



Norges miljø- og
biovitenskapelige
universitet

Masteroppgave 2022 30 stp
Fakultetet for realfag og teknologi

Pikselbasert klassifisering og skyggedeteksjon av urbane områder ved bruk av hyperspektrale bilder og LiDAR

**Semantic Segmentation and Shadow Detection of
Urban Environment by Using Hyperspectral Imaging
and LiDAR**

Vinith Balasingam
Geomatikk

Forord

Med denne masteroppgaven avslutter jeg mitt femårige integrerte masterløp ved Norges miljø- og biovitenskapelige universitet, NMBU. I løpet av disse fem årene har jeg tilegnet meg verdifull kunnskap, som jeg vil ha stor nytte av i fremtiden. Det må dermed rettes en stor takk til alle flinke forelesere som har hjulpet meg på veien.

Jeg vil rette en spesielt stor takk til min hovedveileder, professor Ingunn Burud, som har gitt meg muligheten til å ta del i dette spennende prosjektet. Hovedveilederen har gjennom hele perioden vært svært tilgjengelig, og bidratt med både gode veiledningstimer og faglige diskusjoner.

Jeg vil også benytte denne muligheten til å trekke frem min biveileder, Agnieszka Kuras, som har bidratt med konkrete tilbakemeldinger og gode innspill gjennom hele masterskrivingsprosessen. Hun har også bistått med datasettet, som har hatt stor betydning for utforming av masteroppgaven.

Studietiden på NMBU har vært preget av kameratskap og samhandling på tvers av fagretninger. Derfor ønsker jeg å takke mine medstudenter, som har vært positive gjennom arbeidet med masteroppgaven og motivert meg til å yte mitt beste. Til slutt vil jeg takke venner og familie, som har vært viktige støttespillere gjennom hele prosessen, og bidratt med både rådgivning og korrekturlesing.

Sammendrag

Urbane områder består av ulike typer vegetasjon, komplekse materialer og strukturell informasjon, som kan innhentes ved hjelp av fjernmåling. En av de største utfordringene med fjernmålingsdata er identifisering av skygge, ettersom skygge kan medføre forvirring og misvisende resultater. Skyggedeteksjon er nødvendig for å øke kvalitetsnivået på bildesegmentering og for å kunne skille mellom for eksempel "veier" og "ikke-veier", og "asfalt" og "ikke-asfalt". Skyggedeteksjon basert på maskinlæringsalgoritmer kan redusere graden av feil i predikeringen, og kan dermed benyttes for å oppnå bedre resultater.

I 2019 ble det innhentet hyperspektrale (HS) og LiDAR data fra urbane områder i Bærum kommune. I denne masteroppgaven blir det lagt vekt på skyggedeteksjon og pikselbasert klassifisering, basert på sammenslåing av HS (Hypex VNIR-1800 og SWIR-384) og LiDAR (Riegl VQ-1560i). For å kunne detektere skygge, ble det laget skyggemasker av SWIR og LiDAR, som videre ble brukt til å trene en U-net arkitektur. U-net arkitekturen ble benyttet til å predikere skyggemasker og skyggeklasser for høyoppløselige bilder som for eksempel VNIR. Ved hjelp av genererte skyggemasker, ble skygge innført som en klasse for pikselbasert klassifisering med en U-net arkitektur. Modellen ble trent opp til å kunne predikere fem ulike elementer: "veier", "bygninger", "skygge på vei", "annen skygge" og "alt annet". Innføring av skyggeklasser førte til økt nøyaktighet ved predikering av veier og bygninger. Det ble oppnådd en nøyaktighet på 61% for predikering av veier og 77% for predikering av bygninger. Modellen ble evaluert til en "f1-score" lik 0.81 og "Jaccard score" lik 0.69.

Abstract

Urban regions consist of different types of vegetation, complex materials and structure information, which can be collected by remote sensing. One of the biggest problems within remote sensed data is the appearance of shadows, as it may lead to confusion and misleading results. Shadow detection is required to increase the performance level of the image segmentation, and to distinguish between for example "roads" and "non-roads", and "asphalt" and "shadow on asphalt". Shadow detection based on deep learning algorithms can reduce the impacts of the errors and contribute to produce a better result.

In 2019, hyperspectral (HS) and LiDAR data from urban areas in Bærum municipality in Norway was acquired. In this study, shadow detection and semantic segmentation were carried out based on the fused HS (Hypex VNIR-1800 and SWIR-384) and LiDAR (Riegl VQ-1560i). In order to detect shadow, a shadow mask produced from SWIR and LiDAR, was used to train a U-net architecture. The model was successfully used to predict shadow masks and shadow classes for higher resolution images such as VNIR. With the generated shadow masks, shadow was identified as a class for semantic segmentation with a U-net architecture. The model was trained to predict five different elements: "roads", "building", "shadow on roads", "shadow on everything else" and "everything else". As a result of introducing shadow classes, the classification accuracy increased for roads and buildings. The prediction of roads and building reached an accuracy of 61% and 77%. The model achieved an "f1-score" of 0.81 and a "Jaccard score" of 0.69.

Innhold

1	Introduksjon	14
1.1	Bakgrunn	14
1.2	Prosjektets formål	15
1.3	Forskningsspørsmål	15
1.4	Oppsett og struktur	16
2	Teori	17
2.1	Elektromagnetisk stråling	17
2.2	Fjernmåling	19
2.2.1	Atmosfæriske effekter	20
2.2.2	Radians og reflektans	20
2.2.3	Spektralsignatur	21
2.2.4	Skygge	21
2.2.5	Flybåren fjernmåling	22
2.2.6	Felles kartdatabase	23
2.3	Hyperspektrale bilder	23
2.3.1	Pushbroom-skanner	24
2.4	LiDAR data	25

2.4.1	Digitale høydemodeller	27
2.5	Maskinl�ring	28
2.5.1	Kunstige nevrale nettverk	29
2.5.2	Konvolusjonelle nevrale nettverk	30
2.5.3	Fullstendig konvolusjonelle nettverk	35
2.5.4	Parameterm�ling av modellen	37
3	Metode	39
3.1	Datainnsamling	39
3.1.1	Sensorspesifikasjoner	40
3.1.2	Prosjektomr�de	42
3.2	Maskinvare og programvare	43
3.3	Databehandling	44
3.3.1	Datasammensl�ing	44
3.3.2	Skyggemaske	46
3.4	Arbeidsflyt	47
3.5	Eksperimenter	48
3.6	Generering av treningsdata	49
3.6.1	Treningsdata for predikering av skyggemaske	50
3.6.2	Treningsdata for pikselbasert klassifisering	50
3.7	Oppsett av fullstendig konvolusjonelle nettverk	52
3.7.1	Trening av nettverk	52
4	Resultater	54
4.1	Trening av modellen	54

4.1.1	Trening av modellen for predikering av skyggemaske	55
4.1.2	Trening av modellen for pikselbasert predikering	56
4.2	Prediksjon av skyggemaske	58
4.3	Pikselbasert klassifisering	61
4.3.1	Klassifisering av veier og bygninger	62
4.3.2	Klassifisering av veier, bygninger og skygge	64
4.3.3	Sammenligning av resultater	67
5	Diskusjon	68
5.1	Datasettet	68
5.1.1	Begrensninger med Felles kartdatabase	69
5.2	Analyse av resultatene	70
5.3	Tidligere forskning	73
5.4	Forslag til videre arbeid	75
6	Konklusjon	76
A	Vedlegg	77
A.0.1	Predikering av skyggemaske for 70cm oppløsning	77

Figurer

2.1	En elektromagnetisk bølge bestående av et elektrisk og magnetisk felt som står vinkelrett på hverandre, i retning av lysets hastighet.	17
2.2	Elektromagnetisk spekter ordnet fra korte til lange bølgelengder. Synlig lys består av området $400nm$ til $750nm$, VNIR($400-1000nm$) og SWIR($1000-2500nm$). Figuren hentet fra (Norheim 2020),(CC BY-SA 4.0).	18
2.3	Transmisjon av EM-stråling gjennom atmosfæren som treffer et tre. Strålingen som blir reflektert av treet fanges opp av et kamera.	19
2.4	Atmosfærens transmisjon for ulike bølgelengder. Lave transmisjonsverdier indikerer tett atmosfære og høye verdier indikerer at atmosfæren er gjennomsiktig. Figuren er hentet fra GMBB300 forelesning 4, H21.	20
2.5	Figuren viser reflektans for et materiale i de ulike bølgelengdeområdene for VNIR og SWIR. Spektralsignaturen for tre ulike materialer er gitt ved: rødt betongtak, asfaltert vei og vegetasjon.	21
2.6	Spektralsignaturen for asfaltert vei med og uten skygge. Bildet viser en rød og lyseblå firkant som representerer et asfaltert område med og uten skygge. Grafen viser spektralsignaturen til begge områdene med de samme fargene.	22
2.7	Komponenter på et fly ved flybåren fjernmåling bestående av; kontroll og lagringsenhet, laserskanner, GPS, IMU og kamera. Figuren er inspirert av (Vosselman & Maas 2010, p.22)	22
2.8	Illustrasjon av de ulike FKB standardene. FKB-A data inneholder de mest detaljrike kartene og FKB-D er de minst nøyaktige. Figuren er hentet fra(Georange 2022).	23
2.9	Sammenligning av en hyperkube og et RGB-bilde. Figuren viser et eksempel på pikselinformasjonen fra en hyperkube og tilsvarende område for RGB-bildet. . . .	24

2.10	Metodikken ved en Pushbroom-skanner er illustrert. Flyretningen er definert langs y-retningen, avbildning vil foregå langs x-aksen og spektralkomponentene er gitt i λ -retning.	24
2.11	Figuren til venstre illustrer pulsens bredde på bakken og figuren til høyre illustrer området på bakken som dekkes av utsendt puls.	25
2.12	Illustrasjon av multiple returer av en utsendt puls. Eksemplet viser tre returer og pulsstyrken for hver retur.	26
2.13	En punktsky farget etter høyde for et område i dette prosjektet.	27
2.14	Digital terrengmodell(DTM) beskriver terrengets høydeinformasjon, og digital overflatemodell(DOM) beskriver i tillegg vegetasjon og menneskeskapte objekter. Figuren er hentet fra (Services 2017)	27
2.15	Tre kategorier innen maskinlæringsalgoritmer i henhold til annotert data(fasitdata). Kategoriene består av styrt-læring, forsterkningslæring og ikke-styrt maskinlæring. 28	28
2.16	Et eksempel på hvordan en undertilpasset, godt tilpasset og overtilpasset modell vil håndtere usette data. Modifisert figur hentet fra (Hoffman 2021).	29
2.17	Illustrasjon av ”multilayer perceptron, MLP”, et nevralt nettverk bestående av nevroner satt sammen i tre ulike lag. Inngangsverdier vil bevege seg gjennom nettverket og dette vil gi en prediksjon, \hat{y}	29
2.18	Et eksempel på en CNN-arkitektur bestående av av lag med konvolusjon, ”pooling” og FC. Figuren er hentet fra (Raschka & Mirjalili 2019, p. 542)	30
2.19	Et eksempel på en konvolusjonell operasjon. Inngangsverdi: 5x5, filter:3x3, stride: 1, padding:1. Operasjonen resulterer i lik dimensjon for utgangsbildet som inngangsbildet. Figuren er hentet fra Dumoulin & Visin (2016).	31
2.20	Et eksempel på en transponert konvolusjonell operasjon. Inngangsbildet: 4x4, Padding: 1 og setter inn 0 etter hvert element, som resulterer i et 9x9 bilde som multipliseres med en 2x2 filter med en stride: 2. Utgangsbildets dimensjoner er doblet til 8x8 sammenlignet med inngangsbildet. Figuren er hentet fra (Raschka & Mirjalili 2019, p. 648).	32
2.21	Et eksempel på maks og snitt ”pooling” operasjon. Illustrasjonen viser resultatet etter maks og snitt ”pooling”. Inngangsbilder på 4x4, filter på 2x2 og stride på (2,2) resulterer i et utgangsbilde på 4x4.	32
2.22	Grafisk fremstilling av sigmoid-funksjon.	34

2.23	Grafisk fremstilling av ReLU-funksjon.	34
2.24	U-net arkitekturen til O. Ronneberger. Arkitekturen består av to deler, nedskalering og oppskalering. Venstre del av arkitekturen er nedskaleringsdelen og høyre del er oppskaleringsdelen. Figur hentet fra (Ronneberger et al. 2015)	36
2.25	Et eksempel på en forvirringsmatrise.	37
2.26	Illustrasjon av ”Intersection over Union”. Snittet av prediksjonen og fasiten divideres med unionen av prediksjonen og fasiten.	38
3.1	Flyplan brukt for datainnsamling av Terratec. Figuren er hentet fra prosjektrapporten(10990) utlevert av Terratec.	40
3.2	Tildelt prosjektområde. 04, 05 og 07 representerer de ulike flystripene. De tre bildeflisene som er illustrert benyttes i denne studien.	42
3.3	Hyperkuber for VNIR og SWIR som tilhører område 04.	44
3.4	Ekstrahering av nDOM fra DOM/DTM. Høydemodellene er farget etter høyde ”Shaded relief” og representerer område 04.	45
3.5	Sammenslåing av VNIR, SWIR og nDOM som resulterer i én stor hyperkube. . .	45
3.6	Område 07 er gitt i figur a og skyggemasken produsert av Maximillian Brell for tilsvarende område er gitt i figur b.	46
3.7	Flytdiagrammet for dette arbeidet	47
3.8	Ulike bildekvadratstørrelser vises på fig b(32x32),c(64x64),d(128x128),e(256x256). De fire størrelsene er representert sammen med område 04 i fig a.	49
3.9	Et kakediagram for eksperiment 1 som viser andelen av de ulike klassene og en tabell som viser antall piksler per klasse i hvert bånd.	50
3.10	Et kakediagram for eksperiment 2 som viser andelen av de ulike klassene og en tabell som viser antall piksler per klasse i hvert bånd.	51
3.11	Et kakediagram for eksperiment 3 som viser andelen av de ulike klassene og en tabell som viser antall piksler per klasse i hvert bånd.	51

3.12	Illustrasjon av u-net arkitekturen som blir benyttet i denne studien. Figuren består av blokk A (inngangsbildet), B (prediksjonen), fire nedskaleringsblokker (grønn) og fire oppskaleringsblokker (rosa). Hver nedskaleringsblokk består av to konv-filtre, "dropout, maks-pooling og "skip connection", unntatt blokk 4 som ikke inneholder maks-pooling. Hver oppskaleringsblokk består av to konv-filtre, "dropout, opp-konv og sammenslåing fra "skip connection", og blokk 8 inneholder et ekstra lag med konv-filter.	52
4.1	Grafisk fremstilling av trening og validering av modellen ved bruk av optimaliseringsfunksjonene, Adam og SGD. 200 epoker blir utført fem ganger og gjennomsnittet av de fem testene presenteres for begge funksjonene.	55
4.2	Grafisk fremstilling av trening og validering av den optimaliserte modellen for predikering av skyggemaske	56
4.3	Grafisk fremstilling av trening og validering for modellen i de tre eksperimentene. E1 vises i a, E2 i b og E3 i c.	57
4.4	Prediksjon av skyggemaske med 30cm romlig oppløsning og RGB-visualisering av område 04.	58
4.5	Prediksjon av skyggemaske med 30cm romlig oppløsning og RGB-visualisering av område 05.	59
4.6	Prediksjon av skyggemaske med 30cm romlig oppløsning og RGB-visualisering av område 07.	59
4.7	Fordeling av pikselverdier i de tre skyggemaskene med 30cm romlig oppløsning, visualisert i form av histogram.	60
4.8	Genererte skyggeklasser for eksperiment 3, der klassen "skygge på vei" har blitt innført. Fig a viser skyggeklassene for område 04, fig b viser skyggeklassene for område 05, og fig c viser skyggeklassene for område 07	61
4.9	Forvirringsmatrise for E1 som består av tre klasser.	62
4.10	Modellens prediksjon av de tre klassene i E1 for område 07. Fig a viser fasitdataen, fig b viser RGB-bildet, fig c viser prediksjonen.	63
4.11	Forvirringsmatrise for E2 som består av fire klasser.	64
4.12	Modellens prediksjon av de fire klassene i E2 for område 07. Fig a viser fasitdataen, fig b viser RGB-bildet, fig c viser prediksjonen.	65

4.13	Forvirringsmatriser for E3 før og etter post-prosessering. Fig a viser forvirringsmatrisen med fem klasser, fig b viser forvirringsmatrisen med fire klasser etter post-prosessering.	66
4.14	Modellens prediksjon av de fem klassene i E3 for område 07. Fig a viser fasit-dataen, fig b viser RGB-bildet, fig c viser resultatet etter post-prosessering av prediksjonen.	66
4.15	Sammenligning av prediksjonen fra de tre eksperimentene. Fig a viser for E1, fig b for E2 og fig c for E3.	67
5.1	Predikering av et området bestående av blant annet asfaltert vei med og uten skygge. Illustrasjonen viser tre bilder; testbildet, fasiten og prediksjonen av testbildet.	68
5.2	Prediksjon av et testområde med feil i fasiten. De røde firkantene representerer et bygg på testbildet og prediksjonen. Bygget mangler i fasiten.	69
5.3	De røde firkantene viser det aktuelle bygget i RGB- og LiDAR-visning.	70
5.4	Et testbilde fra område 04 vises sammen med den predikerte skyggemasken på 30cm romlig oppløsning. Den røde sirkelen markerer en parkeringsplass som har blitt feilpredikert som skygge.	70
5.5	Sammenligning av predikert skyggemaske på 30cm romlig oppløsning og 70cm romlig oppløsning for det samme området.	71
5.6	Et testområde blir valgt ut til sammenligning av prediksjonen for de tre eksperimentene.	73
A.1	Skyggemaske for område 04 i 70cm oppløsning.	77
A.2	Skyggemaske for område 05 i 70cm oppløsning.	77
A.3	Genererte skyggeklasser for eksperiment 2	78

Tabeller

3.1	Tabell for HySpex senorene, spesifikasjonene er hentet fra (HySpex-NEO 2022 <i>b</i>) og (HySpex-NEO 2022 <i>a</i>)	41
3.2	Oversikt over de tre bildenes dimensjoner gitt som (bredde, lengde, spektralbånd).	42
3.3	Oversikt over maskinvarespesifikasjoner.	43
3.4	Oversikt over Python-pakker og versjon som ble benyttet.	43
3.5	Dimensjonene til de ulike oppdelingene av treningsdata for pikselbasert klassifisering er gitt som (antall bildekvadrater, kvadratstørrelse i bredde, kvadratstørrelse i lengde og antall bånd).	50
3.6	Tabell for E1 som viser antall piksler per klasse i hvert bånd.	50
3.7	Tabell for E2 som viser antall piksler per klasse i hvert bånd.	51
3.8	Tabell for E3 som viser antall piksler per klasse i hvert bånd.	51
4.1	Test av de ulike modellkonfigurasjonene. Tabellen viser evaluering av trenings- og valideringssettet. Det blir målt tap, ”Jaccard score”, og ”F1 score” for hver modellkonfigurasjon.	54
4.2	Evaluerings av skyggeprediksjon ved bruk av optimaliseringsfunksjonene Adam og SGD. Snittet av 5 gjennomføringer med hver funksjon i 200 epoker presenteres.	55
4.3	Modellevaluering på testdataen for predikering av skyggemasker.	56
4.4	Modellevaluering av prediksjonen på område 07 for de tre eksperimentene. Prediksjonen av område 07 er evaluert etter ”Jaccard score” og ”F1-score”.	62

Forkortelser

CNN Convolutional Neural Network

DOM Digital overflatemodell

DTM Digital terrengmodell

FC Fully connected

FCN Fully convolutional Network

GPS Global Positioning System

IMU Inertial measurement unit

LiDAR Light Detection and Ranging

nDOM Normalisert digital overflatemodell

RGB Red green blue

SWIR Short-wavelength infrared

VNIR Visible and near-infrared

Kapittel 1: Introduksjon

1.1 Bakgrunn

Samfunnets infrastruktur utvikles i takt med den teknologiske tidsalderen. Infrastrukturutbyggingen i urbane områder er under stadig utvikling og påvirker dermed kartdataens kvalitet. Det er derfor sentralt å utforske nye løsninger og metoder for å holde kartdataen kontinuerlig oppdatert. Tradisjonelle metoder som for eksempel feltinspeksjon og manuelt arbeid, regnes som både kostbare og tidkrevende prosesser (Shafri et al. 2012). Ved å benytte flybåren fjernmåling med hyperpektral og LiDAR-sensor, kan informasjon om et urbant område innhentes på en effektiv måte (Skogseth & Norberg 2014, p.290).

Utviklingen av sensorteknologi har økt tilgjengeligheten av hyperspektral data fra urbane områder som består av høy romlig og spektral oppløsning (Liao et al. 2012). Hyperspektrale bilder består av hundrevis av bånd som beskriver de fysiske og kjemiske egenskapene til et materiale. Grunnet de karakteristiske egenskapene for hvert materiale, vil en sensor fange opp dette i de ulike båndene (Franke et al. 2009). En LiDAR-sensor kan benyttes parallelt for å innhente geometrisk informasjon om de ulike objektene som fotograferes. En fusjon av hyperspektrale og LiDAR data, kan gi detaljrik informasjon ved innhenting av data fra det urbane området. Ved å utnytte informasjonen fra den fusjonerte dataen, kan hustak-materialer gjenkjennes, og svakheter som sprekker og hull i veinettet identifiseres (Shafri et al. 2012).

En utfordring ved flybåren fotografering, er skygge. Skygge oppstår ved at ulike objekter, som skyer og urbane materialer, blokkerer for lyskilden. I de fleste tilfeller er lyskilden solen. Dette fører til tap av informasjon i områdene med skygge (Zhou et al. 2021). Skygge vil påvirke graden av nøyaktighet ved ekstrahering av informasjon fra bildene (Shahtahmassebi et al. 2013).

Bruk av datavitenskap og maskinlæring kan gi gode resultater ved klassifisering av fjernmålingsdata (Zhang et al. 2001). Årsaken til dette er at maskinlæring inkluderer en rekke algoritmer som kan brukes til å løse komplekse utfordringer knyttet til store mengder data (Lary et al. 2016). Høy romlig og spektral oppløsning, kan brukes til å detektere og segregere ulike materialer, som for eksempel bygninger og veier, i et urbant område (Jensen 2007, p. 445). Maskinlæring kan benyttes til å kartlegge disse objektene fra hyperspektrale bilder. Konvolusjonelle nevrale nettverk har oppnådd gode resultater ved klassifisering av fjernmålingsdata (Zhang et al. 2019). I denne studien benyttes det dype konvolusjonelle nevrale nettverket, U-net, for pikselbasert predikering.

1.2 Prosjektets formål

Denne masteroppgaven er en del av et større prosjekt, ”FKB Maskinlæring”, ledet av Bærum kommune. Formålet med prosjektet ”FKB Maskinlæring”, finansiert av ”Regionale forskningsfond”, er å utvikle nye metoder for automatisk oppdatering av kommunens kartdata. Ved hjelp av maskinlæring skal dermed prosessen med kartlegging av LiDAR og hyperspektrale data kunne automatiseres.

1.3 Forskningsspørsmål

I denne studien blir det satt fokus på skyggedeteksjon og klassifisering av asfalterte veier og bygninger ved bruk av hyperspektrale bilder og LiDAR. Hensikten med studien er å forske på hvorvidt skygge kan identifiseres, ved hjelp av skyggemasker som blir predikert av et konvolusjonelt nevralt nettverk. Videre blir det forsket på pikselbasert klassifisering av asfalterte veier, bygninger og skyggeklasser. Skyggeklassene blir generert av de predikerte skyggemaskene.

For å kunne avgjøre hvorvidt skyggedetektering og innføring av skyggeklasser vil bidra til å forbedre modellens prediksjonsevne, har følgende problemstilling blitt utarbeidet:

- Vil skyggedeteksjon føre til økt nøyaktighet ved pikselbasert klassifisering av veier og bygninger i urbane områder?

For å utforske denne problemstillingen nærmere, skal det gjennomføres tre eksperimenter. I eksperimentene skal det undersøkes om modellens predikeringsevne påvirkes av at det innføres flere skyggeklasser. Følgende eksperimenter skal utføres i denne masteroppgaven:

- **E1:** Klassifisering av veier, bygg og annet.
- **E2:** Innføring av skyggeklasse. Klassifisering av veier, bygg, skygge og annet.
- **E3:** Innføring av to skyggeklasser. Klassifisering av veier, bygg, skygge på veier, annen skygge og annet.

1.4 Oppsett og struktur

Oppgaven er strukturert i seks kapitler, inkludert introduksjon. I tillegg er det gitt vedlegg. Hvert kapittel beskriver ulike deler av oppgaven. For å veilede leseren gjennom oppgaven, er det gitt en kort oppsummering av hvert kapittel.

Kapittel 2 gir en oversikt over grunnleggende teori og underliggende konsepter som er sentrale for oppgaven. Teorien er fundamental for innholdet i resten av oppgaven.

Kapittel 3 presenterer metodene i studien. Denne delen inneholder metodene som ble brukt i form av datainnsamling, maskin- og programvare, eksperimenter og databehandling.

Kapittel 4 presenterer først resultater fra trening av nettverk. Deretter blir det gitt resultater for prediksjon av skyggemasker og pikselbasert klassifisering for de ulike eksperimentene.

Kapittel 5 diskuterer resultatene som blir presentert i kapittel 4, i henhold til den utarbeidede problemstillingen. I tillegg blir det gitt forslag til videre arbeid.

Kapittel 6 gir en konklusjon basert på oppsummering av hovedaspektene i oppgaven.

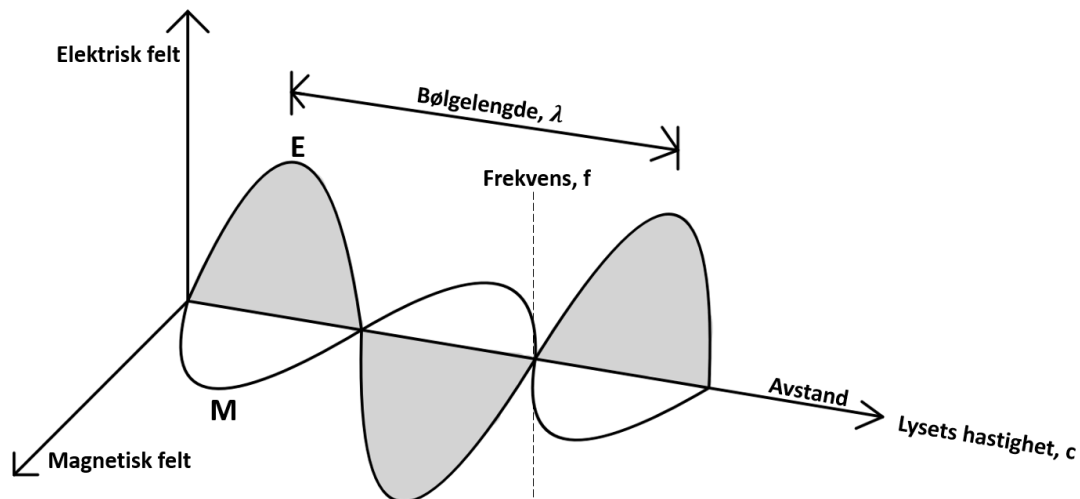
Vedlegg

De fleste figurer i masteroppgaven er selvproduserte. Figurer som ikke er selvproduserte, blir referert i figurtekstene og i kildelisten. Python-kodene som er mest sentrale i denne masteroppgaven er gitt i Github, <https://github.com/M30-GEOM-VB>.

Kapittel 2: Teori

2.1 Elektromagnetisk stråling

Elektromagnetisk stråling kan beskrives i form av partikler og bølger. Elektromagnetiske bølger forplanter seg gjennom rommet med lysets hastighet. Figur 2.1 viser en elektromagnetisk bølge som består av et elektrisk og et magnetisk felt. Figuren består av to ortogonale vektorer, som står vinkelrett på retningsvektoren. Retningsvektoren beskriver lysets hastighet, der c er gitt som en konstant lik $3 \times 10^8 \text{ m/s}$ (Jensen 2007, p. 38).



Figur 2.1: En elektromagnetisk bølge bestående av et elektrisk og magnetisk felt som står vinkelrett på hverandre, i retning av lysets hastighet.

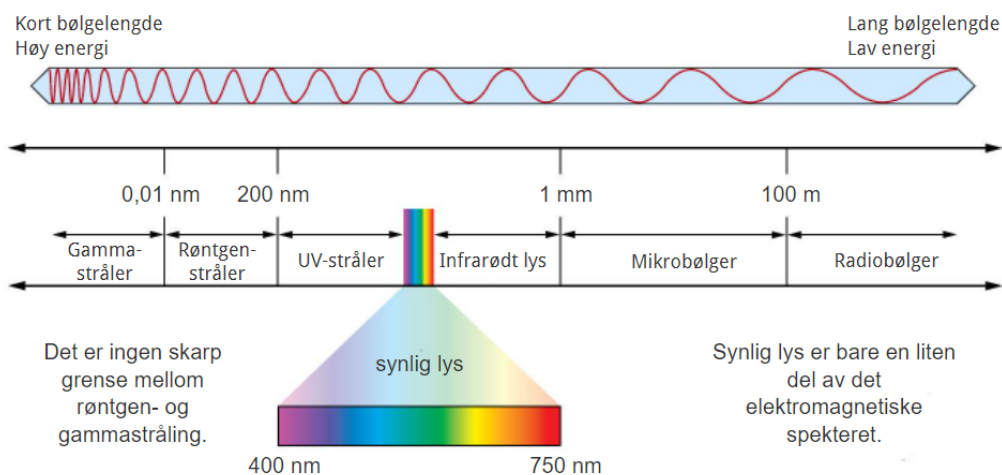
Elektromagnetisk stråling oppstår ved to måter; eksitasjon og absorpsjon. Når et elektron faller fra en høyere til en lavere energitilstand, blir energien sendt ut i form av eksitasjon. Absorpsjon er et fenomen som beskrives av at et elektron går fra en lavere energitilstand til en høyere energitilstand. Forholdet mellom bølglengde og frekvens i elektromagnetisk stråling, er gitt ved følgende formel:

$$c = f \times \lambda, \quad (2.1)$$

der c representerer forplantningshastighet, f angis som frekvens og λ er gitt som bølglengde. Frekvensen er omvendt proporsjonal med bølglengden og dette forholdet beskrives nærmere i figur 2.1 (Jensen 2007, p. 39).

Det elektromagnetiske spekteret utgjør et område ordnet fra korte bølgelengder (høy frekvens og høy energi) til lange bølgelengder (lav frekvens og lav energi). I det elektromagnetiske spekteret vil menneskets øye kun klare å registrere synlig lys. Synlig lys betegnes som området med bølgelengder fra omtrent 400nm til 750nm , bestående av tre overlappende bånd (RGB-bånd). De tre båndene utgjør fargene rød, grønn og blå (Ravikanth et al. (2017)).

I tillegg til synlig lys finnes det andre strålingstyper med både kortere og lengre bølgelengder i det elektromagnetiske spekteret. I denne masteroppgaven er følgende områder i det elektromagnetiske spekteret sentrale for fjernmåling; synlig- og nærinfrarødt område (VNIR) som spenner seg fra 400nm til 1000nm , og kortbølget infrarødt område (SWIR) som spenner seg fra 1000nm til 2500nm (Jenal et al. (2019)).



Figur 2.2: Elektromagnetisk spekter ordnet fra korte til lange bølgelengder. Synlig lys består av området 400nm til 750nm , VNIR($400\text{-}1000\text{nm}$) og SWIR($1000\text{-}2500\text{nm}$). Figuren hentet fra (Norheim 2020),(CC BY-SA 4.0).

Elektromagnetisk stråling kan også beskrives i form av fotonenergi. Når et elektron faller fra en eksitert tilstand til en lavere energitilstand, blir mengden energi utstrålt i form av energipakker. Disse energipakkene kalles fotoner. Forholdet mellom strålingsfrekvens og fotoner, kan uttrykkes slik:

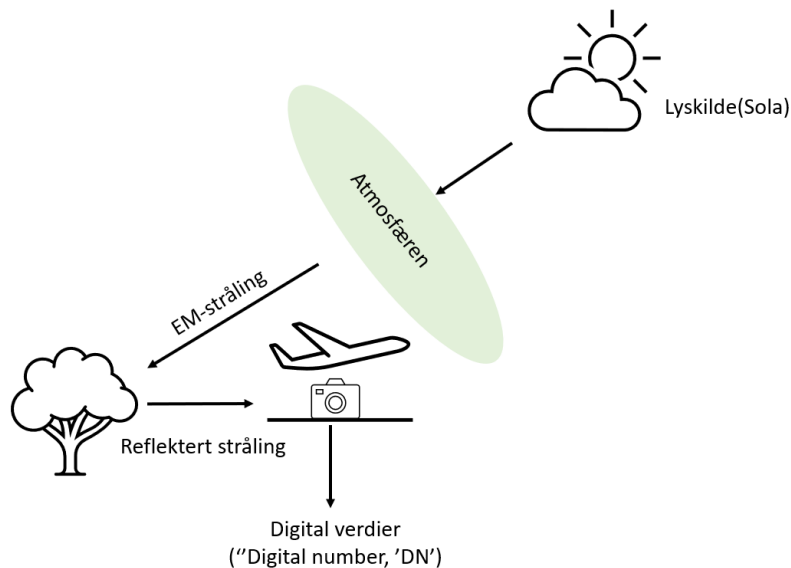
$$E = h \times f, \quad (2.2)$$

der E angir kvanteenergien, h er gitt som Plancks konstant (6.62610^{-34}Js) og f representerer strålingsfrekvens. Kvanteenergien og bølgelengden er omvendt proporsjonale i forhold til hverandre. Dette inverse forholdet er sentralt ved fjernmåling. Ved fjernmåling er det mer utfordrende å detektere langbølget energi i det infrarøde område, enn stråling med kortere bølgelengder i det synlige området (Jensen 2007, p. 45,46).

2.2 Fjernmåling

Fjernmåling er en metode som brukes for å samle inn informasjon om objekter på jordoverflaten (Birkeland (2021)). Ved fjernmåling, samles det inn informasjon fra sensorer uten direkte kontakt med objektet som observeres. Kamera og radar er to type sensorer som i stor grad anvendes innen fjernmåling. Kamera er en passiv sensor og LiDAR er en aktiv sensor. Ved å kombinere bruk av kamera og LiDAR, kan det samles inn data fra to typer sensorer med ulike egenskaper.

Innen fjernmåling brukes det en rekke sensorer til å samle inn informasjon. Disse instrumentene er egnet for ulike bølgelengder og dekker forskjellige deler av det elektromagnetiske spekteret. Ved fjernmåling, er det sentralt å skille mellom passive og aktive sensorer (Schowengerdt 2006, p.10). Figur 2.3 viser prosessen for en passiv sensor og hva som påvirker verdiene i bildefilene som blir tatt opp. Sollyset føres gjennom atmosfæren og treffer et materiale. Materialet reflekterer deler av lyset, som videre tas opp av sensoren. Lyset som tas opp er en skalert versjon av reflektans, og bildefilen består av digitale verdier ("DN - Digital number").



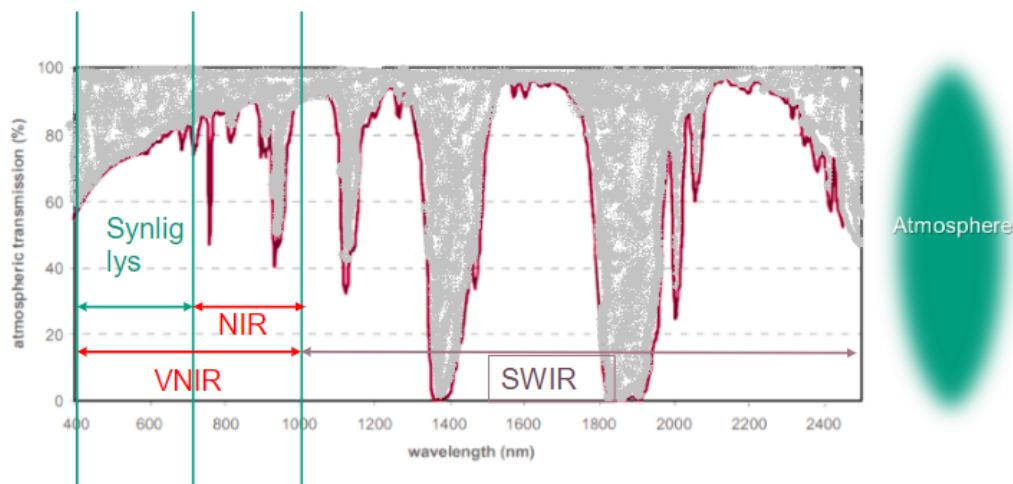
Figur 2.3: Transmisjon av EM-stråling gjennom atmosfæren som treffer et tre. Strålingen som blir reflektert av treet fanges opp av et kamera.

Den aktive sensoren har sin egen lyskilde og krever derfor en mer kompleks prosess. I en aktiv sensor blir strålingen først generert av instrumentet, så sendt ned til jordoverflaten, og deretter reflektert tilbake til sensoren. Det mottatte signalet og den nødvendige informasjonen blir til slutt trukket ut og behandlet (Cracknell 2007, p.22). En aktiv sensor kan i teorien benyttes året rundt ettersom sensoren har sin egen lyskilde. Likevel vil nedbør som regn og snø gi refleksjon av de utsendte pulsene. Dette kan føre til støy og dårligere datakvalitet. Sterk vind kan gi skjev punktfordeling på bakken, noe som kan føre til mindre nøyaktighet (Skogseth & Norberg 2014).

2.2.1 Atmosfæriske effekter

Når lyset føres gjennom atmosfæren blir strålingen påvirket av refraksjon, spredning, absorpsjon og refleksjon. Refraksjon kan oppstå ved at lyset brytes av luft eller vann, slik at lysets retning endres. Spredning og absorpsjon oppstår ved at lyset som føres gjennom atmosfæren blir påvirket av atmosfæriske partikler (Jensen 2007).

Spredning er en viktig faktor å ta hensyn til ved fjernmåling. Spredningen kan føre til tap av informasjon ettersom strålingens retning endres. Dette kan føre til at kontrasten i et bilde reduseres, slik at det kan være utfordrende å skille objekter på bildet fra hverandre. Atmosfæren er kun delvis gjennomsiktig for EM-stråling på grunn av absorpsjon (Jensen 2007). Figur 2.4 viser atmosfærisk transmisjon i prosent. Lave transmisjonsverdier indikerer tett atmosfære og høye verdier indikerer at atmosfæren er gjennomsiktig. Som figuren viser, blir enkelte bølgelengder i SWIR fullstendig blokkert for transmisjon.



Figur 2.4: Atmosfærens transmisjon for ulike bølgelengder. Lave transmisjonsverdier indikerer tett atmosfære og høye verdier indikerer at atmosfæren er gjennomsiktig. Figuren er hentet fra GMBB300 forelesning 4, H21.

2.2.2 Radians og reflektans

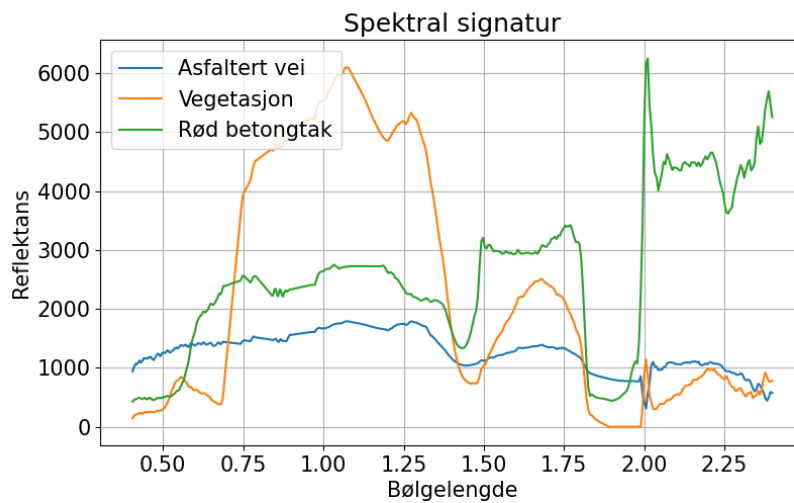
Radians måles direkte av fjernmålingssensorer, og beskriver mengden lys fra det observerte objektet som fanges opp av instrumentet. Lyset føres gjennom atmosfæren, og det skapes dermed atmosfæriske effekter i bildet når lyset tas opp av instrumentet (Ray (2013)). Radians omfatter stråling reflektert fra overflaten. Strålingen blir sendt av nabopiksler og reflektert av skyer over pikselområdet. Radians blir påvirket av strålingskilden. For optiske bilder regnes solen som strålingskilde. Spektrumet til en radianspiksel vil altså ha en fasade lik solspekteret, med en topp på omtrent $500nm$ (Shippert (2013a)).

Signalet tas opp av detektoren ("DN-number"), som så kalibreres til et radiansbilde. Grunnet atmosfæriske effekter vil radiansbildet inneholde støy som for eksempel reflektert sollys fra nabopiksler. Radiansbildet blir atmosfærisk korrigert og omgjort til et reflektansbilde. Reflektansbildet kan da benyttes videre til å trekke ut kvantitativ informasjon, og til å beskrive funksjonene på overflaten. Reflektans beskriver altså forholdet mellom mengden lys som forlater

et objekt og mengden lys som treffer objektet (Shippert (2013a)).

2.2.3 Spektralsignatur

Når elektromagnetisk stråling kommer i kontakt med et materiale, kan strålingen bli absorbert, reflektert eller emittert. Fjernmåling er hovedsakelig basert på deteksjon av reflektert og emittert elektromagnetisk stråling. I enkelte tilfeller er det mulig å identifisere materialer basert på spektralsignaturen (Schowengerdt 2006, p.13). Hvert materiale har sin egen spektralsignatur med bestemte emisjons- og refleksjonsegenskaper, som gjør det mulig å skille objekter fra hverandre. Ved fjernmåling vil det være mulig å innhente denne spektralsignaturen (Khorram et al. 2012, p.7). Figur 2.5 viser et eksempel på spektralsignaturen til tre ulike materialer; rødt betongtak, asfaltert vei og vegetasjon.

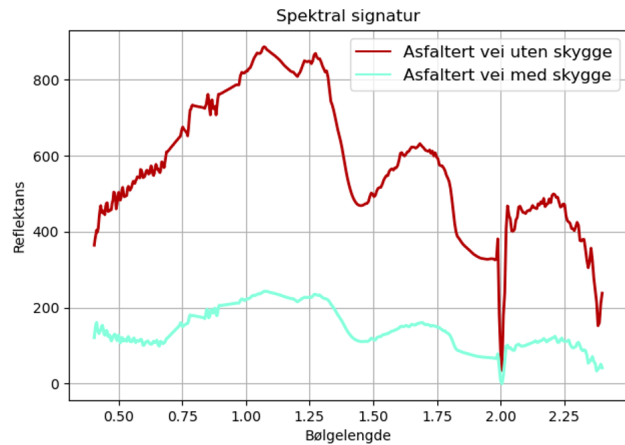


Figur 2.5: Figuren viser reflektans for et materiale i de ulike bølgelengdeområdene for VNIR og SWIR. Spektralsignaturen for tre ulike materialer er gitt ved: rødt betongtak, asfaltert vei og vegetasjon.

2.2.4 Skygge

Data innsamlet ved fjernmåling i urbane områder, kan inneholde skygge som fremkommer av ulike objekter som for eksempel bygninger, trær og andre urbane objekter. Skygge oppstår når et objekt blokkerer et område for lys. Ved passive sensorer brukes sollys som lyskilde i de fleste tilfeller. Et objekt som blokkerer for sollyset vil gi skyggeeffekter på bildet, i området der det blir blokkert for sollys (Shahtahmassebi et al. 2013).

Spektralsignaturen for et materiale vil være ulik avhengig av om området er dekket av skygge eller ikke (Yamazaki et al. 2009). Figur 2.6 viser spektralsignaturen for et område av asfaltert vei. Grafen viser spektralsignaturen til asfaltert vei som ikke blir blokkert for sollys og asfaltert vei dekket av skygge, der et tre blokkerer for tilgang til sollys. Grunnet karakteristiske endringer på områder med og uten skygge, blir skygge et problem. Nøyaktigheten ved klassifisering av objekter vil da også kunne påvirkes (Shahtahmassebi et al. 2013).

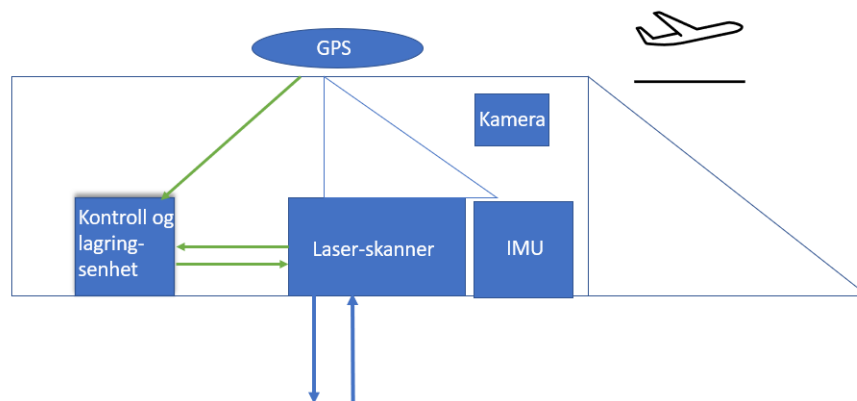


Figur 2.6: Spektralsignaturen for asfaltert vei med og uten skygge. Bildet viser en rød og lyseblå firkant som representerer et asfaltert område med og uten skygge. Grafen viser spektralsignaturen til begge områdene med de samme fargene.

2.2.5 Flybåren fjernmåling

Ved flybåren fjernmåling brukes sensorer montert på flyet, i tillegg til instrumenter som GPS og IMU. GPS og IMU brukes til å beregne sensorens presise posisjon og nøyaktige orientering. Ved å kombinere fjernmålingssensorer med GPS og IMU, vil georefererte data av området bli kartlagt og skannet med høy grad av nøyaktighet (Vosselman & Maas 2010, p.21- 23).

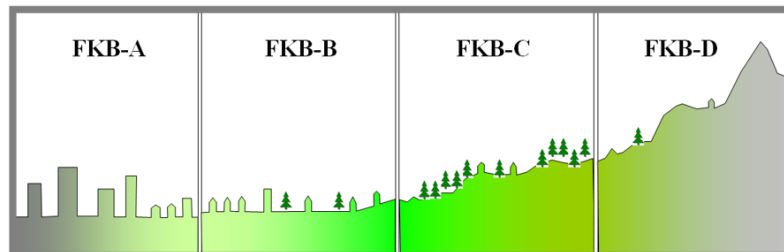
Figur 2.7 viser et eksempel på hvordan de ulike komponentene er satt sammen ved flybåren fjernmåling. Komponentene består av; en laserskanner, GPS og IMU som beregner posisjon og orientering, og en kontroll- og datalagringsenhet. Kontroll- og datalagringsenheten har kontroll over hele systemet og ansvaret for lagring av data fra GPS, IMU og sensorer. Et kamera kan også legges til systemet, slik at laserskanning og avbildning kan skje samtidig.



Figur 2.7: Komponenter på et fly ved flybåren fjernmåling bestående av; kontroll og lagringsenhet, laserskanner, GPS, IMU og kamera. Figuren er inspirert av (Vosselman & Maas 2010, p.22)

2.2.6 Felles kartdatabase

Felles kartdatabase representerer en rekke detaljrike datasett som driftes og vedlikeholdes av Geovekst-partnerne (Berg 2020). FKB inneholder en rekke datasett, og i denne studien blir data for vei og bygninger benyttet. I FKB finnes det fire ulike standarder; FKB-A, FKB-B, FKB-C og FKB-D, der FKB-A data inneholder de mest detaljrike kartene. Standarden på dataen varierer med kost og nytte. Sentrale byområder vil være kartlagt etter FKB-A standarden (Geonorge 2022). Figur 2.8 viser overgangen mellom de ulike standardene.



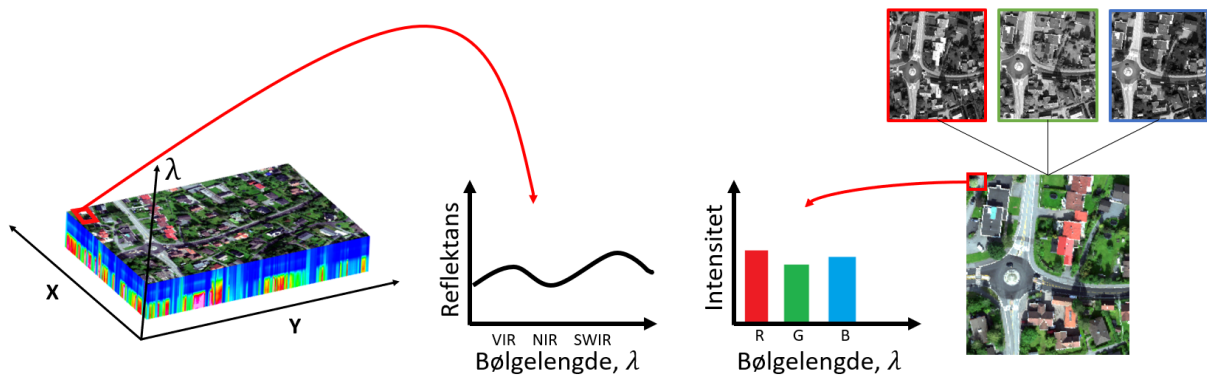
Figur 2.8: Illustrasjon av de ulike FKB standardene. FKB-A data inneholder de mest detaljrike kartene og FKB-D er de minst nøyaktige. Figuren er hentet fra (Geonorge 2022).

2.3 Hyperspektrale bilder

En utfordring med bruk av konvensjonelle kameraer er at kamerafiltrene kun består av RGB-bånd, fordelt på tre kanaler. Det vil altså kun bli registrert stråling fra det synlige området, og betydningsfulle detaljer fra det elektromagnetiske spekteret vil dermed kunne gå tapt. En mulighet for å unngå dette problemet er å ta i bruk hyperspektrale bilder (Ozdemir & Polat (2020)).

Hyperspektrale bilder oppstår ved registrering av hundrevis av bånd fra det elektromagnetiske spekteret. Et standard RGB-bilde består av tre komponenter, i motsetning til hyperspektrale bilder som er bygd opp av hundrevis av komponenter. Ved et hyperspektralt bilde oppstår en datakub (”hypercube”) som inneholder spektralinformasjon (Ozdemir & Polat (2020)). Figur 2.9 viser en sammenligning av hyperspektrale bilder og RGB-bilder. X- og Y-aksene i den hyperspektrale datakuben representerer oppløsningen til bildet, mens Z-aksen gitt ved λ (bølgelengde), representerer antall bånd i bildet.

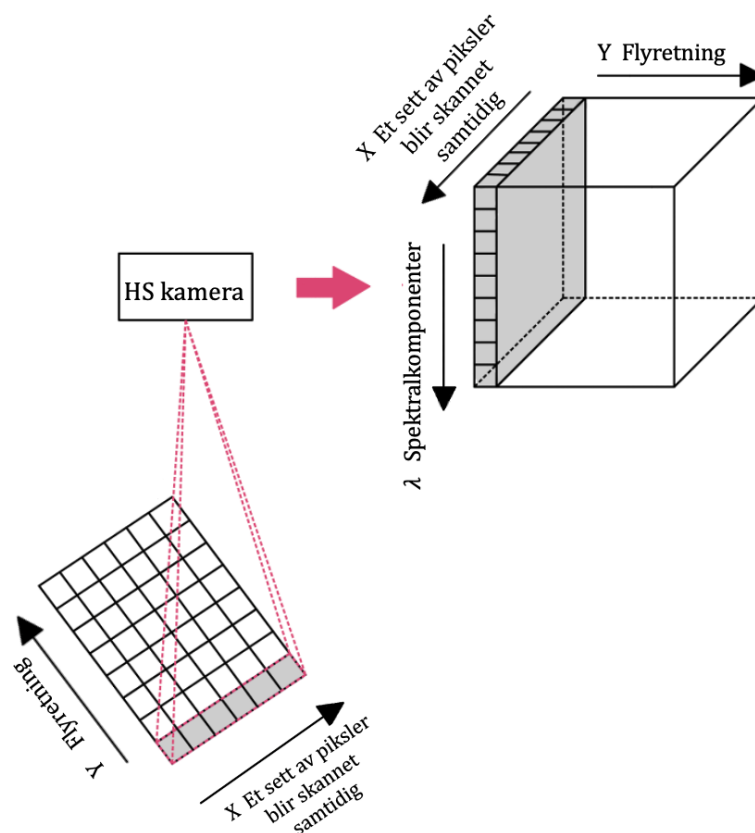
Hver piksel i det hyperspektrale bildet inneholder ”n”-antall bånd. I hvert bånd måles en verdi på refleksans for hver bølgelengde. En piksel kan beskrives som en vektor gitt ved $[1,1,\text{spektralbånd}]$. Pikselvektoren uttrykker da en spektralsignatur for hver enkel piksel i en hyperkub. Spektralsignaturen til en utvalgt piksel består av informasjon som karakteriserer innholdet i pikselen. Denne informasjonen kan benyttes til å skille mellom ulike signaturer (Ozdemir & Polat (2020)).



Figur 2.9: Sammenligning av en hyperkube og et RGB-bilde. Figuren viser et eksempel på pikselinformasjonen fra en hyperkube og tilsvarende område for RGB-bildet.

2.3.1 Pushbroom-skanner

I denne oppgaven har sensoren HySpex blitt benyttet for hyperspektral avbildning. HySpex er en type pushbroom-skanner fra NEO. En pushbroom-skanner består av en "linear array sensor" som vil ta opptak av reflektert eller emittert stråling fra objektet som fotografers, (Jensen 2007). Avbildning av området vil foregå langs en stripe på tvers av flyretningen. Hver piksel som blir fotografert vil inneholde spektralkomponenter. Figur 2.10 viser et eksempel på metodikken ved en Pushbroom-skanner, der flyretningen er definert langs y-retningen, avbildning vil foregå langs x-aksen og spektralkomponentene i λ -retning. (Shippert (2013b)).

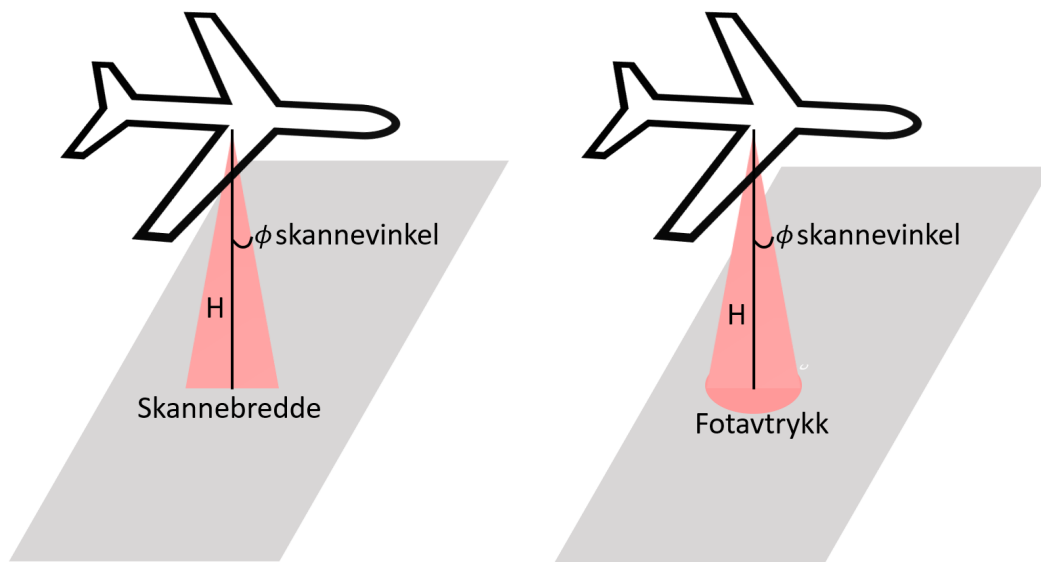


Figur 2.10: Metodikken ved en Pushbroom-skanner er illustrert. Flyretningen er definert langs y-retningen, avbildning vil foregå langs x-aksen og spektralkomponentene er gitt i λ -retning.

2.4 LiDAR data

LiDAR, "Light Detection and Ranging", er en fjernmålingsteknikk som brukes i flybåren laserskanning, for å samle informasjon om jordoverflaten. LiDAR benyttes ved at det sendes ut høyfrekvente laserpulser fra et fly, for å kunne utføre avstandsberegninger og samle inn materialenes intensitetsverdier (Vosselman & Maas 2010, p.14). En LiDAR-skanner for topografisk kartlegging sender ut høyfrekvente laserpulser i den nær-infrarøde delen av det elektromagnetiske spekteret. Delen av det elektromagnetiske spekteret som består av bølgelengder fra $800nm$ til $1550nm$, betegnes som nær-infrarødt område (Vosselman & Maas 2010, p. 25).

Laserpulsene som sendes ut av sensoren vil ha en skannebredde og et fotavtrykk på bakken som vil variere med flyhøyden og sensorens åpningsvinkel. Skannebredden er bredden på bakken som blir skannet på tvers av fly-retningen. Fotavtrykket er området på bakken som dekkes av utsendt puls (Baltsavias 1999). Figur 2.11 viser et eksempel på hvordan skannebredde og fotavtrykk vil se ut.



Figur 2.11: Figuren til venstre illustrerer pulsens bredde på bakken og figuren til høyre illustrerer området på bakken som dekkes av utsendt puls.

En laserskanner sender ut tusenvis av pulser i sekundet nedover mot bakken. Pulsene treffer et objekt og blir reflektert tilbake til sensoren. Materialet til objektet som blir truffet, avgjør hvor stor andel av pulsene som blir reflektert tilbake. Sensoren vil lagre informasjon om tidsforsinkelsen fra da pulsene ble sendt til de ble mottatt. Dermed kan avstanden ned til objektet beregnes. Ved en slik måling av de reflekterte pulsene, kan informasjonen om det truffede objektet behandles ytterligere (Skogseth & Norberg 2014, p. 306, 307).

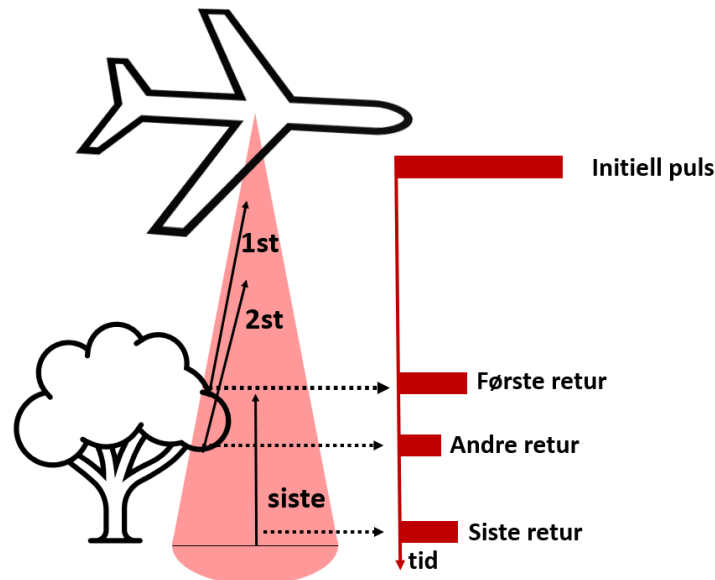
Likningen under viser formelen for beregning av avstanden fra flyet til objektet. Denne avstanden betegnet som d , c står for lysets hastighet og t betegnes som tiden fra da pulsen ble sendt til den ble mottatt.

$$d = \frac{c}{2} \times t \quad (2.3)$$

Multiple returer

Multiple returer er viktige egenskaper ved LiDAR. Det at sensorer klarer å registrere flere returer fra en utsendt puls, betegnes som multiple returer. Antall returer avhenger av materialet som blir truffet av pulsen (Vosselman & Maas 2010, p. 28). Den utsendte pulsen vil gi et fotavtrykk på bakken. Materialets overflate innenfor fotavtrykket avgjør hvor stor del av pulsen som blir reflektert tilbake til sensoren, mens resten vil trenge seg igjennom. (Skogseth & Norberg 2014, p. 307).

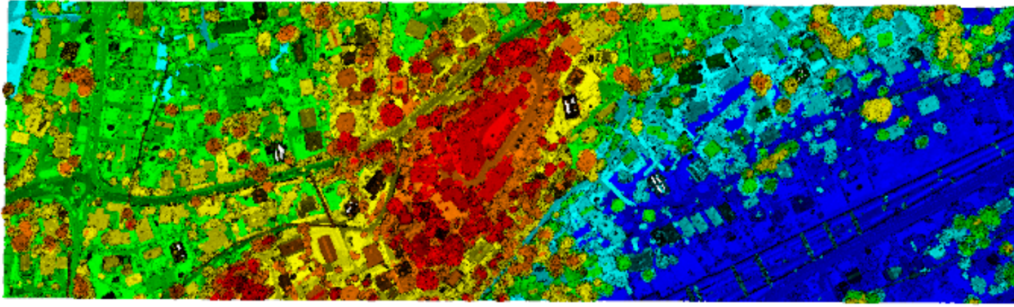
Et eksempel på multiple returer vises i figur 2.12. Når en puls treffer et tre, vil deler av pulsen bli reflektert tilbake. I eksempelet klarer pulsen å trenge ned til bakken, og det registreres tre returer for den utsendte pulsen.



Figur 2.12: Illustrasjon av multiple returer av en utsendt puls. Eksemplet viser tre returer og pulsstyrken for hver retur.

Punktskyer

Punktskyer består av store mengder med informasjon som fanges opp av en laserskanner. De fleste laserskanningene krever videre prosessering av punktskyene, for å kunne ekstrahere informasjon som for eksempel form, trehøyde, tetthet osv. Punktskyer er gitt som X-, Y- og Z-koordinater med attributter som for eksempel beskriver antall returer. Punktskyer kan videre behandles til digitale høydemodeller i 2.5d raster (Vosselman & Maas 2010, p. 45). Figur 2.13 viser et eksempel på en punktsky farget etter høyde.

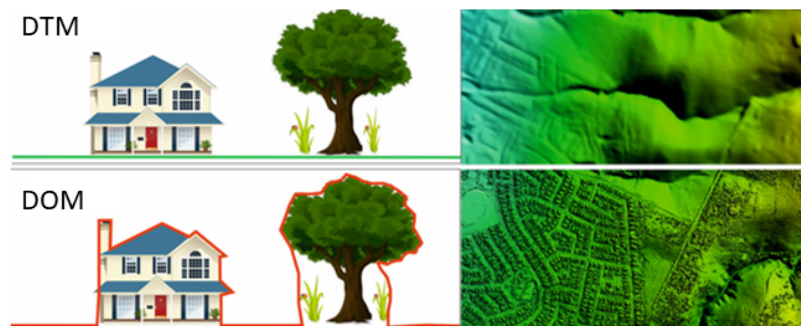


Figur 2.13: En punktsky farget etter høyde for et område i dette prosjektet.

2.4.1 Digitale høydemodeller

Laserskanning fra fly kan benyttes til å lage digitale høydemodeller med høy nøyaktighet (Skogseth & Norberg 2014, p. 307). Digitale høydemodeller beskriver topografi i form av geometrisk modellering, og kan kategoriseres i to typer; digital terrengmodell (DTM) og digital overflatemodell (DOM).

Digital terrengmodell er en matematisk modell som beskriver jordoverflaten. Denne modellen inneholder høydeinformasjon om terrenget (Vosselman & Maas 2010, p. 135). DTM avledes av laserpunkter, klassifisert som bakke fra de siste returene i laserdataen. I likhet med DTM blir også digital overflatemodell benyttet ved topografisk kartlegging. DOM beskriver synlige høydekurver sett ovenfra, men i motsetning til DTM, skildrer DOM vegetasjonsdekke i tillegg til menneskeskapte objekter som for eksempel bygninger. DOM avledes fra de første returpunktene i laserdataen. I figur 2.14 er det gitt en grafisk illustrasjon som viser forskjellen mellom DTM og DOM (Vosselman & Maas 2010, p.135,136).



Figur 2.14: Digital terrengmodell(DTM) beskriver terrengets høydeinformasjon, og digital overflatemodell(DOM) beskriver i tillegg vegetasjon og menneskeskapte objekter. Figuren er hentet fra (Services 2017)

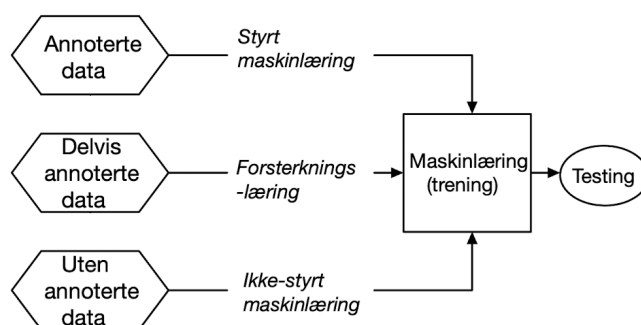
Normalisert Digital Overflatemodell, nDOM, er gitt som differansen mellom DOM og DTM. Etersom nDOM representerer absolutt høyde i henhold til jordoverflaten, må DTM subtraheres fra DOM ved beregning av nDSM (Hashemi (2008). Fordelen ved nDOM er at objektenes høyde beskrives relativt til bakken, slik at bakken får en høyde lik null. Dette er nyttig informasjon for en maskinlæringsalgoritme som skal lære seg forskjellen på for eksempel hus og bakke.

$$nDSM = DOM - DTM \quad (2.4)$$

2.5 Maskinl ring

Maskinl ring er et begrep innen datateknologi som omfatter et bredt spekter av algoritmer og modelleringsverkt y. Maskinl ring er designet for   kunne etterligne menneskets intelligens og brukes til   l se komplekse utfordringer ved dataprosessering. En maskinl ringsalgoritme forbedrer sin klassifikasjonsevne ved hjelp av erfaringer og repetisjon. Formålet med maskinl ring er   l re opp algoritmene til   predikere bedre i l pet av tilpasningsprosessen. I likhet med mennesker vil en god algoritme konstant trenes opp i form av   prosessere ny data, og dermed klare   l re av tidligere feil. ((El Naqa et al. 2015, p.6)).

Figur 2.15 viser en oversikt over ulike kategorier innenfor maskinl ring, i henhold til treningsdata. Maskinl ringsalgoritmer bygger p  matematiske modeller i form av treningsdata og er hovedsakelig klassifisert i tre kategorier; styrt l ring (supervised), ikke-styrt l ring (unsupervised) og forsterkningsl ring (semi-supervised).



Figur 2.15: Tre kategorier innen maskinl ringsalgoritmer i henhold til annotert data(fasitdata). Kategoriene består av styrt-l ring, forsterkningsl ring og ikke-styrt maskinl ring.

Ved styrt maskinl ring l rer modellen   predikere riktig svar, i et annotert datasett som inneholder kjente utgangsverdier for gitte inngangsverdier. Det annoterte datasettet består av store mengder data med sp rsm l og svar. To av de vanligste formene for styrt l ring er klassifikasjon og regresjon. Klassifikasjonsalgoritmen deler inn dataene i ulike kategorier som for eksempel farger, mens regresjonsalgoritmen gir eksakt svar basert p  resultatet ((El Naqa et al. 2015, p.7)).

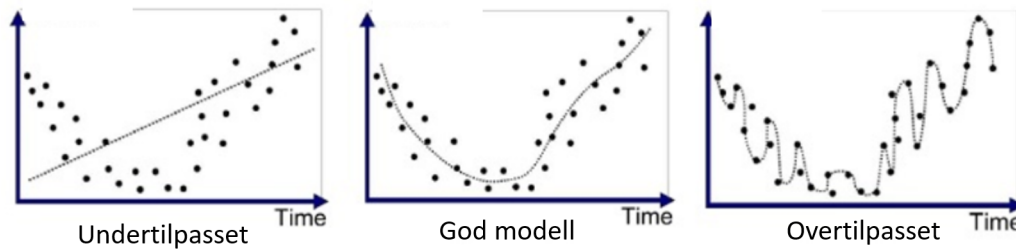
Ikke-styrt maskinl ring derimot, brukes uten annoterte datasett, der det er gitt inngangsverdier uten tilh rende utgangsverdier. Ved denne typen l ring m  modellen pr ve   finne skjulte m nstre i datasettet. Et eksempel p  en slik algoritme er klyngeanalyse (clustering). Klyngeanalyse kan brukes til   finne likheter i datasettet, og gruppere data etter for eksempel lik st rrelse og form ((El Naqa et al. 2015, p.7)).

Overtilpasning

En maskinl ringsmodell trenes opp med et sett av treningsdata, og brukes til   predikere ny data. Formålet med modellen er   klare   predikere mest mulig riktig p  usette data (Dietterich (1995)). Dersom modellen presterer godt p  treningsdataen, men sliter med prediksjon av usette data, kan modellen v re overtilpasset treningsdataen.

En overtilpasset modell kan oppstå dersom modellen er altfor kompleks i henhold til dataen som er gitt. Dersom det er altfor mange parametre å trene på, kan modellen begynne å trekke til seg unødvendig informasjon om treningsdataen, som ikke vil gi en god nok generalisert modell, se figur 2.16. Modellen kan også bli undertilpasset hvis den ikke er kompleks nok i forhold til datasettet, se figur 2.16 (Raschka & Mirjalili 2019, p.75).

Det viktigste ved å trene opp en modell, er å generalisere modellen slik at den predikerer mest mulig rett på usette data.

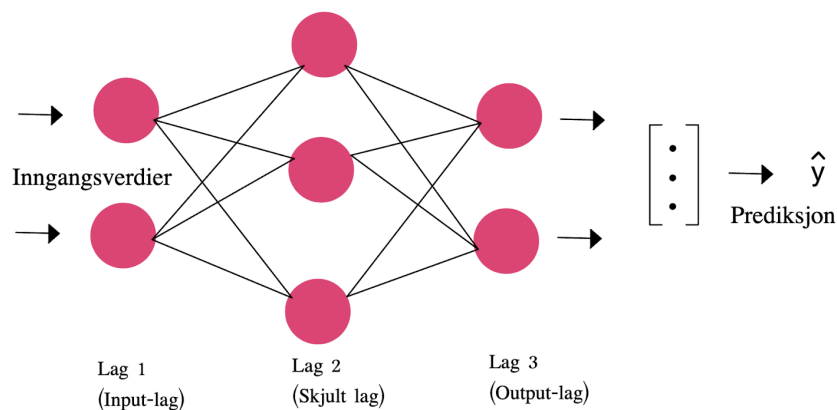


Figur 2.16: Et eksempel på hvordan en undertilpasset, godt tilpasset og overtilpasset modell vil håndtere usette data. Modifisert figur hentet fra (Hoffman 2021).

2.5.1 Kunstige nevralt nettverk

Kunstige nevralt nettverk (ANN) er nettverk designet for å kunne løse komplekse utfordringer, på samme måte som menneskehjernen prosesserer informasjon (Raschka & Mirjalili (2019)). Det kunstige nevralt nettverket tilegner seg kunnskap ved at det trenes opp til å oppdage mønstre og lære av tidligere erfaringer. ANN er organisert i lag, og blir formet av hundre enkeltenheter eller kunstige nevroner. Måten det nevralt nettverket oppfører seg på, avhenger av nevronenes aktiveringsfunksjoner (Agatonovic-Kustrin & Beresford (2000)).

Figur 2.17 viser et eksempel på et kunstig nevralt nettverk. Dette nettverket består av tre lag; et inngangslag, et skjult lag og et utgangslag. Hvert lag inneholder et sett med nevroner som videre er koblet sammen med nevronene i det neste laget. Arkitekturen til en enkel "multilayer perceptron, MLP" vil se slik ut:

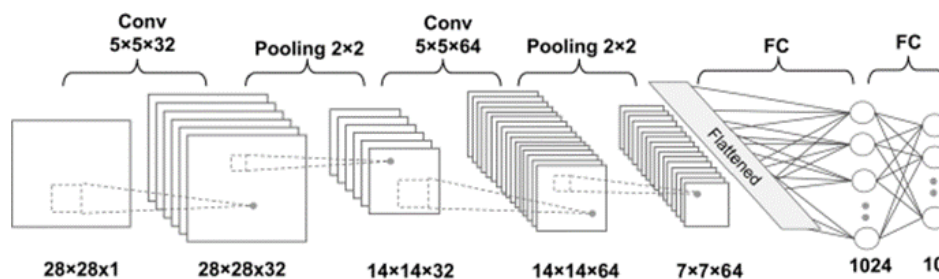


Figur 2.17: Illustrasjon av "multilayer perceptron, MLP", et nevralt nettverk bestående av nevroner satt sammen i tre ulike lag. Inngangsverdier vil bevege seg gjennom nettverket og dette vil gi en prediksjon, \hat{y}

ANN fungerer ved at summen av inngangsverdier og vekt-koeffisienter utgjør nevronets aktive-ring. Aktiveringssignalet passerer gjennom en aktiveringsfunksjon for å kunne produsere nevro-nets utgangsverdi (Agatonovic-Kustrin & Beresford (2000)). Under trening vil sammenlikningen av prediksjonen og fasiten avgjøre hvor nøyaktig prediksjonen er, og eventuelt hvor stort avviket vil være. Dermed kan vekt-koeffisientene oppdateres, slik at modellen forbedres ytterligere. Når et nettverk blir trent opp og testet, blir det gitt ny inngangsinformasjon som igjen brukes til prediksjon av ny utgangsverdi. Som modelleringsverktøy krever ikke kunstige nevrone nettværk kunnskap om selve data-kilden. Ettersom slike nettværk ofte inneholder mange vekt-koeffisien-ter som skal estimeres, krever ANN store treningssett for å kunne gjennomføre simuleringer (Agatonovic-Kustrin & Beresford (2000)).

2.5.2 Konvolusjonelle nevrone nettværk

Konvolusjonelle nevrone nettværk, CNN er en dyplæringsmodell som brukes til bildeklassifise-ring. Dette nevrone nettværket er inspirert av den delen av menneske-hjernen som gjenkjenner objekter. Arkitekturen til en CNN-modell består av konvolusjonelle lag, "pooling" lag og akti-veringsfunksjoner, som er bygd opp av ulike lag. Disse lagene trekker ut relevant informasjon, som bidrar til at nettværket klarer å gjenkjenne objekter på bilder. CNN har ofte lag av nevrone som det siste laget i nettværket, for å utføre prediksjoner basert på informasjonen fra de tidligere lagene. Lagene er fullstendig tilkoblede, FC, med nevrone knyttet sammen som i MLP-modellen (Raschka & Mirjalili 2019, p. 517-520). Et eksempel på hvordan en CNN-arkitektur er bygd opp er gitt i figuren 2.18. Nettværket er bygd opp av lag med konvolusjon, "pooling" og FC.



Figur 2.18: Et eksempel på en CNN-arkitektur bestående av lag med konvolusjon, "pooling" og FC. Figuren er hentet fra (Raschka & Mirjalili 2019, p. 542)

En av de viktigste oppgavene til en maskinlæringsalgoritme, er å kunne ekstrahere relevant informasjon som modellen kan benytte seg av. CNN er spesielt egnet til å trekke ut denne informasjon fra rådataen. På et bilde kan informasjonen deles inn i to kategorier, "high-level"- og "low-level"-informasjon. "Low-level"-informasjon er detaljene til et objekt i et bilde, som alene ikke vil gi nok informasjon til å kunne beskrive noe mer om objektet. Dette kan for eksempel være kanter på et bygg eller enkle mønstre. Ved å sette sammen de små detaljene som nettværket finner, skapes "high-level"-informasjon. "High-level" er den informasjonen vi ønsker å finne ut av i objekt-detektering. Et dypt konvolusjonelt nevrone nettværk er bygd opp slik at de første lagene trekker ut "low-level"-informasjon, som benyttes til å skape "high-level"-informasjon (Raschka & Mirjalili 2019, p. 519).

Konvolusjonelle lag

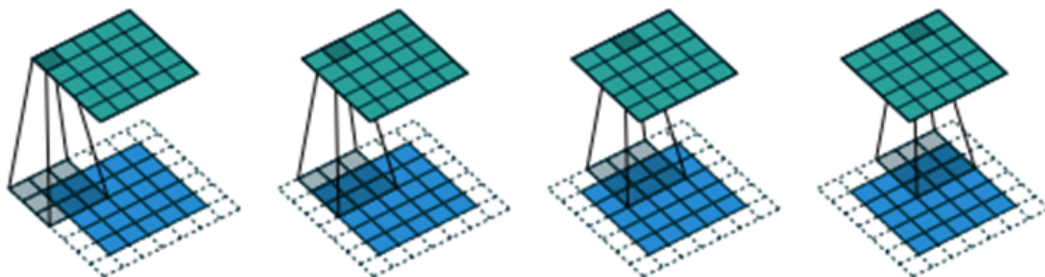
Et konvolusjonelt lag er et filter plassert over bildet og regnes som et av de viktigste lagene i CNN. Dette filteret på $n \times n$, multipliseres med bildets verdier. Verdiene på filterene er vekt-koeffisienter som oppdateres når modellen trenes opp. Hovedoppgaven til dette laget, er å trekke ut informasjon fra bildet og sende informasjonen videre til neste lag. De vanligste størrelsene på et filter er 1×1 , 3×3 eller 5×5 .

Konvolusjonelle operasjoner vil føre til at det opprinnelige bildet vil minke i størrelse. Når filteret flyttes rundt på bildet, vil pikselverdiene i midten av bildet ta del i flere beregninger enn pikselene på kanten av bildet. Informasjon på kanten av bildet kan da gå tapt. For å få med informasjonen langs kantene, kan parametre som "stride" og "padding" justeres. Dette er parametere som brukes til å justere bildet og filteret.

"**Stride**" er en parameter som bestemmer hvor mye filteret skal bevege seg i (x,y) retning (Dumoulin & Visin (2016)). "Stride" er en avgjørende parameter som bestemmer antall vekt-koeffisienter og størrelsen som returneres etter en konvolusjonell operasjon.

"**Padding**" er en operasjon der det legges til nuller på bildets kanter, for å øke størrelsen på bildet (Dumoulin & Visin (2016)). Dette gjøres dersom filteret ikke dekker hele bildet eller hvis effekter på kanten av bildet skal fanges opp.

Figur 2.19 viser en enkel konvolusjonell operasjon. Inngangsbildet er "paddet" med nuller ($p=1$) og et filter på 3×3 forflytter seg med posisjonen til enhver operasjon. Etter en slik operasjon, vil dimensjonen til utgangsbildet være lik dimensjonen til inngangsbildet.



Figur 2.19: Et eksempel på en konvolusjonell operasjon. Inngangsverdi: 5×5 , filter: 3×3 , stride: 1, padding: 1. Operasjonen resulterer i lik dimensjon for utgangsbildet som inngangsbildet. Figuren er hentet fra Dumoulin & Visin (2016).

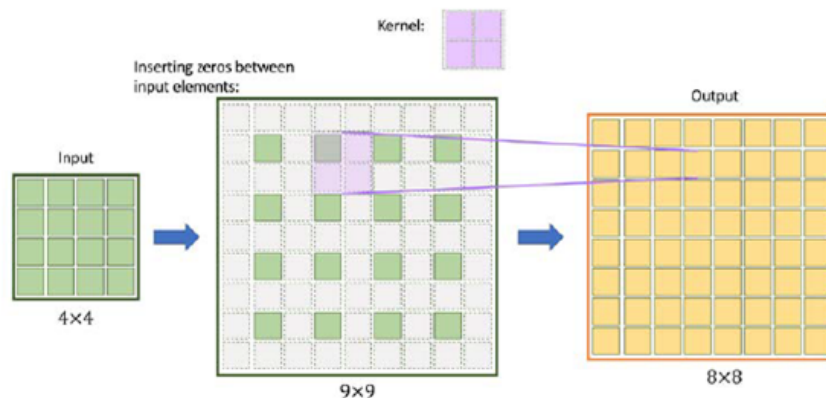
Transponert konvolusjon Transponert konvolusjon, også kalt oppskalingskonvolusjon er en metode som benyttes for å oppskalere et bilde. Gjennom et nettverk vil modellen minimere størrelsen på inngangsbildet. For å få tilbake den opprinnelige størrelsen brukes et transponert konvolusjonelt filter med vekt-koeffisienter. Vektene vil kunne oppdateres når modellen trenes opp. Formålet med dette er å øke bildets dimensjoner, i tillegg til å beholde informasjonen uten at den går tapt (Dumoulin & Visin (2016)).

Ved å manipulere det minimerte bildet, kan et bilde med høyere dimensjon returneres. Denne

prosessen kan beskrives i to steg:

1. Legge til nuller etter hvert element, noe som vil øke størrelsen på bildet.
2. Bildet fra steg 1 multipliseres med et $n \times n$ konvolusjonelt filter med "stride".

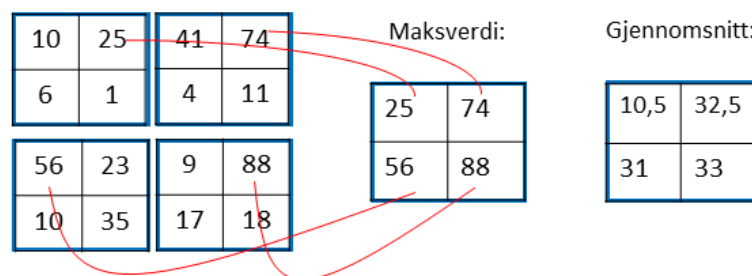
Figur 2.20 er et eksempel med et inngangsbilde på 4×4 . Det legges til nuller og multipliseres med et filter på 2×2 med en "stride" lik (2,2). Dette resulterer i et utgangsbilde som er dobbelt så stort som inngangsbildet.



Figur 2.20: Et eksempel på en transponert konvolusjonell operasjon. Inngangsbildet: 4×4 , Padding: 1 og setter inn 0 etter hvert element, som resulterer i et 9×9 bilde som multipliseres med en 2×2 filter med en stride: 2. Utgangsbildets dimensjoner er doblet til 8×8 sammenlignet med inngangsbildet. Figuren er hentet fra (Raschka & Mirjalili 2019, p. 648).

"Pooling layer"

"Pooling layer" er et lag som brukes i en CNN. Dette er en filteroperasjon, hvorav et filter på $n \times n$ brukes for å tilegne bildets område en ny verdi. Filteret kan tilpasses slik at det hentes ut en maksimumsverdi eller en gjennomsnittsverdi, fra det området filteret dekker, se figur 2.21. Dette laget har ingen trenbare parametre.



Figur 2.21: Et eksempel på maks og snitt "pooling" operasjon. Illustrasjonen viser resultatet etter maks og snitt "pooling". Inngangsbilder på 4×4 , filter på 2×2 og stride på (2,2) resulterer i et utgangsbilde på 4×4 .

"Pooling layer" har to viktige egenskaper; å kompensere for små variasjoner i et område og å redusere antall trenbare parametre. Dette bidrar til raskere beregningstid, i tillegg til færre trenbare parametre som kan redusere sjansen for en overtilpasset modell (Raschka & Mirjalili (2019)).

Tapsfunksjon

En tapsfunksjon er essensiell for at en modell skal kunne lære seg å predikere rett. Etter at modellen har predikert inngangsverdier, sammenlignes resultatet med den sanne verdien. Det er veldig mange tapsfunksjoner som kan benyttes, men i oppgaven blir det valgt å se nærmere på "Dice loss". "Dice loss" er en funksjon som brukes dersom datasettet inneholder ubalanserte klasser (Sudre et al. (2017)). "Dice loss" er gitt slik ved et binært problem:

$$DL(y, \hat{y}) = 1 - \frac{2y\hat{y} + 1}{y + \hat{y} + 1} \quad (2.5)$$

"Dice score"-koeffisient berenger overlapp av prediksjonen og fasiten. Derav får vi en differanse mellom prediksjonen og fasiten som videre brukes for å trene modellen (Jadon (2020)). Målet er å få minst mulig "loss" og mest mulig likhet mellom prediksjonen og fasiten.

"Backpropagation"

"Backpropagation" er metoden som brukes for å beregne gradienten for hvert lag i det nevralt nettverket. (Zhang (2016)). Etter en "forward pass", vil tapsfunksjonen returnere en verdi. Formålet med "Backpropagation", er å beregne avviket i hvert lag slik at vektkoeffisientene kan oppdateres for å minke tapet. Oppdatering av vektkoeffisientene gjøres ved hjelp av en optimaliseringsfunksjon som vil forsøke å finne det globale minimumet til tapsfunksjonen.

Å finne det globale minimumet, er en iterativ prosess som består av å beregne avviket for hvert lag i nettverket ved hjelp av partiell derivering. Prosessen foregår ved at det jobbes baklengs gjennom nettverket. Prosessen starter altså i det siste laget av nettverket og avsluttes i det første laget (Raschka & Mirjalili 2019, p. 417-422). I denne studien blir tre forskjellige optimaliseringsfunksjoner utforsket, SGD ("Stochastic Gradient Descent"), ADAM ("Adaptive Moment Estimation") og RMSprop ("Root Mean Squared Propagation").

Aktiveringsfunksjon

For at et nevralt nettverk skal kunne håndtere komplekse oppgaver, må den lære seg å løse ikke-lineære data. En aktiveringsfunksjon brukes for å transformere dataen. Både lineære og ikke-lineære funksjoner brukes for å løse komplekse oppgaver. Fordelen ved å benytte en lineær funksjon, er tidsbesparing ved beregningsdelen. En aktiveringsfunksjon er sentral for at modellen skal kunne klare å løse problemer som inneholder ikke-lineære data. Aktiveringsfunksjonen har ingen trenbare parametere (Hao et al. (2020)). Det finnes ulike typer aktiveringsfunksjoner, men i denne oppgaven beskrives kun tre av disse.

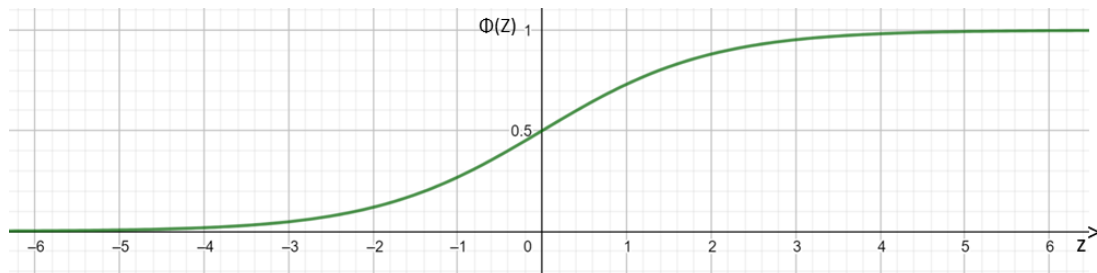
Sigmoid

Sigmoid-kurven beskriver en ikke-lineær aktiveringsfunksjon. Funksjonen vil ta inn en verdi og

returnere en verdi mellom $[0,1]$. Formelen er gitt slik:

$$\phi(z) = \frac{1}{1 + e^{-z}} \quad (2.6)$$

Denne funksjonen vil sørge for at verdiene verken blir altfor store eller altfor små. Sigmoid-funksjonen kan også brukes som det siste laget i et nettverk for binærklassifisering. Funksjonen vil fungere som en sannsynlighetsfordeling som avgrenses mellom $[0,1]$. Ved å sette en grenseverdi kan det avgjøres hvilken klasse inngangsverdien tilhører (Raschka & Mirjalili 2019, p.463).



Figur 2.22: Grafisk fremstilling av sigmoid-funksjon.

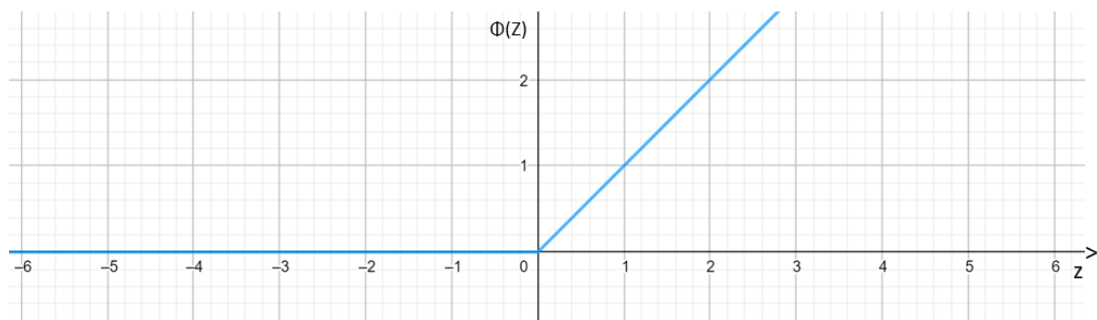
En ulempe med sigmoid-funksjonen er at den raskt vil konvergere mot 0 eller 1, se figur 2.22. Gradienten i disse områdene vil bli liten. Under "backpropagation" multipliseres gradienten med prediksjonsfeilen produsert av "loss"-funksjonen. Hvis gradienten er et lite tall, vil dette medføre at prediksjonsfeilen blir et tilsvarende lite tall. Modellen vil da lære svært langsomt eller ingenting, og dette problemet kalles "the vanishing gradient problem" (Szandala (2021)).

ReLU

Aktiveringsfunksjonen ReLU, "Rectified Linear Unit", er en ikke-lineær aktiveringsfunksjon. Funksjonen er gitt slik:

$$\phi(z) = \max(0, z) \quad (2.7)$$

ReLU vil ta inn verdier, transformere negative verdier til 0 og gi positive verdier en lineær transformasjon, se figur 2.23. ReLU er en enkel funksjon som presterer godt i henhold til å løse kompleksiteten i dype nevralt nettverk (Szandala 2021). Ved at dette er en enkel funksjon, blir beregningstiden kortere når inngangsverdiene transformeres og gradienten beregnes ved "backpropagation".



Figur 2.23: Grafisk fremstilling av ReLU-funksjon.

ReLU lider ikke av problemet "the vanishing gradient problem", ettersom negative verdier blir

satt til 0 og positive verdier får tildelt lik verdi. En ulempe med ReLU er at mange av inngangsverdiene kan ende opp med en verdi på 0, hvis inngangsverdiene består av negative verdier. Dette vil føre til at modellen ikke vil kunne trenes i disse områdene lenger (Szandala (2021)).

Softmax

Aktiveringsfunksjonen ”softmax” betegnes ofte som det siste laget i en multiklasse-klassifisering. I likhet med Sigmoid-funksjonen vil ”softmax” returnere en verdi i intervallet, $[0,1]$. ”Softmax” vil transformere inngangsverdien til en sannsynlighetsverdi. Den transformerte sannsynlighetsverdien gir en indikasjon på hvilken klasse den gitte inngangsverdien tilhører (Raschka & Mirjalili 2019, p. 465). ”Softmax”-funksjonen er gitt slik:

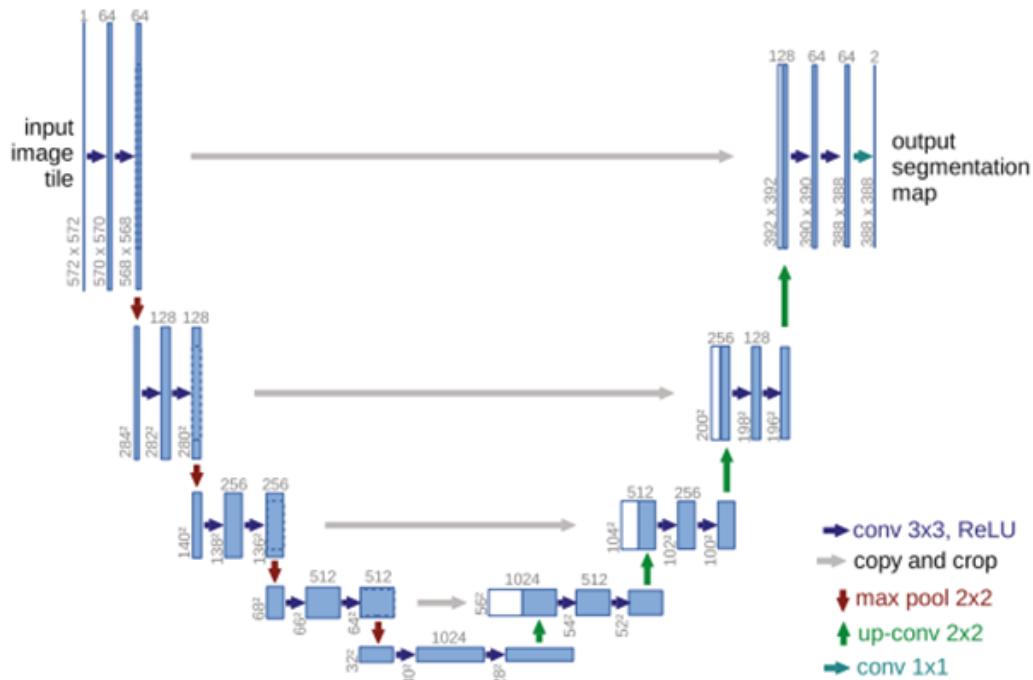
$$p(z) = \phi(z) = \frac{e^{z_i}}{\sum_{j=1}^M e^{z_j}} \quad (2.8)$$

2.5.3 Fullstendig konvolusjonelle nettverk

FCN, ”Fully Convolutional Network”, er nettverk som ikke inneholder fullstendig tilkoblede nevrone. Disse nettverkene inneholder kun konvolusjonelle lag, ”pooling” lag og aktiveringsfunksjoner. Denne type arkitektur ble satt i søkelys av (Long et al. (2015)), for pikselbasert prediksjon. I studien, (Long et al. (2015)), blir det beskrevet at strukturen presterer godt for pikselbasert prediksjon, og at modellen kan trenes fra start til slutt. FCN er ofte satt sammen av to deler som består av nedskalering og oppskalering. Nettverket er i stand til å ta imot en vilkårlig bildestørrelse, trenes effektivt og produsere en pikselbasert prediksjon med lik størrelse som inngangsbildet (Long et al. (2015)).

U-Net

U-net er et fullstendig tilkoblet nettverk(FCN), bygd videre på prinsipper utarbeidet av J. Shelhamer Long (Long et al. (2015)). U-net-arkitekturen for prediksjon av biomedisinske bilder med få mengder med treningsdata, ble utviklet av O. Ronneberger (Ronneberger et al. 2015) . Nettverket har rask beregningstid og er formet som en U. Arkitekturen er bygd opp av konvolusjonelle lag og ”pooling”-lag. U-net er en modell som trener fra start til slutt, og resulterer i presis segmentering selv når modellen trenes på få datamengder (Ronneberger et al. 2015).



Figur 2.24: U-net arkitekturen til O. Ronneberger. Arkitekturen består av to deler, nedskalering og oppskalering. Venstre del av arkitekturen er nedskaleringsdelen og høyre del er oppskaleringsdelen. Figur hentet fra (Ronneberger et al. 2015)

Nettverket gitt i figur 2.24, består av to deler; nedskalering og oppskalering. Nedskalering utgjør venstre del av nettverket og består av fire konvolusjonelle blokker (Trier et al. (2022)). En blokk med konvolusjon består av to 3x3 konvolusjonelle filtre uten "padding", i tillegg til aktiveringsfunksjonen ReLU. Etter at et bilde føres gjennom en slik blokk, føres resultatet videre gjennom et 2x2 maks-"pooling" steg med to i "stride", for å minke inngangsverdien. For hvert steg nedover i nettverket, vil antall informasjonskanaler i z-retning dobles og inngangsverdier i x- og y-retning halveres.

Høyre del av nettverket kalles oppskaleringsdel og består av fire konvolusjonelle blokker (Trier et al. (2022)). Hver blokk består av en 2x2 oppskalingskonvolusjon, to 3x3 konvolusjonelle filtre og aktiveringsfunksjonen ReLU. Forskjellen fra nedskaleringsdelen, er at informasjonen nå skal oppskaleres. Dette gjøres ved hjelp av et filter for oppskalingskonvolusjon og "skip connection". Oppskalingskonvolusjon er en operasjon som halverer antall informasjonskanaler og dobler størrelsen i x- og y-retning. Etter hver "pooling"-operasjon i nedskaleringsdelen, er det en "skip connection" som klipper ut midten av bildet og sender informasjonen videre til oppskaleringsdelen av nettverket. Dette resulterer i at nettverket klarer å trekke til seg informasjon fra de første lagene og kombinere denne informasjonen med de siste lagene (Ronneberger et al. 2015).

Det siste laget i nettverket består av et 1x1 konvolusjonelt filter, som transformerer informasjonskanalene til ønsket antall prediksjonsklasser. Totalt har nettverket 23 lag med konvolusjonelle filtre (Ronneberger et al. 2015).

Trene en U-net

Det er mange likhetstrekk ved trening av U-net og andre typer nettverk. Ved trening av U-net, sendes bildet inn i nettverket, som predikerer hver piksel på bildet og returner en maske. Den første gangen initialiseres vekt-koeffisientene tilfeldig. Den predikerte masken vil da kunne sammenlignes med fasiten, og tapet vil bli kalkulert av tapsfunksjonen. ”Backpropagation” initieres og vekt-koeffisienter oppdateres. På denne måten vil nettverket trene ved å bevege seg fram og tilbake (Ronneberger et al. 2015).

2.5.4 Parametermåling av modellen

Forvirringsmatrise (”Confusion matrix”)

En forvirringsmatrise er en metrikk som brukes for å kunne måle hvor god en modell er på å predikere rett og galt. Forvirringsmatrisen viser den predikerte klassen mot den sanne klassen. I form av en matrise vil klassene som predikerer rett og galt, raskt kunne detekteres. Matrisen vil identifisere klasser der modellen har predikert verdier som er sanne positive (TP), falske negative (FN), falske positive (FP) og sanne negative (TN). Verdiene fra forvirringsmatrisen kan brukes for å regne ut andre typer metrikker som nøyaktighet (”accuracy”), presisjon (”precision”), ”recall” og ”F1-score”.

		Predikerte klasse	
		P	N
Sanne klasse	P	Sann positiv(TP)	Falsk negativ(FN)
	N	Falsk positiv(FP)	Sann negativ(TN)

Figur 2.25: Et eksempel på en forvirringsmatrise.

Nøyaktighet (”Accuracy”)

Nøyaktighet er en viktig metrikk som blir brukt for å måle hvor god en maskinlæringsmodell er til å klassifisere. Dette avgjøres ved at antall riktig predikerte deles på totalt antall prediksjoner. Nøyaktighet er et mål som blir brukt mye. Det kan likevel være utfordrende å avgjøre om en modell er god til å predikere, dersom det er ubalanserte klasser i datasettet. Ved ubalanserte klasser kan nøyaktighet gi et misvisende inntrykk av hvor god modellen er til å predikere (Raschka & Mirjalili 2019, p.21).

$$ACC = \frac{Correct\ Prediction}{Total\ prediction} = \frac{TP + TN}{TP + FN + FP + TN} \quad (2.9)$$

”F1-score”

”F1-score” er en metrikk som brukes for å evaluere hvor god en modell er til å klassifisere data. F1 måler nøyaktigheten på klassifiseringen sammenlignet med fasitdataen og er godt egnet for ubalanserte datasett (Raschka & Mirjalili 2019, p. 214).

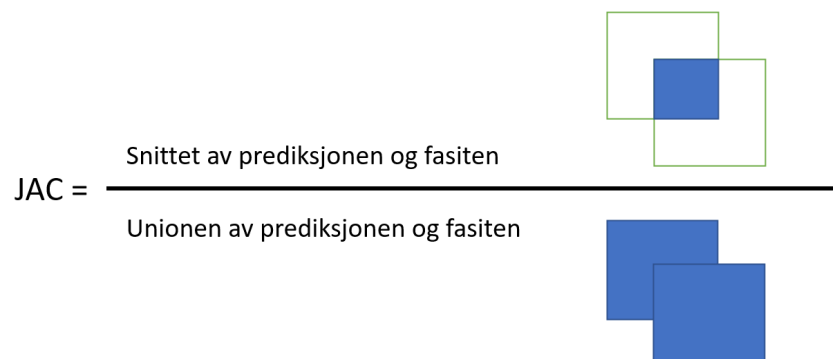
$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + TP)} \quad (2.10)$$

”Jaccard index”

”Jaccard similarity coefficient” også kjent som ”IoU, intersection over union”, er en metrikk som forteller hvor stor andel av predikeringen som er lik fasiten. Formelen er gitt slik:

$$JAC = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (2.11)$$

”Jaccard index” vil gi et tall mellom 0 og 1. Tallet avgjør hvor lik predikeringen er i forhold til fasiten. Formelen beskriver altså andelen av predikeringen som er lik fasiten, dividert med området som dekkes av både predikeringen og fasiten, se figur 2.26:



Figur 2.26: Illustrasjon av ”Intersection over Union”. Snittet av prediksjonen og fasiten divideres med unionen av prediksjonen og fasiten.

R-kvadrert (”R-squared”)

R-kvadrert, også kjent som determinantkoeffisient, er en metrikk som brukes for å evaluere regresjonsmodeller. Formelen for R-kvadrert er gitt som følgende:

$$R^2 = 1 - \frac{\text{Unexplained variance}}{\text{Total variance}} = 1 - \frac{\sum(Y_i - \hat{Y}_i)^2}{\sum(Y_i - \bar{Y})^2} \quad (2.12)$$

I et regresjonsproblem kan det være utfordrende å bruke for eksempel nøyaktighet for å avgjøre hvor god en modell er til å predikere. R-kvadrert er et mål som beskriver forholdet mellom variansen i prediksjonen og fasiten (Redell (2019)). R-kvadrert-verdien gir et uttrykk for modellens evne til å fange variasjonen i datasettet. R-kvadrert returnerer en verdi fra $-\infty$ til 1, der 1 utgjør all variasjon som blir forklart av modellen (Redell 2019).

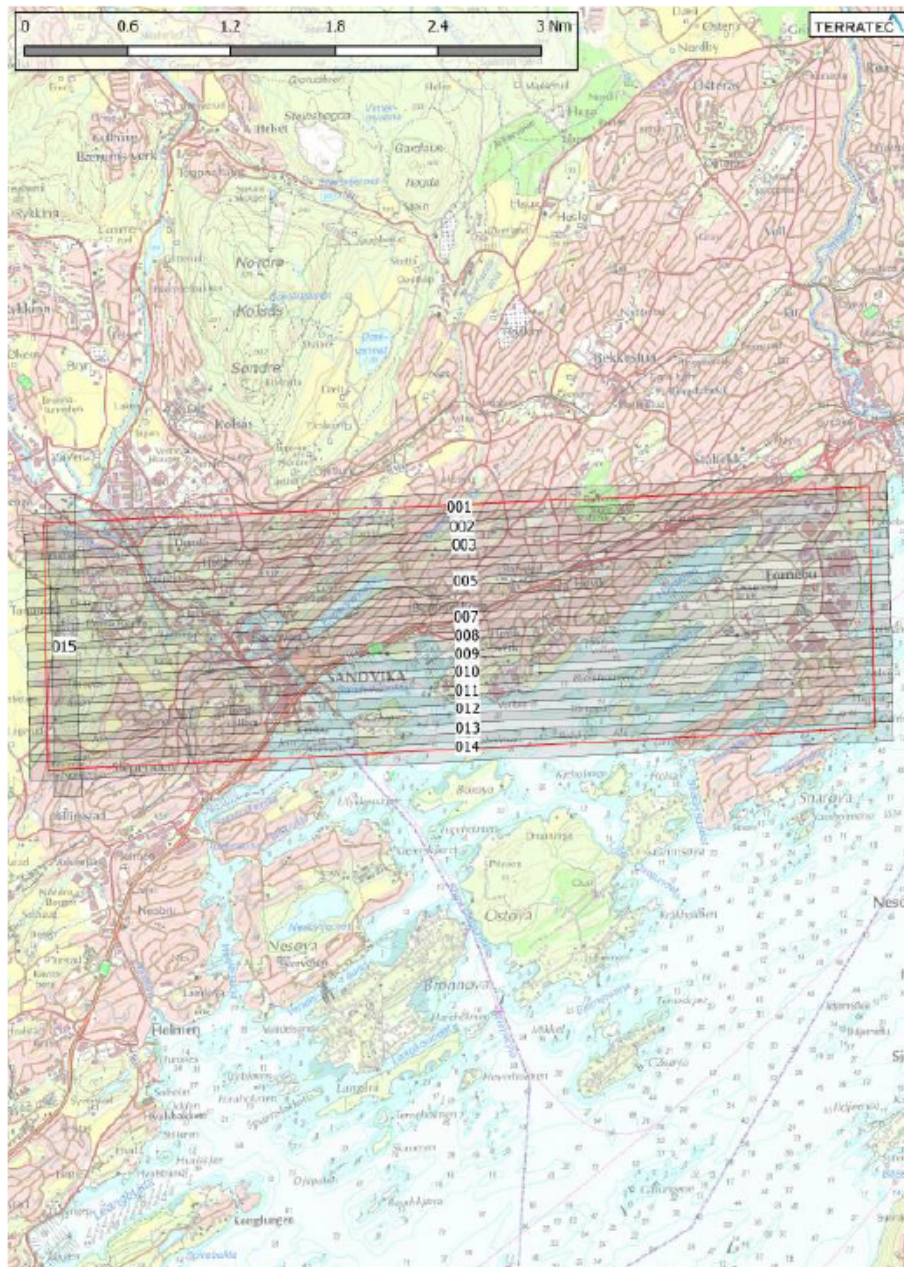
Kapittel 3: Metode

3.1 Datainnsamling

I denne studien har det blitt brukt hyperspektrale bilder og LiDAR data fra fly. Datainnsamlingen har blitt utført av Terratec AS som en del av et prosjekt leder av Bærum kommune, 24. august 2019. Områdedekningen var fordelt over tre ulike områder med et totalt areal på 37.4km^2 . Terratec AS har hatt ansvaret for radiometrisk og geometrisk kalibrering av de hyperspektrale bildekubene. Prosjektets mål er å automatisere kartlegging fra laser og hyperspektrale data ved hjelp av maskinlæring.

Datainnsamling har blitt gjennomført på tre ulike steder på østlandet. I denne studien har innsamlet data fra Høvik i Bærum kommune, blitt tildelt. Figur 3.1 viser flyplanen som ble brukt til datainnsamling av det gitte området. Det har blitt benyttet fire ulike sensorer for innsamling av data for dette området. Hyperspektrale bildekuber har blitt samlet inn ved hjelp av to HySpex sensorer, HySpex VNIR-1800 og HySpex SWIR-384, som begge ble montert i en gyroramme. Laserdata er innhentet samtidig som den hyperspektrale avbildningen og er gjort ved hjelp av to laserskannere, Leica ALS70 og Riegl VQ-1560i.

De hyperspektrale bildekubene har blitt levert med radiansverdier. Oppløsningen er på 0.3m og 0.7m , henholdsvis fra HySpex VNIR-1800 og HySpex SWIR-384. En 30cm DOM innhentet fra Leica ALS70-laserskanneren har blitt brukt for ortorektifisering av bildene. Både georefereringen og ortorektifiseringen er gjort med nærmeste nabointerpolasjon og levert med transformasjonen, WGS84, UTM32.



Figur 3.1: Flyplan brukt for datainnsamling av Terratec. Figuren er hentet fra prosjektrapporten(10990) utlevert av Terratec.

3.1.1 Sensorspesifikasjoner

Hyperspektrale sensorer

To typer HySpex sensorer ble benyttet for anskaffelse av hyperspektrale bildekuber. HySpex VNIR-1800 sensoren skal dekke den synlige og den nær-infrarøde delen av EM-spekteret. HySpex VNIR-1800 sensoren dekker et bølglengdeområde fra $400nm$ til $1000nm$, dette utgjør 186 bånd. Hvert bånd dekker ca $3.22nm$ med bølglengde. Avbildningen fra denne sensoren utgjør en romlig oppløsning på $30cm$. Dette resulterer i at en piksel representerer et område på $30 \times 30cm$ på bakken.

HySpex SWIR-384 sensoren dekker den kortbølgede delen av det infrarøde området(SWIR) i det elektromagnetiske spekteret. Sensoren dekker et bølgelengdeområde fra $930nm$ til $2500nm$ som utgjør 288 bånd. Hvert bånd vil dekke et område på ca $5.45nm$ med bølgelengde. Den romlige oppløsningen er på $70cm$, som tilsvarer et område på $70 \times 70cm$ på bakken. Spesifikasjoner for begge HySpex-sensorene er gitt i tabell 3.1.

Tabell 3.1: Tabell for HySpex sensorene, spesifikasjonene er hentet fra (HySpex-NEO 2022b) og (HySpex-NEO 2022a)

Spesifikasjoner	VNIR-1800	SWIR-384
Bølgelengdeområde	400-1000nm	930-2500nm
Antall spektrale bånd	186	288
Spektraloppløsning	3.26nm	5.45nm
Bakkeoppløsning(GSD)	0.3m	0.7m
Radiometrisk oppløsning	16 bit	16bit
Maks åpningsvinkel (FOV)	17 grader	16 grader
Maks hastighet(med maks oppløsning)	260 fps	500 fps
Piksel FOV(på tvers og langs)	0.16/0.32mrad	0.73/0.73mrad

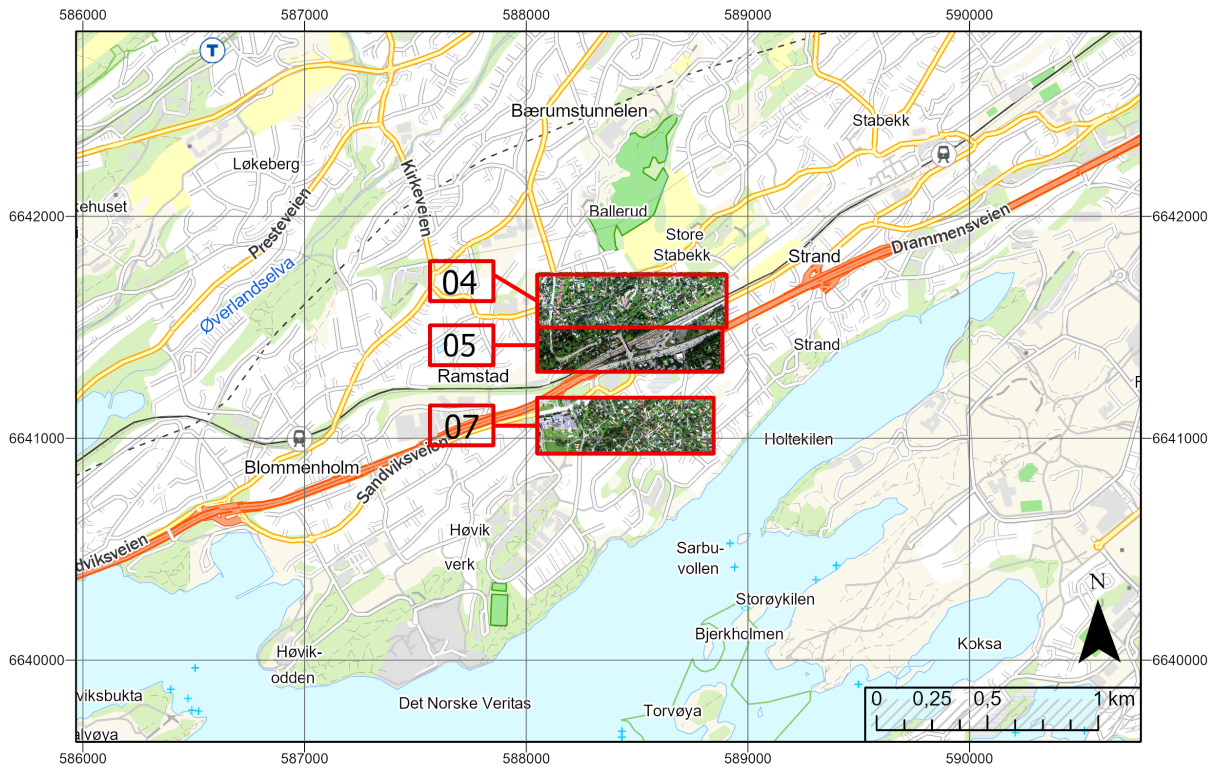
LiDAR sensorer

To type LiDAR-skannere, Leica ALS70 og Riegl VQ-1560i, ble brukt for innsamling av laserdata. Begge laserskannerne sender ut laserpulser på $1064nm$. Det ble valgt å bruke Leica ALS70 i tillegg til Riegl VQ-1560i, fordi det kun var denne som passet fysisk sammen med HySpex sensorene i flyets kamerahull. Leica ALS-70 ble brukt til å produsere en overflatemodell av høy kvalitet fra punktskyen som ble brukt til å ortorektifisere de hyperspektrale bildene. HySpex sensorene har ikke en integrert IMU. Det ble derfor valgt å montere Leica sensoren og Hyspex sensoren sammen på en stiv plate, slik at de eksakt samme vibrasjonene og rotasjonene kunne måles av laserens IMU. Dette ble gjort for presis georeferering av de hyperspektrale bildene. Riegl VQ-1560i ble brukt parallelt ettersom denne sensoren har høyere punkttetthet enn Leica ALS70.

Atmosfærisk korreksjon

I tillegg til radiometrisk og geometrisk korrigering, har det også blitt utført en atmosfærisk korrigering av de hyperspektrale bildene. En atmosfærisk korreksjon vil transformere dataen fra radians til reflektans. Den atmosfæriske korreksjonen brukt i denne studien, ble utført av Agnieszka Kuras i ATCOR4. Atmosfærisk korreksjon er nødvendig for å fjerne eller redusere atmosfæriske effekter på bildene, forårsaket av molekyler og spredning av aerosol i atmosfæren (Yang et al. 2021).

3.1.2 Prosjektområde



Figur 3.2: Tildelt prosjektområde. 04, 05 og 07 representerer de ulike flystripene. De tre bildeflisene som er illustrert benyttes i denne studien.

Fjernmålingsdata for et område på 37.4km^2 , utgjør en stor lagringsplass. I denne studien ble det derfor gitt tre ulike fliser fra tre flystriper, for å begrense problemer knyttet til datamaskinens minne. Disse tre flisene blir heretter omtalt som prosjektområdet i denne oppgaven. Flisene er tatt ut fra flystripe 04, 05 og 07, se figur 3.2. Laserdata for det samme området ble klippet og gitt. Prosjektområdetets størrelse tilsvarer omtrent 0.57km^2 og tar opp 11.4 GB i lagringsplass. Tabellen 3.2 viser de tre områdene med bildets dimensjoner for VNIR og SWIR. Dimensjonene er gitt slik; bredde, lengde, antall bånd. Bredde og lengde er gitt i piksler.

Prosjektområdet er et urbant område i Høvik som byr på mye variasjon med nye og gamle asfalterte veier, ulike hustakmaterialer, vegetasjon, og områder med og uten skygge. Målet er at en maskinlæringsmodell skal bli lært opp til å segmentere et slikt område.

Tabell 3.2: Oversikt over de tre bildenes dimensjoner gitt som (bredde, lengde, spektralbånd).

	Område		
	04	05	07
VNIR (30cm GSD)	783 x 2727 x 176	663 x 2743 x 176	813 x 2630 x 176
SWIR (70cm GSD)	336 x 1169 x 222	284 x 1176 x 222	349 x 1127 x 222

3.2 Maskinvare og programvare

Hyperspektrale bilder og laserdata opptar stor lagringsplass. Da dette skulle leses inn i minnet til datamaskinen, oppstod det derfor problemer. Grunnet begrensninger med egen datamaskin, ble Google Colab Pro tatt i bruk for å trene og teste maskinlæringsmodeller. Colab Pro er en betalingstjeneste fra Google som tilgjengeliggjør RAM og toppmoderne grafikkprosessorer. I dette verktøyet kan det kjøres Pythonkode fra en nettleser som blir kjørt i skytjenesten til Google. Tabell 3.3 viser spesifikasjoner til lokal datamaskin og Google Colab Pro.

Tabell 3.3: Oversikt over maskinvarespesifikasjoner.

	Lokal datamaskin	Colab Pro
GPU	Geforce MX-150	K80, T4 eller P100
CPU	Intel Core i5-8250U @ 1.60 GHz	Intel(R) Xeon(R) CPU @ 2.20GHz
RAM	8 GB	24 GB
Begrenset kjøretid	Nei	Ja (24 timer)

Store deler av arbeidet knyttet til denne masteroppgaven ble utført i Python. Python ble brukt for analyse og modellbygging. Tabell 3.4 viser en oversikt over pakkene som ble benyttet i Python for innlesning, bearbeiding, modellbygging og grafisk fremstilling av resultater. Geomatica Focus ble primært brukt for visualisering av hyperspektrale bilder. I tillegg har QGIS blitt benyttet til å bearbeide terrengmodeller. For visualisering av laserdata, ble Quick Terrain Modeler benyttet.

Tabell 3.4: Oversikt over Python-pakker og versjon som ble benyttet.

Innlesning	Bearbeiding	Modelbygging	Grafisk fremstilling
Spectral (v 0.22)	Numpy (v 1.21.6)	Keras (v 2.8.0)	Matplotlib (v 3.2.2)
Pyrsgis (v 0.4.19)	Patchify (v 0.2.3)	TensorFlow (v 2.8.0)	Seaborn (v 0.11.2)
	Scikit-learn (v 1.0.2)	Segmentation models (v 1.0.1)	

3.3 Databehandling

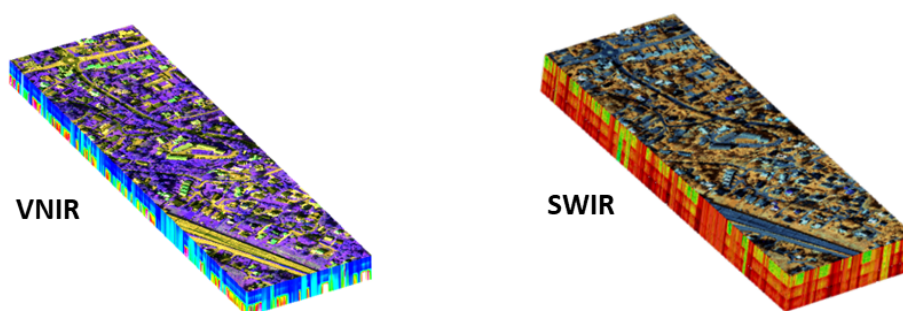
3.3.1 Datasammenslåing

Sammenslåing av data ble utført for hele prosjektområdet med lik fremgangsmåte. VNIR, SWIR og normalisert høydemodell ble slått sammen til en datakube for hvert område.

VNIR og SWIR

Datasettet som blir brukt i denne studien, inneholder VNIR i 30cm romlig oppløsning og SWIR i 70cm romlig oppløsning. VNIR og SWIR inneholder informasjon fra ulike deler av det elektromagnetiske spekteret. Bånd bestående av bølgelengder i intervallene, [1.35,1.42] og [1.8,1.98] ble fjernet grunnet lav transmisjon av strålingen gjennom vann i atmosfæren (Griffin & Burke 2003). For å få størst mulig del av spektralsignaturen til de ulike materialene, ble det besluttet at de resterende båndene fra VNIR og SWIR skulle slås sammen ("stacked") og benyttes. For å kunne slå sammen båndene måtte den romlige oppløsningen være lik. Videre ble det valgt å bruke 30cm romlig oppløsning ettersom dette ga til minst mulig tap av informasjon om pikslene. Dermed ble oppløsningen til SWIR resamplet ned til 30cm romlig oppløsning ved hjelp av "PAN-sharpening" før sammenslåingen. Resamplingen ble utført av Agnieszka Kuras. Figur 3.3 viser hyperkubene til VNIR og SWIR for område 04.

En utfordring ved å resample til høyeste romlige oppløsning er at nye verdier beregnes ved hjelp av estimering (Harris Geospatial Solutions 2019). Estimering av pikselverdier vil ikke nødvendigvis være korrekt og kan inneholde feil. Likevel blir det valgt å resample den romlige oppløsningen til 30cm i denne masteroppgaven, for å kunne utnytte den høyeste romlige oppløsningen i VNIR. Dermed kan det å gjøre om VNIR til 70cm romlige oppløsning ses bort ifra.

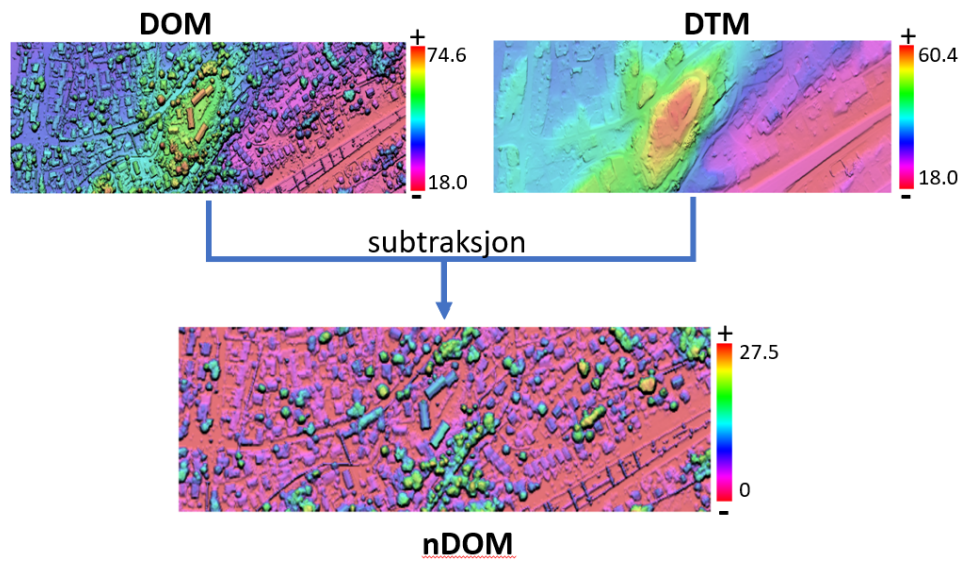


Figur 3.3: Hyperkuber for VNIR og SWIR som tilhører område 04.

Normalisert høydemodell

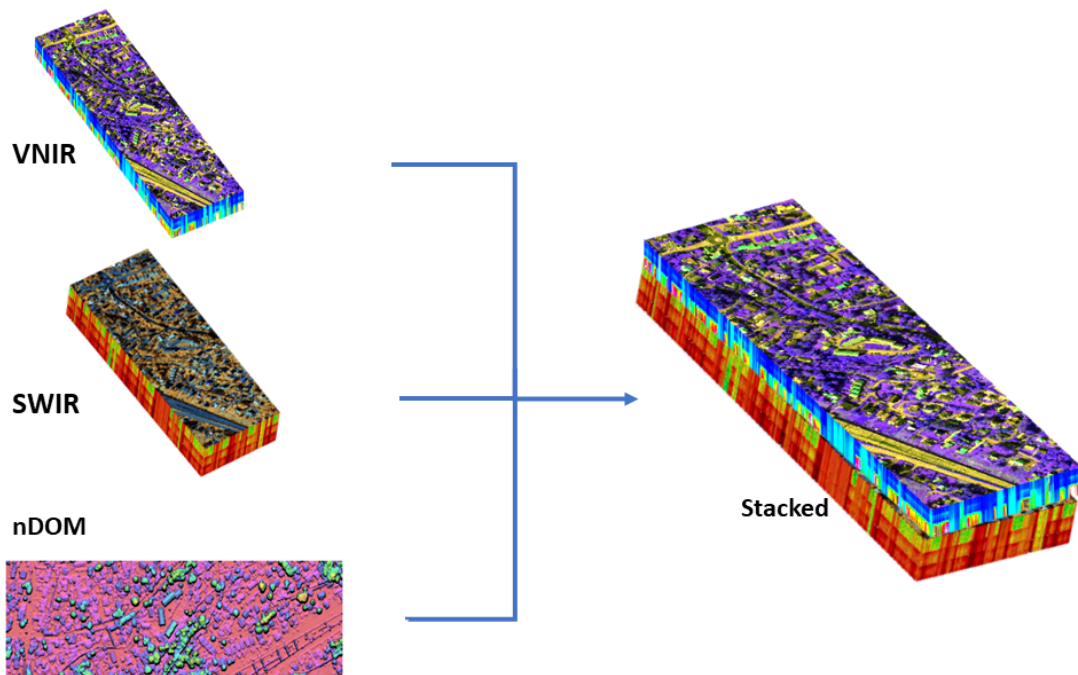
Normalisert høydemodell (nDOM) blir produsert ved å trekke en terrengmodell fra en overflatemodell. Figur 3.4 viser DOM, DTM og nDOM. En normalisert høydemodell viser høyden over bakken for de ulike objektene på bildet. Dette laget legges til i hyperkuben, som består av VNIR og SWIR. Hensikten med å legge inn nDOM er at en maskinlæringsmodell vil klare å bruke informasjonen om høyde til å skille mellom veier og bygninger, og dermed gi mer korrekte

predikeringer.



Figur 3.4: Ekstrahering av nDOM fra DOM/DTM. Høydemodellene er farget etter høyde "Shaded relief" og representerer område 04.

Figur 3.5 viser hvordan den endelige datakuben vil se ut etter "stacking" av VNIR, SWIR og nDOM for et område.



Figur 3.5: Sammenslåing av VNIR, SWIR og nDOM som resulterer i én stor hyperkub.

3.3.2 Skyggemaske

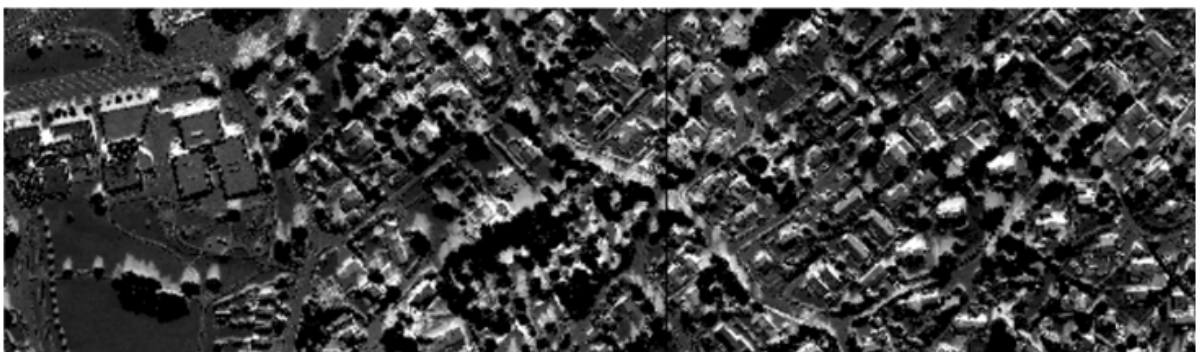
Som en del av forskningsprosjektet, "FKB Maskinl ring" ble det generert en skyggemaske av Maximilian Brell. Denne skyggemasken blir brukt videre i denne masteroppgaven. Skyggemasken ble generert for flystripe 07 med romlig oppl sning p  70cm av SWIR og LiDAR data, se figur 3.6. Pikseldverdiene p  masken strekker seg fra 1 til 43 og representerer graden av skygge. En lav verdi angir ingen skygge/lite skygge og en h y verdi representerer mye skygge.

Skyggemasken ble videre klippet ut til samme st rrelse som prosjektomr det, og brukt til   trene opp en U-net modell for   predikere flere skyggemasker for hele prosjektomr det. Dermed kunne denne masken benyttes som fasitdata, ved generering av skyggeklasser til prosjektomr det. Det ble ogs  testet med   legge til skyggemasken som et b nd til datakuben. Skyggemasken, produsert av U-net, har en romlig oppl sning p  30cm. Ettersom 30cm oppl sning ble valgt i denne studien, blir denne oppl sningen brukt videre i analysen. Hensikten med   legge til skyggemasken er   se om maskinl ringsmodellen klarer   bruke denne informasjonen til   skille mellom veier og bygninger som er dekket av skygge.

Figuren 3.6 viser prosjektomr de 07(fig a) i 70cm romlig oppl sning og skyggemasken som ble generert av Maximilian Brell(fig b).



a) 70cm(GSD) HS bilde, omr de 07

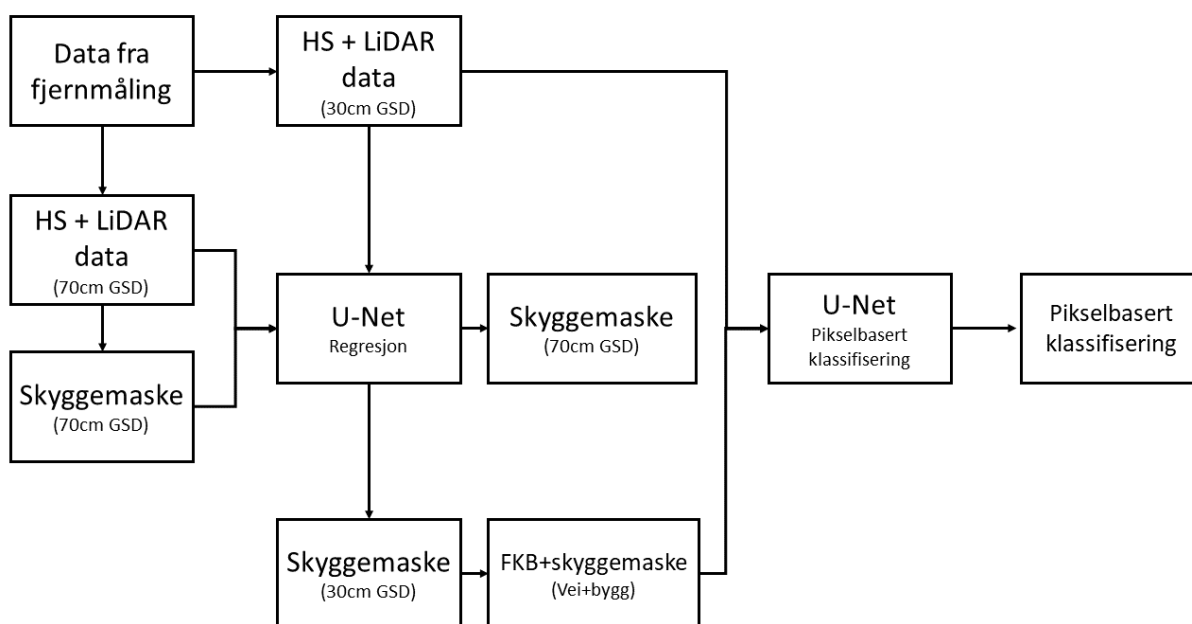


b) 70cm(GSD) skyggemaske, omr de 07

Figur 3.6: Omr de 07 er gitt i figur a og skyggemasken produsert av Maximilian Brell for tilsvarende omr de er gitt i figur b.

3.4 Arbeidsflyt

Figur 3.7 viser arbeidsflyten for denne oppgaven. Arbeidsflyten kan deles inn to deler; prediksjon av skyggemaske og pikselbasert klassifisering. Data innsamlet fra fjernmåling inneholder to romlige oppløsninger og begge blir benyttet. Datasettet på 70cm romlig oppløsning brukes til å trene en U-net modell til å predikere skyggemaske på 30cm romlig oppløsning. I denne studien benyttes skyggemasken fra Maximillan Brell som fasit. Skyggemasken predikeres for de tre områdene i prosjektområdet og blir videre benyttet som fasit for å generere skyggeklasser. Datasettet på 30cm romlig oppløsning blir brukt til å trene en U-net modell for pikselbasert klassifisering. I denne studien benyttes FKB-data for pikselbasert prediksjon.



Figur 3.7: Flytdiagrammet for dette arbeidet

3.5 Eksperimenter

For å utforske forskningsspørsmålet nærmere, blir tre ulike eksperimenter gjennomført i denne studien. Eksperimentene blir utført for å kartlegge om innføring av skyggeklasser, vil føre til at dyplæringsmodellen klarer å klassifisere veier og bygninger bedre, enn uten skyggeklasser. Formålet med dette er å finne hvilket av de tre ulike eksperimentene som gir best resultat av segmenteringen. U-net arkitekturen blir brukt i samtlige eksperimenter.

Eksperiment 1(E1): Klassifisering av veier, bygg og annet.

En U-net modell ble satt opp for segmentering av veier, bygg og annet. Område 04 og 05 ble brukt til å trene, validere og teste modellen. Etter optimalisering av modellen ble det usette området 07 predikert.

Eksperiment 2(E2): Innføring av skyggeklasse. Klassifisering av veier, bygg, skygge og annet.

I likhet med E1 ble den samme modellen og metodikken brukt for trening og predikering av det usette området i E2. Imotsetning til E1, ble det innført en ny klasse som modellen skulle lære og deretter klare å predikere. Ved hjelp av skyggemasken som ble generert kunne det innføres en egen klasse for skygge. Modellen predikerte da totalt fire klasser. Formålet dette var å undersøke om den nye klassen ville få modellen til å segmentere bildet bedre enn i E1.

Eksperiment 3(E3): Innføring av to skyggeklasser. Klassifisering av veier, bygg, skygge på veier, annen skygge og annet.

I E3 ble det innført nok en klasse og modellen predikerte da til sammen fem klasser. Ved hjelp av skyggemasken kunne ”skygge på veier” filtreres ut og bli lagt til som en egen klasse. Formålet med å introdusere denne nye klassen, var å øke andelen rette prediksjoner av veier. Etter prediksjonen, ble det utført en post-prosessering. I post-prosesseringen ble klassene som representerer veier slått sammen. Deretter ble det foretatt en ny evaluering av modellens prediksjonsevne. Dette resultatet kunne videre sammenlignes med resultatene fra E1 og E2.

3.6 Generering av treningsdata

I denne oppgaven blir det brukt styrt maskinl ring ("supervised classification"). De ulike maskinl ringsmetodene blir beskrevet under kapittel 2.5. For   kunne benytte en styrt maskinl ringsmodell, m  det dannes en fasit for hele prosjektområdet. Hvert bilde som skal klassifiseres m  ha en maske som skal representere hvilken klasse de ulike pikslene tilh rer. I denne masteroppgaven blir FKB-data for vei og bygg benyttet som fasit.

De tre omr dene i prosjektet blir delt opp i mindre bildekvadrater. Fordelen med   dele opp et st rre bilde i mindre bildekvadrater, er at beregningstiden og beregningskostnaden av et konvolusjonelt nevralt nettverk vil bli mindre. En annen fordel er at mindre bildekvadrater kan gi  kt n yaktighet av predikering enn ved bruk av et st rre bilde. (Sharma et al. 2017)

I studier som (Sharma et al. 2017) og (Alshehhi et al. 2017), blir bilder samlet inn ved fjernm ling, delt opp i mindre bildekvadrater. Dermed kan det fanges opp naboeffekter til pikselen som skal klassifiseres, slik at en modell enklere kan trekke ut "low-level"-informasjon fra et bilde f r pikselen blir klassifisert. Dette blir forklart i delkapittel 2.5.2. I artikkelen (Sharma et al. 2017), blir det oppn dd 20% h yere n yaktighet ved klassifisering av bildekvadrater i forhold til et st rre bilde. I denne masteroppgaven blir det derfor testet fire ulike bildekvadratst rrelser; 32x32, 64x64, 128x128 og 256x256.

Figur 3.8 viser et eksempel p  hvordan de ulike bildekvadratst rrelsene vil se ut for omr de 04, og hvor stort omr de de dekker p  bakken.



Figur 3.8: Ulike bildekvadratst rrelser vises p  fig b(32x32),c(64x64),d(128x128),e(256x256). De fire st rrelsene er representert sammen med omr de 04 i fig a.

3.6.1 Treningsdata for predikering av skyggemaske

Område 07 med 70cm romlig oppløsning bestående av VNIR, SWIR og nDOM, blir brukt som treningsdata. Ettersom dette er av lavere romlig oppløsning enn 30cm, er det også færre piksler i bildet med 70cm romlig oppløsning. Dette resulterer i mindre treningsdata. Treningsdataen blir delt opp i form av trening, validering og testsett. Omtrent 60% av dataen blir brukt til trening, 20% til validering og 20% til testing.

3.6.2 Treningsdata for pikselbasert klassifisering

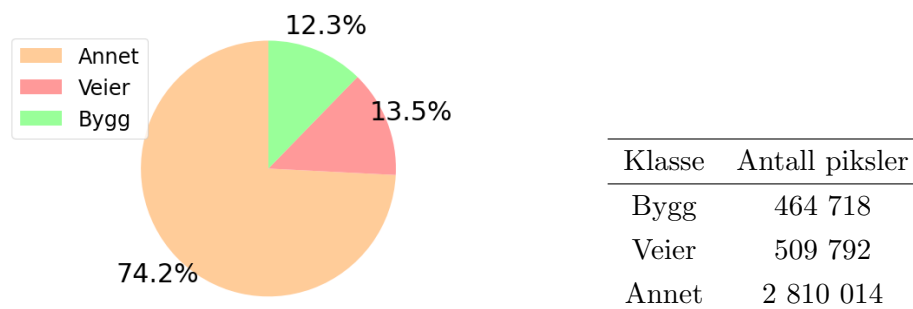
Som treningsdata blir områdene 04 og 05 med 30cm romlig oppløsning brukt for pikselbasert klassifisering. Disse områdene består også av VNIR, SWIR og nDOM, som totalt utgjør 399 bånd. Treningsdataen blir delt opp i trening, validering og testsett på samme måte som treningsdata for predikering av skyggemaske. For pikselbasert klassifisering, blir det generert fire ulike bildekvadratstørrelser, se tabell 3.5. Tabellen viser dimensjonene til de ulike settene gitt ved; antall bildekvadrater, kvadratstørrelse i bredde, kvadratstørrelse i lengde og antall bånd.

Tabell 3.5: Dimensjonene til de ulike oppdelingene av treningsdata for pikselbasert klassifisering er gitt som (antall bildekvadrater, kvadratstørrelse i bredde, kvadratstørrelse i lengde og antall bånd).

Område 04 og 05 kombinert				
	32 x 32	64 x 64	128 x 128	256 x 256
Trening	(2364, 32, 32, 399)	(591, 64, 64, 399)	(147, 128, 128, 399)	(32, 256, 256, 399)
Validering	(592, 32, 32, 399)	(148, 64, 64, 399)	(37, 128, 128, 399)	(8, 256, 256, 399)
Testing	(740, 32, 32, 399)	(185, 64, 64, 399)	(47, 128, 128, 399)	(10, 256, 256, 399)

Generering av fasitdata for eksperiment 1

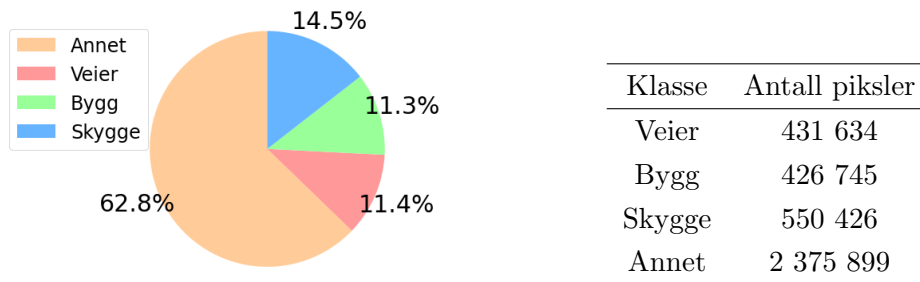
For E1 blir FKB-dataen kun brukt til å generere klassene, "veier" og "bygg". Figur 3.9 viser en tabell og et kakediagram. Tabellen viser antall piksler som tilhører hver klasse og kakediagrammet viser andelen til de tre klassene i E1. For dette eksperimentet dominerer klassen "annet", som nesten har fem ganger flere piksler enn de andre klassene. Dette blir tatt hensyn til under trening av modellen.



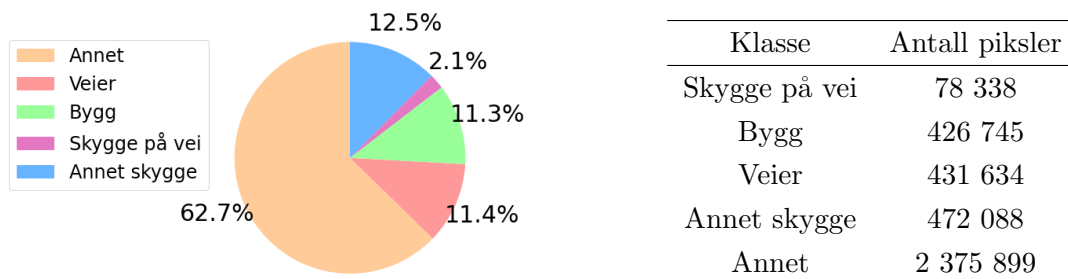
Figur 3.9: Et kakediagram for eksperiment 1 som viser andelen av de ulike klassene og en tabell som viser antall piksler per klasse i hvert bånd.

Generering av fasitdata for eksperiment 2 og 3

I både E2 og E3 blir det innført skyggeklasser. Framgangsmåten er lik for begge eksperimentene. Generering av skyggeklasser blir utført for hele prosjektområdet. Dette kunne først gjøres etter at det ble produsert skyggemaske for hvert område. Skyggemaskene inneholder støy, og det blir derfor valgt å sette en grenseverdi for å luke ut dette. Figur 3.10 viser andelen av de ulike klassene i E2 og figur 3.11 viser andelen av de ulike klassene i E3. I begge disse eksperimentene dominerer klassen "annet" når det gjelder antall piksler.



Figur 3.10: Et kakediagram for eksperiment 2 som viser andelen av de ulike klassene og en tabell som viser antall piksler per klasse i hvert bånd.



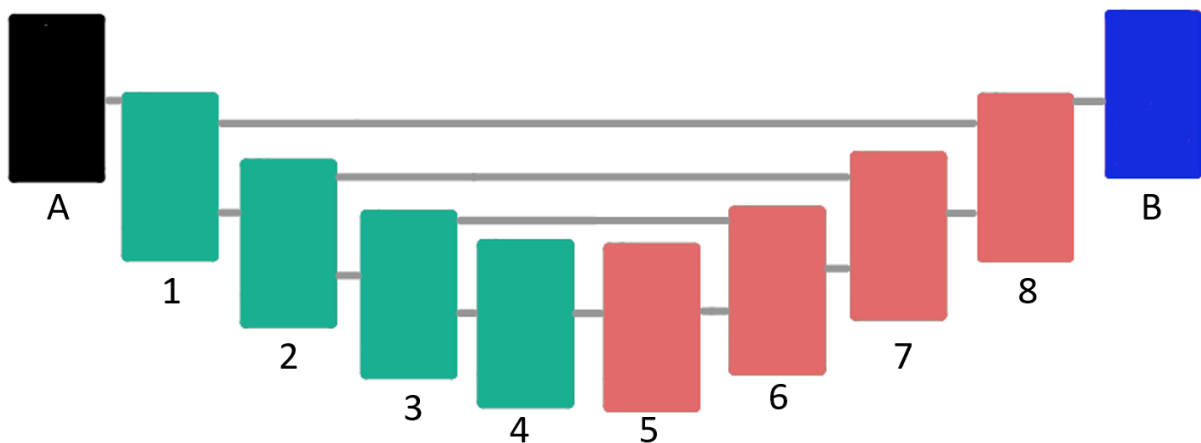
Figur 3.11: Et kakediagram for eksperiment 3 som viser andelen av de ulike klassene og en tabell som viser antall piksler per klasse i hvert bånd.

3.7 Oppsett av fullstendig konvolusjonelle nettverk

I denne studien blir det benyttet en U-net arkitektur for produksjon av skyggemasker og klassifisering av piksler. Figur 3.12 viser arkitekturen for begge modellene. U-net arkitekturen som blir benyttet, bygger videre på modellen til O. Ronneberger (Ronneberger et al. 2015). For begge modellene, blir "padding" satt til "same". Dette gir lik romlig dimensjon for utgangsbildet og inngangsbildet. Dette blir nærmere forklart i delkapittel 2.5.3. Det blir lagt til "drop-out" med 20% i hver konvolusjonell blokk. "Drop-out" stopper en viss mengde informasjon fra å lære videre, ved å ikke la den bevege seg videre i nettverket. Dette legges til for å regularisere modellen, slik at den ikke blir overtilpasset.

For produksjon av skyggemasker blir modellen satt opp for å predikere kontinuerlige verdier. Modellen blir satt opp slik at hver piksel får en verdi som representerer graden av skygge på den pikselen. Dette blir gjort ved at det siste laget med aktiveringsfunksjon blir byttet til en lineær aktiveringsfunksjon, slik at verdiene ikke blir presset ned til en klasse. Når modellen er ferdiglært vil den predikere en skyggemaske som har samme dimensjoner i lengde og bredde som inngangsbildet.

På samme måte som for predikering av skyggemaske, vil modellen for pikselbasert klassifisering predikere hver piksel til bildet som skal klassifiseres. Pikselene som blir predikert blir tilegnet en klasse. Etter at alle pikslene er predikert vil modellen produsere et segmentert bilde, der de ulike pikslene representerer en klasse. Bildet som blir produsert vil ha samme dimensjoner i lengde og bredde som inngangsbildet.



Figur 3.12: Illustrasjon av u-net arkitekturen som blir benyttet i denne studien. Figuren består av blokk A (inngangsbildet), B (prediksjonen), fire nedskalingsblokker (grønn) og fire oppskalingsblokker (rosa). Hver nedskalingsblokk består av to konv-filtre, "dropout, maks-pooling og "skip connection", unntatt blokk 4 som ikke inneholder maks-pooling. Hver oppskalingsblokk består av to konv-filtre, "dropout, opp-konv og sammenslåing fra "skip connection", og blokk 8 inneholder et ekstra lag med konv-filtre.

3.7.1 Trening av nettverk

Før trening av nettverket blir iverksatt, blir hvert bildekvadrat normalisert. Dette gjøres for å skalere pikselverdiene slik at de blir tilegnet verdier mellom 0 og 1. Dermed kan modellen konvergere raskere (Raschka & Mirjalili 2019, p.125). Under trening av nettverk, vil bildene flyte

gjennom U-net arkitekturen fram og tilbake. Her vil de ulike lagene i U-net trekke ut informasjon, og stadig lære seg å gjenkjenne mønstre. En passering av hele datasettet fram og tilbake i et nettverk kalles en epoke. Prosedyren for nettverkstrening er lik for predikering av skyggemaske og pikselbasert klassifisering. For begge metodene blir det satt opp ulike modellkonfigurasjoner som så blir testet. Den beste modellkonfigurasjonen som blir funnet, testes videre ved å endre antall epoker, treningsrate og batch-størrelse. Dette gjøres ved å sette opp sjekkpunkter som måler endringen i tap, som modellen produserer etter hver epoke. Treningen stoppes hvis tapet blir dårligere.

For å utforske hvilken bildekvadratstørrelse som resulterer i best predikering, blir de fire ulike bildekvadratstørrelsene satt opp med tre modellkonfigurasjoner. Totalt utgjør dette 12 modeller. Hver modell blir kjørt fem ganger med 50 epoker, gjennomsnittet av "Jaccard index", "F1-score" og tapsfunksjonen ble brukt til å evaluere konfigurasjonene. Grunnet begrensninger med datamaskinminne blir kun halvparten av treningsdataen brukt for å teste de ulike modellene. Den beste modellen som blir funnet, blir trent med hele treningsdatasettet.

Predikering av skyggemaske

For predikering av skyggemaske er det kun et område for trening og testing av modellen. Grunnet lite data blir det valgt å trene lenge med lav treningsrate, slik at modellen sakte men sikkert klarte å lære mest mulig før konvergeringspunktet nås. Bildekvadratstørrelsen som resulterer i best predikering, blir videre satt opp med to modellkonfigurasjoner for valg av optimaliseringsfunksjon. Begge modellkonfigurasjonene blir trent 2000 epoker hver, med Adam og SGD. Det blir valgt å bruke midlere kvadratisk feil("Mean squared error"), MSE som tapsfunksjon. Dette blir valgt siden denne modellen predikerer kontinuerlige verdier. Predikeringsevnen til modellen blir målt av R-kvadrert("R-squared") og RMSE("Root mean squared error").

Pikselbasert klassifisering

For pikselbasert klassifisering blir "Dice loss" brukt som tapsfunksjon. "Dice loss" blir brukt for å adressere problemet med ubalanserte klasser. "Dice loss" er en lossfunksjon som presterer godt på datasett som inneholder ubalanserte klasser, ved å måle overlapp av prediksjonen mot fasiten (Fidon et al. 2017).

De tre eksperimentene blir trent i 150 epoker hver, med den beste bildekvadratstørrelsen. Etter at den beste modellen har blitt trent opp for hvert eksperiment, predikeres område 07 på modellen. Område 07 holdes separert fra trening og testing fram til den beste modellen er funnet. Dette er fordi det er ønskelig med en god generalisert modell som ikke er overtilpasset. Hver gang område 07 testes, så slippes informasjonen ut til modellen. Ettersom overtilpasning kan forekomme bør det kun predikeres med område 07, når den beste modellen er funnet.

Kapittel 4: Resultater

4.1 Trening av modellen

I masteroppgaven blir det utforsket med ulike bildekvadratstørrelser som beskrevet i delkapittel, 3.6. Tabell 4.1 viser evalueringen av de ulike bildekvadratstørrelsene. Optimaliseringsfunksjonen Adam gir best resultat for de fire kvadratstørrelsene, og kvadratstørrelsen 128x128 presterer best på samtlige evalueringer. Evalueringen for eksperiment 1 setter grunnlaget for hvilken størrelse som skal benyttes videre i denne studien. Ettersom predikeringen med kvadratstørrelsen 128x128 gir best resultat, brukes denne kvadratstørrelsen videre i eksperiment 2 og 3. Derfor blir kvadratstørrelse 128x128 også brukt ved predikering av skyggemasker.

Tabell 4.1: Test av de ulike modellkonfigurasjonene. Tabellen viser evaluering av trenings- og valideringssettet. Det blir målt tap, "Jaccard score", og "F1 score" for hver modellkonfigurasjon.

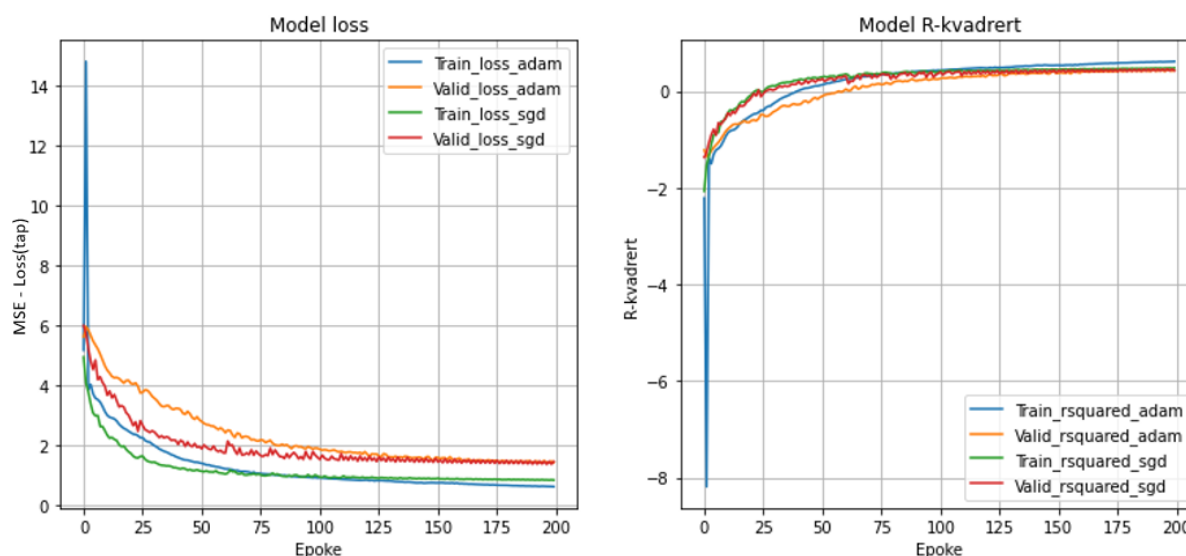
	Loss	Val.Loss	JAC	Val.JAC	F1	F1_val
kvadrat størrelse 32x32						
Adam	0.22120	0.28100	0.64643	0.57244	0.77879	0.71909
SGD	0.59197	0.59199	0.28762	0.28672	0.40802	0.40800
RMSprop	0.36562	0.38994	0.51023	0.47938	0.63437	0.61008
kvadrat størrelse 64x64						
Adam	0.22892	0.27296	0.63971	0.58311	0.77127	0.72732
SGD	0.60000	0.60885	0.27948	0.27294	0.39913	0.39124
RMSprop	0.47320	0.49842	0.42060	0.39131	0.52767	0.50189
kvadrat størrelse 128x128						
Adam	0.13162	0.25841	0.77208	0.60291	0.86901	0.74145
SGD	0.70852	0.74223	0.19189	0.19000	0.29209	0.29121
RMSprop	0.26942	0.31736	0.58755	0.53801	0.72549	0.69049
kvadrat størrelse 256x256						
Adam	0.20999	0.34798	0.66825	0.52068	0.79000	0.65201
SGD	0.62733	0.65948	0.26613	0.23957	0.37266	0.34051
RMSprop	0.24641	0.38500	0.62227	0.48220	0.75358	0.61499

4.1.1 Trening av modellen for predikering av skyggemaske

Predikering av skyggemaskene blir gjort ved å teste optimaliseringsfunksjonene, Adam og SGD. Adam og SGD blir testet fem ganger hver, med 200 epoker. Gjennomsnittet av de fem testene for hver metrikk vises i tabell 4.2. Modellen med Adam presterer best på "loss" og R-kvadrert på treningssettet, men ikke like godt på valideringssettet. Dette kan tyde på en overtilpasset modell. Modellen med SGD presterer best på valideringssettet og har lite avvik mellom trening og validering. Dermed blir SGD brukt som optimaliseringsfunksjon for prediksjon av skyggemaskene. Figur 4.1 viser en grafisk fremstilling av tap og R-kvadrert for hver epoke for begge optimaliseringsfunksjonene, i henhold til trening og validering.

Tabell 4.2: Evaluering av skyggeprediksjon ved bruk av optimaliseringsfunksjonene Adam og SGD. Snittet av 5 gjennomføringer med hver funksjon i 200 epoker presenteres.

	Loss	Val loss	RMSE	Val RMSE	R-kvadrert	Val R-kvadrert
Adam	0.62522	1.4281	3.5604	2.4382	0.61259	0.43610
SGD	0.84317	1.3729	2.2222	2.4428	0.47755	0.45789

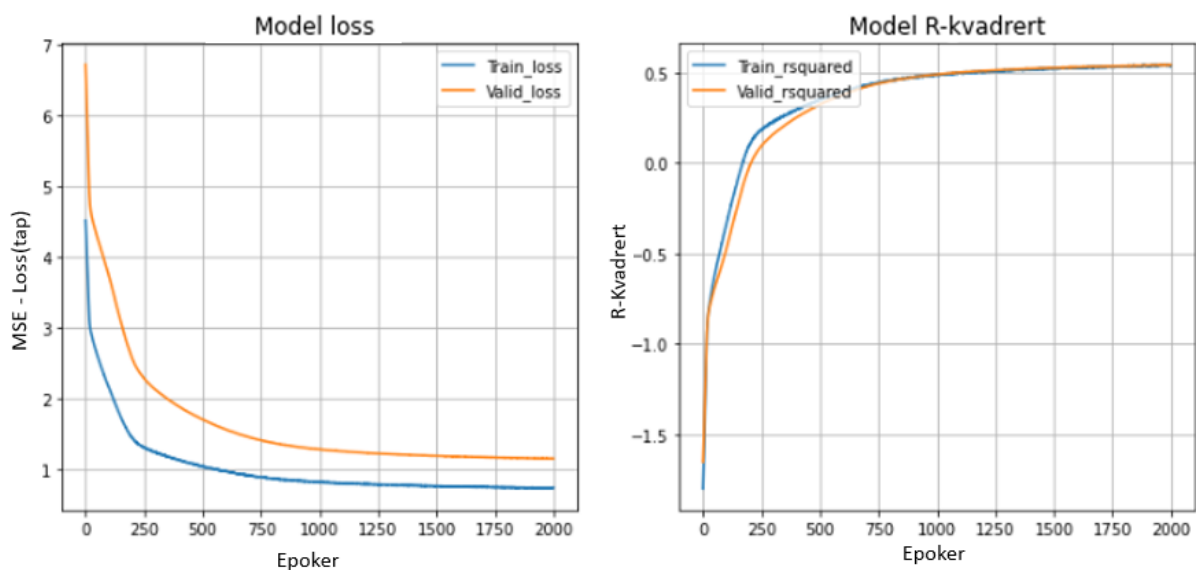


Figur 4.1: Grafisk fremstilling av trening og validering av modellen ved bruk av optimaliseringsfunksjonene, Adam og SGD. 200 epoker blir utført fem ganger og gjennomsnittet av de fem testene presenteres for begge funksjonene.

Den beste modellen for predikering av skyggemaskene blir oppnådd ved bruk av SGD med en læringsrate på 0.001. Modellen blir trent i 2000 epoker uten batch-størrelse. På grunn av lite treningsdata, klarer datamaskinen å lese inn all dataen før vektcoeffisientene blir oppdatert. Dermed blir batch-størrelsen satt til hele treningssettet. Tabell 4.3 viser evaluering av modellen på testsettet. Modellen måles til en R-kvadrert verdi på 0.56 og RMSE på 0.92 i skyggeverdi. Modellen klarer å beskrive 56% av variansen i datasettet og avviker gjennomsnittlig 0.92 i skyggeverdi. Figur 4.2 viser trening og validering av dataen, fig(a) viser kalkulasjon av loss per epoke og fig(b) R-kvadrert tall per epoke. Modellen presterer godt og det er lite avvik mellom trenings-, validerings- og testsettene.

Tabell 4.3: Modellevaluering på testdataen for predikering av skyggemasker.

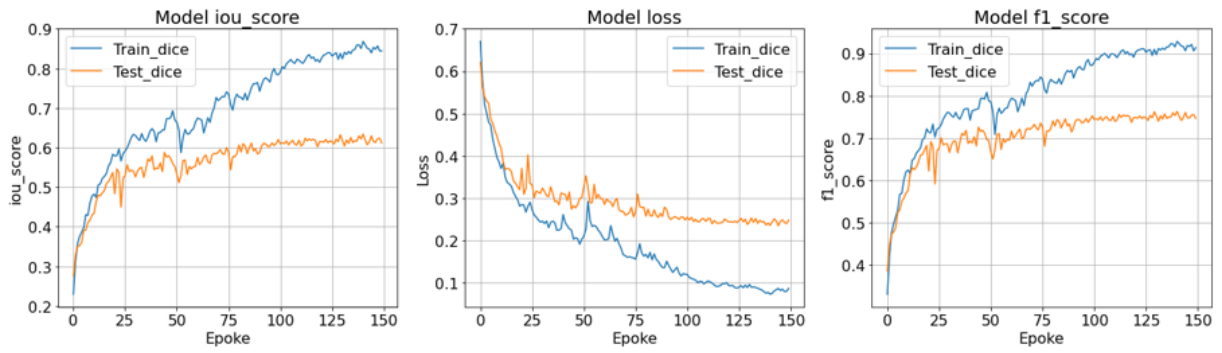
Loss	RMSE	R-kvadrert
0.85287	0.92351	0.56162



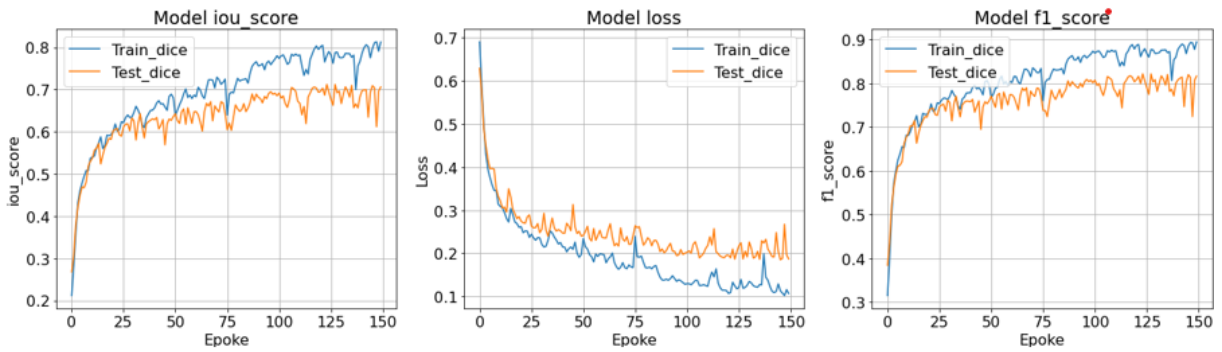
Figur 4.2: Grafisk fremstilling av trening og validering av den optimaliserte modellen for predikering av skyggemaske

4.1.2 Trening av modellen for pikselbasert predikering

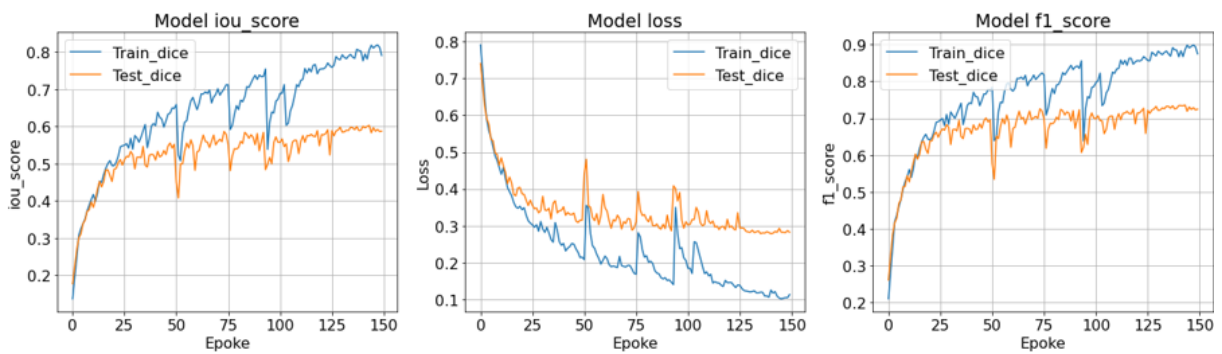
Den beste modellen for pikselbasert predikering blir oppdaget ved bruk av Adam med kvadratstørrelsen 128x128. Modellen blir trent i 150 epoker med en batch-størrelse på 16 bilder. For de tre eksperimentene brukes den beste modellen. Figur 4.3 viser trening og valideringskurver for de tre eksperimentene.



a.) Trening og validering av eksperiment 1



b.) Trening og validering av eksperiment 2



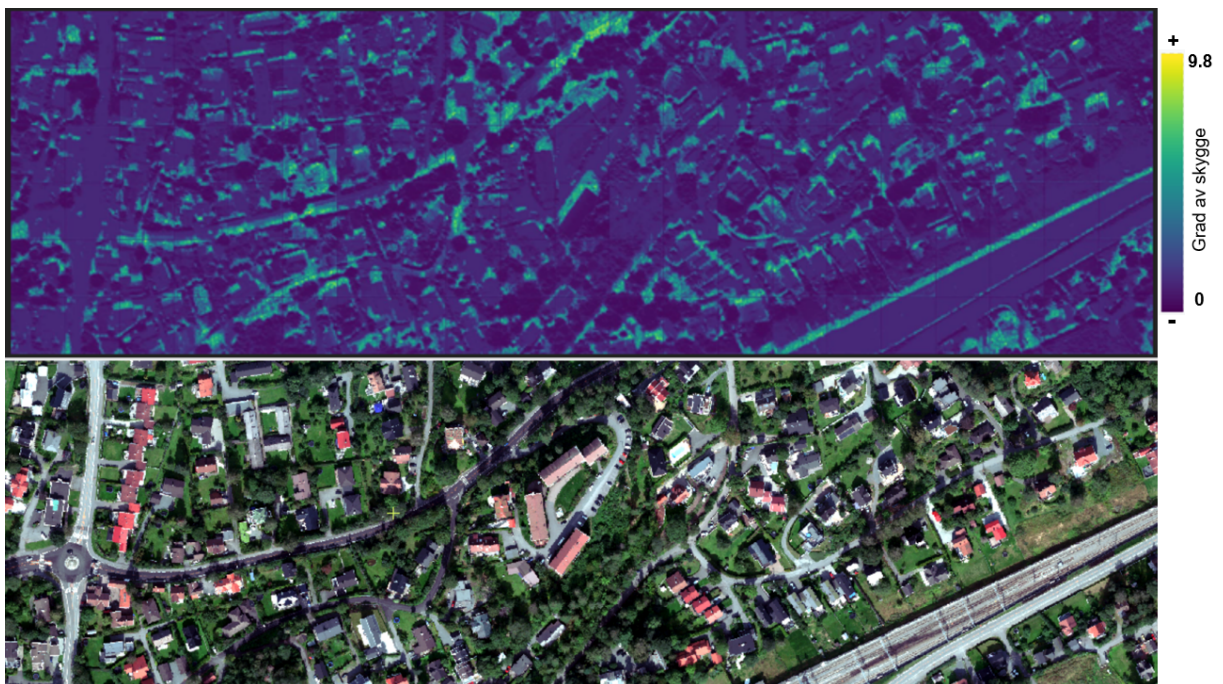
c.) Trening og validering av eksperiment 3

Figur 4.3: Grafisk fremstilling av trening og validering for modellen i de tre eksperimentene. E1 vises i a, E2 i b og E3 i c.

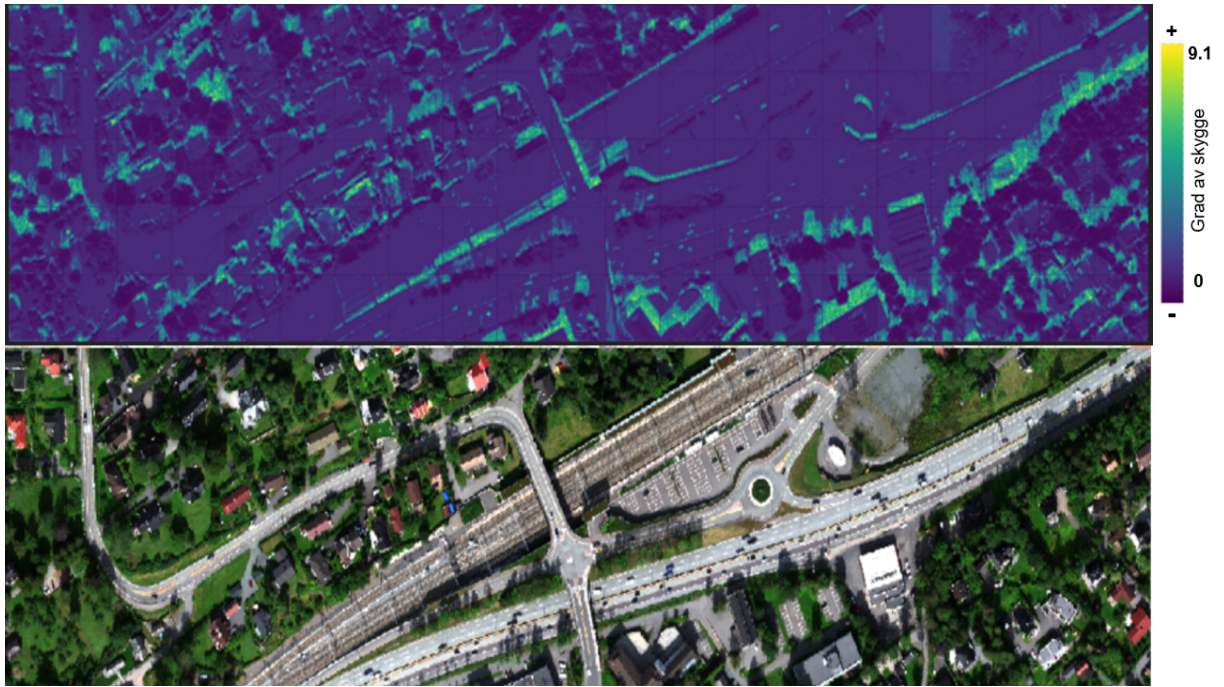
4.2 Prediksjon av skyggeomasker

Resultatene av predikerte skyggeomasker med 30cm romlig oppløsning presenteres i figurene 4.4, 4.5 og 4.6, og 70cm romlig oppløsning er gitt i vedlegg A i figur A.1 og A.2. Skyggeomaskene med 30cm romlig oppløsning brukes til generering av skyggeklasser ved pikselbasert klassifisering. Det blir predikert skyggeomasker for områdene, 04, 05 og 07. Skyggeomaskene er gitt i en fargeskala som representerer graden av skygge, og presenteres sammen med RGB-visualisering for tilsvarende områder. Lyse piksler indikerer sterkere skygge enn mørke piksler. For hver skyggeomasker som blir produsert, genereres et histogram som viser fordelingen av pikselverdiene som representerer graden av skygge.

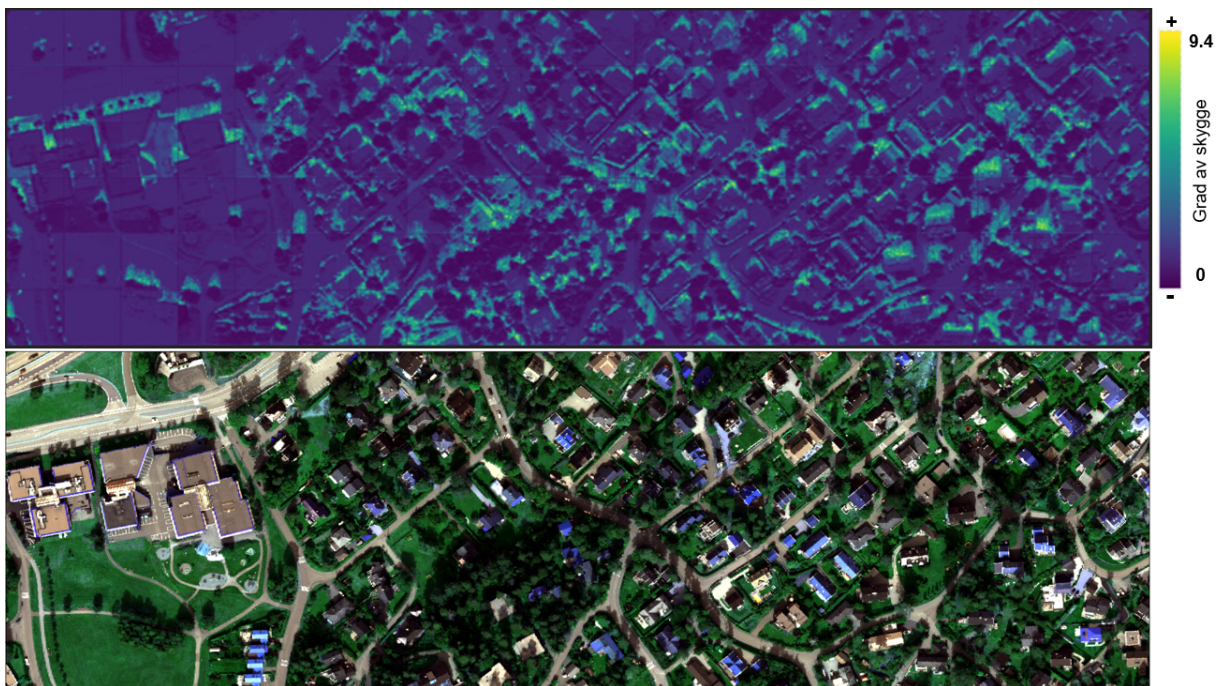
Predikering av skyggeomasker med 30cm oppløsning



Figur 4.4: Prediksjon av skyggeomasker med 30cm romlig oppløsning og RGB-visualisering av område 04.



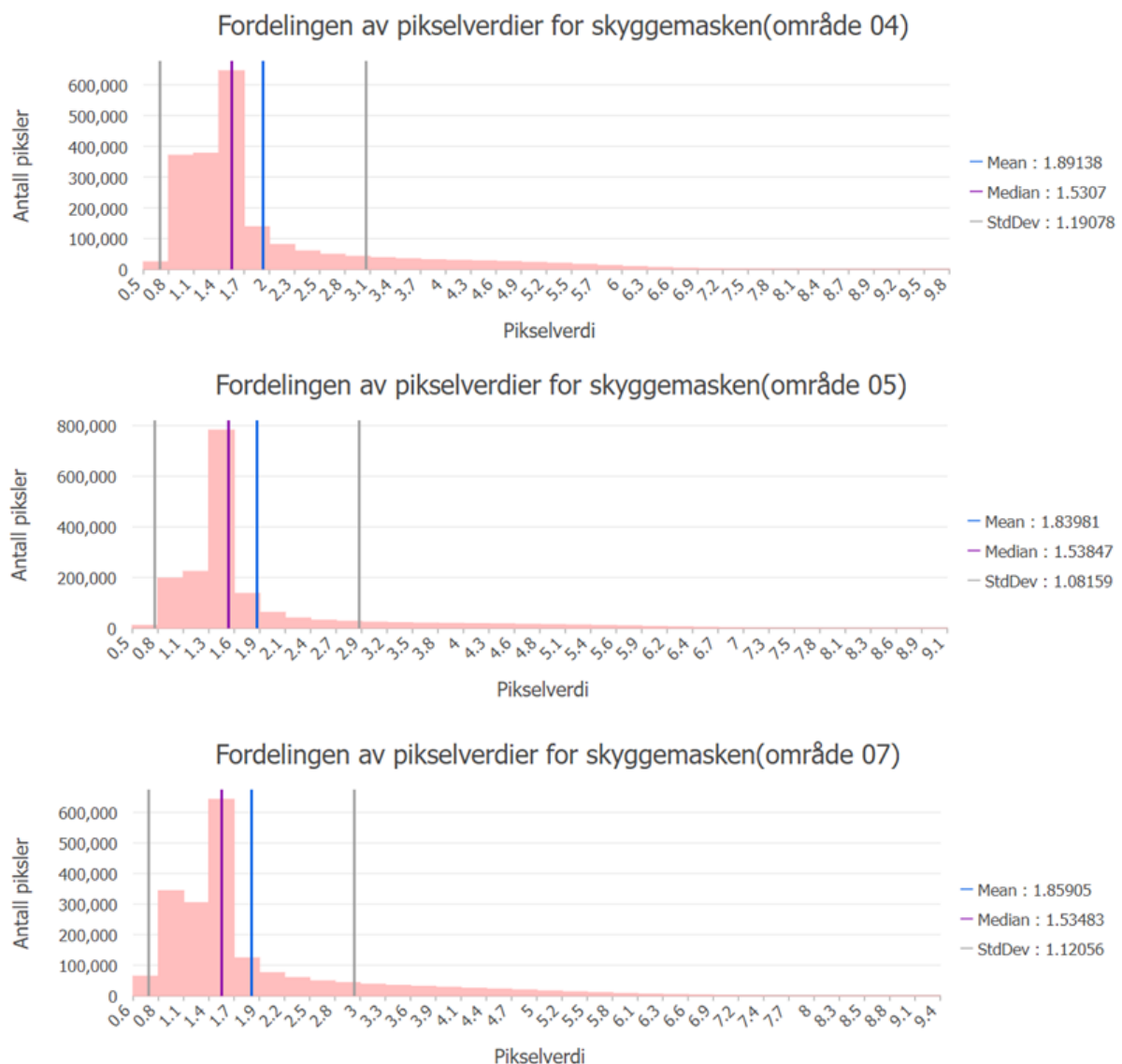
Figur 4.5: Prediksjon av skyggemaske med 30cm romlig oppløsning og RGB-visualisering av område 05.



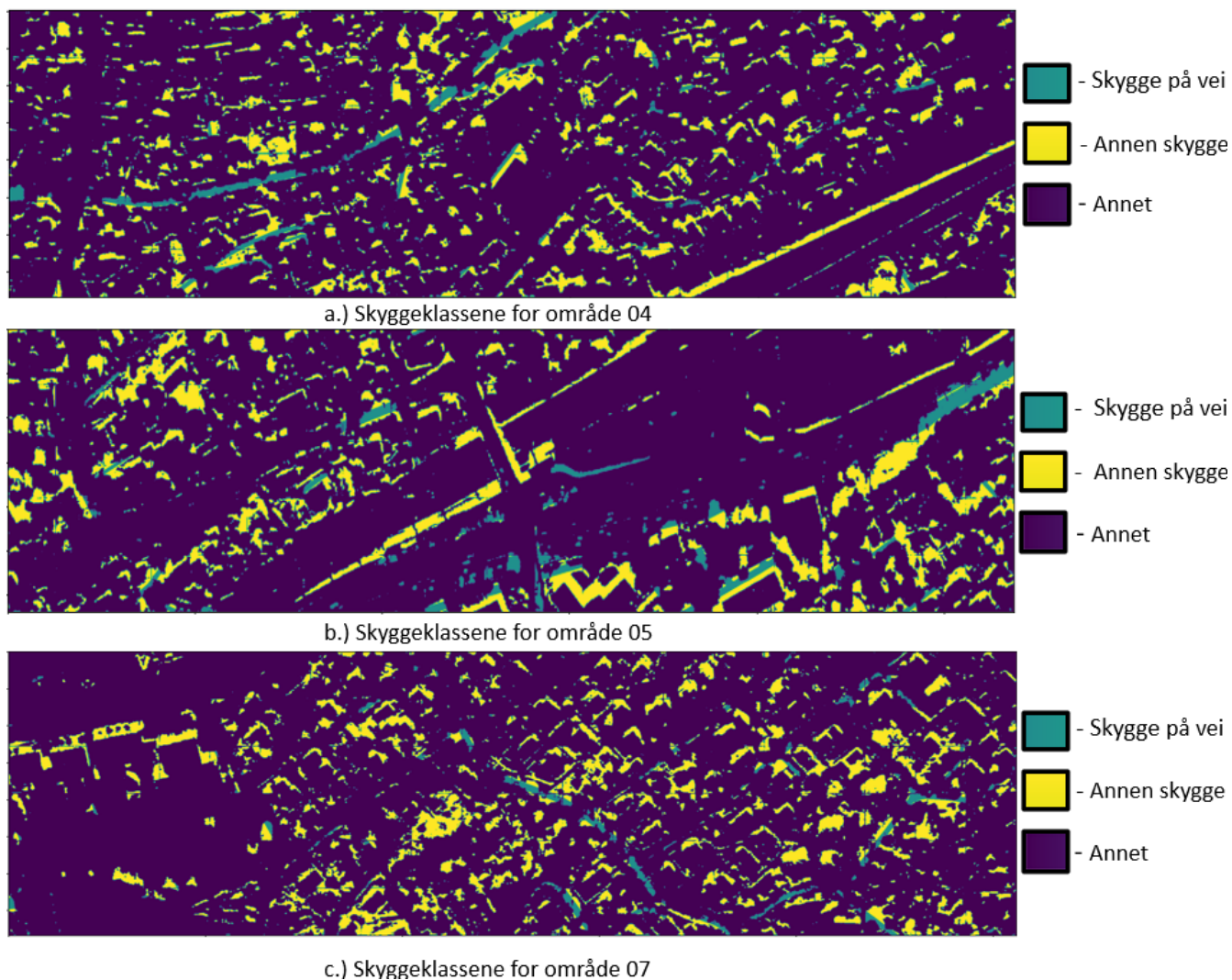
Figur 4.6: Prediksjon av skyggemaske med 30cm romlig oppløsning og RGB-visualisering av område 07.

Histogram for skyggemaskene

Fordelingen av pikselverdiene til de tre skyggemaskene er presentert i form av histogrammer i figur 4.7 for de tre områdene, 04, 05 og 07. De predikerte skyggemaskene inneholder omtrentlige verdier fra 0 til 10 som angir graden av skygge. Gjennomsnittlig pikselverdi på skyggemaskene er omtrent på 1.8 med et standardavvik på omtrent 1.12. For å luke vekk verdier som representerer liten grad av skygge blir det satt en grenseverdi før skyggeklassene blir generert. Ulike grenseverdier blir testet, før grenseverdien til slutt blir satt lik 3. Grenseverdi lik 3 utgjør gjennomsnittet addert med standardavviket. På grunn av gode resultater med grenseverdi lik 3 under testing, blir det valgt å bruke dette videre. Verdier lavere enn 3 blir satt lik 0 og verdier høyere enn 3 blir satt lik 1. Skyggeklassene som blir generert for eksperiment 3 vises figur 4.8, og eksperiment 2 vises i vedlegg A, figur A.3.



Figur 4.7: Fordeling av pikselverdier i de tre skyggemaskene med 30cm romlig oppløsning, visualisert i form av histogram.



Figur 4.8: Genererte skyggeklasser for eksperiment 3, der klassen ”skygge på vei” har blitt innført. Fig a viser skyggeklassene for område 04, fig b viser skyggeklassene for område 05, og fig c viser skyggeklassene for område 07

4.3 Pikselbasert klassifisering

Resultatene som presenteres for pikselbasert klassifisering gjelder for bildekvadratstørrelse 128x128. Denne størrelsen blir valgt fordi den presterer best under trening av modellen. For hvert eksperiment vises område 07 i RGB-farger, sammen med den genererte fasiten og resultatet av prediksjonen. For hvert eksperiment blir det produsert en forvirringsmatrise. Forvirringsmatrisen er normalisert for hver klasse slik at verdiene som presenteres i matrisen vises fra 0 til 1. Matrisen viser andelen piksler i de ulike klassene som blir predikert rett og galt.

Tabell 4.4 viser resultatene av prediksjonen for de tre eksperimentene på testområde 07 som beskrevet i kapittel 3.5. Eksperiment 1 presterer best på begge metrikkene som måles. Eksperiment 3 gir det nest beste resultatet og eksperiment 1 kommer dårligst ut av metrikkene som måles.

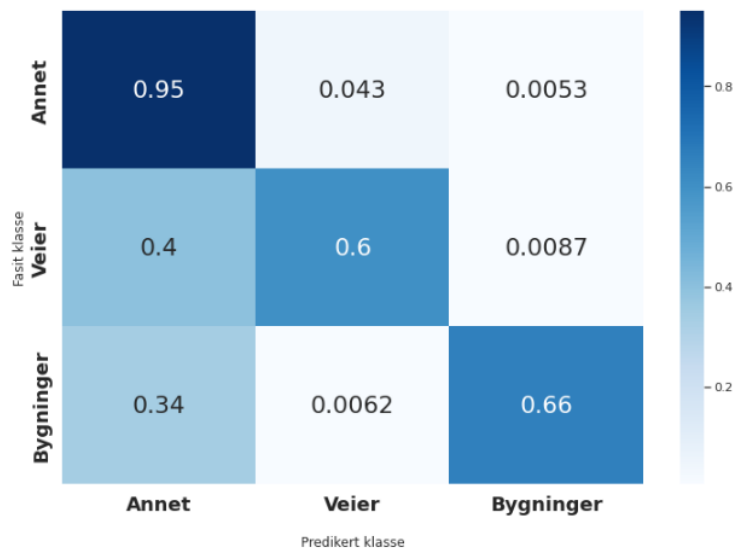
Tabell 4.4: Modellevaluering av prediksjonen på område 07 for de tre eksperimentene. Prediksjonen av område 07 er evaluert etter "Jaccard score" og "F1-score".

	Jaccard score	F1-score
Eksperiment 1	0.7665	0.8610
Eksperiment 2	0.6659	0.7902
Eksperiment 3	0.6950	0.8124

4.3.1 Klassifisering av veier og bygninger

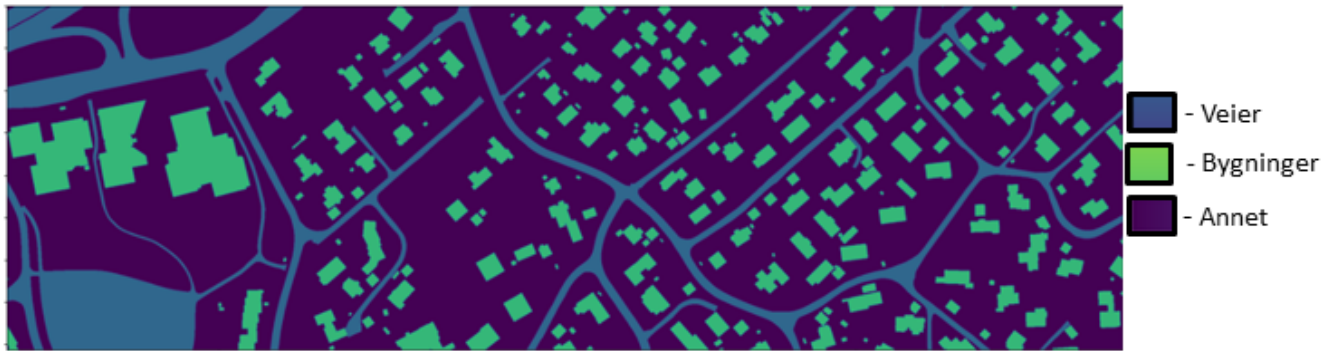
Prediksjon av E1

Figur 4.9 viser forvirringsmatrisen for E1. For klassen "annet" klarer modellen å predikere 96% rett. 60% av veiene blir klassifisert rett, mens 40 % av denne klassen blir predikert som "annet". Det samme gjelder for bygninger, der 66% blir predikert rett og resten av pikslene blir klassifisert som "annet".



Figur 4.9: Forvirringsmatrise for E1 som består av tre klasser.

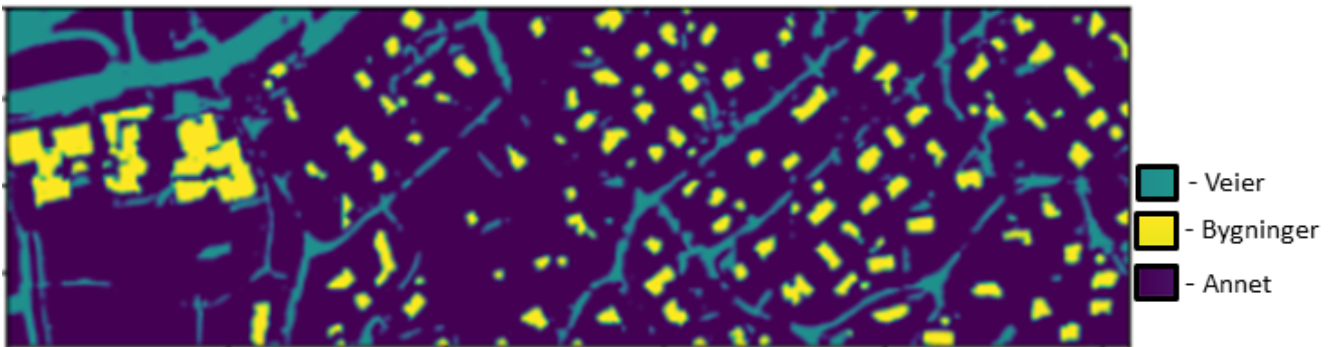
Figur 4.10 viser fasiten(fig a), området(fig b) og resultatet(fig c) fra prediksjonen for E1. Prediksjonen viser tendenser til at modellen klarer å skille veier og bygninger. Likevel har modellen fremdeles utfordringer med korrekt predikering av for eksempel smale veier og bygninger ved tett vegetasjon.



a.) Fasit data for område 07(FKB)



b.) RGB visning av område 07



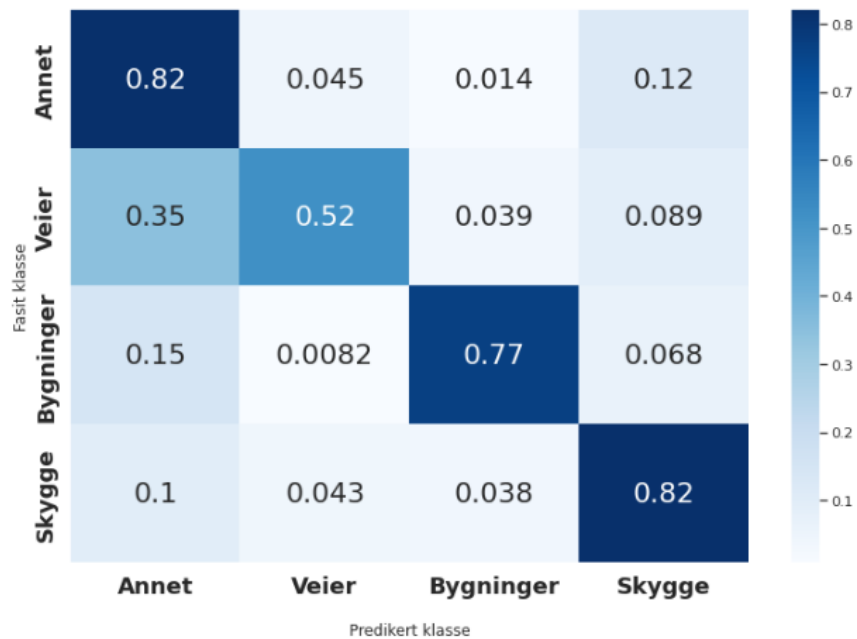
c.) Prediksjon av område 07, eksperiment 1

Figur 4.10: Modellens prediksjon av de tre klassene i E1 for område 07. Fig a viser fasitdataen, fig b viser RGB-bildet, fig c viser prediksjonen.

4.3.2 Klassifisering av veier, bygninger og skygge

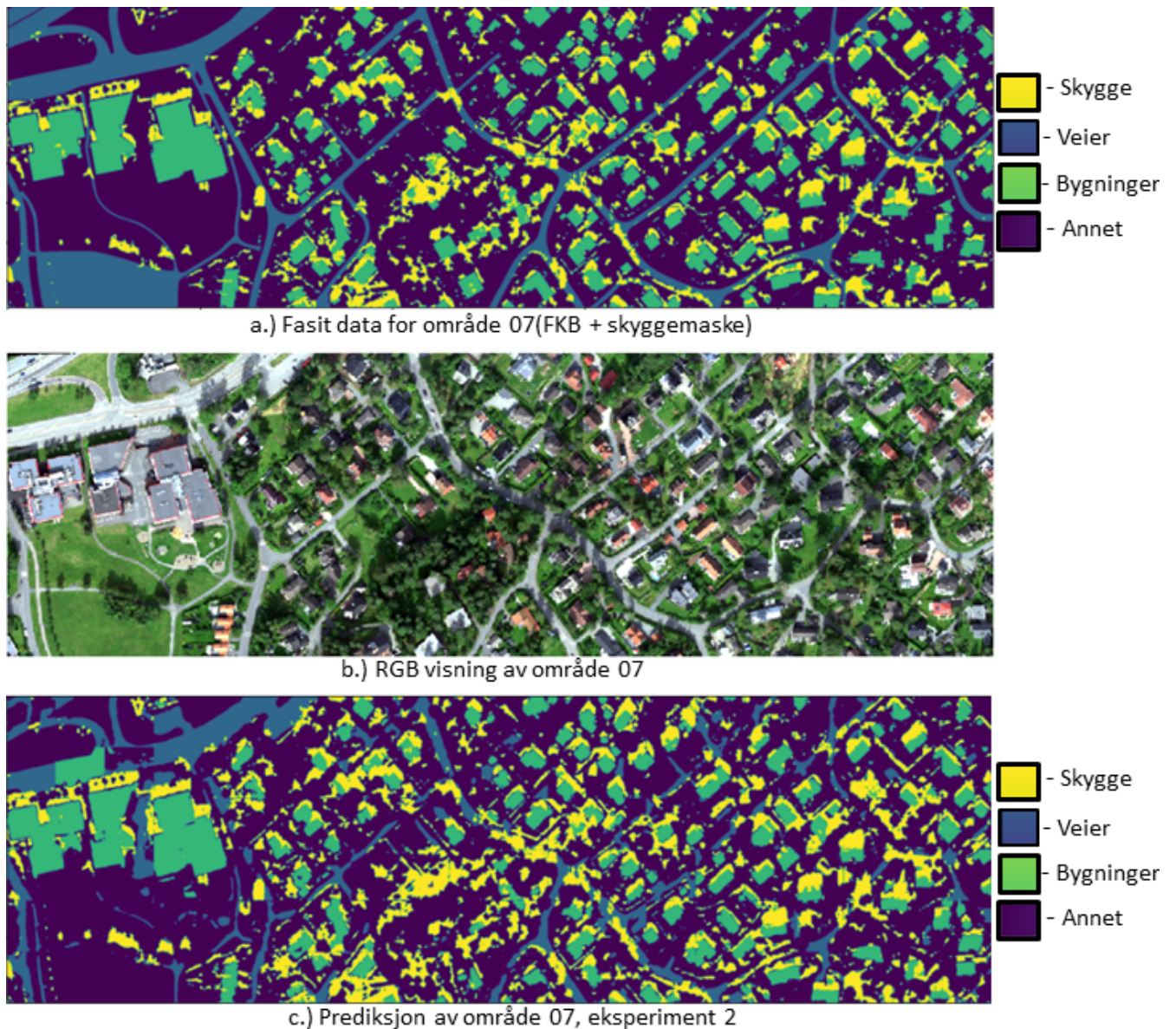
Prediksjon av E2

I E2 predikerer modellen den nye klassen med skygge generert fra skyggemasken. Forvirringsmatrisen gitt i figur 4.11, viser resultatene sammenlignet med fasiten for de fire klassene. Andelen feilklassifiseringer øker for klassene; ”annet”, ”veier” og ”bygninger” sammenlignet med E1. Store deler av feilklassifiseringene gjelder ”skygge” klassifisert som ”annet”, og ”annet” klassifisert som ”skygge”.



Figur 4.11: Forvirringsmatrise for E2 som består av fire klasser.

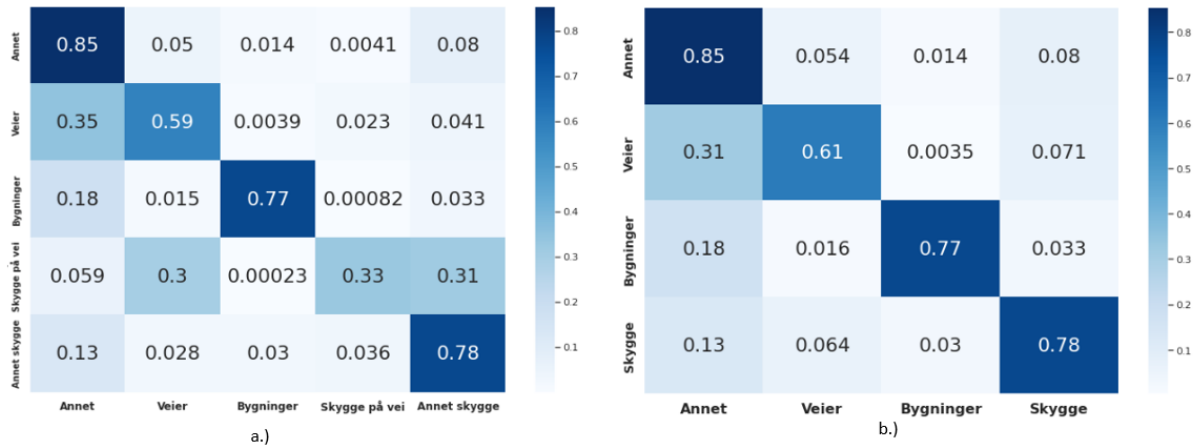
Slik figur 4.12c viser, gir prediksjonen av området i henhold til fasiten og RGB-bildet, gode resultater. Modellen klarer å trekke ut veier som ikke finnes i fasitdataen, men som kan identifiseres på RGB-bildet. Likevel blir enkelte områder feilklassifisert.



Figur 4.12: Modellens prediksjon av de fire klassene i E2 for område 07. Fig a viser fasilitdataen, fig b viser RGB-bildet, fig c viser prediksjonen.

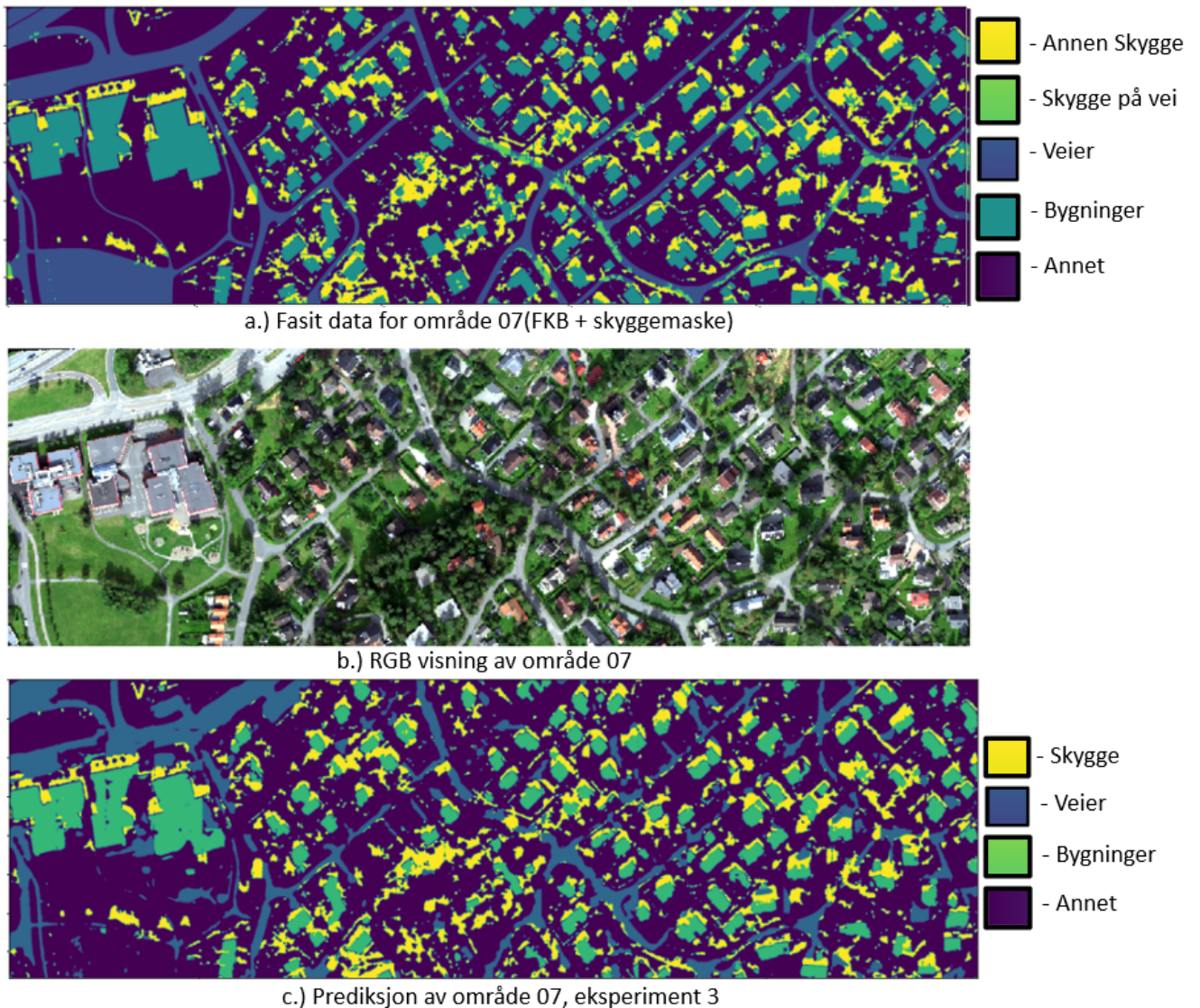
Prediksjon av E3

Figur 4.13 viser to forvirringsmatriser for E3. Figur 4.13a representerer resultatene for de fem klassene; "annet", "veier", "bygninger", "skygge på vei" og "annen skygge". Figur 4.13b viser resultatet av post-prosesseringen, der klassen "skygge på vei" slås sammen med klassen "vei". Nå er det kun fire klasser å ta hensyn til. Figur 4.13b viser at 59 % av "vei" blir predikert rett, før de to klassene blir slått sammen. Etter sammenslåingen derimot øker andel av "vei" som blir predikert rett, til 61%.



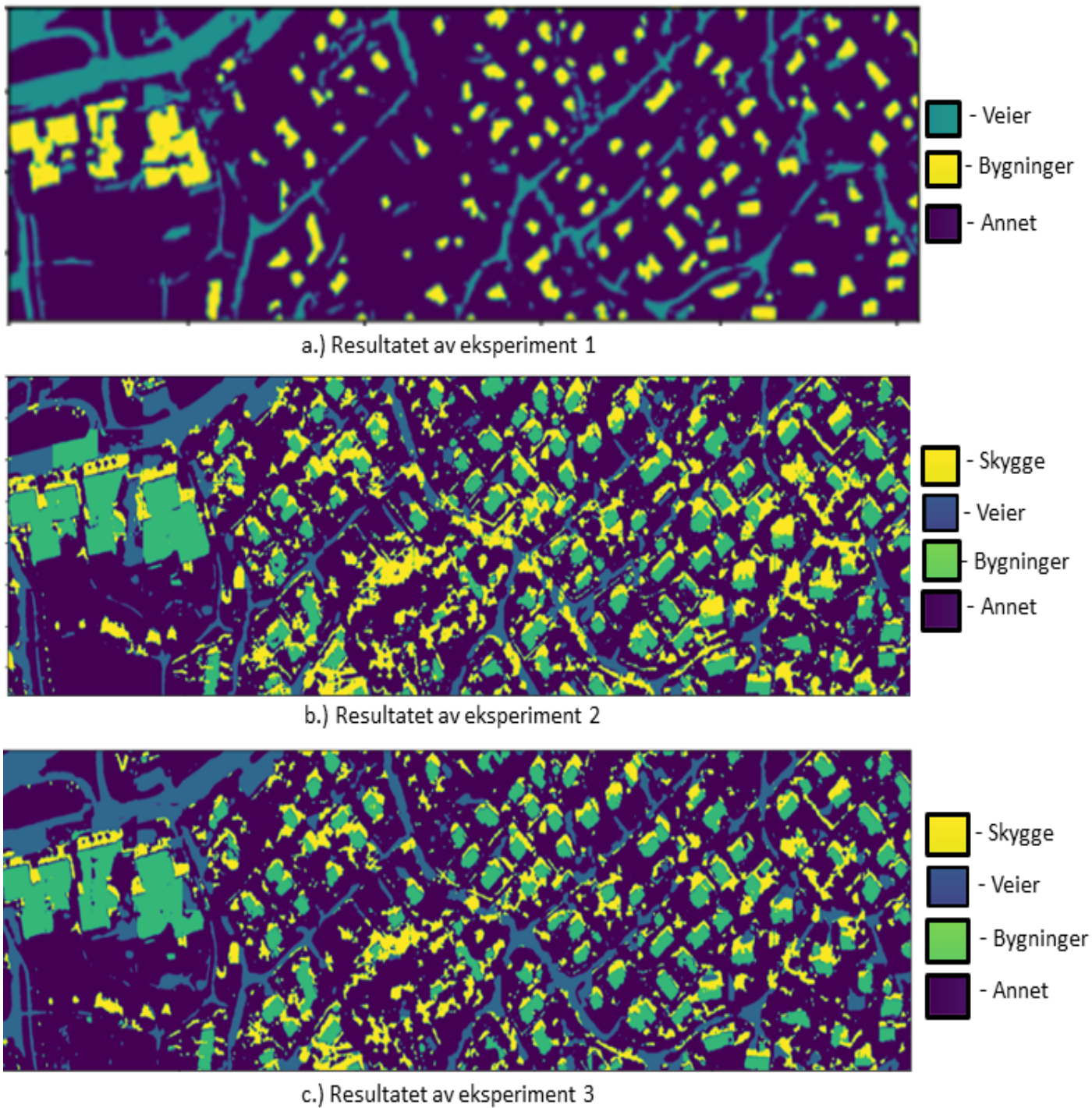
Figur 4.13: Forvirringsmatriser for E3 før og etter post-prosessering. Fig a viser forvirringsmatrisen med fem klasser, fig b viser forvirringsmatrisen med fire klasser etter post-prosessering.

Figur 4.14 viser resultatet av prediksjonen for E3. I dette eksperimentet klarer også modellen å predikere veier som ikke er en del av fasitdataen.



Figur 4.14: Modellens prediksjon av de fem klassene i E3 for område 07. Fig a viser fasitdataen, fig b viser RGB-bildet, fig c viser resultatet etter post-prosessering av prediksjonen.

4.3.3 Sammenligning av resultater



Figur 4.15: Sammenligning av prediksjonen fra de tre eksperimentene. Fig a viser for E1, fig b for E2 og fig c for E3.

Kapittel 5: Diskusjon

I den tidlige fasen av denne studien ble det gjort et forsøk på pikselbasert prediksjon av område 07. I forsøket ble det observert at maskinlæringsmodellen U-net hadde utfordringer med å klassifisere veier som var dekket av skygge. Figur 5.1 viser et eksempel på denne utfordringen. I figuren er det er gitt et utsnitt av område 07 med fasitdata og modellens prediksjon. Sammenligningen av prediksjonen med testbildet og fasiten, viser tydelig at modellen har utfordringer med å predikere områdene av veien som er dekket av skygge. Derfor ble det valgt å forske videre på detektering av skygge og generering av skyggeklasser i denne studien.



Figur 5.1: Predikering av et området bestående av blant annet asfaltert vei med og uten skygge. Illustrasjonen viser tre bilder; testbildet, fasiten og prediksjonen av testbildet.

5.1 Datasettet

I denne studien blir atmosfærisk korrigerede reflektansdata benyttet. Reflektansdataen består av 176 bånd i VNIR og 222 bånd i SWIR. I tillegg til VNIR og SWIR, ble nDOM lagt til som et bånd. Etersom prosjektområdet ikke består av områder med vann, ble vannabsorpsjonsbåndene fjernet fra datasettet til SWIR. Til sammen utgjør datasettet 399 bånd. I initieringsfasen av forskningen, var det sentralt å teste modellens predikeringsevne. Det ble derfor gjennomført et forsøk, der de genererte skyggemaskene ble lagt til som et bånd i datasettet. Å legge til dette båndet ga ingen merkbar endring i modellens predikeringsevne. Derfor ble det valgt å ikke forske videre på dette i studien.

I tidligere studier, (Senchuri 2020) og (Primstad & Stemme 2019), har det blitt benyttet pikselbasert klassifisering av både radians- og reflektansdata. I disse studiene har et enkelt konvolusjonelt nevralt nettverk prestert bedre på radiansdatasettet enn på reflektansdataen. I denne masteroppgaven blir fokuset kun rettet mot reflektansdataen, ettersom atmosfærisk korrigerende for det aktuelle prosjektområdet allerede var utført. Det kunne altså vært interessant å utforske predi-

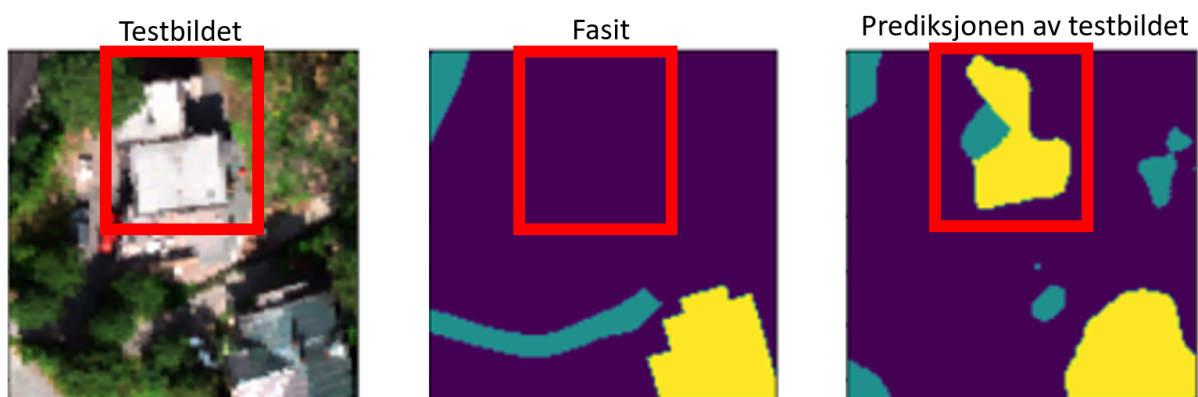
keringsevnen til et dypt konvolusjonelt nevralt nettverk på både radiansdata og reflektansdata.

Datasettet inneholder ubalanserte klasser for de tre eksperimentene som ble utført i studien, som forklart i delkapittel 3.6.2. Klassen "annet" utgjør mellom 60 og 75% av hele datasettet for de tre eksperimentene. For å adressere utfordringen knyttet til ubalanserte klasser, ble det valgt å evaluere modellens predikeringsevne etter "jaccard score" og "F1-score". Modellen ble ikke evaluert etter nøyaktighet. Årsaken til dette er at evaluering av nøyaktighet, kan gi en misvisende representasjon av modellens predikeringsevne, ettersom klassen "annet" utgjør store deler av datasettet.

5.1.1 Begrensninger med Felles kartdatabase

Data hentet fra FKB for bygg og vei ble brukt som fasitdata i denne studien. I FKB-dataen ble det oppdaget mangler, som kan ha ført til at modellen ikke har klart å lære godt nok til å kunne predikere rett. Det ble blant annet funnet mangel på deler av veinettet i FKB-dataen. Det ble også oppdaget feil i form av at områder som ikke blir regnet som veier, ble klassifisert som veier i FKB-vei. Områder som ikke var en del av FKB-dataen ble klassifisert som "annet". Dette førte til en rekke problemer med trening og testing av modellen. Feilklassifisering og evaluering av modellen kan være misvisende når prediksjonen skal sammenlignes med en fasit som ikke er representativ for virkeligheten.

Figur 5.2 viser et område i testsettet, bestående av RGB-vising, fasitdata og predikeringen. De røde firkantene representerer et bygg som vises på RGB-bildet og blir predikert korrekt av modellen, men som ikke er en del av FKB-bygg i fasitdataen. For å avdekke om dette området består av et bygg eller ikke, ble det valgt å bruke hyperspektrale og LiDAR data, se figur 5.3. De røde firkantene viser det aktuelle bygget, og det kunne raskt fastslås at dette var et bygg som manglet i fasitdataen.



Figur 5.2: Prediksjon av et testområde med feil i fasiten. De røde firkantene representerer et bygg på testbildet og prediksjonen. Bygget mangler i fasiten.

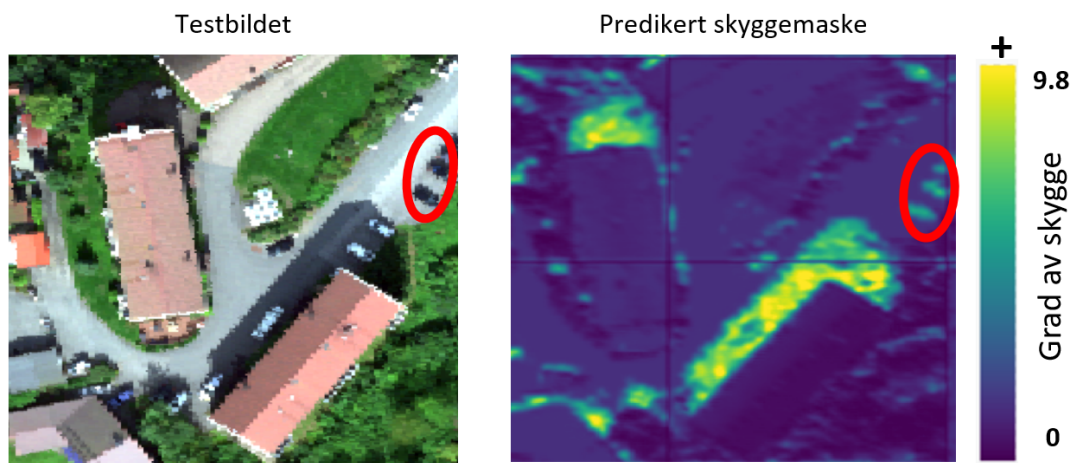


Figur 5.3: De røde firkantene viser det aktuelle bygget i RGB- og LiDAR-visning.

5.2 Analyse av resultatene

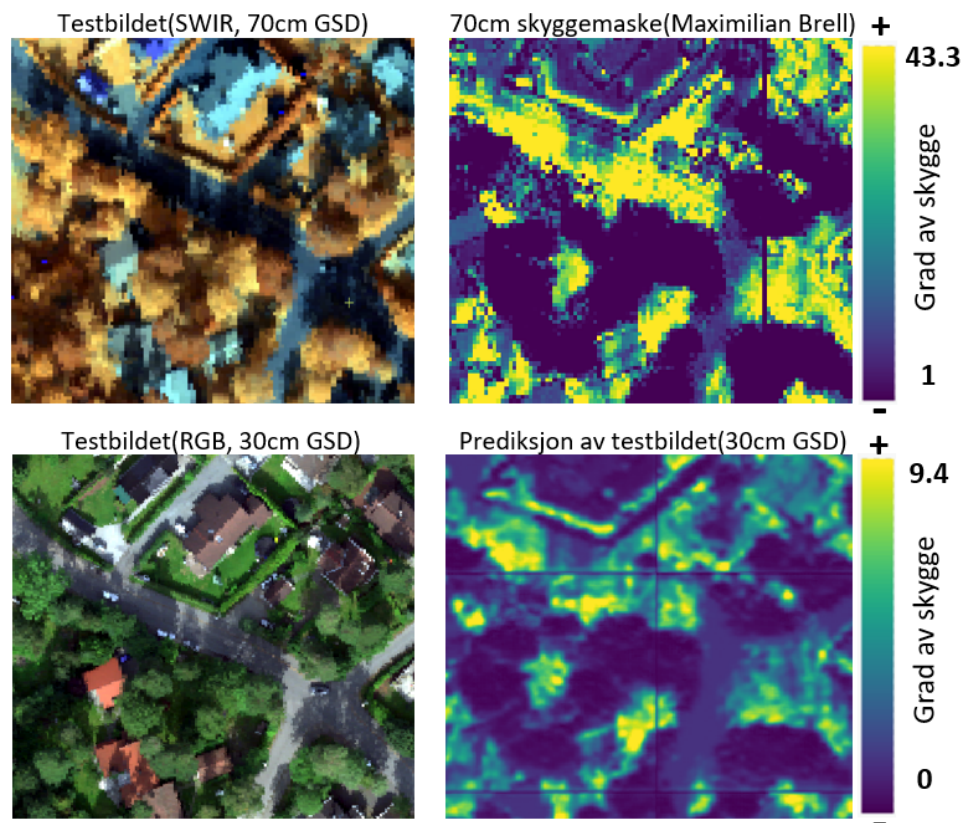
I delkapittel 4.2 vises resultatene av predikerte skyggemasker. For å se nærmere på modellens evne til å predikere skyggemasker, presenteres et utklipp av skyggemasken sammen med et RGB-bilde for tilsvarende område. Figur 5.4 viser et utklipp av skyggemasken som ble produsert for område 04. De gule områdene som vises på den predikerte skyggemasken, angir områder der modellen har predikert skygge. Modellen presterer godt på å identifisere skygge, når skyggemasken blir sammenlignet med testbildet. Dette tyder på at modellen har lært å kjenne igjen mørke piksler og at den dermed klarer å klassifisere disse som skygge. Dette er årsaken til at de mørke områdene på bakken ved siden av bygningen på figur 5.4, blir godt predikert av modellen.

Et konvolusjonelt nevralt nettverk er spesielt god på å bruke informasjon fra nærliggende piksler til å predikere en piksel. En mulig utfordring kan være at modellen kan ha blitt lært opp til å klassifisere mørke partier ved siden av objekter som skygge. I figur 5.4 er det gitt et område markert med en rød sirkel på både testbildet og den predikerte skyggemasken. Her blir en parkeringsplass med tre mørke biler predikert som skygge. Dette kan tyde på at modellen er følsom for mørke pikselverdier. Ved produksjon av skyggeklasser, ble det derfor benyttet en grenseverdi for å luke ut piksler som ikke er skygge. Det er ikke gitt en fasit for skyggemaskene med 30cm oppløsning. Derfor blir evalueringen av modellens predikeringssevne, basert på sammenligningen av skyggemasken og RGB-bildet for tilsvarende område.



Figur 5.4: Et testbilde fra område 04 vises sammen med den predikerte skyggemasken på 30cm romlig oppløsning. Den røde sirkelen markerer en parkeringsplass som har blitt feilpredikert som skygge.

Figur 5.5 viser fire bilder, der SWIR i 70cm romlig oppløsning ble brukt for å lage skyggemaske på 70cm av Maximilian Brell. De to figurene i figur 5.5 illustrerer en RGB-visning og prediksjon av tilsvarende område i 30cm oppløsning. Tatt i betraktning at modellen har blitt trent opp med en skyggemaske på 70cm romlig oppløsning, klarer modellen meget godt å predikere skyggemaske på 30cm romlig oppløsning. I denne studien var det mest sentrale at modellen klarte å identifisere områdene med skygge, slik at det kunne bli generert skyggeklasser. Dette var mulig med skyggemaske som ble produsert av modellen.



Figur 5.5: Sammenligning av predikert skyggemaske på 30cm romlig oppløsning og 70cm romlig oppløsning for det samme området.

Forvirringsmatrisene vist på figurene 4.9, 4.11 og 4.13 representerer resultatene fra de ulike eksperimentene. Ettersom fasitdataen inneholder mangler, blir det utfordrende å kartlegge hvor godt modellen presterer i de ulike eksperimentene. Resultatene for de tre eksperimentene, gitt med et utsnitt av testområde 07 er vist på figur 5.6.

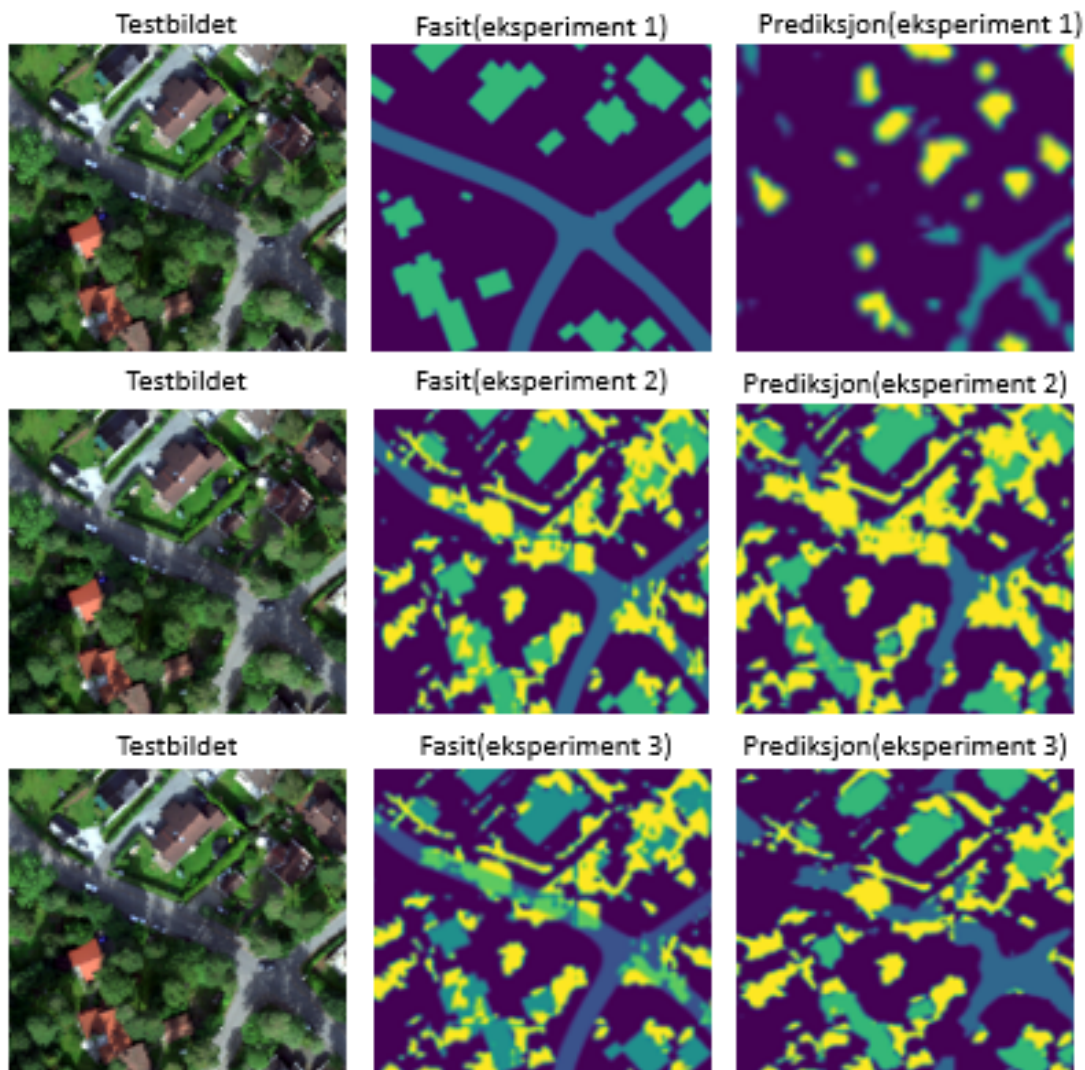
Modellen fra eksperiment 1 har utfordringer med predikering av materialer som er dekket av skygge, se figur 5.6. For å kartlegge om modellen klarte å predikere skygge på materialer, ble det utført et nytt eksperiment. De gule områdene på eksperiment 2, viser områder der modellen har predikert skygge. Sammenlignet med testbildet stemmer det godt. De blå områdene viser veier predikert av modellen. De predikerte veiene ser ufullstendige ut på grunn av den predikerte skyggen på veien. Ettersom modellen klarte å predikere skygge med 82% nøyaktighet, ble det innført en ny klasse for "skygge på veier" i eksperiment 3.

Eksperiment 3 ble gjennomført, for å utforske om modellen hadde klart å lære seg forskjellen på skygge på asfaltert vei og skygge på alle andre materialer. Dersom modellen hadde klart å segmentere skygge på ulike materialer, kunne klassen ”skygge på vei” og klassen ”veier” bli slått sammen til én klasse. Dette ble det gjort et forsøk på i eksperiment 3. I figur 5.6 vises prediksjonen fra eksperiment 3, der de gule områdene som representerer ”skygge på veier” har blitt redusert, sammenlignet med prediksjonen fra eksperiment 2. De gule områdene har i eksperiment 3 blitt predikert som ”vei”.

Modellen har fremdeles utfordringer med å segmentere skygge på ulike materialer som vises på prediksjon (E3). Det er enkelte områder på veien som blir predikert som skygge og ikke vei. En av grunnene til at modellen ikke har klart å segmentere de ulike skyggeområdene, kan være at klassen ”skygge på vei” består av lite treningsdata. ”Skygge på vei” representerer kun 2.1% av datasettet. Figur 4.13a viser forvirringsmatrisen for eksperiment 3, der modellen kun klarte å predikere 1/3 av denne klassen korrekt. ”Skygge på vei” er altså den klassen modellen har mest utfordringer med å predikere rett.

Tabell 4.4 viser en evaluering av de tre eksperimentene. Basert på evalueringen av prediksjonen for alle klassene, oppnår eksperiment 1 høyest verdi i metrikkene. Eksperiment 1 oppnår en ”jaccard-score” på 0.76 og ”F1-score” på 0.86. Sammenligningen av forvirringsmatrisene, 4.9, 4.11 og 4.13 for de tre eksperimentene, viser at klassene ”veier” og ”bygninger” blir predikert mest korrekt i eksperiment 3.

Sammenligning av prediksjonen for de tre eksperimentene vises i figur 4.15. Både prediksjonen fra eksperiment 2 og 3 tyder på at modellen klarer å predikere små asfalterte stier som ikke er en del av fasitdataen. Bygninger har blitt predikert mer korrekt og dette gjenspeiles i forvirringsmatrisene for E2 og E3. Basert på de mest korrekte predikeringene for klassene ”veier” og ”bygninger”, og sammenligningen av prediksjonene, ser eksperiment 3 ut til å gi den beste predikeringen av område 07.



Figur 5.6: Et testområde blir valgt ut til sammenligning av prediksjonen for de tre eksperimentene.

5.3 Tidligere forskning

I en forskning utført av G.Tolt (Tolt et al. 2011) ble skyggedeteksjon av hyperspektrale og LiDAR data utforsket. I forskningen har blitt det satt fokus på bruk av maskinlæring til predikering av områder, både med og uten skygge. En skyggemaske generert fra en overflatemodell blir benyttet til å trene opp en SVM ("Support Vector Machine") og til å predikere en binær skyggemaske for nye områder. En ulempe som blir påpekt i studien til G.Tolt er mangelen på en glidende overgang fra områder med skygge til områder uten skygge. Ettersom det kun blir predikert en binær skyggemaske, mister denne masken informasjonen om graden av skygge per piksel.

Til sammenligning, har det i denne masteroppgaven blitt brukt et konvolusjonelt nevralt nettverk som har blitt trent opp til å predikere graden av skygge. Den predikerte skyggemasken kan også brukes til å lage en binær skyggemaske ved å sette en grenseverdi, slik det blir gjort i forskningsprosjektet utført av G.Tolt (Tolt et al. 2011). Hensikten med å predikere graden av skygge, er at det kan brukes en definert skyggegrad til å generere skyggeklasser. En glidende overgang mellom piksler med og uten skygge kan benyttes ved korrigerende av skygge.

I andre studier som (Zhou et al. 2021) og (Fortuna & Martens 2017) er forskningen rettet mot deteksjon og korrigerings av skygge. Skyggekorrigering er en mer kompleks oppgave som krever beregning av skyggeindekser, slik at overgangen fra områder med skygge til områder uten skygge, blir gradvis korrigert. Dette gir en jevn overgang fra områder uten skygge til områder med skygge. I denne masteroppgaven derimot, blir det tatt i bruk en enklere metode, der skygge blir tatt hensyn til ved å opprette skyggeklasser. Dette blir gjort for å lære en maskinlæringsmodell å identifisere forskjellen på et materiale med og uten skygge. Etter klassifisering kan en post-prosessering utføres, ved at klassen med og uten skygge for et materiale slås sammen til én klasse.

I tidligere studier som (Senchuri 2020) og (Primstad & Stemme 2019), har det blitt forsket på bruk av maskinlæring på hyperspektrale bilder. Begge studiene har hatt fokus på pikselbasert klassifisering av urbane områder og testet bruken av enkle konvolusjonelle nevralt nettverk. Basert på metodikken og oppnådde resultater fra overnevnte studier, blir det i denne studien valgt å forske mer på bruken av dype konvolusjonelle nevralt nettverk. U-net arkitekturen er et dypt konvolusjonelt nevralt nettverk som er kjent for sin evne til å segmentere et bilde presist med få bilder. Det ble derfor valgt å benytte U-net arkitektur, for å utforske om resultatene fra disse studiene, (Senchuri 2020) og (Primstad & Stemme 2019), kunne forbedres.

U-net arkitekturen som blir brukt i denne masteroppgaven inneholder 32 lag med konvolusjonelle filtre og "skip connection", som iverretar og sender ekstrahert informasjon fra nedskaleringsdelen til oppskaleringsdelen, som beskrevet i delkapittel 2.5.3. I denne masteroppgaven ble modellene i de tre eksperimentene evaluert til 0.86, 0.79 og 0.81 i "F1-score". Til sammenligning består arkitekturen brukt i studien (Senchuri 2020), av tre lag med konvolusjonelle filtre uten "skip connection", og modellens predikeringsevne har blitt målt til en "f1-score" på 0.57. I studien (Primstad & Stemme 2019) har det blitt forsket på konvolusjonelle filtre bestående av tre lag uten "skip connection". Ved bruk av det konvolusjonelle nevralt nettverket, ble det målt en nøyaktighet på omtrent 51%. Altså ble det ikke oppnådd like gode resultater som ved bruk av logistisk regresjon i studien (Primstad & Stemme 2019). Logistisk regresjon ble her målt til en nøyaktighet på 86%.

I denne masteroppgaven har det blitt forsket videre på ulike bildekvadratstørrelser for predikering av skyggemasker og pikselbasert klassifisering. Formålet med å teste ulike kvadratstørrelser er å undersøke hvilken størrelse som resulterer i mest korrekt predikering. Bildekvadratstørrelsene, 34x34, 64x64, 128x128 og 256x256 ble testet, som beskrevet i delkapittel 3.6.

I et forskningsprosjekt gjennomført av Sharma Atharva (Sharma et al. 2017) ble det oppnådd 20% høyere nøyaktighet ved klassifisering av bildekvadrater, enn ved predikering av et stort bilde. I denne studien ble resultatene fra valideringssettet sammenlignet og evaluert etter "jaccard-score". Tabell 4.4 viser resultatene fra predikeringen for de ulike størrelsene. Slik tabellen viser, gir størrelsen 128x128 det beste resultatet. Bildekvadratstørrelsen 128x128 gir 13.6% høyere "jaccard-score" enn ved bruk av størrelsen 256x256. Sammenlignet med de to andre størrelsene er differansen på mellom 3%-5%.

5.4 Forslag til videre arbeid

Skyggemaskene som ble produsert i 30cm romlig oppløsning, klarte å identifisere områder med skygge på hyperspektrale bilder, kombinert med nDOM. Å forske videre på bruk av skyggemaske og kartlegge om masken kan brukes til skyggekorrigering, ville vært et interessant forsknings-tema med tanke på videre arbeid. Det ville også vært interessant å utforske om generering av skyggeklasser for flere typer materialer, vil gi mer korrekte predikeringer. Generering av fasitdata er en tidkrevende prosess. Likevel er dette anbefalt å gjøre, ettersom gode fasitdata vil gi direkte utslag i modellens evne til å segmentere korrekt.

Kapittel 6: Konklusjon

Formålet med denne oppgaven er å klassifisere veier og bygninger i urbane områder ved bruk av hyperspektrale bilder og LiDAR. For å nå dette målet ble det gjennomført skyggedeteksjon og klassifisering av asfalterte veier og bygninger. Ettersom skygge påvirker nøyaktigheten av klassifisering, ble det valgt å forske på skyggedeteksjon. Det konvolusjonelle nevrale nettverket U-net, ble benyttet til å identifisere skygge ved predikering av skyggemasker fra hyperspektrale bilder og LiDAR. U-net ble videre brukt til pikselbasert klassifisering av asfalterte veier, bygninger og skyggeklasser. Skyggeklassene ble generert fra de predikerte skyggemaskene.

I denne masteroppgaven blir det forsket på hvorvidt skyggedeteksjon bidrar til økt nøyaktighet ved pikselbasert klassifisering av veier og bygninger i urbane områder. For å utforske dette forskningsspørsmålet nærmere, ble det gjennomført tre eksperimenter.

- **E1:** Klassifisering av veier, bygg og annet.
- **E2:** Innføring av skyggeklasse. Klassifisering av veier, bygg, skygge og annet.
- **E3:** Innføring av to skyggeklasser. Klassifisering av veier, bygg, skygge på veier, annen skygge og annet.

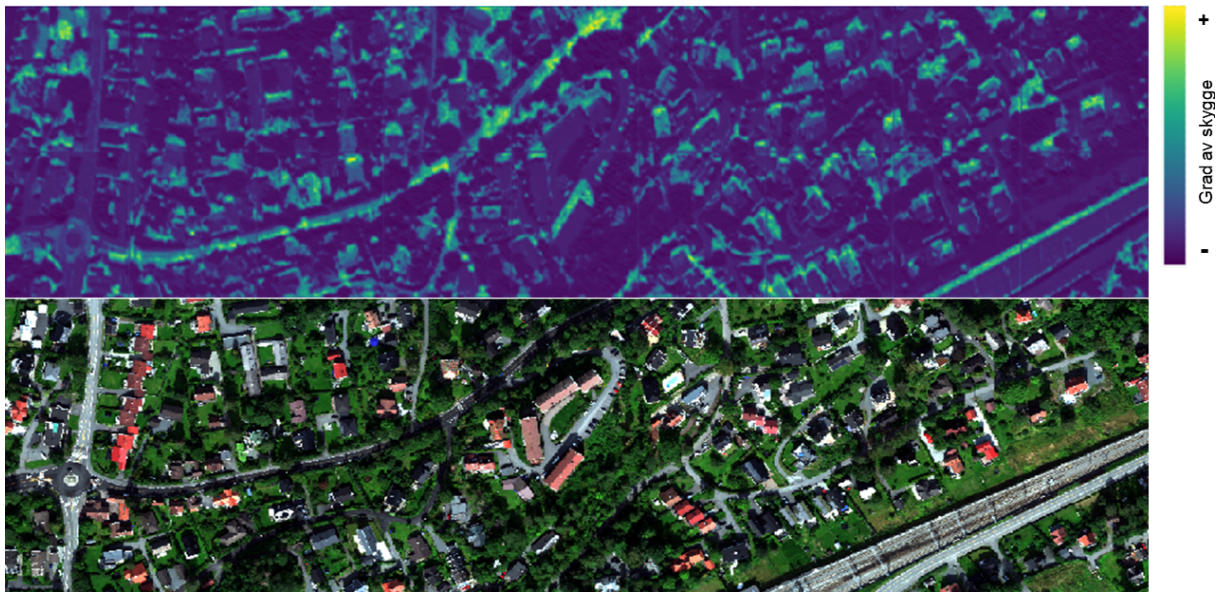
Modellen for predikering av skyggemaske presterte godt sammenlignet med fasiten og RGB-bildet av tilsvarende område. På den måten klarte modellen å identifisere skygge. Resultatene fra de tre eksperimentene vises i tabell 4.4. Basert på målt metrikk av klassifiseringen for hele datasettet, oppnår E1 høyest ”score”. Som følge av ubalanserte klasser og avvik i fasitdataen, kan ikke klassifiseringen bli evaluert etter kun metrikk-”score”. Dersom det kun velges å ta utgangspunkt i klassene ”veier” og ”asfalterte bygninger”, vil E3 gi mest nøyaktige klassifiseringer. I E3 ble det oppnådd en nøyaktighet på 61% for asfalterte veier og 77% for bygninger, vist i figur 4.13b.

Ut ifra resultatene av eksperimentene som ble gjennomført, kan det i denne studien konkluderes med at skyggedeteksjon og innføring av skyggeklasser, vil øke nøyaktigheten på klassifisering av asfalterte veier og bygninger i urbane områder.

Kapittel A: Vedlegg

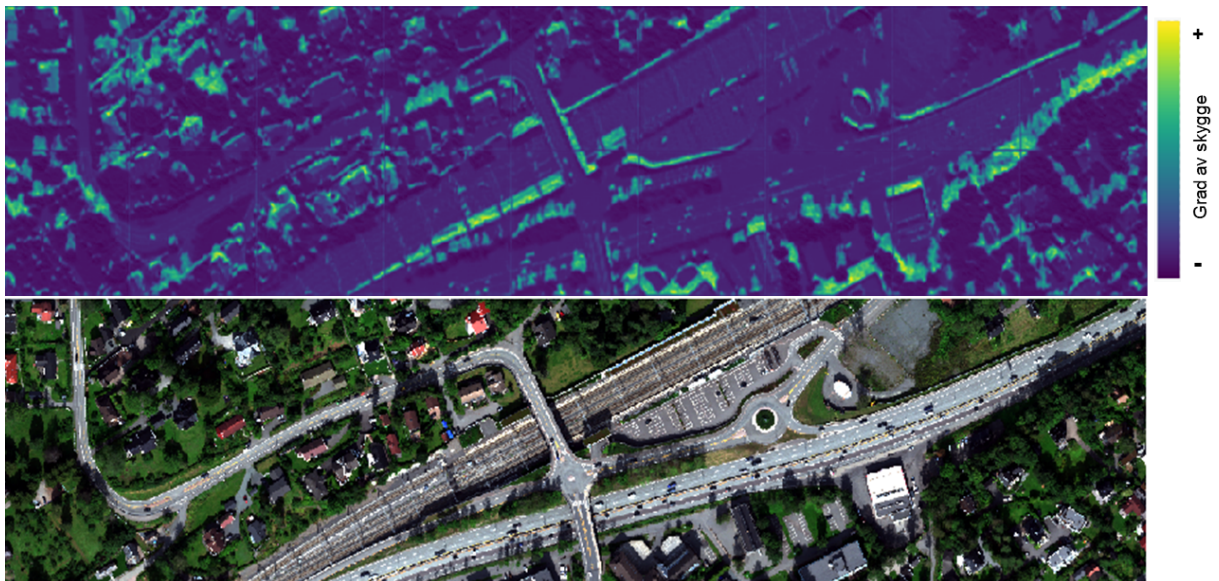
A.0.1 Predikering av skyggemaske for 70cm oppløsning

Skyggemaske for område 04

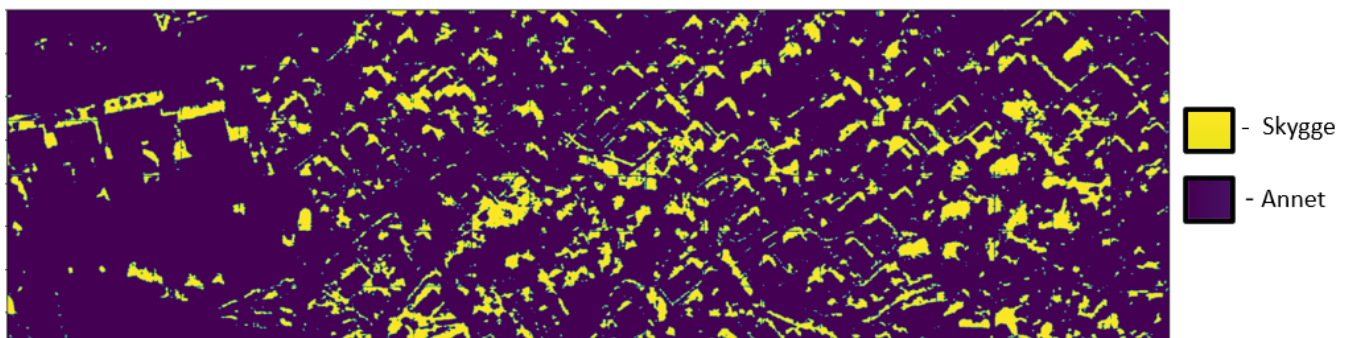
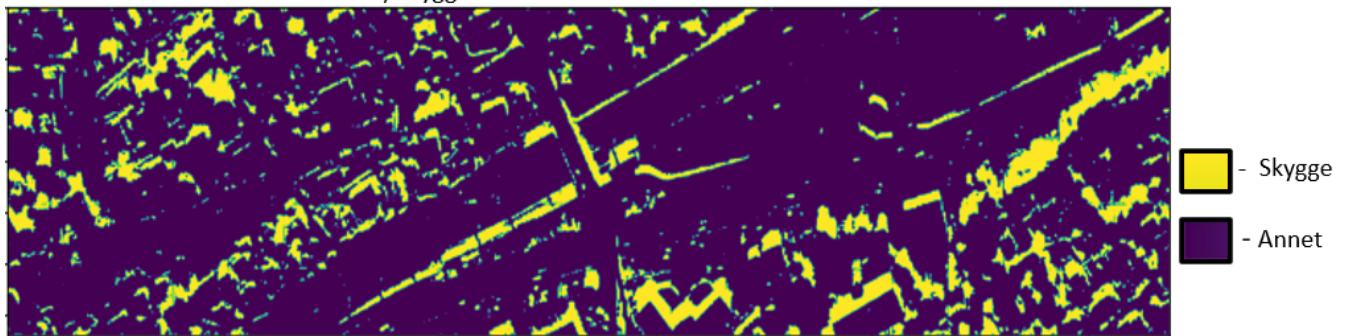
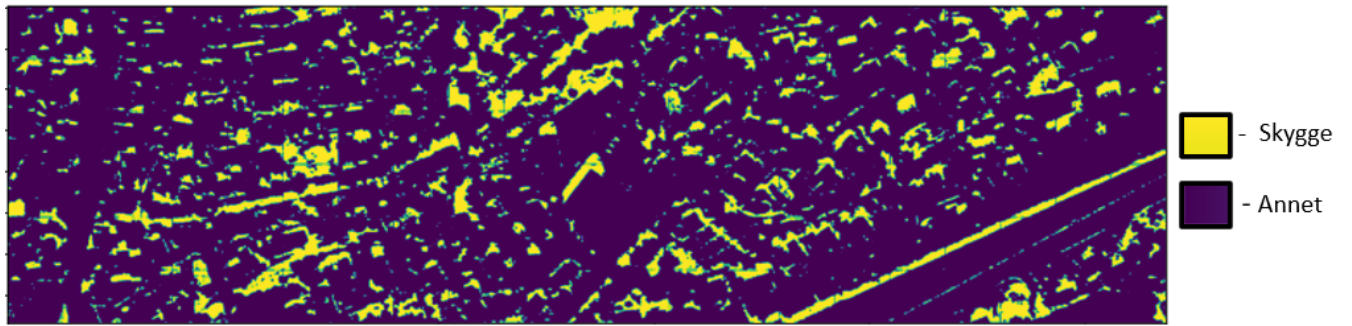


Figur A.1: Skyggemaske for område 04 i 70cm oppløsning.

Skyggemaske for område 05



Figur A.2: Skyggemaske for område 05 i 70cm oppløsning.



Figur A.3: Genererte skyggeklasser for eksperiment 2

Bibliografi

- Agatonovic-Kustrin, S. & Beresford, R. (2000), ‘Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research’, *Journal of pharmaceutical and bio-medical analysis* **22**(5), 717–727.
- Alshehhi, R., Marpu, P. R., Woon, W. L. & Dalla Mura, M. (2017), ‘Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks’, *ISPRS Journal of Photogrammetry and Remote Sensing* **130**, 139–149.
- Baltsavias, E. P. (1999), ‘Airborne laser scanning: basic relations and formulas’, *ISPRS Journal of photogrammetry and remote sensing* **54**(2-3), 199–214.
- Berg, T. R. (2020), ‘Fkb’.
URL: <https://snl.no/FKB>
- Birkeland, R. (2021), ‘Fjernmåling’.
URL: <https://snl.no/fjernmaaling>
- Cracknell, A. P. (2007), *Introduction to remote sensing*, CRC press.
- Dietterich, T. (1995), ‘Overfitting and undercomputing in machine learning’, *ACM computing surveys (CSUR)* **27**(3), 326–327.
- Dumoulin, V. & Visin, F. (2016), ‘A guide to convolution arithmetic for deep learning’, *arXiv preprint arXiv:1603.07285* .
- El Naqa, I., Li, R. & Murphy, M. J. (2015), *Machine learning in radiation oncology: theory and applications*, Springer.
- Fidon, L., Li, W., Garcia-Peraza-Herrera, L. C., Ekanayake, J., Kitchen, N., Ourselin, S. & Vercauteren, T. (2017), Generalised wasserstein dice score for imbalanced multi-class segmentation using holistic convolutional networks, in ‘International MICCAI brainlesion workshop’, Springer, pp. 64–76.
- Fortuna, J. F. & Martens, H. (2017), ‘Multivariate data modelling for de-shadowing of airborne hyperspectral imaging’, *Journal of Spectral Imaging* **6**.
- Franke, J., Roberts, D. A., Halligan, K. & Menz, G. (2009), ‘Hierarchical multiple endmember spectral mixture analysis (mesma) of hyperspectral imagery for urban environments’, *Remote Sensing of Environment* **113**(8), 1712–1723.

- Geonorge (2022), ‘6.2. fkb-områdetyper’.
URL: https://sosi.geonorge.no/standarder/FKBgenerell_d/5.0/
- Griffin, M. K. & Burke, H.-h. K. (2003), ‘Compensation of hyperspectral data for atmospheric effects’, *Lincoln Laboratory Journal* **14**(1), 29–54.
- Hao, W., Yizhou, W., Yaqin, L. & Zhili, S. (2020), The role of activation function in cnn, *in* ‘2020 2nd International Conference on Information Technology and Computer Application (ITCA)’, IEEE, pp. 429–432.
- Harris Geospatial Solutions, I. (2019), ‘Gram-schmidt pan sharpening’.
URL: <https://www.l3harrisgeospatial.com/docs/gramschmidtspectralsharpening.html>
- Hashemi, S. (2008), ‘Automatic peaks extraction from normalized digital surface model (ndsm)’, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* pp. 3–11.
- Hoffman, K. (2021), ‘Machine learning: How to prevent overfitting’.
URL: <https://medium.com/swlh/machine-learning-how-to-prevent-overfitting-fdf759cc00a9>
- HySpex-NEO (2022a), ‘Hyspex classic swir-384 configuration’.
URL: <https://www.hyspex.com/hyspex-products/hyspex-classic/hyspex-swir-384/>
- HySpex-NEO (2022b), ‘Hyspex classic vnir-1800 configuration’.
URL: <https://www.hyspex.com/hyspex-products/hyspex-classic/hyspex-vnir-1800>
- Jadon, S. (2020), A survey of loss functions for semantic segmentation, *in* ‘2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)’, IEEE, pp. 1–7.
- Jenal, A., Bareth, G., Bolten, A., Kneer, C., Weber, I. & Bongartz, J. (2019), ‘Development of a vnir/swir multispectral imaging system for vegetation monitoring with unmanned aerial vehicles’, *Sensors* **19**(24), 5507.
- Jensen, J. R. (2007), *Remote sensing of the environment an earth resource perspective second edition*, Pearson Education, Inc.
- Khorram, S., Koch, F. H., van der Wiele, C. F. & Nelson, S. A. (2012), *Remote sensing*, Springer Science & Business Media.
- Lary, D. J., Alavi, A. H., Gandomi, A. H. & Walker, A. L. (2016), ‘Machine learning in geosciences and remote sensing’, *Geoscience Frontiers* **7**(1), 3–10.
- Liao, W., Bellens, R., Pižurica, A., Philips, W. & Pi, Y. (2012), Classification of hyperspectral data over urban areas based on extended morphological profile with partial reconstruction, *in* ‘International Conference on Advanced Concepts for Intelligent Vision Systems’, Springer, pp. 278–289.

- Long, J., Shelhamer, E. & Darrell, T. (2015), Fully convolutional networks for semantic segmentation, *in* ‘Proceedings of the IEEE conference on computer vision and pattern recognition’, pp. 3431–3440.
- Norheim, B. (2020), ‘Elektromagnetisk spekter’.
URL: <https://ndla.no/nb/subject:1:f18b0daa-6507-4025-8998-b8a11c8ccc70/topic:5:dbc23651-7216-4610-bc38-dde58f013724/topic:3:1b4bd84e-b60f-40cb-83a8-23ad8926298a/resource:fcf00f3b-abfd-411c-9606-d621c59485ca>
- Ozdemir, A. & Polat, K. (2020), ‘Deep learning applications for hyperspectral imaging: A systematic review’, *Journal of the Institute of Electronics and Computer* **2**, 39–56.
URL: <https://iecscience.org/jpapers/49>
- Primstad, A. & Stemme, Å. (2019), Pikkelsbasert arealklassifisering av urbane områder med hyperspektrale flybilder og maskinlæring, Master’s thesis, Norwegian University of Life Sciences, Ås.
- Raschka, S. & Mirjalili, V. (2019), *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*, Packt Publishing Ltd.
- Ravikanth, L., Jayas, D. S., White, N. D., Fields, P. G. & Sun, D. W. (2017), ‘Extraction of spectral information from hyperspectral data and application of hyperspectral imaging for food and agricultural products’.
- Ray, T. W. (2013), ‘Radiance vs. reflectance’.
URL: <https://www.l3harrisgeospatial.com/Support/Self-Help-Tools/Help-Articles/Help-Articles-Detail/ArtMID/10220/ArticleID/19247/3377>
- Redell, N. (2019), ‘Shapley decomposition of r-squared in machine learning models’, *arXiv preprint arXiv:1908.09718*.
- Ronneberger, O., Fischer, P. & Brox, T. (2015), U-net: Convolutional networks for biomedical image segmentation, *in* ‘International Conference on Medical image computing and computer-assisted intervention’, Springer, pp. 234–241.
- Schowengerdt, R. A. (2006), *Remote sensing: models and methods for image processing*, Elsevier.
- Senchuri, R. (2020), Road edge detection using hyperspectral and lidar data based on machine and deep learning, Master’s thesis, Norwegian University of Life Sciences, Ås.
- Services, S. G. (2017), ‘5m digital surface model and digital terrain model’.
URL: <https://www.satpalda.com/5m-dsmdtm>
- Shafri, H. Z., Taherzadeh, E., Mansor, S. & Ashurov, R. (2012), ‘Hyperspectral remote sensing of urban areas: an overview of techniques and applications’, *Research Journal of Applied Sciences, Engineering and Technology* **4**(11), 1557–1565.
- Shahtahmassebi, A., Yang, N., Wang, K., Moore, N. & Shen, Z. (2013), ‘Review of shadow detection and de-shadowing methods in remote sensing’, *Chinese Geographical Science* **23**(4), 403–420.

- Sharma, A., Liu, X., Yang, X. & Shi, D. (2017), ‘A patch-based convolutional neural network for remote sensing image classification’, *Neural Networks* **95**, 19–28.
- Shippert, P. (2013a), ‘Digital number, radiance, and reflectance’.
URL: https://www.l3harrisgeospatial.com/Learn/Blogs/Blog-Details/ArtMID/10198/ArticleID/16278/Digital-Number-Radiance-and-Reflectance?fbclid=IwAR3bPmEVuhjJoH2LUCxGATQy6LaZp6wiTMvWrj_a9kBCe91t8 – S_{I1}
- Shippert, P. (2013b), ‘Push broom and whisk broom sensors’.
URL: <https://www.l3harrisgeospatial.com/Learn/Blogs/Blog-Details/ArtMID/10198/ArticleID/16262/Push-Broom-and-Whisk-Broom-Sensors>
- Skogseth, T. & Norberg, D. (2014), *Grunnleggende landmåling*, Gyldendal Norsk Forlag AS.
- Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S. & Jorge Cardoso, M. (2017), Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations, *in* ‘Deep learning in medical image analysis and multimodal learning for clinical decision support’, Springer, pp. 240–248.
- Szandata, T. (2021), Review and comparison of commonly used activation functions for deep neural networks, *in* ‘Bio-inspired neurocomputing’, Springer, pp. 203–224.
- Tolt, G., Shimoni, M. & Ahlberg, J. (2011), A shadow detection method for remote sensing images using vhr hyperspectral and lidar data, *in* ‘2011 IEEE international geoscience and remote sensing symposium’, IEEE, pp. 4423–4426.
- Trier, Ø., Salberg, A.-B., Larsen, R. & Nyvoll, O. T. (2022), Detection of forest roads in sentinel-2 images using u-net, *in* ‘Proceedings of the Northern Lights Deep Learning Workshop’, Vol. 3.
- Vosselman, G. & Maas, H.-G. (2010), *Airborne and Terrestrial Laser Scanning*, Whittles publishing.
- Yamazaki, F., Liu, W. & Takasaki, M. (2009), Characteristics of shadow and removal of its effects for remote sensing imagery, *in* ‘2009 IEEE International Geoscience and Remote Sensing Symposium’, Vol. 4, IEEE, pp. IV–426.
- Yang, M., Hu, Y., Tian, H., Khan, F. A., Liu, Q., Goes, J. I., Gomes, H. d. R. & Kim, W. (2021), ‘Atmospheric correction of airborne hyperspectral data using polymer, 6s and flaash’, *Remote Sensing* **13**(24), 5062.
- Zhang, J., Zhang, Y. & Zhou, T. (2001), Classification of hyperspectral data using support vector machine, *in* ‘Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)’, Vol. 1, IEEE, pp. 882–885.
- Zhang, W., Tang, P. & Zhao, L. (2019), ‘Remote sensing image scene classification using cnn-capsnet’, *Remote Sensing* **11**(5), 494.

Zhang, Z. (2016), 'Derivation of backpropagation in convolutional neural network (cnn)', *University of Tennessee, Knoxville, TN*.

Zhou, T., Fu, H., Sun, C. & Wang, S. (2021), 'Shadow detection and compensation from remote sensing images under complex urban conditions', *Remote Sensing* **13**(4), 699.



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

Postboks 5003
NO-1432 Ås
Norway