Omvärldsanalys inom signalbehandlingsalgoritmer för EO/IR-sensorer

FREDRIK NÄSSTRÖM, FREDRIK BISSMARCK, VIKTOR DELESKOG,
DAVID GUSTAFSSON, GUSTAF HENDEBY, JÖRGEN KARLHOLM,
JONAS NORDLÖF, JONAS NYGÅRDS

FOI-R--4287--SE
ISSN 1650-1942 Juni 2016

Fredrik Näsström, Fredrik Bissmarck,
Viktor Deleskog, David Gustafsson,
Gustaf Hendeby, Jörgen Karlholm,
Jonas Nordlöf, Jonas Nygårds

Omvärldsanalys inom signalbehandlingsalgoritmer för EO/IR-sensorer

Titel	Omvärldsanalys inom signalbehandlingsalgoritmer för EO/IR-sensorer
Title	A survey of signal processing algorithms for EO/IR sensors
Rapportnr/Report no	FOI-R--4287--SE
Månad/Month	Juni
Utgivningsår/Year	2016
Antal sidor/Pages	64 p
ISSN	1650-1942
Kund/Customer	Försvarsmakten
Forskningsområde	7. Sensorer och signaturanpassning
FoT-område	Sensorer och signaturanpassning
Projektnr/Project no	E72601
Godkänd av/Approved by	Per Johannesson
Ansvarig avdelning	Ledningssystem

Omslagets bildcollage: Översta raden från vänster till höger, den vänstra bilden visar en IR-bild med automatisk måldetektion och den högra bilden visar spaning från stridsfordon 90 (Foto: Janne Bohman/Försvarsmakten). Nedersta raden från vänster till höger, den vänstra bilden visar stridsfordon som tar sig fram där minröjarna har röjt väg (Foto: Anne-Lie Sjögren/Försvarsmakten) och den högra bilden föreställer TUAV i luften under Mali 02 (Foto: Jimmy Croona/Försvarsmakten)

Sammanfattning

Projektet Intelligenta spaningsfunktioner har i uppgift att bedriva forskning med syfte att göra framtida sensorsystem mer intelligenta. Ett intelligent sensorsystem kan i framtiden ge sensoroperatörer bättre situationsuppfattning, högre systemtilltro, mindre risk för felhandlingar (eng. human error) och lägre mental arbetsbelastning.

I denna rapport redovisas en omvärldsanalys som har genomförts inom området signalbehandling för EO/IR-sensorer. I detta arbete har vi analyserat forskningen inom automatisk detektion, måligenkänning, målföljning, positionering och sensorstyrning.

Inom detektion och måligenkänningsområdet har utvecklingen varit mycket kraftig de senaste åren. En relativt ny ansats inom maskininlärning som kallas Deep learning har på ett dramatiskt sätt flyttat fram forskningsfronten inom flera bildbehandlingstillämpningar till nivåer som tidigare var orealistiska och ansågs ligga långt in i framtiden.

Forskning inom positionering är fortsatt ett mycket aktivt område. Framför allt inom SLAM (*simultaneous localization and mapping*) sker mycket forskning. SLAM går ut på att med en rörlig sensorplattform samtidigt kartera och lokalisera plattformen i kartan som byggs ut vartefter plattformen rör sig. Det kan göras genom att hitta och följa stabila och väldefinierade punkter i bilder, s.k. *landmärken*, från olika sensorer som sedan kan användas för att beräkna ett förbättrat positions- och orienteringsestimat. Mer avancerade SLAM-metoder blir allt vanligare där alla landmärken används i ett och samma ekvationssystem för att beräkna hela rörelsebanan (d.v.s. inte bara den senaste positionen).

Forskningen inom sensorstyrningsområdet är fortsatt mycket aktiv och nya metoder och algoritmer utvecklas för att lösa planeringsproblemet. För sensorstyrning av flera sensorsystem innebär det nya frågeställningar om hur och vilken information som ska utbytas mellan sensorsystemen för att de ska samverka. Forskningsområdet kring multi-robot systems (samverkande robotar) är på stor frammarsch inom forskningen med avancerade tillämpningar i olika robottävlingar. Samverkande sensorer har blivit allt mer intressant de senaste åren för att utnyttja alla tillgängliga sensorer på bättre sätt.

Nyckelord: detektion, måligenkänning, målföljning, positionering och sensorstyrning

Summary

The project Intelligent reconnaissance functions has the task to conduct research with the aim to make future sensor systems more intelligent. By being more intelligent, future sensor systems provide sensor operators with better situational awareness, higher system confidence, reduced risk of human errors (eng. Human error) and low mental workload.

This report aims at giving an overview of current research in signal processing for EO/IR sensors. In this work, we have analyzed the research in automatic target detection, target recognition, target tracking, positioning and sensor control.

The research within target detection and target recognition has been very strong in recent years. A relatively new approach of machine learning called Deep learning has produced dramatic advances in several image processing applications to levels that were previously unrealistic and far into the future.

Research within the positioning area is still very active. Especially SLAM (simultaneous localization and mapping) is a very active research area. The goal with SLAM is with a sensor platform in motion at the same time build a map and locate the platform in the map that is being expanded when the platform move. This can be made by finding and following stable and well defined points in images from different sensors, called landmarks, which then can be used to calculate an improved position and orientation estimate. More advanced SLAM methods are becoming more common where all landmarks are used in the same system of equations to calculate the whole trajectory (i.e. not just the last position).

In the sensor control area, the research is still very active with many different applications. Research in this area is constantly advancing with new methods and algorithms applied to solving the planning problem. For sensor control of multiple sensor implies new questions about how and what information that should be exchanged between the sensor systems so that they could cooperate. Research on multi-robot systems (cooperative robots) is emerging quickly in research with advanced applications in various robot competitions. Cooperative sensors have become increasingly attractive in recent years to use all available sensors in a better way.

Keywords: detection, recognition, target tracking, positioning and sensor control

Innehåll

1 Inledning

Målet för det här projektet är att studera generisk funktionalitet för samverkande styrbara sensorsystem. Syftet är att underlätta sensoroperatörens arbete och därmed öka effektiviteten och kvalitén på spaningen. Den nya funktionaliteten som tas fram i projektet ska bidra till framtida intelligenta operatörssystem som erbjuder bra situationsmedvetenhet, utsätter operatören för låg mental arbetsbelastning, hög systemtilltro och liten risk för felhandlingar (eng. human error).

Utvecklingen inom signal- och bildbehandlingsområdet går snabbt över hela världen. Detta gör att nya automatiska funktioner hela tiden utvecklas. Fyra fördelar med automatiska system är att det vanligen leder till:

- Ökad effektivitet (precision/snabbhet)
- Ökad säkerhet
- Minskad arbetsbelastning (mental och fysisk)
- Förbättrad ekonomi

Men även om nya automatiska funktioner utvecklas för signal- och bildbehandling så behövs fortfarande en människa som utifrån det övergripande sammanhanget tolkar skeenden och prioriterar sensorresurser. Människan är därför ofta oersättlig precis som i många andra komplexa system. Förr eller senare är det alltid en människa som ansvarar för att styra, justera och omkonfigurera systemet och dess automatiska funktioner beroende på aktuella krav. Systemen måste därför konstrueras så att människan kan använda dem på ett effektivt och säkert sätt.

En vanlig orsak till att operatörer kan ha svårt att hantera komplexa system på avsett sätt är bristen på återkoppling (feedback). Operatören får helt enkelt för lite eller otydlig information om systemets status och hur den utvecklas [1]. Om systemet och operatören inte kan kommunicera på ett tillfredsställande sätt uppfattar operatören att systemet gör helt oväntade saker, s.k. Automation Surprises. Det här var tidigare ett problem inom den civila luftfarten när man började använda automatiserade funktioner som resulterade i en helt ny typ av tillbud och olyckor [2]. Just bristen på återkoppling till operatören var en av orsakerna till de här problemen, vilket gjorde det mycket svårt för operatören att ingripa och göra rätt saker för att hantera störningar. En felinmatad siffra eller ett glömt kommatecken resulterade i att systemet stumt utförde vad det blivit instruerat att göra hur fel detta än var och utan att visa piloterna resultatet [3].

Det som avgör hur bra operatörens gränssnitt fungerar är vilka förutsättningar det ger för att förstå farkostens förmåga att hantera förväntade situationer och att koordinera hur uppgiften ska lösas [4]. Hur omfattande koordinationsproblemet är beror på vilka krav situationen ställer och farkosternas förmåga att upprätthålla en acceptabel prestationsnivå utan operatörens medverkan. Om koordinationsproblemet är svårt så minskar antalet farkoster som operatören kan hantera. Det är därför viktigt med bra gränssnitt för svåra koordinationsproblem så att operatören inte blir överbelastad.

Det är vanligt att operatören blir överbelastad genom att komplicerade system presenterar för mycket data. En uppgiftsanalys bör därför ligga till grund för vilken information operatören behöver i varje enskild fas av ett uppdrag, och att han/hon ges tillfälle att när som helst ingripa och rätta till felaktigheter. Att i ett tidigt skede undersöka vilken information som operatören behöver gör att man kan undvika dyrbara modifieringar och förseningar av systemet.

För detta projekt har vi valt ett spaningsscenario med bemannade stridsfordon som befinner sig i ett fientligt område med risk för andra mekaniserade förband. Målet med arbetet i projektet är att ta fram generisk funktionalitet för samverkande styrbara sensorsystem för att öka effektiviteten och kvalitén på spaningen, men samtidigt underlätta

sensoroperatörens arbete. För att lyckas med detta har forskningen inriktats mot styrning av sensorer för optimal informationsinhämtning vid spaning mot markmål (människor och fordon), automatisk måligenkänning, följning, positionsbestämning av multipla mål i komplexa miljöer från rörliga sensorplattformar i samverkan, i syfte att skapa en gemensam lägesbild. Geografisk information utnyttjas för förbättrad sensorstyrning, egenpositionering och hotbildsvärdering.

I denna rapport ges en omvärldsanalys över några forskningsområden som är viktiga för projektet. För respektive forskningsområde har vi försökt beskriva state-of-the-art inom området och vad forskningen bör inriktas mot i framtiden. Även viktiga forskningsprojekt inom området finns beskrivna.

2 Detektion

För projektet Intelligenta spaningsfunktioner är det mycket viktigt att kunna detektera både stillastående och rörliga mål i en klottrig bakgrund. För att detta ska vara möjligt behövs en robust detektionsalgoritm som kan analysera och diskriminera former, vilket ställer krav på den spatiella upplösningen (antalet bildpunkter per ytenhet). För att samtidigt få hög detektionssannolikhet, låg falsklarmsnivå och kort beräkningstid har maskininlärning visat sig vara mycket användbart. Kärnan i detektionsalgoritmer med maskininlärning är en klassificerare. Det finns ett flertal kraftfulla klassificeringsmetoder idag och de vanligaste är idag neuronät [5], supportvektormaskiner [6] och boosting [7].

Produkter med automatisk detektion och följning av mål i EO/IR-video från fordon börjar komma ut på marknaden [8]. Rörelsebaserad detektion finns för EO/IR-system i operativa UAV:er [9] och i fordonssystem [10].

För mål med god spatial upplösning (10-15 pixlar längs objektets huvudaxel) har robust detektion av fordon resp. människor erhållits i realtid [11,12]. Detta gäller termisk IR och objekt som inte använder avancerad signaturanpassningsteknik. Fortsatt forskning behövs för detektion av delvis skylda mål, detektion i dåligt väder etc. där detektionsprestanda idag inte är tillfredsställande.

Nedan anges några militär projekt och grupper som behandlar olika aspekter av detektion:

DARPA-projektet TAILWIND - Tactical Aircraft to Increase Long Wave Infrared Nighttime Detection – utvecklar ett system för automatisk detektering och följning av fordon och avsutten trupp för Shadow UAV [13]. Följande funktionalitet adresseras:

- Kontinuerlig övervakning av terrängavsnitt med LWIR-sensor
- Detektering och följning av fordon och avsutten trupp
- Generera 3D-modell (struktur-från-rörelse) av den avspanade terrängen, och utnyttja denna vid målföljningen
- Generera larm när ett mål går in i eller lämnar ett av operatören markerat område
- Videosammanfattning (video summarization, non-cronological video mosaic)
- Skillnadsdetektion

NATO-gruppen ”Unmanned Systems (UMS) Platform Technologies and Performances for Autonomous Operations (AVT-175)” behandlar perception, underrättelse, miljö, plattform och kommunikation för obemannade luft-, land- och sjöfarkoster.

NATO-gruppen “Disposable Multi-Sensor Unattended Ground Sensors Systems for Detecting Personnel (SET-158)” behandlar sensorsystem med låg kostnad och låg energiförbrukning som kan detektera, följa och klassificera människor. Icke-avbildande sensorer och billiga avbildande sensorer studeras för att öka säkerheten i urban och öppen terräng.

I avsnitt 2.1 och 2.2 nedan ges en mer teknisk beskrivning av aktuell forskning inom detektering av utbredda mål respektive rörelse.

2.1 Detektering av utbredda mål

Framstående forskning, huvudsakligen inriktad mot detektering av fotgängare och ansikten, bedrivs vid ett stort antal universitet och forskningsinstitut samt vid IT-företag

som Adobe, Google, Microsoft och Yahoo. Företag inom fordonsbranschen, som Autoliv, Bosch, Continental, Daimler, Honda, MobilEye och Volkswagen har också omfattande verksamhet inriktad mot detektering av fotgängare och fordon.

2.1.1 Detektionsprinciper

De senaste åren har två alternativa principer för måldetektering utkristalliserats. Den vanligaste är *fönsteroperatorn* där bilden delas upp i ett stort antal överlappande rektangulära regioner (fönster). En klassificerare avgör om innehållet i regionen är mål eller bakgrund. Klassificeraren utformas av effektivitetsskäl vanligen som en följd av bearbetningssteg, där i varje steg ett beslut tas om att acceptera eller förkasta regionen; i det senare fallet avbryts bearbetningen, och regionen antas tillhöra bakgrunden. Viola och Jones boostingbaserade detektorkaskad [14] och den SVM-baserade sekventiella detektorn av Romdhani *et al.*[15] är klassiska typexempel. Bearbetningen av en hel bild kan ske genom att bilden avskannas sekventiellt (eventuellt i flera skalor [16]), vilket ger en ström av regioner att bearbeta för en eller flera processorer, eller genom massivt parallell bearbetning i exempelvis en grafikprocessor.

Generaliserad Houghtransform är en alternativ ansats där lågnivåstrukturer (t.ex. hörnpunkter) i bilden först detekteras. Till varje sådan struktur finns sedan en sannolikhetsfördelning över målparametrar som storlek och position (relativt strukturen) kopplad. Sannolikhetsmassorna från olika strukturer summeras och lokala maxima i parameterrummet identifieras. Värden som överstiger en tröskel klassificeras som mål. Proceduren kan betraktas som ett röstningsförfarande där bildstrukturer pekar ut var i bilden mål finns. En fördel med Houghtransformbaserade metoder är att röstningen gör det lättare att hantera objektklasser med stora formvariationer (t.ex. artikulation). Exempel på denna typ av ansats är Gall och Lempitsky [17] samt Okada [18], som båda använder Random Forest-klassificerare [19] för att kategorisera lokala strukturer. Leibe *et al.* [20] presenterar en snarlik metod som dock använder en mer traditionell klustringsmetod för att gruppera strukturer vid träningen och matchning mot klusterprototyper för att bestämma närmaste kategori för testdata. Shotton *et al*. [21] och Opelt *et al*. [22] använder boosting för att bestämma optimala röstvikter för en uppsättning diskriminerande kantsegment.

2.1.2 Hantering av okända målparametrar

En viktig fråga vid detektering är hur man ska hantera de stora signaturvariationer som finns inom en målklass och som bl.a. härrör från målets okända orientering. Två principiella ansatser kan urskönjas. I den ena ansatsen delas träningsdata i förväg upp i olika parameterintervall och separata detektorer tränas för vart och ett av dessa [23]. En region måste klassificeras som bakgrund av samtliga specialiserade detektorer för att förkastas. En mer beräkningseffektiv variant är detektorpyramiden [24] där parameterrummet (t.ex. orientering) delas upp allt finare i ett antal nivåer eller upplösningar. På det första nivån sker ingen uppdelning, utan en klassificerare tränas på samtliga data att förkasta en viss andel bakgrunder. Accepterade regioner går vidare till nästa nivå där en grov uppdelning sker, och klassificerare tränas för specifika parameterintervall. Klassificerarna på samma nivå arrangeras i en sekvens så att en region som förkastas går vidare till nästa steg (ett annat intervall). En region måste alltså förkastas av samtliga klassificerare på samma nivå. Accepterade regioner går vidare till nästa nivå där en ännu finare uppdelning sker, osv. Jones och Viola [25] tränar ett beslutsträd att skatta målets orientering, och fördelar baserat på detta regioner mellan klassificerare tränade på specifika orienteringsintervall. Orienteringsskattning är dock svårt. Huang *et al.*[26] använder liksom i detektorpyramiden en hierarki av upplösningar, men organiserat i en trädstruktur. En viss nod är specialiserad på ett parameterintervall och har under sig noder

specialiserade på delintervall av detta. I den övre noden finns dock separata klassificerare för vart och ett av delintervallen, men deras beräkningar organiseras på ett effektivt sätt, genom att dela särdrag, så att merarbetet i förhållande till att ha en ensam klassificerare är relativt litet. En region som accepteras av en klassificerare skickas vidare till motsvarande dotternod. Det betyder att en region kan skickas vidare till flera dotternoder, vilket ger en större robusthet när parametervärdet (t.ex. orienteringen) är osäkert. Lin och Liu [27] använder en liknande metodik.

Den andra huvudansatsen är att automatiskt dela upp data i mer homogena delmängder, och träna specifika detektorer för dessa. Detta är en mer generell metodik, som kan hitta grupperingar i data som kanske inte är uppenbara. Wu och Nevatia [28] presenterar en metod för att automatiskt bygga en trädstruktur av samma slag som Huang *et al*. Initialt används samtliga träningsdata för att bygga en boostingbaserad detektor. För varje ny boostingiteration kontrollerar man att den nya komponenten tillför tillräcklig diskrimineringsförmåga; om så inte är fallet delas målexemplen upp i två mer homogena grupper genom *k*-meansklustring som skickas till olika dotternoder (bakgrundsexemplen skickas till båda). Splittringen i två grupper propageras uppåt i trädet, ända till rotnoden, och gruppspecifika klassificerare tränas även i dessa noder, dock används samma särdrag som tidigare extraherats. Babenko *et al*. [29] härleder en ny boostingalgoritm, MPLBoost (*Multiple Pose Learning*), som samtidigt delar upp måldata i K grupper och tränar en detektor för var och en av dessa. Träningsdata för detektering kan uttryckas som en uppsättning par (x_i,y_i) där x_i är en bildregion och y_i anger klasstillhörighet, +1 för mål, -1 för bakgrund. Den grundläggande idén i [29] är att ersätta y_i med $\max_k(y_{ik})$, där y_{ik} är en gruppspecifik klasstillhörighet som skattas under träningen. Wojek *et al*. [30] visar att MPLBoost ger en betydande prestandaförbättring jämfört med en traditionell boostingalgoritm (dvs. $K=1$).

Viola *et al*. [31] (generaliserat av Babenko *et al*. [29]) adresserar ett relaterat, mycket viktigt problem vid detektorträning, nämligen att detektorns prestanda är starkt beroende av att målexemplen är ensade. När träningsdata extraheras manuellt genom annotering är det väldigt svårt och tidskrävande att se till att samtliga exempel är korrekt centrerade och i en och samma skala. När målklassen innehåller objekt av starkt varierande utseende (t.ex. fordon) kan det även vara svårt att definiera en korrekt centrering. Genom att ersätta varje målexempel med en uppsättning translaterade och skalade versioner, kan träningen formuleras som att det räcker att minst en av dessa versioner blir korrekt klassificerad. Viola *et al*. kallar detta *Multiple Instance Learning* (MIL) och härleder en ny boostingalgoritm, MILBoost, för denna formulering.

2.1.3 Särdrag

Särdrag, dvs. de mätningar av informationsbärande strukturer som görs i bilden och som används av klassificeraren för att bestämma klasstillhörigheten hos data, är av avgörande betydelse för detektionsprestanda. Mycket har hänt sedan Viola och Jones introducerade integralbildsrepresentationen och beräkningseffektiva rektangelfilter [14] för mer än ett decennium sedan. Ett stort steg framåt vad gäller prestanda togs genom introduktionen av histogram över gradientriktningar (HOG), baserat på Lowes SIFT-deskriptor [32]. Dessa ger en vektoriell beskrivning av kantfördelningen (position och riktning) i ett rektangulärt område som är betydligt mer informativ vad avser områdets form (bl.a. silhuett) än ett Viola-Jonessärdrag. Histogram över rörelsevektorer kan i princip beräknas på liknande sätt som HOG [30,33], men för rörliga kameror måste man kompensera för bakgrundens rörelse. Ytterligare ett steg togs genom att införa särdrag känsliga för en ytas texturering, exempelvis histogram över lokala binära mönster (*local binary patterns*, LBP) [34]. HOG och LBP är komplementära och ger tillsammans en mer fullständig beskrivning av regioner, och en kombinerad deskriptor ger klart förbätt-

rad diskrimineringsförmåga [35]. Etablerade metoder använder ett eller flera endimensionella histogram. Flerdimensionella histogram kan fånga samvariationen mellan olika särdrag, men kan inte skattas utan orealistiskt stora mängder träningsdata. Tuzel *et al*. [36] representerar en bildregion med en kovariansmatris för en uppsättning olika särdrag, vilket är en grövre beskrivning av samvariation. Histogram och kovariansmatriser för särdragsfördelningar i godtyckliga rektangulära omgivningar kan effektivt beräknas från integralbilder [37]. Dollár *et al*. [38] introducerar beteckningen integralkanalsärdrag (*integral channel feature*) som ett allmänt begrepp för regiondeskriptorer baserade på integralbildsrepresentationer.

Ovan nämndes kantsegment som särdrag för Houghbaserade detektionsmetoder. Notera att kantsegment är mer komplexa, sammansatta strukturer (t.ex. linje- och kurvsegment) än lokal orientering, som används i HOG-deskriptorn. Även för fönsteroperatorer har dessa använts, exempelvis [39]. Lim *et al*. [40] visar att HOG och histogram över olika typer av kantsegment är komplementära och tillsammans leder till signifikant bättre detektionsprestanda. Rent allmänt kan sägas att flera studier visar på en påtaglig nytta av komplementära särdragstyper [41,42].

En intressant frågeställning är hur vektoriella särdrag ska hanteras i en boostingklassificerare med beslutsträd som komponenter (i supportvektormaskiner är det rättframt). Viola och Jones filter är skalära och författarna använder ett binärt beslutsträd med en enda intern nod (*decision stump*), där utsagan baseras på utfallet av en jämförelse mellan ett skalärt särdragssvar och ett tröskelvärde. För vektoriella särdrag har två huvudsakliga alternativ använts. Ett sätt är att växa ett större binärt träd där, för varje inre nod, valet av dotternod baseras på jämförelse mellan en av vektorns komponenter och en tröskel. Trädet kan byggas iterativt enligt sedvanlig metodik för klassificerings- och regressionsträd [43], eventuellt med beskärning (*pruning*). Den alternativa och mer generella metoden är att till varje intern nod optimera en skalär funktion vars värde sedan kan jämföras med en tröskel. Funktionen ska vara diskriminerande, dvs. separera mål- och bakgrundsexempel. Linjära funktioner (projektioner) är vanligast. Zhu *et al*. [44] bestämmer en projektion för HOG-vektorer genom att träna en linjär SVM; Laptev [45] applicerar istället en Fisherprojektion på samma deskriptor. Liu och Shum [46] söker en linjär projektion som maximerar Kullback-Leibleravståndet mellan mål- och bakgrundsfördelningarna, men presenterar inte en allmänt tillämpbar algoritm, utan bara en metod att linjärt kombinera en given uppsättning projektionsriktningar. Fisherprojektionen ger maximal separation om mål- och bakgrundsfördelningarna är normalfördelade med identiska kovariansmatriser. En generalisering, Bhattacharyya-projektionen [47], är tillämpbar också när klasserna har identiska medelvärden men olika kovariansmatriser.

2.2 Detektering av rörliga mål från en rörlig plattform

Den vanligaste metodiken för rörelsedetektion från en plattform i rörelse fungerar på så sätt att successiva bilder över samma scen geometriskt ensas mot varandra för att kompensera för förändringar orsakade av sensorns egenrörelse. Därefter analyseras den återstående skillnaden mellan de geometriskt transformerade bilderna.

Om bakgrunden är helt plan, eller höjdvariationerna små i förhållande till avståndet mellan sensorn och markplanet, kan man fullständigt kompensera för egenrörelsen (translation, rotation) samt zoomning genom en linjär avbildning i homogena bildkoordinater (planhomografi) [48]. Avbildningen skattas genom regression från en uppsättning punktkorrespondenser. Ensningen misslyckas (bortsett från brus och fel i rörelseskattningen) endast i områden där man har en rörelse relativt bakgrunden. Denna metodik är ofta tillräcklig vid flyg- och UAV-spaning från hög höjd [49].

Om bakgrundens höjdvariationer relativt avståndet till sensorn väsentligt avviker från ett plan (urban eller kuperad terräng) kan man använda en så kallad *plan + parallax-ansats* [50], där man (automatiskt) försöker finna ett plan i terrängen (t.ex. gatuplanet) och ensa bilderna med avseende på detta. Detta kompenserar fullständigt för sensorns rotation och eventuell zoomning. Strukturer i bakgrunden som avviker från planet (t.ex. hustak) kommer dock pga. sensorns translation att ha en återstående förskjutning mellan bilderna, en s.k. parallaxrörelse. Man kan visa att parallaxvektorerna pekar mot en gemensam punkt i bilden, den s.k. epipolen. Om epipolens position kan skattas med tillräcklig noggrannhet kan förskjutningsvektorer som inte pekar mot denna identifieras som oberoende rörelse. Ofta är det dock svårt att skatta epipolens läge pga. att det inte finns tillräckligt mycket finstruktur i bilden. Dessutom är det vanligt (t.ex. vid UAV-spaning längs en väg) att sensorn rör sig parallellt med målet, vilket medför att målets rörelsevektor pekar mot epipolen – det krävs alltså ytterligare villkor för att detektera sådan rörelse. Genom att ensa tre konsekutiva bilder mot varandra med avseende på ett gemensamt plan kan man härleda villkor som inte kräver skattning av epipolen och även möjliggör detektering av oberoende rörelse i epipolriktningen [49]. Lourakis *et al*. [51] analyserar, efter att ha ensat tre bilder, normalflödesvektorer (dvs. rörelsevektorernas komponent längs mot den lokala bildgradienten), vilka kan skattas mer robust än de fullständiga förskjutningsvektorerna (optiskt flöde). Sawhney *et al*. [52] ensar också tre bilder relativt ett gemensamt plan i scenen, men skattar sedan epipolerna genom robust regression baserat på punktkorrespondenser mellan de tre bilderna (se nedan). Givet detta kan sedan en faktor i varje bildpunkt proportionell mot avståndet skattas från den återstående förskjutningen mellan ett ensat bildpar. Denna formfaktor kan sedan tillämpas på förskjutningsfältet mellan ett annat ensat bildpar; bildpunkter där förskjutningen inte kan kompenseras fullständigt är i rörelse relativt bakgrunden. Yuan *et al*. [53] utvidgar plan + parallaxmetodiken till att fungera även om bilderna inte ensats mot ett gemensamt plan i scenen.

Ett alternativ till plan-parallaxmetodiken är att utnyttja algebraiska samband för punktkorrespondenser i multipla vyer. Dessa kräver inte att man kan identifiera ett plan i scenen. Epipolarvillkoret kan uttryckas som ett bilinjärt samband mellan korresponderande punkter i två bilder. Låt x_i och x_j vara homogena koordinater för punkterna i respektive bild. Då gäller att $x_i^T F x_j = 0$ om den avbildade strukturen tillhör bakgrunden. F kallas för *fundamentalmatris* eller bifokal tensor och kan skattas genom robust regression från en uppsättning punktkorrespondenser [48]. För punktkorrespondenser (x_i, x_j, x_k) i tre bilder finns ett motsvarande samband

$$\begin{bmatrix} 0 & -x_{j3} & x_{j2} \\ x_{j3} & 0 & -x_{j1} \\ -x_{j2} & x_{j1} & 0 \end{bmatrix} \left(\sum_{l=1}^{3} x_{il} T_l \right) \begin{bmatrix} 0 & -x_{k3} & x_{k2} \\ x_{k3} & 0 & -x_{k1} \\ -x_{k2} & x_{k1} & 0 \end{bmatrix} = 0_{3\times 3}$$

De tre matriserna $\{T_1, T_2, T_3\}$ utgör den *trifokala tensorn* i matrisrepresentation, och dess komponenter kan även de skattas genom robust regression från en uppsättning punktkorrespondenser. ter Haar *et al.* [54] jämför prestanda för metoder baserade på plan + parallax, fundamentalmatrisen och den trifokala tensorn.

Ytterligare en alternativ metodik är att skatta sensorns rörelse. Om sensorns rotation mellan bilderna är känd (t.ex. från en tröghetssensor) och sensorn internt kalibrerad (fokallängd m.m. kända) kan man kompensera för denna del av förskjutningen. Det återstående förskjutningsfältet för punkter i bakgrunden pekar mot epipolen. Om även sensorns translation är känd kan epipolens läge beräknas analytiskt [55]. Bildpunkter där den återstående förskjutningen inte pekar mot epipolen rör sig relativt bakgrunden. Man kan gå ytterligare ett steg och även skatta avståndet (djupet) i varje bildpunkt givet sensorns rörelse och det uppmätta rörelsefältet mellan två bilder. Givet detta kan avvikelser

mellan det uppmätta rörelsefältet och det analytiskt beräknade rörelsefältet för bakgrunden detekteras [56].

Ovan refererade metoder är baserade på punktkorrespondenser (optiskt flöde) mellan två eller tre successiva bilder, vilket gör dem känsliga för brus och fel i rörelseskattningen. Korrespondenser mellan ett större antal bilder kan åstadkommas genom s.k. *partikeladvektion* [57]. Optiskt flöde beräknas mellan successiva par av bilder. Från initiala positioner $\boldsymbol{x}_i(0) = (x_i(0),y_i(0))$ i den första bilden propageras ”partiklar” sedan successivt i tiden genom numerisk integration, som om det optiska flödet vore hastighetsfältet för en vätska. Detta ger upphov till trajektorier $\boldsymbol{x}_i(t) = (x_i(t),y_i(t))$. Dey *et al.* [58] propagerar partiklar genom en volym bilder, och använder de resulterande trajektorierna för att skatta parametrar i en fundamentalmatris som uttrycker epipolarvillkoret för samtliga bildpar i volymen. Detta ger betydligt ökad robusthet och detektionsförmåga jämfört med att bara använda ett bildpar. Guo *et al.* [59] och Sheikh *et al*. [60] noterar att för en affin kamera (svag perspektivprojektion) ligger samtliga trajektorier för punkter i bakgrunden i ett tredimensionellt linjärt underrum [61]. En bas i underrummet skattas genom robust regression. Rörelse relativt bakgrunden kan alltså detekteras som trajektorier som inte passar in i underrummet.

Boxer och Loveard [62] diskuterar viktiga praktiska aspekter vid detektering av rörelse från en rörlig plattform, bl.a. kvaliteten hos skattningen av optiskt flöde (dvs. pixelkorrespondenser mellan successiva bilder). Ett stort problem är de felaktigheter som uppkommer genom att scenstrukturer inte är synliga i båda bilderna p.g.a. ocklusion, då det givetvis inte går att hitta en korrespondens. Det är mycket viktigt att kunna detektera sådana situationer, som är vanliga i naturliga miljöer, t.ex. skog, för att undvika falsklarm. Metoder som utvecklats för detta är, exempelvis, att beräkna flödet både framåt och bakåt i tiden och kräva att korrespondenserna överensstämmer (*forward-backward error*), samt att säkerställa att de lokala omgivningarna (ett litet fönster) kring de korresponderande punkterna har samma utseende, vilket exempelvis kan göras genom normaliserad korrelation. Se [57] för en utförlig diskussion av detta problem.

3 Måligenkänning

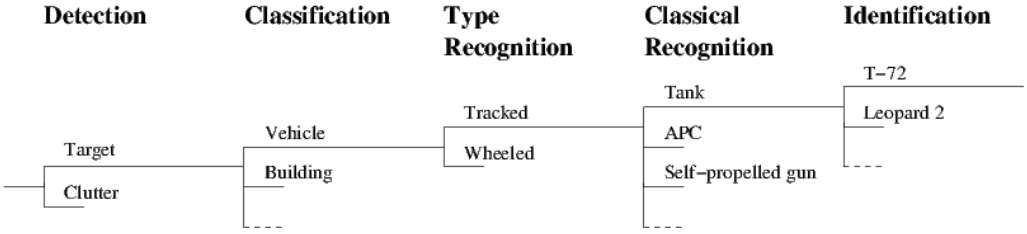
3.1 Definition

Omfattande studier har inom försvarsforskningen gjorts för att beskriva kopplingen mellan sensor- och miljöparametrar (t.ex. brusnivå, upplösning och sikt) och möjligheten att diskriminera mellan olika objektkategorier. I synnerhet U.S. Army Night Vision Laboratory har historiskt haft stor betydelse inom området, och, utgående från psykologiska experiment, producerat en rad programvaror för skattning av möjligheten att under olika betingelser lyckas klassificera objekt. En uppdelning i diskrimineringsnivåer som har använts inom detta område visas i tabell 1, se [63].

Tabell 1. Diskrimineringsnivåer. Efter [63].

Task	Description	Example
Detection	The object has a reasonable probability of being of military interest	A blob of relevant size and shape
Classification	The broad class of object types	Building or vehicle
Type Recognition	The general class of the object	Tracked or wheeled vehicle
Classical Recognition	The specific class of the object	Car, van, pickup, tank, armoured personnel carrier
Identification	The specific type within the class	T-72, Leopard 2

Givet denna uppdelning i nivåer kan man tänka sig en process där klassificeringsuppgiften formuleras i form av ett beslutsträd, se figur 1. För att nå högre nivåer av diskriminering krävs allt bättre kvalitet på bildmaterialet (t.ex. högre upplösning).



Figur 1. Beslutsträd för klassificering som använts inom försvarstillämpningar. Efter [63].

Inom den akademiska forskningen skiljer man vanligen mellan tre nivåer:

- Detektering: att i en bild hitta samtliga instanser av en viss objektkategori, t.ex. ansikten, fotgängare, bilar (vilket i samtliga fall fått kommersiella tillämpningar). *Lokalisering* betecknar specialfallet då man vet hur många instanser som finns i en bild (typiskt en).
- Igenkänning på kategorinivå: att bestämma vilken objektkategori som avbildas. Tävlingen PASCAL Visual Object Classes Challenge 2012 (VOC 2012) [64]

bestod exempelvis av bilder från 20 olika kategorier (bil, buss, båt, cykel, flaska, flygplan, fågel, får, hund, häst, katt, ko, krukväxt, matbord, motorcykel, person, soffa, stol, TV, tåg).

- Finkornig igenkänning: att särskilja varianter inom samma kategori. I tävlingen Fine-Grained Challenge 2013 (FGCOMP) [65] skulle man skilja mellan 196 bilmodeller, 100 flygplansmodeller, 83 fågelarter, 120 hundraser, och 70 skomodeller. Ansiktsigenkänning är den främsta kommersiella tillämpningen inom detta område.

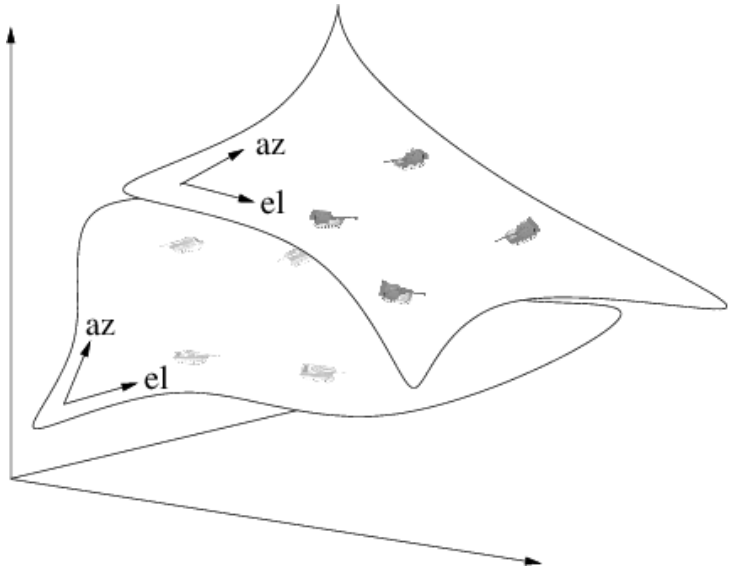
Klassificering är ett beslutsproblem. Givet en signal (t.ex. bild) $\mathbf{x}$ ska vi välja en klass $k \in \{1,\ldots,K\}$. Om vi önskar minimera sannolikheten för felklassificering är det bästa val vi kan göra att välja den klass som har störst aposteriorisannolikhet, dvs.

$$k(x) = \arg \max_{l \in \{1,\ldots,K\}} P(c = l \,|\, \mathbf{x}).$$

För att skatta modeller av aposteriorisannolikheterna använder vi en uppsättning träningsexempel $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$, där $y_i \in \{1,\ldots,K\}$ anger klasstillhörigheten. Träningsdata skall vara ett representativt slumpmässigt sampel från täthetsfunktionen $p(\mathbf{x}\,|\,c)$.

3.2 Pixelrummet och klassernas fördelning

En digitalkamera producerar bilder uppbyggda av en stor mängd bildpunkter eller *pixlar*. Om vi radar upp alla bildpunkter efter varandra kan vi betrakta en bild som en vektor motsvarande en punkt i ett mycket högdimensionellt euklidiskt rum, *pixelrummet*. Om vi håller oss på ett visst avstånd från ett objekt och fotograferar det från alla möjliga orienteringar får vi en uppsättning punkter i pixelrummet som bildar en sammanhängande yta eller *mångfald* (figur 2). I den beskrivna situationen är mångfaldens dimensionalitet två, dvs. det behövs två parametrar (azimut och elevation) för att lokalisera en punkt. Om vi upprepar processen med ett annat objekt fås en mångfald som är separerad från den första, utom i punkter motsvarande orienteringar i vilka objekten ser likadana ut. Om vi är utomhus och upprepar proceduren vid olika tidpunkter fås en mångfald av högre dimensionalitet, beroende på att solens rörelse ger upphov till olika belysningssituationer. Vi kan introducera ännu flera variationer genom exempelvis artikulationer som vridningar av kanontorn eller olika poser hos en människa.



Figur 2. I pixelrummet, där varje bild är en punkt, ligger bilder av ett enskilt objekt på en krökt yta eller mångfald. Mångfalder tillhörande olika objekt överlappar i punkter som motsvarar vyer där objekten ser likadana ut. Varje punkt på en mångfald karakteriseras av objektidentitet och en uppsättning parametervärden, t.ex. orientering (i bilden azimut och elevation för kameran).

En klass av objekt motsvarar en samling mångfalder som inte nödvändigtvis ligger nära varandra i pixelrummet. Aposteriorifördelningen är för varje klass en funktion som har värdet 1 på de till klassen hörande mångfalderna (lägre i punkter med överlapp till andra klasser) och noll överallt annars. Om träningsexemplen kan antas vara ett slumpmässigt sampel är en naturlig tanke att skatta aposteriorifördelningen $P(c|\mathbf{x})$ i punkten $\mathbf{x}$ genom att finna dess närmaste grannar bland träningsdata och bestämma hur stor andel av dessa exempel som tillhör respektive klass. En sådan ansats är dock inte beräkningsmässigt möjlig för högdimensionella signaler.

En intressant egenskap hos den enskilda objektmångfalden är dess låga dimensionalitet i förhållande till pixelrummet. Den är dock i hög grad olinjär (krökt) och mångfalder hörande till olika objekt är sammanflätade (dock, som sagt, utan att beröra varandra utom i undantagsfall). Lokalt kan dock strukturen vara enklare. Ramamoorthi [66] har visat att för en fix vy med varierande belysning (allt annat lika) ligger mångfalden för en konvex matt yta till största delen i ett 5-dimensionellt linjärt rum, vilket bekräftar tidigare empiriska observationer [67]. Resultatet gäller approximativt även för icke-konvexa objekt som ansikten. Olika objekt ligger i olika underrum vilket kan utnyttjas för att formulera en igenkänningsmetod. En uppsättning bilder tas av varje objekt från samma vy men under olika belysningar. För varje objekt bildas en matris $\mathbf{A}$ där kolumnerna är bilderna i vektorform, och sedan beräknas singulärvärdesuppdelningen

$$\mathbf{A} = \sum_{k=1}^{d} s_k \mathbf{u}_k \mathbf{v}_k^T, \qquad s_1 \geq \cdots \geq s_d \geq 0, \;\; \mathbf{u}_k^T \mathbf{u}_l = \delta_{k,l}, \;\; \mathbf{v}_k^T \mathbf{v}_l = \delta_{k,l}.$$

Man finner då att

$$\mathbf{A} \approx \sum_{k=1}^{5} s_k \mathbf{u}_k \mathbf{v}_k^T.$$

Vektorerna $\mathbf{u}_k$, $k=1,\ldots,5$ bildar en ortogonal bas till det sökta underrummet. Det kvadratiska avståndet från en ny bild $\mathbf{b}$ (från samma vy) till ett träningsexempel $\mathbf{t}$ kan nu uttryckas som en summa

$$\|\mathbf{b}-\mathbf{t}\|^2 = \|\mathbf{U}^T\mathbf{b}-\mathbf{U}^T\mathbf{t}\|^2 + \|\mathbf{b}-\mathbf{U}\mathbf{U}^T\mathbf{b}\|^2$$

där $\mathbf{U}=[\mathbf{u}_1,\ldots,\mathbf{u}_5]$. Den första termen är avståndet mellan punkternas projektioner i underrummet och den andra termen är avståndet från $\mathbf{b}$ till underrummet (för träningsexemplet antas detta avstånd vara noll). Eftersom den första avståndsberäkningen nu sker i ett lågdimensionellt rum kan man effektivt implementera närmaste-granneklassificering. Den här typen av metodik var mycket vanlig inom ansiktsigenkänning under 90-talet och tidiga 00-talet [68].

En iakttagelse som i viss mening är komplementär till ovanstående beskrevs först av Ullman och Basri [69]. Istället för objektytans utseende under olika belysningar handlar det här om ett lågdimensionellt linjärt geometriskt samband mellan punkter i olika vyer. Låt $\mathrm{P}=\{\mathbf{P}^1,\ldots,\mathbf{P}^N\}$ vara en uppsättning punkter på ett 3D-objekt och avbilda objektet med en affin projektion (dvs. avståndet till objektet är väsentligt större än objektets utbredning i djupled). Låt vidare $\mathbf{P}^0$ vara en godtycklig annan punkt på objektet eller tyngdpunkten för P. För tre olika kamerapositioner projiceras $\mathbf{P}^i$ till bildkoordinaterna $\mathbf{p}_1^i$, $\mathbf{p}_2^i$, respektive $\mathbf{p}_3^i$. Bilda för varje punkt och kamera skillnadsvektorn $\boldsymbol{\delta}_k^i \equiv \mathbf{p}_k^i - \mathbf{p}_k^0$. Då gäller för en ny kameraposition k':

$$\boldsymbol{\delta}_{k'}^i = \begin{pmatrix} a_1^{k'} & 0 \\ 0 & b_1^{k'} \end{pmatrix}\boldsymbol{\delta}_1^i + \begin{pmatrix} a_2^{k'} & 0 \\ 0 & b_2^{k'} \end{pmatrix}\boldsymbol{\delta}_2^i + \begin{pmatrix} a_3^{k'} & 0 \\ 0 & b_3^{k'} \end{pmatrix}\boldsymbol{\delta}_3^i, \quad i=1,\ldots,N\,.$$

Givet punktkorrespondenser över tre vyer är alltså punkternas koordinater i en godtycklig fjärde vy en linjärkombination av koordinaterna i de tre tidigare. Koefficienterna a_m^k, b_m^k, $m=1,2,3$ kan bestämmas genom minstakvadratmetoden eller, om felaktiga korrespondenser kan förekomma, genom RANSAC eller liknande robust metodik. Minst fyra punktkorrespondenser över de fyra vyerna krävs för en entydig lösning. Användningen av sambandet mellan vyer för objektigenkänning är i princip rättfram. För varje objekt i en databas lagras en uppsättning punktkorrespondenser över tre väl separerade vyer. För en ny bild söker man efter möjliga matchningar till punkterna, och skattar därefter koefficienterna i linjärkombinationen så som beskrivits ovan. Bilden klassificeras som det objekt vars modell ger minsta approximationsfel. Metoden kan generaliseras till allmänna perspektivprojektioner genom användning av den så kallade trifokala tensorn [48], som är en multilinjär relation mellan tre vyer.

Active appearance models [70] kan betraktas som en syntes av de båda ansatserna ovan. I denna metodik som var populär kring millennieskiftet identifierar man ett antal punktkorrespondenser (*landmarks*) i en uppsättning träningsbilder (i ett ansikte exempelvis ögon, nästipp och mungipor). Punktuppsättningarna i de olika bilderna translateras, skalas och roteras så att de är så lika som möjligt. Därefter skattas en lågdimensionell linjär modell som beskriver de återstående formvariationerna. Bilderna transformeras sedan geometriskt (*warping*) så att punkterna flyttas till medelvärdena i uppsättningen träningsdata. I denna ”formnormaliserade” ram skattas sedan en lågdimensionell linjär modell för variationerna i bildernas utseende. Vid matchning av en ny bild används en iterativ process för att skatta koefficienterna som ger bästa anpassning av modellen med avseende på form och utseende.

3.3 Generativa respektive diskriminerande ansatser

3.3.1 Generativa metoder

Som alternativ till närmaste-granneklassificering kan man i exemplet ovan med belysningsunderrummet skatta en parametrisk modell för täthetsfunktionen

$$p(\mathbf{x}|c=k) = p_{1,k}(\mathbf{U}_k^T\mathbf{x}) \times p_{2,k}(\mathbf{x}-\mathbf{U}_k\mathbf{U}_k^T\mathbf{x})$$

Här kan $p_{2,k}$ vara en isotrop normalfördelning $N(\mathbf{0},\sigma^2\mathbf{I})$ och $p_{1,k}$ en Gaussblandning. Klassificeringsmetoder där man modellerar klassernas täthetsfunktioner brukar kallas *generativa*. Bayes sats ger sambandet

$$P(c=k|\mathbf{x}) = \frac{p(\mathbf{x}|c=k)P(c=k)}{\sum_{l=1}^{K} p(\mathbf{x}|c=l)P(c=l)}$$

Här är $P(c=k)$ apriorisannolikheten för klass k.

3.3.1.1 Diskriminantfunktioner

Istället för att gå omvägen och för varje klass skatta en modell av signalen kan man direkt modellera aposteriorisannolikheterna $P(c|\mathbf{x})$ från träningsdata. Ofta har de en mycket enklare struktur än signalen själv. Idealt ska de vara konstanta och lika med 1 på det aktuella objektets mångfald, och noll utanför denna. I själva verket kan vi generalisera detta till att inkludera funktioner som har ett högt värde i objektets mångfald, och ett lågt värde överallt annars. Funktioner δ_k med denna egenskap kallas ofta diskriminantfunktioner. Klassificeringsbeslutet blir

$$k(\mathbf{x}) = \arg\max_{l\in\{1,\ldots,K\}} \delta_l(\mathbf{x})$$

Om vi, i regel helt orealistiskt, i en generativ ansats antar att samtliga klasser är normalfördelade med identiska kovariansmatriser, dvs.

$$p(\mathbf{x}|c=k) = (\det 2\pi\boldsymbol{\Sigma})^{-1/2} \exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_k))$$

fås enligt Bayes sats ovan

$$\log\frac{P(c=k|\mathbf{x})}{P(c=l|\mathbf{x})} = (\boldsymbol{\mu}_k-\boldsymbol{\mu}_l)^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \boldsymbol{\mu}_l^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_l + \log\frac{P(c=k)}{P(c=l)}$$

så att vi kan definiera de linjära diskriminantfunktionerna

$$\delta_k(\mathbf{x}) = \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \log P(c=k), \quad k=1,\ldots,K$$

Parametrarna kan skattas från en uppsättning träningsexempel för varje klass. $\boldsymbol{\Sigma}$ beräknas som ett aprioriviktat medelvärde av de klasspecifika kovariansmatriserna. Klassificeringsmetoden kallas *linjär diskriminantanalys* (LDA).

Istället för att som i LDA göra specifika antaganden om klassernas fördelningar kan vi direkt ansätta en linjär form för logkvoten ovan, dvs.

$$\log\frac{P(c=k|\mathbf{x})}{P(c=K|\mathbf{x})} = \mathbf{w}_k^T\mathbf{x} + b_k, \quad k=1,\ldots,K-1$$

Vi behöver bara $K-1$ modeller eftersom sannolikheterna ska summera till 1; valet av vilken klass som ska vara i nämnaren är godtyckligt. Det följer att

$$P(c=k\,|\,\mathbf{x})=\frac{\exp(\mathbf{w}_k^T\mathbf{x}+b_k)}{1+\sum_{l=1}^{K-1}\exp(\mathbf{w}_l^T\mathbf{x}+b_l)},\qquad k=1,\ldots,K-1$$

Parametrarna kan skattas från en uppsättning träningsexempel $\{\mathbf{x}_i,y_i\}_{i=1}^N$, där $y_i\in\{1,\ldots,K\}$ är klasstillhörigheten, genom att maximera log-likelihoodfunktionen

$$l(\mathbf{w}_1,\ldots,\mathbf{w}_{K-1},b_1,\ldots,b_{K-1})=\sum_{i=1}^N\log P(c=y_i\,|\,\mathbf{x}_i;\mathbf{w}_{y_i},b_{y_i})$$

l är konkav, så gradientsökning leder till globala maximum. Klassificeringsmetoden kallas (*multinomial) logistisk regression*.

En linjär klassificerare som genom sin goda generaliseringsförmåga blivit mycket populär är *supportvektormaskinen* [71]. Den är definierad för diskriminering mellan två klasser (binär klassificering) och placerar ett hyperplan mitt emellan klasserna, så att beslutsregeln blir

$$k(\mathbf{x})=\arg\max_{l\in\{1,2\}}s_l(\mathbf{w}^T\mathbf{x}+b),\quad\text{där}\quad s_1=+1,\ s_2=-1$$

Hyperplanet definieras av $\mathbf{w}^T\mathbf{x}+b=0$, och om vi vill att det ska placeras mitt emellan klasserna kan vi kräva att $\mathbf{w}^T\mathbf{x}+b\geq 1$ för samtliga träningsexempel från klass 1 och att $\mathbf{w}^T\mathbf{x}+b\leq -1$ för klass 2. Avståndet mellan klasserna längs riktningen $\mathbf{w}$ är $2/\|\mathbf{w}\|$, så för att finna den orientering som maximerar separationen mellan klasserna ska vi lösa problemet

$$\min_{\mathbf{w},b}\mathbf{w}^T\mathbf{w},\quad\text{så att}\quad y_i(\mathbf{w}^T\mathbf{x}_i+b)\geq 1,\quad i=1,\ldots,N$$

Här är $(\mathbf{x}_i,y_i)_{i=1}^N$ en uppsättning träningsexempel där $y_i=1$ för exempel från klass 1 (”positiver”) och $y_i=-1$ för exempel från klass 2 (”negativer”). Detta är ett kvadratiskt optimeringsproblem med linjära bivillkor som kan lösas effektivt. Notera att problemets lösning endast beror av de exempel som ligger närmast beslutsplanet, de så kallade supportvektorerna. I allmänhet kan klasserna inte separeras med ett hyperplan och problemet ovan saknar lösning. Vi kan dock generalisera problemet genom att tillåta ett antal exempel att hamna innanför den önskade separationen,

$$\min_{\mathbf{w},b,\boldsymbol{\xi}}\mathbf{w}^T\mathbf{w}+C\sum_{i=1}^N\xi_i,\quad\text{så att}\quad y_i(\mathbf{w}^T\mathbf{x}_i+b)\geq 1-\xi_i,\quad\xi_i\geq 0,\quad i=1,\ldots,N$$

Här är $C>0$ en regulariseringsparameter vars värde kan bestämmas genom att utvärdera klassificeringsresultatet mot en separat uppsättning testdata eller mer systematiskt genom korsvalidering. En mycket viktig egenskap hos supportvektormaskinen (och även LDA) är att lösningen till optimeringsproblemet är en linjärkombination av träningsexemplen, vilket medför att diskriminantfunktionen kan uttryckas som en viktad summa av skalärprodukter $\mathbf{x}_i^T\mathbf{x}_j$ mellan signalvektorer:

$$\mathbf{w}=\sum_{i=1}^N\alpha_i y_i\mathbf{x}_i\Rightarrow\mathbf{w}^T\mathbf{x}+b=\sum_{i=1}^N\alpha_i y_i\mathbf{x}_i^T\mathbf{x}+b$$

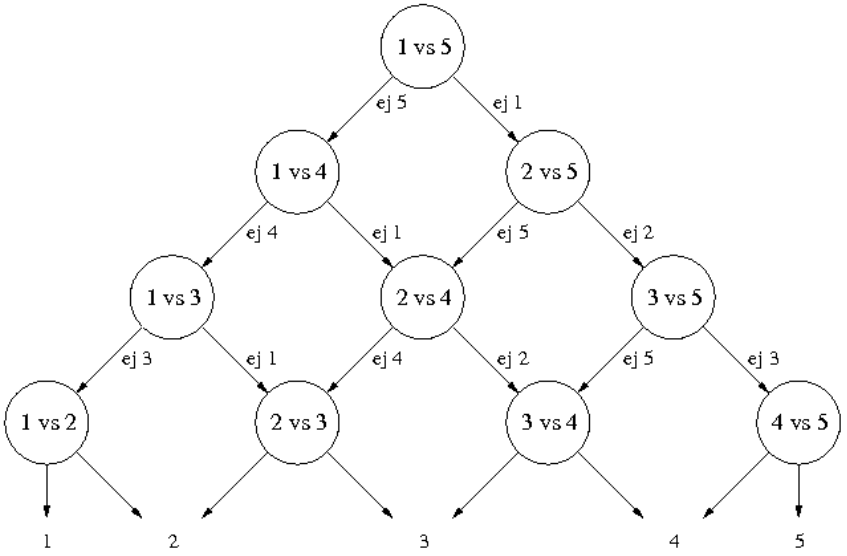
Det betyder att om vi avbildar signalen med en olinjär avbildning $\Phi:\mathbb{R}^n\to\mathrm{H}$ och kan uttrycka skalärprodukter i detta rum på ett kompakt sätt $\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)=k(\mathbf{x}_i,\mathbf{x}_j)$ kan vi finna ett separerande hyperplan i rummet H och evaluera diskriminantfunktionen utan att explicit beräkna avbildningen $\Phi(\mathbf{x})$ som kan vara mycket högdimensionell. Ett par populära så kallade kärnor är polynom- respektive Gaussfunktionerna

$$k(\mathbf{x},\mathbf{x}') = (\mathbf{x}^T\mathbf{x}' + c)^d,\ d \in \mathbb{N},\ c \geq 0 \qquad k(\mathbf{x},\mathbf{x}') = \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2})$$

För polynomfallet är H:s dimensionalitet $(d+n)!/(d!n!)$ och för Gausskärnorna oändligt. Förhoppningen är att signalfördelningarna ska vara mer lättseparerade i det högdimensionella rummet H än i pixelrummet.

För att särskilja fler än två klasser med en supportvektormaskin eller annan binär klassificerare finns flera alternativ:

- En mot övriga: För varje klass tränas en diskriminantfunktion att skilja den aktuella klassen från övriga. Ett testexempel tilldelas den klass vars diskriminantfunktion ger störst värde.
- En mot en (parvis): För varje par av klasser tränas en klassificerare. Ett testexempel tilldelas den klass som vinner flest parvisa tester. Mer sofistikerade sammanvägningsmetoder [72] kan formuleras om varje klassificerare ger en skattning av den betingade sannolikheten $P(c = k \,|\, \mathbf{x}, c = k \text{ eller } c = l)$. För K klasser krävs $K(K-1)/2$ klassificerare, vilket kan bli beräkningsmässigt dyrt.
- Decision Directed Acyclic Graph (DDAG) [73]: Ett beräkningsmässigt effektivt sätt att organisera och evaluera parvisa tester, se figur 3.
- Felrättande koder: (eng. *error-correcting output coding* (ECOC) [74]. Varje klass representeras med en unik binär sträng (samtliga av samma längd) på ett redundant sätt, så att Hammingavståndet (antalet skilda bitar) mellan par av klasser är större än 1. Om kodorden består av B bitar tränas sedan lika många binära klassificerare, där klassificerare b tränas att separera de klasser vars kod har en etta i motsvarande position från de som där har en nolla, se tabell 2. Testexempel klassificeras sedan av samtliga binära klassificerare, vilket genererar en B bitar lång sträng. Exemplet tilldelas sedan den klass vars kodord ligger på minsta Hammingavståndet från strängen.



Figur 3. DDAG grafstruktur för effektiv evaluering av parvisa tester.

Tabell 2. ECOC. 15-bitars kodord för ett problem med 10 klasser. Varje kolumn definierar ett binärt klassificeringsproblem. Redundansen i kodningen ger robusthet mot klassificeringsfel. Efter [74].

KLASS	b_1	b_2	b_3	b_4	b_5	b_6	$\cdots$	b_{15}
1	1	1	0	0	0	0	$\cdots$	1
2	0	0	1	1	1	1	$\cdots$	0
3	1	0	0	1	0	0	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
9	1	1	0	1	0	1	$\cdots$	1
10	0	1	1	1	0	0	$\cdots$	0

Vi ska introducera ytterligare en grundläggande komponent för att bygga en kraftfull klassificerare, nämligen *klassificeringsträdet*. Det finns flera olika metoder att konstruera ett sådant, men här begränsar vi oss till CART (eng. *Classification And Regression Tree*) [75]. Trädet byggs rekursivt, från en ensam rotnod. Vi söker då en funktion $s:\mathbb{R}^n \to \mathbb{R}$, kallad splittringsfunktion, och ett tröskelvärde τ, så att en uppdelning av träningsexemplen i två grupper

$$l_1 = \{(\mathbf{x}_i, y_i) : s(\mathbf{x}_i) \le \tau\}, \qquad l_2 = \{(\mathbf{x}_i, y_i) : s(\mathbf{x}_i) > \tau\}$$

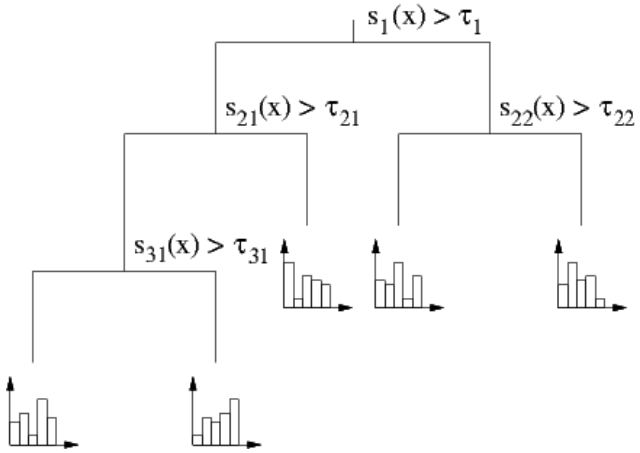
ger en ökad separation av klasserna. Vi kan uttrycka denna separation i termer av entropi (andra mått är möjliga), och skriva minskningen av entropi som

$$\Delta H = H_0 - \frac{N_1}{N_1 + N_2} H_1 - \frac{N_2}{N_1 + N_2} H_2$$

Där N_i anger antalet exempel i respektive grupp, och

$$H_i = -\sum_{k=1}^{K} \hat{p}_{i,k} \log \hat{p}_{i,k}$$

Här är $\hat{p}_{i,k}$ andelen exempel i grupp i från klass k. Ett vanligt val av splittringsfunktion är helt enkelt en komponent av signalvektorn. Man testar då samtliga komponenter och söker för var och en av dessa efter ett tröskelvärde som maximerar entropiminskningen. När den bästa (s, τ)-kombinationen identifierats placeras denna i rotnoden och träningsexemplen flyttas till respektive dotternod (löv). För var och en av dessa upprepas proceduren och datamängden splittrar återigen, och så vidare tills ett stoppkriterium uppfylls, t.ex. att maximalt träddjup uppnåtts eller att antal exempel i ett löv nått ett minimum. För att klassificera ett testexempel $\mathbf{t}$ börjar man i rotnoden och beräknar värdet $s(\mathbf{t})$ och jämför det med tröskelvärdet τ. Beroende på utfallet flyttas exemplet sedan till vänster eller höger dotternod, där proceduren upprepas tills ett löv nås. I lövet finns ett normerat histogram över klasstillhörigheten för de träningsexempel som hamnat där. Detta tas som en skattning av aposteriorifördelningen $P(c|\mathbf{t})$. Ett schematiskt exempel visas i figur 4.



Figur 4. Klassificeringsträd för 5 klasser. I varje löv lagras ett normaliserat histogram över klasstillhörigheten för de träningsexempel som hamnat där.

3.4 Arbetsgång vid klassificering

Det har visat sig svårt att med framgång träna de klassificerare som beskrivits ovan på bilddata i originalformat. Istället har en kedja av transformationer etablerats där data successivt görs allt mer tolerant mot olika slags variationer, för att slutligen kunna klassificeras, se figur 5. Följande moment ingår [76,82]:

1. Sampling av lokala omgivningar
2. Förbehandling
3. Representation av lokala omgivningar
4. Kodning av lokala deskriptorer
5. Spatiell sammanläggning av kodvektorer
6. Klassificering

3.4.1.1 Sampling av lokala omgivningar

Samplingen av omgivningar kan ske tätt (dock ofta med en steglängd större än 1) eller glest, där en *intresseoperator* [77] appliceras för att detektera omgivningar som är informativa (diskriminerande) och stabila (kan detekteras även om avstånd och orientering förändras); typiska exempel är hörnpunkter.

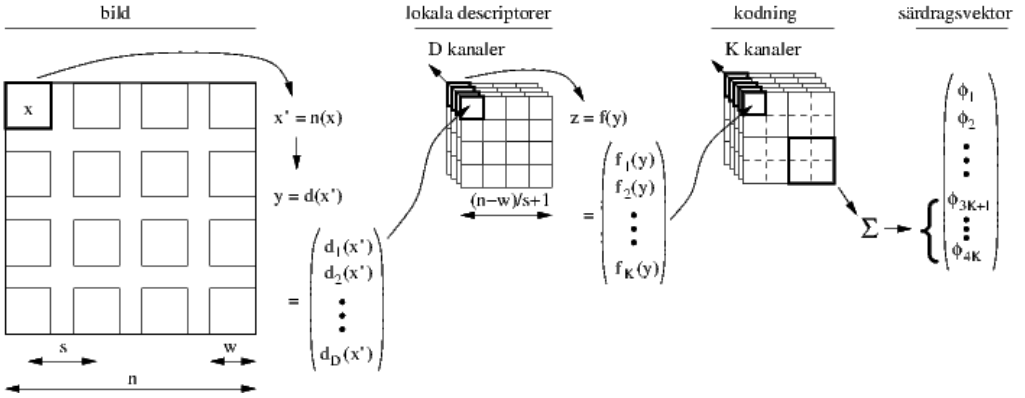
3.4.1.2 Förbehandling

Variationer i omgivningarnas intensitet och kontrast kan kompenseras för genom att normalisera pixelvärdena med avseende på medelvärde och standardavvikelse.

3.4.1.3 Representation av lokal omgivning

Omgivningen representeras matematiskt vanligen med en vektor. Önskvärda egenskaper hos deskriptorn är tolerans mot variationer i belysning, orientering, och smärre skift. Invarians mot orientering i planet kan åstadkommas genom att bestämma den dominanta gradientriktningen i omgivningen och använda denna som en referens (denna rotation är endast nödvändig om stora orienteringsvariationer kan förväntas). Om deskriptorn är en funktion av intensitetsgradienter fås automatiskt en okänslighet mot intensitetsvariationer, medan kontrastnormalisering kan åstadkommas genom att dividera med den genomsnittliga magnituden för gradienten.

Populära deskriptorer med goda egenskaper är exempelvis Gaborfilter [78], SIFT *(Scale Invariant Feature Transform*) [32], SURF (*Speeded-Up Robust Features*) [79] och HOG (*Histogram of Oriented Gradients*) [80]. SIFT och HOG (SURF är en förenklad variant) delar upp omgivningen i mindre områden som representeras med ett magnitudviktat histogram över gradientriktningar. Deskriptorn fås sedan genom att konkatenera histogrammen till en vektor, som normaliseras. En grov uppdelning i riktningsintervall (typiskt används 8 eller 9) ger viss tolerans mot orienteringsvariationer, medan histogrammen ger tolerans mot mindre spatiella skift. Uppdelningen i delregioner bibehåller en känslighet för omgivningens form. Deskriptorer som dessa, vilka beskriver fördelningen av kanter i omgivningen (”form”), används ofta i kombination med sådana som beskriver andra aspekter, typiskt textur (t.ex. LBP, *local binary patterns* [34**]**) eller färg (t.ex. färghistogram [81]).



Figur 5. Typisk bearbetningskedja vid klassificering. Ur bilden samplas lokala $w \times w$-omgivningar x med en steglängd s. Varje omgivning förbehandlas $x \to x'$ (t.ex. intensitets- och kontrastnormalisering), varefter en vektoriell deskriptor $d(x')$ beräknas. Deskriptorn kodas sedan, vilket innebär en olinjär avbildning $f : \mathbb{R}^D \to \mathbb{R}^K$ som typiskt har egenskapen att varje deskriptor y motsvaras av en gles kodvektor $f(y)$, dvs. endast en eller några få komponenter är samtidigt nollskilda. I nästa steg görs en spatiell sammanläggning (eng. pooling) av kodvektorer i en större lokal omgivning, vanligen genom komponentvis medelvärdesbildning eller maximering. Slutligen konkateneras de sammanlagda kodvektorerna till en global särdragsvektor som blir indata till en klassificerare. (Steglängden s är vanligen mindre än omgivningsbredden w, så att omgivningarna överlappar). Efter [82].

3.4.1.4 Kodning av lokal deskriptorer

Nästa steg i bearbetningskedjan är en transformation till ett vektorrum där varje bildomgivning representeras av en vektor där endast ett fåtal komponenter är (signifikant) nollskilda, så kallad *gles kodning*. Detta medför i regel inte en reduktion av signalens dimensionalitet, utan tvärtom kan den öka väsentligt.

Varför gles kodning? Som beskrivits ovan ligger bilderna av ett objekt i en lågdimensionell mångfald i pixelrummet. Mångfalder för olika objekt är krökta och insnärjda i varandra, vilket gör det omöjligt att med en enkel klassificerare separera dem utom i mycket kontrollerade situationer. Målet är att transformera bilden till en representation där objekten är mer tydligt separerade. Vi behandlar i detta steg ännu lokala omgivningar av bilden, dvs. projektioner på jämförelsevis lågdimensionella underrum, där vi inte kan räkna med att kunna associera ett givet mönster till ett specifikt objekt. Men det är ändå möjligt att urskilja distinkta kategorier av mönster och att effektivt modellera hur

de transformeras. Genom att koda olika mönster i separata dimensioner underlättas association i senare steg av samtidiga lokala strukturer över större områden.

Vektorkvantisering

Ett enkelt sätt att skapa en gles representation är genom *vektorkvantisering*, vilket kan åstadkommas via den välkända *k-meansalgoritmen* [83]. Målet för vektorkvantisering är att partitionera signalrummet i K delar och representera signaler i en sådan del med en prototypvektor $\boldsymbol{\mu}_k$, som väljs så att det genomsnittliga kvadratiska approximationsfelet minimeras. Givet ett sampel $\mathbf{y}_1,\ldots,\mathbf{y}_N$ finner k-meansalgoritmen ett lokalt minimum till approximationsfelet

$$E(\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_K)=\sum_{k=1}^{K}\sum_{i:\mathbf{y}_i\in S_k}\left\|\mathbf{y}_i-\boldsymbol{\mu}_k\right\|^2,\qquad \text{där}\quad S_k=\{\mathbf{y}:k=\arg\min_l\left\|\mathbf{y}-\boldsymbol{\mu}_l\right\|\}$$

Man inser lätt att för en given region S_k ska prototypen $\boldsymbol{\mu}_k$ väljas som medelvärdet av samplen i regionen. Vektorkvantisering ger en maximalt gles representation, där varje signal representeras av en binär vektor $f(\mathbf{y})=[0,\ldots,0,1,0,\ldots,0]$. Coates *et al*. [82] föreslår istället

$$f_k(\mathbf{y})=\max\{0,-\left\|\mathbf{y}-\boldsymbol{\mu}_k\right\|+\frac{1}{K}\sum_{l=1}^{K}\left\|\mathbf{y}-\boldsymbol{\mu}_l\right\|\}$$

vilket blir 0 för alla komponenter vars prototyp är på större avstånd från $\mathbf{y}$ än genomsnittet. Detta visar sig ge väsentligt bättre klassificeringsresultat. Författarna visar också att det är mycket gynnsamt att dekorrelera signalens komponenter innan vektorkvantiseringen. Detta betyder att man transformerar signalen linjärt

$$\mathbf{y}'=\mathbf{C}^{-1/2}(\mathbf{y}-\boldsymbol{\mu})$$

där $\mathbf{C}$ är fördelningens kovariansmatris och $\boldsymbol{\mu}$ dess medelvärde.

Gles basutveckling

En på senare år mycket populär formulering av gles kodning är att man, givet en redundant basuppsättning $\mathbf{B}=[\mathbf{b}_1,\ldots,\mathbf{b}_K]$ söker en koefficientuppsättning som minimerar

$$E(\mathbf{c}_i)=\left\|\mathbf{y}_i-\mathbf{B}\mathbf{c}_i\right\|^2+\lambda\sum_{k=1}^{K}\left|c_{i,k}\right|^n,\qquad \lambda>0,\qquad n\in[0,1]$$

För $n=0$ är summan antalet nollskilda koefficienter, och en ofta bra approximativ lösning kan fås genom en girig algoritm, *Orthogonal Matching Pursuit* (OMP) [84], som successivt väljer en bas $\mathbf{B}_k=[\mathbf{b}_{(1)},\ldots,\mathbf{b}_{(k)}]$ och en koefficientvektor $\mathbf{c}_i^k$ som minimerar $\left\|\mathbf{y}_i-\mathbf{B}_k\mathbf{c}_i^k\right\|$. För $n=1$ brukar problemet kallas *Basis Pursuit Denoising* [85] eller *Lasso* [86] och den optimala koefficientvektorn kan bestämmas exakt genom kvadratisk programmering.

Givet en uppsättning träningsexempel bestäms basen $\mathbf{B}$ genom minimering av

$$E(\mathbf{c}_1,\ldots,\mathbf{c}_N,\mathbf{B})=\sum_{i=1}^{N}(\left\|\mathbf{y}_i-\mathbf{B}\mathbf{c}_i\right\|^2+\lambda\sum_{k=1}^{K}\left|c_{i,k}\right|^n).$$

Detta problem kan inte lösas exakt vare sig för $n=0$ eller $n=1$, men det finns dock flera approximativa metoder. För det förra fallet finns exempelvis *K-SVD* [87] som ite-

rativt växlar mellan att uppdatera koefficientvektorerna givet en fix bas $\mathbf{B}$, och att uppdatera basen givet en fix uppsättning koefficienter. Lee *et al*. [88] presenterar en algoritm för fallet $n=1$, liksom Mairal *et al*. [89] vars metod kan hantera en ström (sekvens) av observationer $\mathbf{y}_i$.

En begränsning i ovanstående standardformulering av gles basutveckling är att ingen hänsyn tas till lokalitet i signalrummet, vilket betyder att närliggande signaler kan få helt olika koder. Yu *et al*. [90] förslår *Local Coordinate Coding (LCC)* där följande optimeringskriterium används

$$E(\mathbf{c}_1,\ldots,\mathbf{c}_N,\mathbf{B})=\sum_{i=1}^{N}(\|\mathbf{y}_i-\mathbf{B}\mathbf{c}_i\|^2+\lambda\sum_{k=1}^{K}\|\mathbf{y}_i-\mathbf{b}_k\|^2\,|c_{i,k}|),$$

Regulariseringstermen har nu för varje koefficient $c_{i,k}$ fått en vikt som tydligt uppmuntrar val av basfunktioner nära signalen $\mathbf{y}_i$. I [91] utvidgas metoden till att, efter att basen $\mathbf{B}$ bestämts, genom principalkomponentanalys (PCA) med viktade data $c_{i,k}(\mathbf{y}_i-\mathbf{b}_k)$, $i=1,\ldots,N$, skatta ett lokalt linjärt rum kring varje basvektor $\mathbf{b}_k$. En signal kodas sedan med koefficientvektorn $\mathbf{c}(\mathbf{y})$ och projektionerna $c_k(\mathbf{y}-\mathbf{b}_k)^T\mathbf{U}_k$, $k=1,\ldots,K$, där kolumnerna till $\mathbf{U}_k$ bildar en ortogonal bas till det lokala underrummet. Wang *et al*. [92] presenterar *Locality-constrained Linear Coding (LLC)* där man, givet en redundant bas, för varje signal approximerar denna genom linjär interpolation mellan ett fåtal basvektorer som är *närmaste grannar* till signalen. Basen initialiseras genom k-meansalgoritmen och förfinas sedan iterativt så att de lokala approximationerna av träningsexemplen förbättras. Denna kodningsmetod har gett goda resultat vid klassificering.

Fishervektor

Vi ska nu beskriva en kodningsmetod som visat sig mycket framgångsrik inom klassificering [93]. Den bygger på att en parametriserad probabilistisk modell $p(\mathbf{y}|\boldsymbol{\theta})$ skattats för signalfördelningen. Fishers scorefunktion ges av

$$\mathbf{f}(\boldsymbol{\theta};\mathbf{y})=\frac{\partial\log p(\mathbf{y}|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}$$

och är ett mått på hur känslig sannolikhetsvärdet är för ändringar i parametervektorn $\boldsymbol{\theta}$. Man kan visa att den riktning i parameterrummet i vilken sannolikheten för en observation $\mathbf{y}$ snabbast ökar ges av [94]

$$\tilde{\mathbf{f}}(\boldsymbol{\theta};\mathbf{y})=\mathbf{G}_\theta^{-1}\mathbf{f}(\boldsymbol{\theta};\mathbf{y})$$

vilket kallas den *naturliga gradienten*. Här är

$$\mathbf{G}_\theta=\int_Y\mathbf{f}(\boldsymbol{\theta};\mathbf{y})[\mathbf{f}(\boldsymbol{\theta};\mathbf{y})]^T\,p(\mathbf{y}|\boldsymbol{\theta})d\mathbf{y}$$

Fishers informationsmatris. $\mathbf{G}_\theta$ definierar en metrik och vi kan jämföra två olika observationer genom att ta skalärprodukten mellan deras respektive naturliga gradienter

$$\begin{aligned}k(\mathbf{y}_i,\mathbf{y}_j)&=[\tilde{\mathbf{f}}(\boldsymbol{\theta};\mathbf{y}_i)]^T\mathbf{G}_\theta\tilde{\mathbf{f}}(\boldsymbol{\theta};\mathbf{y}_j)=[\mathbf{f}(\boldsymbol{\theta};\mathbf{y}_i)]^T\mathbf{G}_\theta^{-1}\mathbf{f}(\boldsymbol{\theta};\mathbf{y}_j)\\&=[\mathbf{L}_\theta^T\mathbf{f}(\boldsymbol{\theta};\mathbf{y}_i)]^T\mathbf{L}_\theta^T\mathbf{f}(\boldsymbol{\theta};\mathbf{y}_j)\equiv[\hat{\mathbf{f}}(\boldsymbol{\theta};\mathbf{y}_i)]^T\hat{\mathbf{f}}(\boldsymbol{\theta};\mathbf{y}_j)\end{aligned}$$

där $\mathbf{L}_\theta \mathbf{L}_\theta^T = \mathbf{G}_\theta^{-1}$ är en Choleskyuppdelning. Jaakkola och Haussler [95] kallar skalärprodukten *Fisherkärna*, med tänkt användning i en supportvektormaskin. Sánchez *et al*. [96] kallar $\hat{\mathbf{f}}(\boldsymbol{\theta};\mathbf{y})$ *Fishervektor* och använder den för kodning av deskriptorer. Författarna modellerar deskriptorfördelningen med en Gaussblandning

$$p(\mathbf{y}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{y}), \qquad p_k(y) = [\det 2\pi\boldsymbol{\Sigma}_k]^{-1/2} \exp(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{y}-\boldsymbol{\mu}_k))$$

med antagande om diagonala kovariansmatriser $\boldsymbol{\Sigma}_k = \mathrm{diag}(\sigma_{k,1}^2,\ldots,\sigma_{k,D}^2)$ och skattar dess parametrar från träningsdata. För varje komponent i blandningen finns då tre parametrar: komponentproportionen π_k, medelvärdet $\boldsymbol{\mu}_k$, och varianserna $\boldsymbol{\sigma}_k^2 \equiv [\sigma_{k,1}^2,\ldots,\sigma_{k,D}^2]^T$. Eftersom $0 \le \pi_k \le 1$ och $\sum_{k=1}^{K} \pi_k = 1$ går man lämpligen över till parametriseringen $\pi_k = \exp(\alpha_k) / \sum_l \exp(\alpha_l)$, och får

$$\hat{f}(\alpha_k;\mathbf{y}) = \frac{1}{\sqrt{\pi_k}}(\gamma_k(\mathbf{y}) - \pi_k),$$

$$\hat{\mathbf{f}}(\boldsymbol{\mu}_k;\mathbf{y}) = \frac{1}{\sqrt{\pi_k}}\gamma_k(\mathbf{y})(\frac{\mathbf{y}-\boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k}),$$

$$\hat{\mathbf{f}}(\boldsymbol{\sigma}_k^2;\mathbf{y}) = \frac{1}{\sqrt{2\pi_k}}\gamma_k(\mathbf{y})(\frac{(\mathbf{y}-\boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1).$$

Här är

$$\gamma_k(\mathbf{y}) = \frac{\pi_k p_k(\mathbf{y})}{\sum_{l=1}^{K} \pi_l p_l(\mathbf{y})}$$

aposteriorisannolikheten för komponent *k*. Notera i uttrycken för $\hat{\mathbf{f}}$ att multiplikation och division mellan vektorer ska tolkas elementvis. Den totala Fishervektorn fås genom att konkatenera bidragen från samtliga komponenter, vilket ger en $K(1+2D)$-vektor. Vi ser att Fishervektorn beskriver dels vilken komponent signalen är närmast, men också hur observationen $\mathbf{y}$ avviker från modellen, vilket är mer informativt än vid vektorkvantisering.

Autoencoder

Ett neuronnät, mer specifikt en multilagerperceptron (MLP), kan tränas för att skapa en gles representation av signaler [97]. Ett sådant nät kallas för en *gles autoencoder*. Ett neuronnät består av bearbetningsenheter organiserade i lager, varav det första är för indata. Sedan följer ett eller flera dolda lager, och sist ett utdatalager. Varje enhet, utom i indatalagret, transformerar utdata från föregående lager genom en *aktiveringsfunktion*

$$h(\mathbf{x};\mathbf{w},b) = \sigma(\mathbf{w}^T\mathbf{x} + b)$$

där σ är en monotont växande funktion som kan vara linjär eller olinjär, typiskt

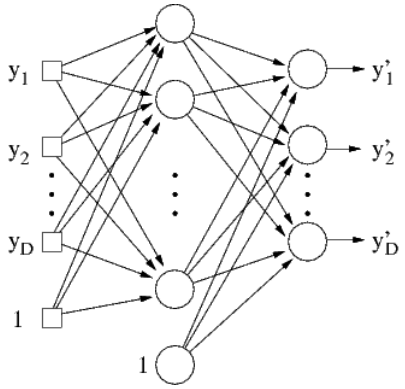
$$\sigma_l(x) = \frac{1}{1+\exp(-x)}, \quad \sigma_t(x) = \tanh(x) \equiv \frac{1-\exp(-2x)}{1+\exp(-2x)}, \quad \sigma_r(x) = \max(0,x)$$

dvs. logistisk funktion, hyperbolisk tangens, respektive ramp (även kallad ReLU, *rectified linear unit*). Nätverket i figur 6 implementerar funktionen

$$h_d^{(2)}(\mathbf{y};\mathbf{w}_d^{(2)},b_d^{(2)}) = \sigma(\mathbf{w}_d^{(2)T}\mathbf{h}^{(1)}(\mathbf{y}) + b_d^{(2)}), \quad d = 1,\ldots,D$$

där komponent k av $\mathbf{h}^{(1)}$ (utsignalen från det dolda mellersta lagret) ges av

$$h_k^{(1)}(\mathbf{y};\mathbf{w}_k^{(1)},b_k^{(1)}) = \sigma(\mathbf{w}_k^{(1)T}\mathbf{y} + b_k^{(1)}), \quad \mathrm{k} = 1,\ldots,K$$



Figur 6. Autoencoder för en D-dimensionell signalfördelning. Notera biasenheterna (konstant värde 1).

Givet en uppsättning träningsdata $(\mathbf{y}_i, \mathbf{t}_i), i = 1,\ldots,N$, där $\mathbf{t}$ är en önskad utsignal, kan nätverkets parametrar bestämmas genom minimering av kostnadsfunktionen

$$E(\mathbf{W}^{(1)},\mathbf{b}^{(1)},\mathbf{W}^{(2)},\mathbf{b}^{(2)}) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{t}_i - \mathbf{h}^{(2)}(\mathbf{y}_i)\right\|^2 + \lambda R(\mathbf{W}^{(1)},\mathbf{W}^{(2)})$$

Här är R en regulariseringsfunktion som förhindrar överanpassning (eng. *overfitting*). Ett vanligt val är (vi generaliserar här till L lager, var och en med K_l enheter)

$$R(\mathbf{W}^{(1)},\ldots,\mathbf{W}^{(L)}) = \sum_{l=1}^{L}\left\|\mathbf{W}^{(l)}\right\|_F^2 = \sum_{l=1}^{L}\sum_{k=1}^{K_l}\left\|\mathbf{w}_k^{(l)}\right\|^2$$

vilket kallas *weight decay*. Ett lokalt optimum kan nås genom gradientsökning (*back-propagation*). Om man nu sätter $\mathbf{t}_i = \mathbf{y}_i$ försöker nätet rekonstruera insignalen, och skapar då en intern representation (kod) i det mellanliggande dolda lagret. För att framtvinga en gles representation kan man sätta en kostnad på den genomsnittliga aktiviteten i det dolda lagret i form av ytterligare en regulariseringsfunktion. För logistisk aktiveringsfunktion eller ReLU kan vi exempelvis använda följande kostnad

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}h_k^{(1)}(\mathbf{y}_i)$$

För tanh-enheter definierar vi representationen som gles om flertalet enheter har aktivering nära -1, och noterar att $\sigma_l(2x) = (\sigma_t(x)+1)/2$. En alternativ formulering är att sätta ett explicit mål $\rho \ll 1$ för den genomsnittliga aktiviteten hos en enskild enhet över hela datauppsättningen, dvs.

$$\hat{\rho}_k = \frac{1}{N}\sum_{i=1}^{N}h_k^{(1)}(\mathbf{y}_i)$$

och använda regulariseringsfunktionen (Kullback-Leibleravståndet)

$$\sum_{k=1}^{K} D_{\mathrm{KL}}(\rho \| \hat{\rho}_k) \equiv \sum_{k=1}^{K} \left[\rho \log \frac{\rho}{\hat{\rho}_k} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_k} \right]$$

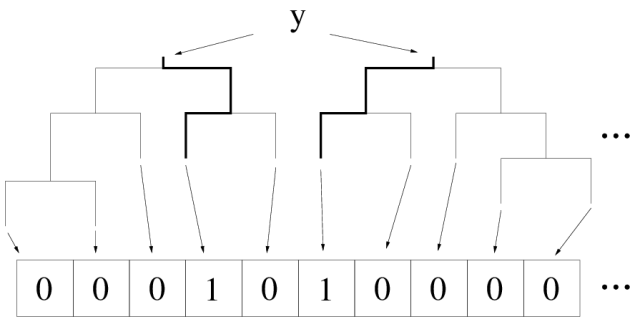
Som diskuterats ovan är det för klassificering fördelaktigt om små till måttliga förändringar av insignalen leder till motsvarande små förändringar i dess kod. Ett sätt att åstadkomma sådan robusthet i en autoencoder är genom regulariseringsfunktionen

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \left\| \nabla_y h_k^{(1)}(\mathbf{y}_i) \right\|^2 = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{J}_h(\mathbf{y}_i) \right\|_F^2$$

där $\mathbf{J}_h$ är Jacobianen för avbildningen $h : \mathbb{R}^D \to \mathbb{R}^K$. Det betyder att kodningen ska sträva efter att vara konstant i närheten av träningsexemplens kodord, vilket för sigmoidenheter tvingar funktionsvärdet till mättnad. Vi måste också kräva viktsymmetri $\mathbf{W}_2 = \mathbf{W}_1^T$. Det resulterande nätverket kallas *contractive autoencoder (CAE)* [98]. Om träningsexemplen huvudsakligen ligger i en lågdimensionell mångfald (lokalt ett linjärt rum) kommer rekonstruktionstermen i den totala kostnadsfunktionen förhindra att kodningen blir konstant längs mångfalden, men ortogonalt mot denna yta kommer kodningen att förändras långsammare, vilket ger robusthet mot brus. Notera att den lokala dimensionaliteten för mångfalden kan bestämmas genom singulärvärdesuppdelning av Jacobianen. En robusthet liknande den just beskrivna kan faktiskt åstadkommas utan att använda några regulariseringsfunktioner, men träna autoencodern på brusade signaler. Detta ger en så kallad *denoising autoencoder* (DAE) [99]. Genom bruset pekar man indirekt ut riktningar i kodningsrummet i vilka koden ska vara konstant. En probabilistisk grafmodell (Markovnät) med stora likheter med DAE är *Restricted Boltzmann Machine (RBM)* [100,101].

Random forest

Vi ska avsluta detta avsnitt om kodning av deskriptorer med en metod baserad på klassificeringsträd som genererar en *klassdiskriminerande* kod (ovan beskrivna metoder fokuserar på rekonstruktion av signalen). Vi har tidigare beskrivit hur CART-algoritmen iterativt bygger upp ett klassificeringsträd där tester, så kallade splittringsfunktioner, i de inre noderna successivt leder ett exempel till ett löv, i vilket aposteriorifördelningen $P(c|\mathbf{x})$ för klasstillhörighet skattas av ett normerat histogram över träningsdata som hamnat där, se figur 4. Uppsättningen löv motsvarar en partitionering av insignalrummet där exempel som liknar varandra och dessutom har samma klasstillhörighet hamnar i samma löv. Om vi numrerar löven kan vi för varje insignal bilda en binär kodvektor med nollor överallt utom för det element som motsvarar lövet i vilket exemplet hamnar, vilket liknar vektorkvantisering. Ett problem med klassificeringsträd är dock att djupa träd, som krävs för att åstadkomma en god klassificering och representation, har hög varians, vilket innebär överanpassning och dålig generaliseringsförmåga. Genom att träna flera träd på olika uppsättningar träningsdata kan större robusthet uppnås vid klassificering. De bästa resultaten fås genom att införa ett slumpmoment under träningen. Utgående från en uppsättning träningsdata samplas för varje träd en egen uppsättning data genom dragning med återläggning (s.k. *bootstrapsampel*). Att väga samman (medelvärdesbilda) modeller som tränats på detta sätt brukar kallas *bagging* [102] och har en påtaglig variansreducerande effekt. Ytterligare variansreduktion fås genom att vid optimeringen av en splittringsfunktion använda en slumpvis vald delmängd träningsdata och inte alltid välja det absolut bästa tröskelvärdet. Medelvärdesbildning av responserna ger en så kallad *Random Forest* [103]. Moosmann *et al*. [104] förslår dock istället att kodvektorerna från de olika träden konkateneras till en sammansatt binär kod, se figur 7.



Figur 7. Kodord genererat genom kombination av en uppsättning beslutsträd tränade som en Random Forest. Efter [104].

3.4.1.5 Inlärning av deskriptorer

Vi har ovan antagit att indata till kodningssteget utgörs av en deskriptorvektor som representerar en lokal bildomgivning. Men ingenting hindrar oss från att istället använda pixelvärdena själva som indata och i kodningssteget från träningsdata optimera en representation av omgivningen med önskade egenskaper.

Coates *et al*. [82] jämför gles autoencoder, gles RBM, vektorkvantisering genom k-meansklustring, samt Gaussblandning för att skapa representationer. Vid vektorkvantiseringen testar man både traditionell kodning där endast ett element i taget är aktivt, och en ny variant, ”triangelkodning”, där de 50 % närmaste prototypernas element aktiveras proportionellt mot hur nära de är till signalen. I samtliga fall normaliserar man först indata (bildomgivningar) genom att subtrahera pixlarnas medelvärde och dividera med deras standardavvikelse för att eliminera känslighet för variationer i intensitet och kontrast. Efter kodning (=inlärning av representationer) och sammanläggning (*pooling*, se nästa avsnitt), används linjära supportvektormaskiner (1-mot-övriga) för klassificering. Man noterar att

- Dekorrelering (*whitening*) av indata radikalt förbättrar klassificeringsresultatet.
- Resultatet förbättras med ökat antal komponenter i kodningsvektorn; upp till 1800 komponenter används.
- Klassificeringen försämras snabbt med ökande steglängd, dvs. minskad samplingstäthet (överlapp) för omgivningar (jmf figur 5).
- För fixt antal kodningskomponenter försämras resultatet när bredden på bildomgivningarna överstiger 6. Förmodligen krävs ett mycket stort antal kodelement för att representera större omgivningar.
- Bästa klassificeringsresultatet uppnås överraskande för vektorkvantisering med triangelkodning. Resultatet är i nivå med djupa neuronnät (se avsnitt 3.4.3).

Shotton *et al*. [105] använder en Random Forest för att skapa en representation på samma sätt som Moosmann *et al*. [104], med den skillnaden att man använder splittringsfunktioner som testar värdet på enskilda pixlar, summan av eller skillnaden mellan par av pixlar, samt absolutvärdet för skillnaden. Sekvensen splittringsfunktioner från rot till löv bildar då komplexa klassdiskriminerande olinjära filter.

3.4.1.6 Spatiell sammanläggning av kodvektorer (*pooling*)

Efter att ha representerat enskilda lokala omgivningar i bilden med en gles kod vill vi nu skapa tolerans/invarians mot transformationer (primärt translation) över större spatiella omgivningar, och minska den globala representationens dimensionalitet. Ett enkelt sätt att åstadkomma detta är att medelvärdesbilda kodvektorerna. Om detta görs globalt

över hela bilden fås en *Bag-of-Visual-Words*-representation (BoVW, även *Bag-of-Features*, BoF) [106], som väsentligen är ett histogram över vilka kodvektorer som finns bilden. Värdet av en gles lokaliserad representation där varje kodelement sällan är aktiverat, och då endast tillsammans med en distinkt mindre grupp andra element (ett lokalt kluster) kan nu inses – gruppen representerar då varianter av en specifik lokal bildstruktur. Histogrammet visar i detta fall i vilka proportioner dessa lokala strukturer finns i den aktuella bilden, vilket kan vara mycket karakteristiskt för enskilda klasser.

För att behålla information om den spatiella fördelningen av lokala strukturer kombineras mer allmänt endast kodvektorer i mindre regioner, så som visas i figur 5. De kombinerade (aggregerade) vektorerna konkateneras sedan till en global kodvektor.

Ett alternativ till medelvärdesbildning som visat sig framgångsrikt är elementvis maximering (*max pooling* [107]). För att intuitivt förstå varför kan man tänka som följer. Om ett kodelement är mycket selektivt, så att det i regionen starkt aktiveras på bara någon enstaka plats (trots överlapp mellan deskriptorernas lokal omgivningar) kan en medelvärdesbildning misstas för en låg aktivering över hela regionen, dvs. att motsvarande lokala struktur inte finns i regionen. Max-operationen ser dock till att den enstaka starka aktiveringen bibehålls i den aggregerade kodvektorn. Om, å andra sidan, ett kodelement ofta är måttligt eller starkt aktivt ger medelvärdesbildningen stor vikt till detta, trots att det då antagligen saknar diskrimineringsförmåga. Man kan sammanfattningsvis säga att max-operationen kompenserar för elementens varierande aktiveringsfrekvens.

Vissa kodningsmetoder har sin egen aggregeringsmetod. Sánchez *et al*. [96] beräknar en gemensam Fishervektor för en region. Scorefunktionen är då

$$\mathbf{f}(\boldsymbol{\theta};\mathbf{y}_1,\ldots,\mathbf{y}_N) = \frac{\partial \log p(\mathbf{y}_1,\ldots,\mathbf{y}_N \,|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

och Fishervektorn

$$\hat{\mathbf{f}}_\theta^Y \equiv \mathbf{L}_\theta^T \mathbf{f}(\boldsymbol{\theta};\mathbf{y}_1,\ldots,\mathbf{y}_N)$$

Om deskriptorerna antas oberoende fås en summa

$$\hat{\mathbf{f}}_\theta^Y = \sum_{i=1}^{N} \mathbf{L}_\theta^T \mathbf{f}(\boldsymbol{\theta};\mathbf{y}_i)$$

vilket motsvararar medelvärdesbildning av de lokala Fishervektorerna över regionen. Författarna noterar dock att klassificering baserad på denna aggregering inte fungerar utan en normalisering av vektorn. Först ersätts varje komponent $\hat{f}_{\theta,k}^Y$ av vektorn med $\operatorname{sign}(\hat{f}_{\theta,k}^Y)\sqrt{|\hat{f}_{\theta,k}^Y|}$ för att kompensera för att observationerna $\mathbf{y}_i$, $i = 1,\ldots,N$ inte är oberoende. Därefter ℓ_2-normeras hela vektorn för att kompensera för varierande mängder brus. I [108] dekorreleras vektorerna innan summationen, vilket ger ett bättre resultat än rotdragningen och i viss mening motsvarar maximering.

En intressant fråga är över vilka områden sammanläggning av kodvektorer ska ske. Det absolut vanligaste är att sammanläggningen sker inom rektangulära regioner i ett reguljärt gitter över bilden, typiskt även i multipla skalor, t.ex. partitionerna 1×1, 2×2, 4×4 rektanglar, osv., så kallad *spatial pyramid pooling* [109,110]. Jia *et al*. [111] bestämmer genom optimering en bästa rektangulär region (*receptive field*) för (max-) aggregering av varje kodord. Feng *et al*. [112] optimerar en viktad summering av kodelement vars koefficienter upphöjts till en exponent $p > 0$; kriteriet för val av vikter och exponent är klasseparation (Fisher). Klassificeringsresultatet är signifikant bättre än med maximering eller summation över en spatiell pyramid. Sanchéz *et al*. [113] föreslår att man vid

beräkning av Fishervektorer för varje deskriptorvektor lägger till dess position i bilden och skapar en Fishervektorkod för denna kombination. Författarna visar att man då kan medelvärdesbilda Fishervektorer över hela bilden och få ett resultat som är bättre än med en spatiell pyramid (eller annan uppdelning); som biprodukt får man också en mindre deskriptor.

3.4.2 System

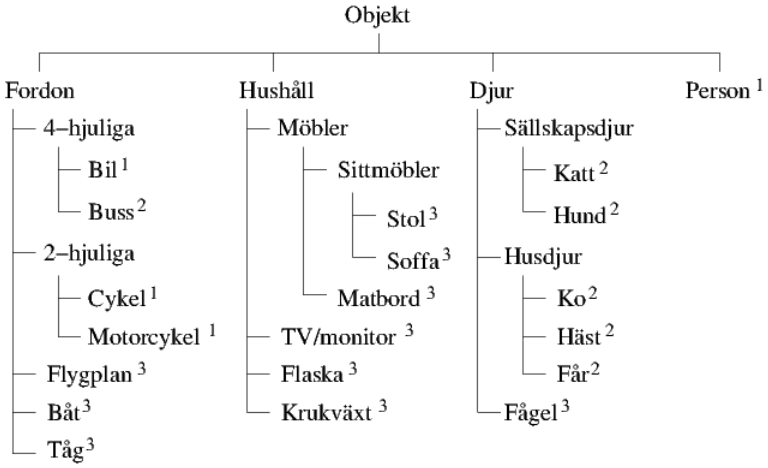
Sista steget i bearbetningskedjan utgörs av en klassificerare. Vi har redan beskrivit några klassificeringsansatser, varav den mest populära under det senaste decenniet tveklöst har varit olika varianter av supportvektormaskinen (SVM). Sedan 2012 [114] har emellertid djupa faltande neuronnät kommit att dominera.

Om vi granskar resultaten av de viktigaste tävlingarna som hållits inom klassificering under de senaste åren framträder en del intressanta, för att inte säga häpnadsväckande fakta om utvecklingen inom området.

PASCAL Visual Object Classes (VOC) Challenge

PASCAL Visual Object Classes (VOC) Challenge [115,116] var en följd tävlingar som organiserades av ett forskarnätverk inom EU och hölls 2005-2012. Ett av delmomenten handlade om klassificering av bilder, andra om detektering och bildsegmentering. Inom klassificering var uppgiften från 2007 att för varje testbild (se figur 8) bestämma vilka av 20 klasser av objekt som fanns i bilden, se figur 9. Från 2009 lades varje år nya exempel till föregående års uppsättning tränings- och valideringsdata. Bilderna samlades in från Internet. Antalet träningsexempel per klass varierade från något hundratal till några tusen. Som prestandamått användes *Average Precision (AP)*. För varje bild förutsätts algoritmen för varje klass producera ett *konfidensvärde* som ska vara ett mått på hur troligt det är att bilden innehåller minst ett objekt i klassen. Genom att sätta ett lägsta tröskelvärde för konfidensen kan för varje bild de troligaste klasshypoteserna listas. *Recall* är då, för ett fixt tröskelvärde, den andel bilder för vilken den korrekta klassen finns i listan. *Precision* är andelen korrekta klasser bland alla listade hypoteser. När tröskelvärdet höjs minskar *recall* men *precision* ökar. Vi kan plotta *precision* som funktion av *recall*. AP är arean under kurvan, och idealt är denna 1, vilket motsvarar att ett tröskelvärde kan sättas så att endast korrekta hypoteser listas, och att listan alltid är fullständig. Vinnarresultat för PASCAL Visual Object Classes (VOC) Challenge för år 2007-2012 visas i tabell 3.

Figur 8. Exempel från VOC. Notera att varje bild kan innehålla objekt från flera klasser.



Figur 9. Objektklasser i PASCAL VOC. Siffran anger året klassen introducerades: 2005[1], 2006[2], respektive 2007[3]. Efter [115].

Det vinnande bidraget vid VOC 2007 [117], som uppnådde AP = 0.575, samplade lokala bildomgivningar dels glest genom användning av två olika intresseoperatorer (blob, hörn), dels tätt i ett reguljärt gitter i multipla skalor. Tre olika deskriptorer användes: SIFT, SIFT+hue samt PAS (kanthistogram). För varje deskriptortyp skapades en kod med 4000 element genom vektorkvantisering (k-means). Koderna aggregerades genom summation (histogram) över ett antal olika uppdelningar av bilden: 1×1 (globalt), 2×2 och 3×1 rektanglar. Resultaten konkatenerades till en särdragsvektor. Bearbetningskedjan applicerades dock separat för varje kombination av samplingsmetod, särdragstyp och bildpartitionering, så att 3×3×3 = 27 olika vektorer skapades per bild. Man optimerade därefter parametrarna i en SVM-kärna som kombinerade de olika vektorerna och klassificerade med 1-mot-övriga-metodiken. Genom att analysera resultat vid användning av olika delmängder av det totala antalet vektorer framgår att den täta samplingen är väsentlig, att SIFT-deskriptorn ensam ger ett mycket bra resultat, samt att aggregering i delregioner är bättre än ett globalt histogram (1×1).

Vinnaren vid VOC 2008 [118], som uppnådde AP = 0,569 (på andra data än 2007), var en ansats mycket lik den föregående vinnaren.

Vinnaren vid VOC 2009, som uppnådde AP = 0,665, använde tät sampling av bildomgivningar vilka representerades genom SIFT. För kodningen användes den glesa kodningsmetoden LCC [90] och vad som kan betraktas som en förenklad version av Fishervektorn kallad Super-Vector Coding [119]. Samma aggregeringsområden användes som av tidigare vinnare. Klassificering kunde utföras med linjära SVM:er, vilket betraktades som ett viktigt framsteg, och ansågs bero på den bättre kodningen.

Vinnaren vid VOC 2010 [120], som uppnådde AP = 0,739, använde som tidigare vinnare multipla samplingar och deskriptorer, följt av vektorkvantisering och multipla aggregeringsområden. En nyhet var att detta kombinerades med dedikerade detektorer [121] som skannade av bilderna, en annan att man vid klassificeringen (via SVM med multipla kärnor [122]) lade på ett bivillkor om ömsesidig exklusivitet, dvs. att vissa klasser inte kunde förekomma samtidigt i samma bild.

Vinnarbidraget vid VOC 2011, som uppnådde AP = 0,767, samplade bildomgivningar tätt och representerade dessa med SIFT, HOG, LBP och färgmoment. Kodningen av deskriptorerna skedde med vektorkvantisering och Fishervektorer. Aggregering gjordes dels över olika rektangulära delområden, men också med en ny metod baserad på adaptiv hierarkisk segmentering av bilden [123], exempelvis baserad på konfidensvärden från objektdetektorer, vilket förbättrar prestanda då objekt inte är centrerade i bilden

eller har ovanlig storlek. Man utnyttjade dessutom en metodik där objektdetektioner användes för att ge kontext till klassificeringen [124] och därmed stötta eller undertrycka hypoteser. Utdata från flera olika olinjära SVM:er (för olika särdrag, kvantiseringar och aggregeringar) kombinerades till en slutlig klassificering.

Vinnare vid VOC 2012, med resultatet AP = 0,822, var samma grupp som föregående år. Ansatsen var i stort densamma, med tillägget att man tränade klassificerare för subkategorier av klasserna [125]. Subkategorierna identifierades genom grafbaserad segmentering av uppsättningen träningsdata.

Tabell 3. PASCAL Visual Object Classes (VOC) Challenge. Vinnarresultat uttryckta i Average Precision. Klassificerare tränade på VOC-data. Se tabell 5 för resultat för klassificerare som tränats på extra data.

2007	2008	2009	2010	2011	2012
0,575	0,569	0,665	0,739	0,767	0,822

ImageNet Large Scale Visual Recognition Competition

ILSVRC [126] startade 2010 och är inriktad mot storskalig objektigenkänning. Klassificeringsuppgiften skiljer sig något från PASCAL VOC Challenge. Skillnaden beror i grunden på att det i detta fall handlar om 1000 klasser och 1,3 miljoner träningsbilder. Bilderna har inte kunnat annoteras fullständigt, utan till varje bild finns endast en klass angiven, även om flera olika objekt är synliga. Därför tillåts upp till fem hypoteser per bild utan att prestanda bestraffas. En viktig skillnad mellan bildmaterialen i VOC och ILSVRC är att i det förra fallet tenderar bilderna att avbilda hela scener (t.ex. ett rum) medan det i ILSVRC huvudsakligen rör sig om objektcentrerade bilder (dock i naturliga bakgrunder). Se figur 10 och jämför med figur 8. Resultat för år 2010-2015 presenteras nedan och sammanfattas i tabell 4.

Figur 10. Exempel på träningsdata från ILSVRC. Från vänster: ”laptop”, ”pomegranate”, ”boat house”, ”springbok”, ”mongoose”.

Vinnare vid ILSVRC2010, som uppnådde 71,8 % korrekta klassificeringar, var samma team och i stort sett samma lösning som vann PASCAL VOC Challenge 2009.

Vinnaren vid ILSVRC2011 [127,128], med 74,2 % rätt, var den grupp som kom tvåa 2010. Metoden använde tät sampling av SIFT och färgdeskriptorer som separat reducerades från 128 respektive 96 till 64 dimensioner genom PCA. Deskriptorerna kodades sedan som Fishervektorer där en Gaussblandning med 1024 komponenter användes, och bilden delades upp i 2×2 regioner över vilka kodvektorerna summerades. Detta gav upphov till två (SIFT resp. färg) särdragvektorer med vardera 520000 komponenter per exempel. Klassificeringen skedde genom linjär SVM enligt principen en-mot-övriga. För att vid träningen hantera den massiva datamängden komprimerades data med en faktor 64 och stokastisk gradientsökning användes vid optimeringen, vilket innebär en iterativ process där ett exempel i taget upprepade gånger dekomprimeras och processas. Slutligen kombinerades resultaten från SIFT och färg till ett beslut.

Vinnaren vid ILSVRC2012 [114], med 83,6 % rätt, var helt överlägsen och innebar ett paradigmskifte inom bildklassificering. Lösningen utgjordes av ett neuronnät (”AlexNet”) med 8 lager, varav enheterna i de 5 första är faltningsoperatorer som verkar på närmast föregående lager, och de tre sista är fullt kopplade. Utdatalagret har 1000 enheter, ett för varje klass. Utdata tolkades som sannolikheter enligt modellen ”soft-max” (variant av logistisk regression beskriven ovan)

$$P(c = k \,|\, \mathbf{y}(\mathbf{x})) = \frac{\exp(y_k(\mathbf{x}))}{\sum_{l=1}^{1000} \exp(y_l(\mathbf{x}))}$$

där $\mathbf{x}$ är en bild och $\mathbf{y}$ utdata från sista lagret. Nätverket tränades genom gradientsökning (*backpropagation*) att maximera log-likelihood

$$l(\boldsymbol{\theta}; \{\mathbf{x}_i, c_i\}_{i=1}^{N}) = \sum_{i=1}^{N} \log P(c = c_i \,|\, \mathbf{y}(\mathbf{x}_i; \boldsymbol{\theta}))$$

Den slutliga klassificeringen skedde genom medelvärdesbildning av utdata från 5 nätverk tränade på samma data.

Vinnaren vid ILSVRC2013, med 88.3 % rätt, var en ansats mycket lik föregående års vinnare. Genom att utveckla en metodik för visualisering av vad de olika lagren i nätverket representerar [129], kunde man identifiera vissa modifieringar av arkitekturen som höjde prestanda. Konkurrensen var detta år betydligt större än 2012, och de flesta av toppresultaten åstadkoms med snarlika neuronnät. Intressant nog kunde två av de bäst presterande bidragen visa att prestandaförbättring kan åstadkommas genom att kombinera utdata från ett neuronnät med det från en klassificerare baserad på Fishervektorkodning. Inspirerad av utvecklingen inom djupa neuronnät, använde [130] Fisherkodning i två steg, dvs., man kodade och aggregerade resultatet av aggregerade Fishervektorer.

Vinnare vid ILSVRC2014, med 93,3 % rätt, var återigen neuronnätsbaserad (”GoogLeNet”) [131]. Den viktigaste skillnaden jämfört med föregående år var nätets djup: 22 lager jämfört med 8 lager för vinnarna 2012-13. Tvåan [132] använde ett nät med 26 lager. Trean [133] är en modifiering av arkitekturen hos vinnarna 2012-13 för att effektivt kunna hantera bilder av godtycklig storlek, både under träning och evaluering, genom användning av spatiell aggregering (*spatial pyramid pooling*) av indata till fullt kopplade lager.

Vinnaren vid ILSVRC2015, med 96,4 % rätt (signifikant över mänsklig nivå), var ett neuronnät (”RelNet”) med en ny struktur [134] som möjliggör effektiv träning av mycket djupa nät; nätet hade 152 lager.

Det är intressant att notera att neuronnät tränade på VOC-data inte ger anmärkningsvärt bra resultat på valideringsdata från den tävlingen, exempelvis uppnår Oquab *et al*. [135] endast AP = 0,709, vilket antas bero på överträning. När författarna däremot tog ett nätverk tränat på ILSVRC-data och justerade detta (*fine tuning*) med VOC-data fick man ett betydligt bättre resultat, AP = 0,828. Med ytterligare modifieringar av nätets arkitektur [136] fås AP = 0,863 (state-of-the-art för VOC 2012 är 2016 AP > 0,93 [137], tabell 5). Skillnaden i mängden träningsdata är således en förklaring till varför neuronnät inte presterat toppresultat tidigare. Rent allmänt har det visat sig att förträning (*pretraining*) på ILSVRC-data följt av finjustering av neuronnätet för ett annat problem är en metodik som fungerar väldigt bra [138].

Chatfield *et al*. [139] analyserar systematiskt skillnaderna mellan den fram till 2011 bästa metoden baserad på Fishervektorkodning av SIFT- och färgdeskriptorer och olika varianter av neuronnät. Man konstaterar att de metoder för att utöka datauppsättningen, exempelvis genom att spegelvända bilder, som använts vid träningen av neuronnäten

även förbättrar prestanda för Fishervektorer, liksom att öka antalet komponenter i Gaussblandningen som används för att modellera deskriptorfördelningen. Trots detta är prestanda på VOC-data klart bättre för neuronnät förtränade på ILSVRC-data (omkring 10 procentenheter). Neuronnäten är också snabbare att evaluera. Man kunde inte heller se någon synergieffekt av att kombinera det bästa neuronnätet med Fishervektorbaserad klassificering.

Tabell 4. ImageNet Large Scale Visual Recognition Competition. Vinnarresultat uttryckta i procentandel av testexemplen för vilka korrekt klass fanns bland fem angivna förslag.

2010	2011	2012	2013	2014	2015
71,8	74,2	83,6	88,3	93,3	96,4

Tabell 5. VOC 2012. Bästa rapporterade resultat (AP) och datum för publicering för metoder tränade på extra data [140]. De flesta metoderna använder djupa faltande neuronnät som först tränats på ILSVRC-data och sedan finjusterats på VOC-data.

1605	1511	1504	1605	1509	1504	1406	1411	1411	1407
0,943	0,940	0,930	0,922	0,917	0,914	0,907	0,904	0,903	0,893

3.4.3 Djupa nätverk

Deep learning (DL) är en relativt ny ansats inom maskininlärning som har gett mycket bra resultat inom väldigt olika tillämpningar [141]. DL har bland annat flyttat fram forskningsfronten inom taligenkänning till nivåer som för bara några år sedan upplevdes som orealistiska [142]. DL har kombinerats med reinforcement learning för att lära sig att spela ett stort antal retro Atari-spel i nivå med eller bättre än bra spelare [143]. DL har, återigen, kombinerats med reinforcement learning för att lära sig spela spelet GO på en nivå som slår världsmästaren [144]. DL har på ett dramatiskt sätt flyttat fram forskningsfronten inom flera bildbehandlingstillämpningar till nivåer som tidigare var orealistiska och långt in i framtiden.

Framgångar inom DL kan karakteriseras på två sätt:

1. DL har flyttat fram forskningsfronten inom flera områden väldigt drastiskt under en relativt kort tid, och tycks fortsätta flytta fram forskningsfronten inom nya tillämpningar framöver. DL har flyttat fram forskningsfronten inom bildbehandling, men även inom flera andra tillämpningar.
2. DL har genom generisk särdragsinlärning och transfer-inlärning presenterat i nivå med forskningsfronten, och i många fall bättre än forskningsfronten, inom många bildbehandlingstillämpningar. Att generera resultat i nivå med, och i många fall bättre än, ytterst specialiserade särdrag utvecklade med stor domänkunskap under många år, med generiska särdrag utan koppling till den aktuella tillämpningen är revolutionerande. Generisk särdragsinlärning har än så länge primärt påverkat bildbehandlingstillämpningar av DL.

Grundläggande koncept inom Deep Learning

Den grundläggande filosofin bakom DL bygger på inlärning av en hierarkisk särdragsrepresentation med hjälpa träningsdata. Den inlärda hierarkiska representation har större kapacitet att representera egenskaper hos data som är central för vidare analys så som

detektion eller klassificering. Varje lager i den inlärda hierarkiska särdragsrepresentationen korresponderar mot en abstraktionsnivå, där abstraktionsnivån ökar med lagren i den hierarkiska representationen. Abstraktionsnivån ökar genom lagren där representation från föregående lager utgör indata till nästkommande lager, vilket genererar en flerlagersstruktur av tilltagande abstraktionsnivå [145,146].

Den helt dominerade tekniken inom DL är neurala nätverk (NN) bestående av flera dolda lager – så kallade djupa neuronnät – och inom bildbehandling är faltande neuronnät (CNN) helt dominerande. DL är i många sammanhang synonymt med djupa neuronnät och CNN inom bildbehandling. Det finns dock inget i den bakomliggande filosofin för DL, inlärning av en hierarkisk särdragsrepresentation, som kopplas till djupa neuronnät. Det existerar alternativa ansatser som bygger på Gaussiska processer så kallade Deep Gaussian Processes (DGP) [147] och kärnbaserade metoder så kallade Deep Kernel Machines (DKM) [148]. De alternativa formuleringarna – DGP och DKM – bygger på en mer solid matematisk grund, vilket kan vara fördelaktigt på längre sikt. Resultaten som de alternativa formuleringarna har genererat är dock betydligt sämre än resultaten som NN och CNN uppnår, detta kan förklarar det helt dominerade intresset för dessa metoder inom DL.

Generellt har tre faktorer varit avgörande för DL [145]:

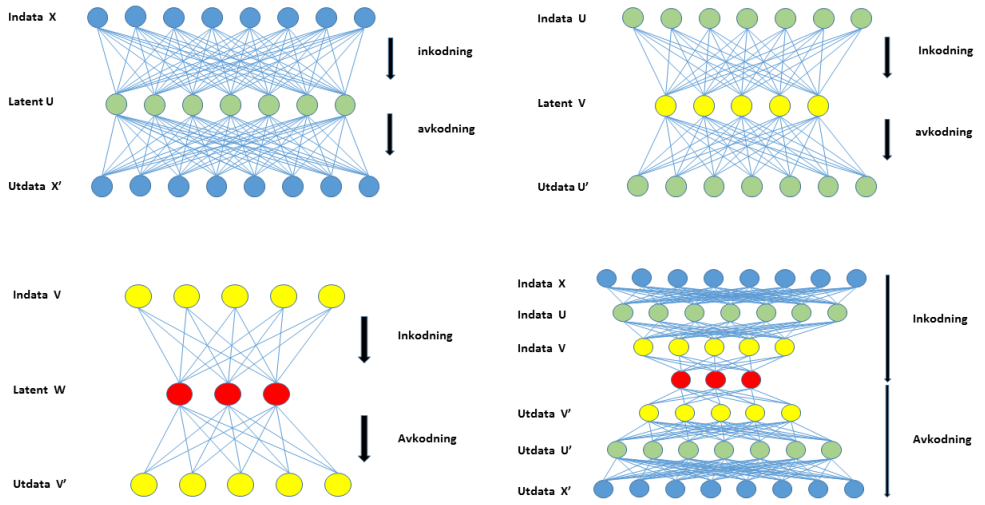
1) Mängden tillgänglig träningsdata har ökat kraftigt. Inom många civila tillämpningar har mängden tillgänglig träningsdata ökat dramatiskt under de senaste åren. Mängden oannoterad data är inom vissa tillämpningsområden nästan obegränsad, samtidigt som mängden annoterad data också har ökat kraftigt. Gemensamt för flertalet av de tillämpningar där DL har flytta fram forskningsfronten är att tillgången på träningsdata har varit stor. Inom bildbehandling har ImageNet datamängden haft en väldigt stor påverkan.
2) Implementation på dedikerad hårdvara. Implantationer av DL algoritmer på dedikerad hårdvara primärt GPU:er har medfört en signifikant uppsnabbning av algoritmerna och signifikant kortare träningstid. Inlärningsexperiment som tidigare vara allt för tidskrävande för att utföras inom rimliga tidsramar är i och med implementation på dedikerade hårdvara nu möjlig.
3) Framsteg inom algoritmutveckling. En stor del av algoritmutvecklingen inom DL handlar om effektivisering av befintliga metoder för att kunna hantera större datamängder och minska träningstiden. Utvecklingen av algoritmer som drastiskt minskar träningstiden på avsevärt större datamängder är en avgörande faktor för DL:s framgångar.

DL kännetecknas idag av en stor del experimentell verksamhet då parameterrummet är väldigt stort och parametervalen till stor del väljs baserat på tidigare erfarenhet med visst teoretiskt stöd. Ett stort antal experiment med mycket stora datamängder med olika parameteruppsättningar krävs för att uppnå de höga resultaten med DL. Utan de förbättrade implementationerna på dedikerad hårdvara (GPU) och de uppsnabbade algoritmerna skulle mängden experiment som krävts för att uppnå DL-resultaten inte varit möjliga.

För-träning och Stackade Auto-Encoder (SAE)

En frekvent förekommande teknik inom DL som haft en väldigt stor påverkan är Auto-Encoder (AE) och Stackade Auto-Encoder (SAE) [145,146,149]. En AE är ett NN som avbildar indata på sig själv dvs. den har indata även som utdata. En AE är ett NN som har lärt sig att reproducera eller rekonstruera dess indata som utdata. Vanligen är antalet noder i det dolda lagren mindre än dimensionen på in- respektive utdata, dimensionen på indata reduceras således i de dolda lagren för att sedan återskapas. En AE består av

två steg, först ett inkodningssteg där indata kodas till en latent representation av lägre dimension, sedan ett avkodningssteg där utdata skall rekonstrueras mha den latenta representationen av lägre dimensionalitet (se figur 11). AE är ett NN där optimal inkodning och avkodning bestäms genom träning. Vanligen med någon variant av backpropagation [BP]. Viktigt att notera är att en AE kan tränas med oannoterad träningsdata då en avbildning av särdragsvektorer på sig själva bestäms.



Figur 11. Stacked Auto-Encoder (SAE). Auto-Encoder (AE) är en viktig teknik inom deep learning där oannoterad träningsdata används för inlärning av en hierarkisk särdragsrepresentation. AE avbildar särdragen på sig själva genom en latent, komprimerad, representation av lägre dimensionalitet. Särdragen inkodas till den latenta representationen, vidare används den latenta representationen för att återskapa, avkoda, de ursprungliga särdragen. Den optimala latenta representationen bestäms genom inlärning. En girig strategi kallad Stacked AutoEncoder (SAE) används för inlärning av en hierarkisk latent särdrag representation. Överst till vänster visas hur en optimal latent representation U av indata X bestäms genom inlärning. Den latenta representationen U används som indata till en AE, överst till höger, där en optimal latent representation V av U bestäms osv. Nederst till höger visas en hierarkisk representation av särdragen som har byggts upp genom att addera ett lager åt gången till hierarkin.

Vanligen innehåller en AE en eller ett fåtal dolda lager då det är svårt att direkt träna ett NN med många dolda lager. Inlärning av djupa AE, innehållande flera dolda lager, görs genom stackning av AE enligt en girig strategi. Särdragsvektorerna används initialt för att träna en AE och bestämma en latent representation av data. När AE:en är färdig tränat appliceras den på träningsmängden och den latenta representationen av lägre dimension används som in- och utdata till nästa AE osv. På detta sätt byggs en hierarkisk representation av särdragen upp genom att addera ett lager åt gången till hierarkin (se figur 11).

Det mittersta dolda lagret i SAE:en utgör en representation av särdragen av lägre dimensionalitet innehållande de centralaste egenskaperna för rekonstruktion. Den latenta representationen kan användas direkt som särdragsvektor innehållande de centralaste egenskaperna för vidare analys.

Huvudanvändningen av SAE:s inom DL är dock inte för inlärning av en komprimerad särdragsrepresentation för direkt användning vid klassificering utan som ett sätt att hitta en bra initiering av vikterna i ett djupt neuronnät. Det har visat sig att träning av ett djupt

neuronnät i praktiken är omöjligt pga. det så kallade vannish gradient problemet. Användning av gradient baserade metoder för parameterinlärning i djupa neuronnät är problematiskt eftersom gradienten blir mycket liten, i princip noll, när felet propageras genom nätverket, det medför att parametrarna nära indatalagren i det djupa neuronnätet i princip inte kommer att uppdateras alls då gradientbaserade metoder används för viktinlärning. SAE:s tränade med oannoterad data används för att hitta bra initieringsvikter till ett djupt neuronnät som sedan kan finjusteras med hjälpa av annoterad data. Ett djupt neuronnät tränas genom följande två steg:

1. Förträning. Oannoterad träningsdata används för inlärning av en hierarkisk särdragrepresentation. En lagervis girig strategi används för att bygga upp en flerlagers SAE-representation.
2. Finjustering. De inlärda vikterna i inkodningssteget i en SAE:en används för initiering av vikterna i ett traditionellt NN. Det har visat sig att vikterna inlärda med en SAE ger en god initiering av vikterna som därefter endast behöver finjusteras med en annoterad träningsmängd.

Genom de två stegen förträning och finjustering kan ett djupt neuralnätverk effektivt tränas. Användningen av SAE för att förträna NN har minskat då möjligheten att träna djupare NN direkt med annoterad träningsdata har förbättrats genom algoritm utveckling primärt genom införandet av ReLU-aktiveringsfunktion. SAE spelar dock fortfarande en stor roll vid tillämpningar med begränsad tillgång på kategoriserad träningsdata [150].

Vincent *et al*. [99] utvärderar olika SAE och Stacked Denoising Auto-Encoder (SDAE) på MNIST datamängden innehållande handskrivna siffor och på ytterligare klassificeringsproblem. Resultaten indikerar att användningen av ett avbrusningskriterie vid AE är av vikt för att bestämma en optimal representation med oannoterad data.

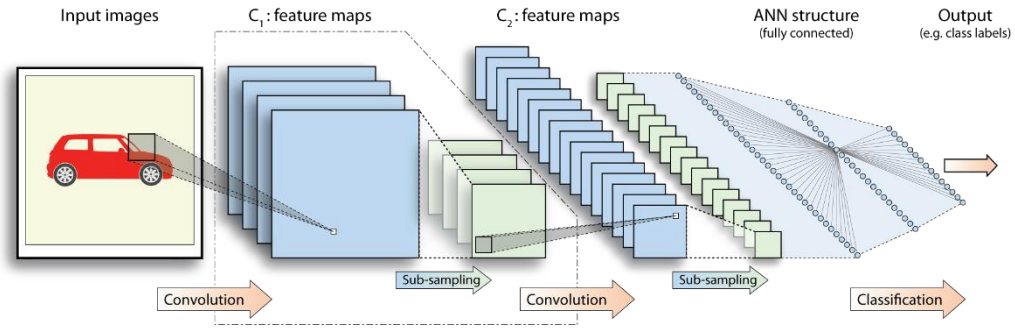
Faltande neuronnät (CNN)

Faltande Neurannät, Convolutional Neural Net [CNN], är en frekvent förekommande teknik inom DL. Inom bildklassificering används ofta CNN synonymt med DL. Med CNN lärs en hierarkisk särdragsrepresentation direkt från en träningsmängd med annoterade bilder utan att någon explicit särdragsextraktion krävs. CNN är ett NN med en speciell struktur bestående av par av lager (se figur 12):

1. Faltande lager. I detta lager faltas föregående bild med ett antal faltningskärnor där varje faltningskärna genererar en särdragskarta av samma spatiala upplösning som bilden/särdragskartan i tidigare lager. Varje faltningskärna kan betraktas som en detektor som detekterar ett specifikt särdrag och den faltade bilden betraktas därmed som en särdragskarta som anger närvaron av det specifika särdraget.
2. Sub-sampling/Pooling lager. Särdragskartorna sub-samplas över mindre regioner i kartan t.ex. 2×2 region genom medelvärdsbildning eller max-pooling. Var efter värdet multipliceras med en vikt för att därefter transformeras med en aktiveringsfunktion. Resultatet blir en transformerad särdragskarta av lägre spatial upplösning (som blir indata till nästa faltningslager).

Ett CNN är sammansatt av ett antal par av lager där den spatiala upplösningen minskar för varje par av lager, emedan antalet särdragskartor vanligen ökar för varje par av lager. Resultatet från det sista sub-samplande lagret stackas till en särdragsvektor och blir indata till ett vanligt NN bestående av ett eller flera dolda lager. Utlagret från CNN är i vanlig ordningen en klassificering och BP kan användas för att träna ett CNN. Storleken och antalet faltningskärnor i varje lager samt antalet lager bestäms manuellt. Valet av

antal lager och antalet särdrag per lager bestäms huvudsakligen baserat på tidigare erfarenheter och experimentella resultat. Vikterna i faltningskärnorna som definierar vilka särdrag som skall extraheras bestäms genom träning. CNN lär sig således de optimala särdragen – vikterna i faltningskärnor – som minimerar klassificeringsfelet på en träningsmängd.



Figur 12. Faltande Neuron Nät (CNN). Ett faltande neuralt nätverk bygger upp en hierarkisk struktur med särdrag. Första nivån i denna struktur skapas genom faltning av den bild som utgör indata. De s.k. faltningskärnor som används är inte specificerade utan tränas in automatiskt i träningsfasen och blir därmed ”optimalt anpassade” för problemet. Man erhåller en särdragskarta för varje faltningskärna man låter träna. Varje särdragskarta undersamplas (dvs. krymps i storlek) för att hålla komplexiteten i nätverket inom rimliga gränser. För att skapa djupare nivåer i hierarkin används lägre nivås särdragskartor som indata för att bygga upp en högre nivå. Den högsta nivån av särdrag används slutligen som indata till ett artificiellt neuralt nätverk som tränas in för att utföra den givna klassificeringsuppgiften.

Det mesta inom DL och CNN bygger på klassiska teorier och metoder från NN som har applicerats på nya sätt. Träning av CNN bygger till stor del på väletablerade metoder utvecklade för NN. Några nya tekniker som framför allt har förkortat träningstiden och gjort klassificerarna mer robusta mot överträning kan dock identifieras:

1. **Rectified Linear Unit – ReLu**. Aktiveringsfunktion i NN var vanligen av sigmoid typ, inspirerade dels av hjärnas neuroner och dels viljan att efterlikna en tröskelfunktion. Sigmoidfunktionen ersätts nu vanligen av aktiveringsfunktionen Rectified Linear Unit – ReLU definierad som

   $$f(x) = \max(0, x)$$

   ReLu resulterar i betydligt kortare träningstid utan att resultaten blir sämre, vilket öppnade upp för fler och större experiment på större datamängder.

2. **Drop-Out**: Drop-out är en teknik för att undvika överträning. Överträning innebär en överanpassning av parametrarna i ett NN till träningsmängden vilket minskar generaliserbarheten hos NN. Vid drop-out sätts responsen från varje dold enhet till 0 med en given sannolikhet vanligen 0.5, men andra sannolikheter kan också förekomma. Vilka enheter som inte skall aktiveras bestäms slumpvis vid varje epok. Drop-out minskar beroendet av vissa särdrag vid klassificering. Klassificeringen kan inte enbart bero på ett värde på ett visst särdrag utan informationen måste vara distribuerad över fler särdrag. En teoretisk genom gång av drop-out metoden redovisas i Srivastava et. al. [151].

3. **Lokal responsnormalisering**. Särdragsnormalisering är ofta avgörande för kortare träningstid och goda klassificeringsresultat. Krizhevsky et. al. [114] använder sig av en normaliseringsmetod – kallad responsnormalisering - i de mel-

lanliggande lagren innehållande särdragskartor. Responsnormaliseringen påminner om lokal kontrastnormalisering av bilder där intensiteten divideras med den lokala variansen i en närliggande region av pixlar. Krizhevsky et. al. [11] rapporterar en klassificeringsförbättring på 1.2 procentenheter pga. responsnormalisering.

Dessa metoder har varit helt avgörande för utvecklingen av DL och CNN, och fick sitt stora genombrott genom Krizhevsky et. al. [114] banbrytande klassificeringsresultat på ILAVRC 2012. Effektivare träningsmetoder har möjliggjort att experiment som tidigare var allt för tidskrävande i större omfattning i dag är möjliga inom en rimlig tidsram. Både ReLu och Drop-out är idag standard metoder inom DL.

Generisk särdragsinlärning och transfer learning

Med CNN bestäms en hierarkisk särdragsrepresentant av ökande abstraktionsnivå genom inlärning med hjälp av en stor mängd annoterade bilder. ImageNet, som diskuteras i avsnitt 3.4.2, är en mycket omfattande mängd annoterade bilder fördelade på ett stort antal kategorier. Kategorierna som ingår är många och särdragen som krävs för att separera klasserna måste vara relativt generiska. Det har framkommit att CNN särdragen som bestämts genom träning på ImageNet-datamängden framgångsrikt kan tillämpas på datamängder med väsentligt annat bildinnehåll än bilderna i ImageNet. Eftersom bildkategorierna i ImageNet är många och variationsgraden mellan de olika kategorierna varierar så krävs relativt generiska särdrag för att separera de olika kategorierna. Med CNN bestäms en hierarkisk särdragsrepresentation där det sista lagret i CNN stackas och används som indata till ett klassiskt NN. Efter träning kan särdragsvektorn, dvs. stackningen av det sista lagret i CNN, användas som särdragvektor till godtycklig klassificerare så som SVM eller ADA-boosting. Den optimala särdragsextraktionen och representation inlärda med CNN blir därmed inte bunden till klassificeringsmetoden som användes för att bestämma särdragen eller till det klassificeringsproblem som används vid inlärningen. Detta utgör grunden för generisk särdragsinlärning och transfer learning.

Azizpour et. al. [152,153] utvärderar särdrag från en CNN tränad på ImageNet på ett antal klassificeringsproblem som avsevärt skiljer sig från bilderna och klasserna i ImageNet. Razavian et. al. [154] använder särdrag från OverFeat [155] som renderade toppresultat på ILSVRC 2013. ImageNet datamängden betraktas som en storskalig datamängd då den innehåller en generisk uppsättning bilder och inte är vinklad mot några specifik tillämpningar. Å andra sidan finns det ett stort antal tillämpningsspecifika bilddatabaser innehållande en speciell typ av bilder, t.ex. vägskyltar eller blommor, dessa bilddatabaser brukar kalas fingryniga – fine grained – då de är inriktad mot en specifik tillämpning. Azizpour et. al. [152,153] använder särdrag från ett CNN inlärda på ImageNet som särdrag till en SVM - så kallad CNN + SVM. Varje max-pooling lager i CNN kan betraktas som en särdragsrepresentation som kan stackas och användas som särdrag till en SVM. Särdrag från lägre pooling-lager, dvs. lager närmare indata-lagret, är mer generiska och mindre knutna till den specifika träningsmängden som den är tränad på. Särdrag från högre pooling-lager, dvs. lager närmare klassificerings-lagret, är mer specialiserade och anpassade till den specifika träningsmängden som den är tränad på. Beroende på hur likt det fingryninga problemet är i bilderna i ImageNet kan olika nivåer i CNN användas som särdrag till SVM.

En komparativ jämförelse mellan CNN + SVM och state-of-the-art metoder som inte använder DL på fin-gryniga problem visar att CNN + SVM presterar mycket bra. CNN + SVM presterar avsevärt mycket bättre än state-of-the-art på många problem och aldrig sämre [152,138]. Resultaten indikerar tydligt att CNN är ett attraktivt alternativ till manuell särdragsextraktion och utvärdering som är mycket tidskrävande och kräver stor domänkunskap (se tabell 6). Användning av CNN särdrag från ett förtränat nät presterar

i många fall bättre än manuellt extraherade särdrag, vilket implicerat att detta borde vara första alternativet i många tillämpningar.

Tabell 6. Sammanställning av resultat från generisk särdragsinlärning – CNN + SVM. Särdragen från ett CNN som har tränats på bilderna i ImageNet har kombinerats med en SVM klassificerare och tillämpats på olika datamängder innehållande bilder relativt olika bilderna i ImageNet. De generiska särdragen från CNN är väl konstruerade för att separera bilder innehållande klasser långt ifrån bildmaterialet i ImageNet. CNN + SVM slår i många fall med god marginal tidigare state-of-the-art metoder. Användning av generiska särdrag som ett första alternativ, istället för den mödosamma processen att implementera och utvärdera ytterst domänspecifika särdrag, i kombination med SVM är ett attraktivt första alternativ (från [152]).

Datamängd	State-of-the-art icke-CNN	CNN + SVM - Standard/Optimized
VOC07	71,1	71.8 / 80,7
MIT	68,5	64,9 / 71.3
SUN	37,5	49,6 / 56
SunAtt	87,5	91,4 / 92,5
UIUC	90,2	90,6 / 91,5
H3D	69,1	73,8 / 74,6
Pet	59,2	78,5 / 88,1
CUB	62,7	62,8 / 67,1
Flower	90,2	90,5 / 91,3

Donahue et. al. [156] utvecklade en generiskt särdragsrepresentation kallad *DeCaf – DEep Convolutional Activation Feature* baserat på ILSVRC 2012 vinnare [114]. Krizhevsky et. al. [114] CNN SuperVision från 2012 består av 5 faltande lager följt av 3 fullt kopplade lager. Donahue et. al. noterar att varje lager i CNN kan betraktas, efter stackning, som särdragsvektor som kan användas vid klassificering med en alternativ klassificeringsmetod. Särdragsvektorer från olika lager – lager 5 till och med 7 - utvärderas på olika datamängder med olika klassificerare. Bäst resultat uppnås då lager 6 eller 7 i det förtränade nätet används, vilket motsvarar det första och andra fullt kopplade lagren. Även här framgår att förtränade särdrag i kombination med lämplig klassificeringsmetod är ett attraktivt alternativ till manuell särdragsextraktion.

Penatti et. al. [157] utvärderar OverFeat särdragen [155] tränade på ImageNet på flygbilder för klassificering av mål. Resultaten visar att särdragen är generaliserbara till helt andra typer av bilder och presterar bättre en alternativa state-of-the-art metoder på vissa datamängder. Dock är resultaten minder entydiga och på vissa datamängder är klassificeringsresultaten sämre.

3.5 Målföljning

Målföljningsalgoritmer används för att skapa målspår från givna observationer (detektioner). Detta ger möjlighet att följa målens position, och kinematik såsom hastighet och acceleration, och i förekommande fall andra viktiga målegenskaper. Målföljningsalgoritmer kan, genom att använda en statistisk modell för hur målet rör sig över tiden, överbrygga begränsade perioder utan observationer och undertrycka både det brus som alltid finns i riktiga observationer och falsklarm som inte beter sig som mål.

Ett viktigt steg i målföljning är associeringen av observationer till målspår. För att göra detta jämförs varje observation med de observationer varje målspårspår förväntas ge upphov till. De bästa associationerna väljs sedan ut för att uppdatera nuvarande målspår och i förekommande fall skapa nya. Två mycket vanliga associationsmetoder är *global nearest neighbour* (GNN) [158, 159], där de totalt sett mest sannolika associationerna väljs ut, och *joint probabilistic data association* (JPDA) [160, 161], där flera observationer ger bidrag till samma målspår baserat på hur sannolika de är. När väl observationer och mål har parats ihop används sedan följefilter för att bestämma önskade måltillstånd mha metoder baserade på bayesiansk statistik. Detta görs vanligen med *kalmanfilter* (KF) [162] eller, när olinjäriteter i de modeller som används förekommer, *extended kalmanfilter* (EKF) [163].

Målföljning är ett förhållandevis moget forskningsområde som varit aktivt sedan mitten av 1900-talet. Forskningsområdet är fortsatt aktivt än idag, vilket kan ses t.ex. från det stora antal publikationer i internationella konferenser såsom International Conference on Information Fusion och IEEE Conference on Aerospace Conference, och journaler såsom IEEE Transactions on Aerospace and Electronic Systems. Forskning inom området är idag inriktat mot metoder för att hantera olinjära problem, och att hantera en förändrad kravbild med flexiblare och mer distribuerade sensornätverk. Olinjära problem uppstår t.ex. vid transformationerna från världskoordinater till bildkoordinater eller då geografisk information utnyttjas i målföljningen. Att hantera olinjära problem är viktigt för att få ut mer information ur sensordata. I sammanhanget bör nämnas olika varianter på kalmanfilter, såsom *unscented kalmanfilter* (UKF) [164] och *cubature kalmanfilter* (CKF) [165], och sekventiella Monte Carlo-metoder som också benämns *partikelfilter* (PF) [166, 167]. Arbetet här går till stor del ut på att anpassa dessa metoder till olika nya problemställningar, samt att anpassa metoderna till att effektivt utnyttja parallell hårdvara och inhomogena sensornätverk med begränsad beräkningskapacitet och begränsad möjlighet till kommunikation.

Ökande beräkningsresurser används även för att förbättra följningen av svaga mål, dvs. mål som i enstaka mätningar ligger nära eller under brusröskeln. Ett sätt att göra detta är *track-before-detect* (TrBD) [158] där måldetektionssteget inte är ett separat steg, utan är integrerat i själva målföljningsalgoritmen. TrBD har funnits sedan andra halvan av 1900-talet, då främst baserat på griddning av tillståndsrummet, dynamisk programmering eller Houghtransformering, och med ökade beräkningsresurser partikelfilterbaserade lösningar [168], med utökningar för t.ex. följning av utbredda mål [169] och heterogena sensorkonfigurationer [170].

Även associationssteget utvecklas i takt med att allt mer beräkningskapacitet blir tillgänglig. Ett exempel på detta är att *multihypotestracking* (MHT) [171, 172] och variationer därav blir allt vanligare, där inte bara en utan flera olika associationshypoteser behandlas samtidigt. Ett alternativt angreppsätt representeras av teorin kring *finite set statistics* (FISST) [173], där ingen association behöver göras utan istället hanteras implicit i den matematiska problemformuleringen. Ur denna formulering har speciellt det senaste decenniet växt fram en stark forskningsfront som jobbar med olika varianter av *probability hypothesis density* (PHD) filter [173] för att approximerar FISST problemet. PHD har rönt stort intresse, men lösningarna kräver dock vissa begränsande antaganden om problemet som studeras. Ett alternativ som därför de senaste åren blivit allt vanligare förekommande inom litteraturen och där för närvarande mest forskning inom området bedrivs är *(multi-target) multi-Bernoulli filter* [173,174], som bygger på en något annan FISST approximation som undviker några av de problem som observerats med PHD-filterlösningar.

Ett resultat av den teknologiska utvecklingen är att det blir allt vanligare att sensorer (ofta commercial off-the-shelf, cots, sensorer med egen målföljning) kombineras i mer eller mindre väldefinierade sensornätverk. Forskningen på området angriper problemet

på två olika sätt. Antingen försöker man att på ett så korrekt som möjligt sätt modellera alla kopplingar som naturligt uppstår mellan de separata sensorerna estimat, och kompenserar för dem [175, 176, 177]. Eller så accepterar man att kopplingar finns mellan observationerna från de olika sensorerna och använder fusions som till priset av något sämre prestanda inte är känsligt för korrelerade observationer [178, 179].

De flesta av ovanstående metoder antas idag finnas implementerade i kommersiellt tillgängliga målföljningssystem. Det är dock oftast svårt att med säkerhet uttala sig om vilka metoder och specialanpassningar specifika system använder sig då detta anses vara känslig information ur konkurrenshänseende.

4 Positionering

Att med god noggrannhet veta position och orientering för sina sensorer är viktigt vid många typer av militära operationer, till exempel vid inmätning av mål för indirekt bekämpning eller för att kunna skapa en gemensam lägesbild mellan flera sensorbärande enheter. Projektet arbetar med positioneringsmetoder för att positionera både enskilda och samverkande stridsfordon med hög global positionsnoggrannhet. För att uppnå det behövs metoder som ger god relativ noggrannhet som sedan kan låsas upp mot kända punkter i terrängen för att på så sätt åstadkomma global positionering med hög noggrannhet.

Positionering med GPS är den vanligaste metoden idag för positionering. En nackdel med GPS-positionering är att den fungerar sämre i bergig och urban terräng. GPS är även relativt enkelt att störa ut. Vissa militära farkoster har därför mycket kvalificerad navigationsutrustning. Detta gäller framför allt flygplan och vissa typer av markgående fordon. Andra plattformar, som t.ex. mindre UAV:er och UGV:er, har däremot ofta betydligt mindre avancerade navigationssystem.

Ett sätt att förbättra navigationsprestanda är att positionera sig genom att matcha bilder eller andra data mot tillgänglig kartinformation eller georefererade flyg-/satellitbilder. Ett annat sätt att förbättra navigationsprestanda är att utnyttja avbildande sensorer som t.ex. visuella kameror och termiska IR-kameror. Genom att hitta och följa stabila och väldefinierade punkter i bilder från dessa sensorer, s.k. *landmärken*, kan ett förbättrat positions- och orienteringsestimat beräknas. Denna metod för att navigera utifrån tidigare okända landmärken kallas ofta *simultaneous localization and mapping* (SLAM). Detta ger ingen absolut världskoordinat, men det ger god relativ noggrannhet. Navigationsdata från plattformens navigationssystem (tröghetsnavigation och GPS - i de fall då GPS anses tillförlitlig) används för att underlätta associationen till tidigare observerade landmärken. Detta görs genom att prediktera var de förväntas synas i nästa bild.

SLAM är ett mycket aktivt forskningsområde, och ett stort antal olika positioneringsmetoder har föreslagits. Positioneringsproblemet består huvudsakligen av två delproblem: tolkning av sensordata (hitta och jämföra bilder, lasermätningar eller landmärken), och följning (prediktion av positionslösningen och uppdatering av densamma baserat på sensordata). Det senare är i princip ett specialfall av målföljningsproblemet, och SLAM-tekniker baserade på Extended Kalmanfilter (EKF-SLAM [180]) och partikelfilter (FastSLAM [181, 182]) är vanligt förekommande. Mer avancerade metoder där alla landmärkesobservationer används i ett och samma ekvationssystem för att beräkna hela rörelsebanan (d.v.s. inte bara den senaste positionen) blir allt vanligare, då beräkningsprestanda och effektiva algoritmer utvecklas [183]. För tolkning av sensordata, i de fall sensorerna är avbildande (t.ex. kameror), används ofta intressepunkter och deskriptorer som exempelvis *scale invariant feature transform* (SIFT) [184].

4.1 Hårdvaruutveckling

Under de senaste åren har det skett en dramatisk förändring inom utbudet av sensorer som kan användas i SLAM-applikationer. I takt med att kvalitén på sensorer förbättras, samtidigt som priser på dessa sjunker, har fler civila och akademiska aktörer visat sitt intresse för SLAM. Framförallt har realtidslösningar med högprestanda och liten drift blivit vanligare samtidigt som tillvägagångsätten blivit mer diversifierade. Vetenskapliga publikationer under de tio senaste åren visar en klar tendens till att använda kameror som de enda avbildande sensorerna i SLAM-lösningar [185] och ofta med stöd från IMU:er. Ny hårdvara, tillsammans med mer avancerade algoritmer, har dock lett till att

lösningar med andra sensorsystem, t.ex. multilinje-lidarsystem, med eller utan stödjande IMU eller INS är realiserbara. Microsofts Kinect-system, som har möjlighet att skapa en högupplöst djupkarta av omgivningen, har också haft en viss påverkan på de senaste årens forskning och sensorval. Dessutom har tillgängligheten till små och billiga ultraljud- och radarsensorer inneburit ett ökat intresse för även dessa sensorer [186].

Civilt finns tillämpningar för positionering inom skogskartering, säkerhetssystem för rökdykare, transportsektor, etc. 3D-modellering av interiör är ytterligare ett exempel med både militär och civil användning. Under 2016 har det också presenterats färdiga civila SLAM-produkter där ett flertal olika sensorer integreras [187].

4.2 Nuvarande forskning inom positionering

4.2.1 Aktiva sensorer

Aktivt avbildade sensorer, som lidar och Microsofts Kinect-system, har möjlighet att skapa modeller av dess omgivning med en detaljnivå och robusthet som överstiger alla motsvarande lösningar med passiva sensorer [188]. Bland annat har forskare från MIT, Carneige Mellon University och National University of Ireland Maynooth tillsammans tagit fram ett realtids SLAM-system där de använder Microsofts Kinect-system för att positionera sig både inomhus och utomhus. Systemet har tillräcklig precision för att klara av att genomför loop-slutning efter det att användaren har färdats över ett hundratal meter [189].

Forskare från Carnegie Mellon University har tagit fram ett landmärksbaserat realtids SLAM-system som bygger på mätningar från en multilinje lidar [190]. Algoritmen går ut på att hitta landmärken som plana ytor och kanter i lidar-punktmoln som sedan matchas med motsvarande landmärken i senare tidssteg. Detta gör det möjligt att beräkna systemets förflyttning mellan de olika tidsstegen. Systemet presterar goda resultat och har en noggrannhet som kan jämföras med icke-realtids system, baserat på resultat på olika referensdatabaser.

4.2.2 Passiva sensorer

Forskningsfältet för passiv sensorer kan idag delas upp i tre generella inriktningar; klassiska filterbaserade lösningar, bildmatchningslösningar samt tekniker inspirerade av naturen [191]. Den sistnämnda har dock inte fått samma utrymme inom SLAM-forskningen utan har istället anammats av andra forskningsgrenar, bland annat med syfte att analysera och förklara djurs beteenden [192]. De flesta SLAM-lösningar som presenterats har använt av någon form av filterbaserad metod [193]. Dock har bildmatchande lösningar ökat i popularitet under de senaste åren och visat mycket goda resultat.

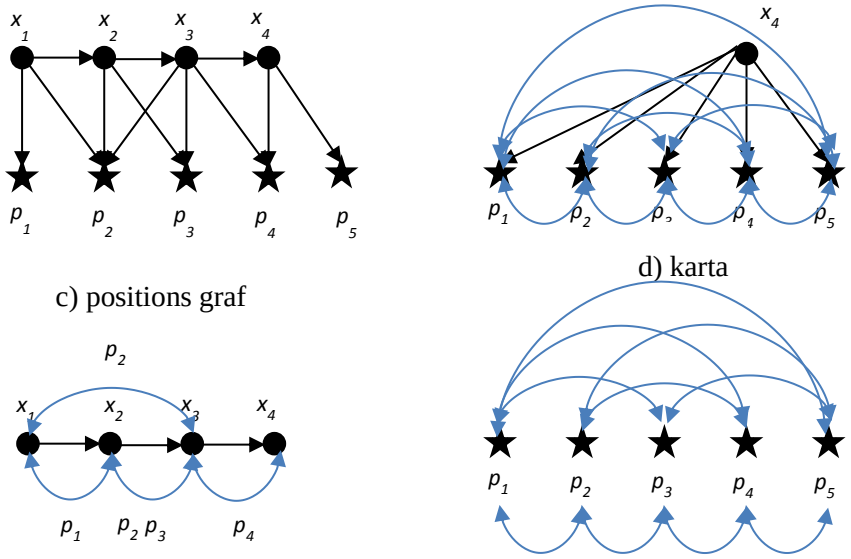
Forskare vid ETH Zurich och Willow Garage har tagit fram ett SLAM-ramverk som klarar av att röra sig över ett stort område utomhus med positionsfel motsvarande ungefär 1% av den totala körda sträckan [194]. Ytterligare ett framgångsrikt koncept inom bildmatchningsgrenen är Large Scale Direct SLAM (LSD-SLAM) som presenterats av en forskningsgrupp från Technical University of Munich. Konceptet går ut på att använda visuella stereokameror för att skatta en relativt högupplöst djupbild av dess omgivning. Denna djupbild matchas sedan mellan olika positioner för att beräkna systemets rörelse. Konceptet har visat mycket goda resultat på olika referensdatabaser för utomhusnavigering [195].

4.2.3 Samverkande system och loop-slutning

För samverkande system såväl som system som arbetar över längre tider och stora områden är metoder för loop-slutning intressanta att studera. I figur 13 ges ett exempel på ett fordon som från fyra positioner mäter på fem träd. Problemet kan representeras i grafform för olika lösningsmetoder som i figur 14. Om vi tänker oss att fordonet fortsätter slingan runt till höger kommer det att återkomma nära position x_1 och återigen kunna se träd p_1 kanske p_2 och senare övriga träd. Om istället ett samverkande fordon anländer nära positionen x_1 och mäter träd p_1 kanske p_2 och senare övriga träd uppnås en likartad situation. Det är känt att den rena filter lösningen (EKF) inte klarar loopslutning efter längre tidsintervall d.v.s. större kartor (Bailey et. al 2006 [196]). Ett exempel på lösningar som klarar att representera det fullständiga problemet (alternativ a i figur 14) är iSAM (Kaess et. al. 2007 [197]). Implementationer finns tillgängliga som ”open-source”. För ännu större områden krävs förenklingar i grafen antingen genom att bryta ned den i sub-kartor eller att reducera länkarna på andra sätt t.ex. genom att på övergripande nivå arbeta på positions grafen (alternativ c i figur 14). Många keyframe baserade metoderna ovan kan ses på exempel av detta. För generell reducering av noder i en graf av iSAM-typ för positionsgrafer finns förslag i [198].

Figur 13. SLAM exempel där ett fordon mäter på fem träd från fyra positioner.

a) Full graf (”smoothing and mapping SAM”) b) Filter (EKF)



Figur 14. Fyra alternativa representationer av situationen i exemplet ovan (figur 13).

I [199] presenterar en realtids SLAM-algoritm baserad på iSAM för samverkande robotar. Lösningen baseras på visuell kontakt mellan robotarna för att avgöra deras relativa förhållande innan kartinformation kan överföras från den ena plattformen till den andra. Algoritmen förutsätter dock en hög bandbredd mellan robotarna.

Kritiskt för loop-slutning (eller samverkan där inte visuell kontakt är tillgänglig som ovan) är att identifiera tidigare landmärken. Inspiration på detta område kan dras från utseendebaserade metoder som RatSLAM [200] och FabMap [201].

Forskare inom positionering/navigering har flera möjligheter att mötas varje år. Bland annat bjuder *The Institute of Navigation* (ION) årligen in till tre tekniska möten: *The International Technical Meeting*, *The Joint Navigation Conference* och *The Satellite Division Technical Meeting* (ION GNSS+). Vartannat år genomför *The Institute of Electrical and Electronics Engineers* (IEEE) och ION även *The Position Location and Navigation System* (PLANS) *Conference*.

För mer robotik inspirerade lösningarna finns de två årliga (IEEE) konferenserna International Conference on Robotics and Automation (ICRA), och Intelligent Robots and Systems (IROS).

5 Sensorstyrning

Sensorstyrning (eng. *sensor management)* är ett generellt systembegrepp för att styra tillgängliga sensorresurser för att i någon mening maximera informationsinnehållet i den insamlade datan för den aktuella tillämpningen [202, 203]. Därav är sensorstyrningen beroende på tillämpningen. Exempel på intressanta tillämpningar är:

- Maximera identifierings- och klassificeringsmöjlighet av okända mål.
- Maximera upptäcktsförmåga av okända mål inom ett intressant område.
- Förbättra omvärldsuppfattning i tidskritiska situationer för sensoroperatörer och beslutsfattare.

Ett sensorsystem kan innehålla flera sensorer av olika typer som lämpar sig bäst för olika uppgifter. Detta medför att dessa måste styras på något sätt för att maximera nyttan av den insamlade informationen. Ett sensorstyrningssystem skall således på något sätt besvara följande frågeställningar:

- Vilken sensor eller grupp av sensorer skall användas?
- Vilka sensorparametrar skall sättas?
- Vart skall sensorn riktas?
- När skall sensorn vara aktiverad?

Eftersom tillämpningsområden och uppgifter kan variera stort för ett sensorstyrningssystem kan dess uppgifter delas in i olika nivåer [203]:

- *Mission planning* (nivå 4)
    - Högsta nivån för att hantera och bestämma uppgifter för flera samverkande system baserat på t.ex. metainformation från operatör eller sensorer
- *Resource deployment* (nivå 3)
    - Hantera vilka typer av sensorresurser som behövs och vart de skall placeras för att lösa uppdraget i en dynamisk miljö.
- *Resource planning* (nivå 2)
    - Hanterar planering av de tillgängliga sensorresurserna, t.ex. förflyttning och samverkan mellan sensorer.
- *Sensor scheduling* (nivå 1)
    - Hanterar hur de erhållna uppgifterna skall utföras beroende på prioritet och sensorbegränsningar.
- *Sensor control* (nivå 0)
    - Bestämma sensorparameterar för att maximera sensorinformation från de utifrån sensors erhållna uppgifter.

5.1 Optimal styrning

För att sensorstyrningssystem skall kunna hantera de olika nivåerna måste olika tekniker användas för att uppnå optimal prestanda givet specificerade kriterier. Den optimala prestandan karaktäriseras av en målfunktion som beskriver önskat beteende hos sensorstyrningssystemet baserat på insignaler, t.ex. sensorinformation och förhandsinformation, och bivillkor, t.ex. begränsad framkomlighet och otillåtna områden. Emellertid är den optimala lösningen väldigt komplex och svår att implementera beräkningseffektivt. Istället för att lösa det optimala problemet finns det suboptimala lösningar som ger tillräcklig prestanda [204]. En suboptimal lösning är att beräkna lösning för n-steg framåt *finite planning horizon*, och använda den första lösningen för att återigen beräkna lösning n-steg framåt vilket kallas *receding horizon control* (RHC) eller *model predictive control* (MPC) [205]. Planeringsproblem kan även formuleras som *travelling salesman*

problem (TSP) nätverksproblem där alla noder ska besökas baserat på olika kostnader för att flytta sig mellan noderna [206].

I fall då flera sensorsystem samverkar kan ytterligare en distinktion mellan olika systemlösningar göras baserat på hur plattformarna koordineras: centraliserat eller decentraliserat. Vid distribuerad koordinering interagerar systemen med varandra emedan vid centraliserad koordinering interagerar systemen genom en huvudnod som delegerar uppgifter till systemen. Samverkande sensorsystem karaktäriseras också av hur stor kunskap de individuella systemen har om varandra, deras förmåga att sammarbeta, samt vilka systemtyper som ingår [207].

Inom projektet kommer inte alla de ovanstående nivåerna att inkluderas. Fokus kommer vara på nivåerna 0-3, vilka innefattar det som är intressant för projektets frågeställningar inom sensorstyrning. Det definierade scenariot för projektet innehåller fordonsburna sensorsystem som kan sitta på både bemannade och obemannade stridsfordon. Sensorstyrning för ett enskilt sensorsystem innefattar nivå 0-1 och flera samverkande sensorsystem innefattar nivå 2-3 enligt definitionen ovan.

5.2 Tillämpningar

Den internationella forskning som bedrivs inom området sensorstyrning är mestadels styrt efter tillämpning, där man vill tillämpa metoder för autonom styrning och planering för att lösa en viss uppgift. På senare tid har dessa metoder börjat tillämpas mer och mer inom den civila sektorn för att nå bättre kvalitet och effektivitet. De intressanta områdena sträcker sig från jordbruk [208], tillverkningsindustri [209] till autonom fordonsstyrning. Det har även bedrivits mycket forskning inom storskaliga system med flertalet plattformar, svärmar och sensorer som ska samarbeta och tillsammans lösa en uppgift [210]. Detta har pådrivits av att det finns färdiga plattformar, både flygande och markgående, att köpa till en relativ låg kostnad.

Inom fordonsbranschen har de automatiska funktionerna blivit mer vanligt än bara för några år sedan. Automatiska funktionerna finns idag från helt autonoma fordon (Google Self-Driving Car) [211] till semi-autonoma fordon som kräver viss interaktion av föraren (Volvo IntelliSafe Autopilot) [212]. Under nästa år ska 100 Volvo bilar kunna köras autonomt på några av Göteborgs vanligaste bilpendlingssträckor. De juridiska förutsättningarna för helt autonoma fordon är annorlunda för Europa än för t ex USA och Kina, då i stort sett samtliga länder i Europa antagit Wien-konventionen för vägtrafik från 1968, som kräver att vägfordon alltid har en förare [213,214].

Inom jordbruket finns en stor potential för autonomi, men också stora utmaningar på många områden, t.ex. koordination av autonoma fordon och sensorer [215], manipulation och igenkänning av grödor, ogräs, m m. Inom EU-projektet crops [216] forskas det bland annat på intelligenta verktyg och manipulatörer, skörd och besprutning.

Sök- och räddningsoperationer (*search and rescue operations*) har på senare år blivit ett hett tillämpningsområde, bland annat på grund av Fukushimakatastrofen, där autonoma robotar skulle kunnat hjälpa till i miljöer där det är för farligt för människor att befinna sig. DARPA:s utmaning för 2015 var att undersöka hur robotar kan lösa ett antal relevanta uppgifter vid en olycka som kärnkraftsolyckan i Fukushima. Detta är ett exempel som speglar en trend inom forskning om robotar inom sök- och räddningsoperationer [217, 218]. Många forskningsprojekt i EU handlar om search and rescue, t.ex. CENTAURO [219], TRADR [220] och INACHUS [221].

Inom projektet är tillämpningar inom övervakning och spaning de mest relevanta. Hur olika resurser ska användas och allokeras för att uppnå bra spaningsförmåga med avseende på olika faktorer, t.ex. rörelsebegränsningar, yttäckningskrav, tid, hinder, opera-

törskrav, uppdragsmål m.m. är frågeställningar som behandlas. Forskningen inom området går ständigt framåt där nya metoder och algoritmer appliceras för att lösa planeringsproblemet. För att planera för flera sensorsystem innebär nya frågeställningar om hur och vilken information som ska utbytas mellan sensorsystemen för att de ska samverka. Detta område innefattas utav *multi-robot systems* (samverkande robotar) som belyser de frågeställningar som är intressanta för projektet i och med att projektscenariot består av flera samverkande sensorsystem. Området är på stor frammarsch inom forskningen med avancerade tillämpningar i robottävlingar [205]. Samverkande sensorer har blivit allt mer intressant de senaste åren för att utnyttja alla dessa sensorer på bättre sätt.

5.3 Forum för autonomiforskning

Bland internationella konferenser såsom IEEE International Conference on Robotics and Automation (ICRA), IEEE International Conference on Intelligent Robots and Systems (IROS) samt International Conference on Automated Planning and Scheduling (ICAPS) publiceras årligen flertalet artiklar inom autonomi, robotik och planering. Området är också representerat av olika internationella journaler såsom IEEE Transactions on Robotics, International Journal of Robotics Research (IJRR) och Journal of Field Robotics (JFR). Olika NATO-grupper t.ex. ”Human-Autonomy Teaming: Supporting Dynamically Adjustable Collaboration (HFM-247)” och “Swarm centric solution for Intelligent Sensor Networks (SET-222)” behandlar samverkan mellan människa och autonoma funktioner samt storskaliga nätverk av sensorer och robotar.

5.4 Framtida forskning går mot riktiga tillämpningar och robusthet

Implementering av området inom robotik och planering är relativt nytt och många av de koncept som har tagits fram är svåra att verifiera med fältprov pga. stora utvecklingskostnader och brist på kontrollerade och upprepbara scenarier. Framtiden går mot mer forskning inom området med stora och skalbara system som dynamiskt ska kunna anpassa sig till olika miljöer och kravställningar. Ett modeord inom autonomiforskningen just nu är ”lifelong” [222, 223]. Det ska påvisas att det automatiska systemet ska kunna fungera och verka under lång tid (systemets livstid), genom att utvärderas på stora, representativa datamängder. Alternativt kan man påvisa att systemet kan anpassa sig till nya situationer. Det blir därmed allt svårare att publicera förslag på algoritmer eller system, om de bara utvärderats på ett fåtal mätningar. Här nämns två exempel på grupper som gjort omfattande utvärdering av hela system:

1) Autonomous Space Robotics Laboratory på Toronto universitet bedriver forskning på autonom styrning av UGV i terräng [224]. De har bland annat tagit fram en metod, V*isual Teach and Repeat*, för att upprepa autonom styrning med en visuell kamera efter en manuell demonstration av färdrutt, i miljöer med vegetation.
2) Mobile Robotics Group på Oxford universitet har jobbat med att kunna lokalisera fordon med visuell kamera i en känd stadsmiljö, oavsett vilket väder eller ljusförhållande som råder, genom att modellera utifrån mycket data i olika väder, klockslag och årstider [225].

Behovet av stora, annoterade och gemensamma data är stort, men det är fortfarande brist på sådana. En stor datamängd för självkörande bilar är KITTI Vision Benchmark Suite (Karlsruhe Inst of Technology and Toyota Institute) [226], som har blivit allt vanligare i utvärderingar.

Ett annat forskningsområde på modet är robust styrning [227, 228]. I tillämpningar där styrning är mer utmanande, t.ex. humanoid gång eller manipulation, har framgångarna

varit mer blygsamma. I sådana tillämpningar är det höga frihetsgrader och komplicerad dynamik, vilket gör att gapet mellan simulerade och verkliga system är stort. När de verkliga systemen körs hamnar de lätt i situationer som inte uppkommit under simulering eller som inte inkluderats i den dynamiska modellen, vilket leder till ett katstrofalt utfall (t.ex. att roboten ramlar, eller tappar objektet den försöker hålla i). Robust styrning går ut på att modellera osäkerheten mellan modellen och verkligheten, så att man kan undvika att hamna på okänt område.

6 Slutsatser och fortsatt arbete

Omvärldsanalysen visar att forskningen inom signalbehandling för EO/IR-sensorer är mycket aktiv runt om i världen. Forskningen inom detta område bedrivs idag vid ett stort antal universitet och forskningsinstitut samt företag. Resultaten från denna forskning kommer troligen att få stor påverkan på både det civila samhället, men även för Försvarsmakten.

Inom måligenkänningsområdet har utvecklingen varit mycket kraftig de senaste åren och deep learning har på ett dramatiskt sätt flyttat fram forskningsfronten inom flera bildbehandlingstillämpningar till nivåer som tidigare var orealistiska och ansågs ligga långt in i framtiden. Deep learning dominerar idag fullständigt inom måligenkänning och en stor del av forskningen har nu inriktats mot algoritmer som även ska kunna lokalisera var i bilden som ett föremål finns.

Många positioneringsalgoritmer fungerar idag bra i enkla miljöer, men en tydlig trend inom forskningen är att nu öka robustheten för dessa så att de ska kunna fungera även i olika miljöer, väder och tidpunkter på dygnet.

I det fortsatta arbetet i projektet kommer olika algoritmer för måligenkänning, positionering och sensorstyrning att implementeras i FOI:s experimentsystem som t.ex. MSP (Multisensorplattform) [229] och sedan testas på scenarier som vi fått från Försvarsmakten för att analysera deras prestanda i militära scenarier.

7 Referenser

[1] Degani, A., *Taming HAL: designing interfaces beyond 2001*, New York: Palgrave Mac millan, 2004.

[2] Sarter, N. B., Woods, D. D., och Billings, C. E., Automation surprises, In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics*, New York: Wiley, 1997.

[3] Billings, C. E., *Aviation Automation: The Search for a Human-Centered Approach*, Mahwah, NJ: Erlbaum, 1997.

[4] Svenmarck, P., "Principles of human-robot coordination for improved manned-unmanned teaming", FOI Memo 1507, Linköping, 2005.

[5] Haykin, S., *Neural Networks, A comprehensive foundation*, Prentice hall, 1999.

[6] Felzenszwalb, P.F., Girshick, R. B., McAllester, D. och Ramanan, D., "Object detection with discriminatively trained part-based models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 1627-1645, 2010.

[7] Torralba, A., Murphy, K.P. och Freeman, W.T., "Sharing visual features for multiclass and multiview object detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(5), 854-869, 2007.

[8] Valpolini, P., "Situational awareness: a lifesaver for vehicle crews", Armada International, June 2012, Internet: http://www.readperiodicals.com/201206/2722305461.html.

[9] Kestrel - Land MTI for Small Unmanned Aircraft Systems, Internet: http://www.avinc.com/uas/kestrel/, besökt 2013-05-31.

[10] Sentinent Develops MTI, Change Detection Capability for Land Combat Vehicles, Internet: http://defense-update.com/20120418_sentinent-develops-mti-change-detection-capability-for-land-combat-vehicles.html, besökt 2013-05-31.

[11] Näsström, F., Allvar, J., Cronström, S., Deleskog, V., Emilsson, E., Engström, P., Hemström, F., Hendeby, G., Karlholm, J., Lindell, R., Möller, S., Rydell, J., Skoglar, P., Stenborg, K-G, Ulvklo, M. och Wikström, J., "Signalbehandling för styrbara sensorsystem - Slutrapport", FOI-R--3570--SE, 2012.

[12] Karlholm, J., "Implementation and evaluation of a method for detection of ground targets in aerial EO/IR imagery", FOI.R--1267--SE, Linköping, June 2004.

[13] Internet: https://www.fbo.gov/utils/view?id=bcc17f4e78b4c7a619ae142c8c37fbbb, besökt 2013-01-16.

[14] Viola, P. och Jones, M.J., "Robust Real-Time-Face Detection", *International Journal of Computer Vision*, 57(2), 127-154, 2004.

[15] Romdhani, S., Torr, P., Schölkopf, B. och Blake, A. "Computationally Efficient Face Detection", *International Conference on Computer Vision*, 2001.

[16] Dollár, P., Belongie, S. och Perona, P. "The Fastest Pedestrian Detector in the West", *British Machine Vision Conference*, 2010.

[17] Gall, J., Lempitsky, V., "Class-Specific Hough Forests for Object Detection", *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[18] Okada, R., "Discriminative Generalized Hough Transform for Object Detection", *IEEE International Conference on Computer Vision*, 2009.

[19] Breiman, L., "Random Forests", *Machine Learning*, 45(1), 5-32, 2001.

[20] Leibe, B., Leonardis, A. och Schiele, B., "Robust Object Detection with Interleaved Categorization and Segmentation", *International Journal of Computer Vision*, 77(1-3), 259-289, 2008.

[21] Shotton, J., Blake, A. och Cipolla, R., "Contour-Based Learning for Object Detection", *IEEE International Conference on Computer Vision*, 2005.

[22] Opelt, A., Pinz, A., Zisserman, A., "A Boundary-Fragment-Model for Object Detection", *European Conference on Computer Vision*, 2006.

[23] Wu, B., Ai, H, Huang, C. och Lao, S., "Fast Rotation Invariant Multi-View Face Detection Based on Real AdaBoost", *International Conference on Automatic Face and Gesture Recognition*, 2004.

[24] Li, S.Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H. och Shum, H., "Statistical Learning of Multi-view Face Detection", *European Conference on Computer Vision*, 2002.

[25] Jones, M. och Viola, P., ”Fast Multi-view Face Detection”, Technical Report TR2003-96, 2003.
[26] Huang, C., Ai, H.Z., Li, Y. och Lao, S.H., ”Vector Boosting for Rotation Invariant Multi-View Face Detection”, *IEEE International Conference on Computer Vision*, 2005.
[27] Lin, Y.-Y. och Liu, T.-L., ”Robust Face Detection with Multi-Class Boosting”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
[28] Wu, B. och Nevatia, R., ”Cluster Boosted Tree Classifier for Multi-View, Multi-Pose Objection Detection”, *IEEE International Conference on Computer Vision*, 2007.
[29] Babenko, B., Dollar, P., Tu, Z. och Belongie, S., “Simultaneous Learning and Alignment Multi-Instance and Multi-Pose Learning”, *European Conference on Computer Vision*, 2008.
[30] Wojek, C., Walk, S. och Schiele, B., “Multi-Cue Onboard Pedestrian Detection”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
[31] Viola, P., Platt, J.C. och Zhang, C., “Multiple instance boosting for object detection”, *Neural Information Processing Systems*, 2005.
[32] Lowe, D.G., “Distinctive image features from scale-invariant keypoints”, *International Journal of Computer Vision*, 60(2), 91-110, 2004.
[33] Dalal., N., Triggs., B. och Schmid, C., ”Human detection using oriented hist. of flow and appearance”, *European Conference on Computer Vision*, 2006.
[34] Ojala, T., Pietikäinen, M. och Harwood, D., “A Comparative Study of Texture Measures with Classification Based on Feature Distributions”, *Pattern Recognition*, 29, 51-59, 1996.
[35] Wang, X., Han, T.X. och Yan, S., ”An HOG-LBP Human Detector with Partial Occlusion Handling”, *International Conference on Computer Vision*, 2009.
[36] Tuzel, O., Porikli, F. och Meer, P., ”Region Covariance: A Fast Descriptor for Detection and Classification”, *European Conference on Computer Vision*, 2006.
[37] Porikli, F., ”Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
[38] Dollár, P., Tu, Z., Perona, P. och Belongie, S., “Integral Channel Features”, *British Machine Vision Conference*, 2009.
[39] Wu, B. och Nevatia, R., “Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors”, *International Journal of Computer Vision*, 75(2), 247-266, 2007.
[40] Lim, J.L., Zitnick, C.L. och Dollár, P., “Sketch Tokens: A Learned Mid-Level Representation for Contour and Object Detection”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
[41] Wu, B. och Nevatia, R., “Optimizing Discrimination-Efficiency Tradeoff in Integrating Heterogeneous Local Features for Object Detection, ”*IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
[42] Dollár, P., Wojek, C., Schiele, B. och Perona, P., “Pedestrian Detection: An Evaluation of the State of The Art”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4), 743-761, 2012.
[43] Breiman, L., Friedman, J. H., Olshen, R. A. och Stone, C. J., *Classification and regression trees*, Chapman & Hall, 1984.
[44] Zhu, Q., Avidan, S., Yeh, M.-C., Cheng, K.-T. och Avidan, S., ”Fast Human Detection Using a Cascade of Histograms of Oriented Gradients”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.
[45] Laptev, I., ”Improving object detection with boosted histograms”, *Image and Vision Computing*, 27, 535-544, 2009.
[46] Liu, C. och Shum, H.Y., ”Kullback-Leibler Boosting”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.
[47] Fukunaga, S., *Introduction to Statistical Pattern Recognition*, Second Edition, Academic Press, 1990.

[48] Hartley, R. och Zisserman, A., *Multiple View Geometry in Computer Vision*, Second Edition, Cambridge University Press, 2003.
[49] Teutsch, M. och Krüger, W., "Detection, Segmentation, and Tracking of Moving Objects in UAV Videos", *IEEE International Conference on Advanced Video and Signal-Based Surveillance*, 2012.
[50] Irani, M. och Anandan, P., "A Unified Approach to Moving Object Detection in 2D and 2D Scenes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6), 577-589, 1998.
[51] Lourakis, M.I.A., Argyros, A.A. och Orphanoudakis, S.C., "Independent 3D Motion Detection Using Residual Parallalx Normal Flow Fields", *IEEE International Conference on Computer Vision*, 1998.
[52] Sawhney, H., Guo, Y. och Kumar R., "Independent motion detection in 3D scenes*", IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10), 1191-1199, 2000.
[53] Yuan, C., Medioni, G., Kang, J. och Cohen, I., "Detecting motion regions in the presence of strong parallax from a moving camera by multiview geometric constraints", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9), 1627-1641, 2007.
[54] ter Haar, F.B., den Hollander, R.J.M. och Dijk, J., "Detection of Moving Objects from a Moving Platform in Urban Scenes", *Proc. SPIE 7701 , Visual Information Processing XIV*, 2010.
[55] Salgian, G., Bergen, J., Samarasekera, S. och Kumar, R., "Moving Target Indication from a moving camera in the presence of strong parallax", Project Report, Robotic Collaborative Technology Alliance, Sarnoff Corporation, 2006.
[56] Yuan, C., Schwab, I., Recktenwald, F. och Mallot, H., "Detection of Moving Objects by Statistical Motion Analysis", *InternationalConference on Intelligent Robots and Systems*, 2010.
[57] Sand, P. och Teller, S., "Particle Video: Long-Range Motion Estimation Using Point Trajectories", *International Journal of Computer Vision*, 80, 72-91, 2008.
[58] Dey, S., Reilly, V., Saleemi, I. och Shah, M., "Detection of Independently Moving Objects in Non-planar Scenes via Multi-Frame Monocular Epipolar Constraint", *European Conference on Computer Vision*, 2012.
[59] Guo, G., Dyer, C.R. och Zhang, Z., "Linear Combination Representation for Outlier Detection in Motion Tracking", *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
[60] Sheikh, Y., Javed, O. och Kanade, T., "Background Subtraction for Free Moving Cameras", *IEEE International Conference on Computer Vision*, 2009.
[61] Weinshall, D. och Tomasi, C., "Linear and Incremental Acquisition of Invariant Shape Models from Image Sequences", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5), 512-517, 1995.
[62] Boxer, P.A. och Loveard, T., "The effect of minimum target size and other factors on the performance envelope of automated moving target indication systems for airborne surveillance with EO sensors", *Proc. SPIE 8020, Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications VIII*, 2011.
[63] Holst, G. C., *Electro-optical imaging system performance*, Fourth Edition, SPIE Press, 2006.
[64] http://host.robots.ox.ac.uk/pascal/VOC/voc2012, besökt 2016-03-30.
[65] https://sites.google.com/site/fgcomp2013, besökt 2016-03-30.
[66] Ramamoorthi, R., "Analytical PCA Construction for Theoretical Analysis of Lighting Variability in Images of a Lambertian Object", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(10), 1322-1333, 2002.
[67] Epstein, R., Hallinan, P. W. och Yuille, A., "5±2 Eigenimages Suffice: An Empirical Investigation of Low-Dimensional Lighting Models", *IEEE Workshop on Physics-Based Modeling in Computer Vision*, 1995.

[68] Shakhnarovich, G. och Moghaddam, B., “Face Recognition in Subspaces”, i S. Z. Li, & A. K. Jain, *Handbook of Face Recognition*, Springer Verlag, 2004.
[69] Ullman, S. och Basri, R., “Recognition by Linear Combination of Models”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10), 992-1006, 1991.
[70] Cootes, T. F., Edwards, G. J. och Taylor, C. J., “Active Appearance Models”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6), 681-685, 2001.
[71] Schölkopf, B. och Smola, A., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002.
[72] Wu, T.-F., Lin, C.-J. och Weng, R. C., “Probability Estimates for Multi-Class Classification by Pairwise Coupling”, *Journal of Machine Learning Research*, 5, 975-1005, 2004.
[73] Platt, J. C., Cristianini, N. och Shawe-Taylor, J., “Large Margin DAGs for Multiclass Classification”, *Advances in Neural Information Processing Systems (NIPS)*, 1999.
[74] Dietterich, T. och Bakiri, G., “Solving Multiclass Learning Problems via Error-Correcting Output Codes”, *Journal of Artificial Intelligence Research*, 2, 263-186, 1995.
[75] Breiman, L., Friedman, J., Stone, C. J. och Olshen, R. A., *Classification and regression trees*, CRC Press, 1984.
[76] Huang, Y., Wu, Z., Wang, L. och Tan, T., “Feature Coding in Image Classification: A Comprehensive Study”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3), 493-506, 2014.
[77] Mikolajczyk, K. och Schmid, C., “Scale & Affine Invariant Interest Point Detectors”, *International Journal of Computer Vision*, 60(1), 63-86, 2004.
[78] Mutch, J. och Lowe, D. G., “Multiclass Object Recognition with Sparse, Localized Features”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, NY, 2006.
[79] Bay, H., Ess, A., Tuytelaars, T. och Van Gool, L., “Speeded-up robust features (SURF)”, *Computer Vision and Image Understanding*, 110(3), 346-359, 2008.
[80] Dalal, N. och Triggs, B., “Histograms of oriented gradients for human detection”, *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2005.
[81] Van der Weijer, J., Schmid, C., Verbeek, J. och Larlus, D., ”Learning Color Names for Real-World Applications”, ”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7), 1512-1523, 2009.
[82] Coates, A., Lee, H. och Ng, A. Y., “An Analysis of Single-Layer Networks in Unsupervised Feature Learning” , *International Conference on Artificial Intelligence and Statistics* (AISTATS), Fort Lauderdale, FL, 2011.
[83] MacQueen, J., “Some Methods for Classification and Analysis of Multivariate Observations”, *Berkeley Symposium on Mathematical Statistics and Probability*, 1967. http://projecteuclid.org/euclid.bsmsp/1200512992, besökt 2016-04-08.
[84] Pati, Y. C., Rezahfar, R. och Krishnaprasad, P. S., ”Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition”, *Asilomar Conference on Signals, Systems, and Computers*, 1993.
[85] Chen, S. S., Donoho, D. L. och Saunders, M. A, “Atomic Decomposition by Basis Pursuit”, *SIAM Review*, 43(1), 129-159, 2001.
[86] Tibshirani, T., “Regression shrinkage and section via the lasso”, *Journal of the Royal Statistical Society, Series B*, 58, 267-288, 1996.
[87] Aharon, M., Elad, M. och Bruckstein, A., ”K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representations”, *IEEE Transactions on Signal Processing*, 54(11), 4311-4322, 2006.
[88] Lee, H., Battle, A., Raina, R och Ng, A., “Efficient sparse coding algorithms”, *Advances in Neural Information Processing Systems (NIPS)*, 2006.
[89] Mairal, J. , Bach, F., Ponce, J. och Sapiro, G., “Online Dictionary Learning for Sparse Coding”, *International Conference on Machine Learning*, Montreal, Canada, 2009.
[90] Yu, K., Zhang, T. och Gong, Y., “Nonlinear learning using local coordinate coding”, *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[91] Yu, K. och Zhang, T., "Improved Local Coordinate Coding Using Local Tangents", *International Conference on Machine Learning*, Haifa, Israel, 2010.
[92] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T. och Gong, Y., "Locality-constrained Linear Coding for Image Classification", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, 2010.
[93] Chatfield, K., Lempitsky, V., Vedaldi, A. och Zisserman, A., "The devil is in the details: an evaluation of recent feature encoding methods", *British Machine Vision Conference*, Dundee, 2011.
[94] Amari, S., "Natural Gradient Works Efficiently in Learning", *Neural Computation*, 10, 251-276, 1998.
[95] Jaakkola, T. och Haussler, D., "Exploiting generative models in discriminative classifiers", *Advances in Neural Information Processing Systems (NIPS)*, 1998.
[96] Sánchez, J., Perronnin, F., Mensink, T. och Verbeek, J., " Image Classification with the Fisher Vector: Theory and Practice" , *International Journal of Computer Vision*, 105, 222-245, 2013.
[97] Ng, A., "Sparse autoencoder", *CS294 Lecture Notes*, Stanford University, 2011. http://web.stanford.edu/class/cs294a/sae/sparseAutoencoderNotes.pdf. Besökt 2016-04-13.
[98] Rifai, S., Vincent, P., Muller, X., Glorot, X. och Bengio, Y., "Contractive Auto-Encoders: Explicit Invariance During Feature Extraction", *International Conference on Machine Learning*, Bellevue, MA., 2011.
[99] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. och Manzagol, P.-A., "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion", *Journal of Machine Learning Research*, 11, 3371-3408, 2010.
[100] Hinton, G. E. och Salakhutdinov, R. R., "Reducing the Dimensionality of Data with Neural Networks", *Science*, 313, 504-507, 2006.
[101] Hinton, G., "A Practical Guide to Training Restricted Boltzmann Machines", UTML TR 2010-003, University of Toronto, 2010. http://www.cs.toronto.edu/~hinton/absps/guideTR.pdf. Besökt 2016-04-15.
[102] Breiman, L., "Bagging predictors", *Machine Learning*, 26, 123-140, 1996.
[103] Breiman, L., "Random Forests", *Machine Learning*, 45, 5-32, 2001.
[104] Moosmann, F., Nowak, E. och Jurie, F., "Randomized Clustering Forests for Image Classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9), 1632-1646, 2008.
[105] Shotton, J., Johnson, M. och Cipolla, R., "Semantic Texton Forests for Image Categorization and Segmentation", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, AK, 2008.
[106] Csurka, H., Dance, C., Fan, L., Willamowski, J. och Bray, C., "Visual categorization with bags of keypoints", *Workshop on statistical learning in computer vision*, *European Conference on Computer Vision*, Prague, Czech Republic, 2004.
[107] Yang, J., Yu, K., Gong, Y. och Huang, T., "Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami Beach, FL, 2009.
[108] Murray, N. och Perronnin, F., "Generalized Max Pooling", *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus OH, 2014.
[109] Lazebnik, S., Schmid, C. och Ponce, J., "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories*", IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, NY, 2006.
[110] He, K., Zhang, X., Ren, S. och Sun, J., "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 1904-1916, 2015.
[111] Jia, Y., Huang, C. och Darrell, T., "Beyond Spatial Pyramids: Receptive Field Learning for Pooled Image Features". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, 2012.

[112] Feng, J., Ni, B., Tian, Q. och Yan, S., ”Geometric ℓ_p-norm Feature Pooling for Image Classification”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, CO, 2011.
[113] Sanchéz, J., Perronnin, F. och de Campos, T., “Modeling the spatial layout of image beyond spatial pyramids”, *Pattern Recognition Letters*, 33, 2216-2223, 2012.
[114] Krizhevsky, A., Sutskever, I. och Hinton, G. E., “ImageNet Classification with Deep Convolutional Neural Networks”, *Advances in Neural Information Processing Systems (NIPS)*, 2012.
[115] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. och Zisserman, A., “The PASCAL Visual Object Classes (VOC) Challenge”, *International Journal of Computer Vision*, 88, 303-338, 2010.
[116] Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J. och Zisserman, A., “The PASCAL Visual Object Classes Challenge: A Retrospective”, *International Journal of Computer Vision*, 111, 98-136, 2015.
[117] Marzalek, M., Schmid, C., Harzallah, H. och Van De Weijer, J., “Learning object representations for visual object class recognition”, *Visual Recognition Challenge workshop, in conjunction with ICCV*, 2007.
[118] Tahir, M. A., Kittler, J., Mikolajczyk, K., Yan, F., van de Sande, K. E. och Gevers, T., ” Visual category recognition using spectral regression and kernel discriminant analysis”, *Visual Recognition Challenge workshop, in conjunction with ICCV*, 2008.
[119] Zhou, X., Yu, K., Zhang, T. och Huang, T. S., ”Image Classification using Super-Vector Coding of Local Image Descriptors”, *European Conference on Computer Vision*, Crete, Greece, 2010.
[120] Yan, Chen, Q., Song, Z., Liu, S., Chen, X., Yuan, X., Chua, T.-S., Huang, Z., Hua, Y. och Shen, S., “Boosting Classification with Exclusive Context”, *PASCAL Visual Object Classes Challenge Workshop*, 2010.
[121] Felzenszwalb, P. F., Girshick, R. B., McAllester, D. och Ramanan, D., “Object detection with discriminatively trained part based models”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 1627-1645, 2010.
[122] Gehler, P. och Nowozin, S., “On Feature Combination for Multiclass Object Classification”, *International Conference on Computer Vision (ICCV)*, Kyoto, Japan, 2009.
[123] Chen, Q., Song, Z., Hua, Y., Huang, Z. och Yan, S., ” Hierarchical Matching with Side Information for Image Classification”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, 2012.
[124] Chen, Q., Song, Z., Dong, J., Huang, Z., Hua, Y. och Yan, S., ” Contextualizing Object Detection and Classification”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1), 13-27, 2015.
[125] Dong, J., Xia, W., Chen, Q., Feng, J., Huang, Z. och Yan, S., ” Subcategory-aware Object Classification”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, OR, 2013.
[126] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. och Fei-Fei, L., “ImageNet Large Scale Visual Recognition Challenge”, *International Journal of Computer Vision*, 115, 211-252, 2015.
[127] Sánchez, J., Perronnin, F., Mensink, T. och Verbeek, J., ” Image Classification with the Fisher Vector: Theory and Practice”, *International Journal of Computer Vision*, 105, 222-245, 2013.
[128] Perronnin, F. och Sánchez, J., “Compressed Fisher vectors for LSVR”, *Large Scale Visual Recognition workshop at ICCV*, Barcelona, 2011.
[129] Zeiler, M. D. och Fergus, R., “Visualizing and Understanding Convolutional Networks”, *European Conference on Computer Vision*, Zürich, Switzerland, 2014.
[130] Simonyan, K., Vedaldi, A. och Zisserman, A., ”Deep Fisher Networks for Large-Scale Image Classification”, *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[131] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. och Rabinovich, A., "Going Deeper with Convolutions", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015.
[132] Simonyan, K. och Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition", *International Conference on Learning Representations*, San Diego, CA, 2015.
[133] He, K., Zhang, X., Ren, S. och Sun, J. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 1904-1916, 2015.
[134] He, K., Zhang, X., Ren, S. och Sun, J., "Deep Residual Learning for Image Recognition", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016.
[135] Oquab, M., Bottou, L., Laptev, I. och Sivic, J., "Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks", ", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, 2014.
[136] Oquab, M., Bottou, L., Laptev, I. och Sivic, J., "Is object localization for free? Weakly-supervised learning with convolutional neural networks", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015.
[137] Wei, Y., Xia, W., Lin, M., Huang, J., Ni, B., Dong, J., Zhao, Y. och Yan, S., " HCP: A Flexible CNN Framework for Multi-label Image Classification", Accepted, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.
[138] Razavian, A. S., Azizpour, H., Sullivan, J. och Carlsson, S., "CNN Features off-the-shelf: an Astounding Baseline for Recognition", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, 2014.
[139] Chatfield, K., Simonyan, K., Vedaldi, A. och Zisserman, A., "Return of the Devil in the Details: Delving Deep into Convolutional Nets", *British Machine Vision Conference (BMVC)*, Nottingham, 2014.
[140] Classification Results: VOC2012. Competition "comp2" (train on own data). http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=2. Besökt 2016-05-24.
[141] Gustafsson, D., Petersson, H. och Enstedt, M., Deep Learning: Concepts and Selected Applications, FOI-D--0701--SE, 2015.
[142] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., och Kingsbury, B., "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups", IEEE Signal Processing Magazine, vol. 29, pp. 82-97, 2012.
[143] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., och Hassabis, D. "Human-level control through deep reinforcement learning", *Nature*, vol. 518, pp. 529-533, 2015.
[144] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. och Hassabis, D., "Mastering the game of Go with deep neural networks and tree search", *Nature*, vol. 529, pp. 484-489, 2016.
[145] Bengio, Y., "Learning Deep Architectures for AI", Foundations and Trends® in Machine Learning, vol. 2, pp. 1-127, 2009.
[146] Deng, L. och Yu, D. "Deep Learning: Methods and Applications", Foundations and Trends® in Signal Processing, vol. 7, pp. 197-387, 2014.
[147] Damianou, A. C. och Lawrence, N. D., "Deep Gaussian processes", Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics (AISTATS-13), pp. 207 - 215, 2013.

[148] Cho, Y. och Saul, L. K., "Kernel Methods for Deep Learning", presented at the Advances in Neural Information Processing Systems (NIPS), 2009.
[149] Hinton, G. E. och Salakhutdinov, R. R."Reducing the Dimensionality of Data with Neural Networks", Science, vol. 313, pp. 504-507, 2006.
[150] LeCun, Y., Bengio, Y. och Hinton G., "Deep Learning ", Nature, Vol 251, May 2015
[151] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I. och Salakhutdinov, R., "Dropout: a simple way to prevent neural networks from overfitting", Journal of Machine Learning Research, vol. 15, pp. 1929-1958, 2014.
[152] Azizpour, H., Razavian, A., Sullivan, J., Maki, A., och Carlsson, S., "Factors of Transferability for a Generic ConvNet Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PP, pp. 1-1, 2015.
[153] Azizpour, H., Razavian, A. S., Sullivan, J., Maki, A. och Carlsson, S., "From generic to specific deep representations for visual recognition", 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 36-45, 2015.
[154] Razavian, A. S., Azizpour, H., Sullivan, J. och Carlsson, S., "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 512-519, 2014.
[155] Sermanet, P., Eigen, D. Zhang, X. Mathieu, M., Fergus, R. och LeCun, Y. "Overfeat: Integrated recognition, localization and detection using convolutional networks", arXiv preprint arXiv:1312.6229, 2013.
[156] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. och Darrell, T., "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", presented at the International Conference on Machine Learning (ICML), Beijing, China, 2014.
[157] Penatti, O. A. B., Nogueira, K. och Santos, J. A. d., "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?", 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 44-51, 2015.
[158] Blackman, S. S. och Popoli, R., *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.
[159] Bar-Shalom, Y. och Li, X. R., *Estimation and Tracking: Principles, Techniques, and Software*, Artech House, 1993.
[160] Bar-Shalom, Y. och Fortmann, T.E., *Tracking and Data Association*, Orlando, FL: Academic Press, 1988.
[161] Fortmann, T.E., Fortmann, T.E. Bar-Shalom, Y. och Scheffe, M., ”Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association”, *IEEE Journal of Oceanic Engineering*, OE-8, July 1983, pp. 173-184.
[162] Kalman, R.E., “A new approach to linear filtering and prediction problems”, *Transactions of the American Society of Mechanical Engineering – Journal Basic Enigneering*, Series D, 82:35-45, March, 1960.
[163] Jazwinski A., *Statistic Processes and Filtering Theory*, Academic Press Inc, 1970.
[164] Julier S., Uhlmann J., “Unscented filtering and nonlinear estimation”, *Proceedings of the IEEE*, 92(3), March 2004.
[165] Arasaratnam, I., Haykin, S., “Cubature Kalman Filters”, in *IEEE Transactions on Automatic Control* 54(6)), June 2009.
[166] Gordon N. J., Salmond D. J. och Smith A.F.M., “A novel approach to nonlinear/non-Gaussian Bayesian state estimation”, in *IEE Proceedings on Radar and Signal Processing*, 140, pages 107-113, 1993.
[167] Doucet A., de Freitas N. och Gordon N., *Sequential Monte Carlo Methods in Practice*, Springer Verlag, 2001.
[168] Salmond, J. och Birch, H., “A particle filter for track-before-detect”, *Proceedings of the American Control Conference*, 2001.
[169] Boers, Y., Driessen, H., Torstensson, J., Trieb, M., Karlsson, R. och Gustafsson, F., ”Track-before-detect algorithm for tracking extended targets”, *IEE Proceedings - Radar, Sonar and Navigation*, 153(4), 2006.

[170] Davey, S.J., Gordon, N.J. och Sabordo, M., "Multi-sensor track-before-detect for complementary sensors", *Digital Signal Processing*, 21, 2011.
[171] Reid, D.B., "A multiple Hypothesis Filter for Tracking Multiple Targets in a Cluttered Environment", Lockheed Missiles and Space Company Report No. LMSC, D-560254, Sept. 1977.
[172] Kurien, T., "Issues in the Design of Practical Multitarget Tracking Algorithms", *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom (Ed.), Norwood, MA: Artech House, 1990, Chap.3.
[173] Mahler, R., *Statistical Multisource-Multitarget Information Fusion*, Artech House, 2007.
[174] Vo, B.T., Vo, B.-N. och Cantoni, A., The cardinality balanced multi-target multi Bernuoulli filter and its impementations, *IEEE Transactions on Signal Processing*, 57(2), February 2009.
[175] Chang, K. C., Saha, R.K. och Bar-Shalom, Y., "On optimal track-to-track fusion", *IEEE Transactions on Aerospace and Electronic Systems*, 33(4), October 1997.
[176] Tian, X. och Bar-Shalom, Y., "Exact algorithms for four track-to-track fusion configurations: All you wanted to ask but were afraid to ask", *Proceedings of the 12th International Conference on Information Fusion*, Seattle WA, Julu 2009.
[177] Chong, C.-Y., Mori, S., Barker, W. H. och Chang, K.-C., "Architectures and algorithms for track association and fusion", *IEEE Aerospace and Electronic Systems Magazine*, 2000.
[178] Julier, S. J. och Uhlmann. J. K., "Non-divergent estimation algorithm in the presence of unknown correlations," *Proceedings of the American Control Conference*, 4, 1997.
[179] Govaers, F., Charlish, A. och Koch, W., "On the decorrelated distributed Kalman filter under measurement origin uncertainty", Proceedings of the 15th International Conference on Information Fusion, 2012.
[180] Veth, M., *Fusion of imaging and inertial sensors for navigation*, Ph.D. dissertation, Air Force Institute of Technology, 2007.
[181] Montemerlo, M., Thrun, S., Koller, D. och Wegbreit, B., "FastSLAM: A factored solution to the simultaneous localization and mapping problem", *Proceedings of the National conference on Artificial Intelligence*. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002.
[182] Montemerlo, M., Thrun, S., Koller, D. och Wegbreit, B., FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 2003.
[183] Dellaert F. och Kaess M., "Square root SAM: Simultaneous localization and mapping via square root information smoothing", *International Journal of Robotics Research*, 25(12):1181-1203, 2006.
[184] Lowe D.G., "Distinctive image features from scale-invariant keypoints", *International journal of computer vision*, 60.2, 2004.
[185] Fuentes-Pacheco, J., Ruiz-Ascebcui, J. och Rendón-Mancha, J. M., "Visual simultaneous localization and mapping: a survey", *Artificial Intelligence Review*, 43.1: 55-81, 2015.
[186] Tardós, J. D., Neira, J., Newman, P. M., och Leonard, J. J., "Robust mapping and localization in indoor environments using sonar data", *The International Journal of Robotics Research*, *21*(4), 311-330, 2002.
[187] Real Earth, http://www.realearth.us/, besökt 2016-03-30.
[188] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J. och Fitzgibbon, A. "KinectFusion: Real-time dense surface mapping and tracking". *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, 127-136, 2011.
[189] Whelan, T., Kaess, M., Johannsson, H., Fallon, M., Leonard, J. J., och McDonald, J. "Real-time large-scale dense RGB-D SLAM with volumetric fusion", *The International Journal of Robotics Research*, *34*(4-5), 598-626, 2015.

[190] Zhang, J., och Singh, S., “Low-drift and real-time lidar odometry and mapping”. *Autonomous Robots*, 1-16, 2016.
[191] Fuentes-Pacheco, J., Ruiz-Ascebcui, J., och Rendón-Mancha, J. M., “Visual simultaneous localization and mapping: a survey”, *Artificial Intelligence Review*, 43.1: 55-81, 2015.
[192] Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., och Furgale, P., “Keyframe-based visual–inertial odometry using nonlinear optimization”, *The International Journal of Robotics Research*, 34.3: 314-334, 2015.
[193] Fuentes-Pacheco, J., Ruiz-Ascebcui, J. och Rendón-Mancha J. M., “Visual simultaneous localization and mapping: a survey”, *Artificial Intelligence Review*, 43.1: 55-81, 2015.
[194] Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., och Furgale, P., “Keyframe-based visual–inertial odometry using nonlinear optimization”, *The International Journal of Robotics Research*, 34.3: 314-334, 2015.
[195] Engel, J., Stuckler, J. och Cremers, D., ”Large-scale direct slam with stereo cameras”, *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, 1935-1942, 2015.
[196] Bailey, T., Nieto, J., Guivant, J., Stevens, M. och Nebot, E., ”Consistency of the EKF-SLAM algorithm”, *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 3562-3568, 2006.
[197] Kaess, M., Ranganathan, A. och Dellaert, F., “iSAM: Fast incremental smoothing and mapping with efficient data association”, *Robotics and Automation, 2007 IEEE International Conference on*, 1670-1677, 2007.
[198] Carlevaris-Bianco, N., Kaess, M. och Eustice, R. M., “Generic node removal for factor-graph SLAM”, *Robotics, IEEE Transactions on, IEEE*, 30:1371-1385, 2014.
[199] Kim, B., Kaess, M., Fletcher, L., Leonard, J., Bachrach, A., Roy, N., *et al*. (2010). Multiple relative pose graphs for robust cooperative mapping. Paper presented at the Robotics and Automation (ICRA), 2010 IEEE International Conference.
[200] Milford, M. J. och Gordon F. W., "Mapping a suburb with a single camera using a biologically inspired SLAM system." Robotics, IEEE Transactions on 24.5 (2008): 1038-1053.
[201] Cummins, M. J. och Newman, P. M., “Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model”, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 3-10, 2010.
[202] Ng G.W. och Ng K.H., “Sensor Management – what, why, and how”, *Information Fusion*, 1(2):67-75, Dec 2000.
[203] Xiong N. och Svensson P., “Multi-sensor management for information fusion: issues and approaches”, *Information Fusion*, 3(2):163-186, June 2002.
[204] Skoglar P., *Tracking and Planning for Surveillance Applications*, Dissertations no 1432, Linköping Studies in Scienece and Technology, April 2012.
[205] Olson E., Strom, J., Morton, R., Richardson, A., Ranganathan, P., Goeddel, R., Bulic, M., Crossman, J. och Marinier, B, Progress toward multi-robot reconnaissance and the MAGIC 2010 competition, *Journal of Field Robotics*, 29(5), 762-792, Sep 2012.
[206] Hagström M., Wirkander S-L. och Ögren P., “Survey of Autonomous Algorithms for Cooperative Sureveillance and Strike Problems”, FOI-R--2016--SE, 2006.
[207] Department of defense, Defense science board, *Task force report: The role and Autonomy in DoD Systems*, July 2012.
[208] Tokekar, P., Vander Hook, J., Mulla D., och Isler, V., "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5321-5326, 2013.
[209] Gronle M. och Osten, W., “View and sensor planning for multi-sensor surface inspection”, *Surface Topography: Metrology and Properties*, 4(2), 2016.
[210] Brambilla, M., Ferrante, E., Birattari, M. och Dorigo, M., ” Swarm robotics: a review from the swarm engineering perspective”, *Swarm Intelligence*, 7(1):1-41, 2013.

[211] https://www.google.com/selfdrivingcar/, besökt 2016-05-24.
[212] http://www.volvocars.com/intl/about/our-innovation-brands/intellisafe/intellisafe-autopilot, besökt 2016-05-24.
[213] "Convention on Road Traffic." United Nations, Vienna, Austria, 1968.
[214] Vanholme, B., Gruyer, D., Lusetti, B., Glaser, S. och Mammar, S., "Highly automated driving on highways based on legal safety," Intelligent Transportation Systems, IEEE Transactions on, vol. 14, no. 1, pp. 333–347, 2013.
[215] Emmi, L., Gonzalez-de-Soto, M., Pajares, G. och Gonzalez-de-Santos, P., "New trends in robotics for agriculture: integration and assessment of a real fleet of robots," The Scientific World Journal, vol. 2014, 2014.
[216] "Crops homepage." [Online]. Available: http://www.crops-robots.eu/. [Accessed: 24-May-2016].
[217] Lim J., Shim, I., Sim, O., Joe, H., Kim, I., Lee, J. och Oh, J.-H., "Robotic software system for the disaster circumstances: System of team KAIST in the DARPA Robotics Challenge Finals," in Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on, 2015, pp. 1161–1166.
[218] Atkeson, C. G., Babu, B. P. W., Banerjee, N., Berenson, D., Bove, C. P., Cui, X., DeDonato, M., Du, R., Feng, S., Franklin, P., *et al*., "No falls, no resets: Reliable humanoid behavior in the DARPA robotics challenge," in Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on, 2015, pp. 623–630.
[219] "Centauro Project," Plone site. [Online]. Available: https://www.centauro-project.eu/front-page. [Accessed: 24-May-2016].
[220] Kruijff-Korbayová, I., Colas, F., Gianni, M., Pirri, F., de Greeff, J., Hindriks, K., Neerincx, M., Ögren, P., Svoboda, T. och Worst, R., "TRADR Project: Long-Term Human-Robot Teaming for Robot Assisted Disaster Response," Künstl Intell, vol. 29, no. 2, pp. 193–201, Feb. 2015.
[221] "INACHUS FP7 Project (607522)," INACHUS FP7 Project (607522). [Online]. Available: http://www.inachus.eu/. [Accessed: 24-May-2016].
[222] Thrun, S., "A lifelong learning perspective for mobile robot control," in Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on, 1994, vol. 1, pp. 23–30.
[223] Churchill W. och Newman, P., "Practice makes perfect? managing and leveraging visual experiences for lifelong navigation," in Robotics and Automation (ICRA), 2012 IEEE International Conference on, 2012, pp. 4525–4532.
[224] Furgale P. och Barfoot, T. D., "Visual teach and repeat for long-range rover autonomy," Journal of Field Robotics, vol. 27, no. 5, pp. 534–560, 2010.
[225] Smith, M., Baldwin, I., Churchill, W., Paul, R. och Newman, P., "The new college vision and laser data set," The International Journal of Robotics Research, vol. 28, no. 5, pp. 595–599, 2009.
[226] Geiger, A. Lenz, P. och Urtasun, R. "Are we ready for autonomous driving? the kitti vision benchmark suite," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 2012, pp. 3354–3361.
[227] Zhou, K. Doyle, J. C., Glover, K. och others, Robust and optimal control, vol. 40. Prentice hall New Jersey, 1996.
[228] Ackermann, J., Robust control: Systems with uncertain physical parameters. Springer Science & Business Media, 2012.
[229] Näsström, F., Allvar, J., Bilock, E., Bissmarck, F., Deleskog, V., Forsgren, R., Habberstad, H., Hemström, F., Hendeby, G., Karlholm, J., Nordlöf, J., Nygårds, J. och Rydell, J. "Intelligent spaning 2012-2015 - Slutrapport", FOI-R--4148--SE, december 2015.

FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1000 anställda varav ungefär 800 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömning av olika typer av hot, system för ledning och hantering av kriser, skydd mot och hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.

FOI
Totalförsvarets forskningsinstitut
164 90 Stockholm

Tel: 08-55 50 30 00
Fax: 08-55 50 31 00

www.foi.se