

# LUNG-SEGMENTERING

## – FÖRBEHANDLING AV MEDICINSK DATA VID PREDICERING MED KONVOLUTIONELLA NEURALA NÄTVERK

Examensarbete Systemarkitekturutbildningen

Gustavsson Robin  
Jakobsson Johan

2018KSAI09



HÖGSKOLAN  
I BORÅS

**Systemarkitekturutbildningen** är en kandidatutbildning med fokus på programutveckling. Utbildningen ger studenterna god bredd inom traditionell program- och systemutveckling, samt en spets mot modern utveckling för webben, mobila enheter och spel. Systemarkitekten blir en tekniskt skicklig och mycket bred programutvecklare. Typiska roller är därför programmerare och lösningsarkitekt. Styrkan hos utbildningen är främst bredden på de mjukvaruprojekt den färdige studenten är förberedd för. Efter examen skall systemarkitekter fungera dels som självständiga programutvecklare och dels som medarbetare i en större utvecklingsgrupp, vilket innebär förtrogenhet med olika arbetssätt inom programutveckling.

I utbildningen läggs stor vikt vid användning av de senaste teknikerna, miljöerna, verktygen och metoderna. Tillsammans med ovanstående teoretiska grund innebär detta att systemarkitekter skall vara anställningsbara som programutvecklare direkt efter examen. Det är lika naturligt för en nytexaminerad systemarkitekt att arbeta som programutvecklare på ett stort företags IT-avdelning, som en konsultfirma. Systemarkitekten är också lämpad att arbeta inom teknik- och idédrivna verksamheter, vilka till exempel kan vara spelutveckling, webbapplikationer eller mobila tjänster.

Syftet med examensarbetet på systemarkitekturutbildningen är att studenten skall visa förmåga att delta i forsknings- eller utvecklingsarbete och därigenom bidra till kunskapsutvecklingen inom ämnet och avrapportera detta på ett vetenskapligt sätt. Således måste de projekt som utförs ha tillräcklig vetenskaplig och/eller innovativ höjd för att generera ny och generellt intressant kunskap.

Examensarbetet genomförs vanligen i samarbete med en extern uppdragsgivare eller forskningsgrupp. Det huvudsakliga resultatet utgörs av en skriftlig rapport på engelska eller svenska, samt eventuell produkt (t.ex. programvara eller rapport) levererad till extern uppdragsgivare. I examinationen ingår även presentation av arbetet, samt muntlig och skriftlig opposition på ett annat examensarbete vid ett examinationsseminarium. Examensarbetet bedöms och betygssätts baserat på delarna ovan, specifikt tas även hänsyn till kvaliteten på eventuell framtagen mjukvara. Examinator rådfrågar handledare och eventuell extern kontaktperson vid betygssättning.



HÖGSKOLAN  
I BORÅS

BESÖKSADRESS: JÄRNVÄGSGATAN 5 · POSTADRESS: ALLÉGATAN 1, 501 90 BORÅS  
TFN: 033-435 40 00 · E-POST: INST.HIT@HB.SE · WEBB: WWW.HB.SE/HIT

**Svensk titel:** Lung-segmentering – Förbehandling av medicinsk data vid predicering med konvolutionella neurala nätverk.

**Engelsk titel:** Lung-segmentation – A pre-processing technique for medical data when predicting with convolutional neural networks.

**Utgivningsår:** 2018

**Författare:** Gustavsson Robin & Jakobsson Johan

**Handledare:** Henrik Linusson

### **Abstract**

In the year of 2017 the Swedish social office reported the most common cancer related death amongst women was lung cancer and the secondmost common amongst men. A way to find out if a patient has lung cancer is for a doctor to study a computed tomography scan of a patients lungs. This introduces the chance for human error and could lead to fatal consequences. To prevent mistakes from happening it is possible to use computers and advanced algorithms for training a network model to detect details and deviations in the scans. This technique is called deep structural learning. It is both time consuming and highly challenging to create such a model. This discloses the importance of decorous training, and a lot of studies cover this subject. What these studies fail to emphasize is the significance of the preprocessing technique called lung segmentation. Therefore we investigated *how is the accuracy and loss of a convolutional network model affected when lung segmentation is applied to the model's training and test data?* In this study a number of models were trained and evaluated on data where lung segmentation was applied, in relation to when it was not. The final conclusion of this report shows that the technique counteracts overfitting of a model and we allege that this study can ease further research within the same area of study.

**Keywords:** Lung cancer, Convolution, Neural network, Lung segmentation, DICOM, Pre-process

## Sammanfattning

Svenska socialstyrelsen presenterade år 2017 att lungcancer är den vanligaste cancerrelaterade dödsorsaken bland kvinnor i Sverige och den näst vanligaste bland män. Ett sätt att ta reda på om en patient har lungcancer är att en läkare studerar en tredimensionell-röntgenbild av en patients lungor. För att förebygga misstag som kan orsakas av den mänskliga faktorn är det möjligt att använda datorer och avancerade algoritmer för att upptäcka lungcancer. En nätverksmodell kan tränas att upptäcka detaljer och avvikelser i en lungröntgenbild, denna teknik kallas deep structural learning. Det är både tidskrävande och avancerat att skapa en sådan modell, det är därför viktigt att modellen tränas korrekt. Det finns flera studier som behandlar olika nätverksarkitekturer, däremot inte vad förbehandlingstekniken lung-segmentering kan ha för inverkan på en modell av denna signifikans. Därför ställde vi frågan: *hur påverkas accuracy och loss hos en konvolutionell nätverksmodell när lung-segmentering appliceras på modellens tränings- och testdata?* För att besvara frågan skapade vi flera modeller som använt, respektive, inte använt lung-segmentering. Modellernas resultat evaluerades och jämfördes, tekniken visade sig motverka överträning. Vi anser att denna studie kan underlätta för framtida forskning inom samma och liknande problemområde.

**Nyckelord:** Lungcancer, Konvolution, Neurtalt nätverk, Lung-segmentering, DICOM, Förbehandling



# Innehållsförteckning

1 Inledning.....	1
1.1 Syfte.....	2
1.2 Frågeställning.....	2
1.3 Avgränsning.....	2
2 Teori.....	3
2.1 Artificiella neurala nätverk.....	3
2.1.1 Inlärning.....	4
2.1.2 Fram-matande och rekurrenta nätverk.....	5
2.2 Konvolutionella neurala nätverk.....	6
2.2.1 Konvolution.....	7
2.2.2 Sparsamma interaktioner och parameterdelning.....	8
2.2.3 Pooling.....	10
2.3 Förbehandlingstekniker.....	11
2.4 Utvärderingsfunktion.....	12
3 Metod.....	13
3.1 Medicinsk data.....	13
3.2 Applicering och utvärdering.....	14
3.3 Lung-segmentering.....	15
3.4 Nätverksarkitektur.....	16
3.5 Hårdvara.....	17
4 Resultat.....	18
5 Diskussion.....	22
5.1 Resultatdiskussion.....	22
5.2 Metoddiskussion.....	23
6 Slutsats.....	24
Appendix A.....	25

# 1 Inledning

År 2012 rapporterade amerikanska cancerinstitutet (2017) 8.2 miljoner cancerrelaterade dödsfall och enligt Svenska socialstyrelsen (2017) var lungcancer den vanligaste cancerrelaterade dödsorsaken bland kvinnor i Sverige och den näst vanligaste bland män. *Kaggle* är en av de största online-gemenskaperna inom maskininlärningstävlingar. Sedan 2014 har Kaggle hållit en årlig tävling, *Kaggle data science bowl* (KDSB), tillsammans med ett eller flera stora företag. I grupp eller individuellt, tävlar medlemmar på Kaggle under en 90-dagars period om prispengar genom att träna en modell att förutse unika dataset. År 2017 gick tävlingen ut på ta fram de bästa algoritmerna för att predicera lungcancer. Datasetet som tävlingen utgick från bestod av över tusen anonyma lungröntgenbilder som amerikanska cancerinstitutet bidrog med (Kaggle 2017).

Att upptäcka cancer i ett tidigt stadiet kan vara livsavgörande för en patient. Ett sätt att ta reda på om en patient har lungcancer, är att en läkare studerar en 3D-röntgenbild av en patients lungor (Kaggle 2017). Det kan vara svårt för en människa att upptäcka tidig cancerutveckling, då både kvalite på lungröntgenbilder kan variera och mänsklig faktor spelar in. För att med större säkerhet kunna ge en patient rätt diagnos, kan en maskin eller dator tränas att upptäcka detaljer och avvikelser i en lungröntgenbild, den tekniken som används för detta kallas *deep structural learning* (deep learning) (Russel, Norvig 2010).

Deep learning är en teknik som bygger på *Artificiell intelligens* (AI) och enligt Goodfellow, Bengio och Courville (2016) har AI haft sina största framgångar i enkla och väldefinierade problemområden, i förhållanden där alla regler är givna och med ett fördefinierat start- och slutresultat. Flera av dessa typer av problemområden kan vara intellektuellt utmanande för människor men desto tydligare för datorer. Det som AI kan ha problem med är något människor gör intuitivt, till exempel att känna igen ord- och meningsstrukturer eller ansikten i bilder. Lösningen på problem av denna struktur kommer från deep learning (Goodfellow, Bengio och Courville 2016). Deep learning är ett försök att efterlikna den mänskliga hjärnan, vilket uppnås genom att använda multilagrade perceptroner för att imitera de biologiska neuronerna. På samma sätt som en hjärna, kan också ett neuralt nätverk tränas att identifiera objekt i en bild. *Convolutional Neural Network* (CNN) är en typ av nätverk som kan användas till att predicera data i två- och tredimensionella bilder (Zeiler & Fergus 2013).

När en patient undersöks för lungcancer utförs en speciell typ av röntgen som kallas *lågdos spiralatortomografi* (DT) (Strålsäkerhetsmyndigheten 2017). När en DT-skanning utförs skapas en serie detaljerade bilder av kroppen i ett specifikt område (Amerikanska cancerinstitute 2017), i vårt fall lungområdet. Enligt amerikanska cancerinstitutet (2017) har DT-skanningar visat sig vara till hjälp, både till att upptäcka cancer i ett tidigt skede och minska antal dödsfall.

## 1.1 Syfte

Den här studiens fokus ligger på utvärdering av förbehandlingstekniken *lung-segmentering*. Lung-segmentering är en förbehandlingsteknik som kan användas till att fokusera på ett problemområde i en bild och på så sätt samtidigt sälla bort irrelevant information, i vårt fall är problemområdet lungpartiet hos en högriskpatients DT-skanning.

Tidigare studier har ofta fokuserat kring nätverksarkitekturen som används vid lungcancerpredicering. Själva tekniken lung-segmentering och dess effekter har inte tidigare lyfts fram på det sätt som görs i denna studie.

För att kunna utvärdera lung-segmentering användes det dataset som fanns tillgängligt på hemsidan för KDSB 2017 tillsammans med ett CNN, vars arkitektur inspirerad av U-net som är känt för att vara bra vid predicering i medicinsk data (Ronneberger, Fischer & Bronx 2015). Datasetet förbehandlades med lung-segmentering och skalades till olika bildupplösningar och jämfördes mot samma data där datasetet inte blivit behandlad med lung-segmentering. Det dataset vi använde kom från KDSB och är en mängd DT-skanningar i *Digital Imaging and Communications in Medicine* (DICOM) format. DICOM är den standard som används för medicinsk data på de flesta sjukhus världen över (DICOM 2017). Varje fil i datasetet innehåller metadata från en högriskpatient och beskriver bland annat bildstorlek på de axiala tvärsnittsbilder en DT-skanning består av.

## 1.2 Frågeställning

Vi vill med denna studie, bidra till att kunna ge en högriskpatient ett säkrare besked om denne drabbats av lungcancer eller inte, därför ställde vi frågan: *Hur påverkas accuracy och loss hos en konvolutionell nätverksmodell när lung-segmentering appliceras på modellens tränings- och testdata?*

## 1.3 Avgränsning

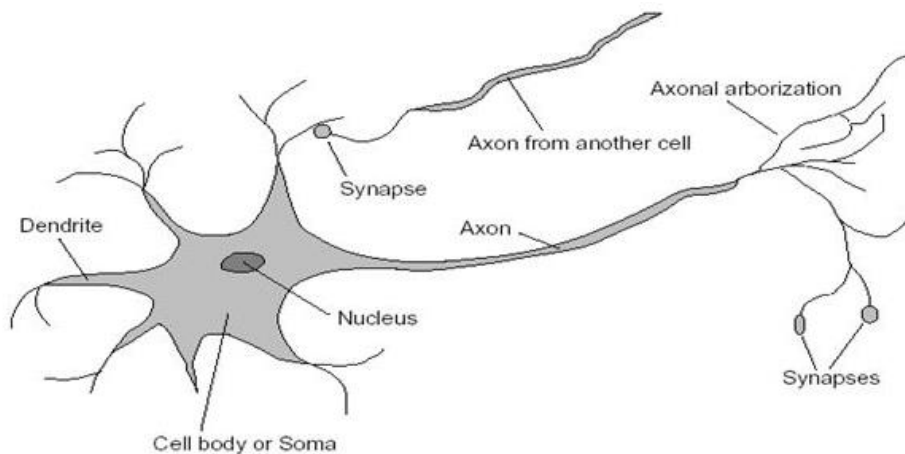
Experimenten som utförts i denna studie är avgränsat till att skala högriskpatienters tvärsnittsbilder från 512x512 pixlar till storlekarna 16x16, 32x32 och 64x64 pixlar och antal tvärsnittsbilder till 32 stycken. Bilderna har använts som test- och träningsdata för ett CNN. Samma experiment utfördes sedan en gång till men med skillnaden att bilderna förbehandlades med lung-segmentering.

## 2 Teori

I detta kapitel presenteras grundläggande information om vad ett artificiellt neuralt nätverk är och dess beståndsdelar. Vidare beskrivs grundläggande teori om hur ett neuralt nätverk lär sig ny information och tre olika typer av nätverk. I kapitlet förklaras också hur konvolutionsfunktionen fungerar och vad som utmärker ett CNN från andra typer av nätverk. Sist beskrivs några vanligt förekommande förbehandlingstekniker och en utvärderingsfunktion för neurala nätverk.

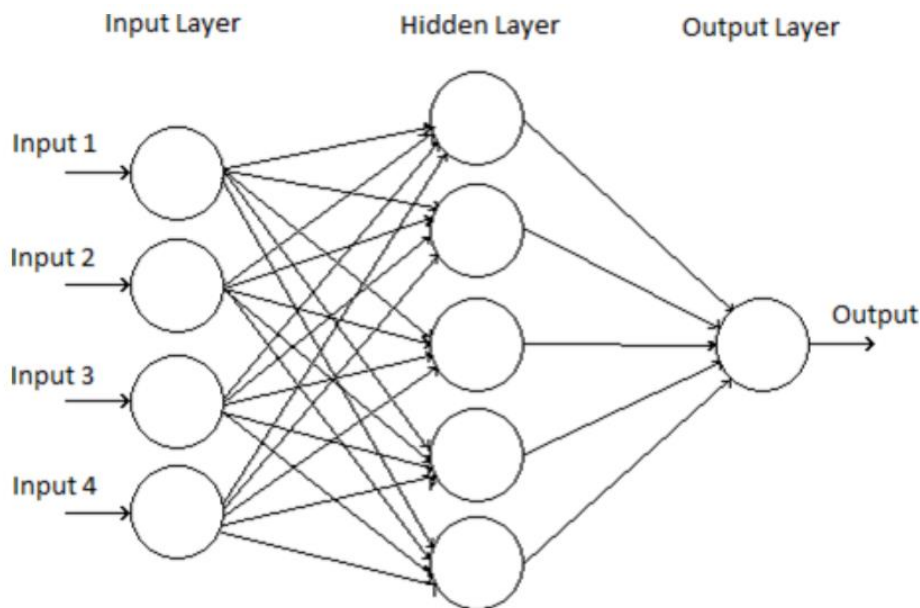
### 2.1 Artificiella neurala nätverk

Namnet *Artificiellt neuralt nätverk* (ANN) härstammar från dess biologiska liknelse, nämligen det mänskliga nervsystemet. Det mänskliga nervsystemet består av neuroner och i Figur 1.0 ses en representation av en mänsklig neuron. Biologiska neuroner av denna typ utgör en stor del av människans hjärna och neuronens funktion möjliggör mänskligt tänkande (Sal Khan 2011). Varje neuron innehåller *dendrit*er som sträcker ut från neuronens (cellens) kropp och dess uppgift är att ta emot elektriska impulser från andra länkade neuroner. Impulserna behandlas i cellkroppen som sedan skickas vidare i form av en impuls, genom dendriten. Dendriten skickar impulsen till axonomet och in till axonterminalerna som därefter skickar signalen vidare till andra förbindande neuroner (Sal Khan. 2011).



Figur 1.0: Mänsklig neuron (Russel. J, S, Norvig, P, 2010).

Ett ANN simulerar detta beteende med matematiska formler och algoritmer för att realisera ett system som kan demonstrera en viss grad intelligens. Ett artificiellt neuralt nätverk består av ett input lager av neuroner, upp till tre *hidden layers* bestående av neuroner och ett outputlager av output neuroner(Wang 2013).



Figur 1.1: Olika lager av neuroner i ett ANN

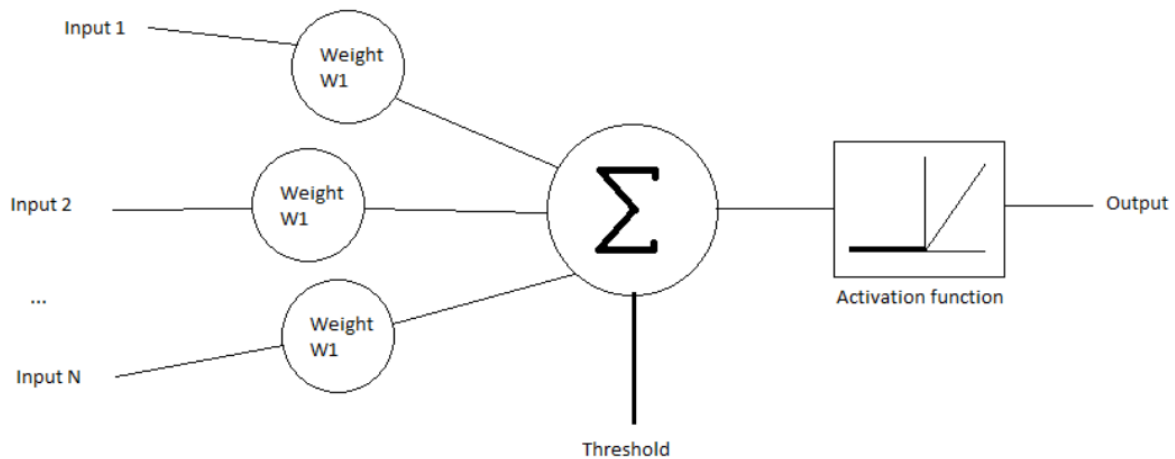
Neuronerna ANN använder är artificiella neuroner. En artificiell neuron är en beräkningsmodell inspirerad av den naturliga neuronerna men förenklad till ett matematiskt uttryck. ANN används för att behandla information och har vissa unika tillämpningsområden som, att studera beteende och kontroll hos djur och maskiner, tekniska områden som mönsterigenkänning, prognostisering och datakomprimering (Gupta 2013).

### 2.1.1 Inläring

Varje lager i ANN har en anslutning som är associerat med ett nummer kallat *vikt* (Wang 2013). Vikten hos en koppling visar influensen av kopplingen, med andra ord hur stark förbindelsen är mellan neuroner. Det biologiska motsvarandet till detta är hur stor inverkan en biologisk neurons avfyrande har på en annan. I denna typ av nätverk så summeras neuronernas input. Då input-neuronerna enbart har en input blir deras utdata den input som matades in multiplicerat med dess associerade vikt. Om värdet av vikten mellan två neuroner är stort så har inputen till den mottagande neuronerna större inflytande på resten av nätverket (Gupta 2013).

Det är möjligt att justera vikterna hos en artificiell neuron för att få fram önskade resultat, givet specifik input. Om nätverket innehåller tusentals neuroner så kan det bli komplicerat och tidskrävande att hitta och justera de rätta vikterna för hand. Lösningen på detta problem kommer i form av algoritmer som justerar vikterna automatiskt för att erhålla önskat resultat. Denna process kallas *inläring* eller *träning* (Gupta 2013). Likt mänsklig inläring lär sig ett neuralt nätverk av exempel. Inläringen för ett neuralt nätverk går till genom att utsätta nätverket för en uppsättning input-data som kallas för träningsdata. Meningen med träningen är att minimera förlusten, vilket görs genom att justera nätverkets vikter. Förlusten är vanligtvis summan av de kvadrerade skillnaderna mellan nätverkets utdata och den önskade utdata (Wang 2013).

Ett neuralt nätverk kan aldrig lära sig något som inte existerar i träningsdata. Därför måste mängden av träningsdata vara stor nog för att det neurala nätverket ska kunna memorera och fånga trender. Finns det däremot för mycket brus i träningsdata kan nätverket förstöra vikterna (Wang 2013). Därför är det viktigt att den data som används är relevant och har en godtycklig varians i sitt utseende.



Figur 1.2: Perceptron, en artificiell neuron.

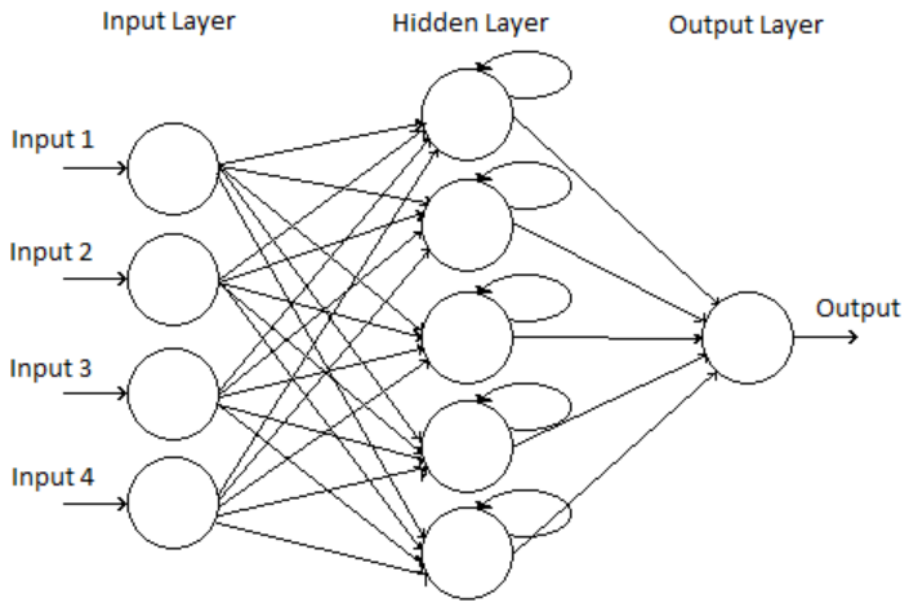
I Ekvation 1.0 visas den matematiska representationen av neuronerna i ANN där  $h_i$  är utdata av neuron  $i$  inom ett gömt lager (hidden layer) och är den så kallade *aktiveringsfunktionen* också kallad *överföringsfunktion*.  $N$  är antalet inmatningsneuroner,  $V_i$  är vikterna,  $x_j$  är indata till inmatningsneuroner och  $T_i^{hid}$  är tröskelvärdet för neuronerna i de gömda lagren (Wang 2013). Aktiveringsfunktionen används för att introducera irregulariteter i nätverkets flöde, men även för att binda värdet av neuroner så att nätverket inte blir paralyserat på grund av avvikande neuroner. En aktiveringsfunktion som förekommer ofta är *sigmoid* funktionen (Wang 2013). Funktionen har en karaktäristisk S-formad kurva ofta hänvisad som en sigmoid kurva.

$$h_i = \sigma \left( \sum_{j=1}^N V_{ij} x_j + T_i^{hid} \right)$$

Ekvation 1.0: Matematisk representation av en neuron

## 2.1.2 Fram-matande och rekurrenta nätverk

Beroende på vad för mönster nätverket följer så klassificeras ett nätverk i en av kategorierna, ett *fram-matande* (feed-forward) eller ett *rekurrent* nätverk. Figur 1.3 är ett praktiskt exempel på ett feed-forward nätverk där inga återkommande loopar existerar. Signalerna har en enkelriktad bana genom nätverket och associerar inputs med outputs. Ett rekurrent nätverk introducerar loopar i nätverket som låter signaler färdas i båda riktningarna. Nätverk av denna typ kan bli extremt komplicerade. Beräkningar som gjorts på input från ett tidigare skede matas tillbaka in i nätverket och ger nätverket ett slags minne (Wang 2013). Denna form av återkoppling är liknande återkopplingskretsar i elektronik. Utdata från neuron  $i$ , i det gömda lagret får nu istället formeln som i Ekvation 1.1 där tid ( $t$ ) är direkt indexerad för att betona den förlängda återkopplingen. När  $W_{ik}$  är noll så reduceras nätverkets beräkningar till det av feedforward nätverket i Ekvation 1.0.



Figur 1.3: Exempel på ett rekurrent nätverk

Utdata från neuron  $i$ , i det gömda lagret får nu istället formeln som i Ekvation 1.1 där tid ( $t$ ) är direkt indexerad för att betona den förlängda återkopplingen. När  $W_{ik}$  är noll så reduceras nätverkets beräkningar till det av feedforward nätverket i Ekvation 1.0.

$$h_i(t) = \sigma\left(\sum_{j=1}^H V_{ij}x_j(t) + \sum_{k=1}^H W_{ik}h_k(t-1) + T_i^{hid}\right)$$

Ekvation 1.1: Matematisk representation av en rekurrent neuron.

## 2.2 Konvolutionella neurala nätverk

*Konvolutionella neurala nätverk* (CNN) är speciellt lämpade för datastrukturer i matrisform, som till exempel bilder, vilka kan ses som tvådimensionella pixelmatriser. CNN är inte enbart ett nätverk med en stor mängd gömda lager, det är ett djupgående nätverk som imiterar den mänskliga hjärnans syncentrum angående processering och klassificering av bilder (Kim 2017). Det som menas med klassificering av bilder är det samma som bildigenkänning, då det kan anses vara samma sak att känna igen ett objekt i en bild, som att associera bilden till ett objekt.

I ett CNN används *konvolution* i stället för matrismultiplikation, vilket är vanligt bland andra neurala nätverk. konvolution utnyttjar två nyckelidéer som gör CNN till ett effektivt neuralt nätverk, dessa är sparsamma interaktioner (*sparse interactions*) och parameter delning (*parameter sharing*) (Goodfellow, Bengio & Courville 2016). För att effektivt kunna erhålla dugliga resultat från ett CNN behövs de viktigaste dragen i en bild lyftas fram, detta är känt som *feature extraction*. Exempel på feature extraction är att omvandla bilder med färg till svartvitt. Detta tar bort attribut som nätverket inte behöver lära sig och möjliggör en lättare träning för en CNN modell (Witten, Frank, Hall & Pal. 2016).

## 2.2.1 Konvolution

Som nämnt ovan är den konvolutionella operationen det som skiljer ett CNN från andra neurala nätverk. Grundformen av konvolution består av två funktioner som använder reella tal som argument. För att förklara konvolution, kan vi låtsas att det är möjligt att hålla koll på var en bil befinner sig med hjälp av en laser som ger en output:  $x(t)$ , där  $x$  är bilens position i tidssteg  $t$ . För att minska eventuellt brus som uppkommer vid mätningarna kan flera mätningar göras och ett medelvärde av dem användas som mätvärde. Senare mätningar har större värde än de äldre därför används en viktfunktion,  $w(a)$ , där  $a$  representerar hur gammal en mätning är. Viktfunktionen  $w$  måste vara en giltig täthetsfunktion (Gooch 2007). Om dessa viktade medelvärdemätningar utförs varje tidssteg kan det beskrivas med en funktion,  $s$ , som talar om var bilen befinner sig, detta illustreras i Ekvation 1.2 och kallas för konvolution (Goodfellow et al 2016).

$$s(t) = \int x(a)w(t-a)da$$

*Ekvation 1.2: Konvolutionsfunktion.*

Funktionen kan även skrivas med en asterisk som visas i Ekvation 1.3. I CNN-terminologi kallas första argumentet i konvolutionsfunktionen för *input* och det andra för *kernel*, det som returneras kallas för *feature map*.

$$s(t) = (x * w)(t)$$

*Ekvation 1.3: Konvolutionsfunktion med asterisk.*

För att exemplet med bilen ovan ska vara realistiskt, kan data inte samlas in i varje tidssteg då mängden hade blivit för stor, däremot i regelbundna intervall, exempelvis varje sekund eller minut. I ett sådant fall skulle tidsvariabeln  $t$  endast vara av heltalstyp, likaså variablerna  $x$  och  $w$ , då kan den matematiska *diskreta konvolutionen* definieras som i Ekvation 1.4 (Goodfellow et al 2016).

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$

*Ekvation 1.4: Diskret konvolution*

Goodfellow et al (2016) skriver att något som är vanligt inom maskininlärning är att input och kernel är en matris bestående av parametrar som anpassas av en algoritm. I CNN-terminologi kallas dessa matriser för tensors. Varje element i input och kernel måste lagras separat och kan antas vara noll på alla ställen förutom i den ändliga uppsättning punkter där värden sparas. I praktiken innebär detta att den icke ändliga summationen kan distribueras över en lista med ändligt många platser. I CNN används ofta konvolution över flera axlar samtidigt, det vill säga att input och kernel är av två- eller fler dimensioner. Exempelvis om den tvådimensionella bilden,  $I$ , används i ett CNN, kommer troligtvis kerneln  $K$  också vara tvådimensionell, detta kan ses i Ekvation 1.5.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i-m, j-n)$$

*Ekvation 1.5: Konvolution vid tvådimensionalitet.*

Konvolution är en kommutativ funktion, vilket innebär att operandernas ordning inte spelar någon roll. Funktionen kan därför skrivas om som i Ekvation 1.6 (Goodfellow et al 2016).

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n)$$

Ekvation 1.6: kommutativ representation av Ekvation 1.5.

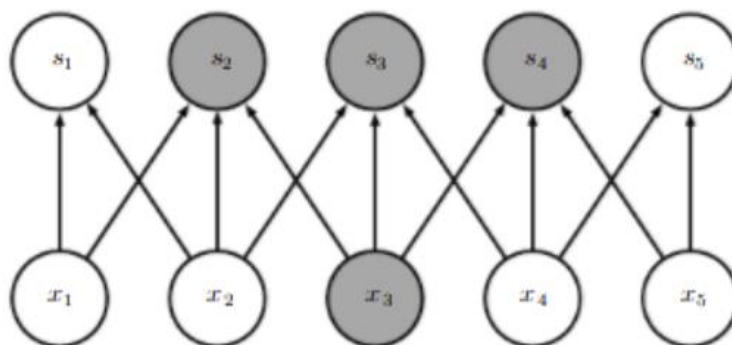
Enligt Goodfellow et al (2016) används den vända konvolutionen oftast, då den är enklare att implementera då det är mindre variation i intervallet av  $m$  och  $n$  värden. Funktionen används oftast för matematiska bevisningar, medan en liknande funktion kallad cross-correlation används vid implementation, där inte kerneln är vänd, se Ekvation 1.7.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n)$$

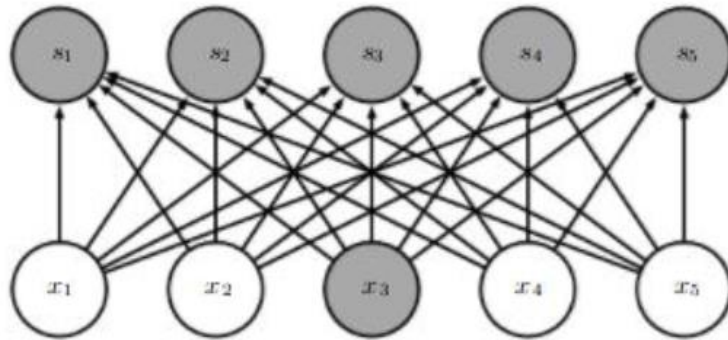
Ekvation 1.7: Cross-correlation

## 2.2.2 Sparsamma interaktioner och parameterdelning

Vid matrismultiplikation i neurala nätverk används en matris bestående av parametrar som bland annat beskriver interaktionen mellan alla inputs och alla outputs, detta kan beskrivas som att input interagerar med output. I CNN råder det däremot *sparsamma interaktioner* som även kallas *sparsamma vikter*. Sparsamma interaktioner, se Figur 1.4, uppnås genom att göra kernel mindre än input. Detta kan innebära en fördel då minnes-komplexiteten under körning blir  $O(k \times n)$  där  $k$  är antal anslutningar varje output  $n$  får ha, till skillnad mot  $O(m \times n)$  som minnes-komplexiteten är hos nätverk där matrismultiplikation, se Figur 1.5, används där  $m$  är varje input och  $n$  är varje output (Goodfellow et al 2016).

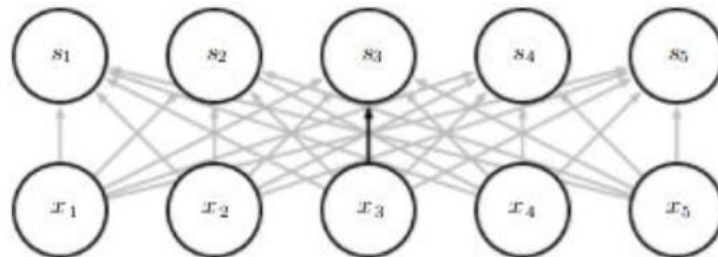


Figur 1.4: Sparsamma interaktioner där  $x$  noderna representerar inputs och  $s$  outputs. Här har  $k$ , som är antal anslutningar varje nod  $x$  får ha, begränsats till tre.



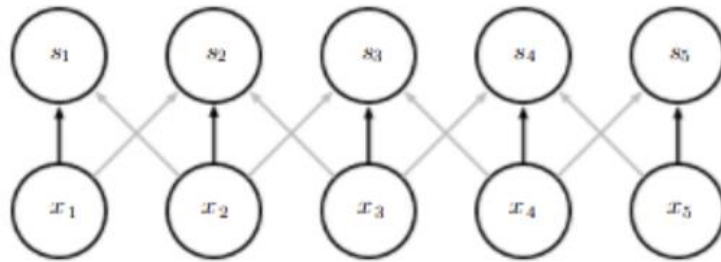
*Figur 1.5: Matrimultiplikation, som gör att komplexiteten ökar gentemot vid sparsamma interaktioner,  $x$  noderna representerar inputs och  $s$  output.*

I neurala nätverk där matrimultiplikation används utnyttjas varje element i en viktmatris endast en gång vid beräkning av output, vilket Figur 1.6 visualiserar. I CNN används parameterdelning vilket innebär att samma parameter kan utnyttjas för mer än en funktion i en modell (Karpathy 2015).



*Figur 1.6: Ingen parameterdelning. Den ensamma svarta pilen indikerar att endast en parameter används till en input.*

Detta innebär att konvolutionsfunktionen i ett CNN endast lär sig en uppsättning parametrar istället för en mängd som visualiseras i Figur 1.7. Minnes-komplexiteten hos ett CNN kommer inte att påverkas av parameterdelning, utan kommer fortfarande vara  $O(k \times n)$ . Däremot kommer minnesanvändningen reduceras till  $k$  parametrar (Goodfellow et al 2016).



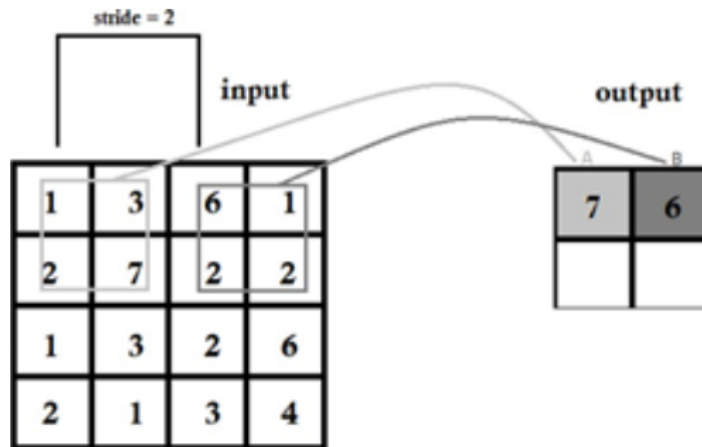
Figur 1.7: Parameterdelning där de svarta pilarna indikerar att samma parameter används till flera inputs.

### 2.2.3 Pooling

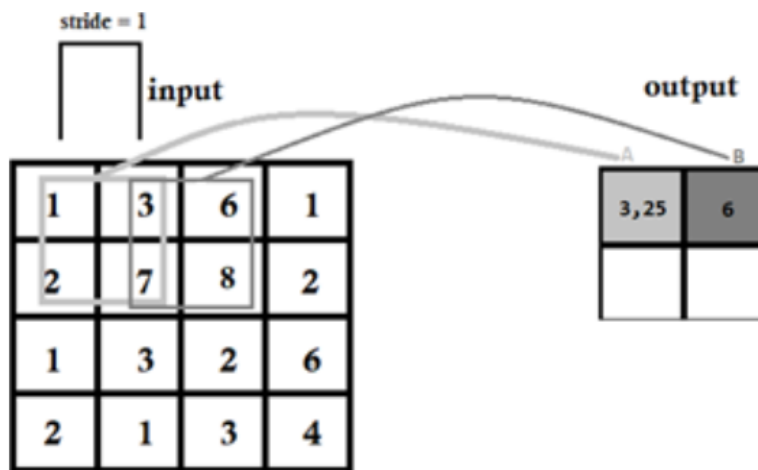
Ett CNN är uppbyggt av flera lager (layers) och enligt Goodfellow et al (2016) består ett typiskt CNN-lager av tre delar, där den första delen innehåller flera konvolutioner som körs parallellt för att producera en mängd linjära aktiveringar. I nästa del, som ibland kallas för *detector stage*, körs alla linjära aktiveringsfunktioner genom en icke-linjär aktiveringsfunktion. I den sista delen används en *poolingfunktion*. Karpathy (2015) skriver att det är vanligt att stoppa in ett pooling lager mellan varje konvolutionellt lager i ett CNN.

Det finns olika sorters poolingfunktioner, dock har de ett gemensamt syfte, att minska storleken av representationen så att mängden parametrar och beräkningar minskar för att minska chansen att nätverket *övertränar*. Diettrich (1995) förklarar att överträning händer då ett nätverk har tränat för länge på en viss data och börjat memorera avvikande värden, detta leder till att nätverket inte längre klassificerar data utöver träningsdata rätt. De vanligaste poolingfunktionerna kallas *max-* respektive *mean-pooling* (Karpathy 2015). I poolingfunktionen går det oftast att ställa in om det så kallade *filtret* ska överlappa (*stride*) och hur mycket.

Pooling kan ses som ett rörligt fönster som flyttas över en matris där ett värde väljs som representant för hela den matrisdelen som fönstret ser på. I max-pooling väljs det högsta och i mean-pooling är det medelvärdet som blir output från poolingfunktionen. I Figur 1.8 används ingen överlappning, i CNN är det dock viktigt att använda överlappning för att inte information om bilden ska försvinna, till exempel som var objekt befinner sig. Överlappning visualiseras i Figur 1.9 där mean-pooling används.



Figur 1.8: Max-pooling med ett 2 x 2 fönster utan överlappning (stride = 2).  
 $A = \max(1, 3, 2, 7) = 7$ .  $B = \max(6, 1, 2, 2) = 6$ .



Figur 1.9: Mean-pooling med ett 2 x 2 fönster med överlappning (stride = 1).  
 $A = (1+3+2+7)/4 = 3,25$ .  $B = (3+6+7+8)/4 = 6$ .

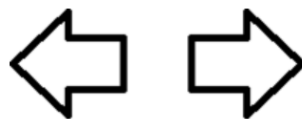
## 2.3 Förbehandlingstekniker

I Stanfords föreläsningmaterial skriver Karpathy (2015) att det finns tre, frekvent använda, förbehandlingstekniker att applicera på en datamatrix  $X$ , där  $X$  antas vara av storleken  $N * D$ , där  $N$  är antal data och  $D$  är antal dimensioner. Två av dem är *medelvärdes subtraktion* (zero-centering) och *normalisering* (normalization).

Medelvärdes subtraktion är den mest använda och innebär att medelvärdet subtraheras från varje individuell egenskap i datamängden, vilket resulterar i att dataklustret centreras kring dess ursprungspunkt längs varje dimension. I bilder kan exempelvis ett enda värde subtraheras från alla pixlar eller subtraheras separat över de tre färgkanalerna (Röd, Grön, Blå).

Normalisering är också en vanligt förekommande förbehandlingsteknik vid användandet av CNN och innebär att normalisera ett datasets dimensioner till samma skala. Det finns två vanliga sätt att uppnå normalisering, det första innebär att dividera varje dimension med sin standardavvikelse, efter att den är nollcentraliserad. Det andra sättet innebär att normalisera varje dimension så att det minsta respektive högsta värdet blir -1 respektive 1. Normalisering är endast meningsfullt att utföra om det finns anledning att tro att olika egenskaper har olika skalor eller enheter, vilket oftast inte är fallet i bilder då pixelvärdet ligger på intervallet 0 till 255.

Det är inte ovanligt att flera tusentals bilder används som dataset vid träning av ett CNN och tumregeln; ju mer data desto bättre, är ett vanligt tänk. Om det finns en brist på data att träna med så kan en teknik som kallas för *data augmentering* användas. Data augmentering innebär att utöka ett dataset, genom att vända och vrida bilder åt olika håll och vinklar, utan att pixlarna i bilden förstörs. Till exempel är bilderna i Figur 2.0 spegelvända och en människa kan snabbt se att det är samma motiv, en pil, medan ett CNN kan uppfatta den flippade bilden som en helt ny input. Vid användandet av fler bilder, det vill säga *ökandet* av antal unika inputs, *minskar* risken för att nätverket blir övertränat (Hammak 2017).



Figur 2.0: Spegelvänd pil.

Data augmentering kan utföras på datorns CPU under tiden nätverket tränas med GPU:n, vilket innebär att augmentering kan ses som en kostnadsfri operation, både sett till beräkningsbelastning och minnesanvändning (Krizhevsky, Sutskever & Hinton 2017).

## 2.4 Utvärderingsfunktion

Det finns en mängd olika utvärderingsfunktioner inom deep learning och maskininlärning, en frekvent använd metod för att visa skillnaden mellan två sannolikhetsfördelningar kallas för *cross-entropy*. I ett fall där det finns två klasser, A = Ej cancer och B = Cancer, så kan  $s(A)$ , där  $s$  är en sannolikhetsfunktion, ge 1.0 och  $s(B)$  ge 0.0. I detta fall skulle det motsvara att det är 100% säkert att det inte finns någon cancer för A respektive 0% cancer för B. Om då modellen skulle gissa på sannolikheterna:  $s(A) = 0.73$  och  $s(B) = 0.27$  så kommer cross-entropy funktionen, se Ekvation 1.8, svara hur nära modellens gissning var den verkliga sanningsdistributionen, i fallet ovan skulle det bli: 0.3147 för  $H = -1.0 \cdot \ln(0.73) + 0.0 \cdot \ln(0.27)$ , det är detta som kallas *förlust* (loss) (de Boer, Kroese, Mannor & Rubinstein 2005).

$$H(p, q) = - \sum_x p(x) \log q(x)$$

Ekvation 1.8: Cross-entropy.

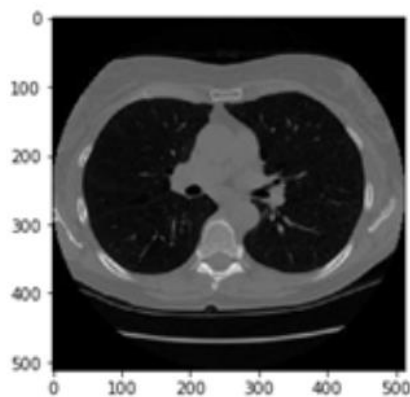
### 3 Metod

I detta kapitel beskrivs tillvägagångssättet för genomförandet av vår studie. För att kunna jämföra resultat från användandet av lung-segmentering har en experimentmetodik använts. Resultaten har analyserats med en kvantitativ metod. Analysen var ämnad att framställa prestandan hos tredimensionella konvolutionella nätverk som jobbat på data där lung-segmentering applicerats gentemot där ingen lung-segmentering använts.

#### 3.1 Medicinsk data

För att träna ett neuralt nätverk, djupgående eller inte, behövs data att träna på. Data som används i denna studie kommer från amerikanska cancerinstitutet och är även det dataset som använts i tävlingen som hölls av Kaggle, Data Science Bowl 2017 (Kaggle 2017). Datasetet består av patient-filer i *Digital Imaging and Communications in Medicine* format (DICOM), vilket är den officiella medicinska datatypen som används i hela världen för att beskriva och spara bildformat (DICOM 2017).

Från en DICOM-fil går det att extrahera en god mängd information bland annat över hur många axiala tvärsnittsbilder en patients lungröntgen består av och hur stort avstånd det är mellan dem, i Bilaga A visas informationen som ges vid öppnandet av en DICOM-fil med pythonbiblioteket Pydicom. En DT-skanning är av *tredimensionell* karaktär och är uppbyggd av flera *tvådimensionella* tvärsnittsbilder. Figur 2.1 är en slumpvald tvärsnittsbild från en av patienternas DICOM-filer. Bilden illustrerar patientens kropp, sedd från ovan. Då vi endast är intresserade av patientens lungor går det att sälla bort de mindre intressanta områden av en bild, såsom övriga organ, ben, muskelmassa och hud. Det går med lite finesse att ställa in det neurala nätverket att ignorera substanser som inte är en del av patientens lungparti.



Figur 2.1: Axial tvärsnittsbild av en DT-skanning, sedd från ovan.

Den officiella enheten för att mäta radiodensitet genom olika substanser i en DT-skanning är *Hounsfield Unit* (HU) (Svenska Strålmyndigheten 2017). I metadata från en DICOM-fil går det att utläsa från en pixel-array vilket HU-värde en viss pixel har i en bild från en DT-skanning. Tabell 1.0 visar vanligt förekommande substanser i en DT-skanning och dess HU-värde (Feeman 2015). Tillsammans med HU-tabellen och uteslutningsmetoden går det att sälla ut substanser som inte är av intresse för lungcancerpredicering.

Substans	HU
Skelett	+1000
Lever	+40 till +60
Vit materia	+20 till +30
Grå materia	+37 till +45
Blod	+40
Muskler	+10 till +40
Njure	+30
Ryggmärgsvätska	+15
Vatten	0
Fett	-100 till -50
Luft	-1000

Tabell 1.0: Hounsfield värde i vanlig förekommande substanser i en DT-skanning.

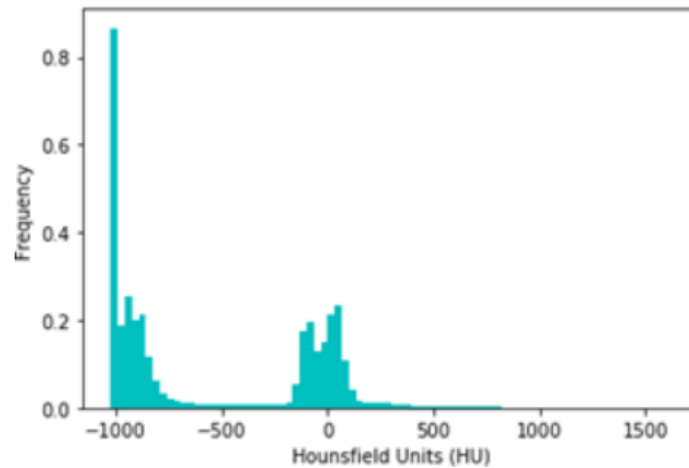
### 3.2 Applicering och utvärdering

Nätverket och de förbehandlingstekniker som använts har implementerats i python 3.6 med hjälp av Keras API och Tensorflow som backend. Data som användes i studien fanns tillgänglig på hemsidan för KDSB 2017. Data var obearbetad i den omfattningen att vissa DT-skanningar innehöll ofullständig och oanvändbar information. Sådan avsaknad information kunde vara om patienten har cancer eller inte, ej tillräcklig mängd bilder och för stort avstånd mellan axiala bilder. Den del av data som hade dessa problem behövdes filtrera bort och resulterade till att användbar mängd data gick från 1595 patienter till 1397 patienter. Fördelningen av patienter i den filtrerade datamängd blev 362 stycken med cancer och 1035 utan.

Keras sekventiella nätverk, vilket är den typ som användes, tar bilder i form av pixel matriser som input, vilket gör att pixlarna måste extraheras från DICOM-filerna. Olika DT-maskiner ger olika skanningar med olika samplingsfrekvenser i djupet på de tredimensionella bilderna. Detta avstånd, mellan successiva bilder, kallas *slice thickness* och kan variera uppemot 4 gånger i avstånd mellan skanningar. För att kunna applicera samma nätverksmodell på skanningar med olika avstånd är det nödvändigt att ändra storleken på bilderna så att alla följer samma generella upplösning.

För att få tag på pixlarna från en DICOM-fil måste filerna konverteras till Hounsfield units och lagras i 3 dimensionella numpy arrayer. När DICOM-filens data är i ett format som kan hanteras av python och Keras, är det möjligt att applicera ytterligare förbehandlingstekniker.

För att generalisera data så att nätverket inte får extra arbete med att lära sig en varierande slice thickness, *resamplas* data. Resampling innebär att avståndet ändras mellan varje pixel till en upplösning som är isotropisk. Avståndet mellan pixlarna för experimentet sattes till 1x1x1 mm. Förutom resampling använde vi oss dessutom av zero-centering och bild-normalisering i samtliga experiment. För att beräkna loss har vi använt oss av cross-entropy.



*Graf 1.0 Grafen visar förekomsten av Hounsfield units i en patients bild.*

### 3.3 Lung-segentering

Förbehandlingstekniken som denna studie har sitt fokus på, är lung-segentering. Att segmentera innebär att reducera ett problemområde genom att klippa bort delar av en bild som aldrig kommer innehålla det vi vill att nätverket ska lära sig att hitta, i detta fallet är målet att identifiera cancernoduler i lung-partiet. Med hjälp av informationen i Tabell 1.0 och Graf 1.0 kan vi se vilka pixlar som är luft och vilka som är cellvävnad. Genom att bestämma mängden luft runt lungorna så kan luftpixlarna utanför lungorna fyllas med ettor och luftpixlarna inuti lungorna fylls med nollor i en binär pixelmatris av bilden. Då har en lyckad separering av lungorna genomförts, detta kallas för segentering och visualiseras i Figur 2.2.



*Figur 2.2 Lung-segentering(Insidan av lungan är markerad som gul).*

Lung-segentering introducerar ett par problem för en del specialfall. Tekniken förlitar sig på att luften utanför lungorna inte kommer i kontakt med luften i lungorna detta gäller dock inte för patienter med en trakeostomi. Bilder innehållande mycket brus kan också få tekniken att fel.

### 3.4 Nätverksarkitektur

Ingen större vikt lades på att trimma arkitekturen till att bli den bäst lämpade för lungcancerpredicering. Detta på grund av att vi saknade djupare kunskap inom området och att det är förbehandlingstekniken lung-segmentering som skulle utvärderas, inte nätverksarkitekturer. Nätverket vi använde i samtliga experiment är inspirerat av U-net som har visats sig vara effektivt vid predicering i biomedicinsk data (Ronneberger, Fischer & Brox 2015). Nätverket kan ses i Figur 2.3. Då Keras uppdaterat sitt api sedan 2015 när U-net skapades, var vi tvungna att göra vår egen approach på nätverket med hjälp av en sekventiell modell med en input och en output.



Figur 2.3: Nätverkets arkitektur, alla lager och dess parametrar.

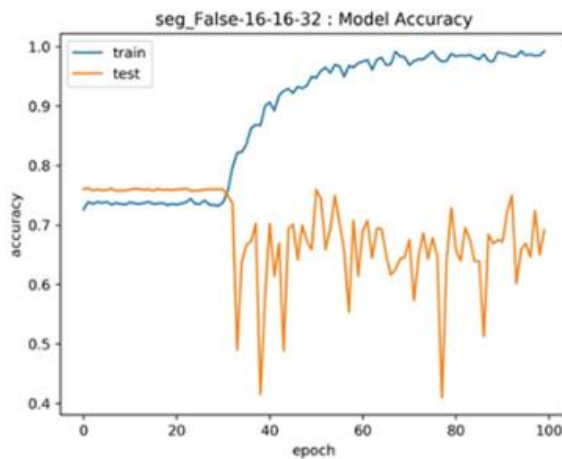
Vårt nätverk bestod av fem stycken block där samtliga innehöll ett eller flera tredimensionella konvolutionella lager. Samtliga konvolutionslager saknade överlappning och använder relu aktiveringsfunktion. Första blocket består av input, average-pooling, konvolution och ett max-poolinglager. Andra blocket inleds med ett konvolutionslager och övergår till ett max-poolinglager. Det tredje blocket består av två konvolutionslager som övergår till ett max-poolinglager. Det fjärde blocket inleds även det med två konvolutionslager som även de övergår till ett max-pooling lager. Det sista, femte, blocket är uppbyggt av ett konvolutions-, utplattnings- och två klassificeringslager.

### 3.5 Hårdvara

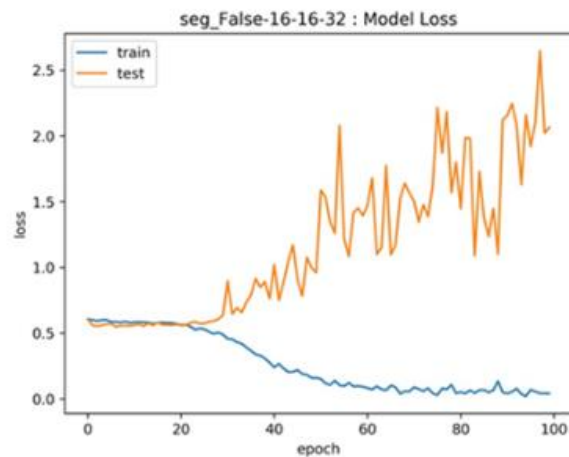
Att skapa ett neuralt nätverk är både avancerat och prestandakrävande. Den lungdata som använts i detta arbete tar upp över 60 *gigabyte* (GB) hårddiskutrymme. Att evaluera på 60 GB data över 100 epoker tar en orimligt lång tid för en persondator att beräkna. Men tack vare Högskolan i Borås så har all evaluering och träning skett på deras maskin. Maskinen som möjliggjort att samtliga experiment kunnat bli utförda inom rimlig tid var belägen på Högskolan i Borås och är en HPE ProLiant DL580 Gen9 Server med 256 GB RAM och 4 Xeon CPU E7-4830 v3 2.10 GHz med totalt 48 kärnor, samt ett Tesla K40c grafikkort med 12 GB.

## 4 Resultat

I detta kapitel presenteras de resultat som tagits fram gällande dataanalysen för att utvärdera lung-segnterings effektivitet och påverkan, när det appliceras på lungdata. Experimenten utfördes på Högskolan i Borås mellan vintern 2017 och våren 2018 och bestod av två huvudsakliga delar. Den första att skala patientens DT-skanning från originalstorleken 512x512 pixlar till storlekarna 16x16, 32x32 och 64x64 pixlar. Den andra delen var att variera körningarna med och utan lung-segntering. Samtliga körningar använde djup 32, det vill säga 32 stycken tvärsnittsbilder. Mätvärdena som erhöles från varje körning var modellens *träffsäkerhet* (accuracy) och modellens loss under varje *epok*. Accuracy är ett mått på statistisk varians (Taylor 1997). För att kunna urskilja något samband med och utan lung-segntering så genererades ett antal grafer som utmärker modellens accuracy och loss per epok i totalt 100 epoker. Den blå linjen i graferna visar modellens prestation på träningsdata och den orangea linjen på testdata.

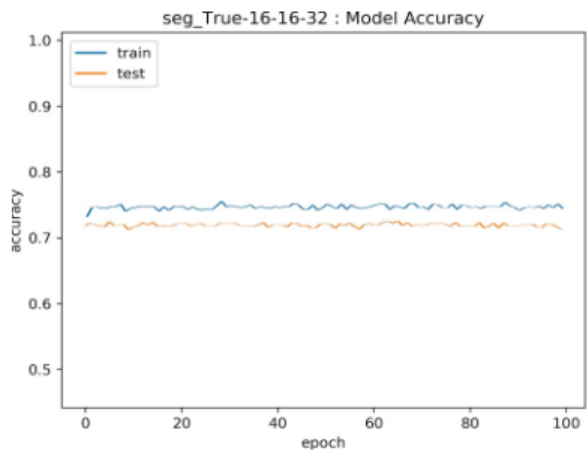


Graf 1.1: Accuracy.

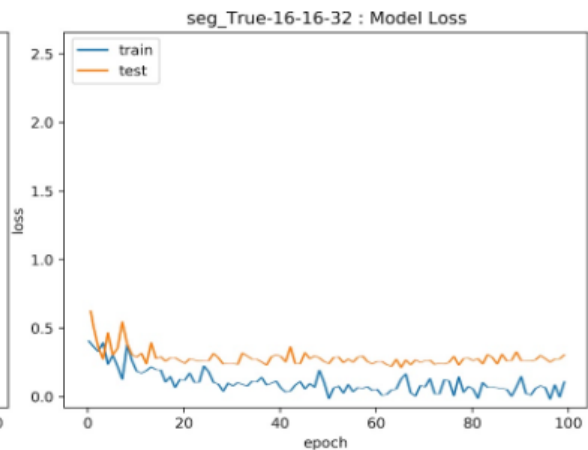


Graf 1.2: Loss.

I Graf 1.1 presenteras en nätverksmodells prestation mätt i accuracy och Graf 1.2 visar samma modells loss på ej lung-segnterad data under 100 epoker med inputstorlek 16x16x32. Graferna visar att träningens accuracy och loss blir bättre, framförallt efter 20 epoker och går upp till 100 procent förtroende. Men för evaluering på testdata så fluktuerar modellens prediktioner vilket redogörs med en ökad loss och minskad accuracy. Här kan det härledas att nätverket förstörs och är oanvändbart.

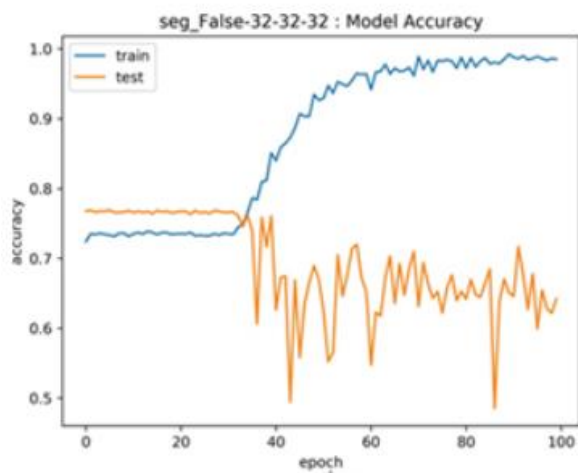


Graf 1.3: Accuracy.

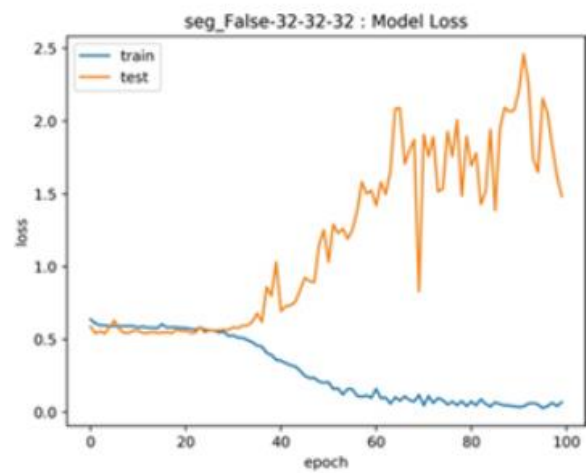


Graf 1.4: Loss

Graferna 1.3 och 1.4 visar också accuracy och loss för inputstorlek 16x16x32 som i Graf 1.1 och Graf 1.2, med skillnaden att data har lung-segenteras. I Graf 1.3 och 1.4 framgår det att accuracy och loss dramatiskt bättras efter de första epokerna. Det blir ingen ökning i accuracy efter 80+ epoker, samtidigt som loss minskar efter var tionde epok. Mindre loss innebär att modellen blir bättre på att utvärdera data.

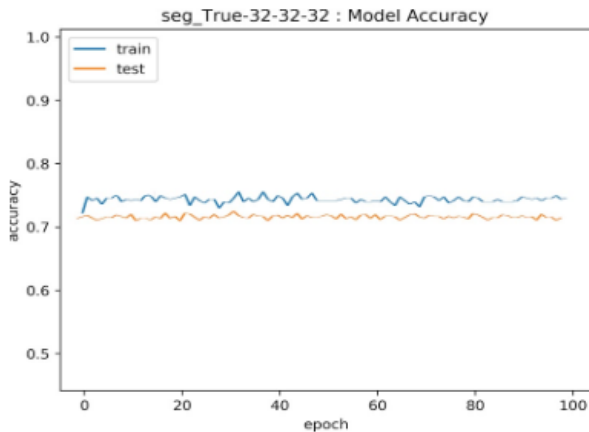


Graf 1.5: Accuracy.



Graf 1.6: Loss.

I Graf 1.5 och 1.6 presenteras en nätverksmodells prestationer på ej lung-segenterad data med inputstorlek 32x32x32. Resultaten följer de som erhöles i körningen med 16x16x32 inputstorlek med skillnaden att det tar dubbelt så många epoker tills det förstörs.



Graf 1.7: Accuracy.

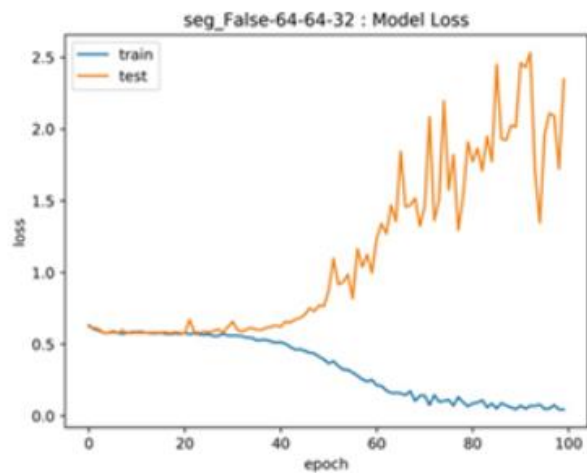


Graf 1.8: Loss.

Graf 1.7 och 1.8 visar resultatet från inputstorlek 32x32x32 då data har lung-segenterats. Här går det att se att modellen tränas normalt och får ett slutresultat på 72 % accuracy och loss på 0.597.

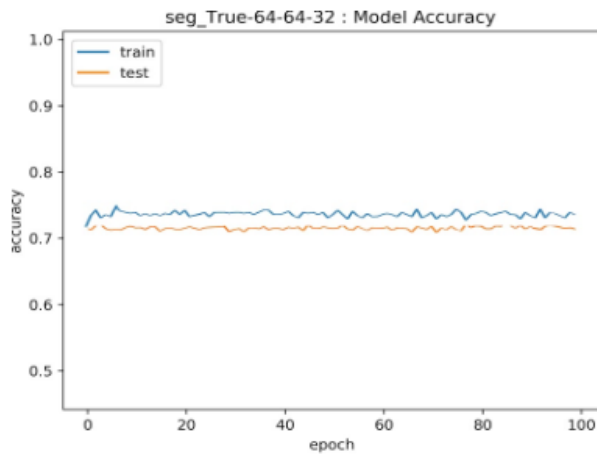


Graf 1.9: Accuracy.



Graf 2.0: Loss.

I Graf 1.9 och 2.0 visas träning- och testresultat av en nätverksmodell med inputstorlek 64x64x32 då lung-segentering inte har använts. Resultaten liknar de som framgår i graferna 1.5 och 1.6 med skillnaden att antal epoker innan nätverksmodellen förstörs ökat med tio.



Graf 2.1: Accuracy.



Graf 2.2: Loss.

I Graf 2.1 och 2.2 presenteras resultatet av en nätverksmodell som tränat på lung-segenterad data med inputstorlek 64x64x32. Likt modellerna som visas i graferna 1.3, 1.4, 1.7 och 1.8 där inputdata har lung-segenterats, tränas även denna modell korrekt. Accuracy efter 100 epoker landar på 72.5%. Samtliga modellers resultat redovisas i Tabell 1.1 nedan.

Storlek	Segmentering	Träning	Test	Accuracy	Loss
16x16	✓		✓	72.5%	0.589
16x16			✓	69.0%	2.065
16x16	✓	✓		74.3%	0.575
16x16		✓		99.5%	0.050
32x32	✓		✓	72.0%	0.597
32x32			✓	65.0%	1.500
32x32	✓	✓		74.6%	0.568
32x32		✓		99.00%	0.100
64x64	✓		✓	74.4%	0.588
64x64			✓	65.8%	2.358
64x64	✓	✓		72.5%	0.571
64x64		✓		98.5%	0.042

Tabell 1.1: Tabellen visar resultat från samtliga körningar, om lung-segentering använts, accuracy i procent, loss, om det är träning- eller testdata efter 100 epoker.

## 5 Diskussion

Resultaten från de tränade modellerna har visat att lung-segmentering har stor betydelse vid predicering av lungcancer, åtminstone vid användandet av den arkitektur vi haft. Vad resultaten betyder och hur användbara de är kommer avhandlas i detta kapitel tillsammans med diskussion om studiens metod samt arbete.

### 5.1 Resultatdiskussion

I de tester där lung-segmentering inte använts kan det härledas att en högre pixelupplösning påverkade antalet epoker det tar för modellen att lära sig data, vilket går att utläsa i graferna 1.1, 1.2, 1.5, 1.6, 1.9 och 2.0. Den förlängda mängd tid det tar för nätverket att lära sig, kan bero på att det finns fler egenskaper att memorera. Precis som för människor, tar det längre tid för ett artificiellt nätverk att lära sig flera saker gentemot en nedskalad mängd. När modellernas träningsaccuracy i graferna 1.1, 1.5 och 1.9 ökar, ökar även testloss drastiskt, se grafer 1.2, 1.6 och 2.0. Ett liknande samband går att se gällande träningsloss kontra testaccuracy. Under träningen ökar modellernas pålitlighet i sina prediktioner ända upp till ett värde av 1.0, vilket innebär 100 % prediktionssäkerhet. Men detta skiljer sig från testresultaten som visar på en lika snabb minskning i testaccuracy ända ner till ett värde av 0.65, eller 65 % prediktionssäkerhet. Från resultaten av testkörningen går det att utläsa ett tydligt fall av överträning. Det är inte helt klart varför modellerna som inte använder lung-segmentering blir övertränade. Diettrich (1995) säger att det finns en risk för överträning när en modell arbetar för hårt för att hitta den bästa matchningen till träningsdata. Överträningen är en konsekvens av att nätverket memorerar diverse särskilda egenskaper som kanske inte existerar i verklig data, i detta fall, testdata. Det som uppstår när obearbetad pixeldata används för träning är att efter en viss tid in i träningen lär sig modellen data som inte hör till cancerpredicering, vilket beror på att bilderna innehåller information om muskelvävnad, skelett och andra substanser som påverkar modellens beslut. I slutändan försöker nätverket troligtvis utvärdera vilken typ av kropp patienterna har, se Figur 2.1, vilket fallerar när det kommer till en ja eller nej fråga om vad patienten ska klassificeras som. I korrelation till experimenten där lung-segmentering används, grafer 1.3, 1.7, och 2.1, finns en stark kontrast i hur probat prognostiseringen av lungcancer förhåller sig. Både träning och tester för lung-segmentering exponerar positiva resultat för modellen. Likaledes framgår det i graferna 1.4, 1.8 och 2.2 att loss för modellerna blir mindre per epok, vilket kan tolkas som att en modell blir mer självsäker på datasetet. En annan märkbar positiv aspekt av lung-segmentering är att chanserna att nätverket övertränas, minskar.

## 5.2 Metoddiskussion

Då denna rapport avser att utvärdera och jämföra vad skillnaden blir i accuracy respektive loss vid predicering med respektive utan lung-segmentering använde vi en kvantitativ experiment-metodik. Chalmers (1999) säger att experiment och påståenden bidrar till den vetenskapliga utvecklingen och att varje nytt forskningsresultat utgör en mer tillförlitlig och stringent bas än dess föregångare. Trots detta finns det människor som påpekar aspekten att teorier härleds från experiment och att experiment skapas av teorier, som negativ. En annan negativ aspekt på kvantitativ forskning skulle kunna vara att resultat från experiment visar vad som sker men inte varför. När man arbetar med neurala nätverk och inlärningsalgoritmer är förbehandling av data som ska användas viktigt och kan påverka prestandan i olika riktningar. På grund av brist på tid och kunskap valde vi att endast utvärdera vårt nätverk med och utan lung-segmentering i tre olika bildupplösningar, vilket högst troligt leder till att modellen inte beter sig optimalt. Samma uppsättning ingångsdata användes för samtliga modeller i varje experiment, vilket är ett krav för en jämförande studie. Valet av data var mycket begränsat, dels då medicinsk data oftast är skyddad av någon form av patient- skyddslag och dels för att den är väldigt specifik. Prestandan av modellerna kan demonstrera vikten av att ha en stor och varierad mängd data för maskininlärning. Den data som modellen utvärderats på innehöll enbart 365 patienter med cancer gentemot 1035 patienter utan, detta kan innebära på att modellerna inte har tränats tillräckligt på att tyda cancer. Data har inte granskats bild för bild vilket betyder att det kan finnas bilder i modellernas tränings- eller test-data som innehåller irrelevant eller ingen pixeldata. Eftersom vi valde att endast köra med 32 stycken bilder per patient kan relevant data gått förlorad. Som en konsekvens från en bristande kontroll kan modellerna ha lärt sig något som ej berör cancer eller lungor. Därför kan det diskuteras om balansen mellan att ha god data gentemot stor mängd data. Det är en förnödenhet att ha en stor mängd data men även att data är av god kvalitet. Som nämnts tidigare i metodkapitlet så kan kvaliteten på bilderna variera från CT-skanner till CT-skanner, vilket också kan ha påverkat modellernas prestanda. Det är mycket krävande för en dator att köra den typen av nätverk som vi använt oss av då det bland annat är stora matriser med bilddata som ska loopas igenom, flera gånger. En konsekvens av detta är att det tar lång tid att segmentera lungbilderna. De toppresterande personerna i KDSB 2017 har kört sina experiment på kraftfulla maskiner för att kunna hinna med förbehandling, träning och validering inom rimlig tid. Det närmaste vi kunde komma sådana typer av maskiner var en belägen på Högskolan i Borås, den var tillgänglig endast ett fåtal stunder tillsammans med vår handledare. Det var därför viktigt att vår kod fungerade som den var tänkt när maskinen var tillgänglig. Det hela blir ett mindre dilemma då det är svårt att testa om koden fungerar som tänkt utan att ha tillgång till en så pass kraftfull maskin. I rådande brist på kunskap valde vi att använda Keras python API vilket också kan ha haft en påverkan på tränings och testresultaten gentemot andra liknande API:n som förekommer.

## 6 Slutsats

Problemområden för bildigenkänning eller klassificering, där konvolutionella nätverk är applicerbara och överträning är ett prevalent problem. Studien visar att det finns goda förmåner med att använda lung-segmentering. Ett svar på frågan, *hur påverkas accuracy och loss hos en konvolutionell nätverksmodell när lung-segmentering appliceras på modellens tränings- och testdata*, är att överträning motverkas.

I den här studien har fokus lagts på segmentering av lungor, detta stoppar inte tekniken från att utvidgas och tillämpas på andra områden där bilder behandlas med neurala nät. Utan bör även kunna appliceras på andra bilddata, speciellt i områden där det finns ett behov av att fokusera på en eller flera delområden av en bild.

## Källförteckning

- American Cancer Institute. (2017). *Cancer Statistics*. Tillgänglig: <https://www.cancer.gov/about-cancer/understanding/statistics>. [Besökt 27/11 2017].
- Chalmers, A., (2013). *What is this thing called science?* 4th ed., St Lucia, Qld.: UQP.
- Data Science Bowl. (2017). *About The Data Science Bowl*. Tillgänglig: <https://datasciencebowl.com/about/>. [Besökt 29/11 2017].
- de Boer, P., Kroese, D., Mannor, S. & Rubinstein, R. (2005). *A Tutorial on the Cross-Entropy Method*. *Annals of operations research*, 134(1), pp.19–67.
- DICOM. (2017). *The DICOM Standard*. [ONLINE] Tillgänglig: <http://www.dicomstandard.org/about/>. [Besökt 27/11 2017].
- Diettrich, G, T. (1995). *Overfitting and Undercomputing in Machine Learning*. ACM Comput. Surv.
- Everitt, BS. (2002). *The Cambridge Dictionary of Statistics*, Cambridge University Press, Cambridge. Tillgänglig: ProQuest Ebook Central. [Besökt 28/12 2017].
- Feeman, T. & SpringerLink Content Provider. (2015). *The Mathematics of Medical Imaging A Beginner's Guide* (2nd ed. 2015. ed., Springer Undergraduate Texts in Mathematics and Technology).
- Gooch, J.W. (2007). *Encyclopedic Dictionary of Polymers*. Springer, New York, NY.
- Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*. MIT Press.
- Gupta, N, (2013). *Network and Complex systems*, Institute of Engineering and Technology.
- Hammak, D. (2017). *2017 Data Science Bowl, Predicting Lung Cancer: 2nd Place Solution Write-up, Daniel Hammack and Julian de Wit*. Tillgänglig: <http://blog.kaggle.com/2017/06/29/2017-data-science-bowl-predicting-lung-cancer-2nd-place-solution-write-up-daniel-hammack-and-julian-de-wit/>. [Besökt 26/1 2018].
- Kaggle. (2017). *Data Science Bowl 2017*. Tillgänglig: <https://www.kaggle.com/c/data-science-bowl-2017/data>. [Besökt 27/11 2017].
- Karpathy, A. (2015). *Convolutional Neural Networks for Visual Recognition*. Tillgänglig: <https://cs231n.github.io/> [Besökt 16/1 2018]
- Kim P. (2017) *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*.
- Krizhevsky, A., Sutskever, I., E.Hinton, G. (2017). *ImageNet Classification with Deep Convolutional Neural Networks*. *Communications of the ACM*, 60(6), pp.84–90.

Pitts, A. (2013). *Nominal Sets: Names and Symmetry in Computer Science*. Cambridge University Press.

Pydicom. (2017). *Pydicom readme, github*. Tillgänglig: <https://github.com/pydicom/pydicom> [Besökt 22/1 2018].

Radcliffe, T. (2016). *Python vs. C/C++ in Embedded Systems*. Tillgänglig: <https://www.activestate.com/blog/2016/09/python-vs-cc-embedded-systems>. [Besökt 20/12 2017].

Ronneberger, O., Fischer, P. & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*.

Russel, J, S, Norvig, P, (2010). *Artificial Intelligence: A Modern Approach*, Pearson Education, Inc.

Sal Khan. (2011). *Anatomy of a neuron*. Tillgänglig: <https://www.khanacademy.org/science/biology/human-biology/neuron-nervous-system/v/anatomy-of-a-neuron>. [Besökt 8/12 2017].

Svenska Socialstyrelsen. (2017). *Statistik om nyupptäckta cancerfall 2015*. Tillgänglig: <http://www.socialstyrelsen.se/Lists/Artikelkatalog/Attachments/20462/2017-1-14.pdf>. [Besökt 27/11 2017].

Svenska Strålmyndigheten. (2017). *Datortomografi*. Tillgänglig: <https://www.stralsakerhetsmyndigheten.se/omraden/stralning-i-varden/berattigande-och-optimering/datortomografi/>. [Besökt 7/12 2017].

Taylor, J.R. (1997). *An introduction to error analysis : the study of the uncertainties in physical measurements* 2. ed., Sausalito, Calif.: University Science.

Wang, S, (2013). *Interdisciplinary Computing in Java Programming*, Springer-Verlag.

Witten, I., Frank, E., Hall, M., & Pal, C. (2016). *Data Mining, 4th Edition (4th ed.)*. Morgan Kaufmann.

Zeiler, M.D. & Fergus, R., (2013). *Visualizing and Understanding Convolutional Networks*. Cornell University.

## Appendix A

No of slices:	195
(0008, 0000) Group Length	UL: 358
(0008, 0005) Specific Character Set	CS: 'ISO_IR 100'
(0008, 0016) SOP Class UID	UI: CT Image Storage
(0008, 0018) SOP Instance UID	UI:
1.3.6.1.4.1.14519.5.2.1.7009.9004.321555830121981826540353244716	
(0008, 0060) Modality	CS: 'CT'
(0008, 103e) Series Description	LO: 'Axial'
(0010, 0000) Group Length	UL: 64
(0010, 0010) Patient's Name	PN: '0015ceb851d7251b8f399e39779d1e7d'
(0010, 0020) Patient ID	LO: '0015ceb851d7251b8f399e39779d1e7d'
(0010, 0030) Patient's Birth Date	DA: '19000101'
(0018, 0060) KVP	DS: ''
(0020, 0000) Group Length	UL: 390
(0020, 000d) Study Instance UID	UI:
2.25.51820907428519808061667399603379702974102486079290552633235	
(0020, 000e) Series Instance UID	UI:
2.25.80077681474898947928901837852399466259680906162240840687333	
(0020, 0011) Series Number	IS: '2'
(0020, 0012) Acquisition Number	IS: '1'
(0020, 0013) Instance Number	IS: '1'
(0020, 0032) Image Position (Patient)	DS: ['-177.500000', '-177.500000', '-384.109985']
(0020, 0037) Image Orientation (Patient)	DS: ['1.000000', '0.000000', '0.000000', '0.000000', '1.000000', '0.000000']
(0020, 0052) Frame of Reference UID	UI:
2.25.39085688508687805643860350141257111202569461626559365918180	
(0020, 1040) Position Reference Indicator	LO: 'SN'
(0020, 1041) Slice Location	DS: '-384.109985'
(0028, 0000) Group Length	UL: 200
(0028, 0002) Samples per Pixel	US: 1
(0028, 0004) Photometric Interpretation	CS: 'MONOCHROME2'
(0028, 0010) Rows	US: 512
(0028, 0011) Columns	US: 512
(0028, 0030) Pixel Spacing	DS: ['0.693359', '0.693359']
(0028, 0100) Bits Allocated	US: 16
(0028, 0101) Bits Stored	US: 16
(0028, 0102) High Bit	US: 15
(0028, 0103) Pixel Representation	US: 1
(0028, 0120) Pixel Padding Value	US or SS: b'0\xf8'
(0028, 0301) Burned In Annotation	CS: 'NO'
(0028, 0303) Longitudinal Temporal Information M	CS: 'MODIFIED'
(0028, 1050) Window Center	DS: '-600'
(0028, 1051) Window Width	DS: '1500'
(0028, 1052) Rescale Intercept	DS: '-1024'
(0028, 1053) Rescale Slope	DS: '1'
(7fe0, 0010) Pixel Data	OB or OW: Array of 524288 bytes

**Högskolan i Borås** är en modern högskola mitt i city. Vi bedriver utbildningar inom ekonomi och informatik, biblioteks- och informationsvetenskap, mode och textil, beteendevetenskap och lärarutbildning, teknik samt vårdvetenskap.

På **institutionen Handels- och IT-högskolan (HIT)** har vi tagit fasta på studenternas framtida behov. Därför har vi skapat utbildningar där anställningsbarhet är ett nyckelord. Ämnesintegration, helhet och sammanhang är andra viktiga begrepp. På institutionen råder en närhet, såväl mellan studenter och lärare som mellan företag och utbildning.

Våra **ekonomiutbildningar** ger studenterna möjlighet att lära sig mer om olika företag och förvaltningar och hur styrning och organisering av dessa verksamheter sker. De får även lära sig om samhällsutveckling och om organisationers anpassning till omvärlden. De får möjlighet att förbättra sin förmåga att analysera, utveckla och styra verksamheter, oavsett om de vill ägna sig åt revision, administration eller marknadsföring. Bland våra **IT-utbildningar** finns alltid något för dem som vill designa framtidens IT-baserade kommunikationslösningar, som vill analysera behov av och krav på organisationers information för att designa deras innehållsstrukturer, bedriva integrerad IT- och affärsutveckling, utveckla sin förmåga att analysera och designa verksamheter eller inrikta sig mot programmering och utveckling för god IT-användning i företag och organisationer.

**Forskningsverksamheten** vid institutionen är såväl professions- som design- och utvecklingsinriktad. Den övergripande forskningsprofilen för institutionen är handels- och tjänsteutveckling i vilken kunskaper och kompetenser inom såväl informatik som företagsekonomi utgör viktiga grundstenar. Forskningen är välrenommerad och fokuserar på inriktningarna affärsdesign och Co-design. Forskningen är också professionsorienterad, vilket bland annat tar sig uttryck i att forskningen i många fall bedrivs på aktionsforskningsbaserade grunder med företag och offentliga organisationer på lokal, nationell och internationell arena. Forskningens design och professionsinriktning manifesteras också i InnovationLab, som är institutionens och Högskolans enhet för forskningsstödande systemutveckling.



HÖGSKOLAN  
I BORÅS

BESÖKSADRESS: JÄRNVÄGSGATAN 5 · POSTADRESS: ALLÉGATAN 1, 501 90 BORÅS  
TFN: 033-435 40 00 · E-POST: INST.HIT@HB.SE · WEBB: WWW.HB.SE/HIT