Restaurering av gamla foton med maskininlärning

Benjamin Hägglund

Kandidatavhandling i datavetenskap
Handledare: Luigia Petre
Fakulteten för naturvetenskaper och teknik
Åbo Akademi
Våren 2022

Innehållsförteckning

1. Inledning

Då man fotograferar i icke-ideala omständigheter, efterbehandlar, komprimerar eller laddar upp bilder, kan den tagna bilden bli oklar bland annat av brus, rörelseoskärpa eller linsoskärpa, det kan också uppstå fysiska defekter på grund av dålig förvaring av bilden som gamla bilder utsetts ofta för. Man kan minska en begränsad mängd av dessa defekter och inverkan med bra utrustning och kunnande, samt olika filter såsom median eller Weiner filter [1]. Båda används mot Gaussisk brus i bilder men Weiner kan appliceras i allt var det finns signaler såsom i audio- och video processering [4]. Med dessa filter det är inte möjligt att ta bort visuell störning, suddighet eller fysisk skada från bilden. Den moderna maskinlärningen (ML) har potentialen att lösa fall som inte är möjligt med något annat verktyg, genom att först analysera bildernas strukturer och sedan filtrera de oönskade pixlarna från bilden [11].

I denna avhandling presenteras två toppmoderna maskninlärningsmodeller för restaurering av gamla bilder, var ena är byggd för skaderestoration och andra för färgläggning. Vi visar typiska exempel på olika sorter av bildkorruption samt matar personliga foton i modellerna. I sektion 2 går vi igenom ML grunderna och processen av bygget av en generisk ML modell, i sektion 3 ingående förklaring på båda modellerna, i sektion 4 presenteras restorationsresultaten, och i sektion 5 befinns slutsatsen.

2. Maskininlärning

IBM beskriver ML som följande: “Maskininlärning är en förgrening av artificiell intelligens (AI) som fokuserar på att bygga applikationer som lär sig av data och förbättrar deras noggrannhet över tid utan att vara programmerad att göra det” [2]. Inom datavetenskap används maskininlärningsmodeller bland annat för datautvinning, klassificering av data och förutsägelse [3]. I detta fall försöker restorationsmodellerna förutsäga hur bilderna ser ut i originaltillstånd och -färg.

2.1 Processen för att bygga en ML modell

Även om det finns dussintals med olika maskininlärningsalgoritmer är modellbyggningsprocessen liknande för alla, de byggs upp av fyra steg: Bygga en datamängd, processera datamängden, träna algoritmen, testa och förbättra modellen [2].

2.1.1 Samla data

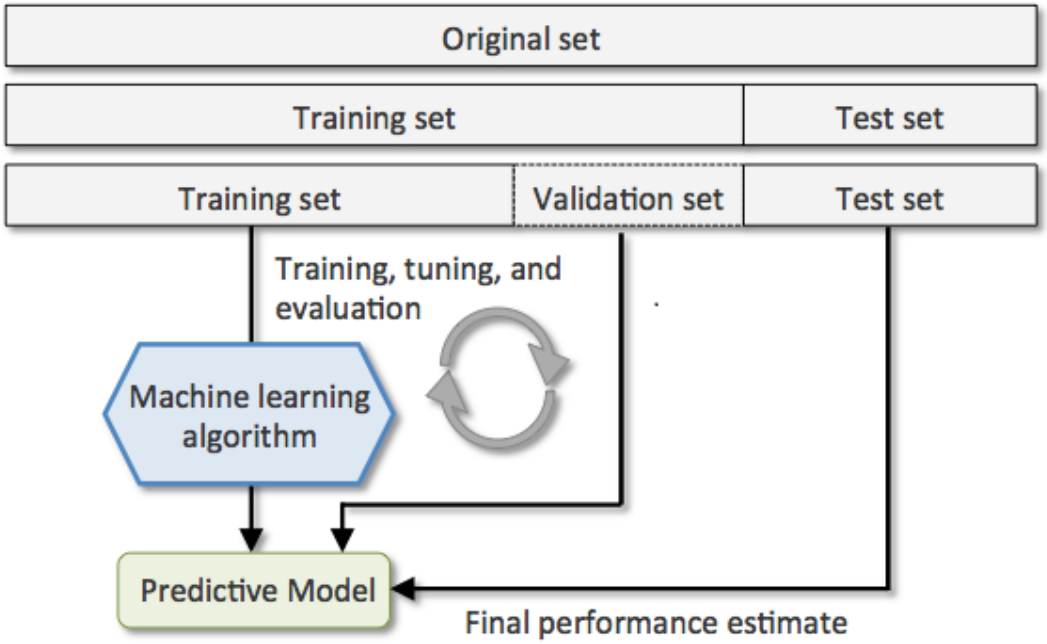
Första steget är att ha tillgång till en datamängd, som är en samling data som används för att träna upp maskinlärningsalgoritmen. Datamängdens innehåll bestämmer modellens riktning, exempelvis då vi har en mängd med engelska meningar, är man begränsad till en modell som känner igen engelsk text. ML är starkt beroende på kvalitén och mängden av tillgängligt data, beroende på fallet kan man själv bygga upp en datamängd genom syntetiskt manipulera varje insättning, genom att exempelvis minska resolutionen på varje bild från en offentlig databas för att simulera kompression. Risken i dessa fall är att simuleringen inte liknar ett riktigt fall och då misslyckas modellen med riktiga fall där kompression har skett. ML applikationer och speciellt neurala nätverk, hänger sig lätt på partiskheter i datalagret, de vill säga att datamängden saknar variation. [7]

2.1.2 Behandling av data

Efter att ha samlat ihop en stor och mångfaldig datamängd, måste den råa informationen slipas för att vara begriplig åt ML algoritmen med till exempel annotera datat [7]. Denna process har en stor inverkan på den slutliga noggrannheten. För oövervakad inlärning hoppar man över denna steg, man matar in den råa datan rakt i algoritmen och sedan skall maskinen själv kunna klassificera varje instans, till exempel om djuret på inmatade bilden är en katt.

2.1.2.1 Funktionsteknik

I förprocessen skall man också identifiera de viktigaste särdragen (engl. Features), alltså de drag som påverkar ML algoritmens prestanda mest. Då undanröjer man redundanta särdragen från algoritmen som minskar antalet inmatningar, vilket påverkar positivt på körningstiden, noggrannheten och mängden minne som krävs för körningen. [9] Funktionsteknik varierar beroende på algoritmen och datatypen man använder, i en matristabell är särdragen kolumnrubrikerna, medan i bilddatalager är det kanterna och linjerna av objekt i bilden som modellen känner till [10].



Figur 2. Hur datamängden splittras och används i ML [7].

2.1.2.2 Tränings- och testmängd

I sista stegen av förprocessen delas det färdiga datamängden in i tre set: Tränings-, valideringsoch testmängd, som vanligtvis delas i förhållandet 60%/20%/20%, dock detta varierar beroende på projektet, några föredrar att inte alls ha en valideringsmängd. Generellt skall man före delningen röra om listan så alla set är ungefär lika slumpmässiga. [7]

Träningsmängden används för att träna ML algoritmen enligt de parametrar man ställt. [12]

Med valideringsmängden mäter man prestandan av den tränande modellen och finjusterar parametrarna, detta repeteras tills man är nöjd med resultatet [12]. Notera att valideringsoch testmängden är totalt okänd åt algoritmen, om den inte var det skulle resultatet vara opålitlig då algoritmen redan vet resultatet. Testmängden ger algoritmen sista utvärderingen av prestanda. [12]

2.1.3 Träna algoritmen

Skapa algoritmens modell med de parametrar du valt och påbörja träningen. Träningsprocessen är iterativ, den kör variabler genom algoritmen och jämför dess utdata med de resultat som den borde ha producerat, sedan ändrar den vikter och partiskheter i algoritmen och kör de variablerna igen ända tills majoriteten av resultaten är korrekta. Nu har algoritmen blivit en ML modell. [2]

2.1.4 Testa och förbättra modellen

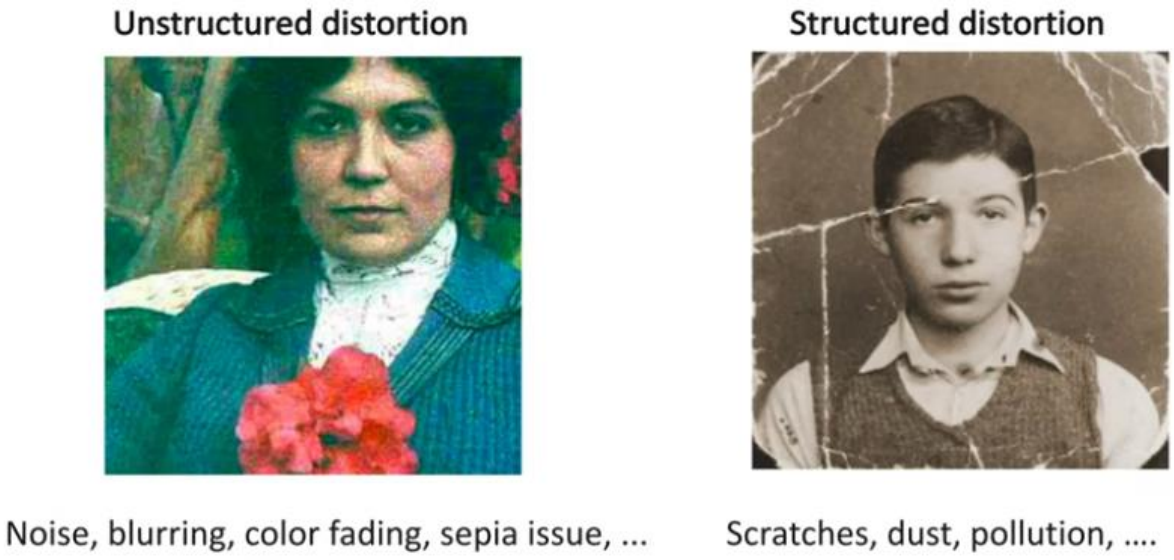
I sista steget används den okända test- eller valideringsmängden för att få den sista utvärderingen av modellen. Om prestandan är inte nöjaktig kan det vara svårt att upptäcka om problemet är med datamängden, modellens uppbyggnad och parametrar, eller typ av modell. Ofta är det bara att pröva sig fram [23]. I optimala fall kommer modellen prestera bättre då den konsumerar mera information. [2]

3. Restaurering av foton med maskininlärning

3.1 Restaurering av gamla foton av skador

Gamla foton är ett vanligt offer för slitning då de förvaras i dåliga omständigheter. Numera kan vi förvara bildens nuvarande kvalité genom att digitalisera fotot med skanner eller telefonkamera, den digitala versionen kan då förbättras antingen med manuellt arbete eller en algoritm.

Degradering av bild kan generellt delas in i två grenar: ostrukturerad degradation såsom suddighet, färgblekning och låg resolution, samt strukturerad degradation såsom hål, repor och fläckar (Fig. 3). Gamla foton brukar ha en kombination av båda vilket ställer en stor utmaning åt datavetare då det är praktiskt taget omöjligt att komma åt en datamängd som innehåller tillräckligt av alla olika degrationskombinationer. Dessutom är det svårt att syntetiskt degradera skarpa bilder för att motsvara ett autentisk utslitet foto, fenomenet kallas för domän gap, alltså då skillnaden i innehållet mellan autentiska och syntetiska datalagren är stor och påverkar modellen märkvärdigt. [13]

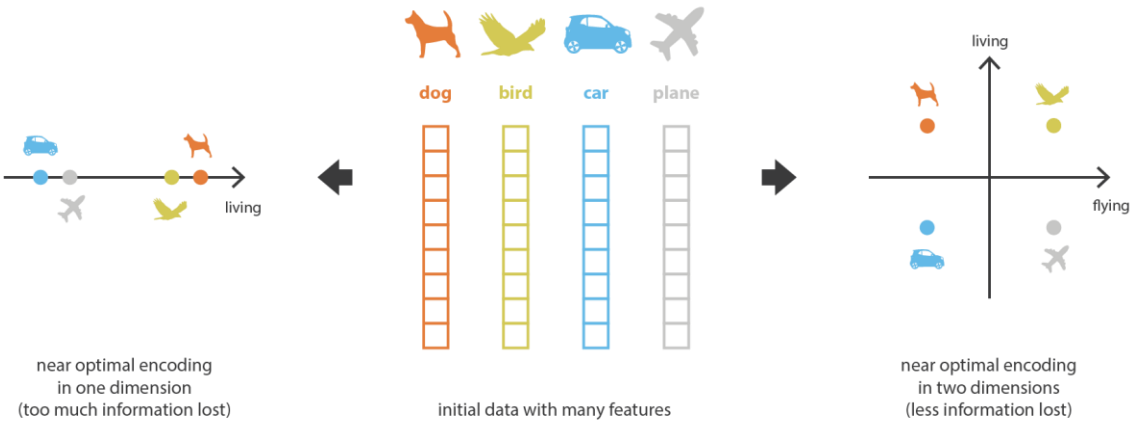


Figur 3. Exempel på skada i gamla bilder [13].

Datavetarna Ziyu Wan, Bo Zhang, Dongdong Chen, Pan Zhang, Dong Chen, Jing Liao och Fang Wen tillsammans med Microsoft presentera en Deep Latent Space Translation (DLST) algoritm för restaurering av bild med blandade former av skador. Algoritmen använder metoder såsom djup maskininlärning, latent utrymme och variationella autoomkodare (VAE). Deras lösning har en ytterligare modell som hanterar bara ansiktet då de vill undvika ändra ansiktsdrag på personerna.

3.1.1 Dimensionell förminskning, latent utrymme och variationella autoomkodare

Dimensionell förminskning används för att sänka antalet särdrag (engl. Features) som beskriver indatat, oftast används detta för krävande beräkningar, datalagring eller datavisualisering där trivialare former av data föredras [15]. I figur 4 sänks dimensionaliteten från sju till två, då kan data visualiseras i en tvådimensionell matris, däremot om dimensionaliteten sänks för mycket försvinner särdragen på de olika inmatningarna som leder till att det är svårare att skilja på indatat.



Figur 4. Dimensionell förminskning till ett- och tvådimensionellt plan [15].

Enligt Wikipedia är latent utrymme definierat som en uppsättning artiklar i en mångfald var de artiklar som liknar varandra är placerade närmare varandra i latenta utrymmet [6], med andra ord är det en representation av komprimerad data som blivit dimensionellt förminskad [14]. Till exempel en stol i latent utrymme behåller endast de viktigaste komponenterna som gör en stol, såsom fyra ben och ryggstöd. Konsekventuellt förlorar man möjligtvis annan information från processen, såsom färgen på stolen. [14]. Gällande bilder och restaurering kan en ML modell använda sig av latent utrymme för att känna igen ett träd för att processerna den annorlunda från resten av bilden, I DLST lösningen tas ansikten för vidare prosessering.

Sedan skall modellen återuppbygga det omkodade datat som blivit dimensionellt förminskat (Fig. 5)

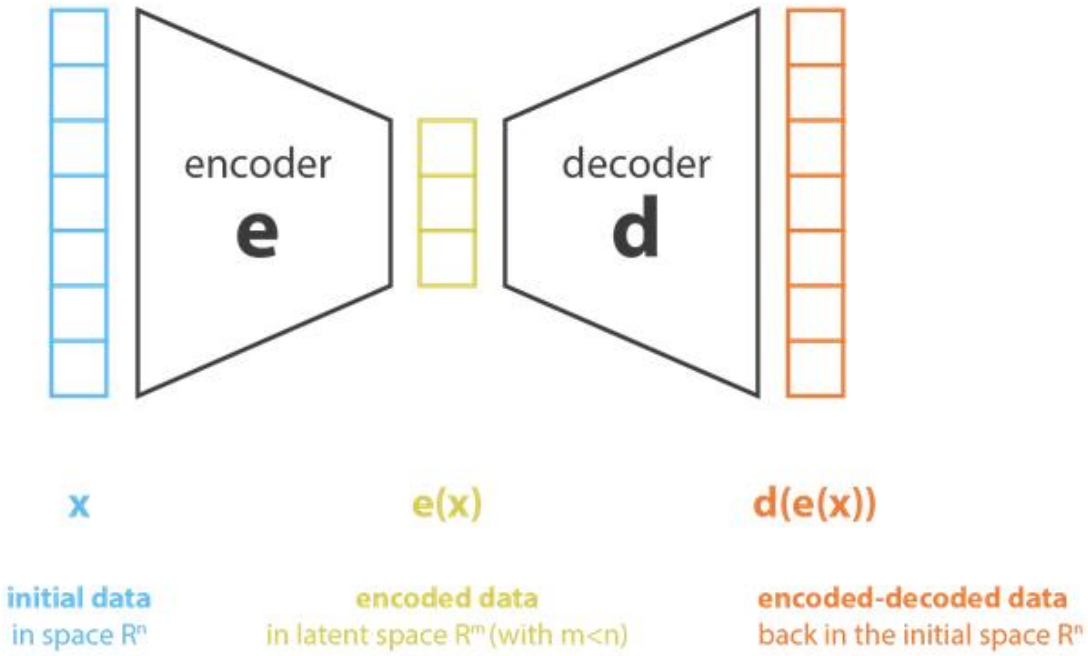
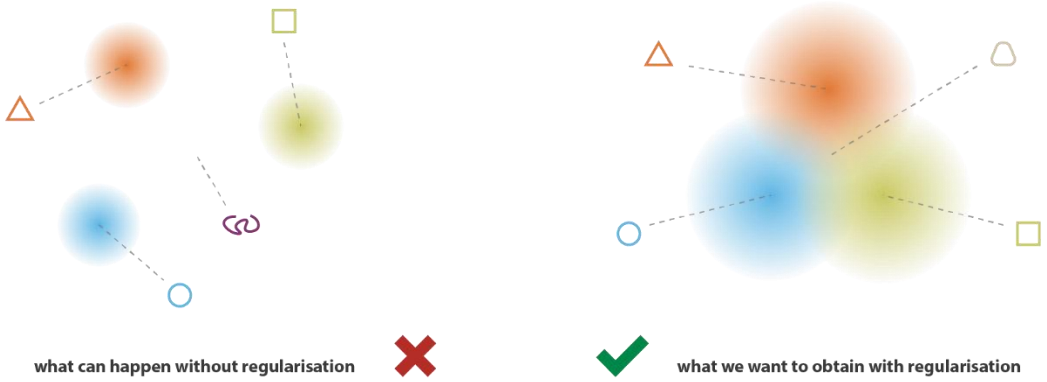


Fig 5. Illustration på en VAE [15]

För att kompressa och rekonstruera datamängden behövs både en omkodare och avkodare (engl. encoder och decoder) generellt är målet att hitta ett sådant par var förlusten av information vid rekonstruktionen är minimal. Informationsförlustmängden kan mätas med en principalkomponentanalys. Neurala nätverk kan användas till om- och avkodning, vilket kallas för autoomkodning, var modellen lär sig iterativt den bästa om- och avkodningsprocessen genom att jämföra in- och utdata och justera vid behov. [15]

3.1.2 Autoavkodning för innehållsgenerering

En väl konfigurerad autoomkodare kan skapa nytt unikt innehåll med det data man redan matat in. Målet är att kunna välja två punkter i latenta utrymmet för avkodning så att de har liknelser i sammanhanget. Generellt vill man använda Variationella autoomkodare (engl förkort. VAE) eftersom de undviker överanpassning (engl. Overfitting) genom att distribuera omkodningen i latenta utrymmet och göra den regelbunden. Regelbundenhet i detta sammanhang innebär att ordningen av varje punkt i latenta utrymmet följer ett mönster [15]. I figur 6 har vi två autoomkodare som båda blivit inmatade trianglar, fyrkanter och cirklar, men på grund av regulariseringsskillnader kan bara den högra generera användbart innehåll. Figur 7 demonstrerar hur innehållsgenerering kan se ut i praktiken, var modellen omvandlar ett ansikte till ett annat.



Figur 6. Exempel på icke reguljär och reguljärt latent utrymme [15]

Sammanfattat tar omkodaren av en VAE en bild från lagret och dimensionellt förminskar den till latenta utrymme, sedan skall avkodaren ta representationen av den inmatade bilden från latenta utrymme och reproducera den originella bilden. Ett färdigt tränat nätverk kan producera representationen av en variation av bilder samt producera nya med att interpolera mellan de redan inmatade.

Figur 7. Interpolation i latenta utrymme [5].

3.1.3 Uppbyggning av DLST datamängd och modell

Till denna modell används Pascal VOC datamängd. För syntetiskt ostrukturerad degradation av bilder användes följande metoder:

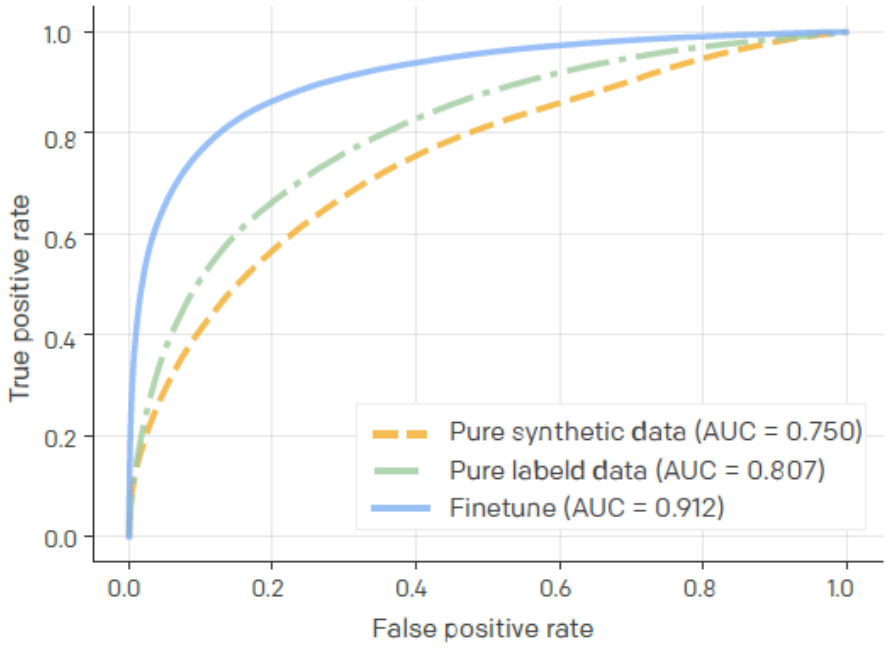
1) Gaussisk vitt brus
2) Gaussisk oskärpa

3) JPEG kompression
4) Färgstörningar med varierande kvalité nivåer
5) Box oskärpa för att simulera kameralinsens inriktning.
6) Filmkornighet. [13]

Varje typ tillämpas i varierande grader och slumpmässig ordning.

För att simulera strukturell degradation används först 62 texturbilder med repor samt 55 bilder med pappertextur, variation på texturerna skapas med elastisk förvrängning. Sedan sätts texturbilderna med repor i slumpmässiga lager på de hela bilderna i Pascals datamängd. Allvarligare skada simuleras med slumpmässig hålgeneration vilket avslöjar papper texturen som är lagrad under bilden. [13]

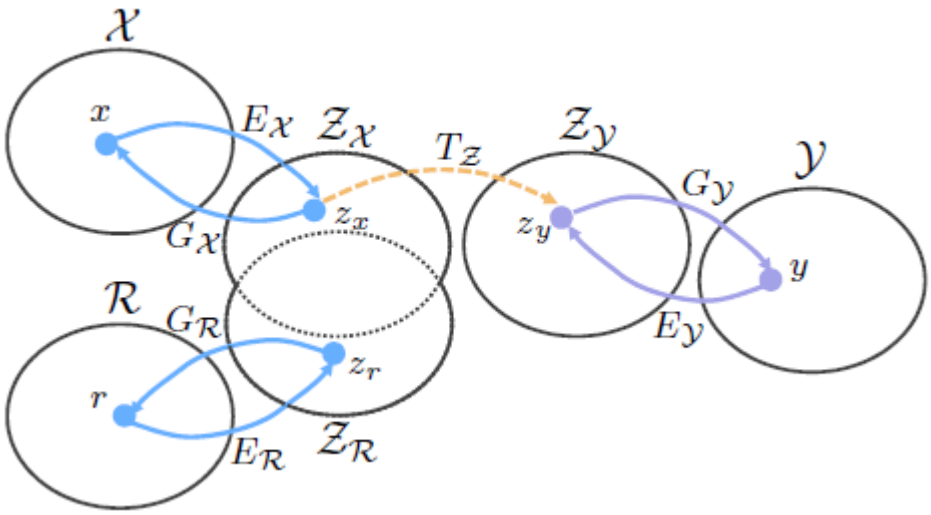
För att vidare förbättra data annoteras skadorna på 783 autentiska gamla foton var 400 används för träningsskedet och resten för testning. Detta har en signifikant inverkan på det slutliga resultatet vilket ROC kurvan demonstrerar i figur 8 [13]. Det skilda nätverket för ansiktsförbättring tränades med en samling av 50,000 bilder med hög resolution [13]



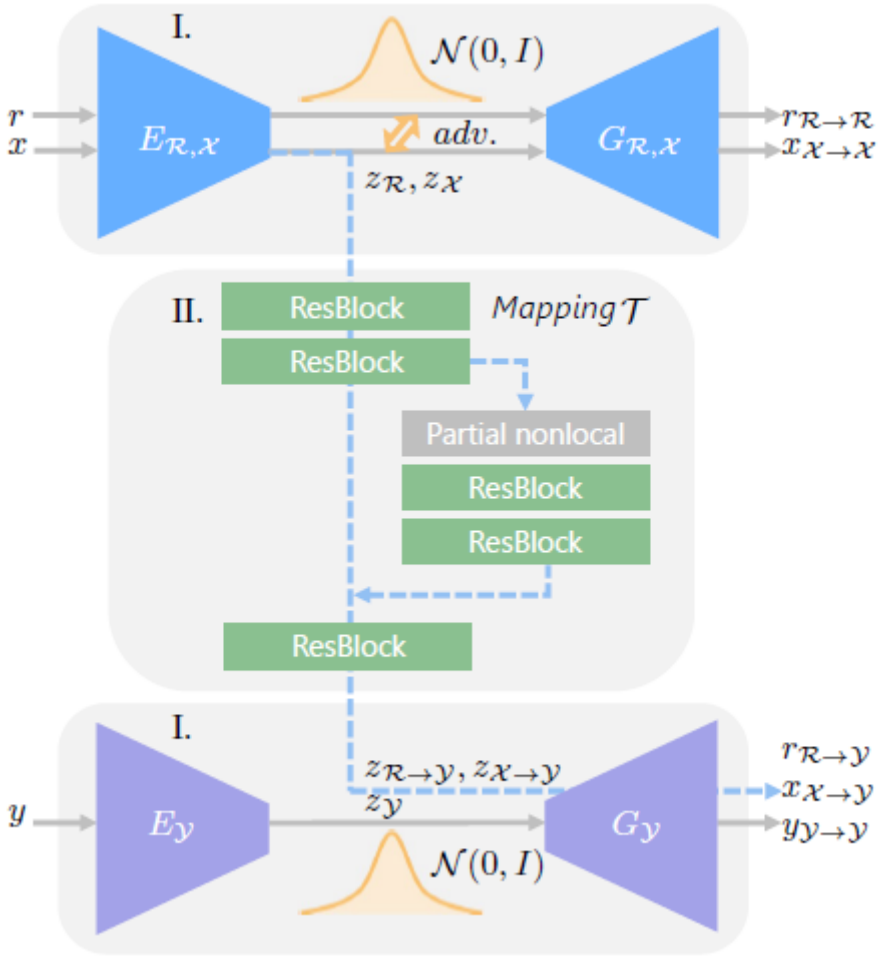
Figur 8. ROC kurva som demonstrerar skillnaden mellan olika databehandlingsmetoder [13].

Huvudmålet med DLST projektet är förbättra restorationen på bilder med blandad skada genom att minska domängapet mellan syntetiska bilder och riktiga bilder. Deras lösning är att använda sig av ett triplettdomänöversättningsnätverk med domänerna: R för autentisk gamla bilder, Y

för rena bilder och X för syntetiskt degraderade bilder av domän Y (fig. 9). Två stycken VAE tränas för att kombinera domänerna. [13]



Figur 9. Illustration på domänerna i DLST modellen [13].



Figur 10. Illustration på DLST-modellen [13]

De tränar VAE1 med att omkoda innehållet av domänerna X och R till ett gemensamt kompakt latent utrymme, medan VAE2 anskaffar den latenta koden av de rena bilderna. Mellan autoavkodarna i latenta utrymmet finns kartläggaren T som består av en partiellt icke-lokalt block med två grenar: en global gren för borttagning av strukturerade skada samt en gren för ostrukturerade defekter. Grenarna är slutna samman i latenta utrymmet vilket sedan förbättrar restorationsförmågan på bilder med mångsidig skada. Med hjälp av de syntetiska bild paren (x och y) lär T modellen hur transformeringen mellan rena och degraderade bilderna kartläggs. För att x och r delar samma latenta utrymme är transformeringen med riktiga gamla bilder också möjlig.[13]

Nätverket ovan förenas med ansiktsförbättringsnätverket, ansiktet går först genom trippeldomännätverket och sedan rekonstruerar ansiktsnätverket indatat till hög resolution, till sist smälts ansiktet samman med resten av den redan processerade bilden (Fig. 15) [13].

I pappret nämns att modellen presterar dåligt med bilder vars belysnivåer är obalanserade på grund av brist på fall i datamängden. [13]

3.2 Färgrestoration

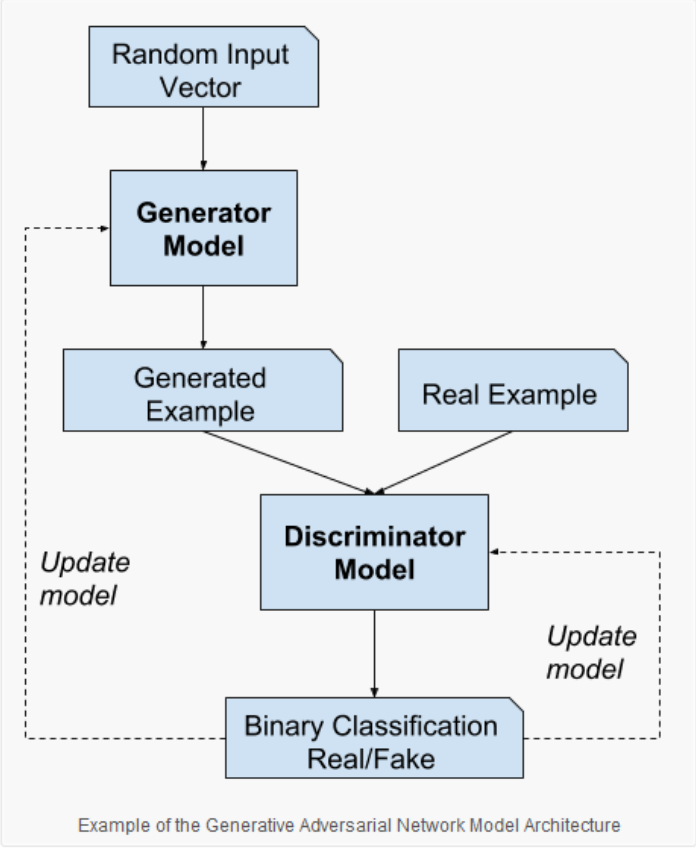
Efter bilden är reparerad från allvarligare skador är målet att återställa de ursprungliga färgerna. Till detta använder vi DeOldify som är tillverkad och uppdaterad ensam av Jason Antic. Deoldify är en djup maskininlärningsmodell som använder sig av en färdig inlärd U-net, som är en typ av faltningsnätverk utveklad för bildsegmentering [16], samt en modifierad Generative Adversial Network (förkort. GAN) som han kallar för NoGAN. Antic påstår att NoGAN får inlärningen från GAN utan den långa träningstiden, var en normal GAN tar flera dagar träning klarar den nya metoden av samma på några timmar med ett hyggligt grafikkort, han använde själv en GTX 1080 Ti. En förtränad U-Net sparar också några dagar av träning då det enda som krävs är finjusteringar för att passa uppgiften.

Till skillnad av restorationsmodellen behöver Antiq själv inte processerna sin data, han tar bilderna rakt från Datalagret ImageNet [17], vilket är en samling av 14 miljoner manuellt annoterade bilder och är ofta använd inom objektdetektion och bildklassificering [8].

3.2.1 GAN

Generative Adversial Network är en maskininlärningsmetod som presenterades år 2014 vars mål är att generera innehåll som är omöjlig att särskilja från autentiskt material, såsom bilder och tal. Ursprungligen var GAN introducerad för oövervakad inlärning, men numera används den i flera former av maskininlärning. [19] GAN delas in i två neurala nätverk som undermodeller: genereringsmodellen som lär sig skapa nya exemplar med hjälp av ett träningsdatalager, samt urskiljandemodellen (engl. Discriminative model), som försöker klassa exemplaren som genererade av den första modellen eller riktiga exemplar tagna rakt från domänen. [18] Om genereraren lyckas lura urskiljaren skall urskiljaren ändra dess insättningar men om urskiljaren gissar rätt måste genereraren uppdatera inställningarna. Målet är att få en 50 % korrekt gissningsrat åt båda undermodellerna [18]

Både VAE och GAN används för innehållsgenerering och därför brukar de jämföras i dessa sammanhang. Generellt generar GANs bättre material men de sägs vara svåra att arbeta med, dessutom är körningstiden betydligt längre.



Figur 11. Demonstration på hur interaktionen mellan generator- och urskiljandemodellen sker i en GAN [18]

3.2.2 NoGAN träning

Antic har skapat tre olika modeller: Artistisk, var färgerna är vibranta men ansikten skapar problem, stabil, som passar bäst för porträtt och landskapsfoton, och video, för att skapa färg i videosnuttar. Alla tre har en likartade träningsprocesser dock längden varierar beroende på antalet gånger man repeterar processen.

I första skedet av NoGAN förtränas generatorn för sig själv med perceptuell förlust, vilket är en funktion som jämför bildpar genom att summera alla kvadratiska fel mellan pixlarna och ta medelvärdet av dem. Bildparet hör till två olika kategorier, i detta fall färglösa och färgade. Modellen lär sig att identifiera skillnaden mellan bildparen och klarar av att omvandla en totalt ny bild från kategori ett till två. [21] Denna steg tar majoriteten av den totala träningstiden och klarar redan av att producera hyfsade färgläggningsresultat, men Antic nämner att modellen har en tendens att fuska, att en del av färgningen är en massa blått, grönt och rött. De genererade bilderna sparas och används i nästa steg. [20]

I andra steget förtränas urskiljandemodellen som en binär klassificerare (engl. Binary classifier) vilket grupperar indatat i bara två klasser. I klassificeraren matas autentiska färggranna bilder samt de generade bilderna från första steget, sedan försöker den gissa om bilden är genererad eller ej. Orsaken att man föredrar detta över en vanlig GAN-process är att det är lättare och snabbare att modifiera klassificeraren till sina behov och generatorn skapar redan rimliga resultat efter första stegets träning. Standard GAN träning kan ta flera dagar. [20]

Till sist tränas generatorn och diskrimineraren i en vanlig GAN miljö, dock man vill inte att modellen skall köra igenom hela processen eftersom efter en viss punkt som ligger mellan 1 och 3 procent av det körda ImageNet datalagret, börjar det uppstå oönskade störningar i färgläggningen. I figur 12 märks att ansiktet påverkas mest av detta: läpparna och hyn blir orange, skuggningen blir slarvigt och detaljerna försvinner. På grund av detta är den egentliga GAN träningstiden av en cykel 30–60 minuter. [20] För att hitta böjningspunkten spara Antiq bilder varje 0,1 % av processerat datalager och granska resultaten manuellt. Videomodellen kör alla steg bara en gång medan stabila och artistiska versionen repeterar de två senaste stegen, tre, respektive fem gånger. Enlig Antic sker det ingen märkbar förbättring med mera repetitioner.[17]

Figur 12. Resultat på olika skeden av Gan träning [17].

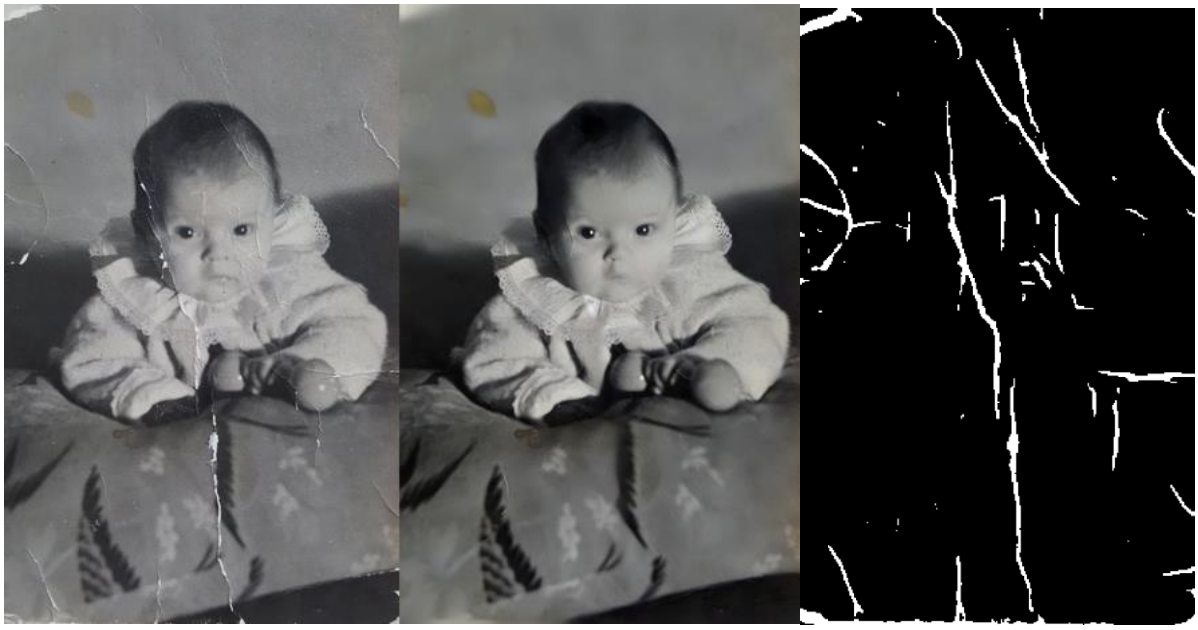
4. Resultat av restoration

För att testa och demonstrera restorationsmetoderna matar vi fem kamerafotograferade bilder från personligt fotoalbum i båda maskininlärningsmodellerna och jämför resultaten med de originella. Bilderna varierar i skador, miljö och ljussättning. Deep Latent Space Translation nätverket har öppen källkod och finns i källförteckningen vid nummer 22, restorationen tar bara några minuter då den färdiga modellen importeras från Github och körs på Google Collab. Collab har dock begränsningar på storleken av filer som kan uppladdas, bilder med hög resolution måste köras på egen dator.

4.1 Skaderestorationsresultat

Figur 13 Restoration av mild strukturerad skada

Fig. 13 är aningen suddig och har vad som ser ut som ett vikmärke i en halvcirkel på högra sidan av bilden, dessutom syns vita prickar runt omkring. Efter behandling är skråman vid vikmärket borta men ljusranden vid cirkeln syns ännu, resolutionen är dock drastiskt förbättrad.



Figur 14. Restoration av svår strukturerad skada

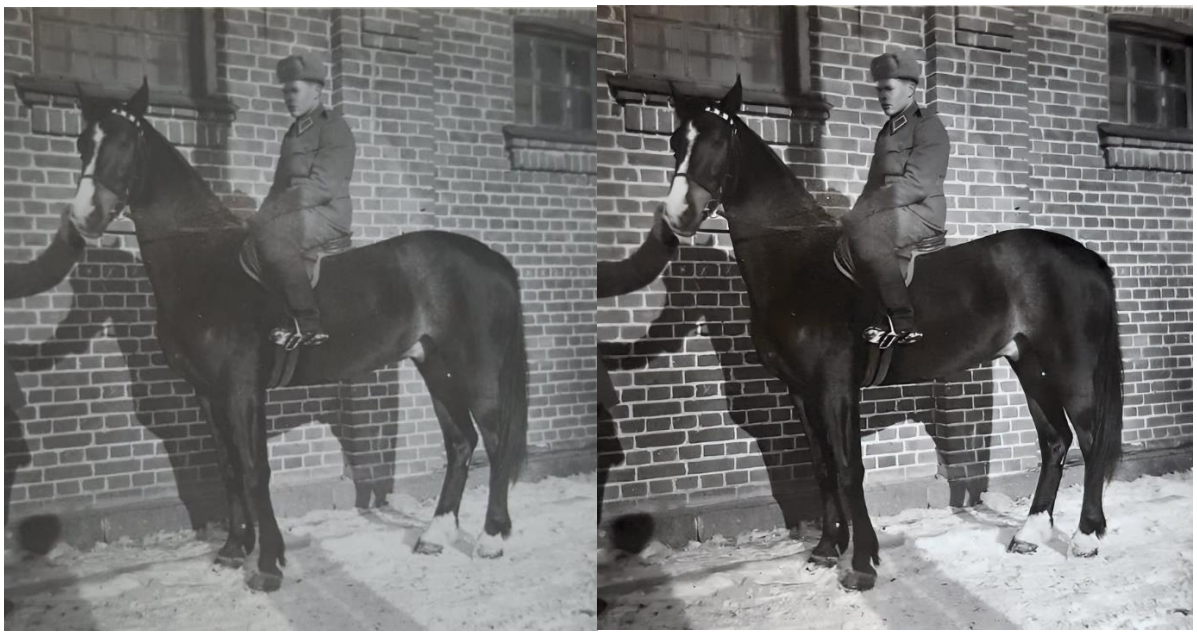
Figur 14 är den enda fallet med stor strukturell skada. Modellen har lämnat skada på babyn, samt tagit bort en del av munnen. Hyn och håret har också missat sin textur och blivit utjämnat. På Github kan man också se skråmorna som modellen upptäckt på den uppladdade bilden.

Figur 15. Svår ostrukturerad skada och exempel på ansiktsrestoration

Bildens (Fig. 15) nedre del är totalt utsliten och bortom reparation. Inzoomat ser man ansiktsförstärkarens resultat vilket har flyttat på ögonens riktning och gett personen ett tomt blick. Bortsett detta har modellen lyckats hålla ansiktsdragen. Sista bilden är vad ansiktsmodellen producerar före maskeras på den restorerade helheten.

Figur 16. Restoration av redan färglagd bild

Gamla färgbilder får ofta en gul färgton då färgmedlet börjar blekna, modellen lyckas få de originella färgerna klarare fram (Fig. 16).



Figur 17. Restoration av ostrukturerad skada

Modellen lyckas bra med att minska ostrukturerad degradation och allmän suddighet.

4.1.1 Resultat

Resultaten är betydligt bättre än originella bilden, det kommer dock med några kostnader, såsom att ansiktsdragen på personerna kan ändras. Vi tror att majoriteten av de uppkomna problemen skulle försvinna med en bättre digitaliserad originalbild var resolutionen skulle

vara högre, en skanner skulle möjligtvis kunna lösa problemet. Skaparna av modellen nämner att komplex ljussättning orsakar problem.

4.2 Färgrestoration

De skaderestorerade bilderna matas sedan in i DeOldify som hittas också på Github [17] För det här syftet använder vi standardmodellen, alltså den artistiska modellen.

Figur 18. Färgläggningsresultat

4.2.1 Färgrestorations resultat

Modellen lyckas behålla den starka belysningen realistisk på motorcykelbilden. Det ser ut som att olika delar av cykeln reflekterar olika starka beroende på materialet, vilket möjligtvis indikerar att modellen förstår sig på olika material. Själva motorcykeln har fått ett rostigt eller kopparfärgat färgtema, inklusive framlyktan, vilket är troligtvis inte den ursprungliga färgen. Babyns hy har fått har blivit gråaktig, den är dock den enda bilden som fått förvrängningar. Modellen har tonat ner den redan färgade bilden, färgrika klädesplagg samt taket har blivit totalt grått.

5. Sammanfattning

Ofta är foton det enda som vi har kvar av den förflutna tiden, med maskinlärning har man möjlighet att återuppliva dem till sin tidigare ära utan att betala åt en professionell fotorestaurerare. Både skaderestoration- och färgrestorationsmodellen klarar av att producera hyfsade resultat. Deoldify lyckas färglägga bilderna utan att skapa mera visuella störningar då den inmatade bilden är hel, likaväl bevarar den avancerad ljussättning såsom reflektioner och skuggningar, hudfärgen verkar dock vara ett problem. Skaderestorationsmodellen tar effektivt bort mildare repor samt allmän suddighet och brus, restoration av starka skador däremot kan orsaka oönskade sidoeffekter såsom ändrade ansiktsdrag. Resultaten skulle lätt kunna

förbättras om bilderna vore digitaliserade med en skanner samt om modellerna körs på egen maskin då det finns resolutionsbegränsningar på båda modellers online versioner.

6. Källförteckning

[1] Rabab Farhan Abbas
Review on some methods used in image restoration

[2] Machine Learning
https://www.ibm.com/cloud/learn/machine-learning
Hämtad 24.3.2021

[3] Top 10 real-life examples of Machine Learning
https://bigdata-madesimple.com/top-10-real-life-examples-of-machine-learning/
Hämtad 24.3.2021

[4] Wiener filter
https://en.wikipedia.org/wiki/Wiener_filter
Hämtad 25.10.2021

[5] Tim Sainburg, Marvin Thielk, Brad Theilman, Benjamin Migliori, Timothy Gentner
Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions

[6] Latent Space
https://en.wikipedia.org/wiki/Latent_space
Hämtad 2.3.2022

[7] How to Build A Data Set For Your Machine Learning Project
https://towardsdatascience.com/how-to-build-a-data-set-for-your-machine-learning-project-5b3b871881ac
Hämtad 24.3.2021

[8] ImageNet
https://paperswithcode.com/dataset/imagenet
Hämtad 25.10.2021

[9] How to Choose a Feature Selection Method For Machine Learning
https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/
Hämtad 26.3.2021

[10] Feature (machine learning)
https://en.wikipedia.org/wiki/Feature_(machine_learning)
Hämtad 26.3.2021

[11] https://en.wikipedia.org/wiki/Computer_vision
https://en.wikipedia.org/wiki/Computer_vision
Hämtad 27.3.2021

[12] Training, validation, and test sets
https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets
Hämtad 27.3.2021

[13] Ziyu Wan, Bo Zhang, Dongdong Chen, Pan Zhang, Dong Chen, Jing Liao, Fang Wen
Old Photo Restoration via Deep Latent Space Translation

[14] Understanding Latent Space in Machine Learning - Ekin Tiu
https://towardsdatascience.com/-latent-space-in-machine-learning-de5a7c687d8d
Hämtad 10.10.2021

[15] Understanding Variational Autoencoders (VAEs) - Joseph Rocca
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73
Hämtad 11.10.2021

[16] U-Net
https://en.wikipedia.org/wiki/U-Net
Hämtad 15.10.2021

[17] DeOldify
https://github.com/jantic/DeOldify
Hämtad 15.10.2021

[18] A Gentle Introduction to Generative Adversarial Networks (GANs)
https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/
Hämtad 15.10.2021

[19] Generative adversarial network
https://en.wikipedia.org/wiki/Generative_adversarial_network
Hämtad 15.10.2021

[20] Decrappification, DeOldification, and Super Resolution
https://www.fast.ai/2019/05/03/decrappify/
Hämtad 20.10.2021

[21] Perceptual Loss Functions
https://deepai.org/machine-learning-glossary-and-terms/perceptual-loss-function
Hämtad 20.10.2021

[22] Bringing Old Photos Back to Life
https://colab.research.google.com/github/mmaithani/data-science/blob/main/Bringing_Old_Photo_Back_to_Life.ipynb#scrollTo=69H2guBfrzqu

[23] My ML Model Fails. Why? Is It the data?
https://towardsdatascience.com/my-ml-model-fails-why-is-it-the-data-d8fbfc50c254
Hämtad 18.3.2022