FFI-RAPPORT

17/17026

Teknologiske muligheter for Tolletaten

mønstergjenkjenning og maskinlæring

–
Idar Dyrdal
Lars Aurdal
Kristin Hammarstrøm Løkken
Thor Engøy

Teknologiske muligheter for Tolletaten

mønstergjenkjenning og maskinlæring

Idar Dyrdal
Lars Aurdal
Kristin Hammarstrøm Løkken
Thor Engøy

Forsvarets forskningsinstitutt (FFI) 29. desember 2017

Emneord

Mønstergjenkjenning
Maskinlæring
Kunstig intelligens
Nevrale nettverk
Dyp læring
Tolletaten

FFI-rapport
FFI-RAPPORT 17/17026

Prosjektnummer
530201

ISBN
P: 978-82-464-3028-7
E: 978-82-464-3029-4

Godkjent av
Rune Lausund, *forskningsleder*
Janet Martha Blatny, a*vdelingssjef*

Sammendrag

Tolletaten har gitt Forsvarets forskningsinstitutt (FFI) i oppdrag å gjennomføre en teknologisk mulighetsstudie som kan bidra til etatens strategi for utvikling av organisasjonen på kort, mellomlang og lang sikt. Våren 2017 gjennomførte FFI en breddestudie som omhandlet teknologiske muligheter for Tolletaten.

Denne rapporten er resultatet av en videre studie av mønstergjenkjenning og maskinlæring. Dette er modne fagfelt som vil kunne være nyttige for Tolletaten, ved at man lærer opp en maskin til å utføre automatisk inspeksjon av varer og reisegods som krysser grensen og utvelgelse av objekter for fysisk kontroll. Alle sensorteknologier, såvel som andre kilder til informasjon, kan brukes som grunnlag for maskinlæring og mønstergjenkjenning. Ved at sensorinformasjonen suppleres med annen tilgjengelig informasjon vil treffsikkerheten kunne bli større. Det anbefales at konkrete kontrollsituasjoner og informasjonskilder studeres nærmere for å kunne vurdere noe om eventuell ytelse til et slikt system.

Systematisk innsamling og lagring av data er en forutsetning for maskinlæring. Et datasett som består av sensordata sammenholdt med en domeneeksperts tolkning av dataene må etableres først. Etter opplæring på et slikt merket datasett vil maskinen kunne finne mønstre og sammenhenger som man er ute etter f.eks. i en strøm av sensordata. Datainnsamlingsprosessen bør være enkel, helst automatisert og tilpasset den øvrige arbeidsflyten i kontrolloppgavene. For å gjøre systemet for mønstergjenkjenning i stand til å oppdager ukjente mønstre anbefales det at det også samles inn data fra kontroller utført etter tilfeldige utvalg.

Rapporten gir noen eksempler på hvordan maskinlæring (med vekt på dyp læring) kan anvendes på informasjonskilder og sensorsystemer som Tolletaten bruker i dag, som bl.a. røntgen. Et datasystem med flere ulike sensorer og informasjonskilder kan over tid lære seg hva som er normalt, og deretter melde fra om avvik fra normalsituasjonen slik at utvalget av objekter for kontroll blir mer treffsikker. Tidlig oppdagelse av avvik fra normalsituasjonen vil også kunne understøtte bedre planlegging og utnyttelse av Tolletatens ressurser.

Summary

The Norwegian Customs Agency has tasked The Norwegian Defense Research Establishment (FFI) to carry out a technology feasibility study that can contribute to the agency's strategy for developing the organization in the short, medium and long term. We have conducted a study on emerging technological opportunities for the Norwegian Customs Agency.

This report is the result of an additional study on pattern recognition and machine learning. These are mature technological areas that can be used by The Norwegian Customs Agency for training a machine to carry out automatic inspection of goods crossing the border. Data from all sensor technologies, as well as additional sources of information, can be used as input to machine learning and pattern recognition. Fusion of sensor information with other available data, will in most cases lead to improved performance of the recognition system. A further study of actual control situations and sources of information is thus recommended for assessing the potential system performance.

Systematic collection and storage of data is a prerequisite for machine learning. First of all a dataset consisting of sensor data interpreted by an expert, i.e. a person with domain knowledge, must be established. After training on such a labelled data set, the recognition system may be able to find patterns and correspondances of interest in a continuous stream of input data. The data collection procedure should be simple, preferrably automatic and an integral part of the daily workflow carried out by the customs officials. To further enhance the pattern recognition systems ability to detect unknown patterns, it is recommended that data from random controls are also included in the data set.

The report provides som examples on how machine learning (with emphasis on deep learning) can be applied to information sources and sensor technologies already in use by the Customs Agency, e.g. X-ray imaging. A pattern recognition system using input from a variety of sensor systems and information sources, may over a period of time be trained to discriminate between the normal situation and anomalies (deviations from the normal situation), thus providing a more reliable selection of objects (e.g. merchandize) for further control. Early anomaly detection will also enable better planning of customs operations and more efficient use of the resources in the Norwegian Customs Agency.

Innhold

1 Innledning

Tolletaten har gitt Forsvarets forskningsinstitutt (FFI) i oppdrag å gjennomføre en teknologisk mulighetsstudie for å underbygge etatens strategi for utvikling av organisasjonen på kort, mellomlang og lang sikt. Våren 2017 utførte FFI en breddestudie, som ble presentert i [1], heretter kalt *breddestudien*. Med utgangspunkt i denne breddestudien har Tolletaten bedt om videre studier på to tematiske områder. Den ene studien dreier seg om automatisering av postmottak, og resultatet fra dette arbeidet, heretter kalt *postmottaksstudien*, er presentert i [2]. Den andre studien er presentert i denne rapporten, og tar for seg bruk av mønstergjenkjenning og maskinlæring, sett i lys av Tolletatens oppgaver.

De overordnede temaene for den teknologiske mulighetsstudien var kunstig intelligens, maskinlæring og sensorsystemer – begreper som er beskrivende for den teknologiske utviklingen. I breddestudien gjorde vi en avgrensning, ved at vi valgte å fokusere på dyp læring som metode for kunstig intelligens. Dyp læring er et fagfelt som er i rask utvikling og som gir svært gode resultater. Målet med denne videre studien er å se nærmere på både dyp læring og andre maskinlæringsmetoder, og hvilken anvendelse slike metoder kan ha for Tolletaten.

Maskinlæring er et fagfelt i grenselandet mellom statistikk, matematikk og informasjonsvitenskap. Fokus for dette fagfeltet er metoder som i større eller mindre grad gir maskiner mulighet til å treffe beslutninger basert på tilgjengelige data. Et dagsaktuelt eksempel på et slikt system er fingeravtrykksleseren som nå finnes på mange ulike håndholdte enheter som for eksempel mobiltelefoner. Systemet er designet slik at det kan analysere data fra en fingeravtrykksleser for så å avgjøre om et gitt fingeravtrykk er eierens fingeravtrykk. Systemet har altså lært seg å kjenne igjen en gitt persons fingeravtrykk (har altså lært noe om sine omgivelser) og kan, basert på dette, ta ulike avgjørelser. Dette kan høres ut som en triviell oppgave, ”det er jo bare å sammenlikne med bildet av fingeravtrykket”, men fagfolk har brukt flere tiår på å få dette til å fungere godt nok for praktisk bruk.

1.1 Historikk

Tradisjonell maskinlæring/mønstergjenkjenning dreier seg i hovedsak om klassifisering av *objekter* (mønstre) til én av flere mulige *klasser* eller kategorier [3]. Objektene kan være trykte bokstaver, håndskrevne tegn eller fysiske gjenstander i bilder, elektromagnetiske signaler, ulike tilstander i f.eks. medisinsk sammenheng, og mye mer. Her trener man opp en *klassifikator* til å skille mellom forskjellige kategorier ut fra *egenskaper* ved objektene. Maskinen lærer da å skille mellom klassene, basert på eksempler hvor det er kjent hvilken klasse hvert objekt tilhører. Egenskapene som maskinen bruker til å klassifisere objektene, er tallstørrelser avledet fra objektene. For et objekt i et bilde kan f.eks. høyde og bredde være mulige egenskaper.

Som fagfelt har maskinlæring røtter tilbake til årene rundt 1950 og har siden dette vært i rivende utvikling. En viktig gren innen forskningen på dette store området har vært utviklingen av systemer som på et eller annet nivå etterligner menneskehjernens antatte måte å gjøre beregninger på. Allerede i 1957 foreslo Frank Rosenblatt at man kunne bruke et såkalt perceptron (en enkel modell for et biologisk nevron) for å gjøre beregninger [4]. Disse tidlige arbeidene ga senere opphav til forskning

på såkalte nevrale nett, et forskningsfelt som har vært en underdisplin av maskinlæring siden den gang. Selv om nevrale nett lenge har vært gjenstand for omfattende forskning var resultatene ikke spesielt imponerende, og ønsket om at man skulle kunne klare å etterligne menneskehjernens ytelse virket lenge som et svært fjernt håp. I de siste årene, grovt anslått i tiden etter 2010, har dette imidlertid forandret seg kraftig.

Denne utviklingen kan forklares på en rekke måter, men det virker naturlig å peke på to drivkrefter som har hatt en spesiell innflytelse. En underliggende årsak har vært den dramatiske endringen i tilgangen på billig datakraft, en utvikling man kan takke spillindustrien for. I dag kan rimelige grafikkkort for massemarkedet yte 10 teraflops (10 milliarder flyttallsoperasjoner per sekund) over tid, noe som gjør at dagens nevrale nett kan være enormt mye større og mer komplekse enn hva man tidligere har kunnet bruke. En annen årsak er at man i den senere tiden har gjort metodiske framskritt, som har vist seg å være nøkler for å oppnå bedre ytelse. Konvolusjonsnett er et viktig eksempel på dette.

1.2 Vanlige ord og begreper

Begrepene *mønstergjenkjenning* og *maskinlæring* betraktes ofte som synonymer i litteraturen, men maskinlæring er vanligvis avgrenset til såkalt *ledet læring* (se avsnitt 2.2). Fagfeltet mønstergjenkjenning omfatter derimot også *ikke-ledet læring* (trening av en klassifikator basert på ukjente data) og *klyngeanalyse* (Cluster Analysis), dvs. å finne struktur og sammenhenger i ukjente data. I tillegg brukes begrepet mønstergjenkjenning også om selve gjenkjenningsprosessen, dvs. gjenkjenning av ukjente objekter med en ferdigtrent klassifikator.

Her bruker vi derfor maskinlæring om den prosessen som leder frem til en algoritme eller maskin som foretar gjenkjenning av nye data, f.eks. klassifisere objekter i bilder. Det er verdt å merke seg at mens maskinlæring kan være tidkrevende og kreve mye regnekraft, kan selve mønstergjenkjenningen gå svært raskt, og med minimal prosessorkraft. Dyp læring er som oftest langt mer krevende enn klassiske maskinlæringsmetoder, mens et ferdigtrent dypt nett for bildegjenkjenning i mange tilfeller kan kjøres i sann tid på en smarttelefon.

Både mønstergjenkjenning og maskinlæring er fagfelt innen det større området *kunstig intelligens*. Et beslektet fagfelt er *maskinsyn*, som i videste forstand har som formål å lære maskiner til å forstå verden omkring seg ved hjelp av kameraer. Her inngår både mønstergjenkjenning og maskinlæring, men da rettet mot analyse av bildeinformasjon. Kunstig intelligens omfatter også *ekspertsystemer* (regelbaserte systemer, konstruert av menneskelige eksperter) og til en viss grad også *stordata* (Big Data), fordi metoder fra mønstergjenkjenning også brukes her.

1.3 Rapportens oppbygning

Denne rapporten gir en innføring i ulike metoder fra mønstergjenkjenning og maskinlæring. Kapittel 2 tar for seg klassisk mønstergjenkjenning og ulike maskinlæringsmetoder for trening av klassifikatorer, som benytter håndlagde egenskaper som input. Kapittel 3 beskriver et utvalg av nyere metoder, mens kapittel 4 gir en innføring i dype nett og dyp læring. De ulike metodene beskrevet i disse kapitlene er forsøkt illustrert gjennom relevante eksempler. Kapittel 5 beskriver

forslag til systematisk datainnsamling som Tolletaten bør iverksette, siden store, merkede datasett er forutsettningen for vellykket maskinlæring. Rapporten avsluttes med konklusjoner og anbefalinger for videre arbeid i kapittel 6.

2 Klassisk mønstergjenkjenning

Dette kapitlet tar for seg mønstergjenkjenning (tradisjonell maskinlæring), som kan betraktes som et fagområde innen kunstig intelligens. Målsettingen med hele fagfeltet er i videste forstand å lære maskiner opp til å sanse og forstå verden omkring seg, og bli i stand til å løse mange av de praktiske oppgaver vi mennesker presenteres for til daglig.

Mer konkret er hensikten med mønstergjenkjenning i første rekke å klassifisere objekter eller tilstander til én av flere mulige klasser. Tradisjonelt løses dette problemet ved å trene opp en såkalt klassifikator til å skille mellom de ulike kategoriene ut fra målte egenskaper ved objektene. Sagt på en annen måte, ønsker man å lære en maskin å skille mellom klassene ved å presentere den for eksempler på objekter som allerede er klassifisert. Det er denne treningsprosessen som går under betegnelsen maskinlæring.

Klassifiseringsprosessen bygger på sensorinformasjon, ofte i form av bildeinformasjon fra kameraer (synlig lys eller termisk) og andre bildedannende sensorer (Lidar, røntgen, terahertz osv.), men kan også være informasjon i form av tidsrekker eller punktmålinger (f.eks. vekt). I denne rapporten bruker vi bilder som gjennomgående eksempel, slik at det er ulike objekter i bildene som skal klassifiseres. Vi understreker imidlertid at maskinlæring kan brukes til mange andre – og langt mer komplekse – datasett.

Mønstergjenkjenning og maskinlæring inngår i dag i en rekke praktiske anvendelser. Her kan nevnes lesing av tekst, talegjenkjenning, deteksjon og gjenkjenning av ansikter (brukes bl.a. i mobiltelefoner, digitale kameraer og bilderedigeringsprogrammer), gjennkjenning av fingeravtrykk og annen biometrisk informasjon, analyse av satelittbilder for kartlegging av jordressurser, maskinsyn og robotikk, medisinske anvendelser (deteksjon av kreftceller, analyse av EKG- og EEG-signaler), kvalitetskontroll i produksjonsprosesser og ikke minst i inspeksjon av gjenstander på et transportbånd. Det siste er vel den anvendelsen som minner mest om mulige bruksområder innen Tolletaten.

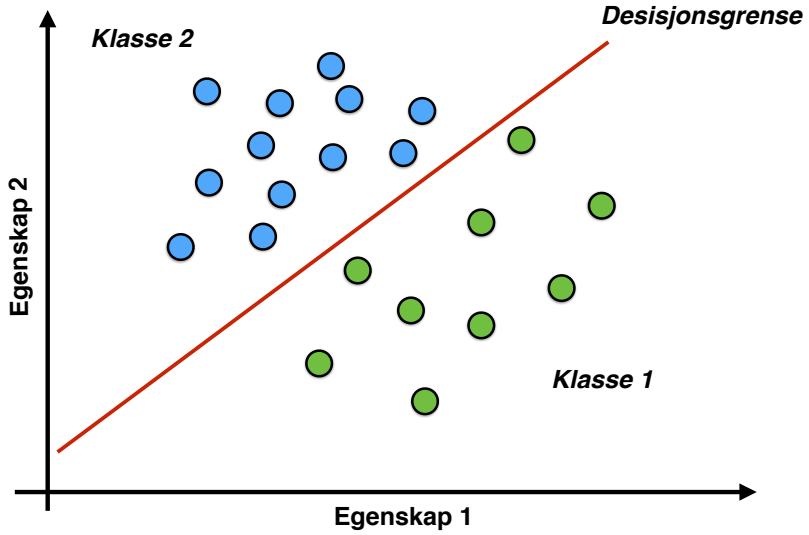
Formen på klassifikatoren kan være lineær, kvadratisk eller av høyere orden. I våre eksempler bruker i lineære klassifikatorer. Man kan bruke bruke treningssettet til å finne optimale verdier for parametrene som inngår i klassifikatoren, slik at en feilfunksjon (avledet fra antall feilklassifiseringer i treningssettet) blir så lav som mulig. Eksempler på slike metoder er de såkalte perceptron- og relaksasjonsalgoritmene og *Support Vector Machines* (SVM). Grunnidéen bak perceptronmetodikken har senere blitt videreført til nevrale nett. Metoder for å trene klassifikatorer av disse typene er beskrevet i bl.a. [3] og [5].

2.1 Klassifisering

Når objekter i bilder skal klassifiseres, trekkes det ut egenskaper i form av tallstørrelser, eksempelvis høyde, bredde, areal og omkrets, eller egenskaper avledet fra farge, lysstyrke og tekstur. Disse egenskapene er egnet til å skille mellom objektklasser, eller for å skille spesifikke objekttyper fra alt mulig annet i bildet. Her vil digital bildebehandling brukes til å trekke ut de ønskede egenskapene. I den videre maskinelle behandlingen er objektene representert ved de valgte egenskapene.

Dette er den såkalte beslutningsteoretiske metodikken som typisk brukes i sammenhenger der det er

snakk om et begrenset antall mulige klasser (se f.eks. [3] eller andre lærebøker i mønstergjenkjenning). Figur 2.1 viser et eksempel med to egenskaper plottet mot hverandre for et utvalg av eksempler fra to klasser.



Figur 2.1 *Eksempel på objekter fra to klasser (henholdsvis grønne og blå sirkler), representert ved to egenskaper. Den røde linjen (desisjonsgrensen) deler her dette todimensjonale egenskapsrommet inn i separate regioner, én for hver klasse. Ukjente objekter blir målt på de to egenskapene, slik at resultatet utgjør et punkt i egenskapsrommet. Objektene kan dermed klassifiseres ut ifra hvilken hvilken side av desisjonsgrensen disse punktene ligger på.*

La oss si at problemet består i å avgjøre hvorvidt en kunstgjenstand laget av tre er ekte eller en forfalskning, der ekte gjenstander typisk er laget av lyst trevirke med lav kornethet i veden, og la egenskap 1 være lysheten og egenskap 2 være kornetheten. Hvis nå de grønne sirklene i figur 2.1 representerer ekte gjenstander og de blå falske, kan man trene opp en klassifikator for å skille mellom falske og ekte gjenstander. Den heltrukne, røde linjen i figuren er en såkalt *desisjonsgrense* som deler inn rommet i regioner svarende til hver av klassene man ønsker å skille mellom. I dette eksemplet skiller desisjonsgrensen perfekt mellom eksempelobjektene fra de to klassene, men dette vil generelt ikke være tilfelle.

Målet med treningen (maskinlæringen) er å komme frem til desisjonsgrenser som skiller klassene best mulig, slik at sannsynligheten for å tilordne et objekt til feil klasse blir så lav som mulig. I vårt eksempel med kunstgjenstander, vil maskinen kunne lære å gjenkjenne en forfalskning ved å trenes opp på et sett med manuelt klassifiserte bilder av ekte og falske gjenstander.

2.2 Ledet læring

Maskinlæring foretas vanligvis ved såkalt *ledet læring*, som forutsetter at klassetilhørigheten til eksempelobjektene i treningssettet er kjent. Gangen i denne prosessen er illustrert i figur 2.2. Ut fra objektene i treningssettet (til venstre i figuren) vil man tradisjonelt gjøre et valg av egenskaper basert på forhåndskunnskap om problemstillingen. Tallverdiene til disse egenskapene blir beregnet i

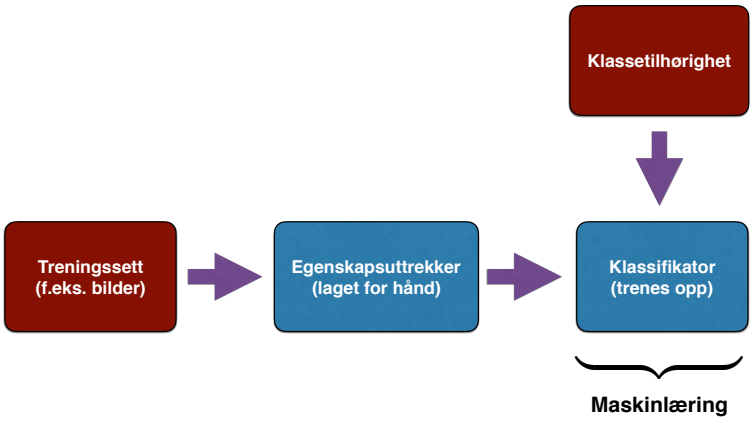
den såkalte egenskapsuttrekkeren for hvert av objektene i treningssettet. Her vil det typisk foretas en form for bildeanalyse, og resultatet er et sett av tallstørrelser $x_1, x_2, \ldots, x_d$.

Egenskapene for objektet organiseres i en *egenskapsvektor*

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

Generelt kan antall egenskaper d og antall klasser være vilkårlig. I våre eksempler bruker vi imidlertid to egenskaper, slik at disse kan spenne ut et todimensjonalt egenskapsrom, slik som i figur 2.1. Målet er å dele egenskapsrommet inn i adskilte desisjonsregioner med minst mulig overlapp mellom klassene. Liten overlapp leder til relativt sikker klassifisering (lav feilrate). Bruk av flere egenskaper gir vanligvis mer informasjon for klassifisering, og derved lavere feilrate.

Legg merke til at i denne tradisjonelle fremgangsmåten er egenskapene definert på forhånd; de er laget for hånd ut fra kunnskap om problemstillingen, og er ikke en del av selve treningsprosessen. I vårt eksempel med kunstgjenstander laget i tre, er det to forhåndsbestemte egenskaper; lyshet og kornethet. De forhåndsberegnede egenskapsvektorene sendes sammen med klassetilhørigheten inn til selve treningsprosessen som genererer selve klassifikatoren for problemet.



Figur 2.2 *Illustrasjon av prinsippet for ledet læring. Et sett av objekter (treningssettet), representert ved utvalgte (håndlagde) egenskaper og med kjent klassetilhørighet, tas som input til én av mange mulige treningsmetoder for å generere en klassifikator.*

Parametrene i klassifikatoren finnes fra treningssettet ved ledet læring. Eksempler på metoder her er de såkalte perceptron- og relaksasjonsalgoritmene, som søker etter en separerende rett linje dersom klassene er lineært separable eller en best mulig rett linje dersom problemet er ikke-separabelt. Andre muligheter er Widrow-Hoff og Ho-Kashyap algoritmene, som bruker minste kvadraters metode til å finne en rett linje som skiller klassene best mulig, enten datasettet er lineært separabelt eller ikke. Slike treningsmetoder, som er grundig behandlet i mange lærebøker i mønstergjenkjenning, f.eks. [3] og [5], kan generaliseres til problemer med mange klasser.

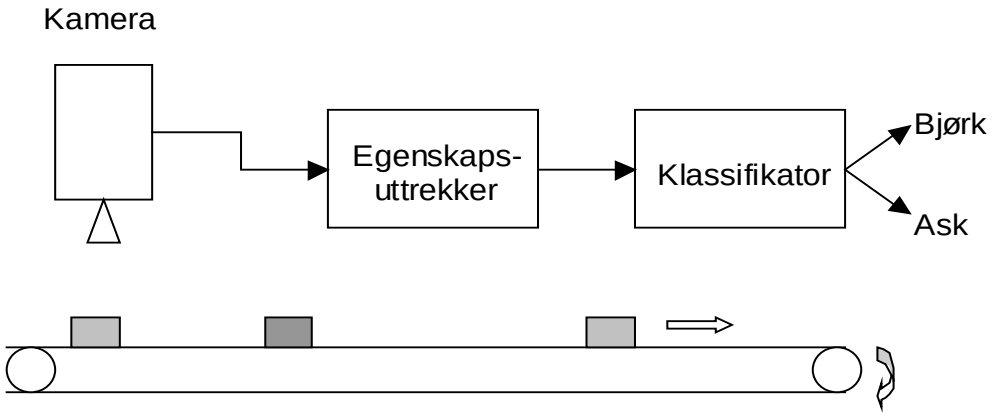
En beslektet maskinlæringsmetode er såkalte "Support Vector maskiner" (SVM). Her forutsettes i utgangspunktet at klassene er lineært separable (dvs. kan separeres av en lineær klassifikator),

men metodikken kan generaliseres for å trene klassifikatorer av høyere orden, og har vist seg å gi svært gode resultater på mange tollrelevante problemer (se eksempler i neste avsnitt). Forøvrig kan perceptronmetodikken betraktes som en forløper til det store underfeltet innen mønstergjenkjenning, som går under navnet nevrale nett. SVM og nevrale nett behandles nærmere i kapittel 3. Se også lærebøkene [3] og [5] og artikkelen [6], som tar for seg ulike anvendelser av SVM.

Et problem med alle typer klassifikatorer, ikke minst nevrale nett og andre høyereordens, ikke-lineære klassifikatorer, er faren for overtrening, dvs. at klassifikatoren blir spesialisert til å gjenkjenne objektene i treningssettet men dårlig til å gjenkjenne nye, ukjente objekter. Dette skyldes at det blir for få treningsobjekter i forhold til antall parametre i klassifikatoren, slik at treningen blir lite robust. Problemet vokser med økende antall egenskaper som det skal regnes på, og med økende kompleksitet for klassifikatoren.

2.3 Eksempler på bruk

Metodene beskrevet her egner seg best der problemet omfatter et begrenset antall mulige klasser, og er svært aktuelle i tilfeller med to klasser, f.eks. "treff" og "ikke treff". Typiske anvendelser er inspeksjon av gjenstander på et samlebånd, f.eks. for kvalitetskontroll i en industriell produksjonsprosess der man ønsker å skille ut feilproduserte gjenstander eller for å sortere objekter av forskjellig type (se figur 2.3).



Figur 2.3 *Eksempel på bruk av mønstergjenkjenning for automatisk sortering av gjenstander på et transportbånd. Et kamera tar bilder av trestykker fra to mulige arter (ask og bjørk), trekker ut egenskaper fra bildene (lysstyrken til veden, kornethet mm.). De målte egenskapene sendes inn til klassifikatoren, som foretar valget av klasse.*

Første skritt i denne gjenkjenningsprosessen er å skille de interessante objektene fra omgivelsene. For bilder vil dette ofte innebære en såkalt "segmentering" av bildet; en prosess der bildet deles inn i logisk sammenhørende grupper av piksler. Målet er å trekke ut segmenter (omriss) fra bildet, som i størst mulig grad omslutter selve objektet uten å ta med omgivelsene.

I den etterfølgende egenskapsuttrekkingen vil man da beregne ulike tallstørrelser ut fra sensorinformasjonen innenfor de segmentene som er trukket ut fra rådataene. Det valgte settet av egenskaper går

deretter inn til klassifikatoren, som foretar valget av klasse. Det som kjennetegner disse metodene er at de er basert på tallinformasjon om objektene. I figur 2.1 tilsvarer dette tallene på de to aksene, slik at de to tallene til sammen utgjør et punkt (vektor) i egenskapsrommet (der hvor man finner grønne og blå prikker i figuren). To punkter som ligger nær hverandre i dette egenskapsrommet vil da mest sannsynlig tilhøre samme klasse, mens punkter langt fra hverandre trolig tilhører forskjellige klasser.

Vareinspeksjon er kanskje den anvendelsen av bildebasert mønstergjenkjenning innen Tolletaten som minner mest om illustrasjonen i figur 2.3, men da hovedsakelig på røntgenbilder i stedet for kamera for synlig lys.

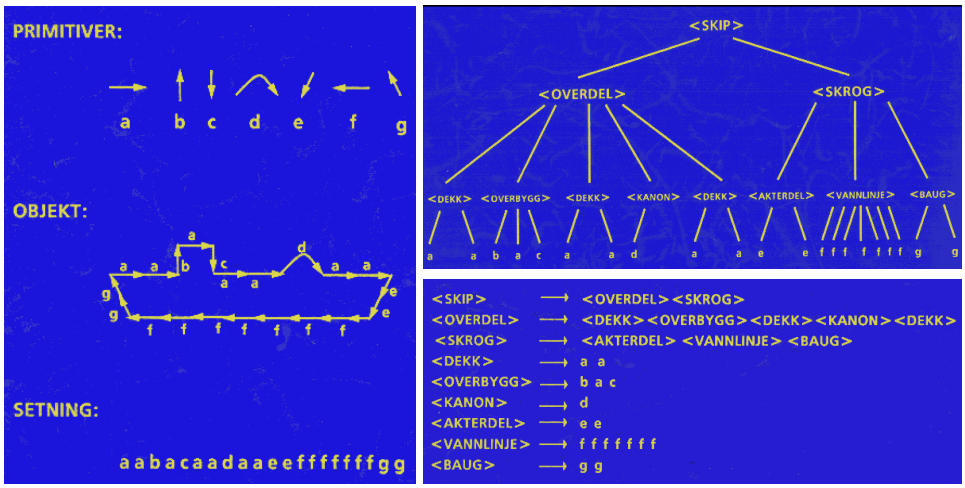
Andre anvendelser er ANPR, basert på dagkamera, men det er også verdt å nevne mønstergjenkjenning basert på annen type input, f.eks. for å avdekke "Pattern-of-life" (POL). Her er det aktuelt å bruke maskinlæring for å avdekke mønstre (f.eks. adferd) over tid, en type problemstillinger der metoder beskrevet i de følgende kapitlene kanskje vil være mer aktuelle.

3 Moderne metoder

Mens det foregående kapitlet tok for seg mønstergjenkjenningsmetoder med opphav tilbake til 1950- og 1960-årene, omhandler dette kapitlet i hovedsak metoder som fikk sitt gjennombrudd fra ca. 1980 og frem til begynnelsen av 2000-tallet. Unntaket er temaene "beslutningstrær" og "syntaktiske metoder" som går lengre tilbake, men er tatt med her som en innledning til nyere teknikker.

3.1 Syntaktiske metoder

I noen tilfeller, spesielt der antall mulige klasser er stort, kan mønstergjenkjenning også bestå i å komme frem til en *beskrivelse* av sammensatte objekter ved hjelp av mer grunnleggende komponenter. Denne greinen av faget refereres ofte til som *strukturell mønstergjenkjenning*. Her benyttes såkalte *syntaktiske* metoder. Input til denne prosessen er symbolsk informasjon, ofte en streng av symboler som analyseres i henhold til et gitt regelverk. Resultatet av analysen er en hierarkisk beskrivelse av objektet. Denne beskrivelsen kan omfatte relasjoner mellom de ulike komponentene, for eksempel hvor ulike deler av objektet er plassert i forhold til hverandre, og kan derfor være utgangspunkt for en mer detaljert klassifisering av objektet.



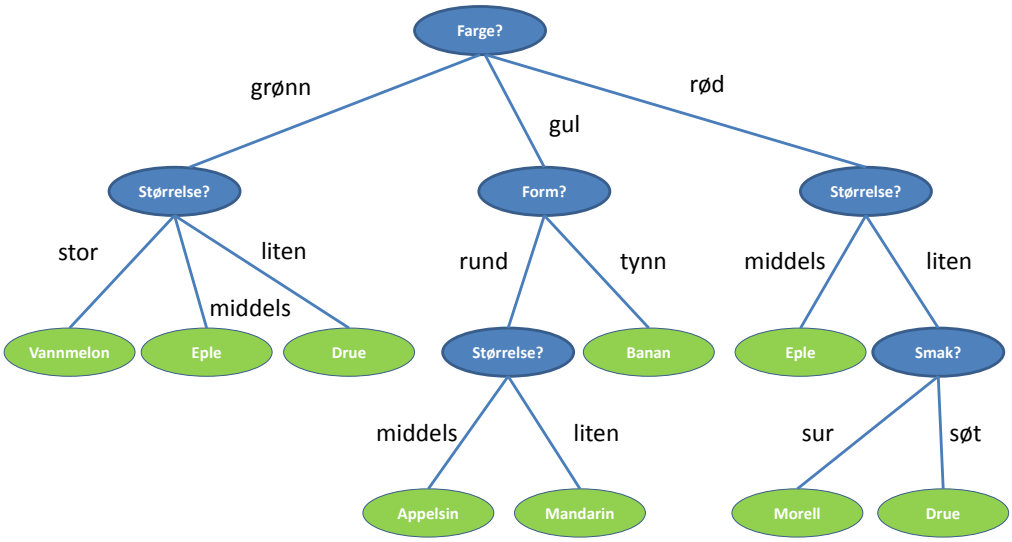
Figur 3.1 *Syntaktisk beskrivelse av skipskontur i form av en streng av primitiver (til venstre), hierarkisk beskrivelse (øverst til høyre) og regelverket (grammatikken) som angir hvilke sammenhenger som er tillatt mellom de ulike elementene (nederst til høyre). Illustrasjon fra tidligere FFI-prosjekt.*

Strukturelle metoder kan som nevnt brukes der det er svært mange mulige klasser og/eller der det er viktig å kartlegge sammenhengen mellom objekter/delobjekter av forskjellig type. Eksempel på bruk av syntaktiske metoder (beskrivelse av skipskontur) er vist i figur 3.1. I dette eksempelet blir omrisset av skipet (til venstre) først delt opp i linjesegmenter av forskjellig form, symbolisert ved a, b, c, d, e, f eller g. Disse grunnsymbolene, *primitivene*, settes sammen til en streng (setning), som vist nederst til venstre. Setningen analyseres (*parses*) for å fastslå om den er i overensstemmelse med regelverket (*grammatikken*) nederst til høyre. I dette eksempelet beskriver grammatikken hvordan omrisset av et skip skal se ut. Dersom parsingen lykkes, bygges det samtidig opp en hierarkisk

beskrivelse av skipet, med skipet som helhet på toppen av hierarkiet og primitivene på laveste nivå, som vist øverst til høyre i figuren.

3.2 Beslutningstrær

Klassifisering av et objekt kan i mange tilfeller foretas ved å besvare en sekvens av spørsmål. En slik beslutningsprosess kan representeres som en trestruktur; et flertrinns klassifiseringssystem der prosessen starter i en "rotnode", med forgreininger til undernoder fra nivå til nivå i treet. I hver node i treet foretas således et valg av veien videre til to eller flere mulige noder på neste nivå. Dette gjentas fra nivå til nivå inntil man ender opp i en "blad-node" som forhåpentligvis svarer til riktig klasse (riktig beslutning).



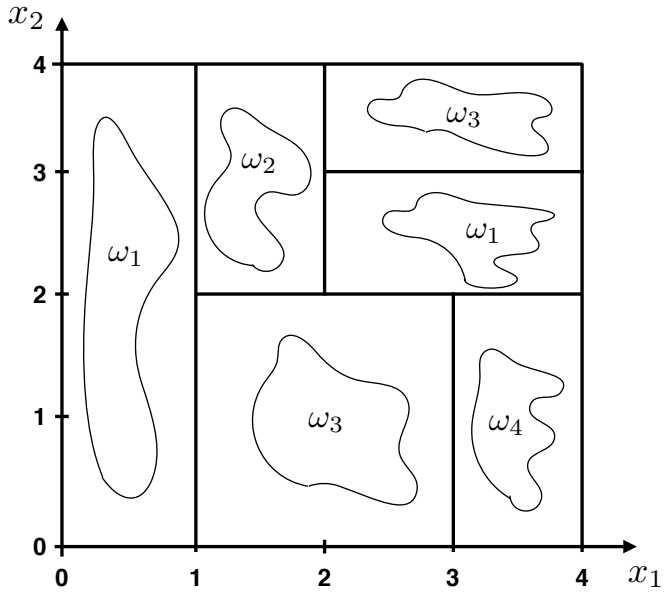
Figur 3.2 *Eksempel på beslutningstre som bruker kategorisk informasjon. Dette treet skal klassifisere frukt ut fra farge, størrelse, form og smak. En frukt som er beskrevet som "gul", "rund" og "middels stor" vil av dette beslutningstreeet bli klassifiser som en appelsin.*

Beslutningstrær brukes ofte for klassifisering basert på kategorisk (ikke-numerisk) informasjon, i likhet med syntaktiske metoder beskrevet i foregående avsnitt. Et eksempel på et slikt tre er vist i figur 3.2. Her skal treet finne frem til hvilken frukt det er snakk om ut fra en kvalitativ beskrivelse av farge, form, størrelse og smak. De blå nodene i figuren er såkalte "ikke-terminale" noder. Her stilles et spørsmål, som avgjør veivalget til neste nivå i treet. Beslutningsprosessen starter i rotnode øverst i figuren, og avhengig av svaret på det første spørsmålet velges en av de tre etterkommerne, der det stilles nye spørsmål med to eller flere alternativer. Til slutt ender prosessen i en av de grønnfargene bladene nedes i treet, forhåpentligvis med riktig resultat. Legg merke til at de ulike kriteriene godt kan forekomme flere steder i treet.

Beslutningstrær kan også brukes på numeriske data, i likhet med metodene behandlet i kapittel 2. Dette kan være informasjon i form av bilder eller signaler (tidsrekker) eller andre typer målinger

som kan karakteriseres ved hjelp av tallstørrelser. Valget i hver node kan utføres på bakgrunn av verdien til én enkelt egenskap i form av en *terskling* av verdien til egenskapen. Et eksempel på dette kan være at hvis den målte verdien er større enn x, så velges klasse A, og hvis verdien er mindre enn eller lik x, så velges klasse B. Valget kan også være basert på en kombinasjon av egenskaper, der det kan brukes forskjellige kombinasjoner i de ulike nodene.

Det første alternativet brukes ofte ved "sekvensiell klassifisering", der klassifiseringsprosessen foretas ved å benytte én egenskap om gangen i en sekvens av beslutninger. Et slikt beslutningstre gir god innsikt i hvordan maskinen løser problemet, og kan forøvrig lett trenes opp av en ekspert på problemstillingen ved at vedkommende avgjør ut fra domenekunnskap hvilke egenskaper og hvilke terskler som skal brukes i hver enkelt node.

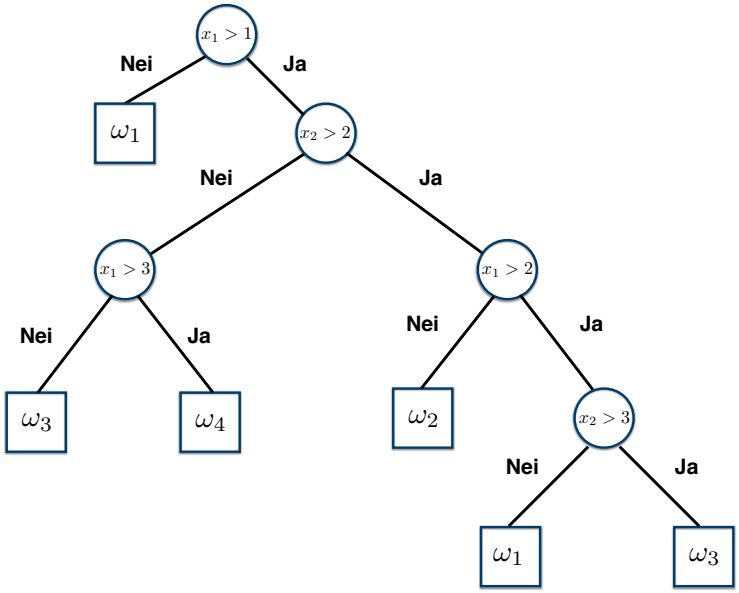


Figur 3.3 *Todimensjonalt egenskapsrom for et problem med fire klasser. Rommet er her delt inn i rektangulære beslutningsregioner, ut fra hvor objekter fra de ulike klassene typisk vil befinne seg i rommet.*

La oss se på et eksempel. Figur 3.3 viser et todimensjonalt egenskapsrom med egenskapene x_1 og x_2 for et problem med fire klasser, som her er kalt $\omega_1, \omega_2, \omega_3$ og ω_4. De lukkede kurvene viser hvilke områder i dette egenskapsrommet hvor objekter fra de ulike klassene typisk vil befinne seg. I dette tilfellet er man så heldige at det ikke er noen overlapp mellom klassene. Det er da mulig å dele inn egenskapsrommet i perfekte "beslutningsregioner" (rektanglene i figuren), der det knyttes en klassetilhørighet (symbolene $\omega_1, \ldots, \omega_4$ i figuren) til hver region. Ukjente objekter vil da bli tilordnet den klassen som svarer til regionens klassetilhørighet.

Klassifiseringsprosessen kan her utføres som en sekvens av tersklingsoperasjoner, og kan fremstilles som et beslutningstre som vist i figur 3.4.

Beslutningstrær av den andre typen kan betraktes som en sammenstilling av flere ordinære klassifikatorer; én klassifikator i hver node, som tar seg av forskjellige delproblemer i klassifiseringsprosessen. Hver av disse klassifikatorene er da delt opp på relevante undermengder av treningssettet. Forskjellen



Figur 3.4 *Eksempel på beslutningstre som bruker numerisk informasjon. Dette tréet er en (av mange) mulige implementasjoner av klassifiseringsprosessen for problemet i figur 3.3.*

på et slikt tre, og tréet i figuren ovenfor er at logikken i hver node er mer omfattende; i stedet for å foreta en enkel terskling, velges de ulike alternativene (to eller flere) ut fra en mer omfattende logikk som tar hensyn til sammenhengen mellom flere egenskaper.

Fordelen ved å formulere klassifikatoren som et beslutningstre er bl.a. at beslutningsprosessen blir enklere å forstå. Man kan i mange tilfeller bygge opp en regelbasert beslutningsprosess, basert på kunnskap om problemet, som har mye til felles med ekspertsystemer, og forsåvidt også med maskesystemet som er i bruk i Tolletaten i dag. Det er også enklere å trene et slikt system manuelt ut fra forhåndskunnskap, enn f.eks. å trene et nevralt nett på mangedimensjonale inputdata.

Det eksisterer en rekke metoder for å trene beslutningstrær automatisk, se f.eks. [7] eller lærebøkene [3] og [5]. Problemet med disse metodene er at trærne ofte blir svært store, og spesialisert til treningssettet (overtrening). Det er utviklet teknikker for å beskjære ("prune") slike trær, for å holde størrelsen på et rimelig nivå, slik at problemet med overtrening også reduseres [8]. Trening av beslutningstrær kan forøvrig ofte være ustabil, ved at små endringer i treningsdataene kan føre til store variasjoner i trestrukturen. Metodene beskrevet i neste avsnitt har imidlertid vist seg å redusere ulempene ved slik ustabilitet, og alt i alt gi bedre klassifiseringsresultater.

3.3 Kombinerte klassifikatorer

Det finnes en rekke måter å kombinere klassifikatorer på. Hensikten er å oppnå et sikrere klassifiseringsresultat, enn det man er i stand til å oppnå med bare én enkelt klassifikator. I dette avsnittet blir det ganske kort gjennomgått ulike metoder som i senere år har vist seg å gi gode resultater i mange anvendelser. Disse er "Bagging", "Boosting", med "AdaBoost" som et spesielt eksempel, og til slutt "Random Forest".

3.3.1 Bagging

Prinsippet med "bagging" er å trene opp en rekke klassifikatorer på forskjellige utvalg av eksempler fra ett og samme treningssett. La oss si at det er n sampler i treningssettet. Av disse trekker man et tilfeldig utvalg på n' sampler der $n' < n$, og trener en klassifikator på dette utvalget. De utvalgte samplene legges tilbake i treningssettet, og prosessen gjentas et antall ganger; hver gang med en ny klassifikator som resultat. Selve komponentklassifikatoren kan være et beslutningstre, et nevralt nett eller en annen klassifikatortype. Vanligvis brukes samme type for alle komponenter. Det endelige klassifiseringsresultatet oppnås vanligvis ved en flertallsavgjørelse mellom resultatene fra komponentklassifikatorene.

3.3.2 Boosting

Boosting gjør det mulig å oppnå svært lav feilrate for den kombinerte klassifikatoren, selv om hver enkelt klassifikator ikke nødvendigvis er særlig god. Brukt på beslutningstrær har boosting som nevnt også vist seg å redusere problemene med overtrening. Prinsippet her er å trene opp et stort antall klassifikatorer (ikke nødvendigvis beslutningstrær), men disse skal være enkle. Slike "svake" klassifikatorer ("weak learners") har så lav kompleksitet at klassifiseringsnøyaktigheten bare er litt bedre enn å foreta et tilfeldig valg av klasse. Selv om klassifiseringsevnen er dårlig, vil imidlertid slike svake klassifikatorer ha god evne til å generalisere. Det vil si at de klassifiserer nye, ukjente data omtrent like godt som treningsdataene. Kombinasjonen av et stort antall slike svake klassifikatorer har vist seg å gi svært godt klassifiseringsresultat.

Boosting minner om bagging ved at man trener opp mange klassifikatorer på forskjellige utvalg av treningssettet. Måten utvalgene plukkes ut på er imidlertid forskjellig i de to metodene. I en variant av boosting trener man først en klassifikator, kall den C_1, på et tilfeldig utvalg, f.eks. ved å plukke ut n_1 sampler tilfeldig fra settet på n. Når klassifikatoren er trent på dette settet, anvendes den på et nytt utvalg, av treningssettet, og den trenger bare å gi såvidt bedre resultat enn det man får ved å trekke en tilfeldig klasse. C_1 er med andre ord en svak lærer. Neste utvalg, n_2, av treningssettet velges slik at klassifikator C_1 klassifiserer halvparten av disse riktig og den andre halvparten feil. En ny klassifikator C_2 trenes så på dette settet. Deretter kjører man klassifikatorene C_1 og C_2 på de resterende samplene og velger de samplene der klassifikatorene er uenige om resultatet. Ved å følge denne strategien trener man nye klassifikatorer på det "mest informative" utvalget av sampler med hensyn på de klassifikatorene man allerede har. Målet er å ende opp med et ensamble av komponentklassifikatorer, som ved flertallsavgjørelse gir svært lav feilrate.

3.3.3 AdaBoost

En mye brukt boosting-algoritme går under navnet "AdaBoost", fra *adaptive boosting*. Med denne metoden er det mulig å legge til et stort antall svake lærere inntil feilraten for den kombinerte klassifikatoren blir vilkårlig lav. Mange svake klassifikatorer utgjør til sammen én sterk klassifikator.

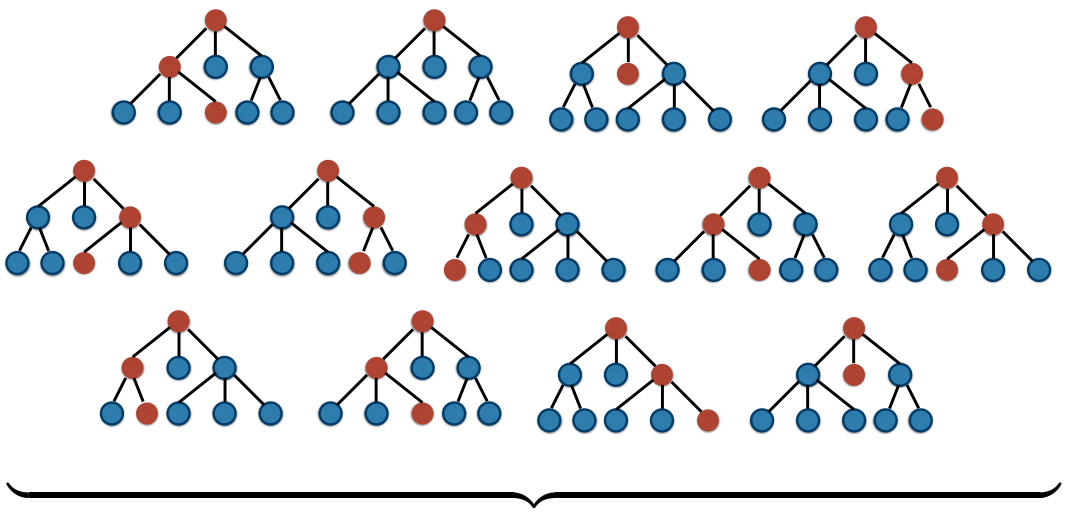
Sentralt i denne algoritmen er at hvert treningssample blir gitt en vekt, som styrer sannsynligheten for at samplet blir plukket ut for trening av en av komponentklassifikatorene. Hvis et gitt sample blir riktig klassifisert av den nåværende klassifikatoren, reduseres vekten og derved sannsynligheten

for at det skal bli med i treningen av den neste komponentklassifikatoren; i motsatt fall økes vekten. Man kan si at algoritmen konsentrerer oppmerksomheten om informative eller vanskelige sampler.

AdaBoost har bl.a. blitt brukt med gode resultater i ansiktsdeteksjon.

3.3.4 Random forest

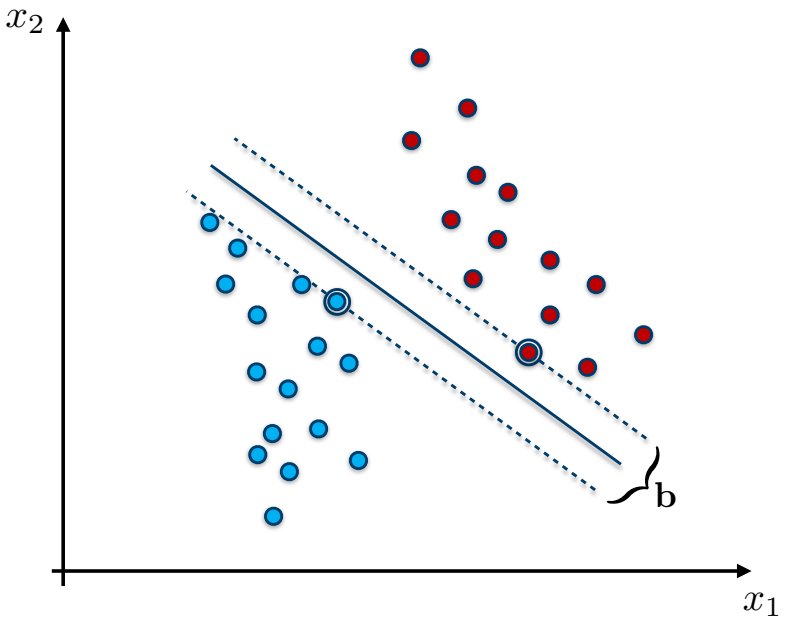
En nyvinning på området beslutningstrær er "random forest". Her genereres en "skog" av beslutningstrær, der hvert enkelt tre i skogen trenes opp på tilfeldige utvalg av egenskaper fra treningssettet; en fremgangsmåte som har vist seg å gi svært gode resultater i mange anvendelser. Random forest har mye til felles med boosting, men skiller seg ved at hver node i hvert enkelt tre i skogen bruker et tilfeldig utvalg av egenskaper.



Figur 3.5 *Illustrasjon av Random Forest. Mange forskjellige beslutningstrær tar imot samme input på toppen av treet. Klassifiseringsresultatet fra hvert tre (valget av blader nederst i trærne) kombineres ved f.eks. majoritetsavgjørelse eller midling, for å gi det endelige valget av klasse.*

3.4 Support Vector Machines

En "Support Vector Machine" (SVM) er i utgangspunktet en lineær klassifikator for å skille mellom to klasser [6]. Denne metoden ble introdusert på 1960-tallet. Grunnidéen består i å finne en desisjonsgrense for å skille datapunkter fra de to klassene perfekt. Dette forutsetter at klassene er såkalt lineært separable. Figur 3.6 viser et eksempel på et slikt separabelt problem for to klasser. Klassene kan her skilles ved hjelp av en desisjonsgrense (den heltrukne, rette linjen). SVM forsøker å finne en best mulig desisjonsgrense, dvs. et løsning med størst mulig "margin" slik at sannsynligheten for å feilklassifisere nye, ukjente objekter skal bli så liten som mulig. Marginen svarer til bredden



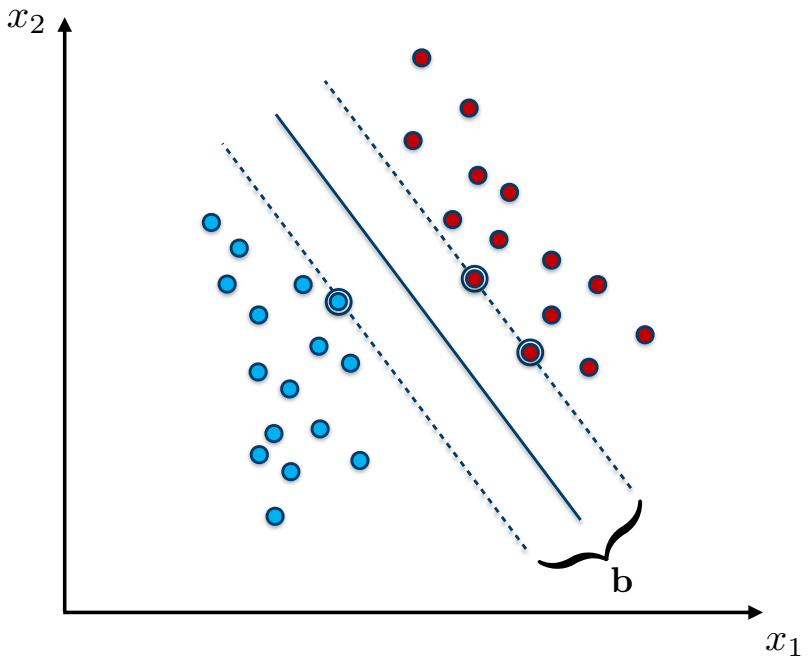
Figur 3.6 *Desisjonsgrense (heltrukken linje) for et lineært separabelt problem. Marginen b svarer til avstanden mellom de stiplede linjene, som markerer yttergrensene for en stripe i egenskapsrommet, som er fri for sampler. SVM-algoritmen finner en plassering og orientering til desisjonsgrensen, slik at stripen blir så bred som mulig, dvs. den prøver å finne en maksimal verdi på b).*

på stripen (avstanden b i figuren), avgrenset av de stiplede linjene som omgir desisjonsgrensen. Algoritmen tar utgangspunkt i såkalte "støttevektorer" (support vectors), dvs. sampler på hver side av desisjonsgrensen som kan brukes som forankringspunkter for desisjonsgrensen (se de innsirklede punktene i figurene). Målet er å finne en plassering og orientering av desisjonsgrensen slik at bredden på stripen (dvs. marginen b) kan gjøres så stor som mulig før den treffer støttevektorene. Et eksempel på en slik løsning er vist i figur 3.7.

Våre eksempler viser todimensjonale problemer, dvs. tilfeller der man bruker kun to egenskaper som input til klassifikatoren. I slike tilfeller vil en lineær klassifikator, som det er snakk om her, gi rette linjer som desisjonsgrenser. Med tre egenskaper vil desisjonsgrensen bli et plan, og i høyere dimensjon et såkalt "hyperplan", men matematisk sett er det uansett snakk om en "lineær" desisjonsgrense.

Som nevnt forutsetter SVM i sin opprinnelige form at datasettet er lineært separabelt. Det er imidlertid mulig å modifisere algoritmen ved å introdusere såkalte "slack" variabler, der man tillater et antall sampler innenfor marginen omkring desisjonsgrensen. Det gjøres altså en avveining av størrelsen på marginen, som man også her ønsker å gjøre så stor som mulig, uten at det går på bekostning av for mange uønskede sampler innenfor stripen omkring hyperplanet. Denne utvidelsen gjør det mulig å bruke SVM på det mer vanlige tilfellet av lineært ikke-separable data, men på bekostning av å måtte løse et mer innfløkt optimaliseringsproblem.

På 1990-tallet ble det introdusert en elegant videreutvikling av SVM som viste seg å bane vei for



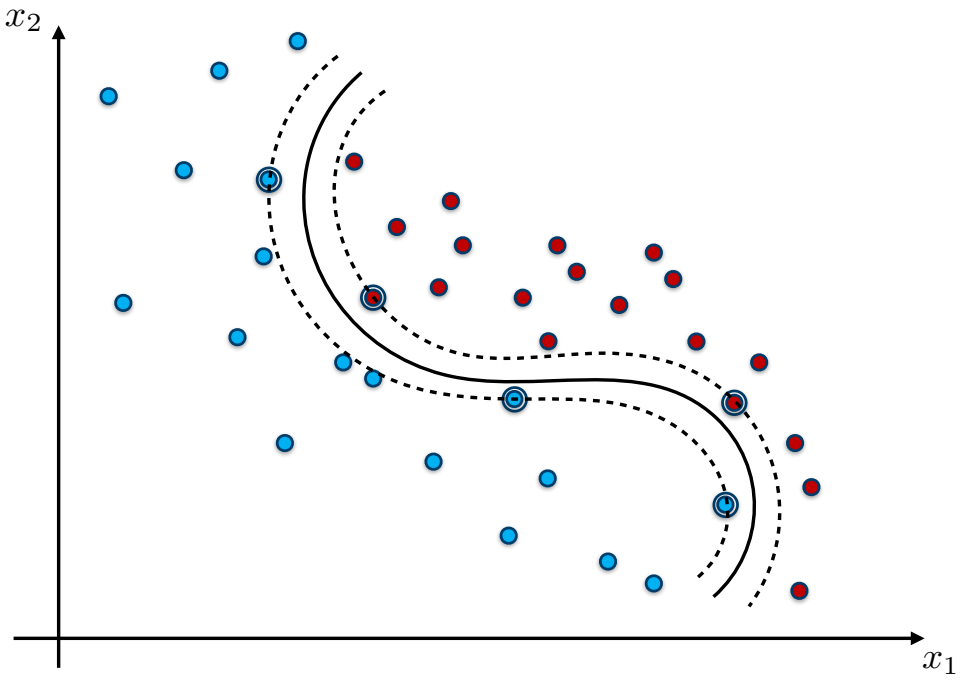
Figur 3.7 *Desisjonsgrense med størst mulig margin. Støttevektorene (innsirklede sampler) er "forankringspunkter" for den frie stripen på hver side av desisjonsgrensen (den heltrukne linjen i dette todimensjonale eksempelet). SVM-algoritmen søker å finne en løsning med størst mulig bredde på denne stripen.*

svært gode resultater. Nyvinningen bestod i å innføre en transformasjon av de opprinnelige egenskapsvektorene til punkter i et nytt egenskapsrom av mye høyere dimensjon. Dersom dimensjonen til det nye rommet er høy nok, er det mulig å transformere datasettet til et lineært separabelt sett. Treningen kan derved foretas i det nye rommet. Figur 3.8 viser et todimensjonalt eksempel med sampler fra to klasser som ikke er lineært separable, med ikke-lineær desisjonsgrense og tilhørende støttevektorer og margin. Her er løsningen funnet i det høyeredimensjonale rommet, der klassene er lineært separable, og deretter projisert tilbake til det opprinnelige egenskapsrommet.

3.5 Nevrale nett

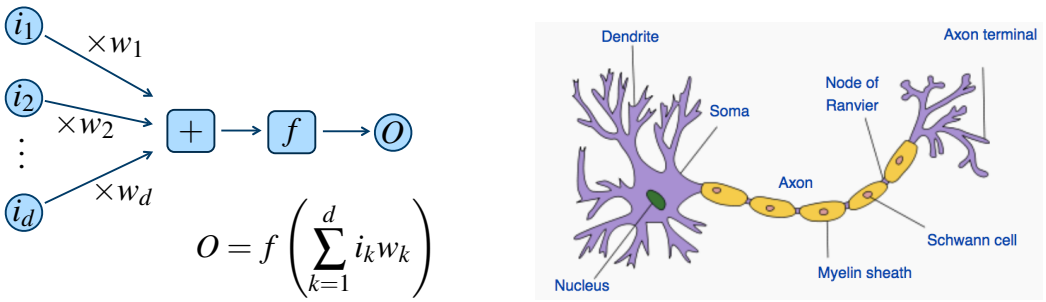
3.5.1 Bakgrunn

Nevrale nett – eller rettere sagt *kunstige* nevrale nett – er en samling av kunstige nevroner, som er enkle modeller av biologiske nerveceller. Figur 3.9 viser et slikt kunstig nevron og en skisse av et biologisk nevron. Kunstige nevroner ble introdusert av Frank Rosenblatt i 1957 [4]. I likhet med biologiske nevroner, tar det kunstige nevronet inn signaler fra én eller flere naboceller, vekter signalene og summerer dem til et nytt signal. Dette signalet sendes så gjennom en aktiveringsfunksjon for å produsere utsignalet, som i sin tur går videre til nye nevroner i nettverket. Dette er inspirert av biologiske nevroner som sender en impuls videre, bare dersom signalstyrken er over et visst nivå. Figur 3.10 viser et enkelt nevralt nett med inputlag, outputlag og et skjult lag. Nodene i det skjulte



Figur 3.8 *Desisjonsgrense, støttevektorer og tilhørende margin for lineært ikke-separable data.*

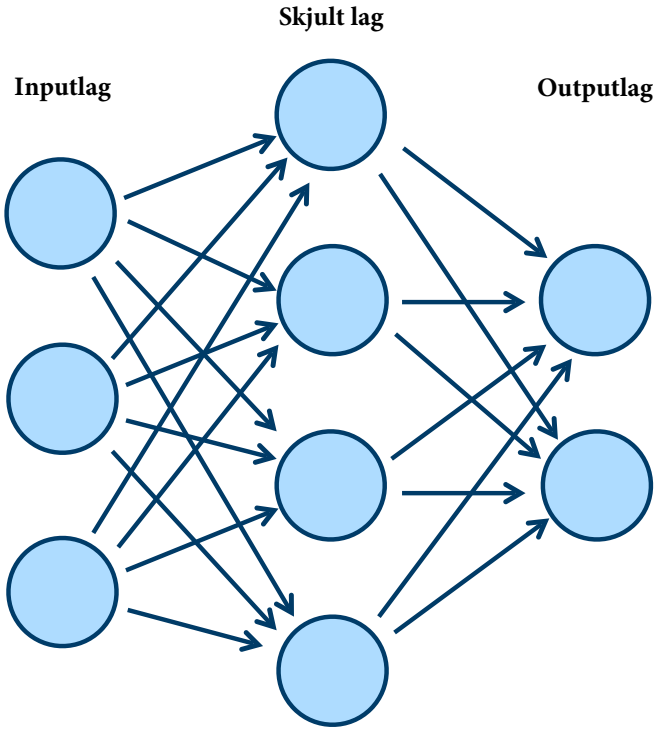
laget tar imot informasjon fra alle inputnodene og sender resultatet til alle noder i outputlaget. Hver av de skjulte nodene tilsvarer én enkelt perceptron. Et typisk nevralt nett kan ha flere skjulte lag.



Figur 3.9 *Illustrasjon av Rosenblatts Perceptron. Inputverdiene $i_1, \ldots, i_d$ til venstre blir multiplisert med vekter $w_1, \ldots, w_d$, summert og sendt gjennom aktiveringsfunksjonen f for å gi outputverdien O. I det biologisk nevronet (t.h.) kommer de elektriske impulsene fra nabocellene inn på forgreiningene til venstre, og ved tilstrekkelig høy aktivitet genererer cellen nye impulser som sendes til andre naboceller via terminalene til høyre (Quasar Jarosz, English Wikipedia).*

Rosenblatts Mark I perceptron[1] var en implementasjon av perceptron-algoritmen i maskinvare. Maskinen var koblet til et kamera som registrerte bilder med en matrise av 20 x 20 fotoceller, og kunne læres opp til å gjenkjenne objekter av en gitt geometrisk form. Maskinlæringen foregikk ved å justere på potensiometrene. Dette svarte til å justere vektene på inputsiden i perceptronen. Dette ble foretatt automatisk ved hjelp av elektromotorer koblet til potensiometrene, hver gang bildet ble klassifisert feil.

[1] Wikipedia: Perceptron



Figur 3.10 *Illustrasjon av et konvensjonelt nevralt nett med ett skjult lag.*

Perceptron-algoritmen har helt siden introduksjonen vært en av standardmetodene i for trening av klassifikatorer, men da hovedsakelig implementert i programvare, og kan som nevnt generaliseres til å håndtere problemer med mange klasser. I utgangspunktet leter algoritmen etter en perfekt løsning på lineært separable problemer, men finner uansett en best mulig løsning også på ikke-separable problemer. For å kunne håndtere mer kompliserte problemer, introduserte man senere såkalte flerlags perceptroner (multilayer perceptrons). Nevrale nett er en videreutvikling av dette konseptet, og har siden introduksjonen på 1970-tallet vært en mye brukt klassifikatortype.

I et konvensjonelt ferdigtrent nevralt nett (se illustrasjonen til venstre i figur 3.10) kommer ukjente data inn via nodene i inputlaget, bearbeides og sendes videre til alle nodene i neste lag. Resultatene av beregningene underveis (gjentatt vekting, summering og transformasjon via aktiveringsfunksjoner) ender opp i nodene i outputlaget. For typiske klassifiseringsproblemer er det vanlig å ha en outputnode for hver klasse, og maskinlæringsprosessen settes da opp slik at den ønskede klassen (for gitt input) kommer ut med den største tallverdien, mens helt feilaktige klasser vil ha svært lav verdi. Dette er et eksempel på et foroverrettet (feed forward) nevralt nett, der informasjonen bare går i én retning, fra input til output. Det finnes også andre typer nett med tilbakekoblinger (rekursive nett), men vi kommer ikke inn på dette her.

3.5.2 Trening av nevrale nett

I likhet med tradisjonelle klassifikatorer, og de andre typene beskrevet tidligere i dette kapitlet, trenes også nevrale nett ved ledet læring. Det er helt avgjørende for et vellykket resultat at man har

et størst mulig treningssett tilgjengelig, der samplene (eksemplene fra de ulike klassene) er merket med klassetilhørighet. Grunnprinsippet i denne treningsprosessen er at man sender sampler fra gitt klasse inn på inputlaget, og deretter sammenligner resultatet i outputlaget med den ønskede responsen for vedkommende klasse. Dersom resultatet er feil, må man gå tilbake og justere vektene i nettverket inntil ønsket resultat oppnås (i likhet med justeringen av potensiometrene i Rosenblatts Mark I Perceptron). Dette må så gjøres for alle sampler i treningssettet.

En slik trening av nevrale nett gjøres vanligvis ved såkalt "back-propagation". Dette er en algoritme som tar utgangspunkt i en feilfunksjon; et mål på avviket mellom den faktiske responsen til nettverket, og den ønskede responsen for et sample av gitt klasse. Ved å gjennomføre et såkalt "gradientsøk", en systematisk justering av vektene i det nevrale nettet i retning mot lavere verdi på feilfunksjonen, vil man (forhåpentligvis) etter mange iterasjoner (mange gjennomløp av treningssettet) nå frem til et ferdigtrent nettverk med så lav feilrate som det er mulig å oppnå på det aktuelle treningssettet.

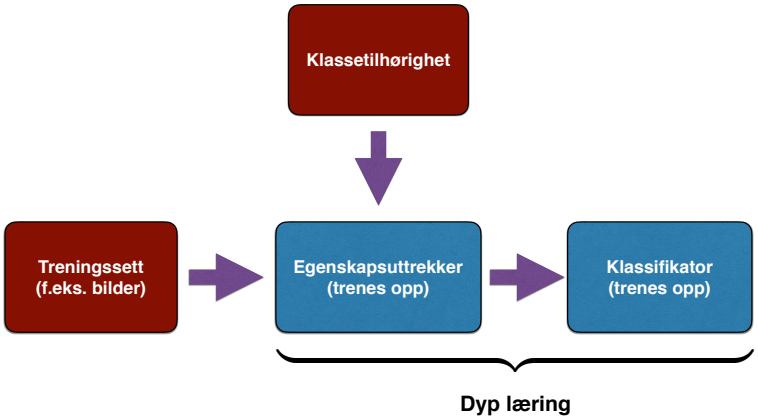
Et problem man må være oppmerksom på i forbindelse med nevrale nett, er faren for overtrening. Et stort nett, med flere skjulte lag, vil ha stor fleksibilitet til å tilpasse seg kompliserte fordelinger for de ulike klassene man ønsker å skille, men samtidig mange vekter som skal trenes opp. Opptrening av et robust nevralt nett, dvs. et nett som vil fungere godt på nye data, vil kreve et mye større treningssett for samme dimensjon på egenskapsrommet (antall inputnoder) enn for enkle, tradisjonelle klassifikatorer. Av den grunn falt nevrale nett i mange år i bakgrunnen, til fordel for bl.a. SVM, men dette bildet har blitt dramatisk endret i løpet av de siste årene. Dette leder over i temaet "dyp læring" som behandles i neste kapittel.

4 Dyp læring

I tiden etter 2010 har store nevrale nett blitt en mer og mer dominerende teknologisk base for maskinlæring [9]. Milepæler i den siste tiden har for eksempel vært at nevrale nett nå brukes for å spille go (komplisert japansk brettspill) på toppnivå. De siste ukene (høsten 2017) er det også trent opp et dypt nett som slår ut de beste sjakkprogrammene i verden [10]. Videre slår nå nevrale nett mennesker i en rekke komplekse bildegjenkjenningsoppgaver.

4.1 Grunnleggende teori

Hovedforskjellen mellom tradisjonell maskinlæring og dyp læring er at i det siste tilfellet blir også egenskapsuttrekkingen en del av treningsprosessen. Dette er illustrert i 4.1. Et dypt nett for klassifisering av bilder består typisk av et antall "konvolusjonslag", med en tradisjonell klassifikator (typisk et nevralt nett) på toppen, som vist i figur 4.2. Man kan godt si at konvolusjonslagene til sammen utgjør egenskapsuttrekkeren i et tradisjonelt klassifiseringssystem, med den forskjellen at disse lagene også blir trent ved å utnytte klassetilhørigheten til objekter i treningsbildene.

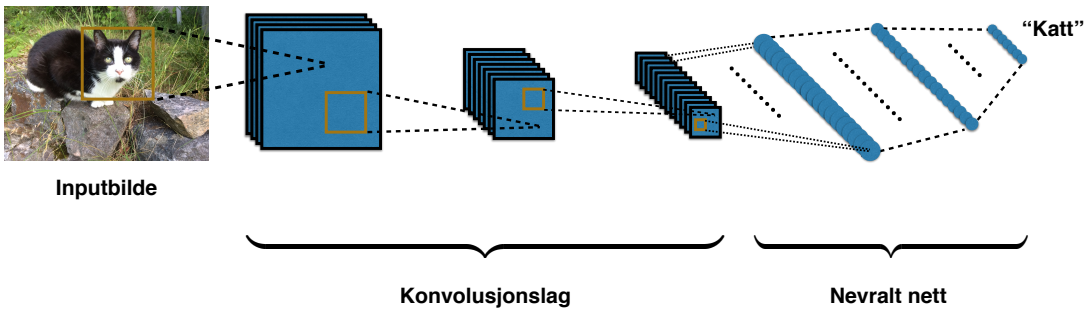


Figur 4.1 *Illustrasjon av dyp læring. En database av f.eks. bilder, sammen med klassetilhørigheten, er input til treningen av et dypt nett. De dypere lagene i nettet er det som i tradisjonell mønstergjenkjenning kalles egenskapsuttrekkeren (og som er laget for hånd, ref. figur 2.2), mens de høyere lagene utgjør klassifikatoren. I dyp læring trenes både egenskapsuttrekkeren og klassifikatoren.*

Den underliggende teknologien omtales gjerne som dyp læring (deep learning), der dyp refererer til antall såkalte lag i de nevrale nettene som benyttes. I praksis kan man si at dyp læring er det fagfeltet som arbeider med store og komplekse nevrale nett for å løse maksinlæringsoppgaver. Fordelen ved bruk av dyp læring for slike anvendelser er den svært gode ytelsen som disse systemene ofte oppnår. Videre er analysen typisk rask slik at en høy analyserate kan opprettholdes.

Den primære ulempen er at disse systemene må trenes med data. For at dette skal virke tilfredsstillende kreves det ofte svært store datamengder. Dette problemet forsterkes ved at dataene ofte må sorteres manuelt for å indikere for systemene for eksempel hva som er et funn og hva som ikke er det. Det er

viktig å påpeke at dette er en rent praktisk utfordring knyttet til bruken av disse metodene, ikke en svakhet som påvirker ytelse.



Figur 4.2 *Illustrasjon av et konvolusjonsnett for bildegjenkjenning.*

La oss nå beskrive gangen i et konvolusjonsnett, med utgangspunkt i figur 4.2. Inputbilde sendes inn på det første laget (representert ved stabelen av blå kvadrater til venstre i figuren). Den grunnleggende operasjonen her er en konvolusjon (folding) mellom bildet og et digitalt filter. Filteret opererer på et lokalt område i originalbildet, som vist i figuren, og beregner verdien til et nytt piksel i ett av bildene i det første laget (såkalte feature-maps). La oss si at det nye bildet er det fremste av de blå kvadratene i stabelen. Filteret kjøres over hele bildet, slik at kvadratet fylles ut med nye pikselverdier, som inneholder informasjon fra ulike deler av originalbildet.

Filteret kan være konstruert slik at det fremhever f.eks. vertikale eller horisontale kanter i originalbildet. Det dannes altså en filtrert versjon av originalbildet, der nettopp slike karakteristika er fremhevet. Konvolusjonslaget inneholder imidlertid en stabel av filtere, som hver for seg fremhever forskjellige karakteristika, og bygger opp en stabel av nye bilder (de blå kvadratene) der forskjellige bildeegenskaper er fremhevet. Denne stabelen sendes så videre til neste konvolusjonslag, der et nytt sett av filtere opererer på bildene i foregående lag. Prosessen fortsetter gjennom alle konvolusjonslagene, til den når inputlaget i det konvensjonelle nevrale nettet på toppen (til høyre i figuren).

Trening av et slikt nett innebærer at man må justere inn filterverdiene i alle filtere i alle konvolusjonslagene. Dette betyr at det er et svært stort antall parametre som må tilpasses, med behov for et enormt treningssett for å unngå overtrening av nettet. Noe av grunnen til at dyp læring overhode er realiserbart er de forenklinger som gjøres ved å gå fra et konvensjonelt nevralt nett, der det vil være koblinger mellom alle piksler i inputbildet og alle noder i neste lag (*fully connected network*), til et bl.a. konvolusjonsnett. I et konvolusjonsnett vil det bare være *lokale* koblinger mellom et bilde i ett lag og det nye pikselet i neste lag. Dette alene gir en betydelig reduksjon i antall vekter som må trenes. Videre brukes det samme filteret over hele bildet, med en enorm reduksjon i antall vekter.

Alt i alt viser det seg at dyp læring er mulig såfremt man har tilstrekkelig prosessorkraft. Videre forutsettes det at det legges arbeid i datainnsamling og merking av et mye større treningssett enn det som har vært vanlig å bruke i tradisjonell maskinlæring. Det har også vist seg at man, overaskende nok, har fått gode resultater med langt mindre treningsdata enn forventet. Det er i mange anvendelser også oppnådd gode resultater ved å starte med ferdigtrenede nett, og bare gjøre en finjustering av de øverste lagene i nettet ved hjelp av et moderat antall egne treningsbilder. For slik "tuning" av eksisterende nett kan man også endre outputlaget til å gi ut de klassene man selv ønsker.

4.2 Resultater innen relevante problemstillinger

Dyp læring har i de seneste årene gitt betydelige fremskritt innen talegjenkjenning, deteksjon og gjenkjenning av objekter i bilder, lesing av håndskreven tekst, deteksjon av narkotika og medikamenter, genanalyse og mye mer. Et område der dyp læring har gitt spesielt gode resultater er automatisk språkforståelse og automatisk oversetting fra ett språk til et annet.

Et av de store områdene er maskinsyn; det å lære maskiner å forstå verden omkring seg. Dette har mange anvendelser innen bl.a. robotikk. Her kan nevnes selvkjørende biler, som ved hjelp av kameraer og andre sensorer må lære seg å forstå omgivelsene. Det er utviklet dype nett som forstår typiske byscener, og kan skille mellom klasser av typen gate, fortau, bygninger, biler, fotgjengere osv. Det arbeides også med dyp læring for karakterisering av mer generelle (ikke-bymessige) scener. En rekke medisinske anvendelser kan trekkes frem, bl.a. roboter som kan foreta kirurgiske inngrep, der bilder fra kameraer brukes til å styre roboten. Et fellestrekk ved slike anvendelser er deteksjon og gjenkjenning av objekter i bilder.

I de siste årene er det trent opp dype nett på store bildedatabaser, for å utføre slike oppgaver. Disse nettene har vist seg å slå tradisjonelle metoder med stor margin. Et eksempel er automatisk lesing av registreringsnummer på biler (ANPR), der dyp læring har gitt stor gevinst. Et annet område er ansiktsgjenkjenning, der man også har høstet betydelig gevinst. Slike dype nett brukes nå i nyere mobiltelefoner og bilderedigeringsprogrammer for hjemmebruk (husk at selv om treningen av et dypt nett typisk krever lang tid og store maskinvareressurser, vil eksekveringen av nettet på et inputbilde gå svært hurtig, selv med liten prosessorkraft).

Automatisk talegjenkjenning har også gjort store fremskritt de siste årene, ved hjelp av dyp læring. Som eksempel kan nevnes “Siri” fra Apple. Her er mange av forbedringene siden 2014, oppnådd gjennom bruk av dyp læring. Store forbedringer har også vært oppnådd innen språkoversetting. Oversetting kan også være bildebasert, ved lesing av trykt eller håndskrevet tekst. Lesing av håndskrift har forøvrig også gjort store fremskritt vha dyp læring. Tolking av tekst vha dype nett brukes også i såkalte “chat bots”.

En medisinsk anvendelse av relevans for Tolletaten er forøvrig analyse av termiske bilder. Tollmyndigheter i andre land har vist interesse for dette, for å kunne detektere f.eks. flypassasjerer med feber; dette for å kunne forhindre spredning av farlige sykdommer. En annen mulighet her, kan være å detektere en persons sinnsstemning ved termisk deteksjon av hjertefrekvens, noe som kan indikere nervøsitet fordi vedkommende er involvert i smugling eller annen kriminell aktivitet.

Man regner forøvrig med at dyp læring og dype nett vil kunne gi vesentlige forbedringer innen analyse av store datamengder, for å avdekke sammenhenger og mønstere som er vanskelig å fange opp med tradisjonelle metoder.

4.3 Eksempler på bruk innen Tolletaten

Den viktigste anvendelsen av dyp læring for Tolletaten synes å være automatisk analyse av røntgenbilder. Røntgen blir brukt i stor grad i dag, til inspeksjon av pakker og brev og skanning av biler og containere. I noen grad foretas også røntgenundersøkelse av personer for å avdekke bl.a. narkotika i mage og tarm. Bruk av røntgen er i utgangspunktet tidsbesparende, fordi det gjør

det mulig å luke ut pakker/biler/containere der bildene tyder på at det er lite å finne ved en fysisk inspeksjon.

Spesielt containertrafikken er et stort problem for tollmyndigheter i alle land, og det er i praksis bare mulig å fysisk inspisere en svært liten del av alle containere som passerer grensen. En manuell undersøkelse av en container innebærer full utpakking av alt innhold for å være sikker på å finne eventuelle ulovlige varer skjult blant lovlig gods, med etterfølgende pakking av containeren. Dette er svært kostbart og tidkrevende, og skaper forsinkelser i vareflyten. Røntgenskanning gir mulighet til å undersøke flere containere, men visuell inspeksjon av røntgenbildene er også tidkrevende og beheftet med usikkerhet. Ulovlige gjenstander kan være helt eller delvis skjult av andre objekter og være vanskelige å oppdage på grunn av dårlig kontrast og mye støy i bildene.

Det synes ikke å ha vært gjort mye vitenskapelig arbeid innen automatisk analyse av røntgenbilder for slike formål, men [11] beskriver et rammeverk for bruk av moderne maskinlæring i denne sammenhengen. Her har man sett på røntgeninspeksjon av biler som fraktes på jernbanevogner, og har spesielt sett på deteksjon av små, metalliske objekter ved hjelp av konvolusjonsnett. Selv om artikkelen konkluderer med at det gjenstår mye arbeid, er resultatene såpass lovende at temaet er verdt å følge opp.

Det er fullt mulig å starte utviklingen av dype nevrale nett for røntgendeteksjon av ulovlig eller tollbelagt innhold i alle typer sendinger, så snart et tilstrekkelig stort bildemateriale er tilgjengelig. Selve treningen krever prosessorkraft og regnetid, men i den store sammenheng er både kostnader og innsats av beskjeden størrelse, og ikke til hinder for å sette i gang en slik aktivitet. I postmottaksstudien [2] er det beskrevet et eksempel på deteksjon av pistoler i røntgenbilder av bagasje med et dypt nett som er trent på et lite antall (44) håndmerkede bilder. Her brukes en variant av et konvolusjonsnett (et såkalt U-nett). I tillegg til å varsle om at det er en pistol i bagasjen, returneres et bilde som viser omrisset til pistolen og hvor den ligger i bagasjen. Det foretas en såkalt "semantisk segmentering", der man skiller objekter av interesse fra resten av bildet. Dette er bare et enkelt eksempel på hva som er mulig å oppnå med beskjedne midler. Man kan tenke seg å bygge opp et bibliotek av objekter som det er ønskelig å detektere.

Selv om slike eksperimenter viser forbløffende gode resultater, er det viktig å understreke at maskinlæring, og ikke minst dyp læring, krever størst mulig treningssett med merkede data. Bare slik er det mulig å utføre ledet læring. For røntgenbilder kan denne merkingen bestå i at det legges masker over bildet, som viser omrisset til objekter av spesifikk klassetilhørighet. Datainnsamling og merking av data blir behandlet i kapittel 5.

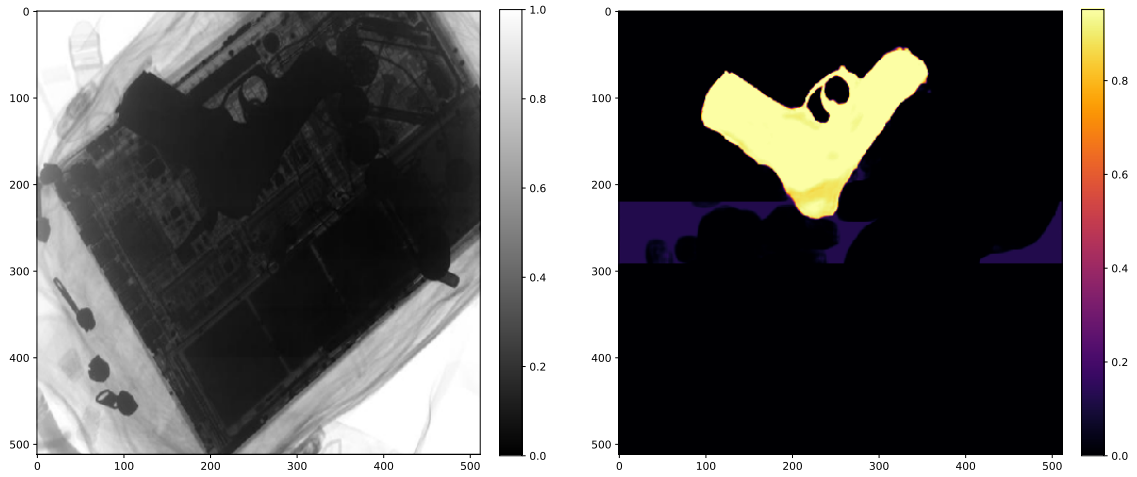
5 Innsamling og bearbeiding av data

Behovet for treningsdata har blitt nevnt flere ganger i denne rapporten. Slike data er nødvendig for å kunne la en maskin lære av erfaring. Tolletaten sitter i dag på mye data, både bildedata av forskjellig type og andre typer data, men denne informasjonen er spredd på ulike systemer, i mange forskjellige formater. Det er viktig at datamaterialet blir tatt vare på og organisert i en felles database, som gjør det enkelt å hente ut treningsdata for å drive kontinuerlig maskinlæring (finjustering av eksisterende klassifikatorer) etter hvert som nye data lagres i databasen.

5.1 Merking av data

Den mest arbeidskrevende delen av datainnsamlingen vil sannsynligvis være merkingen av materialet, basert på de funn som blir gjort. En del av denne jobben blir allerede gjort i dag. Det blir skrevet rapporter hver gang det gjøres funn. Problemet er å få knyttet informasjonen til rådataene som sensorsystemene har samlet inn. Her må det lages rutiner/tekniske løsninger som gjør at denne prosessen går mer eller mindre automatisk.

La oss igjen ta røntgenbilder som eksempel. Hvis det gjøres et funn i en pakke, lastebil eller container som er skannet, er det viktig ikke bare å registrere at det er gjort et funn, og hva som er funnet, men også hvor i bildet gjenstanden befinner seg. Det ideélle hadde vært å markere omrisset til objektet eller legge en maske over objektet, som vist i eksempelet i figur 5.1. Les mer om pistoleksemplet i [2]. Samme metodikk kan brukes for andre typer bilder (synlig lys, termisk, terahertz, hyperspektral avbildning).



Figur 5.1 *Røntgenbilde hentet fra GDXray-datasettet (se `http://dmery.ing.puc.cl/index.php/material/gdxray/`) som viser en pistol sammen med korresponderende maske som viser hvor pistolen befinner seg. Masken er i dette bildet generert automatisk av et nevralt nett.*

5.2 Metadata/kontekst

Med metadata menes i denne sammenhengen all mulig informasjon knyttet til en sending eller varetransport. Dette kan typisk være informasjon fra tollmanifestet: varebeskrivelse, avsender, avsenderland, mottaker, mottakers adresse, pris på sendingen, tidspunkt for ankomst osv. Det er viktig at slik vareflytsinformasjon også knyttes til en sending, slik at den kan inngå i treningssettet. Den sammenhengen (konteksten) en observasjon er gjort i kan bidra til å styrke eller redusere sannsynligheten for at en pakke det er tatt et røntgenbilde av, virkelig inneholder smuglergods. Dette kan gi økt sannsynlighet for å stoppe smuglergods, og redusert behov for unødvendige fysiske inspeksjoner.

Vareflytsdata kan i seg selv også gjøres til gjenstand for maskinlæring. Analyse av vareflyten over tid kan avdekke avvik fra normale forhold (ref. Pattern-of-life), og gi grunnlag for økt kontroll på gitt sted og tid. Ved å koble slike data med faktiske funn (hva, hvor og når), kan det være mulig å avdekke andre sammenhenger enn kun avvik fra det normale.

5.3 Forenklet prosess for datainnsamling

Data som skal brukes for å vurdere om et objekt inneholder ulovlige varer må samles inn på en måte som sikrer god kvalitet, dvs at metoden for innhenting av data (sensormåling) er veldefinert og gir verdier innenfor et kjent område og med kjente begrensninger (feil).

Vi skisserer nedenfor en prosess for datainnsamling som følger en typisk arbeidsflyt for kontroll av objekter. Poenget med prosessen er å gjøre datainnsamlingen enkel slik at den krever få manuelle grep og ikke forstyrrer arbeidsflyten. Denne skissen er et utgangspunkt for prosessdesign og må tilpasses og optimaliseres for den konkrete arbeidsflyten.

Vi tenker oss at hvert objekt som velges ut for kontroll, registreres i et IT-system og gis en unik ID. For at ID-en skal kunne forbindes med det fysiske objektet, kan det være aktuelt å sette et ID-merke på det, eller lagre et fotografi av det, slik at det kan gjenkjennes. Objektets lagrede ID gir grunnlag for å knytte sammen alle tilhørende data. Dette kan være egne sensordata eller ulike hendelser som inntreffer før, under og etter kontrollen. Alle data som genereres, knyttes automatisk til objektets ID og lagres. Merking med ID og eventuelt forsegling av objektet kan være nødvendig for videre behandling etter kontroll.

Skisse til delvis automatisert prosess for datainnsamling:

- Når et objekt velges ut for kontroll, genereres det automatisk en sak med unik ID i IT-systemet. Grunnlagsdata som dato og saksbehandler registreres også automatisk.
- Begrunnelse for utvelgelse registreres.
- Automatisk innhenting av digital forhistorie som reiserute, transportmiddel mm. Knyttes til objektets ID.
- Automatisk fotografering med synliglyskamera.
- Automatisk / halvautomatisk røntgenskanning.
- Manuell kontroll av objekt. Kan inneholde flere dataelementer:
  - Halvautomatisk fotografering av innhold, én eller flere ganger.

  - Manuell datainnhenting av supplerende sensorer.
  - Manuell innføring av positivt resultat fra kontroll, type funn, foreløpig konklusjon. Operasjonen bør helst kunne gjøres uten å gå bort fra arbeidsstasjonen og uten bruk av tastatur.
- Halvautomatisk/manuell innføring av negativt resultat. Kan for eksempel skje ved at objektet lukkes og merkes som ''åpnet av tolletaten" og føres under et kamera som dokumenterer at objektet er lukket. Merkingen kan knyttes til objektets ID.
- Automatisk lagring og lukking av sak og videresending for videre saksbehandling eller påtale, for eksempel ved å føre objektet mot stedet hvor det skal lagres fysisk.

Sensordata som samles inn i denne prosessen har to anvendelsesområder:

- Automatisk dataanalyse i sanntid, for å foreslå type funn. Eksempelvis mønstergjenkjenning i bildedata.
- Lagrede sensordata kan benyttes til maskinlæring, ved at de er knyttet til informasjon om funn/ikke funn (se avsnitt 2.2 om ledet læring).

5.4 Tilgjengelig programvare

Det finnes i dag mye tilgjengelig programvare for mønstergjenkjenning og maskinlæring, både kommersielle systemer og gratisprogramvare (open source). Slike systemer dekker alt fra klassiske metoder, via nyere teknikker, til de siste metodene innen feltet dyp læring. For noen få tiår siden hadde man ikke slike hjelpemidler tilgjengelig, og måtte programmere bl.a. maskinlæringsalgoritmer fra bunnen av, slik en av forfatterne av denne rapporten gjorde på 1980-tallet. Situasjonen i dag er en helt annen. Eksempler på tilgjengelige verktøy er:

- Matlab; generelt programsystem for matematiske problemstillinger. Inneholder diverse verktøykasser for bl.a. nevrale nett, maskinlæring og dyp læring, bildeanalyse og maskinsyn.
- OpenCV; programbibliotek (gratis), hovedsakelig for bildeanalyse og maskinsyn, men inneholder også biblioteker for maskinlæring.
- Python; generelt programmeringsspråk, men distribusjonene (gratis) inneholder diverse numeriske pakker. Python er antakelig det mest brukte programmeringsspråket for dyp læring.
- PRTools; separat (gratis) Matlab-verktøykasse for mønstergjenkjenning og maskinlæring.
- perClass; kommersiell Matlab-verktøykasse for mønstergjenkjenning og maskinlæring, basert i stor grad på PRTools
- TensorFlow; numerisk bibliotek spesielt rettet mot dyp læring (gratis).
- Caffe; rammeverk for dyp læring (gratis). Inneholder bl.a. ferdigtrente nett som kan brukes som et utgangspunkt for å fintrene egne nett på mer spesifikke problemstillinger.

Bruk av slike standardiserte pakker muliggjør rask utvikling av programvare for egne problemstillinger.

5.5 Innhenting av data fra tilfeldige utvalg

Som beskrevet i postmottaksstudien [2] handler randomiserte tester om å gjøre et tilfeldig utvalg av pakker, biler, containere mm. for nærmere inspeksjon. I et postmottak kan man f.eks. plukke ut hver n-te pakke på samlebåndet for nærmere kontroll, uavhengig av etterretningsinformasjon eller hva det automatiske systemet måtte ha funnet. Dette tjener to formål.

For det første får man gjennom slike randomiserte tester muligheten til å kartlegge den faktiske smuglerhyppigheten. Det er en kjent sak at smugling av bl.a. narkotika har et mye større omfang enn beslag som gjøres på grensen skulle tyde på. Slike beslag gjøres som oftest på bakgrunn av etterretningsinformasjon, tilslag på masker og "mageføelse", men man har ingen oversikt over hvor mye som slipper forbi. Forbruket av narkotika i Norge tyder på at dette er ganske mye. Avsnitt 5.6 beskriver et enkelt eksempel på hvordan man kan komme frem til statistikken ved tilfeldig utvalg over tid.

Det andre formålet er å skaffe et mest mulg realistisk/balansert treningssett. Som påpekt i [2] er ikke maskinene bedre enn man gjør dem; hvis de først læres opp til å finne piller, og velger ut pakker med piller til kontroll, så er det fare for at man stirrer seg blind på piller og ikke finner noe annet. Ved å tvinge systemet til å velge ut et *representativt* utvalg fra pakkestrømmen, får man et bedre grunnlag for maskinlæringen. På sikt vil systemet bli bedre til å skille mellom en rekke forskjellige smuglervarer, og ikke minst bli bedre til å skille smuglervarer fra alt mulig annet. Over tid vil man også kunne finne mer avanserte mønstre i varestrømmen. For at maskinene skal kunne lære hvilke pakker som inneholder noe ulovlig og hvilke som ikke gjør det, må pakkene kontrolleres, og resultatene registreres. Det forutsettes da at eventuelt smuglergods blir avdekket ved en kontroll.

I det følgende skal vi se hvordan det er mulig å anslå volumet av smugling basert på tilfeldig utvalg, dvs. at man her stopper biler helt tilfeldig, uten hensyn til annen informasjon. Dette kan gjøres ved bruk av en randomgenerator på en datamaskin. I utgangspunktet er antall smuglere pr. time ukjent, og vi skal finne ut om det er mulig å estimere dette antallet ved hjelp av tilfeldige kontroller. I eksemplet nedenfor tar vi utgangspunkt i at hvis en smugler blir stoppet, så blir vedkommende tatt. Selvsagt vil ikke dette alltid være tilfelle, men uansett kan vi ikke vite noe som helst om de smuglere som er så dyktige (eller heldige) at de unngår å bli avslørt under en fysisk kontroll av bilen.

5.6 Eksempel på beregning av smuglerrate

Et visst antall biler som passerer en av de mange grensestasjonene på vei inn i Norge, vil bringe med seg smuglergods. Dette kan være alt fra litt mer enn tillatt kvote av kjøttvarer, til et større parti narkotika. Tolletaten overvåker grensestasjonene og kontrollerer visse, utvalgte biler. Målsettingen med dette er å håndheve norske lover og regler for import av varer til landet. For å kunne oppnå denne målsettingen er det viktig å kartlegge *omfanget* av brudd på tollbestemmelsene, dvs. hvor mange biler som har med seg ulovlige eller ufortollede varer over grensen. Hensikten med dette eksempelet er å vise hvordan en slik kartlegging kan oppnås gjennom tilfeldig utvalg av biler for kontroll på grensen. I vårt eksempel tar vi utgangspunkt i biler som passerer inn i Norge over Svinesund.

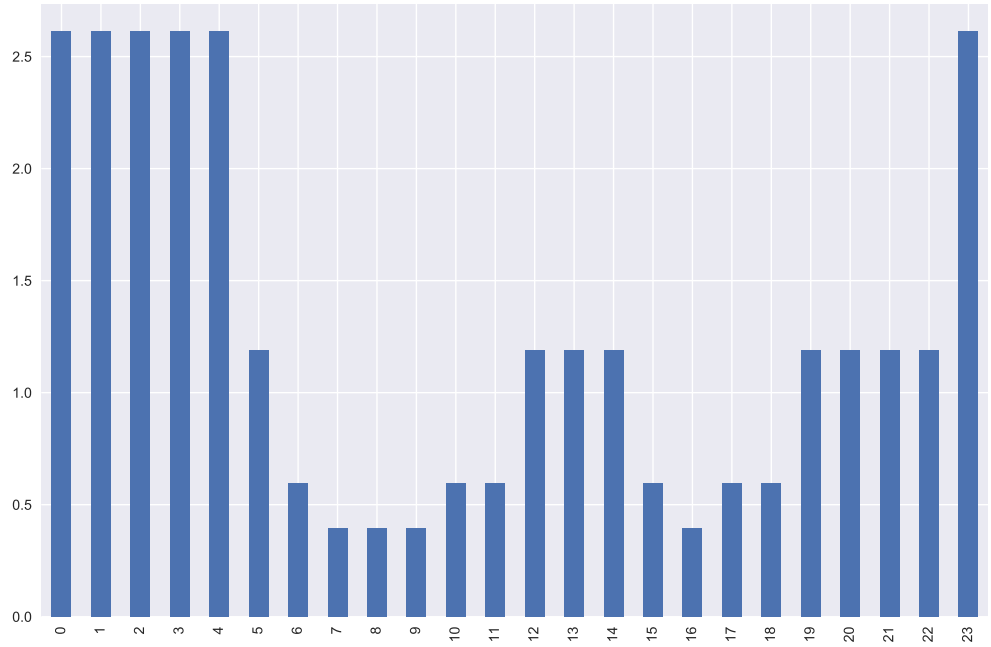
5.6.1 Innledende antagelser

Vi gjør først noe antagelser om biltrafikken, den faktiske smuglerraten og antallet kontroller, for å kunne sette opp en simulering over hvor lang tid man trenger for å beregne et estimat for smuglerraten.

Vi gjør en antagelse om 16 000 biler i døgnet. Hyppigheten vil variere med tid på døgnet. Her skal vi anta at antall biler varierer fra time til time gjennom døgnet, med flest biler i rushtiden morgen og ettermiddag. For eksempelets skyld gjør vi også en forutsetning om hyppigheten av smugling. I dette eksempelet skal vi anta en smuglerrate på 1 % på døgnbasis, dvs. at det totalt passerer 160 biler med smuglergods i løpet av et døgn. Videre antar vi at antallet biler med smuglergods pr. time er dobbelt så høyt om natten (mellom kl. 23:00 og 05:00) som om dagen.

Vi gjør også en antagelse om at tollfunksjonærene som er på vakt i løpet av døgnet, vil ha kapasitet til å stoppe et gitt antall biler pr. time. I dette eksempelet antar vi at det stoppes 8 biler i timen på dagtid og 3 biler i timen om natten.

Ut fra de antagelsene vi har gjort ovenfor, kan vi beregne prosentandelen av biler som er involvert i smugling for hver time i døgnet, som vist i figur 5.2. Prosentandelen er lavest i rushtiden, fordi det da er stor trafikk og relativt lite smugling. Tilsvarende er prosentandelen biler som er involvert i smugling stor om natten, fordi det da er mindre trafikk, og et større antall smuglere som passerer grenseovergangen.



Figur 5.2 *Våre innledende antagelser om biltrafikk og smuglerrate, gir denne prosentvise fordelingen av biler involvert i smugling, fordelt over døgnet. Vi ser at andelen av biler involvert i smugling er høyest om natten. Dette er resultat av generelt lite trafikk og relativt høy smuglerrate om natten. I rushtiden morgen og ettermiddag er prosentandelen smuglerbiler lav fordi antallet biler er høyt, mens antallet smuglere er konstant gjennom dagen. Tilsvarende er det en noe høyere andel smuglerbiler midt på dagen, fordi den øvrige trafikken er lavere enn i morgen- og ettermiddagsrushet.*

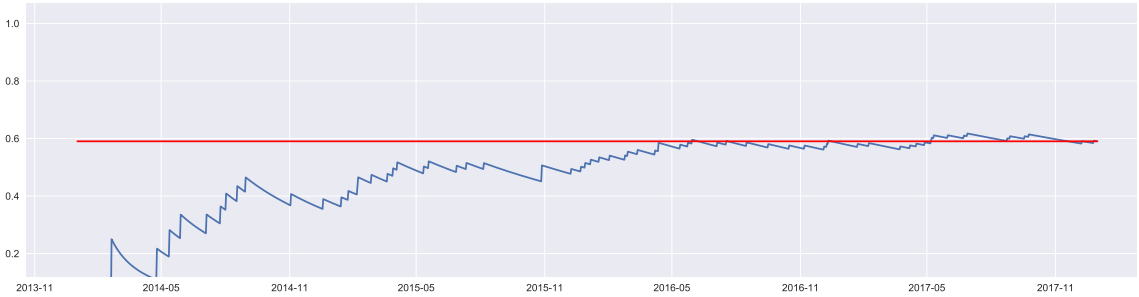
5.6.2 Databehov for beregning av smuglerrate

Vi har nå generert en "fasit" med utgangspunkt i noen antagelser om trafikk og smuglerrate.

La oss nå ta et konkret eksempel på smugling om morgenen, mellom 06:00 og 07:00. I dette tidsintervallet antar vi at det passerer noen få hundre biler, og som vi ser i figur 5.2 vil ca. 0,6 % av bilene vil være involvert i smugling. Dette betyr at hvis vi stopper en tilfeldig bil i denne tidsperioden, er det en sannsynlighet på 0,6 % for at denne bilen er involvert i smugling. Vi antar også at det kontrolleres tre biler om dagen (etter tilfeldige utvalg) i dette tidsintervallet. Spørsmålet er nå hvor lang tid det vil ta før vi kan oppnå et godt estimat av denne (for oss i utgangspunktet ukjente) sannsynligheten.

Her vil vi anta at variasjonen i ankomsthyppigheten av biler og hyppigheten av smuglere er som beskrevet ovenfor, men at statistikken er den samme fra dag til dag. Vi lar altså de ulike sannsynlighetene variere fra tidspunkt til tidspunkt gjennom døgnet, mens døgnet som helhet alltid er det samme.

Med utgangspunkt i disse antagelsene har vi gjort en simulering over fire år.



Figur 5.3 *Predikert sannsynlighet for smugling mellom 06:00 og 07:00, som funksjon av lengden på datainnsamlingsperioden. Den røde linjen viser den sanne sannsynligheten (vår fasit i simuleringen), mens den blå kurven viser estimert sannsynlighet på bakgrunn av tilfeldig utvalg over en periode på 4 år. Simuleringen viser at det tar nærmere to år før estimatet ligger innenfor 10% av den sanne sannsynligheten.*

Resultatet av simuleringen er vist i figur 5.3. Figuren viser hvilken tilnærming det er mulig å oppnå, til den sanne sannsynligheten for at en gitt bil frakter smuglergods, som funksjon av tidsperioden datainnsamlingen foretas over. Vi ser her at det vil gå nærmere to år før estimatet er innenfor 10% av den faktiske sannsynligheten, gitt de forutsetningene vi har satt i eksperimentet.

5.7 Anbefaling

Det anbefales at Tolletaten så snart som mulig starter innsamling og systematisk lagring av data som underlag for maskinlæring. Dette er lavthengende frukt, som med gode rutiner kan utføres som en del av det daglige arbeidet. Det som allerede finnes av sensordata, f.eks. røntgenbilder, må tas vare på og organiseres i en database, som man kan fortsette å bygge opp etter hvert som nytt materiale blir registrert. Det er viktig at informasjon om alle treff blir knyttet til rådataene, slik at man oppnår den merkingen av dataene, som kreves for effektiv maskinlæring.

Det anbefales også å samle inn data, basert på tilfeldig utvalg. I tillegg til å gi et større, og mer variert datamateriale, som omfatter også 'normale" tilfeller, har slike data en egenverdi ved at de over tid kan brukes til å bygge opp statistikk over faktisk smugling.

6 Konklusjon og anbefalinger

Rapporten beskriver fagfeltene mønstergjenkjenning og maskinlæring fordi vi tror at disse kan bidra til å bedre tolletatens utvelgelse av ulovlig vareførsel. Begrepene brukes i noen grad om hverandre og hører begge innunder det større fagfeltet kunstig intelligens. Forenklet kan man si at mønstergjenkjenning er den operative bruken av ferdige tilpassede prosesseringsalgoritmer og -metoder, mens maskinlæring er en systematisk metode for å tilpasse en prosesseringsalgoritme til et datasett. Flere ulike metoder blir beskrevet i rapporten, fra tradisjonelle klassifikatorer, som ble introdusert på 1950-tallet, via nyere metoder introdusert i perioden fra ca. 1980 frem til omkring 2010, og dyp læring som har fått sitt gjennombrudd de siste årene.

Alle metodene som er nevnt her har sin anvendelse også i dag, på en rekke problemstillinger, men hovedkonklusjonen i denne rapporten er at Tolletaten bør vurdere bruk av dyp læring for flere formål. Et område som er nevnt i rapporten er automatisk analyse av røntgenbilder. Selv om enkle eksperimenter viser at dyp læring har et stort potensiale på dette problemet, finnes det helt klart også store utfordringer som må overkommes. Det er først ved å arbeide med reelle og sammenlignbare data av tilstrekkelig omfang, for eksempel i et avgrenset pilotprosjekt, at det kan klargjøres hvilken ytelse et slikt system vil ha. Gitt at systemytelsen for gjenkjenning av viktige varer er minst like god som en domeneekspert har, vil det kunne vurderes å skalere opp til et operativt system.

Den største utfordringen er antakelig datainnsamling og merking av data. Det kan være en manuelt krevende jobb å skulle gå gjennom eksisterende (historiske) data pånytt for å merke dem. Her kan man eventuelt i første omgang involvere fagmiljøer som driver med maskinlæring for å vurdere eksisterende datakvalitet. Imidlertid er det enda viktigere å se framover på nye data som hele tiden genereres. Dette er en aktivitet Tolletaten kan starte opp med i dag, ved å designe en god datainnsamlingsprosess som er tilpasset arbeidsflyten og iverksette systematisk lagring og merking av data i det daglige arbeidet.

Det anbefales at datamaterialet som allerede finnes, men ligger lagret på ulike systemer, organiseres i en felles database slik at de er samlet tilgjengelig både for mønstergjenkjenning, analyse og maskinlæring. Nye registreringer av positive og negative funn bør også legges fortløpende inn i dette felles lagringssystemet, som ledd i den daglige virksomheten. Her er det viktig at all informasjon om funn blir knyttet til råmaterialet, slik at man oppnår den merkingen av data som er forutsetningen for effektiv maskinlæring.

For å oppnå en bedre situasjonsforståelse av de ulovlige varestrømmene bør det settes av ressurser til systematisk innsamling av data etter tilfeldige utvalg. Foreløpige vurderinger peker på at en slik prøvetakingsmetodikk vil måtte ha et perspektiv over noen år for å ha tilstrekkelig kvalitet. Det vil unektelig være en utfordring å bruke sårt tiltrengte kontrollressurser til å gjennomføre slik prøvetaking som vil ha et høyt innslag av negative resultater. Men å unnlate å bruke slike ressurser vil innebære å forbli blinde for ukjente mønstre i varestrømmene. Det kan neppe intelligente mennesker, organisasjoner eller maskiner være tilfreds med.

Referanser

[1] T. Engøy, J. I. Botnan, K. H. Løkken, T. R. Frømyr, M. Aronsen, A. Stolpe, T. A. Blix, I. Dyrdal, and L. Aurdal. Teknologiske muligheter for tolletaten – breddestudie. Technical Report FFI-RAPPORT 17/16605, FFI, 2017.

[2] T. Engøy, K. H. Løkken, I. Dyrdal, and L. Aurdal. Teknologiske muligheter for tolletaten – postportal. Technical Report FFI-RAPPORT 17/16605, FFI, 2017.

[3] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, New York, 2 edition, 2001.

[4] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

[5] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition*. Elsevier, San Diego, third edition edition, 2006.

[6] Hyeran Byun and Seong whan Lee. Applications of support vector machines for pattern recognition: a survey. In *Verri (Eds.), Lecture Notes in Computer Science*, pages 213–236, 2002.

[7] Paul E. Utgoff. Incremental induction of decision trees. *Machine Learning*, 4(2):161–186, Nov 1989.

[8] Michael J. Kearns and Yishay Mansour. A fast, bottom-up decision tree pruning algorithm with near-optimal generalization. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 269–277, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[10] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *ArXiv e-prints*, December 2017.

[11] Nicolas Jaccard, Thomas W. Rogers, Edward J. Morton, and Lewis D. Griffin. Tackling the x-ray cargo inspection challenge using machine learning. *Proc.SPIE*, 9847:9847 – 9847 – 13, 2016.

About FFI

The Norwegian Defence Research Establishment (FFI) was founded 11th of April 1946. It is organised as an administrative agency subordinate to the Ministry of Defence.

FFI's MISSION

FFI is the prime institution responsible for defence related research in Norway. Its principal mission is to carry out research and development to meet the requirements of the Armed Forces. FFI has the role of chief adviser to the political and military leadership. In particular, the institute shall focus on aspects of the development in science and technology that can influence our security policy or defence planning.

FFI's VISION

FFI turns knowledge and ideas into an efficient defence.

FFI's CHARACTERISTICS

Creative, daring, broad-minded and responsible.

Om FFI

Forsvarets forskningsinstitutt ble etablert 11. april 1946. Instituttet er organisert som et forvaltningsorgan med særskilte fullmakter underlagt Forsvarsdepartementet.

FFIs FORMÅL

Forsvarets forskningsinstitutt er Forsvarets sentrale forskningsinstitusjon og har som formål å drive forskning og utvikling for Forsvarets behov. Videre er FFI rådgiver overfor Forsvarets strategiske ledelse. Spesielt skal instituttet følge opp trekk ved vitenskapelig og militærteknisk utvikling som kan påvirke forutsetningene for sikkerhetspolitikken eller forsvarsplanleggingen.

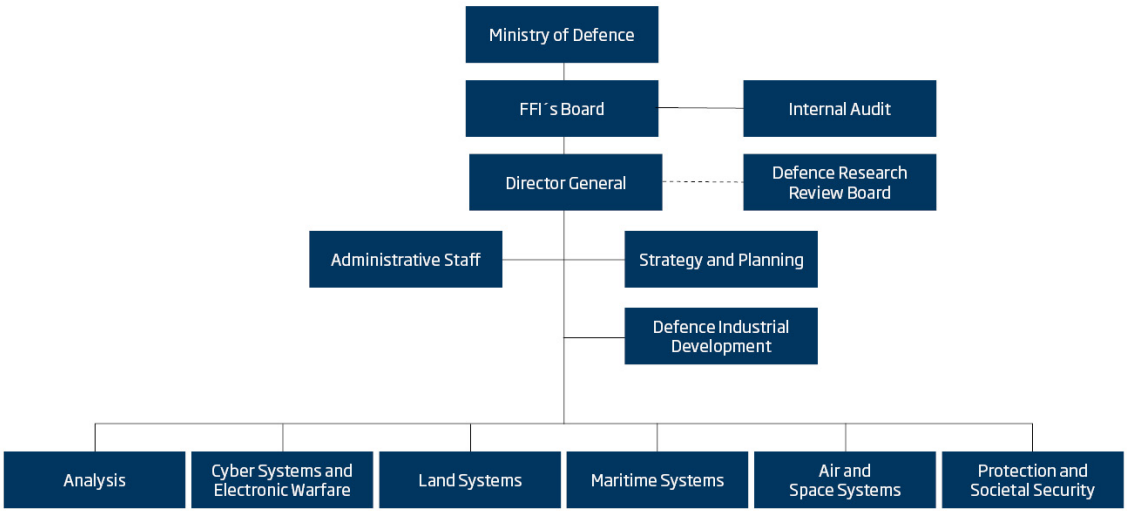
FFIs VISJON

FFI gjør kunnskap og ideer til et effektivt forsvar.

FFIs VERDIER

Skapende, drivende, vidsynt og ansvarlig.

FFI's organisation



Forsvarets forskningsinstitutt
Postboks 25
2027 Kjeller

Besøksadresse:
Instituttveien 20
2007 Kjeller

Telefon: 63 80 70 00
Telefaks: 63 80 71 15
Epost: ffi@ffi.no

Norwegian Defence Research Establishment (FFI)
P.O. Box 25
NO-2027 Kjeller

Office address:
Instituttveien 20
N-2007 Kjeller

Telephone: +47 63 80 70 00
Telefax: +47 63 80 71 15
Email: ffi@ffi.no

FFI Forsvarets forskningsinstitutt
Norwegian Defence Research Establishment