Universitat Oberta de Catalunya

uoc.edu

Implementació d'un procés de transferència d'estil mitjançant una GAN

Ricard Deza Tripiana
Màster Universitari en Ciència de Dades
Àrea 5

Julian Vicens Bennasar
Albert Solé Ribalta

24/06/2020

FITXA DEL TREBALL FINAL

Títol del treball:	*Implementació d'un procés de transferència d'estil mitjançant una GAN*
Nom de l'autor:	*Ricard Deza Tripiana*
Nom del consultor/a:	*Julian Vicens Bennasar*
Nom del PRA:	*Albert Solé Ribalta*
Data de lliurament (mm/aaaa):	*06/2020*
Titulació o programa:	*Màster Universitari en Ciència de Dades*
Àrea del Treball Final:	*Àrea 5*
Idioma del treball:	*Català*
Paraules clau	*Generative adversarial network, deep learning, style transfer*

Resum del Treball (màxim 250 paraules): *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

Aquest treball es basa en l'aplicació de xarxes generatives adversarials (GAN) per realitzar la transferència de l'estil d'un conjunt d'imatges, específiques d'un autor, a una imatge d'entrada. Concretament, es volen aconseguir uns models capaços de generar imatges noves, donada una fotografia real d'entrada, aplicant la transferència d'estil de obres pictòriques de van Gogh, Picasso i Pollock.

Aquest estudi, s'endinsa en les diferents característiques de les imatges tractades per les xarxes i en els components intervinents en el procés de transferència d'estil. Es basa en la configuració i el tractament de les pèrdues descrites en l'article "Artsy–GAN: A style transfer system with improved quality, diversity and performance" de Liu et al. (2016).

En aquest article es proposa un enfocament generatiu adversarial utilitzant pèrdues perceptives, processant imatges amb chroma subsampling, introduint soroll a les imatges d'entrada al generador i una funció objectiu de pèrdua que fomenta generar detalls diferents per a la mateixa imatge de contingut. Amb aquestes modificacions es pretén millorar el rendiment i la qualitat dels resultats obtinguts amb anteriors estudis, com per exemple la utilització de CycleGan's.

Abstract (in English, 250 words or less):

This work is based on the application of generative adversarial networks (GAN) to transfer the style of a set of images, specific to an author, to an input image.

Specifically, we want to achieve models capable of generating new images, given an input real photograph as input, applying the transfer of style of paintings by van Gogh, Picasso and Pollock.

This study delves into the different characteristics of the images processed by the networks and the components involved in the style transfer process. It is based on the configuration and treatment of losses described in the article “Artsy – GAN: A style transfer system with improved quality, diversity and performance” by Liu et al. (2016).

This paper proposes an adversarial generative approach using perceptual loss, processing images with chroma subsampling, introducing noise into generator input images, and a loss target function that encourages generating different details for the same content image. These modifications are intended to improve the performance and quality of the results obtained with previous studies, such as the use of CycleGan’s.

Índex

Llista de figures

1. Introducció

1.1 Context i justificació del Treball

Els resultats d'aquest estudi poden ajudar a la millora de software d'edició d'imatges i vídeos en transferència d'estil. Una de les aplicacions més utilitzades, actualment, de la transferència d'estil, és la transformació de imatges o vídeos en temps real en aplicacions mòbil.

També s'està provocant, actualment, un canvi en la forma de pensar en l'art de forma comercial. Els resultats poden ajudar a investigar nous camins en l'àmbit de les campanyes publicitàries a partir de la generació d'imatges per part de models d'aprenentatge automàtic.

Aquest treball es nodreix d'obres pictòriques per tal de dur a terme la transferència d'estil. Però aquesta transferència es pot extrapolar a molts diversos conjunts d'imatges.

1.2 Objectius del Treball

Els objectius del treball són:

- Estudi i definició de l'arquitectura de la xarxa a utilitzar.
- Decidir la tecnologia utilitzada per la implementació de la solució.
- Recollir dels conjunts de dades d'entrenament pels models a construir. Aquest conjunt de dades els separem en 3 subconjunts:
    - Recull d'obres pictòriques de Picasso i Pollock per l'entrenament dels models.
    - Reutilització dels conjunts de dades d'entrenament del projecte pix2pix per la seva comparació amb els resultats dels nostres models.
    - Recull d'imatges d'entrada a transformar. Preferiblement fotografies reals, ja siguin paisatges o "selfies".
- Investigació amb altres estructures de transferència d'estil.
- Construcció i entrenament dels models definits.
- Anàlisis i interpretació dels resultats obtinguts.
- Conclusions sobre el mètode utilitzat.
- Comparació amb els resultats obtinguts amb el projecte CycleGAN.

1.3 Enfocament i mètode seguit

A partir de la revisió bibliogràfica es definirà l'arquitectura del sistema a construir, com per exemple les funcions de pèrdues que s'utilitzaran o com s'entrenaran les diverses xarxes que la composen.

Per tal de generar el conjunt d'imatges d'entrenament, es realitzarà cerques en bases de dades obertes de museus, entitats artístiques o altres estudis previs.

La construcció i entrenament dels models es realitzarà en el llenguatge Python basant-se amb els frameworks de Deep learning PyTorch i Keras.

Els resultats obtinguts s'analitzaran a partir de mètriques de distancia com per exemple "Frechet Inception Distance" (FID). Aquests resultats s'utilitzaran per comparar, amb les mateixes mètriques amb altres experiments anteriors.

1.4 Planificació del Treball

En la Figura 1, es pot observar la planificació del Treball, les tasques i subtasques en que es divideix.

A banda de la proposta inicial del treball, la primera tasca realitzada és la revisió bibliogràfica dels estudis previs, tan de la transferència d'estil, com l'aplicació de les GAN en aquesta disciplina. Durant l'estudi de les referències històriques del tema del treball, i quan ja es comença a adquirir prou coneixement sobre ell, es defineix i es revisa l'estructura objectiu del treball de forma reiterativa.

Seguidament, un cop definida l'estructura a construir, es comença la recol·lecció de conjunts d'imatges per l'entrenament. Per tal de duu a terme aquesta tasca, es recopilen conjunts d'imatges utilitzades en estudis previs, com també es recullen imatges de bases de dades artístiques.

Finalment, es realitza la construcció de l'estructura definida, i el seu posterior entrenament amb les imatges recol·lectades per aquest propòsit. Un cop, obtingut els models precisats, s'avaluaran i es validaran els resultat obtinguts de les transferències d'estil aplicades.

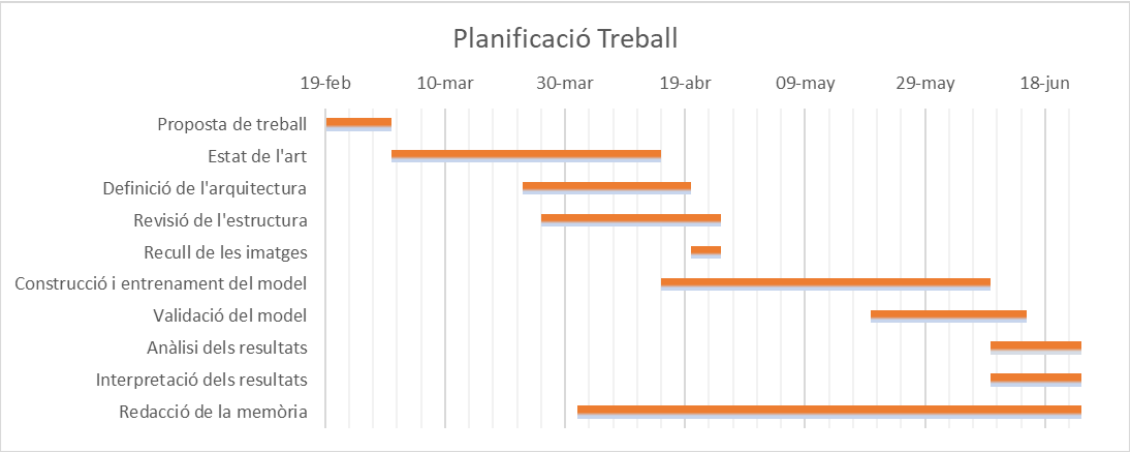


Figura 1: Planificació del Treball final de Màster

1.5 Breu sumari de productes obtinguts

Aquest estudi té com a objectiu obtenir uns models de transferència dels estils pictòrics dels autors Vincent van Gogh, Jackson Pollock i Pablo Picasso, entrenats amb les arquitectures que es descriuran en els següents capítols. El codi generat en aquest treball es troba en el repositori https://github.com/rdezat/style-gan.

També es generaran transferències d'estil, dels mateixos autors, a partir d'una replica del codi del notebook Neural style transfer with Keras. El codi generat en aquest estudi previ es troba en el repositori https://github.com/rdezat/keras-neural-st.

A mesura del possible, s'intentarà construir els mateixos models utilitzant la arquitectura del projecte CycleGAN (Zhu, 2017).

1.6 Breu descripció dels altres capítols de la memòria

La resta de capítols es divideixen en 3 parts:

- Estat de l'art
- Un algorisme neuronal d'estil artístic
- Disseny i construcció de l'arquitectura

En el capítol del "Estat de l'art", es realitza un repàs del treballs anteriors referents a la transferència d'estil i la translació d'imatge a imatge, posant especial èmfasi a la utilització del Deep learning i, sobretot de les GAN's.

En el capítol “Un algorisme neuronal d’estil artístic”, es descriu l’arquitectura d’un model de transferència d’estil sense utilitzar GAN’s, com a estudi previ al l’objectiu del treball. Es mostrarà com es realitza l’entrenament d’aquest model i els resultats obtinguts.

El capítol de “Disseny i construcció de l’arquitectura”, es tracta de la part principal del treball on s’especifica les arquitectures construïdes de transferència d’estil utilitzant GAN’s i basant-se en l’estudi previ de (Liu H. N., 2018). També es descriu com es realitza la avaluació els models obtinguts i es presentaran els resultats i les mètriques aconseguides. Com a mode de comparació, s’introdueixen resultats d’altres projectes, com pot ser CycleGAN (Zhu, 2017).

2. Estat de l'art

2.1 Introducció

Una de les aplicacions més recents i interessants del Deep Learning és la transferència d'estil, és a dir, la capacitat de crear una nova imatge a partir de dues imatges input, una que representa l'estil a transferir i una altra que representa el contingut en el que es vol transferir l'estil (Xie, 2007).

2.2 Primeres aplicacions de la transferència d'estil

Inicialment, els estudis de transferència d'estil, com el de (Xie, 2007), utilitzaven la tècnica de la síntesis de textura. Aquesta tècnica pot generar un gran *patch* de textura sota la norma d'un petit *patch* de textura subjacent sense repetició de patrons obvis. Aquesta similitud perceptiva es pot modelar amb un *camp aleatori de Markov*. Un *camp aleatori de Markov* o *MRF* és similar a una xarxa bayesiana en la seva representació de dependències. La diferència és que les xarxes bayesianes estan dirigides i acícliques, mentre que els *MRF* no són dirigits i poden ser cíclics. Així, un *MRF* pot representar determinades dependències que una xarxa bayesiana no pot (com ara dependències cícliques). En el domini de la intel·ligència artificial, un *MRF* s'utilitza per modelar diverses tasques de nivell baix o mig en processament d'imatges i visió per ordinador. La propietat estadística de la similitud perceptiva d'una textura sintetitzada es pot definir bé a partir del marc *MRF*. Modelant una textura amb *MRF*, es pot il·lustrar la síntesi de textures amb un procés d'inferència bayesiana.

Bàsicament existeixen dos classes de mètodes de síntesis de textura: els mètodes basats en píxels, els quals generen textures píxel a píxel, i els mètodes basats en patchs que generen textures patch a patch. Per incorporar el contingut de la imatge d'origen durant el procés de transferència d'estil artístic, es desenvolupen tècniques de síntesis d'imatges basades en mètodes de síntesis de textura restringida. La majoria dels mètodes existents duen a terme la compensació mitjançant l'ús de paràmetres de control global.

2.3 Inicis del Deep learning

Uns dels primers estudis en introduir l'arquitectura del Deep Learning en els processos de transferència d'estil fou (Gatys, A Neural Algorithm of Artistic Style.; Gatys, Image

Style Transfer Using Convolutional Neural Networks., 2016), en els quals realitzen aquesta transferència combinant el contingut d'una imatge amb l'estil d'una altra minimitzant conjuntament la pèrdua de reconstrucció de característiques i una pèrdua de reconstrucció d'estil utilitzant xarxes neuronals convolucionals (*CNN*). Tal com comenten en el seu estudi (Gatys, A Neural Algorithm of Artistic Style.), els treballs anteriors sobre la separació del contingut de l'estil, es basava en inputs de menor complexitat, com per exemple, caràcters amb diferents lletres o imatges de cares o figures petites en diferents postures. El seu mètode, per un parell específic d'imatges d'origen, es pot ajustar la compensació entre contingut i estil per crear imatges visualment atractives i produeix resultats d'alta qualitat, però té un cost computacional elevat degut a un problema d'optimització.

Un dels treballs que intenten solucionar aquest problema, (Johnson, 2016), indiquen que es poden generar imatges d'alta qualitat optimitzant les funcions de pèrdua de percepció (*Perceptual loss functions*) basades en característiques d'alt nivell extretes de xarxes preentrenades. Proposa l'ús de funcions de pèrdues de percepció per entrenar xarxes de retroalimentació per tasques de transformació d'imatges. Una funció de pèrdua de percepció s'utilitza quan es comparen dues imatges diferents que semblen similars, però es canvien per un píxel. La funció s'utilitza per comparar diferències d'alt nivell, com ara discrepàncies de contingut i d'estil.

Degut a les limitacions dels mètodes anteriors, apareixen nous treballs que busquen mètodes que incorporin menys complexitat. (Cui, 2018), per exemple, proposen mètodes basats en les relacions, o no, entre els mapes de característiques (*Feature map*) i les capes de la *CNN*, reformulant la matriu de Gram, utilitzada en anteriors estudis, en una matriu de covariàncies. Un mapa de característiques està format per diferents unitats d'una *CNN* que comparteixen els mateixos pesos i biaixos. La matriu de Gram és la matriu que defineix el producte escalar, les entrades del qual venen donades per $G_{ij} = \left(v_i \middle| v_j\right)$.

2.4 Generative Adversarial Networks (*GAN*)

Un dels mètodes amb més calatge en la traducció d'imatge a imatge va ser introduït per (Goodfellow, 2014) i es va anomenar Generative Adversial Networks (*GAN*). Es tracta d'un procés adversatiu, en el qual s'entrenen dos models de forma simultània. Un model generatiu que capta la distribució de les dades i un model discriminatori que estima la probabilitat que una mostra provingui de les dades de formació del model generatiu. El procediment d'entrenament del model generatiu és maximitzar la probabilitat que el

model discriminatori s'equivoqui. Degut que els models es defineixen amb perceptrons multicapa (*MLP*), tot els sistema es pot entrenar amb backpropagation, per tant elimina la necessitat de cadenes de Markov i de xarxes d'inferència aproximades sense control. Un perceptró multicapa és un tipus de xarxa neuronal artificial de classe directa. Consisteix, almenys, de tres capes de nodes: una capa d'entrada, una capa oculta i una capa de sortida. Excepte pel node d'entrada, cada node empra una funció d'activació no lineal, la qual cosa permet de classificar conjunts de dades que no estan separats linealment. La retropropagació o backpropagation, és un mètode que s'empra per a calcular el gradient que és necessari aplicar als pesos o coeficients dels nodes de la xarxa. Una cadena de Markov, és una sèrie d'esdeveniments, en la qual la probabilitat que passi un esdeveniment depèn de l'esdeveniment immediat anterior. En efecte, les cadenes d'aquest tipus tenen memòria.

Posteriorment, altres treballs [(Denton, 2015); (Radford, 2016); (Isola, 2017); (Liu, 2017); (Kim, 2017); (Yi, 2017)], van abordar el problema de la traducció d'imatge a imatge utilitzant una combinació de pèrdues adversatives.

Un dels més citats, és probablement el treball de (Isola, 2017). En aquest estudi introdueixen les *GAN* condicionals (*cGAN*). Les *cGAN* són una extensió de les *GAN* on les dues xarxes (generador *G* i discriminador *D*) reben un vector addicional d'informació com a entrada. Aquest vector pot contenir informació sobre la classe de l'exemple de formació. D'aquest treball apareix el programari pix2pix àmpliament utilitzat.

En l'estudi de (Denton, 2015) apliquen les cGAN introduint piràmides laplacianes, model anomenat *LAPGAN*. Les piràmides laplacianes són una representació lineal d'imatge invertida que consisteix en un conjunt d'imatges passabanda separades entre si una octava, més un residu de baixa freqüència.

En (Liu, 2017), indiquen que, com existeix un conjunt infinit de distribucions conjuntes que poden arribar a les distribucions marginals donades, no es pot inferir res sobre la distribució conjunta de les distribucions marginals sense suposicions addicionals. Per tant, fan una hipòtesis d'espai latent compartit i proposen un marc de traducció no supervisat basat en *GAN's* acoblades.

Per donar resposta al problema de l'aparellament costós, en l'estudi de (Kim, 2017) proposen un mètode basat en *GAN's* que aprenen a descobrir relacions entre diferents dominis (*DiscoGAN*).

En l'estudi (Yi Z. Z., 2017), es vol esmenar el fet que les xarxes *cGAN* per a la traducció d'imatges a imatges depenent de la complexitat de la tasca, es necessiten milers a milions de parells d'imatges etiquetades per entrenar-les. Inspirats en l'aprenentatge

dual de la traducció del llenguatge natural, desenvolupen un nou mecanisme *GAN* dual, que permet entrenar als traductors d'imatges a partir de dos conjunts d'imatges sense etiquetar de dos dominis.

2.5 CycleGAN

Una de les aplicacions de les *GAN's* més influents en els últims temps, es tracta d'una evolució de les *cGAN* anteriorment descrita en el projecte pix2pix. (Zhu, 2017) apliquen la consistència de cicle (*Cycle consistency*) al nivell de la característica amb la pèrdua de confrontació, per aprendre una traducció d'imatges no emparellades. El model proposat conté dos generadors (*F* i *G*) i dos discriminadors (D_x i D_y), un per cada domini. Introdueixen dues pèrdues de consistència de cicle que pretenen minimitzar les diferències entre les transformacions de domini creuades. En minimitzar les dues pèrdues, les xarxes aprenen com realitzar transformacions creuades sense utilitzar dades d'entrenament per parelles. Aquestes xarxes reben el nom de *CycleGAN's*.

Altres recerques recents [(Lazo, 2020); (Li, 2018); (Perera, 2018); (Wang, 2016)] realitzen petites modificacions en el concepte de les *CycleGAN's* per tal de millorar la qualitat i la resolució de la traducció.

(Li, 2018), introdueix les xarxes adversatives consistents de cicle aplicat (*SCAN*), les quals descomponen una sola traducció en transformacions de múltiples etapes. En el seu estudi, a més, defineixen un nou bloc de fusió adaptatiu per integrar dinàmicament la sortida de l'etapa actual i la de l'etapa anterior, que supera directament les piles en diverses etapes.

En l'estudi (Perera, 2018), introdueixen la traducció no supervisada de múltiples imatges a imatges (*In2I*). Utilitzant un esquema basat en *GAN's*, combinen informació de múltiples modalitats per produir la sortida corresponent del domini desitjat. També introdueixen un nou terme de pèrdua de consistència latent (Latent Consistency Loss) en la funció objectiu. La pèrdua de consistència latent és la representació comuna entre els dominis d'entrada i de destí.

En el treball (Wang, 2016), es proposa una modificació de les *GAN's* anomenada *Style and Structure Generative Adversarial Network (S2-GAN)*. Aquestes xarxes tenen dos components: *Structure-GAN* que genera un mapa normal de superfície i *Style-GAN* que agafa el mapa anterior com entrada i genera la imatge en dues dimensions. A més, utilitzen una pèrdua addicional amb normals de superfície computada a partir d'imatges generades.

3. Un algorisme neuronal d'estil artístic

Com a estudi previ i presa de contacte amb codis de transferència d'estil, es genera una estructura de translació imatge a imatge sense *GAN's*, utilitzant un model preentrenat.

Es construeix una estructura basada en l'article de (Gatys, A Neural Algorithm of Artistic Style.), prenent com a referencia el notebook Neural style transfer with Keras.

Com indica el nom del notebook, aquest codi està implementat Python, utilitzant el framework Keras per fer ús del Deep learning. El codi es troba en el repositori del GitHub https://github.com/rdezat/keras-neural-st.

L'objectiu d'aquest projecte consisteix en generar una imatge amb el mateix contingut que una imatge base, però amb l'estil d'una imatge diferent, és a dir, transferir l'estil d'una imatge a una altra. Això s'aconsegueix mitjançant l'optimització d'una funció de pèrdua que té tres components: pèrdua d'estil, pèrdua de contingut i pèrdua de variació total.

La pèrdua d'estil és la que es manté l'aprenentatge profund. Precisament, consisteix en una suma de distàncies euclidianes (L^2 *distance*) entre les matrius de Gram de les representacions de la imatge de contingut i la imatge d'estil, extretes de diferents capes d'una xarxa convolucional preentrenada amb ImageNet (VGG19). La idea general és capturar informació de color i textura a diferents escales espacials.

La pèrdua de contingut és la distància euclidiana entre les característiques de la imatge de contingut i les característiques de la imatge de combinació, mantenint la imatge generada prou a prop de l'original.

La pèrdua total de variació imposa una continuïtat espacial local entre els píxels de la imatge de la combinació, donant-li coherència visual.

3.1 Entrenament

Per tal de realitzar l'entrenament, s'executa l'arxiu *main.py* passant-li els següents paràmetres d'execució:

- --base-image-path: Ruta a la imatge amb el contingut a transferir
- --style-reference-image-path: Ruta a la imatge amb l'estil a transferir
- --output-path: Ruta de les imatges de sortida

- --n-epochs: Nombre d'èpoques d'entrenament, per defecte 5000
- --total-variation-weight: Pes de la pèrdua de total de variació, per defecte 1e-6
- --style-weight: Pes de la pèrdua d'estil, per defecte 2e-6
- --content-weight: Pes de la pèrdua de contingut, per defecte 2e-8
- --img-nrows: Nombre de files de píxels de la imatge generada, per defecte 400
- --initial-learning-rate: Taxa d'aprenentatge inicial, per defecte 100
- --decay-steps: Èpoques de decaïment de la taxa d'aprenentatge, per defecte 100
- --decay-rate: Decaïment de la taxa d'aprenentatge, per defecte 0.96
- --epoch-save: Nombre d'èpoques per desar les sortides

Abans de començar l'entrenament, es configura l'extractor de característiques preentrenat, es configura l'optimitzador utilitzat i es preprocessen les imatges.

Per tal de configurar l'extractor de característiques, es carrega un model VGG19 preentrenat amb els pesos d'Imagenet i es crea un model, que s'anomenarà *feature_extractor*, amb els inputs i outputs del model preentrenat.

Posteriorment, es configura l'optimitzador. S'utilitza un optimitzador SGD (gradient descendent incremental) i s'aplica una programació de decaïment de la taxa d'aprenentatge amb els paràmetres d'execució inicials *initial_learning_rate*, *decay_steps* i *decay_rate*.

Per últim, abans de començar l'entrenament, les imatges utilitzades, és a dir, la imatge de contingut i la imatge d'estil, es preprocessen. La funció de preprocessament d'imatges realitza diversos passos. En primer lloc, es carreguen i es redimensionen segons el nombre de files de píxels que s'informi en el paràmetre inicial d'execució *img_rows* i el nombre de columnes de píxels calculat mantenint la relació de resolució original. Seguidament, és converteix la imatge en una matriu i s'insereix una dimensió. Utilitzant el mètode de preprocessament de la llibreria VGG19 de Keras, es converteixen les imatges de format RGB a BGR i es centra cada canal de color respecte al conjunt de dades ImageNet. Finalment, la funció retorna el resultat en forma de Tensor.

Les èpoques d'entrenament venen establertes pel paràmetre inicial d'execució *n_epochs*. Per cada època, es calculen les pèrdues i els gradients, els quals s'apliquen al optimitzador. En el cas que l'època d'entrenament sigui un múltiple del paràmetre inicial d'execució *epoch_save*, es grava un log per consola i es desa la imatge de combinació obtinguda.

Abans de calcular les pèrdues, es concatenen les imatges en un Tensor i s'extreuen les característiques amb l'ajuda del model *feature_extractor*.

Per calcular la pèrdua de contingut, s'utilitza la capa de característiques "block5_conv2" del *feature_extractor*. S'extreu les característiques de les imatges de contingut i de combinació, i es calcula la pèrdua de contingut, tal i com s'ha descrit anteriorment, aplicant un pes definit pel paràmetre inicial d'execució *content_weight*.

Per la pèrdua d'estil, s'utilitzen la capes de característiques "block1_conv1", "block2_conv1", "block3_conv1", "block4_conv1" i "block5_conv1" del *feature_extractor*. Per cadascuna d'aquestes capes, s'extreu les característiques de les imatges d'estil i de combinació, i es calcula la pèrdua d'estil aplicant un pes definit pel paràmetre inicial d'execució *style_weight* dividit pel nombre de capes utilitzades. La pèrdua d'estil total és la suma de la pèrdua d'estil de cadascuna de les capes.

Cadascuna de les pèrdues d'estil, es calcula generant la matriu de Gram per les característiques extretes de la imatge d'estil i de combinació i s'aplica la distància euclidiana.

La pèrdua de variació total es calcula amb la següent funció aplicada sobre la imatge de combinació multiplicada pel pes definit pel paràmetre inicial d'execució *total_variation_weight*:

```
# La tercera funció de pèrdua, pèrdua de variació total, dissenyada per mantenir la imatge generada coherent localment
def total_variation_loss(x, img_ncols, img_nrows):
  a = tf.square(
      x[:, :img_nrows - 1, :img_ncols - 1, :] - x[:, 1:, :img_ncols - 1, :])
  b = tf.square(
      x[:, :img_nrows - 1, :img_ncols - 1, :] - x[:, :img_nrows - 1, 1:, :])
  return tf.reduce_sum(tf.pow(a + b, 1.25))
```

Es sumen totes les pèrdues aplicades, i es calcula el gradient amb la imatge de combinació, per tal de poder introduir-lo en el optimitzador.

Com es comenta anteriorment, un cop calculades les pèrdues i el gradient, en el cas que l'època d'entrenament sigui un múltiple del paràmetre inicial d'execució *epoch_save*, es desa la imatge de combinació obtinguda. Per tal de convertir el Tensor corresponent a la imatge de combinació en una imatge vàlida, es realitza un deprocessament. Aquest mètode consisteix en redimensionar la matriu sense canviar

les dades, eliminar el centre zero amb el píxel mitjà, convertir de format BGR a RGB i aplicar el mínim 0 i màxim 255 als valors de la matriu.

3.2 Resultats

Amb aquesta arquitectura s'ha realitzat diversos experiments amb diferents combinacions d'imatges i de pesos. S'explica els resultats segons l'estil a transferir.

3.2.1 Vicent van Gogh

Com s'observa en el notebook de referència, l'exemple allí exposat, es tracta de la transferència d'estil del quadre "*La nit estrellada*" de Vicent van Gogh amb una fotografia d'un paisatge urbà de París:

Figura 2: La nit estrellada de Vicent van Gogh

Figura 3: Paisatge urbà de París

I el resultat de la transferència d'estil en el notebook de referència, és el següent:

Figura 4: Resultat del Notebook de referència

Per tant, la primera prova realitzada ha estat amb els paràmetres per defecte, la mateixa imatge d'estil, però modificant la imatge de contingut:

Figura 5: Fotografia del port de Dinan

El resultat de la transferència d'estil amb aquest paràmetres i imatges és el següent:

Figura 6: Resultat experiment van Gogh sense GAN

Com s'observa, el resultat es força similar a l'exemple del notebook de referència. La translació de l'estil s'aconsegueix amb suficient èxit i amb bona qualitat.

3.2.2 Jackson Pollock

Els següents experiments es realitzaran amb l'estil pictòric de Jackson Pollock.

Experiment 1

El primer experiment s'han realitzat amb els paràmetres per defecte, la imatge de contingut anterior (Figura 4) i com a imatge d'estil, el quadre "*No.5, 1948*" de Jackson Pollock:

Figura 7: No.5, 1948 de Jackson Pollock

El resultat de la transferència d'estil amb aquest paràmetres i imatges és el següent:

Figura 8: Resultat Experiment 1 Pollock sense GAN

Podem observar com hi ha un intent de transferir l'estil de Pollock a la imatge de contingut, però es mantenen els colors i la estructura de la imatge original.

Experiment 2

El segon experiment, per tal d'esmenar les mancances del resultat anterior, es realitzen petits canvis en els paràmetres d'execució.

Pollock era un pintor abstracte, i com s'ha comentat abans, la pèrdua de variació total, manté la coherència visual. Per tant, per dur a terme l'experiment, s'han modificat els següents paràmetres:

- total-variation-weight: 1e-10
- content-weight: 2e-4

El resultat amb aquest paràmetres i les mateixes imatges que l'experiment 1 és el següent:

Figura 9: Resultat Experiment 2 Pollock sense GAN

Es veu com queda més modificat tant el color de la imatge com l'estructura. S'observa en la part inferior (riu) on la transferència d'estil és més clara. Tot i així, sembla que es sobre impressioni l'estil a la imatge de contingut i no generar una imatge amb l'estil desitjat.

3.2.3 Pablo Picasso

Els següents experiments es realitzaran amb l'estil pictòric de Pablo Picasso.

Experiment 1

El primer experiment s'ha realitzat amb els paràmetres per defecte. La imatge d'estil seleccionada es el quadre "*Les senyoretes d'Avinyó*" de Pablo Picasso:

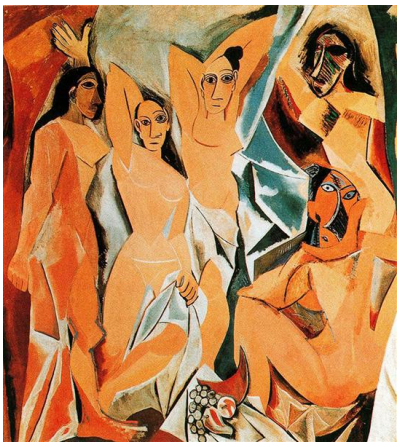


Figura 10: Les senyoretes d'Avinyó de Pablo Picasso

Degut a l'estructura i contingut del quadre de Pablo Picasso, s'ha intentat utilitzar una imatges de contingut el més similar possible:

Figura 11: Grup de persones a l'estiu

El resultat de la transferència d'estil amb aquest paràmetres i imatges és el següent:

Figura 12: Resultat Experiment 1 Picasso sense GAN

Es pot observar com es transfereix molt correctament l'estil cubista del quadre seleccionat als objectes propers a l'objectiu de la càmera i com es distorsionen més exageradament els objectes llunyans. Per aquest motiu es realitza una altre experiment.

Experiment 2

El segon experiment, per tal d'intentar trobar una mica més de coherència als objectes llunyans, s'ha modificat el pes de la pèrdua de variació total:

- total-variation-weight: 1e-5

El resultat amb aquests paràmetres i les mateixes imatges que l'experiment 1 és el següent:

Figura 13: Resultat Experiment 2 Picasso sense GAN

S'observa com es transfereix l'estil cubista, fins i tot amb més intensitat que en el cas anterior però amb molta menys nitidesa en la imatge. Per aquest motiu es realitza una altre experiment.

Experiment 3

El tercer experiment, es realitza per trobar una altre camí per resoldre la coherència als objectes llunyans. En aquest cas s'ha modificat la pèrdua de contingut amb l'objectiu de limitar la seva distorsió:

- content-weight: 2e-6

El resultat amb aquests paràmetres i les mateixes imatges que l'experiment 1 és el següent:

Figura 14: Resultat Experiment 3 Picasso sense GAN

Es pot observar com es transfereix correctament l'estil, però amb menys intensitat als objectes propers. També s'observa una pèrdua de la qualitat notable.

4. Disseny i construcció de l'arquitectura

L'objectiu principal d'aquest treball és replicar, i a mesura del possible, millorar l'arquitectura de transferència d'estil descrit el l'article *Artsy-GAN: A style transfer System with improved quality, diversity and performance* de Liu, H., Michelini, P. N., i Zhu, D. El model descrit en aquest article té l'objectiu d'aprendre un mapeig d'un domini *X* a un domini *Y*, donat un conjunt de dades d'entrenament $\{x_i\}_{i=1}^{N}$ on $x_i \in X$ i un conjunt de dades generades $\{y_i\}_{i=1}^{M}$ on $y_i \in Y$. El model consta de dues xarxes neuronals de convolució, un generador i un discriminador. L'objectiu del generador és aprendre a generar imatges indistingibles de les imatges reals del domini *X*. L'objectiu del discriminador és aprendre a detectar quan una imatge generada pel generador és real o falsa. Per tant, es tracta d'un aprenentatge contraposat on l'objectiu és adquirir l'equilibri entre discriminador i generador. La funció objectiu conté tres tipus de termes: pèrdua adversativa (Goodfellow, 2014), per fer coincidir la distribució d'imatges produïdes i el domini objectiu, la pèrdua perceptiva (Johnson, 2016) per mantenir l'objecte i el contingut en les sortides i la pèrdua de diversitat per millorar la diversitat de les imatges generades. També s'aplica un chroma subsampling en el generador per tal de millorar el rendiment d'execució de l'entrenament dels models.
Aquest sistema es troba esquematitzat en la següent imatge:

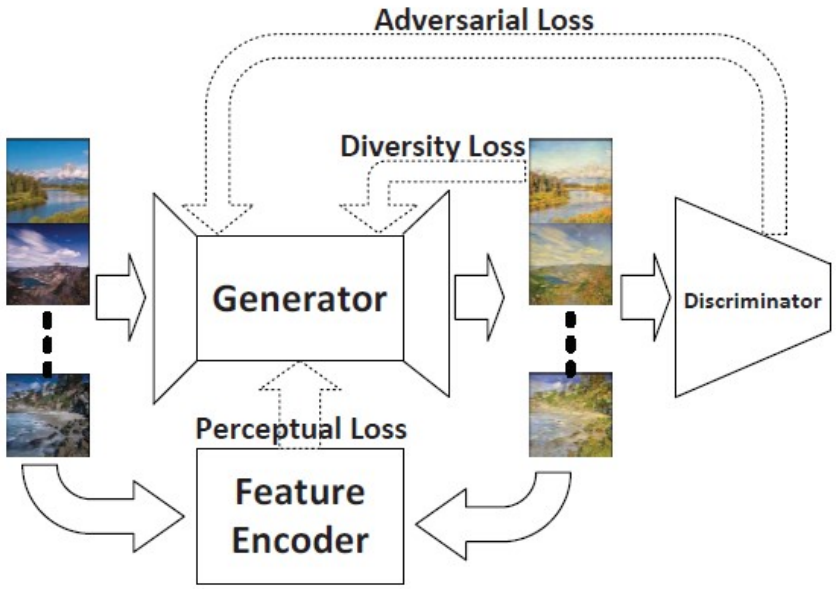


Figura 15: Esquema de l'arquitectura de la GAN implementada en aquest treball

Per tal de dur a terme la construcció s'ha utilitzat com a entorn integrat de desenvolupament *Spyder* sobre la distribució *Anaconda*. El model ha estat implementat amb el *Pytorch* utilitzant *Phyton 3.6*.

4.1 Conjunts d'entrenament

En l'entrenament dels models s'ha utilitzat els següents conjunts d'imatges:

- van Gogh: Les imatges han estat recuperades del projecte CycleGan.
- Picasso: Les imatges han estat extretes mitjançant la API del repositori web WikiArt.
- Pollock: Les imatges han estat extretes mitjançant la API del repositori web WikiArt.

Per tal d'utilitzar l'API de WikiArt, s'ha implementat el codi *datasets.py* ubicat en la carpeta wikiArt.

4.1.1 Recollida d'imatges de wikiArt

El primer pas per poder utilitzar la API de wikiArt és crear-se un compte d'usuari a la url https://www.wikiart.org/. Un cop es tingui usuari de wikiArt, s'accedeix a la url https://www.wikiart.org/es/App/GetApi/GetKeys per tal d'adquirir les claus necessàries per consumir l'API: Api access key i Api secret key. Un cop aconseguides les claus, informarem les claus en les variables *api_access_key* i *api_secret_key* de l'arxiu *dataset.py*.

Seguidament, s'ha de recollir el *id* del Artista de la pàgina de wikiArt. Es podria realitzar mitjançant un codi de web-scraping, o bé manualment de la següent forma:

- Cerquem l'artista objectiu en la pàgina de wikiArt.
- Cliquem botó dret sobre la foto del artista i seleccionem Inspeccionar.
- Recuperem el valor de l'atribut *data-id* immediatament superior al codi html de la imatge.

Desem aquest valor en un json pels paràmetres de cerca, per exemple:

```
searchParam = {'id': '57726d84edc2cb3880b48b01'}
```

Indiquem el tipus de cerca que es vol realitza, en aquest cas:

```
searchType = 'PaintingsByArtist'
```

Amb aquestes variables, *searchType* i *searchParam*, es realitza un get de la url: https://www.wikiart.org/en/api/2/<searchType>?<param1>=<value1>&<param2>=<value2>, on en el nostre cas d'exemple seria: https://www.wikiart.org/en/api/2/PaintingsByArtist?id=57726d84edc2cb3880b48b01

Es recull el json de resposta, i mentre aquest contingui el paràmetre *hasMore*, es torna a crida a la url anterior concatenant el valor del paràmetre. El json de resposta és una llista de llistes de diccionaris amb la informació de cadascuna de les imatges.

Un cop es recull tota la informació retornada pel tipus de cerca "PaintingsByArtist", es recupera els id's i les url's de cadascuna de les imatges que contingui el json de resposta anterior i es crea una llista de diccionaris amb els id's i urls de les imatges.

A continuació, recuperarem els estils que classifiquen cadascuna de les imatges. Per tal efecte, es realitza un nou get a la url anterior modificat el tipus de cerca per "Painting" i el paràmetre id amb el valor de l'id de la imatge recuperat en el json anterior. Aquesta informació la utilitzarem per poder filtrar les imatges recuperades pels estils que ens interessa.

Finalment, un cop es tingui la llista d'imatges final, es desaran les imatges en local de les url's recuperades per cadascuna d'elles.

El codi corresponent a la utilització d'aquesta API es troba desat en el repositori del projecte https://github.com/rdezat/style-gan.

4.2 Entrenament

S'han construït dues arquitectures amb una petita diferència entre si. Una conté un generador amb chroma subsampling i, l'altra sense. L'entrenament, es realitza executant l'arxiu *main_wchr.py* (o *main.py*, amb l'arquitectura amb chroma subsampling) amb el paràmetre principal *train* i els següents subparàmetres:

- --n_epochs: Nombre de èpoques d'entrenament, per defecte 200.
- --epoch: Època en que comença l'entrenament, per defecte 0.
- --batch-size: Mida del lot d'entrenament, per defecte 1.
- --dataset: Ruta de la carpeta que conté el conjunt d'entrenament. Obligatori.
- --dataset_name: Nom del tipus de conjunt d'entrenament. Obligatori.
- --save-model-dir: Ruta a la carpeta on es desarà el model entrenat. Obligatori.
- --image-size: Mida de les imatges d'entrenament, per defecte 256.
- --cuda: Entorn d'execució utilitzat en l'entrenament. S'informa 1 per executar sobre GPU, 0 per CPU. Obligatori.

- --seed: Llavor aleatòria per l'entrenament, per defecte 42.
- --lr: Taxa d'aprenentatge, per defecte 2e-4
- --b1: En l'optimitzador, decaïment del moment de gradient de primer ordre, per defecte 0.5
- --b2: En l'optimitzador, decaïment del moment de gradient de primer ordre, per defecte 0.999
- --decay_epoch: Època en la qual comença el decaïment de la taxa d'aprenentatge, per defecte 100.
- --content-weight: Pes de la pèrdua de contingut, per defecte 1.

En primer lloc, cal anotar que el codi permet la càrrega d'un generador i un discriminador preentrenats amb el mateix codi, per tal de seguir l'entrenament. Això s'aconsegueix informant al subparàmetre *epoch*, el valor de les èpoques en que varen ser entrenats el generador i discriminador que es vol carregar. Per tal que això sigui possible, els models han de tenir el nom amb el format següent:

- generator_epoch_<*epoch*>_wcrh.pth o generator_epoch_<*epoch*>.pth, segons l'arquitectura elegida per l'entrenament.
- discriminator_epoch_<*epoch*>_wcrh.pth o discriminator _epoch_<*epoch*>.pth, segons l'arquitectura elegida per l'entrenament.

on <epoch> és el nombre de èpoques en que varen ser entrenats.

Abans de començar l'entrenament, es preparen les configuracions necessàries per dur-lo a terme.

Els conjunts d'imatges utilitzats per l'entrenament són imatges de color en format RGB. Per tal de convertir-les en entrades del generador, es realitzen unes transformacions. Aquestes transformacions, redimensionen les imatges posant en l'eix més curt la dimensió informada en el subparàmetre d'entrada *image_size*, retallen les imatges redimensionades com un quadre de dimensions (*image_size*, *image_size*) centrat en la imatge i les converteixen en un Tensor amb les dimensions (C x H x W) en el rang [0.0, 1.0]. Per carregar el conjunt d'imatges i aplicar les transformació, s'utilitza un *ImageFolder*, es genera un *loader* mitjançant *DataLoader* aplicant la mida de lot (*batch-size*) informada com a subparàmetre.

Tal i com s'ha dit, la diferència entre les arquitectures construïdes radica en l'estructura del generador utilitzat. L'estructura del generador és molt similar a la descrita per Johnson et al. a Perceptual Losses for Real-Time Style Transfer and Super-Resolution. Conté tres blocs de downsampling seguits per una capa de normalització d'instància (Instance Normalization) cadascun d'ells, cinc blocs residuals, dos blocs de upsampling seguits per una capa de normalització d'instància cadascun d'ells i una

capa convolucional de sortida. Tots els blocs de downsampling i upsampling apliquen com a funció d'activació la funció d'unitat lineal rectificada per elements (ReLU).

Les capes convolucionals dels blocs de downsampling tenen la següent estructura:

1. Es carrega el tensor d'entrada mitjançant la reflexió del límit d'entrada (ReflectionPad2d).
2. S'aplica una convolució 2D sobre el tensor d'entrada compost per diversos plans d'entrada.

Els blocs residuals es construeixen amb dues capes convolucionals com les anteriors, amb els mateixos canals d'entrada i sortida, i una mida de Kernel igual a 3, seguides d'una capa de normalització d'instància. En el primer grup de capes dels blocs residuals s'aplica la funció d'unitat lineal rectificada per elements com a funció d'activació.

Les capes convolucionals dels blocs de upsampling tenen la mateixa estructura que les capes de downsampling, però s'aplica una augment de resolució d'entrada al factor d'escala donat (interpolació amb un factor d'escala).

En el cas de l'estructura que inclou el chroma subsampling, es deixen les mateixes capes pel canal 1 de luminància, i es substitueix el primer bloc de downsampling i l'últim de upsampling per l'estàndard corresponent en els canals 2 i 3 de cromància.

L'estructura del discriminador, implementada en l'arxiu discriminator.py, és la mateixa que la descrita per les Cycle-GAN en l'article Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks, utilitzant Patch-GAN de 70x70. Conté una capa convolucional amb la versió *Leaky* de la funció d'unitat lineal rectificada per elements com a funció d'activació i tres capes com l'anterior però intercalant una capa de normalització d'instància. Posteriorment s'emplenen els límits del tensor d'entrada amb zeros i per últim una capa convolucional.

Com a optimitzadors s'utilitza la versió Adam, tant pel generador com pel discriminador amb els subparàmetres *lr* com a taxa d'aprenentatge i, *b1* i *b2* com a factors d'oblit pels gradients i pels segons moments dels gradients, respectivament. Es defineixen uns programadors per aplicar un decaïment en la taxa d'aprenentatge amb el subparàmetres *n_epochs*, *epoch* i *decay_epoch*.

Les èpoques d'entrenament venen establertes per la diferència entre els subparàmetres inicials *n_epochs* i *epoch*. Per cada època i cada lot del *DataLoader* generat amb el conjunt d'imatges d'estil, es calculen les pèrdues i es computen els gradients, els quals s'apliquen als optimitzadors corresponents.

L'entrenament es divideix en dues parts, l'entrenament del generador i l'entrenament del discriminador.

En el cas del generador, es defineixen 3 funcions de pèrdua: la pèrdua adversarial, la pèrdua perceptiva i la pèrdua de diversitat. La pèrdua total esdevé una combinació de les anteriors.

Per la pèrdua adversativa, s'apliquen els mínims quadrats de la *GAN* a la funció de mapeig $G: \{x, z\} \rightarrow y$ i al seu discriminador, és a dir, l'objectiu del generador és:

$$L_{GAN} = E_{x \sim \rho_{data}(x)} \left[\left(D\left(G(x, z)\right) - L_{real} \right)^2 \right],$$

on *z* és el tensor del soroll i L_{real} és l'etiqueta per les dades reals, és a dir un tensor de uns. El generador tendeix a produir imatges $G(x, z)$ que s'assemblin a les imatges del domini *Y*. Per tant, aquest funció mesura la capacitat del generador per "enganyar" al discriminador en la generació d'imatges *fake* que s'assemblin al conjunt d'imatges reals.

Per la pèrdua perceptiva, s'aplica la funció descrita en l'article (Johnson, 2016) per mesurar les diferències semàntiques i perceptives entre les imatges. Un xarxa de codificació de característiques entrenada per la classificació d'imatges s'utilitza com extractor de característiques perceptives de les imatges. Siguin $\phi_j(x)$ les sortides de la capa $j - èssima$ de la xarxa de codificació de característiques ϕ en processar la imatge x. Si $j - èssima$ és una capa convolucional, $\phi_j(x)$ és un mapa de característiques de la forma $C_j \times H_j \times W_j$. Llavors la pèrdua perceptiva del generador és la distància euclidiana entre les representacions de les característiques, és a dir:

$$L_{PERCEPTUAL} = \frac{1}{C_j H_j W_j} \left[\left\| \phi_j(x) - \phi_j\left(G(x, z)\right) \right\|^2 \right]$$

Com a extractor e característiques s'utilitza una xarxa VGG-19 preentrenada en ImageNet.

Per poder plasmar la diversitat del conjunt d'entrada en la generació de sortides, s'introdueix soroll a l'entrada del nostre sistema. Com realitza (Ulyanov, 2016), s'utilitza una funció objectiu que pot afavorir la diversitat en la transferència d'estil, descrita com:

$$L_{TN} = -\frac{1}{N}\sum_{i=1}^{N} \lambda \ln \min_{j \neq i} \left\| g(z_i) - g(z_j) \right\|,$$

on N és el nombre de sorolls d'entrada i també el nombre de sortides, λ és el pes de la pèrdua de diversitat en la pèrdua total, i $g(z)$ és la xarxa per produir les imatges estilitzades. En aquest cas es proposa una nova funció de pèrdua per fomentar la diversitat en la traducció d'imatge a imatge. Es mesura la mitjana de la distància entre les imatges de sortida i s'utilitza la reciprocitat per maximitzar-la. La pèrdua de diversitat ve donada per:

$$L_{DIVERSITY} = -\frac{1}{N}\sum_{i=1}^{N} \frac{1}{mean_{j \neq i} \left\| g(z_i) - g(z_j) \right\| + \epsilon}$$

Per tal de produir el soroll en els lots d'entrada, es crea una funció per afegir soroll gaussià a les imatges.

Per tant, la pèrdua total a minimitzar, en el cas del generador quedaria:

$$L_{TOT} = L_{GAN} + \alpha L_{PERCEPTUAL} + \beta L_{DIVERSITY},$$

on α i β són el pesos de les pèrdues de percepció i de diversitat respectivament, i tenen un valor de 1e-2 cadascun d'ells.

En el cas del discriminador, l'objectiu és minimitzar la pèrdua adversarial, la qual és formula de la següent forma:

$$L_D = E_{y \sim p_{data}(y)} \left[\left(D(y) - L_{fake} \right)^2 \right] + E_{x \sim p_{data}(x)} \left[\left(D\left(G(x,z)\right) - L_{fake} \right)^2 \right],$$

on L_{fake} és l'etiqueta per les dades falses, és a dir, un tensor de zeros. El discriminador intenta distingir entre les imatges traslladades $G(x,z)$ de les imatges reals del domini *Y*.

Per tant, aquest funció mesura la capacitat del discriminador per endevinar si la imatges d'entrada és un *fake* o és real.

Per cada lot del conjunt d'entrada, s'actualitzen els optimitzadors del generador i del discriminador, i per cada època s'actualitzen les programacions del decaïment de la taxa d'aprenentatge.

Un cop finalitzat l'entrenament, es desen els models entrenats en mode d'avaluació.

4.3 Avaluació

Pel tal d'avaluar el model i realitzar la transferència d'estil, s'executa l'arxiu main.py (o main_wchr.py, segons com s'hagi entrenat el model) amb el paràmetre eval i amb els següents subparàmetres:

- --content-image: Ruta i nom de la imatge que volem estilitzar.
- --content-scale: Factor de disminució de la imatges a estilitzar, per defecte None.
- --output-image: Ruta i nom de la imatge de sortida.
- --model: Ruta i nom del model que es vol utilitzar per l'estilització.
- --cuda: Entorn d'execució utilitzat en l'estilització. S'informa 1 per executar sobre GPU, 0 per CPU. Obligatori.

Es carrega la imatge de contingut que s'hi vol transferir l'estil i es converteix en tensor. Es carrega el model del generador que es vol utilitzar per realitzar la transferència d'estil i "passem" el tensor de la imatge de contingut pel generador carregat. Per tal de convertir el tensor generat en una imatge vàlida, es realitza un deprocessament. Aquest procés multiplica de forma escalar el tensor per 255, aplica el mínim 0 i màxim 255 als valors del tensor convertit en matriu, es transposa els canals (0, 1, 2) → (1, 2, 0) i es converteix en imatges utilitzant la llibreria *Image*. Finalment, es desa la imatge estilitzada.

4.4 Resultats

S'han realitzat diversos experiments amb diferents estils a transferir i utilitzant les dues arquitectures construïdes, és a dir, utilitzant el chroma subsampling i sense. S'explica els resultats segons l'estil i arquitectura utilitzada a transferir. Cal anotar que els models s'han entrenat per etapes degut a les limitacions de hardware i la duració dels entrenaments.

4.4.1 Vicent van Gogh

Els experiments de transferència de l'estil de Vicent van Gogh s'ha realitzat amb un conjunt d'obres de l'autor de 400 imatges, entre les quals hi ha pintures de paisatges, retrats, escenes o natura morta. Un exemple del conjunt d'imatges és el següent:

Figura 16: Escala a Auvers de Vicent van Gogh

Experiment 1

El primer experiment amb aquest estil pictòric s'ha realitzat utilitzant l'arquitectura sense chroma subsampling. El entrenament s'ha dut a terme de 100 èpoques en 100 èpoques fins les 1400 èpoques amb el decaïment de la taxa de d'aprenentatge per defecte a partir de les 50 èpoques. A partir de les 1400 èpoques, s'ha entrenat de 200 en 200 fins les 2000 èpoques amb la mateixa configuració pel decaïment. En tots els entrenaments s'ha utilitzat una mida de lot de 4. Per tal de mostrar els resultats i degut a la heterogeneïtat de les imatges d'entrenament, s'ha volgut experimentar amb diverses imatges de contingut i les seves versions en resolució de 256. Les primeres imatges provades són la Figura 5: Fotografia del port de Dinan amb una resolució de 4000X3000 i la seva versió en resolució de 340x256. El resultat obtingut respectivament és:

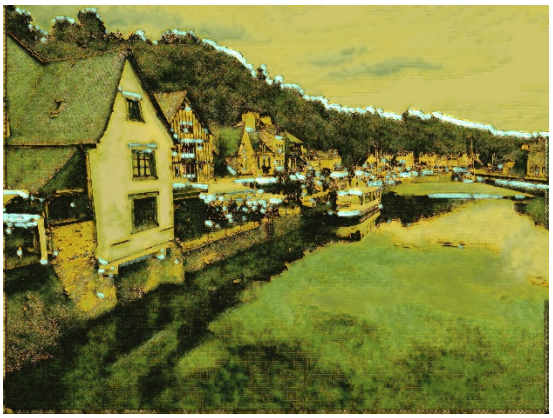


Figura 17: Resultat van Gogh amb paisatge de resolució 4000X3000 sense chroma subsampling

Figura 18: Resultat van Gogh amb paisatge de resolució 340X256 sense chroma subsampling

Es pot observar com en el cas de la fotografia amb resolució 4000X3000 el resultat no es satisfactori i no s'aprecia una transferència d'estil. En el cas de la fotografia amb resolució 340x256 es pot observar un inici de transferència d'estil però amb una qualitat i una coherència en la imatge insuficient.

La segona prova realitzada amb aquest model, es duu a terme amb un selfie personal, en aquest cas amb les versions de resolució 3000X4000 i de resolució 256X344:

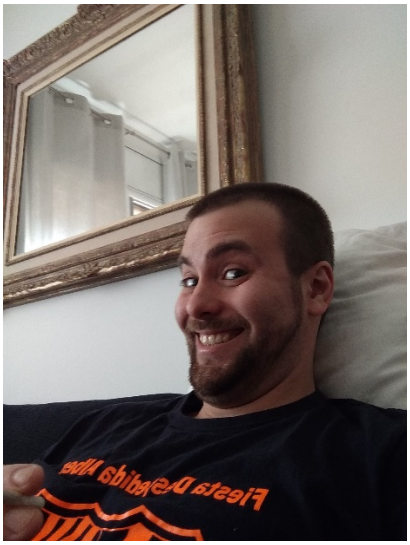


Figura 19: Selfie d'una persona

El resultat obtingut respectivament per les diferents versions és:

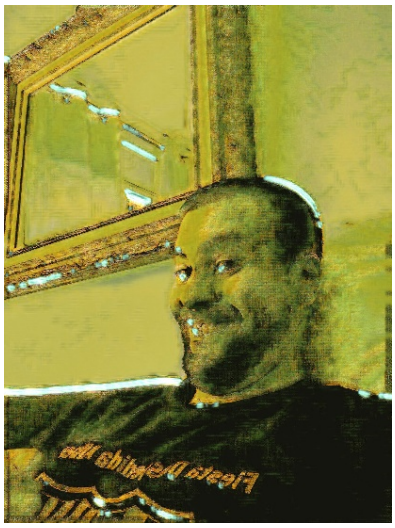


Figura 20: Resultat van Gogh amb selfie de resolució 3000X4000 sense chroma subsampling

Figura 21: Resultat van Gogh amb selfie de resolució 256X344 sense chroma subsampling

Es pot observar com en el cas de la fotografia amb resolució 3000X4000 el resultat no es satisfactori i no s'aprecia una transferència d'estil. Com en el cas anterior la fotografia amb resolució inferior, 340x256, es pot observar un inici de transferència d'estil però també amb una qualitat i una coherència en la imatge insuficient. El motiu de la qualitat pobra d'aquest resultats és la falta d'entrenament. S'entén que si l'entrenament s'hagués dut a terme durant més èpoques, es podria millorar els resultats. Aquesta falta d'entrenament ve donada per les limitacions del hardware i la duració dels entrenament que han fet impossible evolucionar més aquests entrenaments.

Amb l'objectiu de comparació, mostrem els resultats de la transferència d'estil del mètode CycleGAN (Zhu, 2017), utilitzant el notebook facilitat pels autors, amb les mateixes imatges:

Figura 22: Resultat van Gogh amb paisatges amb el mètode CycleGAN

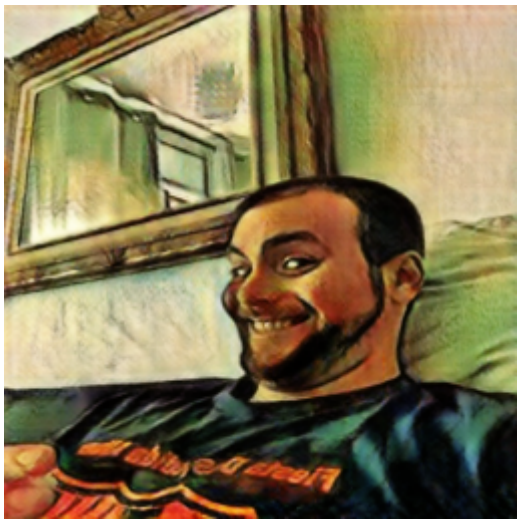


Figura 23: Resultat van Gogh amb selfie amb el mètode CycleGAN

Observem com el resultat visual és millor, però en el cas del "selfie", la transferència d'estil no és tan clara.

Per tal de calcular la mètrica FID d'aquest model s'ha utilitzat el codi facilitat en el repositori del GitHub https://github.com/mseitzer/pytorch-fid.git, s'ha comparat amb les mètriques obtingudes en altres projectes com CylceGAN i el mateix article de referència d'aquest estudi per l'estil pictòric de van Gogh:

	van Gogh
Cycle-GAN	180.37
Artsy-GAN	168.03
El nostre model	**155.67**

Figura 24: FID Experiment 1 van Gogh

Tal com indica (Liu H. N., 2018), la mètrica FID, podria copsar la similitud de les imatges generades amb les reals. Tanmateix, per a la transferència d'estil, no hi ha mètriques d'objectius reconegudes. El mètode obté puntuacions FID més petites, que es poden interpretar com a sortides més realistes que els mètodes en que els comparem, tot i observar resultats visuals no satisfactoris.

Experiment 2

El segon experiment amb aquest estil pictòric s'ha realitzat utilitzant l'arquitectura amb chroma subsampling. L' entrenament s'ha dut a terme de 100 èpoques en 100 èpoques fins les 300 èpoques amb el decaïment de la taxa de d'aprenentatge per defecte a partir

de les 50 èpoques. En tots els entrenaments s'ha utilitzat una mida de lot de 4. Les imatges utilitzades per l'avaluació d'aquest model han estat les mateixes que en el cas del primer experiment d'aquest estil pictòric. Els resultats obtinguts són els següents:

Figura 25: Resultat van Gogh amb paisatge de resolució 4000X3000 amb chroma subsampling

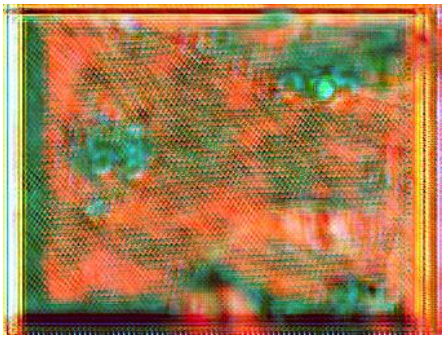


Figura 26: Resultat van Gogh amb paisatge de resolució 340X256 amb chroma subsampling

Figura 27: Resultat van Gogh amb selfie de resolució 3000X4000 amb chroma subsampling

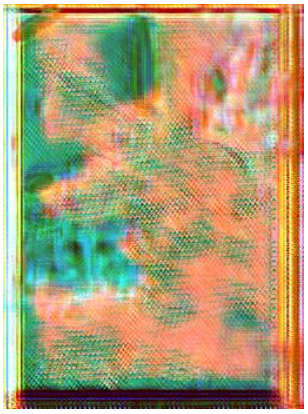


Figura 28: Resultat van Gogh amb selfie de resolució 256X344 amb chroma subsampling

Es veu clarament la falta d'entrenament del model. El model entrenat es incapaç de generar una imatges amb coherència. En el cas de les imatges amb les resolucions superior s'intueix el contingut de la imatges original, però en el cas de les resolucions inferiors es veu clarament que el generador no ha après el suficient dur a terme un transferència d'estil satisfactòria.

S'ha calculat la mètrica FID d'aquest model, de la mateixa forma que l'experiment anterior, i s'ha comparat amb les mètriques obtingudes en altres projectes com CylceGAN i del mateix article de referència d'aquest estudi per l'estil pictòric de van Gogh:

	van Gogh
Cycle-GAN	180.37

Artsy-GAN	168.03
El nostre model	**308.63**

Figura 29: FID Experiment 2 van Gogh

El mètode obté puntuacions FID bastant més altes, que es poden interpretar com a sortides menys realistes que els mètodes en que els comparem. En aquest cas, si es veu una correlació entre la mètrica i el resultat.

4.4.2 Jackson Pollock

Els experiments de transferència de l'estil de Jackson Pollock s'ha realitzat amb un conjunt d'obres de l'autor de 74 imatges, totes elles de l'estil d'expressionisme abstracte característic d'aquest autor. Un exemple del conjunt d'imatges és el utilitzat en l'estudi previ de l'arquitectura de transferència d'estil sense GAN's (Figura 7: No.5, 1948 de Jackson Pollock).

En aquest cas, tan sols s'ha pogut realitzar un experiment amb aquest estil pictòric, i aquest ha estat amb l'arquitectura amb chroma subsampling. L'entrenament s'ha realitzat, en una primera part de 100 èpoques amb el decaïment de la taxa de d'aprenentatge per defecte a partir de les 50 èpoques. Posteriorment, un entrenament de 300 èpoques, i a partir d'aquí, entrenaments consecutius de 400 èpoques cadascun fins les 2000 èpoques amb el decaïment de la taxa d'aprenentatge per defecte però a partir de la època 100. En tots els entrenaments s'ha utilitzat una mida de lot de 4. Per mostrar els resultats s'han utilitzat les mateixes imatges que en les proves de l'estil pictòric anterior, i els ha estat els següents:

Figura 30: Resultat Pollock amb paisatge de resolució 4000X3000 amb chroma subsampling

Figura 31: Resultat Pollock amb paisatge de resolució 340X256 amb chroma subsampling

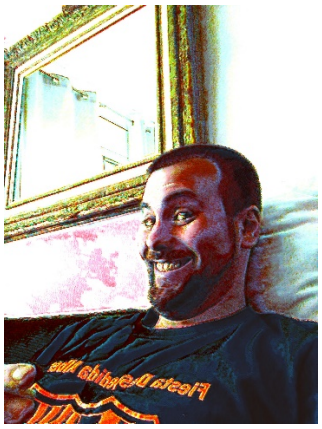


Figura 32: Resultat Pollock amb selfie de resolució 3000X4000 amb chroma subsampling

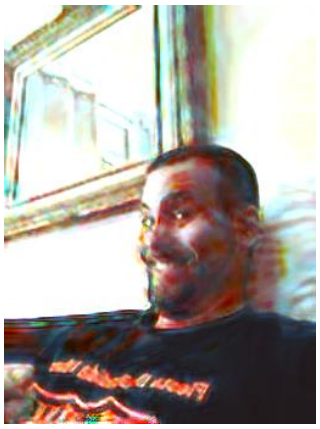


Figura 33: Resultat Pollock amb selfie de resolució 256X344 amb chroma subsampling

En aquest cas, es preveu que la falta de resultats satisfactoris, a banda de la falta d'entrenament ja comentat anteriorment, es suma la menor quantitat d'imatges disponibles per dur a terme l'entrenament, la qual cosa és un punt clau per l'èxit de l'entrenament d'una xarxa neuronal.

S'ha calculat la mètrica FID d'aquest model, de la mateixa forma que en el cas de la transferència d'estil de van Gogh. En aquest cas no tenim les mètriques dels mètodes CycleGan i de l'article de referència. La mètrica per aquest model, ens a donat **265.18**, tot i no ser comparable amb els valors de l'estil van Gogh, es veu que és un nombre superior, per tant es podria interpretar com a sortides poc realistes.

4.4.3 Pablo Picasso

Els experiments de transferència de l'estil de Pablo Picasso s'ha realitzat amb un conjunt d'obres de l'autor de 197 imatges, totes elles de l'estil cubista, on s'incorpora, tant paisatge, retrats com natura morta. Un exemple del conjunt d'imatges és el utilitzat en

l'estudi previ de l'arquitectura de transferència d'estil sense GAN's (Figura 10: Les senyoretes d'Avinyó de Pablo Picasso).

Com en el cas de l'estil de Pollock, tan sols s'ha pogut realitzar un experiment amb aquest estil pictòric, però aquest ha estat amb l'arquitectura sense chroma subsampling. L'entrenament també s'ha realitzat en tres parts:

- Primera part: Intervals de 200 èpoques amb el decaïment de la taxa de d'aprenentatge per defecte a partir de les 50 èpoques fins les 600 èpoques.
- Segona part: Intervals de 300 èpoques amb la mateixa configuració pel decaïment que el primer entrenament fins les 1200 èpoques.
- Tercera part: Intervals de 200 èpoques amb el decaïment de la taxa de d'aprenentatge per defecte a partir de les 50 èpoques fins les 1800 èpoques.

En tots els entrenaments s'ha utilitzat una mida de lot de 4. Per mostrar els resultats s'han utilitzat les mateixes imatges que en les proves dels estils pictòrics anteriors, i els ha estat els següents:

Figura 34: Resultat Picasso amb paisatge de resolució 4000X3000 sense chroma subsampling

Figura 35: Resultat Picasso amb paisatge de resolució 340X256 sense chroma subsampling

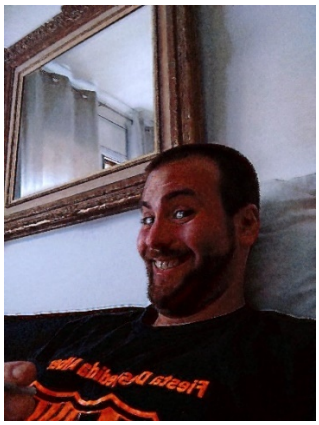


Figura 36: Resultat Picasso amb selfie de resolució 3000X4000 sense chroma subsampling

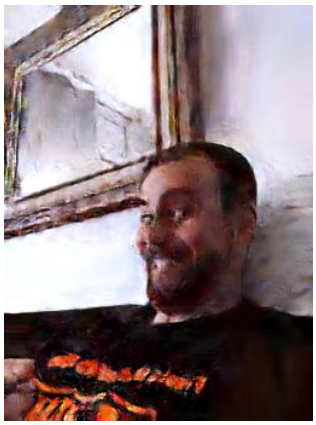


Figura 37: Resultat Picasso amb selfie de resolució 256X344 sense chroma subsampling

Els resultats tornen a ser no satisfactoris degut a la falta d'entrenament dels models i, probablement la falta d'imatges d'entrenament. En aquest cas, addicionalment, com en el cas de les imatges amb resolució superior, el model genera imatges quasi idèntiques que les originals, la qual cosa no és l'objectiu de l'estudi.

S'ha calculat la mètrica FID d'aquest model, de la mateixa forma que en els casos anteriors. Con en la transferència d'estil de Pollock, en aquest cas no tenim les mètriques dels mètodes CycleGan i de l'article de referència. La mètrica per aquest model, ens a donat **152.31**. Com en el cas de Pollock, no és comparable amb els valors de l'estil van Gogh, i es veu que el valor és lleugerament inferior. Aquest valor por ser degut a l'absència de transferència d'estil i la similitud de la imatges generada a la original.

5. Conclusions

En primer terme cal destacar, que a partir de la revisió bibliogràfica dels estudis anteriors referents a la transferència d'estil, s'ha descobert el interès que hi ha per aquesta rama de la visió per ordinador. Com, per exemple, els investigadors cada cop troben noves i millors formes per millorar els resultats de la transferència d'estil, fins i tot, mesclant disciplines com el processament natural del llenguatge (NLP) (Yi, 2017).

En el transcurs d'aquest estudi s'ha après a construir una arquitectura de xarxes generatives adversarials, la qual cosa ens ha permès endinsar en el món del Deep Learning. Degut al procés de construcció, i per poder anar resolent les problemàtiques que va plantejar l'article de referència, s'ha assolit un coneixement d'utilització framework Pytorch, construint xarxes neuronals i gestionant les pèrdues amb els optimitzadors. Sense oblidar que l'estudi previ de l'algorisme amb una xarxa neuronal convolucional, s'ha construït amb framework Keras, i també s'ha assolit cert nivell en la seva utilització. Per tant, he après dos frameworks per dur a terme projectes de Deep Learning en Python els quals eren prou desconeguts per mi.

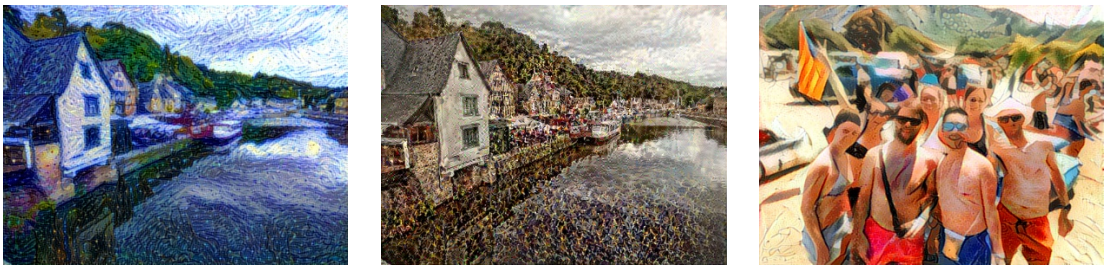
A nivell funcional, m'ha fascinat el funcionament d'una GAN. L'idea d'entrenar un model per tal que aprengui a crear imatges amb un estil, i a l'hora, entrenar un model que "auditi" la generació d'imatges del model anterior, ho he trobat molt interessant.

Referent a la construcció del model de GAN, podem recalcar les dificultats generades en l'aplicació del mètode de chroma subsampling. Segons l'article de referència, l'aplicació d'aquest tractament dels tensors en l'estructura del generador, millora substancialment el rendiment de l'entrenament del model. En el nostre codi, no hem observat cap millora entre l'arquitectura que no utilitza chroma subsampling de la que si l'utilitza. Un motiu pel qual no s'ha pogut investigar i millorar més aquest mètode és el temps de execució que requereixen aquests sistemes (el projecte de CycleGAN, utilitzant GPU, l'entrenament pot arribar a tardar dies).

A nivell de resultats, podem diferenciar-los en les dues construccions creades, és a dir, el cas de l'arquitectura sense GAN i l'arquitectura amb GAN.

En el cas de la primera construcció, els resultats han estat prou satisfactoris visualment. Podríem comentar que en el cas de la transferència d'estil de Jackon Pollock, el resultat ha estat una mica més escàs, però un motiu podria ser l'alt nivell d'abstracció de les seves obres. Per l'experiència rebuda durant l'estudi, tendim a pensar que els resultats d'aquests tipus de models són millors amb estil més figuratius que abstractes.

van Gogh Pollock Picasso



Resultats de l'estudi previ d'una arquitectura de transferència d'estil sense GAN.

En el cas de la segona construcció, i principal del present treball, els resultats no han estat satisfactoris. A banda dels resultats visuals, veiem resultats en els entrenaments que indiquen que els models no estan aprenen adequadament. Més concretament, en el cas del discriminador, en tots els models entrenats, la pèrdua total ha tendit a 0, la qual cosa indica un entrenament fallit. En el cas del generador, no s'han trobat signes erronis, ja que la pèrdua total del generador ha estat oscil·lant al voltant del valor 1. En la Figura 38 podem observar la evolució de les pèrdues totals del discriminador (a dalt) i el generador (a sota) en les últimes 200 èpoques de l'entrenament del model sense chroma subsampling de la transferència d'estil de Pablo Picasso.

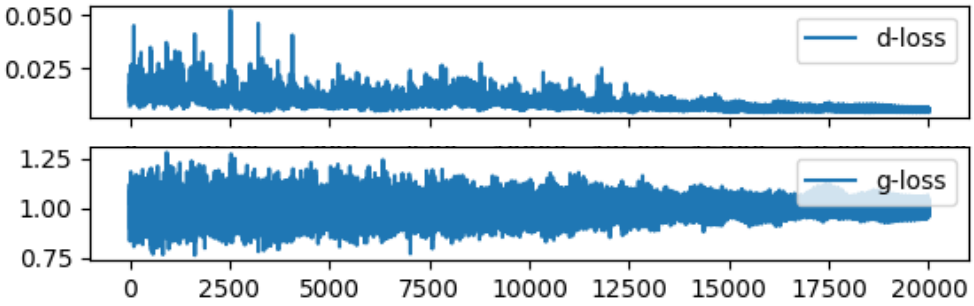


Figura 38: Gràfic de l'evolució de les pèrdues totals

van Gogh Pollock Picasso

Resultats de l'estudi principal amb l'arquitectura de transferència d'estil amb GAN.

Per tal de millorar els resultats d'aquest models en treballs futurs es poden realitzar diverses millores. En primer lloc, abastir-se del hardware necessari per tal de dur a terme

aquests entrenaments tant tediosos, o bé, la utilització de plataformes de pagament que posen a la nostra disposició els entorns necessaris per la seva execució. Amb l'equipament idoni, es poden realitzar múltiples experiments modificant els paràmetres del model per tal de trobar les configuracions més idònies. Un altre dels temes claus seria millorar la quantitat i la qualitat dels conjunts d'imatges d'entrenament. I per descomptat, la optimització del codi i dels mètodes del model, com podria ser la inserció de soroll aleatori a les entrades del discriminador amb un decaïment en el temps o la normalització de les entrades del generador en un rang de [-1, 1].

6. Glossari

GAN: Xarxa generativa adversarial

cGAN: Xarxa generativa adversarial condicional

Transferència d'estil: És una tècnica de visió per ordinador que ens permet recompondre el contingut d'una imatge a l'estil d'una altra.

Transferència d'estil neuronal: Fa referència a una classe d'algorismes de programari, que utilitzen Deep Learning, que manipulen imatges digitals o vídeos per adoptar l'aparença o l'estil visual d'una altra imatge.

Aprenentatge automàtic: És una aplicació d'intel·ligència artificial (AI) que proporciona als sistemes la capacitat d'aprendre i millorar automàticament a partir de l'experiència sense ser programats explícitament.

Deep Learning: és un conjunt d'algorismes d'aprenentatge automàtic que intenta modelar abstraccions d'alt nivell en dades utilitzant arquitectures computacionals que admeten transformacions no lineals múltiples i iteratives de dades expressades en forma matricial o tensorial.

7. Bibliografia

Cui, X. Q. (2018). The Intra-Class and Inter-Class Relationships in Style Transfer. *Applied Sciences*, 1681. doi:10.3390/app8091681

Denton, E. C. (2015). Deep generative image models using a Laplacian pyramid of adversarial networks. *NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems.*, (p. 1486-1494). Recollit de https://arxiv.org/abs/1506.05751

Gatys, L. E. (2016). Image Style Transfer Using Convolutional Neural Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* doi:10.1109/cvpr.2016.265

Gatys, L. E. (sense data). A Neural Algorithm of Artistic Style. *Journal of Vision*, (p. 326). doi:10.1167/16.12.326

Goodfellow, I. J.-A.-F. (2014). Generative adversarial nets. *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, (p. 2672-2680). Recollit de https://arxiv.org/abs/1406.2661

Isola, P. Z. (2017). Image-to-Image Translation with Conditional Adversarial Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* doi:10.1109/cvpr.2017.632

Johnson, J. A.-F. (2016). Perceptual Losses for Real-Time Style Transfer an Super-Resolution. *Computer Vision - ECCV*, (p. 694-711). doi:10.1007/978-3-319-46475-6_43

Kim, T. C. (2017). Learning to discover cross-domain relations with generative adversarial networks. *ICML'17: Proceedings of the 34th International Conference on Machine Learning.*, (p. 1857-1865). Recollit de https://arxiv.org/abs/1703.05192

Lazo, J. R. (2020). Mejorando el Proceso de Transferencia de Estilo Neuronal en Imágenes Añadiendo Mid-Level Representation. Recollit de http://hdl.handle.net/20.500.12590/16233

Li, M. H. (2018). Unsupervised Image-to-image Translation with Stacked Cycle-Consistent Adversarial Networks. *European Conference on Computer Vision (ECCV).* Recollit de https://arxiv.org/abs/1807.08536

Liu, H. N. (2018). Artsy-GAN: A style transfer system with improved quality, diversity and performance. *24th International Conference on Pattern Recognition (ICPR).* doi:10.1109/ICPR.2018.8546172

Liu, M. Y. (2017). Unsupervised image-toimage translation networks. *In Advances in Neural Information Processing Systems, NIPS 2017.* Recollit de https://arxiv.org/abs/1703.00848

Perera, P. A. (2018). In2I: Unsupervised Multi-Image-to-Image Translation Using Generative Adversarial Networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).* doi:10.1109/icpr.2018.8545464

Radford, A. M. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *Proceedings of the 4th International Conference on Learning Representations, ICLR 2016.* Recollit de https://arxiv.org/abs/1511.06434

Ulyanov, D. L. (2016). Texture Network: Feed-forward Synthesis of textures and Stylized Images. *Conference on Computer Vision and Pattern Recgnition (CVPR).* Recollit de https://arxiv.org/pdf/1603.03417.pdf

Wang, X. &. (2016). Generative Image Modeling Using Style and Structure Adversarial Networks. *European Conference on Computer Vision (ECCV)*, (p. 318-335). doi:10.1007/978-3-319-46493-0_20

Xie, X. T. (2007). Feature Guided Texture Synthesis (FGTS) for artistic style transfer. *Proceedings of the 2nd internacional conference on Digital interactive media in entertainment and arts - DIMEA '07.* doi:10.1145/1306813.1306830

Yi, Z. Z. (2017). DualGAN: Unsupervised Dual Learning for Image-to-Image Translation. *IEEE International Conference on Computer Vision (ICCV).* doi:10.1109/iccv.2017.310

Zhu, J. P. (2017). Unpaired Image-to-image Translation Using Cycle-Consistent Adversarial Networks. *IEEE International Conference on Computer Vision (ICCV).* doi:10.1109/iccv.2017.244