



UNIVERSITÉ PARIS 8 - VINCENNES À SAINT-DENIS

Master Informatique des Systèmes Embarqués

Memoire de projet tuteuré

Fakhri YAHIAOUI - Roman BOURSIER

Date de soutenance : le 09/06/2020

Tuteur – Université : Farès BELHADJ

Résumé

A faire en dernier ...

Remerciements

Idem ...

Table des matières

Résumé	1
Remerciements	2
Introduction	4
0.1 Problématique	4
1 Etat de l’art	7
1.1 Généralités sur les GANs	7
1.2 CGANs et traduction d’image	8
1.2.1 Pix2Pix	8
1.2.2 GauGan	8
1.2.3 CycleGan	8
1.2.4 Transfert de style neuronal	9
1.2.5 Learning to Sketch	9
1.2.6 Sketching : Inferring Contour Drawings from Images . .	10
2 Proposition de solution	11
2.1 Datasets	11
2.2 Modèle	11
2.3 Abstraction vers photo	12
2.4 Photo vers peinture	13
3 Résultats	15
4 Conclusion et Perspectives	17

Introduction

Dans le cadre de notre projet de fin d'études, nous souhaitons utiliser un modèle de Deep Learning, afin de produire un moteur de rendu capable d'adopter une stylisation « type » tel que la peinture chinoise. Dans un premier temps, il s'agira de proposer un modèle d'abstraction des peintures sélectionnées comme base d'apprentissage et d'utiliser le couple « peinture originale » / « abstraction » pour l'entraînement. Par la suite, un moteur de rendu d'abstractions sera connecté au réseau profond qui produira une peinture sur la base de l'abstraction.

Le modèle généré devra d'une part adopter la stylisation retenue mais aussi interpréter l'abstraction d'origine.

0.1 Problématique

La "traduction image-image" (Image-to-image translation) permet d'apprendre le mapping entre une image d'entrée et de sortie. En figure 1, nous testons la génération d'un paysage à partir d'un croquis simple. En figure 2, c'est un échec.

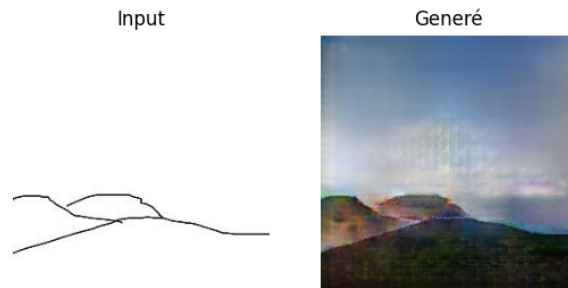


FIGURE 1 – Test du framework pix2pix [IZZE16] sur notre dataset composé de photos de paysages, labellisées en appliquant un filtre Canny [Can86] sur chacune d'elles.

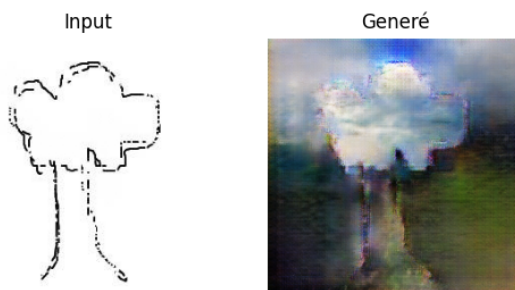


FIGURE 2 – Utilisation du même modèle avec une abstraction d'arbre

Comme évoqué dans [SPS⁺18] les algorithmes du type "traduction image-image" se basent essentiellement sur la corrélation d'une image à l'autre, et relève d'un apprentissage supervisé. Le rendu en figure 2 s'explique par la nature du dataset et par la distance importante qui sépare une abstraction d'une photo.

En figure 3 nous avons demandé à plusieurs personnes de dessiner un paysage composé de montagnes et d'arbres, éléments courants de la peinture chinoise. Ces dessins sont des abstractions, que nous souhaitons traduire en peintures chinoises.

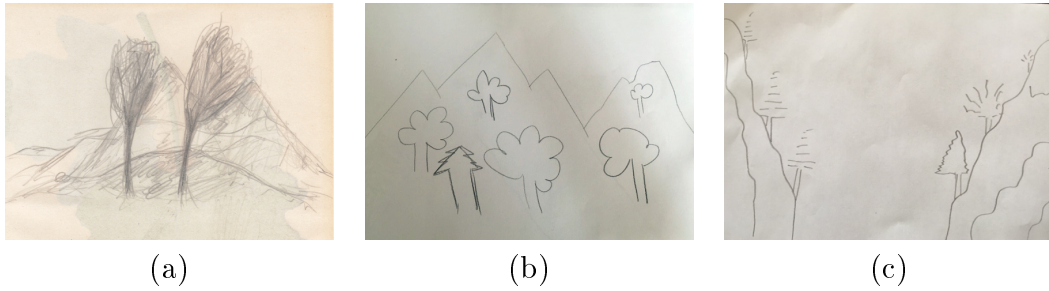


FIGURE 3 – Dessin de paysages réalisés par des personnes différentes. On note que les niveaux d'informations sont plus ou moins élevés. Le dessin (b) est très abstrait tandis que nous observons plus de détails pour (a) .

Chapitre 1

Etat de l'art

Nous présentons dans un premier temps les GANs qui sont au coeur de notre problématique. En un second lieu, nous étudions différents modèles existants basés sur les GANs conditionnels. Nous évoquons aussi, de manière plus succincte, des travaux liés à notre sujet mais n'utilisant pas les réseaux de neurones afin de pouvoir identifier les avantages et inconvénients des deux approches. Nous terminons par une liste des datasets utiles à notre sujet.

1.1 Généralités sur les GANs

"Les GANs (en anglais generative adversarial networks) sont une classe d'algorithmes d'apprentissage non-supervisé. Ces algorithmes ont été introduits dès 2014 par Goodfellow et permettent de générer des images avec un fort degré de réalisme." [Wik20]

Un générateur fabrique des données et les soumet au discriminateur dont le but est d'évaluer leurs degrés de crédibilité.

Le générateur $G(z, \theta_1)$ représente un réseau de neurones capable de mapper du bruit z vers l'espace désiré x . Le discriminateur $D(z, \theta_2)$ retourne la probabilité dans l'intervalle $[0, 1]$ que x vient du dataset original. θ_i représente les poids définis par chacun des modèles. Le générateur tente de maximiser la probabilité que les données x , soient classifiées comme appartenant au dataset d'origine et inversement, le discriminateur minimise la probabilité que de fausses images appartiennent au dataset d'origine.

Il existe aujourd'hui une très grande variété de travaux de recherche basés sur les GANs (BCGAN, AmbientGAN, ORGAN, Perceptual GAN). Le dépôt "The GANs Zoo" [hin18] référençait déjà en 2018 plus de 502 noms de GANs différents !

1.2 CGANs et traduction d'image

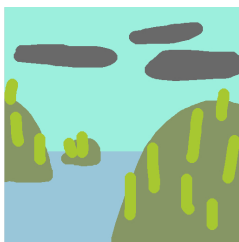
Les GANs conditionnels ajoutent une information supplémentaire y , partagée par le discriminateur et le générateur. Grâce aux cGANs il est possible de générer des images réalistes basées sur des labels de classes, des textes ou des images.

1.2.1 Pix2Pix

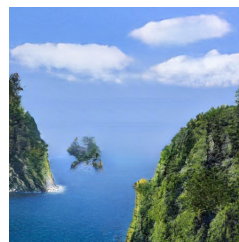
[GEB15] nous montre en quoi les cGANs peuvent permettre de résoudre efficacement les problèmes de traductions d'images et propose un framework applicable à n'importe quel domaine.

1.2.2 GauGan

[PLWZ19] permet de générer des paysages réalistes basés sur des masques de segmentation sémantiques. Les auteurs introduisent la "normalisation conditionnel" ou "adaptative", permettant de prendre en compte notamment les informations spatiales. Les couches de normalisations ont tendance à faire perdre de l'information contenu dans les masques sémantiques d'entrées car ils ne dépendent pas de données externes.



(a) Input - masques de segmentation sémantiques



(b) Output

FIGURE 1.1 – Test de rendu de paysage à partir de l'application GauGAN : <http://nvidia-research-mingyuliu.com/gaugan/>

1.2.3 CycleGan

CycleGan [ZPIE17], présente une approche pour la traduction d'une image d'un domaine source vers un domaine cible lorsque le dataset n'est pas apparié.

Le modèle possède deux générateurs $G : X \rightarrow Y$ et un second $F : Y \rightarrow X$ ainsi que deux discriminateurs D_Y and D_X

Les auteurs nous expliquent que si nous pouvons passer du domaine X vers le domaine Y et vice-versa, alors le résultat final devrait être identique à l'entrée initiale X .

1.2.4 Transfert de style neuronal

Introduit par Leon A. Gatys [GEB15] le TDN consiste à transférer un style à partir d'une image de référence vers une image de contenu. L'objectif est de transformer l'image d'entrée (bruit) en minimisant la distance avec l'image de contenu et avec l'image de style. On obtient alors par rétropropagation, une image qui correspond au contenu de l'image d'origine et au style souhaitée.

L'avantage de cette technique est qu'elle ne nécessite pas de dataset, seules deux images sont nécessaires.

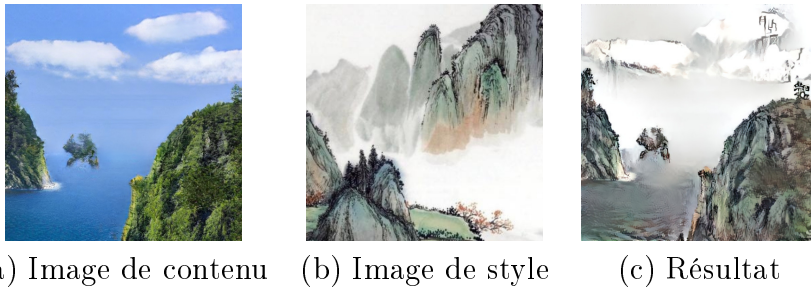


FIGURE 1.2 – Test de transfert de style réalisé à partir de la figure 1.3(b)

1.2.5 Learning to Sketch

[SPS⁺18] présente une méthode permettant de transformer une photo en croquis, en essayant d'imiter la façon de faire d'un humain. Le modèle est capable de réaliser le croquis séquentiellement, c'est-à-dire trait par trait. Les auteurs proposent de résoudre le problème des styles subjectifs et variés des dessins fait à la main en utilisant un modèle hybride supervisé/non-supervisé. L'objectif étant de palier "au signal de supervision faible et bruyante" induit par l'écart important entre un croquis et sa photo correspondante.

L'architecture est décomposé en 4 sous-modèles contenant chacun leurs propres sous-réseaux d'encodeurs et de décodeurs. Deux réseaux supervisés traduisent respectivement une photo en croquis $D(E(photo)) \rightarrow sketch$ et un croquis en photo $(D(E(sketch)) \rightarrow photo)$. Deux autres réseaux non-supervisés se chargent de la reconstruction. $D(E(photo)) \rightarrow photo$ et

$$D(E(\textit{sketch})) \rightarrow \textit{sketch}$$

1.2.6 Sketching : Inferring Contour Drawings from Images

[LLM⁺19] propose une nouvelle approche concernant la détection des contours dans une image. L'article montre que les solutions traditionnelles comme Canny [Can86] , captent uniquement les signaux de haute fréquence dans l'image sans la comprendre. Les auteurs ont collecté un dataset de 5000 paires "croquis humain/photos", crée manuellement via la plateforme de crowdsourcing "Amazon Mechanical Turk". En effet aucun dataset existant ne convenait (nombre d'éléments dans l'image, limites internes manquantes, le contenu non reconnaissable, les zones ombrées vides etc ..).

Ce modèle également du type "traduction image-image", permet d'avoir plusieurs labels différents pour la même photo, donc plusieurs interprétations différentes.



(a) Input



(b) Output

FIGURE 1.3 – Test du modèle pré-entraîné sur une photo de paysage

Chapitre 2

Proposition de solution

2.1 Datasets

Malgré nos recherches, nous n'avons pas pu trouver de dataset correspondant exactement à nos besoins, c'est-à-dire des paires d'abstraction/peinture chinoise. Pour les croquis, les plus grosses bases existantes sont TU-Berlin [EHA12], Sketchy [SBHH16] et Quickdraw. Après plusieurs essais, nous avons décidé d'abandonner leurs utilisations car il s'agit le plus souvent d'objets isolés et non appairés à une image réaliste. Pour la peinture chinoise, nous avons récupéré un premier dataset de 5000 images ([ych18]) et scrappés des plateformes comme google/baidu et pinterest.

Au final notre dataset est composé de 624 peintures de paysage chinoises réparties en 90/10 (90% des images destinées à l'entraînement et 10% pour les tests). Nous les avons ensuite réduites essentiellement pour des raisons de temps et d'apprentissage trop long. Chaque image sera par la suite redimensionnée et recadrée au format 256x256.

2.2 Modèle

Nous proposons de diviser l'apprentissage en deux étapes. La première consiste à traduire un croquis en dessin détaillé et la seconde d'un dessin détaillé vers la peinture. Notre proposition se base sur deux cGan de type "image-image", entraîné sur deux datasets appairées.

L'implémentation utilise le framework Keras et se base sur l'architecture pix2pix[1].

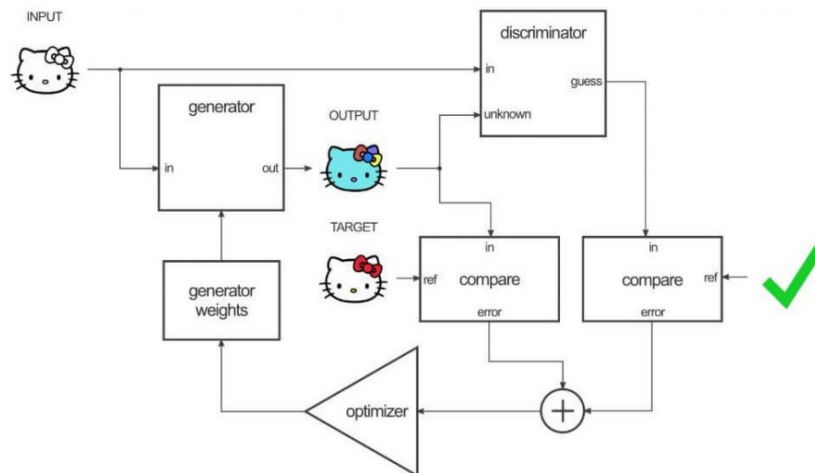


FIGURE 2.1 – Pix2pix 1.3(b)

2.3 Abstraction vers photo

Au départ, nous avons réalisé directement un pix2pix sur le dataset croquis/photos. Cependant, nous avons remarqué que les résultats n'étaient pas celles qu'on espérait (voir figure 2.2). Pour un dessin fait main nous pouvons constater que le résultat est un peu brouillon tandis qu'avec une image en entrée où le filtre Canny est appliqué le résultat est plus satisfaisant (voir figure 2.3)



Croquis

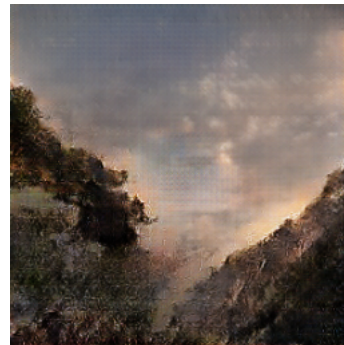


Résultat en couleur

FIGURE 2.2 – Résultat pix2pix croquis en entrée.



Image avec filtre Canny



Résultat en couleur

FIGURE 2.3 – Résultat pix2pix avec image Canny en entrée.

C'est pour cette raison, qu'après plusieurs expériences décevantes, nous avons choisi d'utiliser un algorithme de détection de bord nommé "holistically-nested edge detection" (HED) pour traduire nos peintures en dessins détaillés. Cette solution basée sur un modèle d'apprentissage profond permet de résoudre l'ambiguïté liée à la détection des contours et des objets.

A partir de cette sortie notre but est d'extraire le maximum d'éléments de l'image en ne gardant que les caractéristiques les plus représentatives. Nous avons testé plusieurs méthodes.

[Todo inclure les tests]

Après essais sur le modèle d'apprentissage, il s'avère que la labellisation manuelle reste dans notre cas le plus satisfaisant.

Nous pensons néanmoins que deux méthodes auraient mérité d'être approfondies, celle la segmentation ainsi que l'algorithme de Ramer–Douglas–Peucker. Dans les deux cas, l'avantage majeur est d'épargner la labellisation manuelle.

2.4 Photo vers peinture

Afin de réaliser la peinture chinoise à partir de la photo, nous avons pensé directement au cyclegan. Lors de nos recherches, nous avons pu constater des exemples de cyclegan générant des peintures de style monet à partir d'une photo.

N'ayant pas de photos de paysage qui ressemblent aux paysage dans les peintures chinoises, nous avons tout de même réussi à appliquer le style propre aux peintures chinoise sur ces photos.

De plus, pour plus de rapidité nous avons réduit le nombre de dataset donc il est probable que ça a eu un impact sur le résultat. Aussi, en raison de la durée limitée sur google colab l'entraînement n'a pas pu être fini (voir les

figures ci-dessous pour les résultats).



Photo donné au cyclegan



Photo avec peinture chinoise appliquée

FIGURE 2.4 – Résultat cyclegan de la photo à la peinture chinoise.



Photo donné au cyclegan



Photo avec peinture chinoise appliquée

FIGURE 2.5 – Résultat cyclegan de la photo à la peinture chinoise.

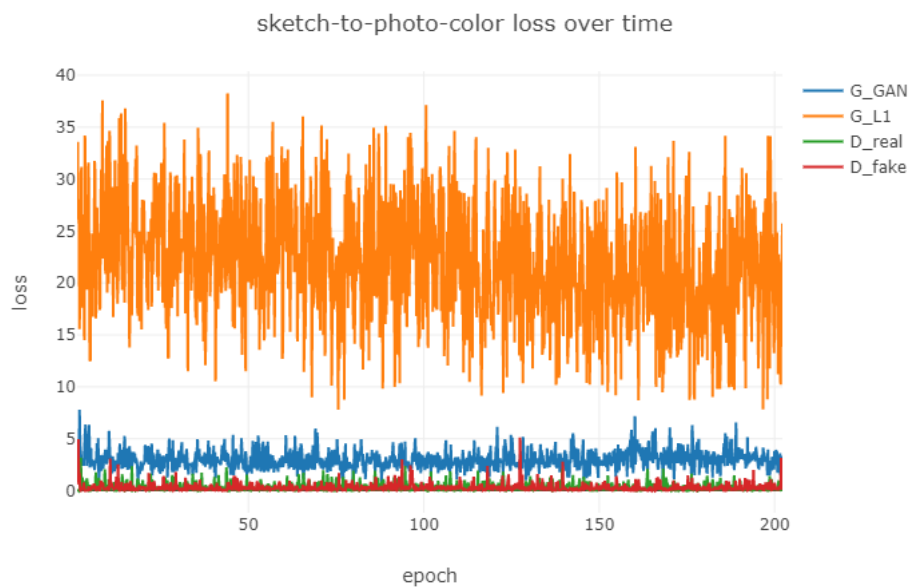
Comme on peut le constater, malgré le modèle non fini, à la figure 2.4 le résultat est assez réussi. Cependant on peut remarquer que dans la figure 2.5 les mêmes couches de peintures sont appliquées mais qui donne un résultat peu convaincant.

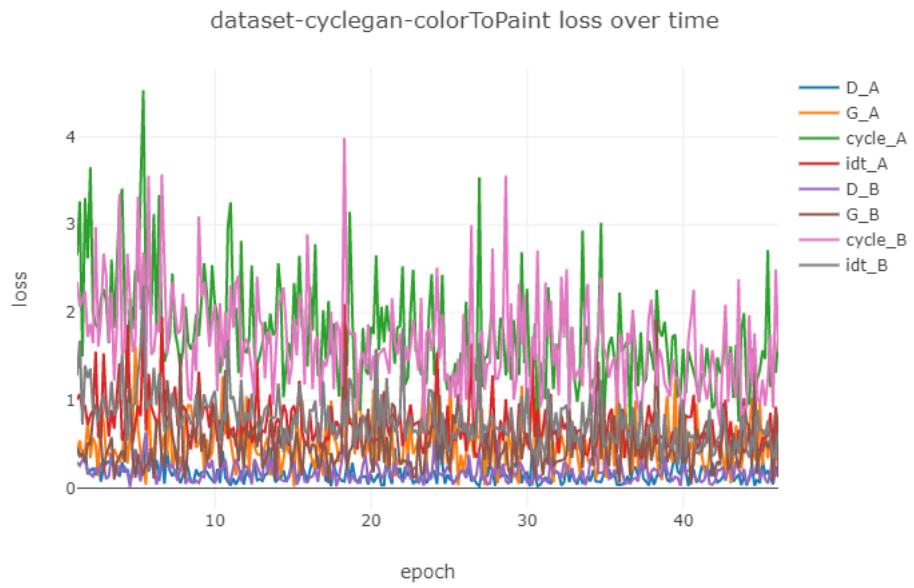
Ce résultat est probablement dû au fait que l'entraînement n'est pas fini (ça s'est arrêté à la 45ème epoch sur 200).

Chapitre 3

Résultats

- Graphique temps d'apprentissages (CPU collab/pc perso) / exécutions
- Plot d'images





Chapitre 4

Conclusion et Perspectives

Comme nous l'avons évoqué dans la section état de l'art, il est possible d'appliquer efficacement un style donné à une image, mais pour que le rendu soit crédible, il faut que l'entrée provienne d'une image réaliste. Notre principale difficulté était donc de transformer une abstraction en dessins réalistes. Les résultats présentés montrent que des modèles comme pix2pix permettent de réaliser ce transfert à condition d'avoir un dataset spécialisé (paysage) et appairé. Ainsi plus y a de variation et de disparité dans le dataset, moins la prédiction sera convaincante.

Par ailleurs, nous savons que les RCNNs, grâce aux couches de convolutions, sont capables de détecter les caractéristiques visuelles communes d'un ensemble d'images. Comme le prouve ce travail : <https://medium.com/artists-and-machine-intelligence/perception-engines-8a46bc598d57>, il est peut-être possible de générer une représentation abstraite de nos peintures en utilisant ce principe. Il serait alors théoriquement possible d'obtenir des paires abstractions/peintures, permettant l'entraînement d'un cGAN type "traduction image-image".

Les résultats pourraient être améliorés en augmentant le nombre d'éléments dans le dataset ainsi que le nombre d'époques.

Bibliographie

- [Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6) :679–698, 1986.
- [EHA12] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4) :44 :1–44 :10, 2012.
- [GEB15] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.
- [hin18] hindupuravinash. The gan zoo. <https://github.com/hindupuravinash/the-gan-zoo/>, 2018.
- [IZZE16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.
- [LLM⁺19] Mengtian Li, Zhe Lin, Radomír Mech, Ersin Yumer, and Deva Ramanan. Photo-sketching : Inferring contour drawings from images. *CoRR*, abs/1901.00542, 2019.
- [PLWZ19] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. *CoRR*, abs/1903.07291, 2019.
- [SBHH16] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database : Learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, 2016.
- [SPS⁺18] Jifei Song, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy M. Hospedales. Learning to sketch with shortcut cycle consistency. *CoRR*, abs/1805.00247, 2018.
- [Wik20] Wikipedia. Réseaux antagonistes génératifs — Wikipedia, the free encyclopedia. <http://fr.wikipedia.org/w/index.php?title=R%C3%A9seaux%20antagonistes%20g%C3%A9n>

- %C3%A9ratifs&oldid=170457375, 2020. [Online; accessed 17-May-2020].
- [ych18] ychen93. The gan zoo. <https://github.com/ychen93/Chinese-Painting-Dataset>, 2018.
- [ZPIE17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.