Norges miljø- og biovitenskapelige universitet

Masteroppgave 2024 30 stp
Fakultetet for realfag og teknologi

Automatisk klassifisering av naturlandskap ved bruk av maskinlæring og terrestriske bilder

Automated Classification of Natural Landscapes Using Machine Learning and Terrestrial Imagery

Lars Michaloff Nordby
Geomatikk

Forord

Ved fullførelsen av denne masteroppgaven markerer dette slutten på min femårige studietid her ved Norges miljø- og biovitenskapelige universitet (NMBU). Det har vært fem fantastiske år!

Jeg vil starte med å rette en stor takk til min hovedveilederen, Misganu Debella-Gilo ved NMBU, og biveileder Stefano Puliti ved NIBIO for veiledning av oppgaven. Takk for gode tilbakemelinger og veilding. Den kunne ikke blitt gjort uten dere. Videre vil jeg takke medstudenter på geomatikkrommet for å gjøre skriveperioden særdeles mye lettere, artigere og for alle de faglige diskusjonene vi har hatt.

Jeg vil også rette en stor takk til Den X-Clusive Stiftelsen PB, Studentsamfunnet i Ås, Atlas; linjeforeningen for geomatikk, samt medstudenter og professorer for å ha gjort studietiden så bra som den har blitt, og for å ha gitt meg rom til å vokse faglig og sosialt.

Fem år har gått, jeg har ledd, jeg har grått

Ikke viste jeg at dette var selve livet

Og at man skulle ende opp med å gå rundt en enebærbusk,

Enebærbusk,

Enebærbusk

Tidlig en onsdags morgen i Mai

15 Mai 2024, Ås NMBU

Lars Mcihaloff Nordby

Sammendrag

Denne masteroppgaven undersøker muligheten for å automatisk identifisere og segmentere ut enebærbusker fra terrestriske bilder tatt i forbindelse med Landsskog-takseringsprosjektet til NIBIO. Målet har vært å utvikle en maskinlæringsmodell som kan identifisere og lokalisere enebærbusker i bildet. Det er også undersøkt om modelltreningen kan suppleres med et eksternt datasett. Segmentering og klassifisering er brukt for å oppnå disse målene. Modellene er trent opp på datasett inneholdende bilder fra NIBIO og iNaturalist. Modellene har deretter blitt testet på bilder fra NIBIO, og valideringsparameterne har blitt sammenlignet for å finne den beste modellen.

Modeller fra YOLOv8-arkitekturen har blitt benyttet i både klassifiserings- og segmenteringsoppgavene. Resultatene viser at alle segmenteringsmodellene generelt presterer dårlig, mens klassifiseringsmodellen med en justert konfidensnivå leverer de beste resultatene.

Imidlertid viser resultatene at segmenteringsmodellene har potensial til å oppnå gode resultater med tilstrekkelig datagrunnlag. Modellen som er trent på kombinasjonsdatasettet tyder på at ved å bruke et eksternt datasett, oppnås en modell som er mer sikker og nøyaktig i prediksjonene sine, og viser tegn til mer robusthet. Oppgaven konkluderer derfor med at det er mulig å lage en maskinlæringsmodell for automatisk klassifisering av enebærbusker i terrestriske bilder. Fremtidig arbeid bør fokusere på mer datainnsamling og utprøving av andre modellarkitekturer.

Abstract

In this master thesis, the aim is to find out if it is possible to identify and segment out Juniper from terrestrial pictures taken during the fieldwork of NIBIO's National Forest Inventory (NFI) project. The goal of the thesis has been to develop a machine learning model that can identify and localize juniper in the pictures. It is also studied if the model training could benefit from the addition of more data from an external dataset. Segmentation and classification are used to try to achieve these goals. The models are trained on datasets containing images from NIBIO and iNaturalist. The models have then been tested on pictures from NIBIO, and the validation parameters compared to find the best model.

The model architectures used are from the YOLOv8 models, and this architecture has been used in both classification and segmentation tasks. The results show that all the segmentation models generally perform poorly, while the classification model with an adjustment in the confidence level delivers the best results.

However, the results show that the segmentation models still have some potential to perform better by training on a sufficient dataset. The model trained on a combination dataset shows that using an external dataset could result in a model that is more accurate and confident in its predictions and shows signs of delivering a more robust model. The thesis concludes, therefore, that it is possible to develop a machine learning algorithm that can detect juniper in terrestrial pictures. Further work should be focused on gathering a sufficient dataset and trying out different model architectures.

Innhold

Figurer

Tabeller

Forkortelser

CNN Convelution nural network

COCO Common Objects in Context

ESA European Space Agency

FN falsk negativ

FP falsk positiv

GIS geografiske informasjons systemer

IoU Intersection over union

KI kunstg intelligens

mAP mean average precision

NASA National Aeronautics and Space Administration

NFI National Forest Inventory

NIBIO Norsk institutt for bioøkonomi

NIN Natur i Norge

SAM Segment Anything Model

TN sann negativ

TP sann positiv

YOLO You Only Look Once

1 Introduksjon

1.1 Bakgrunn

Norge er et langt og variert land. Det er derfor viktig at vi som nasjon har en oversikt over hvilke arter og naturtyper vi har i hele landet. Dette arbeidet, som er vedtatt av Stortinget og er utviklet i samarbeid med Miljødirektoratet, har resultert i Natur i Norge (NIN)-systemet. Målet med NIN-systemet er å gi et felles vokabular for norsk natur og tilby et verktøy som muliggjør sammenligning av all natur i landet. Systemet er bestemt at skal brukes i alle offentlige kartlegginger av natur[1].

Som en del av dette arbeidet har Norsk institutt for bioøkonomi (NIBIO) fått ansvaret med å utvikle landskog-takseringen (NFI). Målet er å gi en oppdatert oversikt over landets skogressurser, som inkluderer areal-, ressurs- og miljødata[2]. Kartleggingen omfatter permanente prøveflater som besøkes med en kartleggingsfrekvens på hvert femte år, noe som innebærer at en femtedel av flatene kartlegges hvert år. Med unntak av noen regionale forskjeller, er størrelsen på hver prøveflate 3x3 km. Hovedarbeidet med etableringen av prøveflatene ble utført fra 1986 til 1993, og i 2011 ble etableringen av flatene for hele landet fullført[3, s.7].

Flatene skal kartlegges ved hjelp av NIN-systemet. Fra 2023 og frem mot 2027 vil det også bli tatt bilder av disse flatene for å kvalitetssikre dataene. Dette gjør det mulig å verifisere i ettertid at de er klassifisert i riktig NiN-naturtype. Fremtidig gjentatte bildetaking vil bidra til å dokumentere eventuelle endringer som har funnet sted siden sist prøveflaten ble undersøkt. Dette vil også tillate etterregistrering av andre arter som ikke var en del av den opprinnelige feltkartleggingen.

Med utviklingen innenfor maskinlæring og KI som har skjedd det siste årene er det på vei til å implementeres innen flere og flere fagkretser og applikasjoner. Når det kommer til maskinlæring innenfor vegetasjonskartlegging er mye av fokuset og utviklingen vært fokusert på fjernmåling fra satellitt og fly/drone bilder. Terrestriske bilder har ikke blitt brukt like mye, innenfor terrestriske bilder er det mer hverdagslige objekter og datasyn, til for eksempel selvkjørende biler,

som har vært fokusert på.

NIBIO jobber med å utvikle en modell som kan segmentere ut en rekke arter og objekter fra disse bildene automatisk. Som en del av dette arbeidet ønsker de å kartlegge hvilke av prøveflatene som inneholder arten enebærbusk. Tilstedeværelsen av enebærbusker i en skog tyder ofte på at det er tidligere kulturlandskap. Og tilstedeværelsen av enebærbusker blir ikke registrert i felt, så det er noe som må finnes fra bildene

Sånn NIBIO jobber nå, er det at de tar for seg ett og ett bilde og segmenterer alle artene de ser. Dette er tidkrevende, og det kan gå lang tid mellom hvert bilde med den arten de søker. Så som en annen del av oppgaven ønsker de å vite om man kan trene opp en modell på eksterne bilder. Dette i håp om at det skal være enklere og raskere å trene opp modellen på spesifikke arter av interesse.

1.2 Problemstilling

Denne oppgaven skal undersøke om det er mulig å automatisk finne enebærbusker i terrestriske bilder tatt i forbindelse med National Forest Inventory (NFI) prosjektet. Hovedpunktene man ønsker at en modell skal kunne oppfylle er:

1) Er det enebærbusker i bildet?

2) Hvor i bildet er enebærbusken?

3) Hvor stor andel av bildet er enebærbusk?

I forbindelse med utviklingen av en modell som oppfyller nevnte punkter, skal det også undersøkes om det er mulig å forbedre modellen ved å supplere modelltreningen med et eksternt datasett

2 Teori

2.1 Vegetasjonskartlegging

Vegetasjonskartlegging har siden 1900-tallet hatt økende fokus og interesse. I første del av 1900-tallet utviklet den sveitsiske botanikeren Josias Braun-Blanquet en standardisert metode for å klassifisere vegetasjon. Denne metoden, senere kjent som Braun-Blanquet-metoden, ble standard for vegetasjonskartlegging i Europa i 1930. Utover 1900-tallet utviklet metoden seg og ble adoptert verden over, noe som skapte et felles sammenligningsgrunnlag fordi mange brukte det samme systemet for klassifisering. På slutten av 1990-tallet økte fokuset på naturvern, og interessen for vegetasjons- og naturkartlegging steg. Dette førte til at flere land startet egne prosjekter for å kartlegge egne områder og utvikle egne standardiserte metoder. Eksempler på dette er EUs arbeid med Natura 2000, USAs 'National Vegetation Classification standards' og senere NIN-systemet i Norge. I de senere årene har teknologisk utvikling spilt en viktig rolle i vegetasjonskartlegging, noe som har økt interessen for faget og gjort analyser og kartlegging enklere. Utviklingen innen teknologi for fjernmåling, databaser og geografiske informasjons systemer (GIS) har bidratt til å utvikle fagfeltet og understreket dets betydning for institusjoner og land over hele verden.[4, 5].

Natur i Norge

Arbeidet med NIN systemet i Norge startet i 2005, etter at man så behovet for et mer omfattende og presist system for kartlegging av natur. Den første utgaven av NIN ble lansert i 2009. Siden den gang har det kommet to versjoner til, hvor den siste ble lansert i 2023. Behovet for å presist og systematisk kunne kartlegge naturtyper kommer fra et ønske om at man skal kunne kommunisere på tvers av fagfelt og være sikker på at man snakker om det samme når man refererer til naturtyper. Systemet er bygget opp i tre deler, hvor man deler naturmangfoldet inn i biodiversitet, geodiversitet, og økodiversitet. Naturtyper klassifiseres under økodiversitet[6].

2.2 Bildebruk i vegetasjonskartlegging

I den senere tid har fremskritt innen fjernmålingsteknologi ført til at bilder blir stadig mer tilgjengelige og enklere å produsere. For vegetasjonskartlegging har man flere fjernmålingsplattformer å velge mellom.

Satellittbilder

Bilder fra satellitt har blitt mer tilgjengelige, og mange satellitter leverer bilder gratis. Satellittene bidrar med oppdaterte bilder på faste intervaller over samme område, noe som gir en tidsserie av bilder som kan vise endringer over tid. Satellittbilder har ofte god spektral oppløsning, da de fleste satellittene leverer enten multispektrale eller hyperspektrale bilder. De har også høy temporal oppløsning, det vil si hvor ofte satellitten tar bilder av samme område. På den andre siden har satellittbilder ofte dårlig geometrisk oppløsning, det vil si hvor stort område et piksel dekker på bakken, som ofte er flere meter. Noen av satellittene som finnes og som produserer data tilgjengelig gratis, er fra National Aeronautics and Space Administration (NASA) og European Space Agency (ESA). NASA har siden 1972 drevet et pågående jordovervåkingsprogram med sine Landsat-satellitter. Disse satellittene er hyperspektrale og har siden den første oppskytingen levert en kontinuerlig bildeserie av jordens overflate[7]. ESA har på sin side drevet et jordovervåkingsprogram med sitt Copernicus-program, som startet i 1998. Programmet består av flere satellitter som leverer både radarbilder og multispektrale bilder gratis[8]. Bilder fra satellitter egner seg til å overvåke store områder og over lengre tid, som naturtyper og større geografiske områder[9].

Fly- og dronebilder

Flyfotografering har vært brukt siden første verdenskrig og har blitt stadig mer vanlig ettersom fly- og kamerateknologien har utviklet seg. Flyfoto har ofte en høy geometrisk oppløsning ettersom man selv kan bestemme flyhøyden. På grunn av kostnadene er den temporale oppløsningen sjelden særlig god, ofte med noen års mellomrom, avhengig av hvilket prosjekt og hva som er hovedfokuset med flyvningen. Fordelen med flyfoto er at man raskt og effektivt kan dekke større områder med god geometrisk oppløsning, og metoden er ofte brukt over kommuner, byer og skoger. De senere årene har droneteknologien utviklet seg kraftig. Dette har ført til at man enkelt og svært kostnadseffektivt kan ta bilder som oppnår god geometrisk oppløsning. På grunn

av kostnadseffektiviteten kan man også øke den temporale oppløsningen etter behov. Men på grunn av lovverk og batterikapasitet er området som kan dekkes på en gang begrenset. Droner egner seg til kartlegging av mindre områder som jorder, skogeiendommer og privateiendommer. Med bilder tatt fra luften kan man oppnå en geometrisk oppløsning på mindre enn en cm[9].

Terrestriske bilder

Terrestriske bilder omhandler bilder som er tatt fra bakkenivå, uavhengig av hvilken type kamera som har tatt bildene, så lenge de er tatt fra bakken. Kameraene kan være håndholdt eller festet i et stativ. Det er vanlig å bruke profesjonelle kameraer, men ettersom mobilkameraene nå begynner å levere god kvalitet, blir dette også et godt alternativ. Bildene brukes ofte i profesjonelle sammenhenger, spesielt i forbindelse med nærfotogrammetri, hvor man ønsker å lage 3D-modeller eller utføre nøyaktige målinger av objekter i bildet[10, s.1-27]. Bildene tatt på bakkenivå har ofte god geometrisk og temporal oppløsning. I motsetning til satellitt- og luftbårne kameraer som produserer vertikalbilder, produserer terrestriske bilder skråfoto, altså bilder tatt fra siden. Dette gjør det vanskelig å dekke større områder. Terrestriske bilder egner seg til kartlegging av individuelle tilfeller av vegetasjon, eller områder med liten geografisk utstrekning som skogholt og stillbilder av landskapet[11].

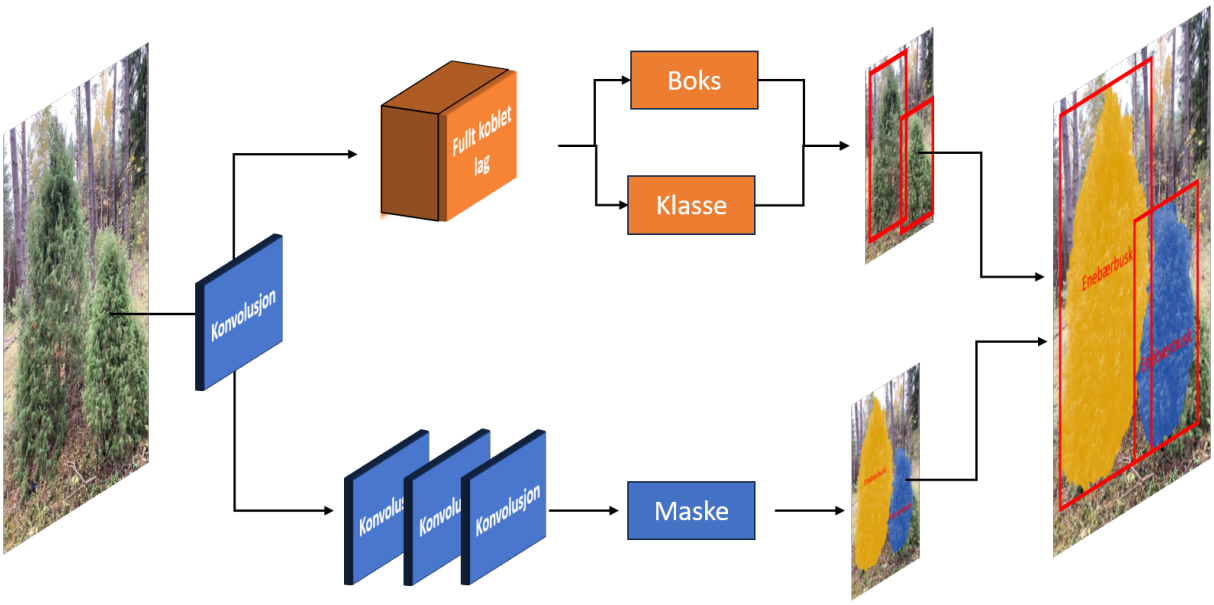
2.3 Maskinlæring i vegetasjonskartlegging

Den mest vanlige bruken av maskinlæring i vegetasjonskartlegging er anvendelsen av ikke-veiledede modeller som K-means og ISODATA, veiledede modeller med treningsdata, og kartlegging ved bruk av indekser. Mesteparten av vegetasjonskartleggingen gjennomføres på bilder fra satellitter og fly/droner, hvor mye av fokuset ligger på dekningsgraden av området for den aktuelle vegetasjonstypen. Disse modellene brukes ofte på multispektrale data som gjør det mulig å skille enkelte vegetasjonstype fra hverandre, men klassene for vegetasjon blir ofte veldig store og går ikke ned på enkeltartsnivå. De holder seg på generelle nivåer som 'Skog', 'Åker', 'Gress'[12]. Ettersom mengden tilgjengelige bilder har vokst, har også behovet for å automatisere mange analyser og kartlegginger økt. Dette, kombinert med utviklingen innen kunstg intelligens (KI) og maskinlæring, har ført til økt fokus på hvordan KI kan implementeres i vegetasjonskartleggingen. Blant metodene er segmentering og klassifisering de mest brukte. Klassifisering er en metode hvor man trener en modell til å bestemme hvilken klasse et bilde

tilhører. Resultatene kommer ut som en sannsynlighet for at bildet tilhører hver enkelt klasse. Dette kan både være flerklassifisering eller binær klassifisering[13]. Et skritt videre er segmentering, hvor man går ned på pikselnivå for å finne ut om hver enkelt piksel tilhører en klasse. Innenfor segmentering finnes det flere typer, hvor semantisk og instanssegmentering er de mest brukte. Semantisk segmentering klassifiserer hver enkelt piksel til å tilhøre en klasse og skiller ikke objekter fra hverandre. Instanssegmentering segmenterer de ulike klassene, men skiller også ulike tilfeller fra hverandre. I et bilde med flere busker vil semantisk segmentering klassifisere alle buskene som samme objekt, mens instanssegmentering vil skille hver busk hver for seg som et eget objekt[14]. Figur 2.1 viser hva resultatet av de forskjellige metodene ville vært på et bilde av enebærbusker.

Figur 2.1: Sammenligning av tre forskjellige maskinlæringsmetoder for prediksjon av enebærbusk. Øverst til venstre: originalbildet, øverst til høyre: klassifisering, viser prosent sannsynlighet for å tilhøre hver klasse, nederst til venstre: instanssegmentering, har segmentert ut hver enkelt busk som egen maske, nederst til høyre: semantisk segmentering, har predikert enebærbuskene som én maske.

Det finnes mange forskjellige arkitekturer å velge mellom innenfor både segmentering og klassifisering. I de senere årene har imidlertid Convelution nural network (CNN) blitt den mest populære. Etter hvert har det kommet nyere og mer avanserte modeller som bygger på CNN. Felles for disse modellene er at de ofte bruker et CNN som første lag for å hente ut bildeegenskaper og strukturer. De nyeste og mest avanserte modellene, slik som Mask R-CNN[15] og YOLO[16], har tatt steget videre fra CNN og kombinerer nå både avgrensningsbokser og instanssegmentering i samme modell. Dette gjør at modellene er mye raskere og mer nøyaktige enn tidligere metoder[17]. Figur 2.2 viser en forenklet modellarkitektur om hvordan de nye modellene kombinerer masker og avgrensningsbokser i én enkelt modell. Det er en forenklet konseptuell figur; en komplett modellarkitektur for YOLO er vist i vedlegget under figur C.1.



Figur 2.2: Forenklet modellarkitektur av Mask-RCNN og YOLO. Bildet viser hvordan Mask-RCNN og YOLO bearbeider et bilde av enebærbusker gjennom ulike trinn. Starter med originalbilde. Konvolusjonslagene ekstrakterer funksjoner fra bildet. Øverste vei:Fullt koblet lag genererer boks- og klasseprediksjoner. Nederste vei: maskeprediksjon som markerer ulike segmenter i bildet

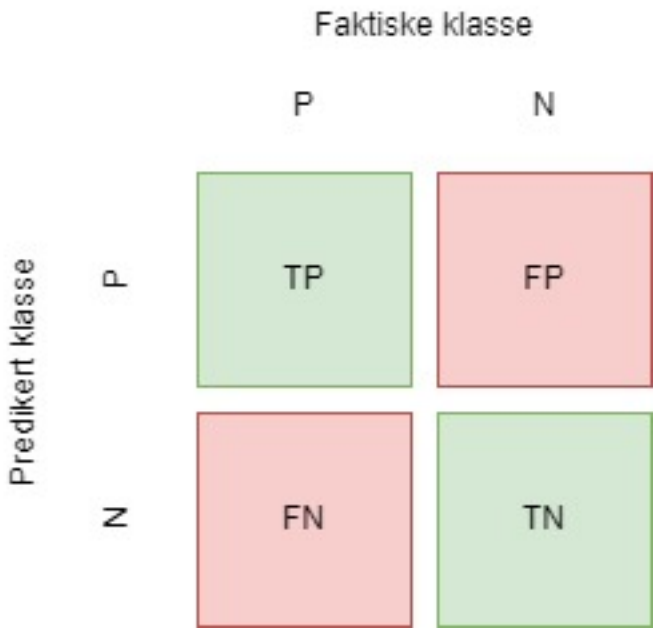
I motsetning til CNN og Mask R-CNN, er ikke YOLO kun en modellarkitektur; det er også en fullt integrert Python-pakke levert av Ultralytics. Dette bidrar til å gjøre modellen mer implementerbar, ettersom mye av infrastrukturen, modulene, og parameterne allerede er utviklet. Dermed er implementeringen av YOLO mye enklere sammenlignet med et tradisjonelt CNN, som må utvikles fra bunnen av enten selvstendig eller gjennom andre pakker som scikit-learn eller lignende maskinlæringspakker.

Ved trening fra bunnen av trenes en modell oftest ved å initialisere alle trenbare vekter i nettverket med tilfeldige verdier. Gjennom treningen justeres disse vektene. Justeringen skjer ved å optimalisere vektene slik at de beveger seg mot et minimum i den valgte kostfunksjonen. Dette er en prosess som ofte tar lang tid og krever store datamengder. Utviklingen de senere årene innenfor maskinlæring har ført til det som kalles overførbar trening (transfer learning). Dette er en metode som brukes for å redusere tiden det tar å trene opp nye nettverk. Metoden innebærer å overføre vekter fra et eksisterende trent nettverk til et nytt nettverk. Dette betyr at det nye nettverket ikke starter med tilfeldig initialiserte vekter, men med vekter fra et nettverk som allerede er trent. Dette konseptet ligner på hvordan menneskelige ferdigheter overføres; for eksempel vil en person som er dyktig til å spille fiolin sannsynligvis lære andre lignende instrumenter raskere på grunn av sin eksisterende musikalske forståelse. Ved å benytte vekter som allerede er nærmere et minimum i kostfunksjonen, kreves det også mindre data for å trene det nye nettverket effektivt. Suksessen til denne metoden avhenger av at nettverket man overfører vektene fra har relevante likheter med oppgaven som det nye nettverket skal løse[18].

2.4 Evalueringsmetode

Forvvirngsmatrise

En forvirringsmatrise er en matrise som fremstiller antallet sann positiv (TP), sann negativ (TN), falsk positiv (FP), og falsk negativ (FN) som en matrise. Denne matrisen gir en oversikt over antallet korrekte og feilaktige prediksjoner, hvor radene representerer de predikerte klassene, mens kolonnene representerer de faktiske klassene. Hver celle i matrisen viser antallet korrekte eller feilaktige prediksjoner for hver kombinasjon av predikert og faktisk klasse. Når det er flere klasser, vil dimensjonene i matrisen være lik antallet klasser[19, s.211].



Figur 2.3: Strukturen til en forvirringsmatrise som illustrerer forholdet mellom faktiske og predikerte klasser. TP (True Positive), FN (False Negative), FP (False Positive), og TN (True Negative) representerer de fire mulige utfallene av en binær klassifiseringsmodell. Illustrasjonen er inspirert av [19, s.212]

Accuracy

Nøyaktigheten (ACC) av en modell gir et mål på hvor mange prosent av det totale antallet prediksjoner som er korrekt predikert. I maskinlæringsammenheng refereres dette ofte til som 'Accuracy'[19, s.213].

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

Precision

Presisjonen (PRE) av en modell omtales i maskinlæringsammenheng som 'Precision' og indikerer hvor godt modellen klarer å identifisere sanne positive tilfeller (TP) blant alle positive klassifiseringer (TP+FP). Dette gir et mål på hvor nøyaktig modellen er til å predikere positive tilfeller. Presisjon er et viktig mål i modeller hvor det er kritisk å minimere antallet falske positive (FP)[19, s.214].

$$PRE = \frac{TP}{TP + FP} \tag{2.2}$$

Recall

Gjenkalling (REC) av en modell refereres til som 'Recall' i maskinlæringskontekster og angir hvor godt modellen klarer å identifisere sanne positive tilfeller (TP) blant alle faktiske positive tilfeller (TP+FN). Dette er et viktig mål i situasjoner hvor det er essensielt å minimere antallet falske negative (FN)[19, s.214].

$$REC = \frac{TP}{FN + TP} \tag{2.3}$$

F1-score

F1-scoren gir et mer helhetlig bilde av modellens ytelse ved å balansere både gjenkall og presisjon. Denne metrikken er nyttig i situasjoner hvor både gjenkall og presisjon anses som like viktige, det vil si at man ønsker å minimere både antall falske negative (FN) og falske positive (FP). F1-scoren gir dermed en bedre indikasjon på modellens totale prestasjon i slike tilfeller[19, s.214].

$$F1 = 2 \times \frac{PRE \times REC}{PRE + REC} \tag{2.4}$$

IoU

Intersection over union (IoU) er et mål brukt i modeller som predikerer en avgrensningsboks (bounding box). Det anvendes i instanssegmentering, objektlokalisering og objektsporing. IoU måler hvor mye overlapp det er mellom den predikerte boksen og den faktiske boksen, og gir dermed et mål på hvor nøyaktig modellen er i å estimere den faktiske avgrensningsboksen. Formelen for IoU er en fraksjon der overlappet mellom de to boksene deles på det totale området som de to boksene sammen dekker. En IoU-score på 1 indikerer perfekt overlapp, mens en score på 0 betyr at det ikke er noen overlapp i det hele tatt[20].

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{2.5}$$

Her representerer A den predikerte avgrensingsboksen, mens B er den faktiske avgrensingsboksen

mAP

mean average precision (mAP) er et mål brukt for å evaluere hvor effektivt modeller for objekt-deteksjon, instanssegmentering eller objektsporing presterer. mAP er definert som arealet under en presisjon-gjenkallkurve. For å beregne AP, må man først regne ut gjenkall og presisjon for hver klasse. Tallet som følger mAP, ofte 50 eller 95, angir IoU-terskelen som kreves for at en deteksjon skal regnes som en sann positiv (TP), falsk positiv (FP), falsk negativ (FN), eller sann negativ (TN). For eksempel innebærer mAP50 at modellen må predikere en avgrensningsboks med en IoU på over 50% for at det skal regnes som en TP. Når man har beregnet Average Precision (AP) for hver klasse, tar man gjennomsnittet av disse verdiene for å finne mAP-verdien. Hvis det kun er én klasse involvert, vil mAP og AP være identiske[21].

$$mAP = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i \tag{2.6}$$

Her er N antallet klasser i modellen, mens AP er arealet under presisjon-gjenkall kurven

3 Metode

3.1 Datagrunnlag

3.1.1 NIBIO bildene

National Forest Inventory (NFI) landskog-takserings på norsk, er en del av NIBIO sitt arbeid med å karlegge landest skogressurser. Data som samles inn er med på å gi et godt grunnlag for landsdekkende areal-,ressurs- og miljødata. Dataen oppdateres jevnlig for å øke det samfunnsnyttige resultatet. Prosjektet består av en rekke faste prøveflater som dekker hele landet[2].

Fra 2023 og fremover er det besluttet at kartleggingen skal inkludere bilder fra testfeltene for å kvalitetssikre den innsamlede dataen. For hvert testfelt skal det tas ett bilde i hver av de fire himmelretningene, samt ett nedover for å dokumentere hvilke vekster som finnes i området[3, s.65].

Bildene som brukes i denne oppgaven er levert av NIBIO og er allerede annotert. Det var tilgang på 1547 bilder, hvorav 173 inneholdt forekomster av enebærbusker. Hvert bilde er tatt med et kamera på en smarttelefon og har en oppløsning på 4160 x 3120 piksler. Bildene ble tatt i forbindelse med feltarbeid utført i sommerhalvåret 2023. Bildene i oppgaven representerer et tilfeldig utvalg fra alle bildene som ble tatt.

Figur 3.1: Eksempler på NIBIO bilder tatt i forbindelse med NFI feltarbeidet sommeren 2023

3.1.2 Inaturalist bildene

iNaturalist er et ideelt sosialt nettverk som bidrar til å spre informasjon om biomangfold og hjelper brukerne med å lære mer om naturen. Nettstedet tilbyr muligheten til å laste opp bilder av ulike levende biologiske arter, sammen med artsnavn og posisjonen for observasjonen. Dette gjør det til en verdifull database hvor man kan hente ut mange bilder av spesifikke arter gratis[22].

Gjennom en åpen API ble det opprinnelig lastet ned 10 000 bilder av arten 'Juniperus communis'. Av disse ble 5000 bilder gjennomgått, og det ble valgt ut bilder som ikke var nærbilder, ikke inneholdt snø, og var fri for fremmede objekter som sko og hender. Blant de egnede bildene ble 1000 lastet opp til TrainYOLO (3.2), hvor de ble annotert. Etter nærmere kvalitetssikring resulterte dette i 969 bilder med segmenterte enebærbusker. Bildene er tatt med forskjellige kameraer, og de fleste har en oppløsning på 375x500 piksler. Bildene er tatt over hele verden og til alle årstider, og kan stamme fra hvilket som helst år siden oppstarten av tjenesten i 2014. Fleste-

parten av bildene er trolig fra 2017 og senere, da nettstedet åpnet seg opp for ikke-akademiske brukere.

Bilde med snødekke, ikke egnet for identifikasjon

Bilde med hånd som holder planten, distraherende element

Bilde som er for nærme, vanskelig å identifisere hele planten

Godt bilde som viser hele planten tydelig og på god avstand

Figur 3.2: Eksempler på bilder fra iNaturalist med kvalitetsvurderingene brukt i sorteringen av egnede bilder

3.2 Språk, plattform og programvare

Python

Python er et mye brukt programmeringsspråk som er kjent for å være raskt og enkelt å lære. Språket er populært verden over og tilbyr mange integrerte pakker. Koden er open-source, noe som betyr at all koden er tilgjengelig. Dette gjør det lett å utvikle egne pakker for spesialiserte formål[23]. I denne oppgaven er Python brukt som hovedprogrammeringsspråk. Koden er kjørt

i Google Colab.

Google Colab

Google Colaboratory, ofte kalt Google Colab, er en tjeneste fra Google som tillater kjøring av kode i skyen på kraftigere utstyr enn det brukeren selv har tilgjengelig. Denne plattformen gir også mulighet til å koble seg til Google Drive, noe som forenkler lagring av resultater og bruk av data.

Tjenesten er gratis, men ved intensiv bruk eller store datamengder kan man gå tom for de tildelte ressursene. Derfor er Google Colab Pro+ benyttet for å få tilgang til mer lagringsplass og kraftigere grafikkort for behandling av bilder i oppgaven. Grafikkortets ytelse har stor innvirkning på hastigheten for trening av modeller. Tilgjengeligheten av de kraftigste grafikkortene er begrenset og tildeles tilfeldig basert på abonnementstype. Det har derfor ikke alltid vært mulig å sikre tilgang til det kraftigste grafikkortet[24].

Tabell 3.1: Tilgjengelige grafikkkort og GPU-ram i Google Colab brukt i oppgaven

Grafikk kort navn:	Tilgjengelig GPU-ram:
A100	40Gb
v100	16Gb
T4	15Gb
CPU	0Gb

Ultralytics

Ultralytics er et firma som arbeider for å gjøre kunstig intelligens-teknologi mer tilgjengelig og brukervennlig. De er kjent for å utvikle AI-modeller innenfor computer vision, samt programvarepakker knyttet til dette. De er spesielt kjent for sin You Only Look Once (YOLO) modell, som har blitt meget populær de siste årene[25].

Train YOLO

TrainYOLO er en plattform basert på Ultralytics'(3.2) YOLO modell, og tilbyr funksjonalitet for å laste opp bilder, annotere dem, trene modeller og sammenligne forskjellige versjoner av modellene som utvikles[26]. Plattformen har sin egen Python-pakke, som muliggjør bruk av deres API.

NIBIO benytter denne plattformen til å automatisere bildeanalysen i NFI prosjektet. Bildene fra NIBIO og iNaturalist er først lastet opp og annotert på TrainYOLO før de benyttes videre i treningen av modellen.

Comet ML

Comet ML er et firma som fokuserer på å utvikle en programvareplattform som hjelper til med å optimalisere prosessene for maskinlæring og dyp læring. Plattformen er gratis og tilbyr en API som kan integreres i treningskoden. Denne APIen logger alle treningsparametre og produserer automatisk visuelle fremstillinger av disse. Programvaren lagrer også tidligere versjoner og gjør det enkelt å sammenligne forskjellige versjoner av modellen og de anvendte hyperparametrene. Det kreves en spesifikk Python-pakke for å kunne bruke APIen til denne nettsiden[27].

3.3 Preprosessering

3.3.1 Nedlasting

Bildene fra iNaturalist ble lastet ned ved hjelp av Inaturalist sin API i R-studio. Pakken som brukes heter rinat"og gir tilgang til databasen, noe som muliggjør nedlasting av et spesifisert antall bilder av bestemte arter. For denne oppgaven ble det lastet ned 10 000 bilder av arten 'Juniperus communis'.

3.3.2 Annotering

Etter nedlastingen ble bildene lastet opp til TrainYOLO-nettstedet (se kap:3.2). Her ble den innebygde segmenteringsfunksjonen brukt til manuelt å annotere hvert bilde. Programmet inneholder et penselverktøy som gjør det enkelt å tegne inn i bildet hvor enebærbusken befinner seg, og programmet genererer da en tilhørende maskeringsfil automatisk. For å effektivisere prosessen benyttes Meta sin Segment Anything Model (SAM) til automatisk å segmentere ut objekter i bildet[28]. Dette akselererer prosessen med å annotere enebærbusker betraktelig, da man umiddelbart får en maske på busken. Brukeren kan deretter justere masken for å sikre at den er så nøyaktig som mulig. Bildene fra både NIBIO og iNaturalist ble annotert på denne måten, men ikke av samme person.

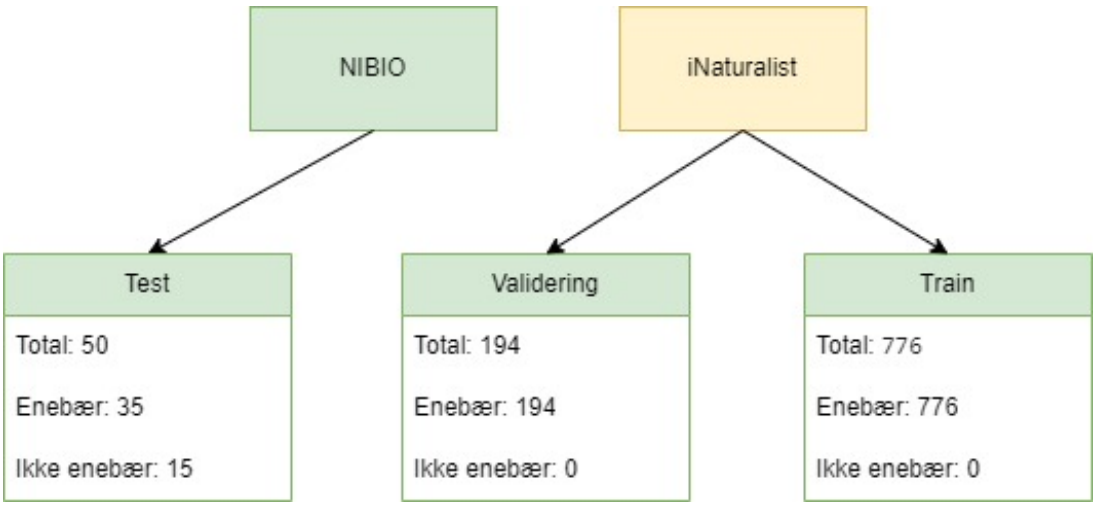
3.3.3 Datainndeling

Siden oppgaven skal teste forskjellige datasett og hvordan modellen håndtere dette var det nødvending å lage fire forskjellige trening- og validerings-sett. Tre til segmentering og en til klassifisering. For testdataen ble det bruk samme datasett for å gi et sammenligningsgrunnlag. Testsettet inneholder kun bilder fra NIBIO og består av både bilder med og uten enebærbusker for å sjekke om den klarer å skille mellom dem.

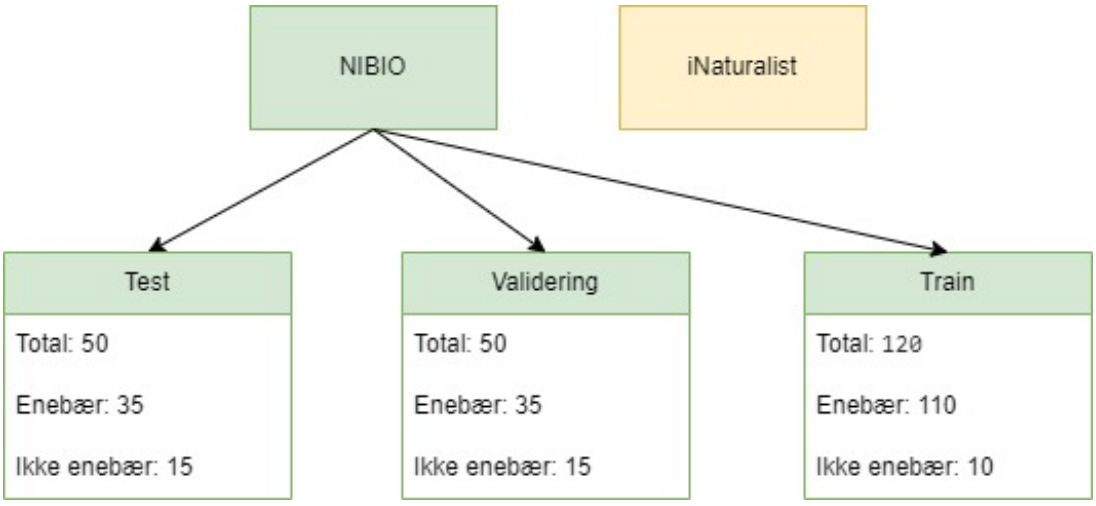
Siden oppgaven skal teste påvirkningen av eksterne datasett og hvordan modellen håndterer dette, var det nødvendig å lage fire forskjellige trenings- og valideringssett. Tre av disse er for segmentering og ett for klassifisering. For testdataene ble det brukt det samme datasettet for å ha et sammenligningsgrunnlag. Testsettet består kun av bilder fra NIBIO, og inkluderer bilder både med og uten enebærbusker for å teste om modellen klarer å skille mellom dem.

Det første datasettet inneholder kun bilder fra iNaturalist (Figur 3.3). Det andre datasettet består utelukkende av bilder fra NIBIO (Figur 3.4), mens det tredje datasettet inneholder en kombinasjon av bilder fra både NIBIO og iNaturalist (Figur 3.5). Ettersom fokuset primært er på nøyaktigheten av prediksjoner på NIBIO-bildene, inneholder valideringssettet for kombinasjonsmodellen kun bilder fra NIBIO.

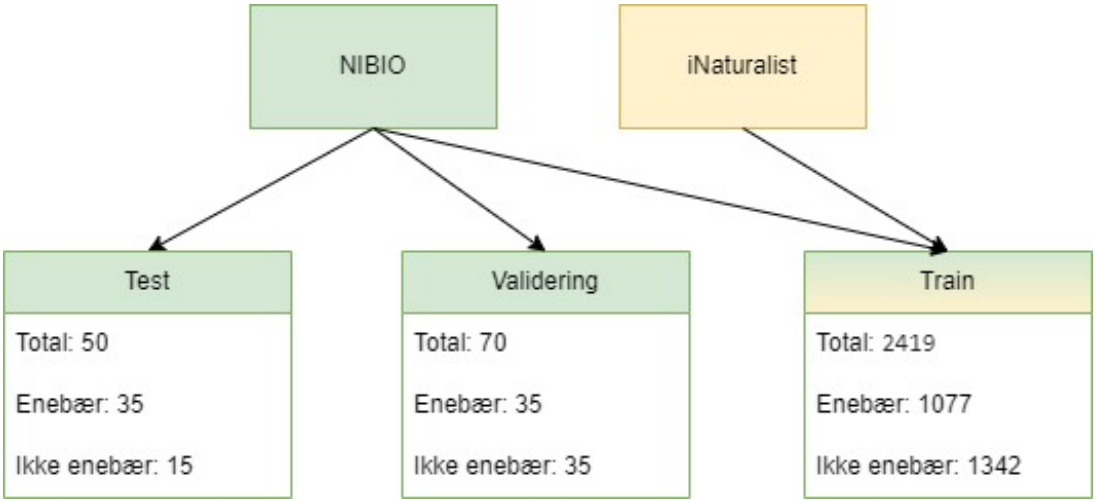
For klassifiseringsdatasettet (Figur 3.6) er dataene delt slik at 30% utgjør valideringsdata og 70% treningsdata. Her er 30% tatt fra både NIBIO- og iNaturalist-bildene tatt i bruk for validering.



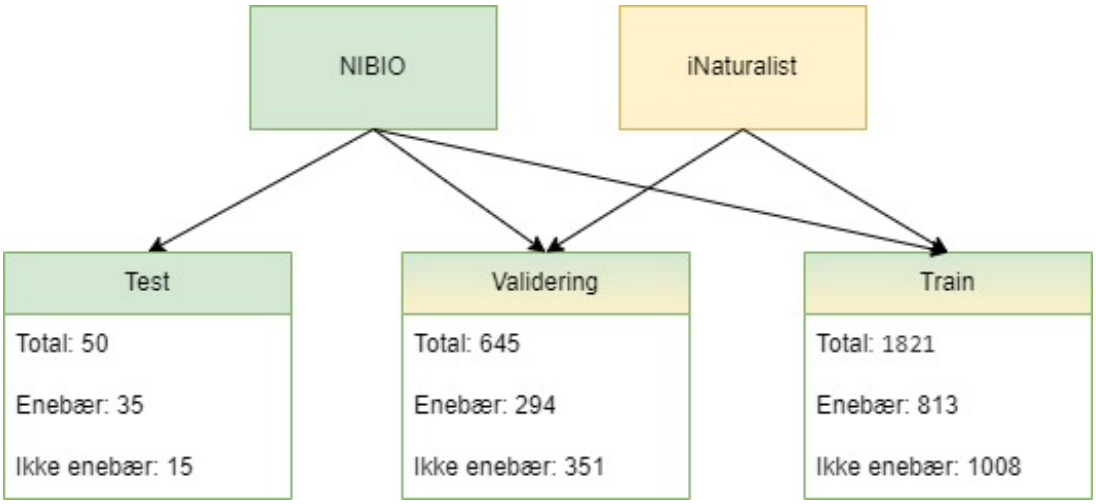
Figur 3.3: Oppdeling av iNaturalist datasettet i trenings-, validerings- og testsett, med spesifisering av antall bilder av enebær og ikke enebær for hvert sett



Figur 3.4: Oppdeling av NIBIO datasettet i trenings-, validerings- og testsett, med spesifisering av antall bilder av enebær og ikke enebær for hvert sett



Figur 3.5: Oppdeling av Combo datasettet i trenings-, validerings- og testsett, med spesifisering av antall bilder av enebær og ikke enebær for hvert sett



Figur 3.6: Oppdeling av Klassifiserings datasettet i trenings-, validerings- og testsett, med spesifisering av antall bilder av enebær og ikke enebær for hvert sett

Mappestruktur

For å kunne trene modellene effektivt med YOLOv8, må bildene være organisert i en korrekt mappestruktur og inneholde de riktige filene. Det er forskjell i filstrukturen mellom segmentering sammenligneg og klassifisering. For en detaljert beskrivelse av den nødvendige mappe- og filoppsettet, se B.

3.4 Segmentering

I håp om å adressere alle tre problemstillingene som ble stilt i begynnelsen av oppgaven, ble instanssegmentering valgt som en passende metode. Denne teknikken tillater oss å svare på om det er en enebærbusk i bildet, hvor i bildet enebærbusken befinner seg, og med litt ekstra prosessering, hvor stor andel av bildet som er enebærbusk.

3.4.1 YOLOv8-segmentering

Modeller

YOLOv8 tilbyr fem forskjellige versjoner av forhåndstrente modeller, som alle er trent på Microsoft sitt Common Objects in Context (COCO) datasett. Dette datasettet gir grunnlaget for å overføre vekter og biaser som allerede er optimalisert for å segmentere objekter. Bruk av disse forhåndstrente modellene bidrar til mer effektiv trening siden modellene allerede har kunnskap

om hvordan objekter segmenteres. De fem modellene varierer i størrelse og kompleksitet fra det minste, YOLOv8 nano (YOLOv8n), til det største, YOLOv8 Extra large (YOLOv8x). Ved å velge en mer kompleks modell øker man antall parametere, noe som påvirker prosesseringshastigheten, men man oppnår også mer nøyaktig segmentering på COCO datasettet (se tabell:3.2)

Tabell 3.2: Sammenligning av YOLO-segmenteringsmodellene. Verdiene er fra tester utført på COCO og er hentet fra Ultralytics[29]

Modell	mAP(50-95% mask)	Antall prametere(M)	Hastighet(A100 (ms))
YOLOv8n-seg	30.5	3.4	1.21
YOLOv8s-seg	36.8	11.8	1.47
YOLOv8m-seg	40.8	27.3	2.18
YOLOv8l-seg	42.6	46.0	2.79
YOLOv8x-seg	43.4	71.8	4.02

Moduser

YOLOv8 tilbyr seks forskjellige moduser å velge mellom: 'Train', 'Val', 'Predict', 'Export', 'Track', og 'Benchmark'. For denne oppgaven ble 'Train', 'Val', og 'Predict' brukt. Hver av disse modusene har sitt eget sett med hyperparametere og innstillinger som kan tilpasses når koden kjøres. For en komplett oversikt over alle innstillingene, se dokumentasjonssidene for Trening[30], Validering[31], og Predikering[32]. Detaljer om de brukte hyperparameterne for hver modell er inkludert i vedlegg A.

Modellene ble først trent i 'Train'-modus med de spesifikke trenings- og valideringsdatasettene. Etter å ha trent de beste modellene for hvert datasett, ble hver modell evaluert i 'Val'-modus for å etablere et sammenligningsgrunnlag. Dataene som ble brukt er organisert som vist i figur:B.1.

Hyperparametere

På grunn av treningens ressurskrav, ble det ikke utført noen systematisk justering av hyperparametrene, som 'grid search' eller 'random search'. De mest testede og justerte parameterne inkluderer bildestørrelse (imgsz), modelltype (model), antall epoker (epochs), og batch-størrelse (batch). Øvrige hyperparametre ble satt til standard for den aktuelle modellen eller automatisk justert via modellens 'auto'-funksjon der det var mulig. Dette gjelder spesielt parametere som 'batch', 'optimizer', 'lr0', 'momentum', og 'augmentation', der modellen selv optimaliserer de beste verdiene for den gitte oppgaven.

Hver modell ble trent flere ganger med ulike parametre for å vurdere eventuelle endringer i

prestasjon og nøyaktighet. Til dette formålet ble CometML (kap:3.2) brukt for å sammenligne de ulike iterasjonene av modellen.

Trening

Flere modeller har blitt trent på de datasettene som er illustrert i figurene:3.3, 3.4, og 3.5. For hvert datasett har flere modeller blitt trent med mål om å identifisere den modellen som yter best ved å justere hyperparameterne. Treningsprosessen ble utført sekvensielt, startende med iNaturalist-bilder, etterfulgt av NIBIO-bilder, og til slutt en kombinasjon av begge i Combo-datasettet.

For bildene fra iNaturalist (iNaturalist (iNat)) startet treningen med en modell basert på halvparten av datasettet, det vil si 500 bilder. Dette ble gjort for å evaluere hvor mange bilder som var nødvendige for å oppnå en funksjonell modell. Deretter ble det trent en annen modell på hele datasettet, 1000 bilder, for å sammenligne effektiviteten og forbedre ytelsen.

3.5 Klassifisering

For å svare på det første spørsmålet stilt i problemstillingen (kapittel:1.2), om det er en enebærbusk i bildet eller ikke, ble klassifisering benyttet. Målet var å utvikle en robust modell som effektivt kan identifisere tilstedeværelsen av enebærbusker i bildene.

3.5.1 YOLOv8-klassifisering

Som i kapittel: 3.4.1, finnes det også fem varianter av klassifiseringsmodellen, som er oppsummert i tabell 3.3. I motsetning til segmenteringsmodellene, er klassifiseringsmodellene forhåndstrent på ImageNet-datasettet, ikke på COCO. Dette skyldes at oppgaven krever klassifisering fremfor segmentering. På grunn av deres spesifikasjon er disse modellene betydelig mindre og raskere enn segmenteringsmodellene.

Tabell 3.3: Sammenligning av YOLO-klassifiseringsmodellene. Verdiene er fra tester utført på ImageNet og er hentet fra Ultralytics[30]

Modell	Acc (Top-1)	Antall parametere (M)	Hastighet (A100 ms)
YOLOv8n-cls	69.0	2.7	0.31
YOLOv8s-cls	73.8	6.4	0.35
YOLOv8m-cls	76.8	17.0	0.62
YOLOv8l-cls	76.8	37.5	0.87
YOLOv8x-cls	79.0	57.4	1.01

Klassifiseringen innebar trening av flere modeller, hvor størrelsen på modellen ble endret som den primære hyperparameteren. Treningen ble utelukkende utført på klassifiserings-datasettet(figur: 3.6, mens fremgangsmåten ellers var som beskrevet i segmenteringskapittelet (3.4.1). Et unntak var valideringsprosessen, som ble utført i 'Predict'-modus på testdataene i stedet for 'Val', for å muliggjøre etterbehandling av prediksjonene.

3.5.2 Tweak

For å forbedre ytelsen til klassifiseringsmodellen, ble det foretatt en justering i etterprosesseringen av den beste klassifiseringsmodellen. Modellen ble satt i 'Predict'-modus, noe som innebærer at den predikerer hvilken klasse et bilde tilhører. Det lagres også en fil med sannsynligheten for at bildet tilhører hver klasse. I den opprinnelige modellen er det satt en terskel på 50%, noe som betyr at modellen klassifiserer et bilde som tilhørende den klassen som oppnår en sannsynlighet over denne terskelverdien. 'Tweaken' bestod i å endre denne terskelen i etterprosesseringen, ved å gå gjennom disse filene og justere når og hvordan klassene blir klassifisert.

3.6 Evaluering

For å kunne sammenligne hver av modellene på en konsistent måte, er det viktig å benytte samme testdatasett. Dette muliggjør en objektiv vurdering av hvilken modell som presterer best. Videre er det essensielt å anvende valideringsparametere som reflekterer modellens ytelse. De seks vanligste målene i dyplæringssammenheng er: 'Accuracy' (kapittel:2.4), 'Precision' (kapittel:2.4), 'Recall' (kapittel:2.4), 'F1-score' (kapittel:2.4), mAP50 og mAP50-95 (kapittel:2.4).

Disse beregnes ut i fra forvirringsmatrisen (kapittel:2.4) og med gjeldende IoU terskel (kapittel: 2.4).

3.7 Bruk av KI i oppgaven

Denne oppgaven bruker kunstig intelligens (KI) i form av OpenAI sin "ChatGPTsom et hjelpemiddel for feilsøking og dokumentasjon av programvaren. KI er også brukt til å gjennomgå avsnitt for å korrigere skrive- og grammatikkfeil, samt til å hjelpe med oversettelse av fagutrykk. ChatGPT er ikke brukt til å finne kilder eller som kilde i oppgaven, men utelukkende som et hjelpemiddel. Det er benyttet med et kritisk blikk, der man ikke stoler blindt på resultatene. Noen av spørsmålene som er stilt til ChatGPT er:

- Hvordan endrer jeg språkinnstillingene i LaTeX?
- Jeg vil vise en mappestruktur i LaTeX, hvordan gjør jeg det?
- Jeg får denne feilmeldingen når jeg kjører koden [...] har du noen forslag til hvorfor det ikke fungerer?
- Les over dette avsnittet og gi tilbakemeldinger på språket

4 Resultat

De beste modellene som er brukt til sammenligning og videre i resultatene, omtales heretter som NFI-, iNat- (iNat-1000), Combo-, Klassifisering-, Tweak- og iNat-500-modellen. Klassifisering og Tweak er samme modell, men med forskjellige konfidensverdier. NFI-modellen er navnet på modellen som er trent på bildene fra NIBIO-datasettet (figur: 3.5).

Tabell 4.1: Modellnavn og parametere, inkludert antall epoker, for de beste YOLO-modellene i hver kategori brukt videre i resultatene

Navn / Parameter	Modell	Epoker
NFI	YOLOv8m-seg	300
iNat	YOLOv8n-seg	100
Combo	YOLOv8n-seg	200
Klassifisering	YOLOv8n-cls	100
Tweak	YOLOv8n-cls	100
iNat-500	YOLOv8n-seg	122

4.1 Kvantitative resultater fra segmentering

For å evaluere hvor gode modellene er og for å kunne sammenligne dem, er det brukt forvirringsmatriser (kapittel:2.4) og valideringsparametrene nøyaktighet (kapittel:2.4), presisjon (kapittel:2.4), gjenkall (kapittel:2.4), F1-score (kapittel:2.4), mAP50 og mAP50-95 (kapittel:2.4). Alle forvirringsmatrisene til segmenteringsmodellene har 0 tilfeller av TN. Dette er fordi det bare er en klasse som segmenteres, og derfor er det ingenting som har faktisk klasse som bakgrunn/ikke enebærbusk. Resultatene er fra validering på testdataen som består av kun bilder fra NFI (kapittel: 3.3.3).

NFI-modellen har de høyeste verdiene for nøyaktighet, presisjon og F1-score, henholdsvis 0.28, 0.38 og 0.44, og nest høyest gjenkall, mAP50 og mAP50-95 med henholdsvis 0.54, 0.27 og 0.12.

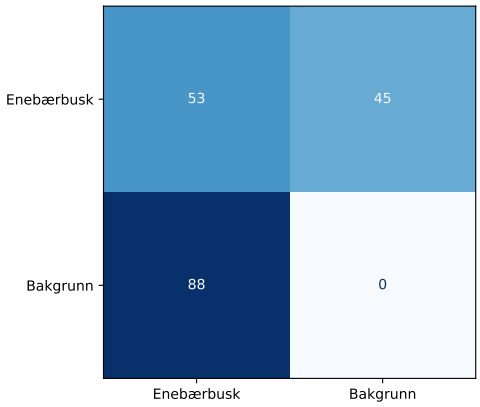
Combo-modellen har best resultat på gjenkall, mAP50 og mAP50-95 med henholdsvis 0.64, 0.29 og 0.16. Modellen har også nest best resultater på nøyaktighet, presisjon og F1-score, med

0.21, 0.24 og 0.35.

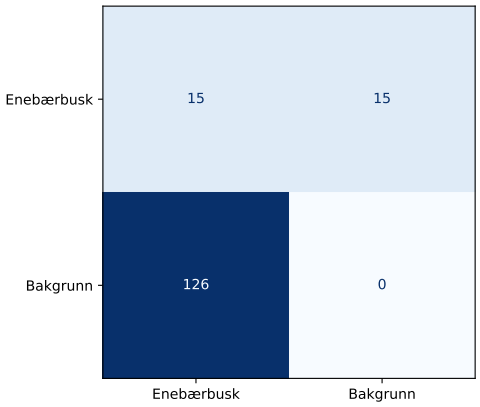
iNat-modellen gjør det klart dårligst på alle parametrene, med nøyaktighet på 0.10, presisjon på 0.11, gjenkall på 0.50, F1-score på 0.18 og mAP50 på 0.03. mAP50-95 var ikke tilgjengelig å regne ut, men den antas å være tilnærmet lik 0.

Tabell 4.2: Resultater fra valideringsparameteren for segmenteringsmodellene NFI, iNat og Combo

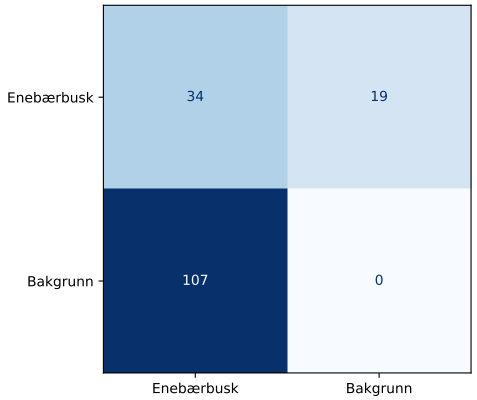
Score / Modell	NFI	iNat	Combo
Accuracy	**0.28**	0.10	0.21
Precision	**0.38**	0.11	0.24
Recall	0.54	0.50	**0.64**
F1-score	**0.44**	0.18	0.35
mAP50	0.27	0.03	**0.29**
mAP50-95	0.12	NaN	**0.16**



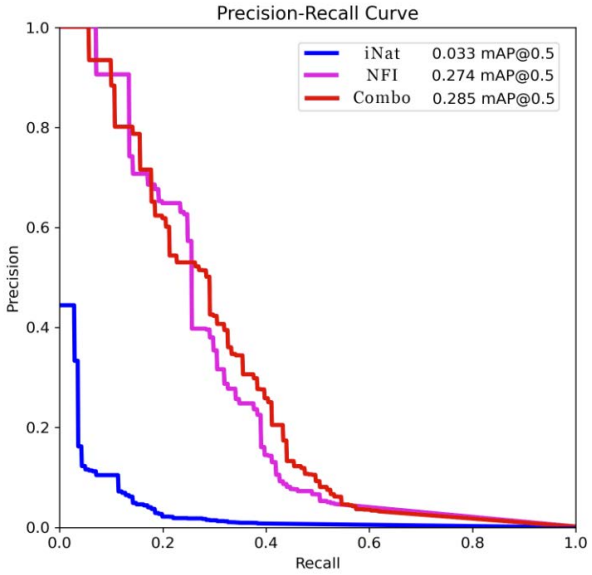
Figur 4.1: Forvirringsmatrise for NFI-modellen, hvor tallene representerer antall masker i testsettet. Matrisen viser hvordan enebærbusker og bakgrunn er klassifisert riktig og feil



Figur 4.2: Forvirringsmatrise for iNat-modellen, hvor tallene representerer antall masker i testsettet. Matrisen viser hvordan enebærbusker og bakgrunn er klassifisert riktig og feil



Figur 4.3: Forvirringsmatrise for combo-modellen, hvor tallene representerer antall masker i testsettet. Matrisen viser hvordan enebærbusker og bakgrunn er klassifisert riktig og feil

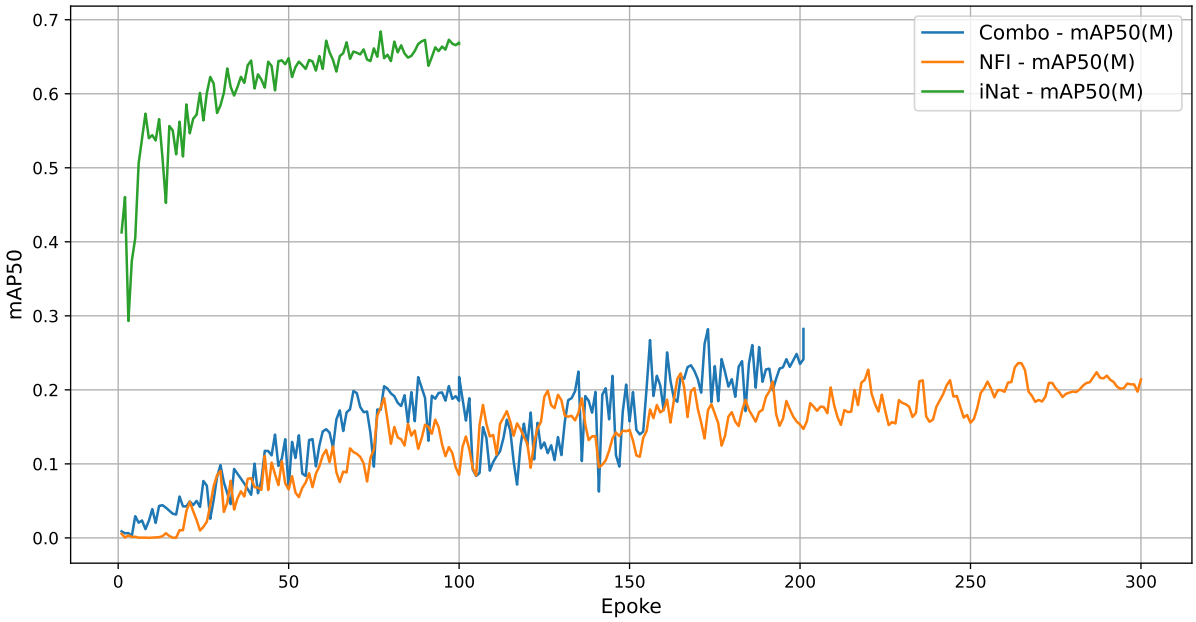


Figur 4.4: Precision-Recall graf for de tre segmenteringsmodellene: iNat, NFI, og Combo. Grafen viser presisjon mot gjennkall, hvor arealet under kurven tilsvarer mAP50 scoren

Trening

Under er en oversikt over hvordan modellene utviklet seg gjennom treningen. iNat-modellen er validert på egne bilder, mens Combo- og NFI-modellene er validert på NIBIO-bilder. Derfor er det avvik fra tidligere presenterte resultater for iNat.

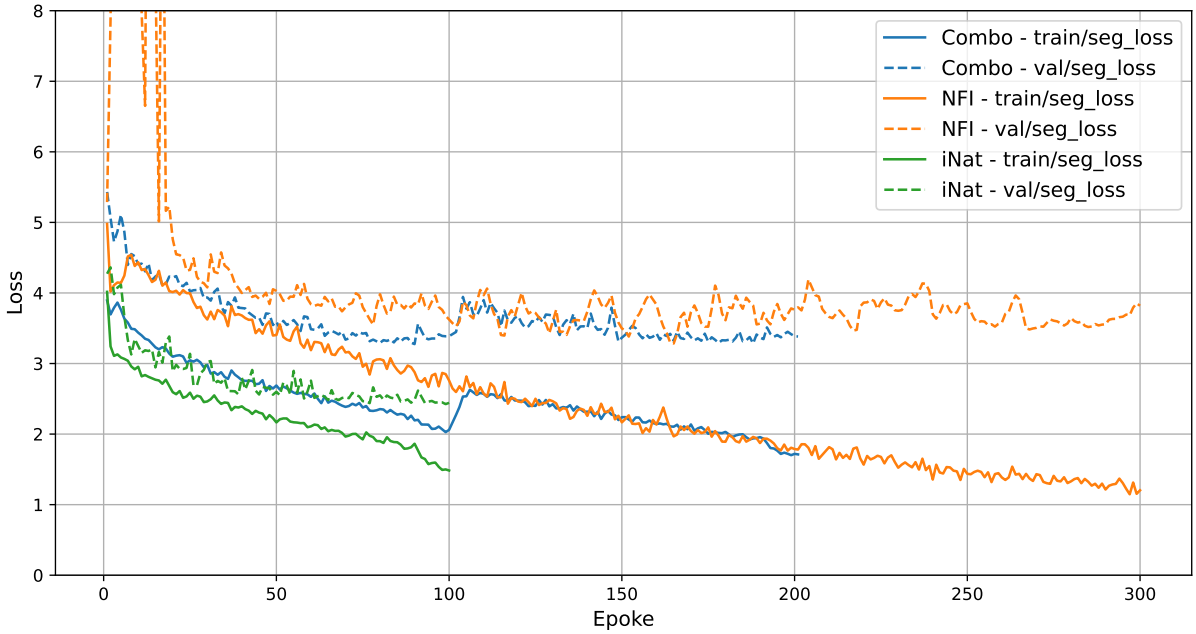
Figur 4.5 viser mAP50-verdien (kapittel: 2.4) som modellene har oppnådd per epoke under treningen. Den viser at iNat-modellen, på kun sine egne bilder, gjør det relativt bra, med en høyeste verdi opp mot 0.7. Mens NFI- og Combo-modellene følger hverandre og gjør det dårligere, med en verdi som beveger seg opp mot 0.3.



Figur 4.5: utviklingen av mAP50 segmentering gjennom treningen av de tre segmenteringsmodellen.Grafen viser mAP50-score for modellene Combo, NFI, og iNat over 300 epoker

Figur 4.6 viser trenings- og valideringsloss for hver av modellene. Verdiene til NFI-modellen har en periode på ca. 10 epoker hvor den viser uteliggere med loss-verdier over 1000. For synlighetens skyld er ikke dette vist i grafen. Videre ser man at alle modellene har gradvis nedadgående grafer for trening, mens valideringslossen går nedover i starten før den flater ut. Igjen ser vi at iNat-modellen gjør det bedre enn de to andre, med lavere loss for både trening og validering. Vi ser også at Combo- og NFI-modellene følger hverandre, spesielt tydelig etter epoke 100 hvor Combo-modellen ser ut til å gjøre et hopp i verdi.

Hoppet kommer trolig av måten grafen er satt sammen på. Grafen er satt sammen av tre deler, hvor den første delen viser treningen fra epoke 0-48. Del to er en fortsettelse av denne modellen, hvor treningen starter fra siste punktet til modell 1. Modell 2 er da trent fra epoke 49-100. Modell 3 er trent i 100 epoker fra den beste modellen som ble trent i modell 2. Den beste modellen ble oppnådd tidligere i treningen enn epoke 100, og dette forårsaker hoppet i verdien.

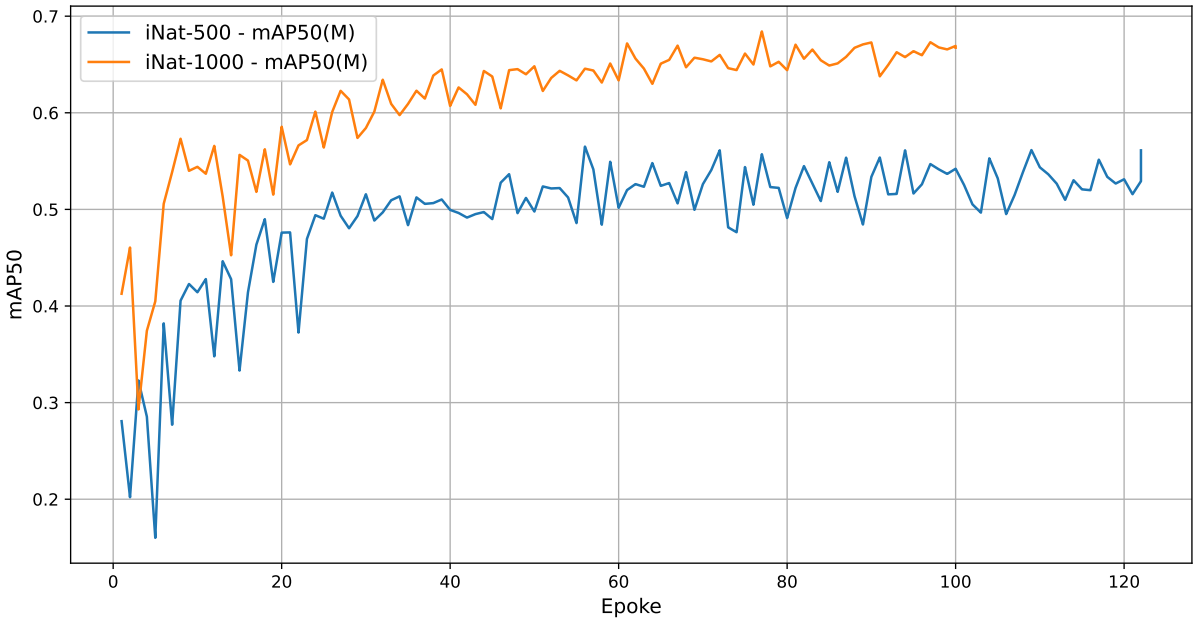


Figur 4.6: Trening og valideringsloss for de tre segmenteringsmodellene gjennom treningsprosessen. Grafen viser trenings- og valideringsloss for modellene Combo, NFI, og iNat over 300 epoker

4.2 iNaturalist data

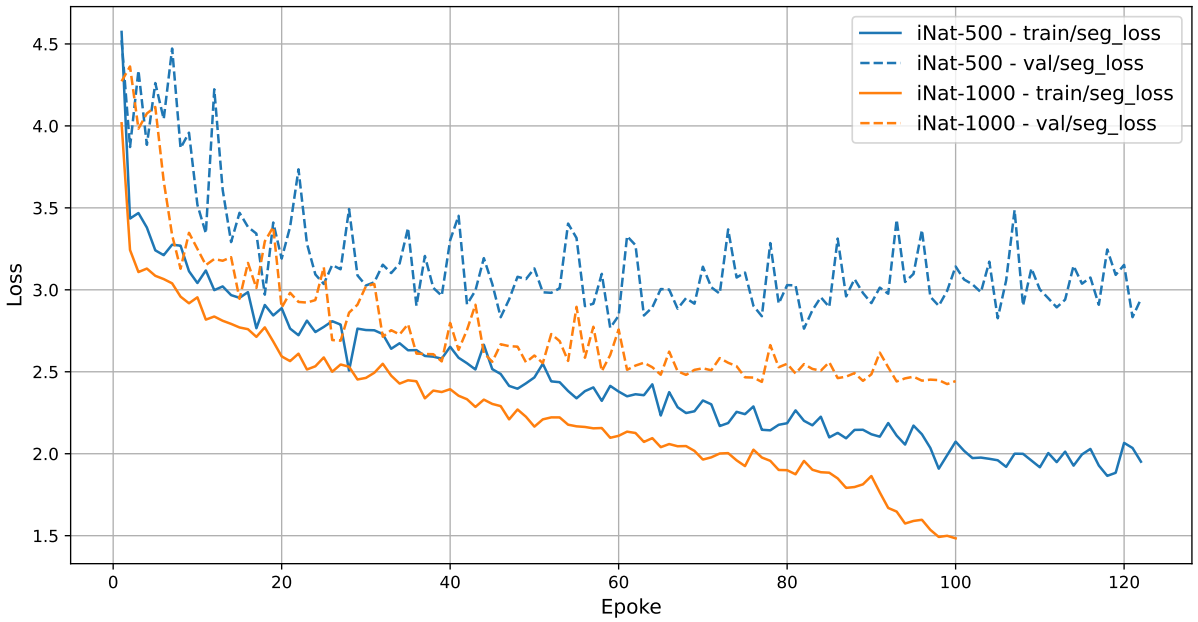
For å undersøke hvor mange bilder som potensielt var nødvendig for å oppnå en god modell, ble det analysert hvor godt modellene presterte med henholdsvis 500 og 1000 tilgjengelige bilder. Grafen er basert på verdier hentet fra validering på kun iNat-bilder.

Figur 4.7 viser mAP50-scoren for hver av modellene. Man ser at begge modellene følger samme mønster med en stigende kurve i starten, før trenden begynner å flate ut og nå et platå. iNat-1000 har den høyeste verdien på rundt 0.65, mens iNat-500 har en topp på 0.55.



Figur 4.7: mAP50 score for iNat modell på 500 og 1000 bilder

Figur 4.8 viser trenings- og valideringsloss for modellene. Her ser vi at valideringslossen til iNat-500-modellen flater ut relativt tidlig, mens treningslossen fortsetter å synke. Vi ser det samme mønsteret i iNat-1000-modellen, men her skjer det senere i treningen, og begge grafene ligger lavere gjennom hele treningsforløpet enn de til iNat-500.



Figur 4.8: Segmenteringsloss for iNat modell tret på 500 og 1000 bilder

4.3 Kvantitativ resultater fra klassifisering

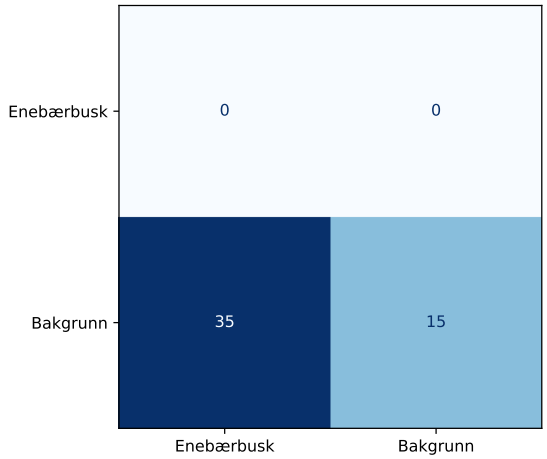
Klassifiseringen ble kun gjennomført på klassifiseringsdatasettet(figur: 3.5. Under vises forvirringsmatriser og valideringsparametrene for den beste klassifiseringsmodellen og samme modell med en "tweak" i vektingen av klassene. Dataen er fra validering på testdatasettet. Klassifiseringen er satt med en terskelverdi på 50% for når et bilde klassifiseres som enebærbusk eller bakgrunn. Tweak er satt til en terskelverdi på 3%, som var verdien som ga høyest score på testdatasettet.

Vi ser fra resultatene at Tweak gir den beste scoren i alle felter bortsett fra recall. Tweak har en nøyaktighet på 0.86, presisjon på 0.87, recall på 0.94 og F1-score på 0.90. Klassifiseringen på den andre siden har henholdsvis verdier på 0.36, 0.09, 1.00 og 0.16. Recall på 1.0 skyldes at det ikke er klassifisert noen enebærbusk i noen av bildene og er derfor missvisende.

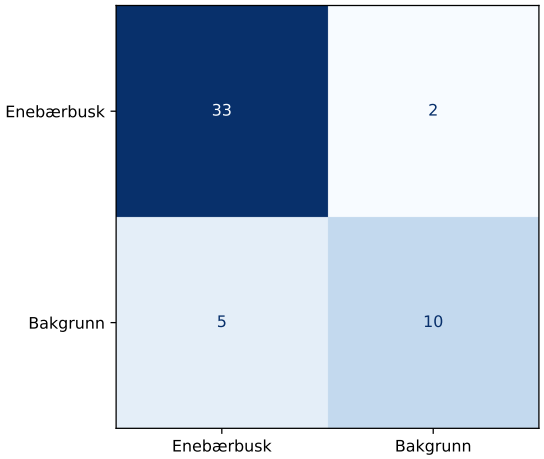
Tabell 4.3: Resultater fra valideringsparameteren for klassifiseringmodellene klasssifisering, og Tweak

Score / Modell	Klassifisering	Tweak
Accuracy	0.36	0.86
Precision	0.09	0.87
Recall	1.00	0.94
F1-score	0.16	0.90

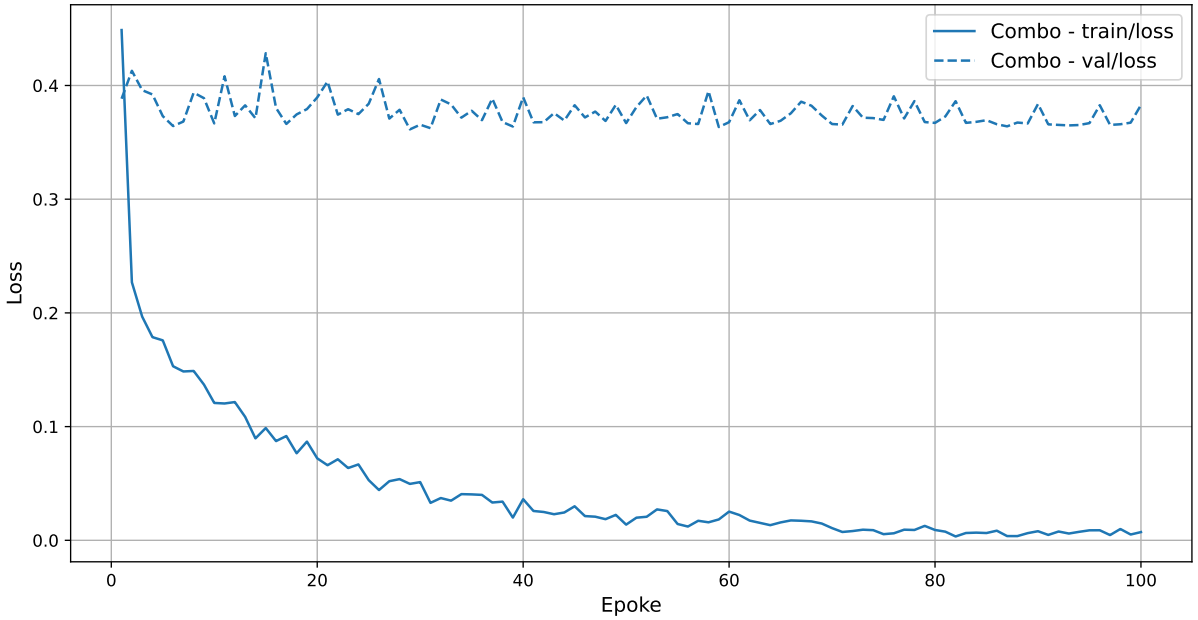
Figur 4.11 viser trenings- og valideringsloss for klassifiseringsmodellen gjennom treningen. Man ser at treningslossen er helt ned til 0, noe som tilsier at den har lært treningsdatasettet nesten helt perfekt. Valideringslossen er imidlertid høyere og har flatet ut uten noen nedadgående bevegelse.



Figur 4.9: Forvirringsmatrise for klassifiserings-modellen, hvor tallene representerer antall masker i testsettet. Matrisen viser hvordan enebærbusker og bakgrunn er klassifisert riktig og feil



Figur 4.10: Forvirringsmatrise for tweak-modellen, hvor tallene representerer antall masker i testsettet. Matrisen viser hvordan enebærbusker og bakgrunn er klassifisert riktig og feil



Figur 4.11: Trening og valideringsloss for klassifiserings-modellen gjennom treningen

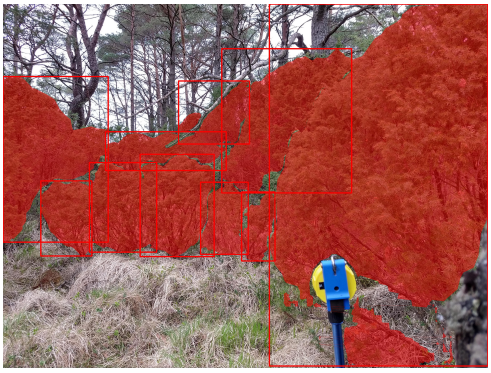
4.4 Kvalitative resultater for segmenteringsmodellene

For å bygge opp under de kvantitative resultatene og få en dypere forståelse av hvordan modellene faktisk predikerer, følger det under noen kvalitative analyser.

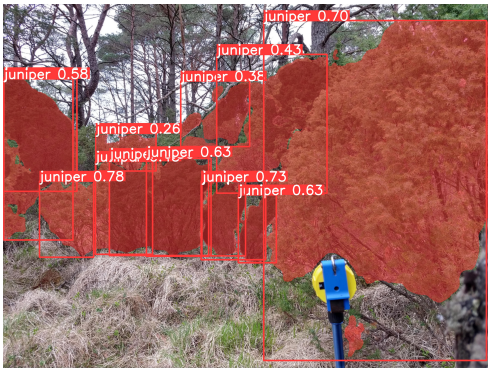
4.4.1 Segmenteringsmaske

Under følger fire eksempler på masker som modellene har predikert. Originalbildet og tilhørende maske er også vist.

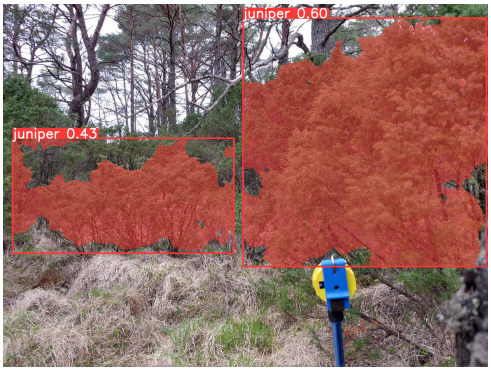
Bildet A18027_N vist i figur: 4.12a er et bilde hvor alle modellene klarer å predikere relativt bra. Det er mye enebærbusk i bildet, men samtlige modeller klarer å skille enebærbuskene fra trærne som er i bakgrunnen. Combo modellen sin prediksjon, vist i figur: 4.12b, og NFI modellen sin prediksjon, vist i figur: 4.12d, predikerer ganske likt og finner flere tilfeller av busken, mens iNat modellen sin prediksjon, vist i figur: 4.12c, foretrekker å segmentere ut større områder og flere busker som ett tilfelle av enebærbusk.



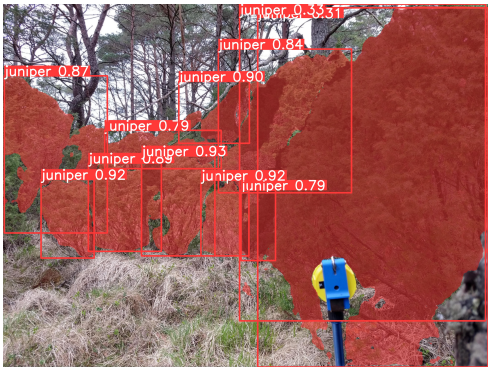
(a) Originalbilde med manuelt annotert maske



(b) Combo-modellen sin predikerte maske



(c) iNat-modellen sin predikerte maske



(d) NFI-modellen sin predikerte maske

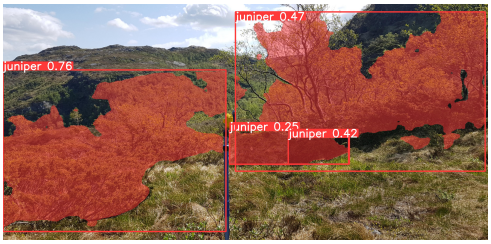
Figur 4.12: Predikert segmenteringsmaske på bildet A18027_N

På bildet A34027_O, vist i figur: 4.13a, varierer det litt hvor bra hver modell klarer seg. Combo modellen, vist i figur: 4.13b, finner to av fire busker og har ingen feilklassifiseringer. iNat modellen, vist i figur: 4.13c, er overivrig og finner områdene hvor det er enebærbusker, men tar med større områder rundt som ikke er en del av busken. Den ser også ut til å slite med å skille mellom enebærbuskene og bjørketrærne rundt. NFI modellen, visst i figur: 4.13d, finner

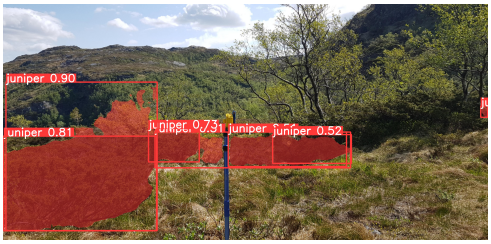
også to av fire busker, men er litt mer ivrig og feilklassifiserer flere områder hvor det ikke er enebærbusker. Felles for alle modellene er at de finner den store busken helt til venstre, som er godt belyst og står for seg selv, men de sliter med busken helt til høyre som ligger mer i skyggen og er omgitt av bjørketrær.

(a) Originalbilde med manuelt annotert maske

(b) Combo-modellen sin predikerte maske



(c) iNat-modellen sin predikerte maske



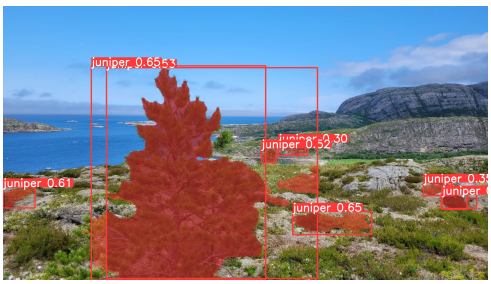
(d) NFI-modellen sin predikerte maske

Figur 4.13: Predikert segmenteringsmaske på bildet A34027_O

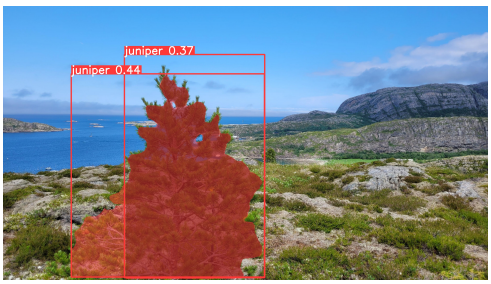
På bildet C59024_N, vist i figur: 4.14a, er det et bra testbilde siden det ikke har noen enebærbusker i seg, men har med et lite bartre som lett kan forveksles med en enebærbusk. Som man ser, klarer Combo modellen, vist i figur: 4.14b, å la være å klassifisere noen enebærbusker i bildet. iNat modellen, vist i figur: 4.14c, blander litt og tolker mange av lyngtuer som enebær og mener også at furutreet i fronten av bildet er en enebærbusk. NFI modellen, vist i figur: 4.14d, predikerer furutreet som en enebærbusk, men klarer å skille lyngtuene fra enebærbusker.

(a) Originalbilde med manuelt annotert maske

(b) Combo-modellen sin predikerte maske



(c) iNat-modellen sin predikerte maske



(d) NFI-modellen sin predikerte maske

Figur 4.14: Predikert segmenteringsmaske på bildet C59024_N

Ingen av modellene gjør det noe bra på predikeringen på bildet A38023_S, vist i figur: 4.15a. iNat modellen, vist i figur: 4.15c, og NFI modlelen, vist i figur: 4.15d, har funnet noen små busker, men som man ser på originalbildet, er det store deler de ikke har klart å detektere. Combo-modellen, som vist i figur: 4.15b, har ikke funnet noe. Enebærbuskene er i bakgrunnen av bildet på litt avstand, noe som tyder på at ingen av modellene klarer å håndtere små busker på lang avstand.

(a) Originalbilde med manuelt annotert maske

(b) Combo-modellen sin predikerte maske

(c) iNat-modellen sin predikerte maske

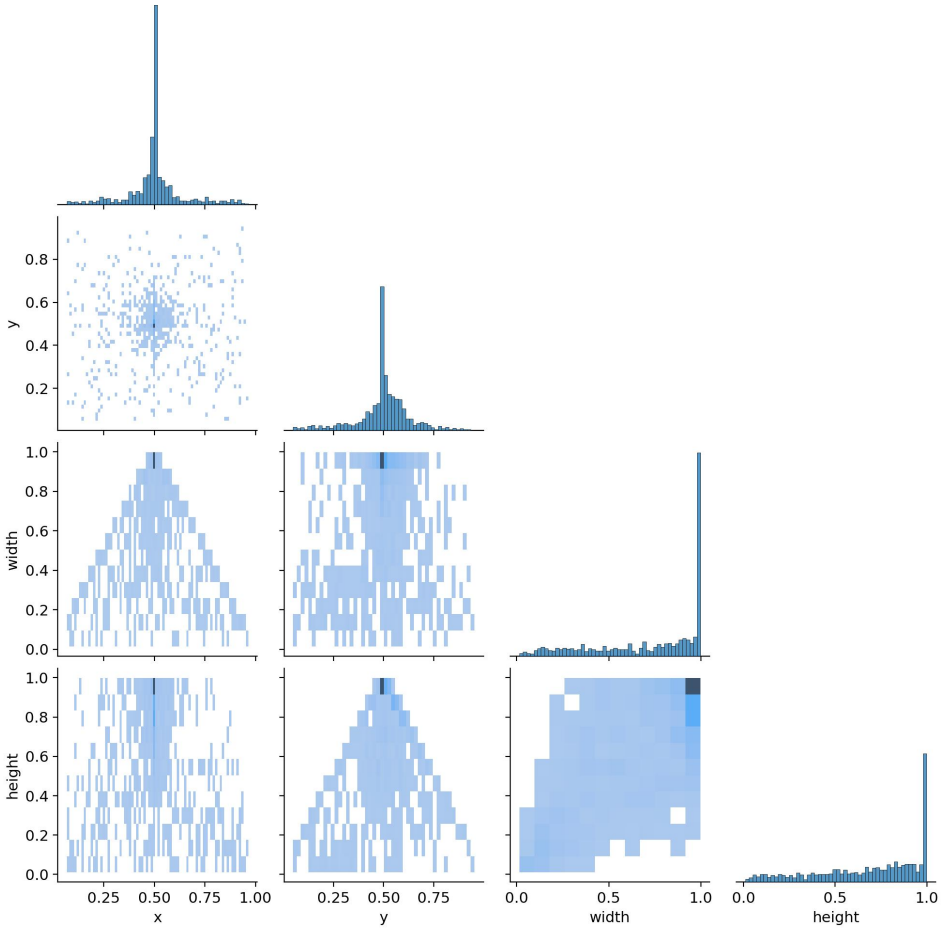
(d) NFI-modellen sin predikerte maske

Figur 4.15: Predikert segmenteringsmaske på bildet A38023_S

4.4.2 Korrelogram

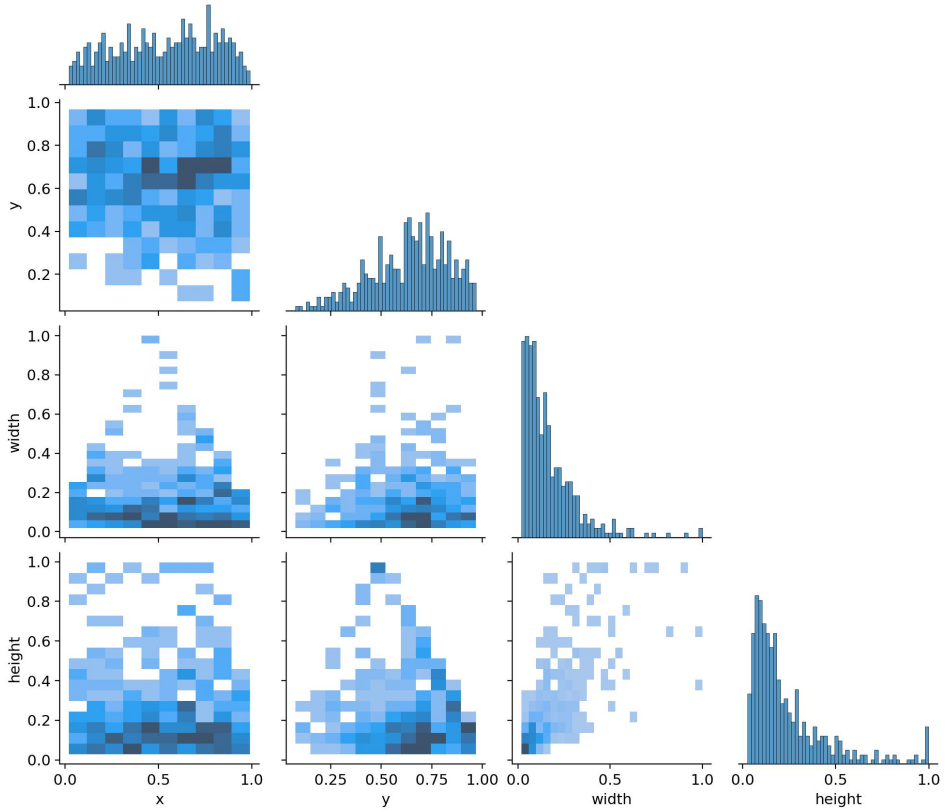
Under følger en analyse av korrelogrammene til hver av modellene som viser fordelingen av hvert treningsdatasett. Et korrelogram er en grafisk fremstilling av korrelasjonskoeffisientene mellom to variabler. Korrelogrammet viser korrelasjonen mellom fire forskjellige variabler: x-koordinatet, y-koordinatet, bredden (width) og høyden (height) til treningsmaskene som er representert og korrelert med hverandre. x- og y-koordinatet er sentrum av avgrensningsboksen til masken. Hvert histogram viser fordelingen innenfor den variabelen. Hvert spredningsdiagram viser fordelingen mellom de to variablene, hvor mørkere farge betyr flere tilfeller med den verdien. Verdiene er normalisert ned til verdier mellom 0 og 1 for å kunne sammenligne variablene.

I korrelogrammet til iNatutralist-datasettet (figur: 4.16) ser man at objektene er spredt over hele bildet i noe som kan minne om en normalfordelingskurve, men med en tydelig vekt midt i bildet. Dette ser man både i histogrammet for fordelingen av x og y, men også i spredningsdiagrammet for korrelasjonen mellom x og y. Ser man på bredden og høyden til objektene, ser man at det er representasjon over hele spekteret, men en skjevfordeling med mange verdier opp mot 1.0 i både bredde og høyde. Ut fra dette kan man si at maskene som modellen er trent på har store avgrensningsbokser som dekker store deler av bildet i både bredde og høyde, og har sentrum plassert midt i bildet.



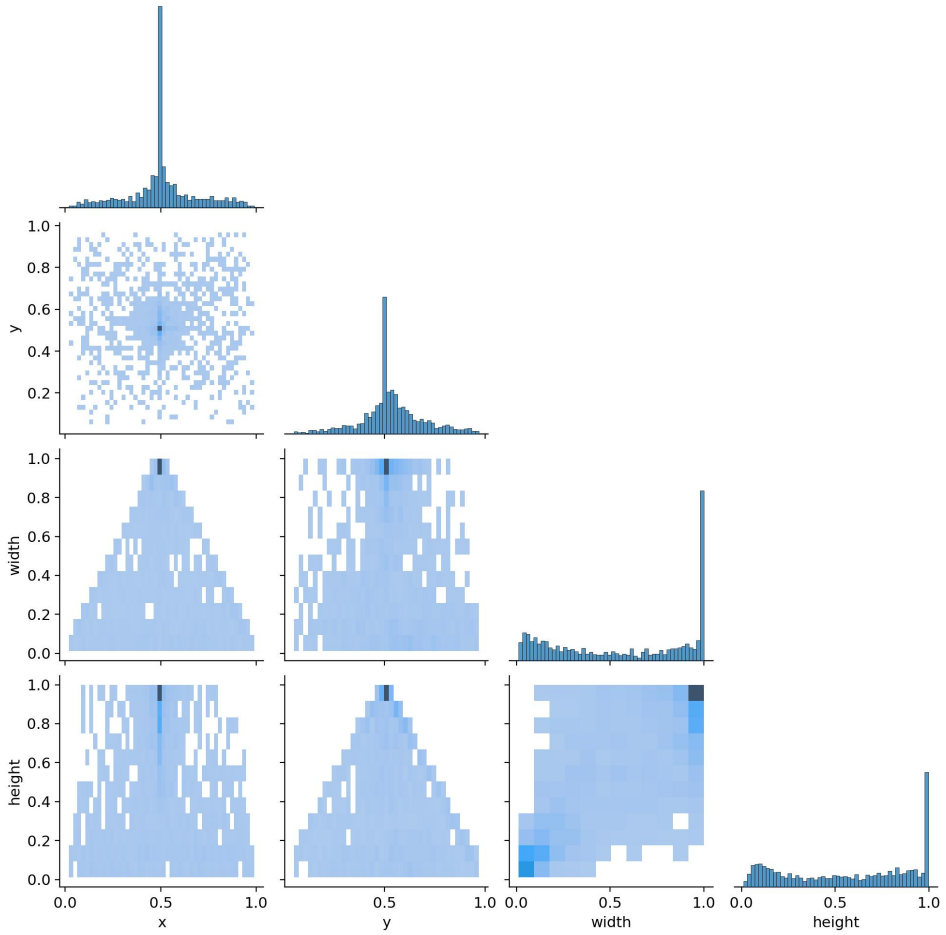
Figur 4.16: Korrelogram for iNaturalist-datasettet, som visualiserer sammenhengen mellom variablene x, y, width og height. Inkluderer scatter plots og histogrammer for hver variabel

Korrelogrammet til NIBIO-datasettet (figur: 4.17) har en mer jevn spredning av hvor i bildet masken er, basert på x- og y-koordinatene. Man ser likevel en liten tendens til flere masker opp mot høyre hjørne, med nesten ingen tilfeller av masker nede i venstre hjørne. Når man ser på bredde og høyde, ser man at det er en skjevfordeling hvor vekten ligger på lave verdier. Dette betyr at mange av avgrensningsboksene i dette datasettet er små i utstrekning. Man kan derfor si at tilfellene av enebærbuskene i NIBIO-datasettet er små og jevnt fordelt utover i bildene.



Figur 4.17: Korrelogram for NIBIO-datasettet, som visualiserer sammenhengen mellom variablene x, y, width og height. Inkluderer scatter plots og histogrammer for hver variabel

Ser man på korrelogrammet til Combo-datasettet (figur: 4.18), ser man at det er en kombinasjon av begge datasettene. Man ender derfor opp med en litt bedre og mer jevn spredning i x- og y-koordinatene. Man får også en viss balanse i bredde og høyde, med de store avgrensningsboksene fra iNaturalist-datasettet og de mindre avgrensningsboksene fra NIBIO-datasettet. Men på grunn av skjevfordelingen i antallet bilder i hvert av datasettene, ender man hovedsakelig opp med mange av de samme karakteristikkene som iNaturalist, altså at modellen har en stor tyngde av masker midt i bildet med stor utstrekning i både høyde og bredde.



Figur 4.18: Korrelogram for combo-datasettet, som visualiserer sammenhengen mellom variablene x, y, width og height. Inkluderer scatter plots og histogrammer for hver variabel

5 Diskusjon

Dette kapittelet tar for seg metoden og resultatene den førte til, og diskuterer dem. Del én tar for seg resultatene, del to tar for seg betydningen av eksterne datasett, del tre tar for seg metodens påvirkning, og del fire tar for seg forslag til videre arbeid.

5.1 Analyse av resultatene

5.1.1 Segmenteringsmodellene

Tabell 4.1 viser at ingen av modellene presterer på ønsket nivå. Man ser en tydelig rangering blant de tre modellene. Den rene NFI-modellen har best nøyaktighet (accuracy), presisjon (precision) og F1-score, noe som gjør den til den beste modellen av de tre. På andre plass kommer Combo-modellen, som gjør det best på gjenkall (recall), mAP50 og mAP50-95, samt har nest høyeste score på nøyaktighet, presisjon og F1-score. Til slutt kommer iNat-modellen, som leverer dårligst på alle valideringsparametere med klar margin.

NFI-modellen gjør det best fordi testdataen består av den samme typen bilder som den er trent på. Den rene iNat-modellen er trent på en annen type bilder, og derfor er NIBIO-bildene som brukes i testdataen helt nye for modellen. Combo-modellen burde ha vært nærmere i verdiene til valideringsparametrene til NFI-modellen, men trolig har den plukket opp noen strukturer fra iNat-bildene i treningen som gjør at nøyaktigheten er dårligere. På den andre siden ser det ut som om den ekstra dataen bidrar til bedre recall, mAP50 og mAP50-95 score

5.1.2 Robusthet og pålitelighet

Datagrunnlaget

Hovedgrunnen til de dårlige prestasjonene stammer nok fra mangel på data. NIBIO-dataen inneholder kun 171 bilder med enebærbusk. Dette er for lite til å trene opp en god modell. Det fører også til en usikkerhet til valideringsparametrene, da testsettet inneholder 50 bilder, dette er et relativt lite utvalg fra hele populasjonen og derfor ikke kan ses på som et stort nok utvalg til

å si noe om hvor bra modellen generaliserer seg til ny data og hvordan den ville fungere fullt integrert hos NIBIO.

En mulig usikkerhet som kan påvirke modellens prestasjon er tilknytningen til datasettene, spesielt i iNat-datasettet. Her er det knyttet noe usikkerhet til annoteringen på grunn av menneskelig feil. Siden annoteringen ikke ble utført av en fagkyndig person på hva som er enebærbusk og ikke. Det ble oppdaget tilfeller hvor noen av objektene som var oppgitt som enebærbusk viste seg å være andre arter av vegetasjon, som bartrær og lyng. Videre ble det også oppdaget i bildene hentet fra iNaturalist at de var feilkategorisert som enebærbusk når det egentlig var andre arter. De tilfellene som ble oppdaget, ble fjernet enten før eller under annoteringen, men det kan være tilfeller som ikke har blitt oppdaget. Dette kan bidra til at modellen gjør det dårlig på å skille enebærbusker fra for eksempel bartrær. Dette er trolig ikke hovedgrunnen til modellens dårlige prestasjon, men kan være en faktor hvor dataen har bidratt til feillæring.

Valideringsparametrene er regnet ut basert på tallene man får fra forvirringsmatrisen. Siden segmenteringsmodellen kun inneholder én klasse, ser den ikke på alle de gangene modellen lar være å predikere enebærbusk. Et slikt tilfelle skulle ha blitt registrert som en sann negativ (TN). Forvirringsmatrisen ser kun på de tilgjengelige tilfellene av enebærbusk som forekommer i alle 50 bildene. Det er derfor noe usikkert om resultatene gir et fullstendig bilde, siden det ikke er noen tilfeller av sanne negativer. Nullen som er registrert i TN-ruten i matrisen indikerer at testdataen ikke inneholder noen tilfeller av en predikert og faktisk bakgrunnklasse. Et eksempel på dette er bildet i figur 4.14. Siden bildet ikke inneholder noen tilfeller av enebærbusk, registreres dette som falske positiver hos iNat og NFI, mens det ville blitt registrert som en TN hos Combo hvis det var inkludert. Dette ville ført til en økning i nøyaktighetsscoren til modellen. Det er derfor ikke en selvfølge at den modellen med best valideringsparametere er den beste modellen. Det er derfor man har innført parameterne F1 og mAP50, da de gir et mer helhetlig bilde. Men igjen er heller ikke disse parameterne perfekte. F1-scoren er utsatt for samme usikkerhet som nøyaktighet. mAP50 har et problem hvor den ikke tar hensyn til hvor nærme modellens prediksjon er den faktiske masken. Dette er fordi den baserer seg på IoU. Alle avgrensningsbokser som er utenfor vil automatisk få en IoU-score på 0[20]. mAP50 gir et godt bilde på hvor nøyaktige prediksjonene til modellen er, men siden NFI og Combo-modellen har ganske like mAP50-scoringer, kan det likevel være forskjeller i modellene om hvor nærme maskene er, men dette måles altså ikke.

Visuell kontroll

Ut fra de kvalitative resultatene vist i figur 4.15 ser man at alle modellene sliter med å finne tilfeller av enebærbusker som er langt unna eller i bakgrunnen av bildet. Når buskene tar opp en liten del av bildet, blir det vanskeligere å oppdage for modellen på grunn av nedskaleringen av oppløsningen i forbindelse med predikeringen.

I figur 4.13 ser man at alle modellene sliter med busker i skyggen. Dette er trolig fordi det er vanskelig å fange opp strukturen i busken når det er skygge. Dette kunne nok blitt bedre ved hjelp av mer augmentasjon. Videre ser man at spesielt iNat-modellen sliter med å skille enebærbusk fra resten av vegetasjonen i bildet, spesielt bjørketrærne. Dette er nok fordi strukturen mellom en enebærbusk på nært hold kan ligne på strukturen man får i bjørketrær med lite løv. Noe av det samme problemet ser man i bildet, hvor den også feilsegmenterer annen vegetasjon som enebærbusk.

I figur 4.14 ser man et eksempel på et av bildene som ikke har noen enebærbusker i seg. Her predikerer både iNat og NFI at det er enebærbusk i bildet. Begge modellene tror at furutreet i fronten av bildet er en enebærbusk. I tillegg feilklassifiserer iNat-modellen lyngtuer som enebærbusk. Imidlertid klarer Combo-modellen å la være å feilklassifisere lyng og furutreet som enebærbusk. Dette er trolig fordi Combo-modellen er trent på flere bilder som kun er betegnet som bakgrunner, altså bilder som ikke har noen maske i seg. iNat og NFI-modellene har kun et fåtall av slike bilder, og tror derfor at det skal være enebærbusk i hvert bilde. Combo-modellen er trent opp med mesteparten av bildene som bakgrunner, noe som kan være årsaken til at modellen ikke segmenterer ut feil i dette tilfellet.

I figur 4.12 ser vi at alle modellene predikerer relativt bra. Man ser imidlertid tendensen som ligger i dataen hver modell er trent opp på. iNat-modellen foretrekker å segmentere ut store deler av bildet og tar med flere busker i samme segment, mens NFI og Combo-modellen foretrekker flere mindre masker til å dekke hele området.

Overtilpassing

Ut fra grafen vist i figur 4.6 ser man at både Combo- og NFI-modellen viser tegn på overtilpassing (overfitting). Dette fordi treningslossen i begge tilfeller fortsetter å ha en nedadgående trend, mens valideringslossen flater ut. Det er også tegn på underrepresentasjon i datasettet, da

avstanden mellom kurvene er stor[33]. iNat har en bedre trend hvor begge grafene er nære hverandre og synker. Mot slutten ser det ut til at modellen begynner å vise tegn til overfitting siden det ser ut som valideringsgrafen begynner å flate ut. Selv om alle treningsgrafene fortsetter å synke, noe som kan indikere at modellen fortsatt lærer og derfor kunne trengt flere epoker, viser mAP50-grafene i figur 4.5 at de flater ut, noe som understreker at det er snakk om overtilpassing på samtlige modeller.

5.1.3 Klassifiseringsmodellene

Klassifiseringsmodellen gir heller ikke håp for en god modell, da den oppnår en nøyaktighet på 0.36 og en F1-score på 0.16. Men ved å endre terskelen i modellen for hvor sikker den må være på at bildet inneholder enebærbusk eller ikke, altså sikkerheten (confidence), kan man oppnå svært gode resultater. Ved å sette denne sikkerhetsverdien til 3% oppnår man en nøyaktighet på 0.86 og en F1-score på 0.90. Dette er såpass gode resultater at man kan begynne å vurdere å implementere modellen i videre arbeid hos NIBIO.

Det er likevel usikkerhet knyttet til funnet om at en modifisert klassifiseringsmodell presterer godt. Metoden for å komme frem til det beste sikkerhetsnivået er gjort ved å gå gjennom alle verdiene og regne ut valideringsparametrene. Den beste modellen er derfor plukket ut ved å finne den sikkerhetsverdien som ga best resultat. Dette er et eksempel på overtilpassing av dataen, da dette ikke gir noen indikasjon på om dette sikkerhetsnivået er den beste verdien globalt. Hadde det vært flere bilder i testdataen, ville det gitt et mer sikkert resultat.

Også klassifiseringsmodellen viser tegn til overtilpassing, som vist i grafen 4.11. Avstanden mellom trenings- og valideringsgrafen tyder på overtilpassing. Det at treningslossen også nærmer seg 0, betyr at modellen har lært seg treningssettet nesten helt perfekt.

5.2 Betydningen av ekstern data

Generelt vil man, når man trener opp maskinlæringsmodeller, gjerne ha så mye data som mulig slik at modellen har et godt utgangspunkt for å lære, samtidig som man har nok data til testing. Mange som utvikler en bildegjenkjenningsmodell ønsker at modellen skal gjøre det bra på en spesifikk oppgave eller med en spesifikk type bilde. Av den grunn er det ikke alltid man har et stort nok datasett selv. Dette problemet kan løses ved bruk av eksterne datasett. Felles for

mange av disse eksterne datasettene er at de er et produkt av en fellesdugnad fra brukere. Tilgjengeligheten av disse datasettene er derfor uvurderlig i mange tilfeller, da man kan supplere sitt eget datasett med ofte ferdig klassifiserte/segmenterte bilder.

Dette blir også en viktig faktor når man bruker ferdigtrente nettverk, slik som i denne oppgaven hvor YOLO-modellen er trent på COCO-datasettet. Vekten fra den trente modellen er overført til den modellen som vi ønsker skal finne enebærbusk. COCO-datasettet er et av disse kjente eksterne datasettene som brukes mye til å evaluere prestasjonen til forskjellige arkitekturer, slik at man har et sammenligningsgrunnlag. COCO-datasettet består av rundt 80 klasser og 330 000 bilder[34]. Det ferdigtrente nettverket er derfor kjent med å segmentere ut forskjellige hverdagslige objekter. Ingen av klassene er planter/vegetasjon. YOLO-modellen er derfor ferdigtrent på å skille objekter fra hverandre, men ikke biologiskearter fra hverandre. Det er trolig tilstrekkelig likhet mellom oppgavene til at modellen har fordel av denne treningen, men det kan også være at bildene og objektene er såpass annerledes at modellen ikke har noen fordelaktig forbedring fra denne treningen. Men i mangel på andre forhåndstrente modeller er det ikke mulig å sjekke dette. Det er utviklet et eksternt datasett for fjernmåling som kan brukes som en erstatning for COCO, hvor de kombinerer generell objektkunnskap fra ImageNet og tilfører fjernmålingsspesifikk kunnskap med egne bilder[35]. Muligheten for at dette også er et behov i vegetasjonskartlegging eller for terrestriske landskapsbilder er derfor stor og vil trolig bidra til raskere læring av slike modeller i fremtiden.

5.2.1 Betydning av iNat-datasettet

Ser man på betydningen av iNat-datasettet, viser valideringsparametrene at det ikke gir en bedre modell med tanke på nøyaktighet, F1-score og presisjon. Den blir faktisk verre. Men man kan se at inkluderingen av iNat-datasettet forbedrer gjenkall (recall) og mAP50-scoren. Bedre gjenkall betyr at Combo-modellen er mer sikker på å finne alle tilfellene av enebærbusk i bildet. Marginalt høyere mAP50 tilsier at Combo klarer å segmentere enebærbuskene mer nøyaktig enn de andre modellene. Den klarer å treffe med mer enn 50% overlappende predikert mot faktisk maske oftere enn de andre modellene. En høyere mAP50-95-score enn NFI-modellen tyder på at modellen har en høyere IoU og er mer sikker på sine prediksjoner. På grunn av at resten av parameterne er lavere, men man får høyere score på disse tre parameterne, er det logisk å anta at det er supplementeringen av NFI-dataen med data fra iNat som er årsaken til dette.

Siden den rene NFI-modellen er trent opp på kun et fåtall bilder uten enebærbusk, og Combo er trent opp på alle de tilgjengelige NIBIO-bildene, er det vanskelig å konkludere om forbedringen av valideringsparametrene kommer fra det eksterne datasettet eller om det er inkluderingen av tomme NIBIO-bilder som bidrar mest til høyere mAP-score.

Det iNat-modellen viser i denne oppgaven er at siden den rene iNat-modellen oppnår akseptable verdier på mAP50 ved validering med egne bilder (figur: 4.7), tyder det på at problemstillingen om å kunne lage en segmenteringsmodell som segmenterer ut enebærbusk er mulig å få til med tilstrekkelig mengde bilder.

5.2.2 Korrelogram

En forklaring på prestasjonen kommer frem når vi ser på korrelogrammene til hvert av datasettene. Ut i fra korrelogrammet til iNat (figur:4.16) ser vi at maskene modellen er trent på er store i utstrekning og plassert midt i bildet. Ser man på korrelogrammet til NFI (figur:4.17) ser man at maskene er mer spredt i bildet og opptar mye mindre plass. Siden det er NIBIO-bildene man ønsker at modellen skal prestere bra på, gir det mening at iNat-modellen gjør det dårlig, siden modellen er trent opp på en helt annen type fordeling av objekter enn det man ønsker at den skal finne.

Ser man på korrelogrammet til Combo-modellen (figur: 4.18) ser man at man får en bedre spredning i x og y, og man ser også at man får en liten motvekt i bredde og høyde. Combo-datasettet ender opp med å holde på mange av karakteristikkene til iNat-datasettet, siden det er en skjevfordeling i størrelsen på datasettene. Hadde datasettene vært like store, med lik karakteristiske fordeling som de har nå, ville det resulterende korrelogrammet vært mer jevnt fordelt, med likevekt på begge sider av skalaen. Dette ville ført til et robust datasett som trolig ville kunne dekke de fleste tilfeller av enebærbusk, uavhengig av hvilken type utsnitt bildet hadde, det være seg nærbilde som i iNat eller landskapsbilder som i NIBIO bildene

5.3 Metodens betydning

Metoden kunne vært mer strømlinjeformet og standardisert. Implementeringen av en pipeline ville gjort det enklere å holde styr på hva man hadde gjort og sørget for et bedre sammenligningsgrunnlag på alle modellene. Slik metoden er utført nå, ble mye lært underveis. Først ble

den beste parameteren for iNat-modellene funnet, deretter ble den beste NFI-modellen laget, og til slutt den beste Combo-modellen. En del av parameterne som ble funnet å ikke ha noen effekt i iNat-modellen, ble derfor ikke prøvd ut på NFI og Combo modellene, under antakelsen om at de ikke ville ha noen effekt der heller. Siden datasettene er forskjellige, kan det være at de samme parameterne ville hatt en effekt på de senere modellene. På samme måte ble det prøvd ut parametere i de senere modellene som muligens kunne ha gitt bedre resultater på de tidligere modellene. Ved å implementere en pipeline og sette opp en grid search av disse parameterne, kunne man ha funnet den beste modellen for hvert datasett og forutsatt lik trening og testing.

Selv om Ultralytics sine YOLO-modeller regnes som 'state-of-the-art' innen segmentering og objektdeteksjon, kan det være at valg av modellarkitektur burde vært annerledes, og andre metoder og dybder burde vært prøvd ut. Selv om YOLO tilbyr flere størrelser på sine modeller (tabell:3.2 og 3.3), endrer det ikke dybden på nettverket, kun antall trenbare parametere som vist i modellarkitekturen i figurC.1. Siden modellen viser tegn til overtilpassing, kunne man ved å lage arkitekturen fra bunnen av, styrt dybden på nettverket og sjekket om det hadde hjulpet. Spesielt når det å legge til dropout i treningen ikke viste seg å ha en signifikant effekt. På den andre siden ville det å lage et eget nett fra bunnen tatt mer tid og gått på bekostning av andre funn i oppgaven.

Som nevnt i kapittel 5.2.1, er det ikke mulig å skille mellom om forbedringen i mAP-scoren kommer fra økt antall NIBIO-bakgrunnsbilder eller inkluderingen av iNat enebærbusker. Trolig er det litt av begge. For å finne ut av dette ville det vært bedre å trene en modell på alle NIBIO-bildene. Dette ville trolig også resultert i at NFI-modellen scoret høyere på sine parametere, og det ville gitt et bedre bilde av om Combo-modellen faktisk ble forbedret på grunn av det eksterne datasettet.

5.4 Videre arbeid

Som anbefaling til videre arbeid ville det være nødvendig å skaffe flere bilder med enebærbusk fra NIBIO-datasettetet. Med flere bilder ville det være mulig å teste ut disse modellene mer i detalj. Første steg vil være å teste hvor bra Tweak-klassifiseringsmodellen gjør det på flere bilder, og gjøre justeringer for å finne en sikkerhetsverdi som gir gode resultater på et mer representativt utvalg bilder.

Hvis en segmenteringsmodell er å foretrekke over en klassifiseringsmodell, vil det være nødvendig å gjenta metoden i denne oppgaven. Trene opp en ren NFI-modell på like mange bilder som iNat og en kombinasjon av disse for å se om resultatene forbedrer seg. Deretter vil det være viktig å finne ut hvor mange NIBIO-bilder som er nødvendig før tilførselen av iNat-bilder blir lønnsom eller ikke har noen effekt.

Det bør prøves ut flere modellarkitekturer for å se om andre arkitekturer presterer bedre. Siden modellen viser tegn til overtilpassing, kan det være et tegn på at modellarkitekturen ikke er optimal for denne typen problem. Å eksperimentere med andre arkitekturer kunne vært lurt for å se hvordan det påvirker prestasjonen. Som en start ville et vanlig CNN med variert dybde være en god begynnelse. Det kunne også vært en idé å prøve ut andre forhåndstrente modeller med mer domene-spesifikk trening for å se om det har noen effekt.

Videre arbeid med å besvare spørsmål tre i problemstillingen, hvor stor andel av bildet som er enebærbusk, er også noe man burde se på. Dette er i teorien bare en etterprosessering og bør ikke være vanskelig å få til. Men det er et mål som kan være nyttig å ha hvis man fortsetter å ta bilder fra samme punkt hver gang, og kan derfor brukes til å kartlegge tilvekst av enebærbusk.

6 Konklusjon

Oppgaven har tatt for seg problemstillinger relatert til å automatisk finne enebærbusk i bilder tatt i forbindelse med NIBIO sitt NFI-prosjekt. Videre ble det undersøkt om en modell ville oppleve forbedring hvis den ble supplert med et eksternt open-source datasett. Oppgaven har fokusert på modellenes evne til å besvare spørsmål én og to i problemstillingen, samt betydningen av eksterne datasett. Spørsmål tre har ikke vært fokusert på i denne oppgaven. Originalspørsmålene var som følger:

1) Er det enebærbusker i bildet?

2) Hvor i bildet er enebærbusken?

3) Hvor stor andel av bildet er enebærbusk?

Resultatene viser at instanssegmenteringsmodellen presterer dårlig på alle datasettene. Den modellen med best nøyaktighet er den rene NFI-modellen. Men hvis man vektlegger evnen til å finne alle tilfeller og nøyaktigheten på hver maske, er Combo-modellen den beste. Likevel er det klassifiseringsmodellen med en tweak som gjør det aller best og som er den eneste modellen som faktisk kunne blitt implementert med en gang. Det er imidlertid håp om at segmenteringsmodellene kan fungere, men det er avhengig av mer data for å slå dette fast. Data fra iNat-modellen viser at med nok data er det mulig å oppnå en segmenteringsmodell som presterer bra. En segmenteringsmodell vil klare å svare på alle spørsmålene stilt i problemstillingen og vil derfor være modellen å foretrekke fremfor en klassifiseringsmodell som kun kan svare på spørsmål én.

Når det kommer til spørsmålet om en modell vil ha fordel av å trenes opp med et eksternt datasett, viser resultatene at modellens nøyaktighet ikke går opp, men dens evne til å mer nøyaktig predikere masken går opp. Det tyder på at en modell trent på et eksternt datasett er en mer robust modell og mer sikker i prediksjonene sine enn den rene NFI-modellen.

Bibliografi

[1] Artsdatabanken. Grunnleggende om natur i norge. nettside, 14.11 2023. URL https://artsdatabanken.no/Pages/345448/Grunnleggende_om_Natur_i_Norge. hentet: 11.04.2024.

[2] NIBIO. Landsskog-takseringen. Nettside, 2024. URL https://www.nibio.no/tema/skog/skog-og-miljoinformasjon-fra-landsskogtakseringen/landsskog-takseringen. hentet:08.04.2024.

[3] Knut Ole Viken. *Landsskogtakseringens feltinstruks — 2023*, volume 9(5)2023. NIBIO, 2023.

[4] Milan Chytrý, Joop H. J. Schaminée, and Angelika Schwabe. Vegetation survey: a new focus for applied vegetation science. *Applied Vegetation Science*, 14(4):435–439, 01.11 2011. https://doi.org/10.1111/j.1654-109X.2011.01154.x.

[5] Artsdatabanken. Historien om nin. nettside, 13.11 2023. URL https://artsdatabanken.no/Pages/345454/Historien_om_NiN. hentet: 07.05.2024.

[6] Artsdatabanken. Dette er nytt i nin 3.0. nettside, 01.03 2024. URL https://artsdatabanken.no/Pages/352853/Dette_er_nytt_i_NiN. Hentet: 07.05.2024.

[7] NASA. Landsat 9 fact sheet. Faktaark, 2019. URL https://landsat.gsfc.nasa.gov/satellites/landsat-9/. Hentet: 07.05.2024.

[8] European Space Agency. Copernicus programme. Nettside, 2024. URL https://sentiwiki.copernicus.eu/web/copernicus-programme. Hentet: 07.05.2024.

[9] Jinru Xue and Baofeng Su. Significant remote sensing vegetation indices: A review of developments and applications. *Journal of Sensors*, 2017:1–17, 23.05 2017. https://doi.org/10.1155/2017/1353691.

[10] Thomas Luhmann, Stuart Robson, Stephen Kyle, and Jan Boehm, editors. *Close-Range Photogrammetry and 3D Imaging*. De Gruyter, 2 edition, 2014.

[11] K. Džubáková, P. Molnar, K. Schindler, and M. Trizna. Monitoring of riparian vegetation response to flood disturbances using terrestrial photography. *Hydrology and Earth System Sciences*, 19(1):195–208, 13.01 2015. https://doi.org/10.5194/hess-19-195-2015.

[12] Yichun Xie, Zongyao Sha, and Mei Yu. Remote sensing imagery in vegetation mapping: a review. *Journal of Plant Ecology*, 1(1):9–23, 01.03 2008. https://doi.org/10.1093/jpe/rtm005.

[13] Leiyu Chen, Shaobo Li, Qiang Bai, Jing Yang, Sanlong Jiang, and Yanming Miao. Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*, 13(22), 21.11 2021. https://doi.org/10.3390/rs13224712.

[14] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*, 9(3):171–189, 03.07 2020. https://doi.org/10.1007/s13735-020-00195-x.

[15] Ross Girshick. Fast r-cnn, 27.09 2015. https://doi.org/10.48550/arXiv.1504.08083.

[16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 09.05 2016. https://doi.org/10.48550/arXiv.1506.02640.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 27.01 2018. https://doi.org/10.48550/arXiv.1703.06870.

[18] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning, 23.06 2020. https://doi.org/10.48550/arXiv.1911.02685.

[19] Sebastian Raschka and Vahid Mirjalili. *Python Machine Learning - Third Edition - Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd., 2019.

[20] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression, 15.04 2019. https://doi.org/10.48550/arXiv.1902.09630.

[21] Shivy Yohanandan. What is mean average precision (map) and how does it work. Nettside, 05,06 2020. URL https://xailient.com/blog/what-is-mean-average-precision-and-how-does-it-work/. Hentet: 07.05.2024.

[22] iNaturalist. About. Nettside, 11.07 2023. URL https://www.inaturalist.org/pages/about. hentet: 08.04.2024.

[23] Python. About, 2024. URL https://www.python.org/about/. Hentet: 09.05.2024.

[24] Google colab. Få mest mulig ut av colab-abonnementet. Nettside, 2024. URL https://colab.research.google.com/#scrollTo=qB3bdLe8jkAa. hentet: 08.04.2024.

[25] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8. Software available from GitHub repository, 2023. URL https://github.com/ultralytics/ultralytics. Version 8.0.0.

[26] TrainYOLO. The data engine for yolo. Nettside, 2024. URL https://www.trainyolo.com/. Hentet:08.04.2024.

[27] COMET. About us. Nettside, 2024. URL https://www.comet.com/site/about-us/. hentet: 08.04.2024.

[28] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 05.04 2023. https://doi.org/10.48550/arXiv.2304.02643.

[29] Glenn Jocher, Ayush Chaurasia, Jing Qiu, Muhammad Rizwan Munawar, and Fatih Cagatay Akyon. Image classification. Nettside, 17.04 2024. URL https://docs.ultralytics.com/tasks/classify/. Hentet: 19.04.2024.

[30] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Instance segmentation. Nettside, 18.04 2024. URL https://docs.ultralytics.com/tasks/segment/. Hentet: 19.04.2024.

[31] Glenn Jocher, Burhan Qaddoumi, and Muhammad Rizwan Munawar. Model prediction with ultralytics yolo. Nettside, 18.04 2024. URL https://docs.ultralytics.com/modes/val/. Hentet: 23.04.2024.

[32] Burhan Qaddoumi, Dzmitry Plashchynski, Glenn Jocher, Jason Sohn, Ayush Chaurasia, and Jing Qiu. Model prediction with ultralytics yolo. Nettside, 18.04 2024. URL https://docs.ultralytics.com/modes/train/. Hentet: 19.04.2024.

[33] Jason Brownlee. How to use learning curves to diagnose machine learning model performance. Nettside, 06.10 2019. URL https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/. Hentet: 03.05.2024.

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 21.02 2015. https://doi.org/10.48550/arXiv.1405.0312.

[35] Ziyue Huang Wang, Mingming Zhang, Yuan Gong, Qingjie Liu, and Yunhong. Generic knowledge boosted pre-training for remote sensing images, 21.01 2024. https://doi.org/10.48550/arXiv.2401.04614.

[36] Glenn Jocher. Instance segmentation datasets overview. Nettside, 18.04 2024. URL https://docs.ultralytics.com/datasets/segment/#generate-segmentation-dataset-using-a-detection-model. Hentet: 19.04.2024.

[37] Michael Goodwin Tasmiha Khan. What is yaml. Nettside, 2023. URL https://www.ibm.com/topics/yaml. Hentet: 11.12.2023.

[38] RangeKing. Brief summary of yolov8 model structure, 10.01 2023. URL https://github.com/ultralytics/ultralytics/issues/189. Hentet:09.05.2024.

A Hyperparametere

Tabell A.1: Noen av hyperparamteren som er brukt i NFI-modellen

augment: true	overlap mask: true
auto augment: autoaugment	patience: 100
batch: -1	perspective: 0.0
classes: null	plots: true
conf: null	pose: 12.0
device: null	pretrained: true
dropout: 0.2	scale: 0.5
epochs: 300	seed: 0
erasing: 0.4	shear: 0.0
freeze: null	split: val
imgsz: 1200	stream buffer: false
iou: 0.7	task: segment
lr0: 0.01	translate: 0.1
lrf: 0.01	val: true
mixup: 0.0	verbose: true
mode: train	vid stride: 1
model: yolov8m-seg.pt	visualize: false
momentum: 0.937	warmup bias lr: 0.1
mosaic: 1.0	warmup epochs: 3.0
name: NFI2	warmup momentum: 0.8
optimizer: auto	weight decay: 5.0E-4

Tabell A.2: Noen av hyperparamteren som er brukt i Combo-modellen

augment: false	overlap mask: true
auto augment: randaugment	patience: 100
batch: -1	perspective: 0.0
classes: null	plots: true
conf: null	pose: 12.0
device: null	pretrained: true
dropout: 0.0	scale: 0.5
epochs: 100	seed: 0
erasing: 0.4	shear: 0.0
freeze: null	split: val
imgsz: 1200	stream buffer: false
iou: 0.7	task: segment
lr0: 0.01	translate: 0.1
lrf: 0.01	val: true
mixup: 0.0	verbose: true
mode: train	vid stride: 1
model: yolov8n-seg.pt	visualize: false
momentum: 0.937	warmup bias lr: 0.1
mosaic: 1.0	warmup epochs: 3.0
name: nummer 3	warmup momentum: 0.8
optimizer: auto	weight decay: 5.0E-4

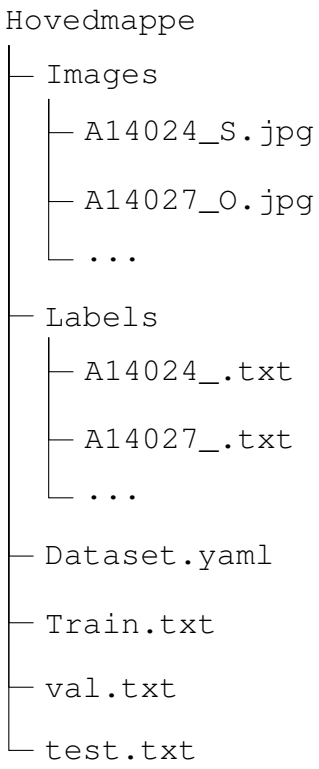
Tabell A.3: Noen av hyperparamteren som er brukt i iNat-modellen

augment: false	overlap mask: true
auto augment: randaugment	patience: 50
batch: 8	perspective: 0.0
classes: null	plots: true
conf: null	pose: 12.0
device: null	pretrained: true
dropout: 0.0	scale: 0.5
epochs: 100	seed: 0
erasing: 0.4	shear: 0.0
freeze: null	split: val
imgsz: 640	stream buffer: false
iou: 0.7	task: segment
lr0: 0.01	translate: 0.1
lrf: 0.01	val: true
mixup: 0.0	verbose: true
mode: train	vid stride: 1
model: yolov8n-seg.pt	visualize: false
momentum: 0.937	warmup bias lr: 0.1
mosaic: 1.0	warmup epochs: 3.0
name: train	warmup momentum: 0.8
optimizer: auto	weight decay: 5.0E-4

B Mappestruktur og filer

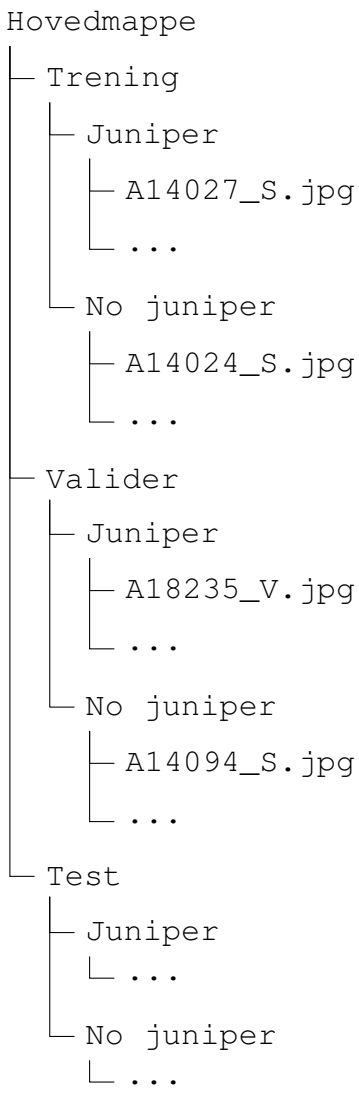
Mappestruktur

Segmenteringsmodellen til YOLO har en spesifikk filstruktur den krever for å fungere. Denne består av en hovedmappe med undermapper for alle bildene og alle maskene. Videre i hovedmappen vil det være en .yaml-fil (kapittel: B) som inneholder databane til hovedmappen, samt filnavnene på trenings-, valideringss- og testdataen. Den inneholder også navnene på alle klassene. [36]



Figur B.1: Mappestruktur segmentering

Mappestrukturen til klassifiseringsmodellen består av en hovedmappe med tre undermapper, en for trening, en for validering og en for test. Videre har hver av mappene en mappe for hver av klassene. I denne oppgaven er det to klasser, 'juniper' og 'no juniper'. Det gir følgende mappestruktur:



Figur B.2: Mappestruktur klassifisering

.yaml fil

YAML står for 'Yet Another Markup Language' og brukes vanligvis som konfigurasjonsfil. Det er laget for å kunne forstås enkelt av mennesker [37]. Formatet er brukt i oppgaven som konfigurasjonsfil i forbindelse med treningen av modellene. Strukturen er vist i figur B.3.

'names:' er en nummerert liste med alle klassene som man ønsker å segmentere. I denne oppgaven er det kun én klasse. Videre er 'path' datastien til hovedmappen (figur:B.1). 'train:' spesifiserer navnet eller datastien til txt-filen som inneholder bildene som skal brukes til trening (se kapittel:B). 'val:' spesifiserer navnet eller datastien til txt-filen som inneholder bildene som skal brukes til validering. 'test:' spesifiserer navnet eller datastien til txt-filen som inneholder bildene som skal brukes til testing.

```
names:
    0: juniper

path: /Master/NFI
train: train.txt
val: val.txt
test: test.txt
```

Figur B.3: Yaml fil

.txt fil

For å vise hvilke bilder som er del av trenings-, validerings- og testdatasettene, må man lage separate .txt-filer som inneholder datastien til bildene i forhold til det som er oppgitt som 'path' i YAML-filen (kapittel: B).

```
./images/A26167_O.jpg
./images/D63237_N.jpg
        ...
./images/D09239_V.jpg
```

Figur B.4: Eksempel på hvordan train.txt ser ut

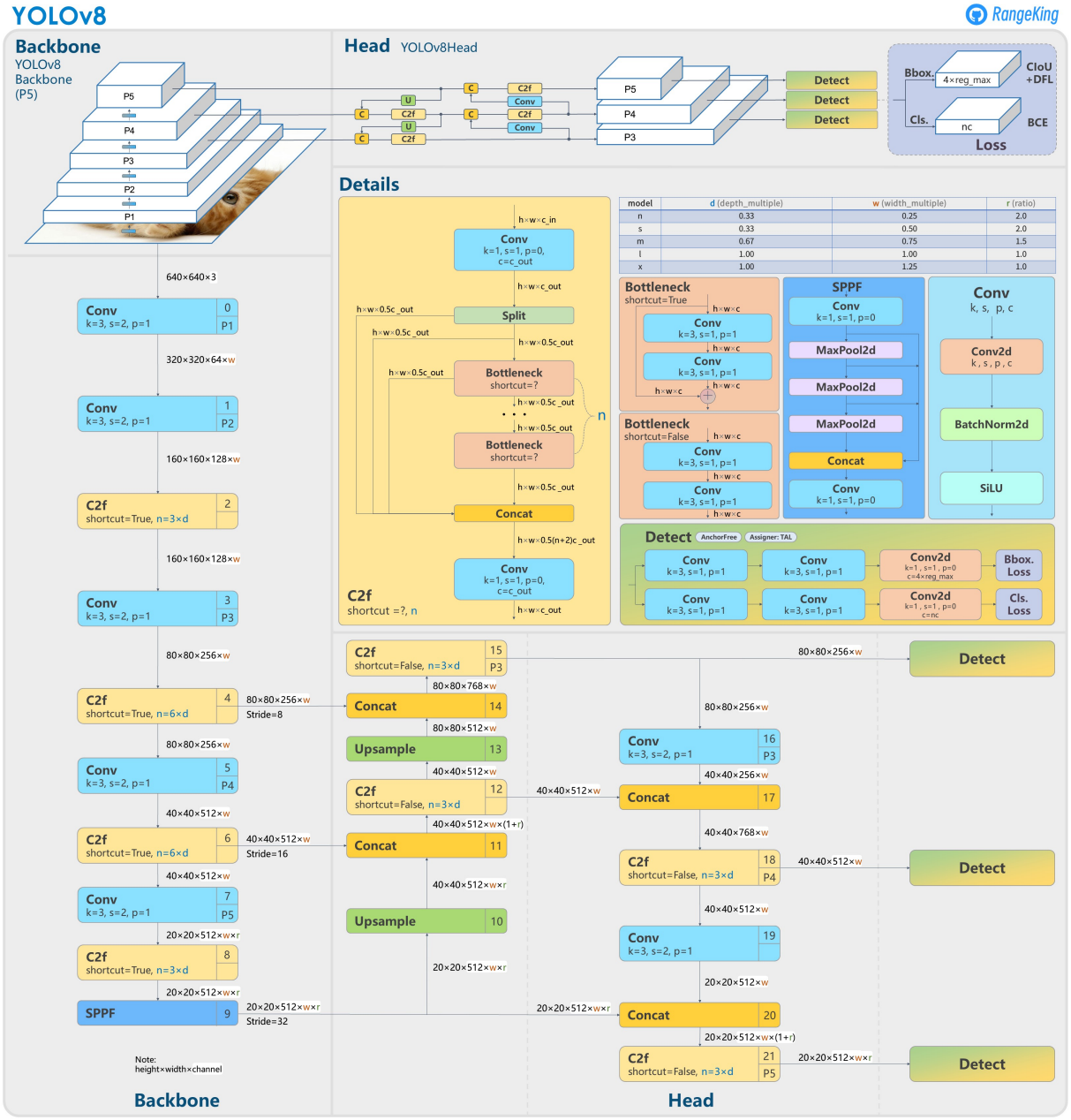
Maske .txt fil

Maskene til bildene finner man i 'labels'-mappen (figur: B.1). Masken til bildet har samme navn som bildet, men er en .txt-fil i stedet for en .jpg-fil. Disse filene lages automatisk av trainYolo og har følgende struktur [36]:

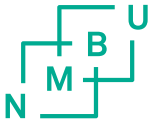
```
0 0.74050 0.73067 0.73975 0.73100 0.74025 0.73133 0.74100 0.73100

        <class-index> <x1> <y1> <x2> <y2> ... <xn> <yn>
```

Figur B.5: Eksempel på hvordan maskefilen ser ut

C YOLOv8 modellarkitektur



Figur C.1: YOLOv8-modell arkitekturen vist grafisk. Bilde er laget av githubbruker 'RangeKing' og hentet fra github[38]



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

Postboks 5003
NO-1432 Ås
Norway