add > Radar > Geometric > Terrain correction > Terrain correction
8. Add > Raster > Subset
9. Add > Input-Output > Write

The different steps then have to be connected, and the resulting graph should look like this:



Now the settings must be specified:

- Read: Add the two input images for the interferogram pair
- TOPSAR-Split: choose subswath and burst(s) which cover the AOI. Choose HH polarisation.
- Apply-Orbit-File: default settings
- Back-geocoding: DEM = choose type (auto-download or manual file), unselect “mask out area with no elevation” (optional)
- Interferogram: default settings
- TOPSAR-Deburst: default settings
- Terrain correction: Source bands = coherence; Digital Elevation Model = pick DEM wanted (auto-download or manual); Pixel Spacing (m) = 10m; Map projection = WGS84 UTM 33, unselect “mask out areas with no elevation” (optional)
- Subset: One subset per AIO. Press “geographic coordinates” and specify AOI under the worldview. Add here a polygon statement in WKT format, e.g. POLYGON((7.930165640451276 58.111809, 7.930165640451276 58.131931, 7.967902136589363 58.131931, 7.967902136589363 58.111809, 7.930165640451276 58.111809)). The coordinates must be in order “longitude latitude” and in decimal degrees.
- Write: Specify output path to respective folder (for each subset region). Output format can be specified under “save as”, and formats include GeoTIFF (would lose metadata) or SNAP’s own format (.dim) which includes the metadata.

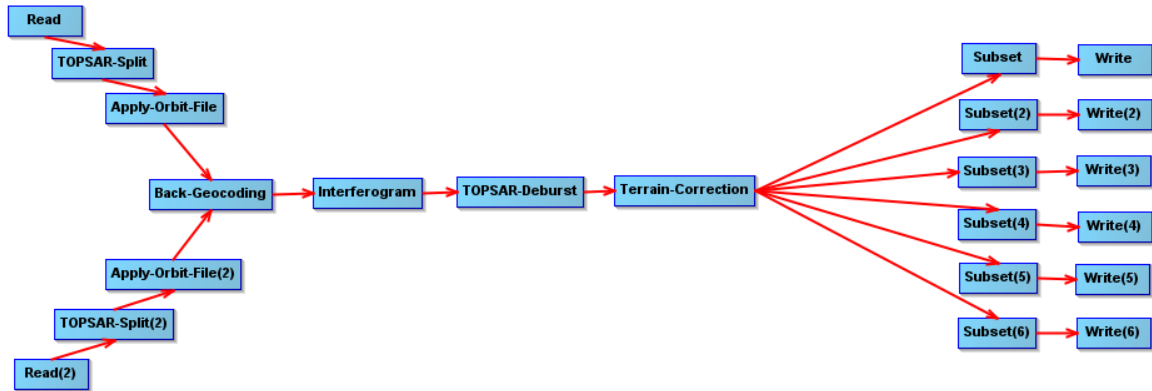
To process several interferograms, the read operators have to be adjusted to the new image pair. The remaining settings can stay the same.

Batch processing for a defined master is possible. For this, the second read option in the graph should specify the master image. Add the graph to the batch-processing window as previously done for the Backscatter processing. Add all images that should be paired with the master image via the “+” symbol. Click on run and for each added image, one interferometric pair will be created and processed with the specified master and following the graph steps.

If also incidence angle and layover-shadow masks are needed, the graph can be extended by adding the following:

1. 4x Add > Raster > Subset
2. 4x Add > Input-Output > Write

The additional processing steps should be connected like this:



The settings should be updated:

- Terrain-Correction: Source bands = coherence; Digital Elevation Model = pick DEM wanted (auto-download or manual); Pixel Spacing (m) = 10m; Map projection = WGS84 UTM 32, unselect "mask out areas with no elevation" (optional). Select Output bands for: "Selected source band", "Local Incidence Angle", "Shadow-Layover-Mask".
- Additional Subset following Terrain-Correction: One subset per AIO and per output product. Press "geographic coordinates" and specify AOI under the worldview. Add here a polygon statement in WKT format, e.g. POLYGON((7.930165640451276 58.111809, 7.930165640451276 58.131931, 7.967902136589363 58.131931, 7.967902136589363 58.111809, 7.930165640451276 58.111809)). The coordinates must be in order "longitude latitude" and in decimal degrees. Select source band to be exported (e.g. coherence, or shadow-layover-mask).
- Write: Specify output path to respective folder (for each subset region). Output format can be specified under "save as", and formats include GeoTIFF (would lose metadata) or SNAP's own format (.dim) which includes the metadata.

Example results:

AOI_V= [83 880,86 330,6 463 000,6 465 000] -245x200 pixel (UTM 32N WGS1984)

→ POLYGON ((7.930165767669678 58.11180877685547, 7.967902183532715 58.11180877685547, 7.967902183532715 58.13193130493164, 7.930165767669678 58.13193130493164, 7.930165767669678 58.11180877685547, 7.930165767669678 58.11180877685547))

AOI_E= [91 220, 93850, 6 465 100,6 468 000] - 263x290 pixel (UTM 32N WGS1984)

→ POLYGON ((8.050148963928223 58.13736343383789, 8.089455604553223 58.13736343383789, 8.089455604553223 58.1632194519043, 8.050148963928223 58.1632194519043, 8.050148963928223 58.13736343383789, 8.050148963928223 58.13736343383789))

