Vision and Learning in the Context of Exploratory Rovers

J. de Curtò

J. de Curtò.

Vision and Learning in the Context of Exploratory Rovers.

ETH Zürich.

J. DE CURTÒ

VISION AND LEARNING IN THE CONTEXT OF EXPLORATORY ROVERS

Doctor of Science at

ETH Zürich.
Department of Information Technology and Electrical Engineering.
curto@vision.ee.ethz.ch

May 2021. Zürich.

DISSERTATION ETHZ

VISION AND LEARNING IN THE CONTEXT OF EXPLORATORY ROVERS

A dissertation submitted to attain the degree of

DOCTOR OF SCIENCE of ETH ZÜRICH
(Dr. sc. ETH Zürich)

presented by

J. DE CURTÒ
Master of Science (CS) at Carnegie Mellon.
Master of Science (EE) at City University of Hong Kong.
Master of Science (Telecommunications) at
Universitat Autònoma de Barcelona.
Master of Science (Telecommunications) at
Universitat Politècnica de Catalunya.

Born on 19 September 1988.
Citizen of Catalunya (Regne d'Espanya).

accepted on the recommendation of

Prof. Dr. Luc van Gool, examiner.

2021

J. de Curtò: *Vision and Learning in the Context of Exploratory Rovers*, © 2021

DOI: 10.3929/ethz-a-

To my grandma and my grandpa. Bel i Bosch and De Curtó i Berengué.

ABSTRACT

Generative Adversarial Networks (GANs) [1, 2] have had tremendous applications in Computer Vision. Yet, in the context of space science and planetary exploration the door is open for major advances. We introduce tools to handle planetary data from the mission Chang'E-4 and present a framework for Neural Style Transfer using Cycle-consistency [3] from rendered images.

We also introduce a new real-time pipeline for Simultaneous Localization and Mapping (SLAM) and Visual Inertial Odometry (VIO) in the context of planetary rovers. We leverage prior information of the location of the lander to propose an object-level SLAM approach that optimizes pose and shape of the lander together with camera trajectories of the rover. As a further refinement step, we propose to use techniques of interpolation between adjacent temporal samples; videlicet synthesizing non-existing images to improve the overall accuracy of the system.

The experiments are conducted in the context of the Iris Lunar Rover, a nano-rover that will be deployed in lunar terrain in 2021 as the flagship of Carnegie Mellon, being the first unmanned rover of America to be on the Moon.

ZUSAMMENFASSUNG

Generative Adversarial Networks (GANs) [1, 2] hatten enorme Anwendungen in Computer Vision. Im Kontext der Weltraumforschung und der Erforschung der Planeten steht die Tür jedoch offen für große Fortschritte. Wir stellen Werkzeuge für den Umgang mit Planetendaten aus der Mission Chang'E-4 vor und präsentieren ein Framework für die Übertragung des neuronalen Stils unter Verwendung der Zykluskonsistenz [3] aus gerenderten Bildern.

Wir führen auch eine neue Echtzeit-Pipeline für Simultaneous Localization and Mapping (SLAM) und Visual Inertial Odometry (VIO) im Kontext von Planetenrovern ein. Wir nutzen vorherige Informationen über den Standort des Landers, um einen SLAM-Ansatz auf Objektebene vorzuschlagen, der die Pose und Form des Landers zusammen mit den Kameratrajektorien des Rovers optimiert. Als weiteren Verfeinerungsschritt schlagen wir vor, Interpolationstechniken zwischen benachbarten zeitlichen Abtastwerten zu verwenden. videlicet synthetisiert nicht vorhandene Bilder, um die Gesamtgenauigkeit des Systems zu verbessern.

Die Experimente werden im Rahmen des Iris Lunar Rover durchgeführt, eines Nano-Rovers, der 2021 als Flaggschiff von Carnegie Mellon als erstem unbemannten Rover Amerikas auf dem Mond im Mondgelände eingesetzt wird.

ACKNOWLEDGEMENTS

I would like to thank De Zarzà and my parents for their support during the development of this thesis.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

PLANETARY ROVERS

1.1 INTRODUCTION

Generative Adversarial Network (GAN) [1] are able to produce good quality high-resolution samples from images, both in the unscontrained and conditional setting [3–15]. Nonetheless, applications in the context of NASA missions and space exploration are scarce.

Given the difficulty to handle planetary data we provide downloadable files in PNG format from the missions Chang'E-3 and Chang'E-4[1]. In addition to a set of scripts to do the conversion given a different PDS4 Dataset. Example samples from the dataset can be seen in Figure 1.1. We also provide the corresponding labels, where localization information is present. We run extensive experiments to train a model able to be used as a hyperrealistic feature of the current simulator used in the Iris Lunar Rover [16].

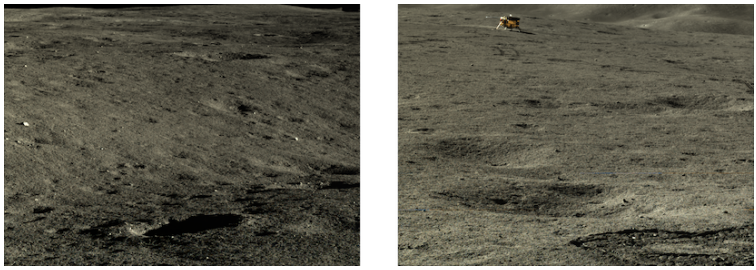

FIGURE 1.1: Images from the Moon. Panoramic camera of the rover. Chang'E-4.

1.2 OVERALL SYSTEM

Following the design principles and the perception pipeline proposed in [17] in the context of the NASA Mission Resource Prospector, we intend

1 Original PDS4 and PDS3 images and labels from missions to the Moon Chang'E can be obtained at moon.bao.ac.cn.

to design a simulator with hyperrealistic characteristics of the Moon that helps us deploy VIO/SLAM in a rover of the same characteristics. The intention is also that helps us address object detection and segmentation in this unmapped environment, where training data is very difficult and costly to obtain. Although at the present time data from the Moon is scarce, there are already some open datasets available in analogue environments such as the POLAR Stereo Dataset [18] that includes stereo pairs and LiDAR information or [19], that contains IMU, stereo pairs and odometry plus some additional localization data, all obtained on Mount Etna. Our intention is to provide downloadable files from the mission Chang'E-4 [20] that could be easily used in CV and ML pipelines. We also provide scripts to handle alternate PDS4 Datasets. The context where this tools are being used is our specific sensor suite, that will be on-board the Iris Lunar Rover, a project led by Carnegie Mellon that intends to put forward a four pound rover into the surface of the Moon by 2021 and that will be America's first rover to explore the surface of the planet, consists on IMU, two high-fidelity cameras and odometry sensors. Furthermore, it also has a UWB module [21–24] on-board to localize the rover with respect to the lander.

1.3 APPROACH, LONG-TERM GOAL AND PRIOR WORK

Generative image generation is a key problem in Computer Vision and Computer Graphics. Variational Autoencoders (VAE) [11, 25] try to solve the problem with an approach that builds on probabilistic graphical models. Autoregressive models (for instance PixelRNN [26]) have also achieved relative success generating synthetic images. In the past few years, Generative Adversarial Networks (GANs) [1, 2, 10, 27–30] have shown strong performance in image generation. Some works on the topic pinpoint the specific problem of scaling up to high-resolution samples [31], where conditional image generation is also studied while some recent techniques focus on stabilizing the training procedure [32–40]. Other promising novel approaches include score matching with LANGEVIN sampling [41, 42] and the use of sequence transformers for image generation [43].

The use of these techniques though have seen little or no applications in space exploration and planetary research. We propose here a framework that could be used to generate abundant data of the Moon, Mars and other celestial bodies, so that learning algorithms could be trained on Earth and

studied in simulation before being deployed in the real missions.

The proposed approach consists on using a technique of Neural Style Transfer or Generative Image Generation, such as the criteria of cycle-consistency, together with an augmentation of the given dataset (in our case using data from the lunar missions Chang'E-3 and Chang'E-4, but the same applies to Mars or other planets) using GANs in the setting of unconstrained image generation.

1.4 CYCLE-CONSISTENT GENERATIVE ADVERSARIAL NETWORKS

Our focus here is on Cycle-consistent Generative Adversarial Networks [3], where we work on unpaired image-to-image translation [44].

Image-to-image translation is a type of problem in Computer Vision and Computer Graphics where the objective is to learn a correspondence function between an input sample and an output sample, using a training set of aligned or non-aligned image pairs.

More precisely, our goal is to learn a function

$$G : X \to Y, \tag{1.1}$$

in a way that the distribution of samples $G(X)$ is as close as possible to the distribution Y. To accomplish this we are going to use an adversarial loss. Therefore, we couple it with the inverse correspondence

$$F : Y \to X, \tag{1.2}$$

and use a criteria of cycle-consistency to address the fact that the problem is highly under constrained

$$F(G(X)) \approx X \qquad \textit{and} \qquad G(F(Y)) \approx Y. \tag{1.3}$$

When we talk about paired training data, we refer to the fact that the training data consists of training examples $\{x_c, y_c\}_{c=1}^{N}$, where the correspondences between x_c and y_c are given. Instead, we say that we are using unpaired training data, when the set consists of two training sets $\{x_c\}_{c=1}^{N}$ and $\{y_a\}_{a=1}^{N}$, where there is not a correspondence explicitly given between which x_c corresponds to which y_a.

Formally, the GAN objective [1] involves finding a NASH equilibrium to the following two-player game:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}}\left[\log D(x)\right] + \tag{1.4}$$

$$+\mathbb{E}_{z\sim p_z}\left[\log(1-D(G(z)))\right], \tag{1.5}$$

where x is a ground truth image sampled from the true distribution p_{data}, and z is a noise vector sampled from p_z (that is, uniform or normal distribution). G and D are parametric functions where $G : p_z \to p_{data}$ maps samples from noise distribution p_z to data distribution p_{data}.

1.5 NEURAL STYLE TRANSFER

Neural Style Transfer [45–47] is based on the idea of synthesizing an original image by combining the content of one image together with the style of another sample. Here we will use cycle-consistent networks to attack this specific problem, with the aim of using a more general method that could help us solve concomitantly other tasks in the future. Moreover, the criteria of cycle-consistency assumes there is a bijection between the two domains, a constrain that could be often too restrictive, but that is very appropriate in our particular problem at-hand.

1.6 UNCONSTRAINED IMAGE GENERATION

To tackle the problems that arise when training Cycle-consistent networks with a dataset with few samples, i.e. mainly mode collapse and artifacts, we propose to use Unconstrained Image Generation using GANs to enlarge the original dataset with unseen examples, that is, as a way to generate additional training samples that will help the learning procedure converge to the desired solution. To achieve this we make use of the construction developed in [40].

1.7 EXPERIMENTS

Extensive experiments using data from Chang'E-3 and Chang'E-4 have been conducted, in particular we are using images from the panoramic camera

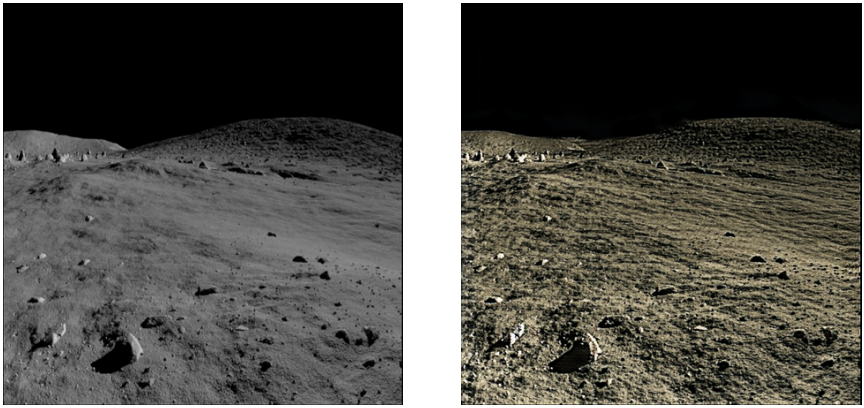

FIGURE 1.3: **Cycle-consistent gan**. Left: image from Kaggle, rendered simulator of the Moon. Right: style-Moon using our model. Trained at image size 512.

of the rover and from the terrain camera of the lander. Some examples can be seen in Figures 1.2 and 1.3, model trained at image size 256 and 512, respectively. As a source domain we are using samples from a rendered simulator of the Moon provided by Kaggle. The intention is to use the model in our actual renderer environment of the mission.

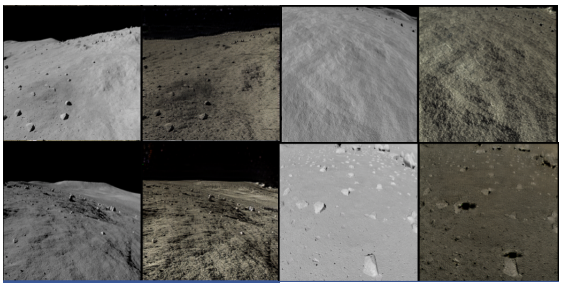

FIGURE 1.2: **Cycle-consistent gan**. Left: images from Kaggle, rendered simulator of the Moon. Right: style-Moon using our model. Trained at image size 256.

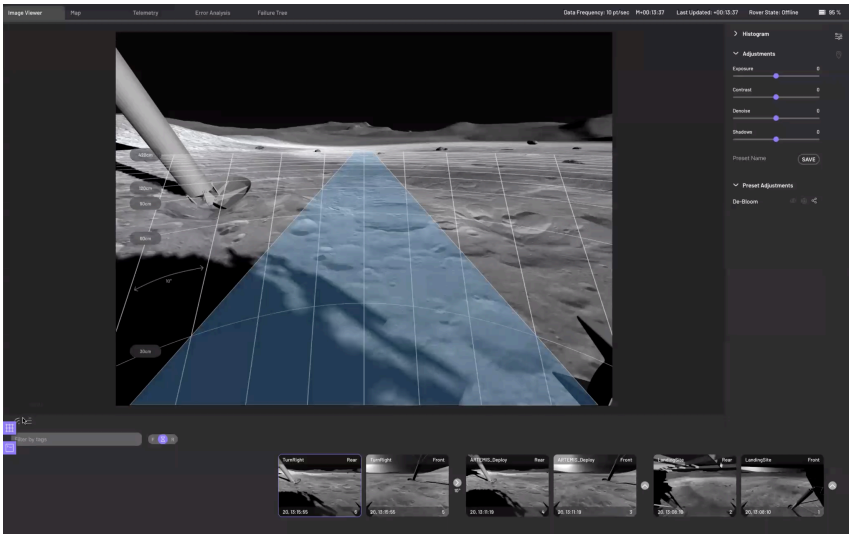

FIGURE 1.4: **Iris Lunar Rover**. Simulator used in the actual mission.

1.8 SIMULATOR

The context where this feature is being integrated is the actual simulator, see Figure 1.4, of the Iris Lunar Rover, the rover of Carnegie Mellon that will fly to the Moon onboard the Peregrine Lander of Astrobotic in 2021. Data from the simulator will be of the utmost importance to train and test localization algorithms such as SLAM/VIO [48, 49]. The ability to have ample data to train will also amplify the capabilities of the modules designed for segmentation [35, 50–52] and object detection [53–55]. As well as to test the software design before the real mission.

2

VIO AND SLAM IN LUNAR ROVERS

2.1 INTRODUCTION

Our aim is to present a novel pipeline to deploy state-of-the-art DL techniques in planetary rovers. With the advent of a new wave of planetary exploration missions, the need to call on generalizable perception and control systems that can operate autonomously in other worlds will become ubiquitous in the coming years.

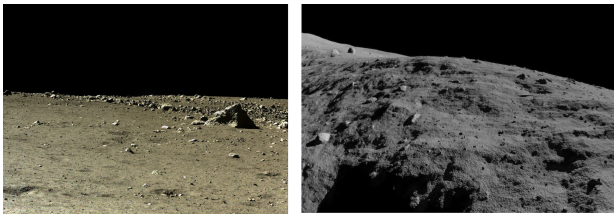

FIGURE 2.1: Left: real image from the Moon. Right: synthetic Moon.

2.2 OVERALL SYSTEM

Following the design principles and the perception pipeline proposed in [17] in the context of the NASA Mission Resource Prospector, we put forward an improved technique for Visual Odometry (VIO) that could be exploited in a rover of the same characteristics. Although at the present time data from the Moon is scarce, there are already some open datasets available in analogue environments such as the POLAR Stereo Dataset [18] that includes stereo pairs and LiDAR information or [19], that contains IMU, stereo pairs and odometry plus some additional localization data, all obtained on Mount Etna. Specifically for the task of semantic segmentation, Kaggle provides images from a rendered environment of the Moon and masks. More recently, as a benchmark for tasks of Computer Vision in the context of space exploration, a dataset containing PNG images and positioning information from the mission Chang'E-4 to the Moon has been released [16], the data from CE4 consists on post-processed original files

from the mission Chang'E[1]. Our specific sensor suite, that will be on-board the Iris Lunar Rover [16, 56], a project led by Carnegie Mellon that will deploy a four pound rover into the surface of the Moon by 2021 and that will be the first unmanned rover of America to explore the surface of the Moon, consists on IMU, two high-fidelity cameras and odometry sensors. Furthermore, it also has a UWB module [21–24] on-board to localize the rover with respect to the lander.

2.3 SLAM/VIO

Simultaneous Localization and Mapping (SLAM) and Visual Inertial Odometry (VIO) are defined as a function that transform raw data from the sensors into a distribution over the states of the robot. SLAM and VIO [48, 49] have been for decades unparalleled problems in robot perception and state estimation. Although typical dense SLAM systems are not differentiable, new approaches to solve this problem propose gradient-based learning over computational graphs to go all the way from 3D maps to 2D pixels [57].

The first task to tackle in geometric computer vision, being SLAM [58–60], Structure-from-Motion (SfM) [61–66], camera calibration or image matching, is to extract interest points [67, 68] from still images. We can define interest points as 2D specific locations in a given sample which can be considered stable and repeatable along different ambient conditions and viewpoints. The techniques used to traditionally attack this problem pertain to Multiple View Geometry [69], a subfield of mathematics that sets forth theorems and algorithms built on the assumption that those interest points can indeed be reliably extracted and matched across overlapping frames. Natheless, real-world computer vision operates on raw images that are far from the idealized conditions assumed in the proposed theory. Blending traditional modules with learning representations have lately been proven to be incredibly effective [68, 70, 71] as a way to bridge the gap between the conditions that we face in the real world and the assumptions made to design the algorithms. Plentiful of approaches also explore unsupervised learning of depth and ego-motion [72–74].

State-of-the-art approaches also deal with related problems such as SLAM object-level, that is, a system capable of optimizing object poses and shapes

1 moon.bao.ac.cn.

together with camera trajectory [75–77]. Although a SLAM system capable of incrementally mapping multi-object scenes seems not related to our task, its importance is revealed when we understand the fact that in many occasions the rover will localize itself with respect to the lander, which location is known; therefore a SLAM solution capable of optimizing the pose and shape of the lander along camera trajectory of the rover, would be distinctly adequate. With respect to this, we have to bear in mind that the principal technique that the rover will be using on-board to localize itself will be the UWB module [24]; that will indeed use the lander as a way-station for data communication. The reason for this is that critical weight and power can be hugely saved using RF for communication and state estimation. Thus, SLAM and VIO computation will be done on-ground. Using the same philosophy, it seems natural also to rely on a technique that will jointly optimize pose and shape of the lander together with camera trajectories.

2.4 SHAPE AND POSE OF THE LANDER

We assume here that we have a segmentation mask of the lander that in our specific case is obtained by the use of semantic segmentation [35, 51, 52, 73, 78–84]. On some of these approaches, the segmentation process is guided by the use of a prior object detector [54, 55, 85–88]. Specifically, we finetune our model building on DilatedResnet-101 [37, 89] and UperNet-101 [90, 91] trained on ADE20K [73]. Some examples of the mask given by our segmenter can be observed in Figure 2.2. To infer the shape and pose we will leverage existing techniques [77] that given a depth image, full shape and pose is determined. These techniques normally address multi-object categories; where a previous classification step and object observation is necessary, however our approach is somewhat simpler in the sense that the only object under consideration will be the lander per se.

2.5 TEMPORAL INTERPOLATION BETWEEN SUBSEQUENT SAMPLES

In the absence of continuous data between adjacent temporal samples given by the camera and to mitigate the effects that this will incur in the algorithms used to localize the rover, we propose to adopt techniques from video frame interpolation. Although signaling breakthroughs have been achieved by the use of recent deep convolutional neural networks, the quality of the resulting samples is often dubious due to object motion

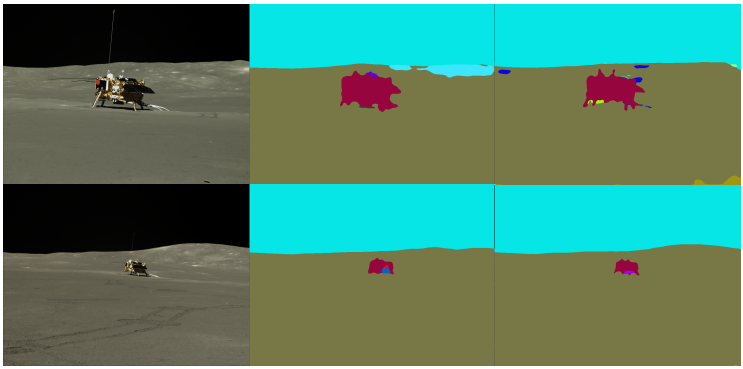

FIGURE 2.2: **Segmentation of the Lander**. Left: Image from CE4 [16]. In particular we are using color images from the panoramic camera of the rover of the mission to the Moon Chang'E-4. Middle: Generated mask given by model DilatedResnet-101 [37, 89]. Right: Generated mask given by model UperNet-101 [90, 91].

or occlusions. The main aim here is to synthesize non-existent frames in-between original samples to improve accuracy in the proposed VIO/SLAM approaches. Specifically for this purpose, we build on a recent depth-aware flow projection layer that achieves compelling upshots to synthesize intermediate sequences [92].

3

GENERATIVE ADVERSARIAL NETWORKS

3.1 INTRODUCTION

Developing a Generative Adversarial Network (GAN) [1] able to produce good quality high-resolution samples from images has important applications [3–16, 56] including image inpainting, 3D data, domain translation, video synthesis, image edition, semantic segmentation and semi-supervised learning.

FIGURE 3.1: **HDCGAN Synthetic Images**. A set of random samples. Our system generates high-resolution synthetic faces with an extremely high level of detail. HDCGAN goes from random noise to realistic synthetic pictures that can even fool humans. To demonstrate this effect, we create the Dataset of Curtò & Zarzà, the first GAN augmented dataset of faces.

In this paper, we focus on the task of face generation, as it gives GANs a huge space of learning attributes. In this context, we introduce the Dataset of Curtò & Zarzà [93], a well-balanced collection of images containing 14,248 human faces from different ethnical groups and rich in a wide range of learnable attributes, such as gender and age diversity, hair-style and pose variation or presence of smile, glasses, hats and fashion items. We also ensure the presence of changes in illumination and image resolution. We propose to use Curtò as de facto approach to empirically test the distribution learned by a GAN, as it offers a challenging problem to solve, while keeping the number of samples, and therefore training time, bounded. It can also be used as a drop-in substitute of MNIST for simple tasks of classification, say for instance using labels of ethnicity, gender, age, hair

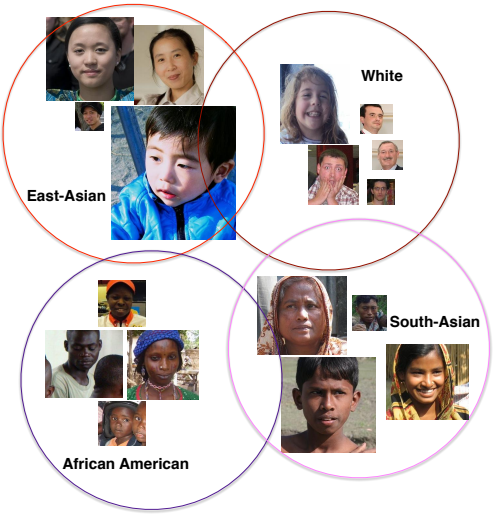

FIGURE 3.2: **Samples of Curtò**. A set of random instances for each class of ethnicity: African American, White, East-asian and South-asian. See Table 3.1 for numerics.

style or smile. It ships with scripts in TensorFlow and Python that allow benchmarks of classification. A set of random samples can be seen in Figure 3.2.

Despite improvements in GANs training stability [32–34] and specific-task design during the last years, it is still challenging to train GANs to generate high-resolution images due to the disjunction in the high dimensional pixel space between supports of the real image and implied model distributions [94, 95].

Our goal is to be able to generate indistinguishable sample instances using face data to push the boundaries of GAN image generation that scale well to high-resolution images (such as 512×512) and where context information is maintained.

In this sense, Deep Learning has a tremendous appetite for data. The question that arises instantly is, what if we were able to generate additional realistic data to aid learning using the same techniques that are later used to train the system. The first step would then be to have an image

generation tool able to sample from a very precise distribution (e. g. faces from celebrities) which instances resemble or highly correlate with real sample images of the underlying true distribution. Once achieved, what is desirable and comes next is that these generated image points not only fit well into the original distribution set of images but also add additional useful information such as redundancy, different poses or even generate highly-probable scenarios that would be possible to see in the original dataset but are actually not present.

Current research trends link Deep Learning and Kernel Methods to establish a unifying theory of learning [96–98]. The next frontier in GANs would be to achieve learning at scale with very few examples. To achieve the former goal this work contributes in the following:

- Network that achieves compelling results and scales well to the high-resolution setting where to the best of our knowledge the majority of other variants are unable to continue learning or fall into mode collapse.

- New dataset targeted for GAN training, Curtò, that introduces a wide space of learning attributes. It aims to provide a well-posed difficult task while keeping training time and resources tightly bounded to spearhead research in the area.

3.2 PRIOR WORK

Generative image generation is a key problem in Computer Vision and Computer Graphics. Remarkable advances have been made with the renaissance of Deep Learning. Variational Autoencoders (VAE) [11, 25] formulate the problem with an approach that builds on probabilistic graphical models, where the lower bound of data likelihood is maximized. Autoregressive models (scilicet PIXELRNN [26]), based on modeling the conditional distribution of the pixel space, have also presented relative success generating synthetic images. Lately, Generative Adversarial Networks (GANs) [1, 2, 10, 27–30] have shown strong performance in image generation. However, training instability makes it very hard to scale to high-resolution (256×256 or 512×512) samples. Some current works on the topic pinpoint this specific problem [31], where conditional image generation is also tackled while other recent techniques [32, 35–40] try to stabilize training.

3.3 DATASET OF CURTÒ & ZARZÀ

Curtò contains 14,248 faces balanced in terms of ethnicity: African American, East-asian, South-asian and White. Mirror images are included to enhance pose variation and there is roughly 25% per image class. Attribute information, see Table 3.1, is composed of thorough labels of gender, age, ethnicity, hair color, hair style, eyes color, facial hair, glasses, visible forehead, hair covered and smile. There is also an extra set with 3,384 cropped labeled images of faces, ethnicity white, no mirror samples included, see Column 4 in Table 3.1 for statistics. We crawled Flickr to download images of faces from several countries that contain different hair-style variations and style attributes. These images were then processed to extract 49 facial landmark points using [99]. We ensure using Mechanical Turk that the detected faces are correct in terms of ethnicity and face detection. Cropped faces are then extracted to generate multiple resolution sources. Mirror augmentation is performed to further enhance pose variation.

Curtò introduces a difficult paradigm of learning, where different ethnical groups are present, with very varied fashion and hair styles. The fact that the photos are taken using non-professional cameras in a non-controlled environment, gives us multiple poses, illumination conditions and camera quality.

TABLE 3.1: Attribute Information. Descending order of class instances by number of samples, Column 3.

Attribute	Class	# Samples	# Extra
Age	Early Adulthood	3606	966
	Middle Aged	2954	875
	Teenager	2202	178
	Adult	1806	565
	Kid	1706	85
	Senior	1102	402
	Retirement	436	218
	Baby	232	14
Ethnicity	African American	4348	0
	White	3442	3384

	East Asian	3244	0
	South Asian	3214	0
Eyes Color	Brown	9116	2119
	Other	4136	875
	Blue	580	262
	Green	416	128
Facial Hair	No	12592	2821
	Light Mustache	466	156
	Light Goatee	444	96
	Light Beard	258	142
	Thick Goatee	168	39
	Thick Beard	166	68
	Thick Mustache	154	62
Gender	Male	7554	1998
	Female	6694	1386
Glasses	No	12576	2756
	Eyeglasses	1464	539
	Sunglasses	208	89
Hair Color	Black	8402	964
	Brown	3038	1241
	Other	1554	253
	Blonde	616	543
	White	590	347
	Red	48	36
Hair Covered	No	12292	3060
	Turban	1206	76
	Cap	722	237
	Helmet	28	11
Hair Style	Short Straight	5038	1642
	Long Straight	2858	857
	Short Curly	2524	287
	Other	2016	249

	Bald	1298	187
	Long Curly	514	162
Smile	Yes	8428	2118
	No	5820	1266
Visible Forehead	Yes	11890	3033
	No	2358	351

3.4 APPROACH

Generative Adversarial Networks (GANs) proposed by [1] are based on two dueling networks, Figure 3.3; Generator G and Discriminator D. In essence, the process of learning consists of a two-player game where D tries to distinguish between the prediction of G and the ground truth, while at the same time G tries to fool D by producing fake instance samples as closer to the real ones as possible. The solution to a game is called NASH equilibrium.

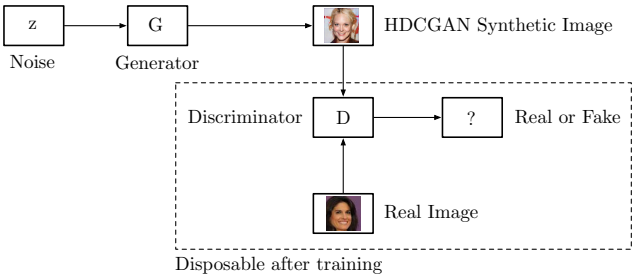

FIGURE 3.3: **Generative Adversarial Networks**. A two-player game between the Generator G and the Discriminator D. The dotted line denotes elements that will not be further used after the game stops, namely, end of training.

The min-max game entails the following objective function

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}}\left[\log D(x)\right] + \tag{3.1}$$

$$+\mathbb{E}_{z\sim p_z}\left[\log(1-D(G(z)))\right], \tag{3.2}$$

where x is a ground truth image sampled from the true distribution p_{data}, and z is a noise vector sampled from p_z (that is, uniform or normal distribution). G and D are parametric functions where $G : p_z \rightarrow p_{data}$ maps samples from noise distribution p_z to data distribution p_{data}.

The goal of the Discriminator is to minimize

$$L^{(D)} = -\frac{1}{2}\mathbb{E}_{x\sim p_{data}}\left[\log D(x)\right] - \tag{3.3}$$

$$-\frac{1}{2}\mathbb{E}_{z\sim p_z}\left[\log(1 - D(G(z)))\right]. \tag{3.4}$$

If we differentiate it w.r.t $D(x)$ and set the derivative equal to zero, we can obtain the optimal strategy

$$D(x) = \frac{p_{data}(x)}{p_z(x) + p_{data}(x)}. \tag{3.5}$$

Which can be understood intuitively as follows. Accept an input, evaluate its probability under the distribution of the data, p_{data}, and then evaluate its probability under the generator's distribution of the data, p_z. Under the condition in D of enough capacity, it can achieve its optimum. Note the discriminator does not have access to the distribution of the data but it is learned through training. The same applies for the generator's distribution of the data. Under the condition in G of enough capacity, then it will set $p_z = p_{data}$. This results in $D(x) = \frac{1}{2}$, that is actually the NASH equilibrium. In this situation, the generator is a perfect generative model, sampling from $p(x)$.

As an extension to this framework, DCGAN [2] proposes an architectural topology based on Convolutional Neural Networks (CNNs) to stabilize training and re-use state-of-the-art networks from tasks of classification. This direction has recently received lots of attention due to its compelling results in supervised and unsupervised learning. We build on this to propose a novel DCGAN architecture to address the problem of high-resolution image generation. We name this approach HDCGAN.

3.4.1 *HDCGAN*

Despite the undoubtable success, GANs are still arduous to train, particularly when we use big images (e. g. 512×512). It is very common to see D beating G in the process of learning, or the reverse, ending in unrecognizable imagery, also known as mode collapse. Only when stable learning is achieved, the GAN structure is able to succeed in getting better and better results with time.

This issue is what drives us to carefully derive a simple yet powerful structure that leverages common problems and gets a stable and steady training mechanism.

Self-normalizing Neural Networks (SNNs) were introduced in [100]. We consider a neural network with activation function f, connected to the next layer by a weight matrix $\mathbf{W}$, and whose inputs are the activations from the preceding layer x, $y = f(\mathbf{W}x)$.

We can define a mapping g that maps mean and variance from one layer to mean and variance of the following layer

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} \longmapsto \begin{pmatrix} \tilde{\mu} \\ \tilde{\nu} \end{pmatrix} : \begin{pmatrix} \tilde{\mu} \\ \tilde{\nu} \end{pmatrix} = g \begin{pmatrix} \mu \\ \nu \end{pmatrix}. \tag{3.6}$$

Common normalization tactics such as batch normalization ensure a mapping g that keeps (μ, ν) and $(\tilde{\mu}, \tilde{\nu})$ close to a desired value, normally (0, 1).

SNNs go beyond this assumption and require the existence of a mapping $g : \Omega \longmapsto \Omega$ that for each activation y maps mean and variance from one layer to the next layer and at the same time have a stable and attracting fixed point depending on (ω, τ) in Ω. Moreover, the mean and variance remain in the domain Ω and when iteratively applying the mapping g, each point within Ω converges to this fixed point. Therefore, SNNs keep activations normalized when propagating them through the layers of the network.

Here (ω, τ) are defined as follows. For n units with activation x_c, $1 \leq c \leq n$ in the lower layer, we set n times the mean of the weight vector $w \in \mathbb{R}^n$ as $\omega := \sum_{c=1}^{n} w_c$ and n times the second moment as $\tau := \sum_{c=1}^{n} w_c^2$.

Scaled Exponential Linear Units (SELU) [100] is introduced as the choice of activation function in Feed-forward Neural Networks (FNNs) to construct a mapping g with properties that lead to SNNs.

$$selu(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha \exp^x - \alpha & \text{if } x \leq 0. \end{cases} \tag{3.7}$$

Empirical observation leads us to say that the use of SELU greatly improves the convergence speed on the DCGAN structure, however, after some iterations mode collapse and gradient explosion completely destroy training when using high-resolution images. We conclude that although SELU gives theoretical guarantees as the optimal activation function in FNNs, numerical errors in the GPU computation degrade its performance in the overall min-max game of DCGAN. To alleviate this problem, we propose to use SELU and BatchNorm [101] together. The motivation is that when numerical errors move $(\tilde{\mu}, \tilde{\nu})$ away from the attracting point that depends on $(\omega, \tau) \in \Omega$, BatchNorm will ensure it is close to a desired value and therefore maintain the convergence rate.

Experiments show that this technique stabilizes training and allows us to use fewer GPU resources, having steady diminishing errors in G and D. It also accelerates convergence speed by a great factor, as can be seen after some few epochs of training on CelebA in Figure 3.8.

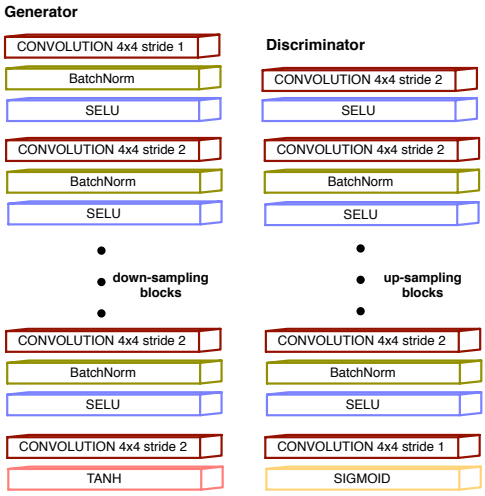

FIGURE 3.4: **HDCGAN Architecture**. Generator and Discriminator.

As SELU + BatchNorm (BS) layers keep mean and variance close to $(0,1)$ we get an unbiased estimator of p_{data} with contractive finite variance. These are very desirable properties from the point of view of an estimator as we are iteratively looking for a MVU (Minimum Variance Unbiased) criterion and thus solving MSE (Minimum Square Error) among unbiased estimators. Hence, if the MVU estimator exists and the network has enough capacity to actually find the solution, given a sufficiently large sample size by the Central Limit Theorem, we can attain NASH equilibrium.

HDCGAN Architecture is described in Figure 3.4. It differs from traditional DCGAN in the use of BS layers instead of ReLUs.

We observe that when having difficulty in training DCGAN, it is always better to use a fixed learning rate and instead increase the batch size. This is because having more diversity in training, gives a steady diminishing loss and better generalization. To aid learning, noise following a Normal $N(0,1)$ is added to both the inputs of D and G. We see that this helps overcome mode saturation and collapse whereas it does not change the distribution of the original data.

We empirically show that the use of BS induces SNNs properties in the GAN structure, and thus makes learning highly robust, even in the stark presence of noise and perturbations. This behavior can be observed when the zero-sum game problem stabilizes and errors in D and G jointly diminish, Figure 3.9. Comparison to traditional DCGAN, WASSERSTEIN GAN [102] and WGAN-GP [103] is not possible, as to date, the majority of former methods, such as [104], cannot generate recognizable results in image size 512×512, 24GB GPU memory setting.

Thus, HDCGAN pushes up state-of-the-art results beating all former DCGAN-based architectures and shows that, under the right circumstances, BS can solve the min-max game efficiently.

3.4.2 *Glasses*

We introduce here a key technique behind the success of HDCGAN. Once we have a good convergence mechanism for large input samples, that is a concatenation of BS layers, we observe that we can arbitrarily improve the final results of the GAN structure by the use of a Magnifying Glass

approach. Assuming our input length is $N \times M$, we can enlarge it by a constant factor, $\zeta_1 N \times \zeta_2 M$, which we call telescope, and then feed it into the network, maintaining the size of the convolutional filters untouched. This simple procedure works similar to how contact lenses correct or assist defective eyesight on humans and empowers the GAN structure to appreciate the inner properties of samples.

Note that as the input gets bigger so does the neural network. That is, the number of layers is implicitly set by the image size, see up-sampling and down-sampling blocks in Figure 3.4. For example, for an input size of 32 we have 4 layers while for an input size of 256 we have 7 layers.

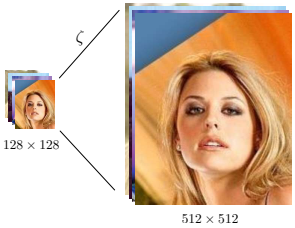

FIGURE 3.5: **Glasses on a set of samples from CelebA**. HDCGAN introduces the use of a Magnifying Glass approach, enlarging the input size by a telescope ζ.

We can empirically observe that BS layers together with Glasses induce high capacity into the GAN structure so that a NASH equilibrium can be reached. That is to say, the generator draws samples from p_{data}, which is the distribution of the data, and the discriminator is not able to distinguish between them, $D(x) = \frac{1}{2} \forall x$.

3.5 EMPIRICAL ANALYSIS

We build on DCGAN and extend the framework to train with high-resolution images using Pytorch. Our experiments are conducted using a fixed learning rate of 0.0002 and ADAM solver [105] with batch size 32 and 512×512 samples with the number of filters of G and D equal to 64.

In order to test generalization capability, we train HDCGAN in the newly introduced Curtò, CelebA and CelebA-hq.

Technical Specifications: 2 × NVIDIA Titan X, Intel Core i7-5820k@3.30GHz.

3.5.1 *Curtò*

The results after 150 epochs are shown in Figure 3.6. We can see that HDCGAN captures the underlying features that represent faces and not only memorizes training examples. We retrieve nearest neighbors to the generated images in Figure 3.7 to illustrate this effect.

FIGURE 3.6: **HDCGAN Example Results. Dataset of Curtò & Zarzà**. 150 epochs of training. Image size 512×512.

FIGURE 3.7: **Nearest Neighbors. Dataset of Curtò & Zarzà**. Generated samples in the first row and their five nearest neighbors in training (rows 2-6).

3.5.2 *CelebA*

CelebA is a large-scale dataset with 202,599 celebrity faces. It mainly contains frontal portraits and is particularly biased towards groups of ethnicity white. The fact that it presents very controlled illumination settings and good photo resolution, makes it a considerably easier problem than Curtò. The results after 19 epochs of training are shown in Figure 3.8.

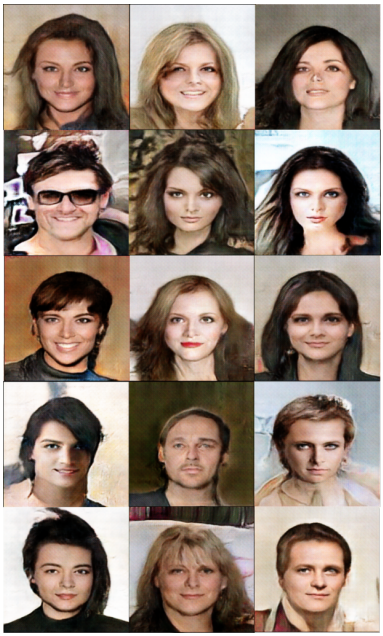

FIGURE 3.8: **HDCGAN Example Results. CelebA**. 19 epochs of training. Image size 512×512. The network learns swiftly a clear pattern of the face.

In Figure 3.9 we can observe that BS stabilizes the zero-sum game, where errors in D and G concomitantly diminish. To show the validity of our method, we enclose Figure 3.10, presenting a large number of samples for epoch 39. We also attach a zoomed-in example to appreciate the quality and size of the generated samples, Figure 3.11. Failure cases can be observed in Figure 3.12.

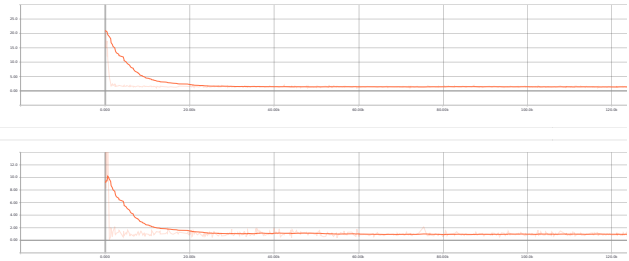

FIGURE 3.9: **HDCGAN on CelebA**. Error in Discriminator (top) and Error in Generator (bottom). 19 epochs of training.

FIGURE 3.10: **HDCGAN Example Results. CelebA**. 39 epochs of training. Image size 512×512. The network generates distinctly accurate and assorted faces, including exhaustive details.

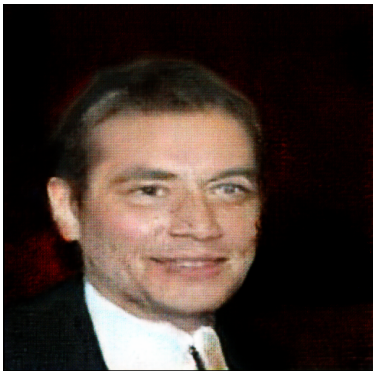

FIGURE 3.11: **HDCGAN Example Result. CelebA**. 39 epochs of training. Image size 512×512. 27% of full-scale image.

FIGURE 3.12: **HDCGAN Example Results. CelebA**. 39 epochs of training. Image size 512×512. Failure cases. The number of failure cases declines over time, and when present, they are of more meticulous nature.

Besides, to illustrate how fundamental our approach is, we enlarge Curtò with 4,239 unlabeled synthetic images generated by HDCGAN on CelebA, a random set can be seen in Figure 3.13.

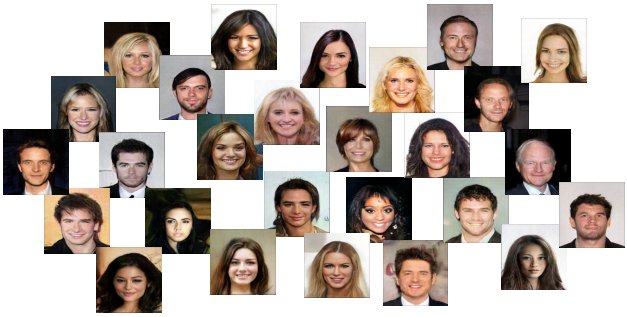

FIGURE 3.13: **HDCGAN Synthetic Images**. A set of random samples.

3.5.3 *CelebA-hq*

CelebA-hq in introduced in [38], a set of 30.000 high-definition images to improve training on CelebA. A set of samples generated by HDCGAN on CelebA-hq can be seen in Figures 3.1, 3.14 and 3.15.

FIGURE 3.14: **HDCGAN Example Results. CelebA-hq**. 229 epochs of training. Image size 512×512. The network generates superior faces, with great attention to detail and quality.

FIGURE 3.15: **HDCGAN Example Result. CelebA-hq**. 229 epochs of training. Image size 512×512. 27% of full-scale image.

To exemplify that the model is generating new bona fide instances instead of memorizing samples from the training set, we retrieve nearest neighbors to the generated images in Figure 3.16.

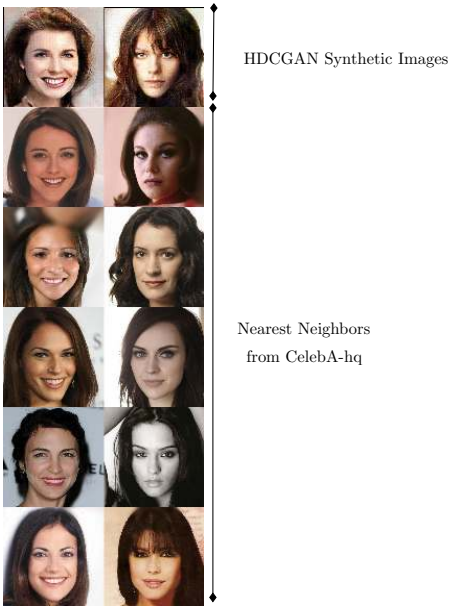

FIGURE 3.16: **Nearest Neighbors. CelebA-hq**. Generated samples in the first row and their five nearest neighbors in training (rows 2-6).

3.6 ASSESSING THE DISCRIMINABILITY AND QUALITY OF GENERATED SAMPLES

We build on previous image similarity metrics to qualitatively evaluate generated samples of generative models. The most effective of these is multi-scale structural similarity (MS-SSIM) [27]. We make comparison at resized image size 128×128 on CelebA. MS-SSIM results are averaged from 10,000 pairs of generated samples. Table 3.2 shows HDCGAN significantly improves state-of-the-art results.

	MS-SSIM
Gulrajani *et al.* [103]	0.2854
Karras *et al.* [38]	0.2838
HDCGAN	**0.1978**

TABLE 3.2: Multi-scale structural similarity (MS-SSIM) results on CelebA at resized image size 128×128. Lower is better.

We monitor MS-SSIM scores across several epochs averaging from 10,000 pairs of generated images to see the temporal performance, Figure 3.17. HDCGAN improves the quality of the samples while increases the diversity of the generated distribution.

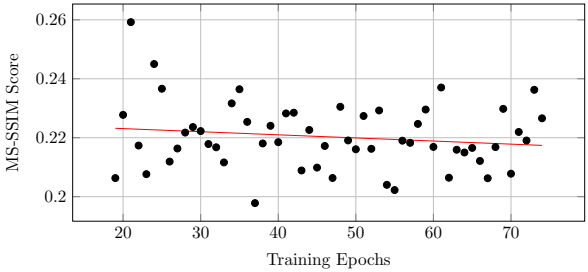

FIGURE 3.17: **MS-SSIM Scores on CelebA across several epochs**. Results are averaged from 10,000 pairs of generated images from epoch 19 to 74. Comparison is made at resized image size 128×128. Affine interpolation is shown in red.

In [106] they propose to evaluate GANs using the FRÉCHET Inception Distance, which assesses the similarity between two distributions by the difference of two Gaussians. We make comparison at resized image size 64×64 on CelebA. Results are computed from 10,000 512×512 generated samples from epochs 36 to 52, resized at image size 64×64 yielding a value of 8.44, Table 3.3, clearly outperforming current reported scores in DCGAN architectures [107].

	Fréchet
Karras *et al.* [38]	16.3
Wu *et al.* [107]	16.0
HDCGAN	**8.44**

TABLE 3.3: FRÉCHET Inception Distance on CelebA at resized image size 64×64. Lower is better.

3.7 DISCUSSION

In this chapter, we propose High-resolution Deep Convolutional Generative Adversarial Networks (HDCGAN) by stacking SELU + BatchNorm (BS) layers. The proposed method generates high-resolution images (e. g. 512×512) in circumstances where the majority of former methods fail. It exhibits a steady and smooth mechanism of training. It also introduces Glasses, the notion that enlarging the input image by a telescope ζ while keeping all convolutional filters unchanged, can arbitrarily improve the final generated results. HDCGAN is the current state-of-the-art in synthetic image generation on CelebA (MS-SSIM 0.1978 and FRÉCHET Inception Distance 8.44).

Further, we present a bias-free dataset of faces containing well-balanced ethnical groups, Curtò & Zarzà, that poses a very difficult challenge and is rich on learning attributes to sample from. Moreover, we enhance Curtò with 4,239 unlabeled synthetic images generated by HDCGAN, being therefore the first GAN augmented dataset of faces.

4

AN EFFICIENT SEGMENTATION TECHNIQUE

4.1 INTRODUCTION

Detection and Segmentation are key components in any toolbox of Computer Vision. In this paper we present a technique of hashing to segment an object given its bounding box and therefore attain simultaneously both Detection and Segmentation. At its heart lies a novel way to retrieve and generate a high-quality segmentation, which is crucial for a wide variety of CV applications. Simply put, we use a state-of-the-art convolutional network to detect the objects, but hashing on top of a high-quality hierarchy of regions to generate the segmentations [108].

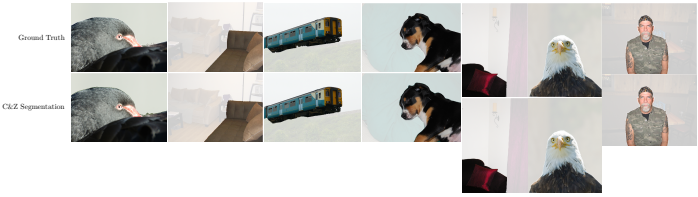

FIGURE 4.1: **Top Detections**. Top: VOC 2012 Ground Truth. Bottom: C&Z Segmentation.

Detection and Segmentation of Objects are two popular problems in Computer Vision and Machine Learning, historically treated as separated tasks. We consider these strongly related vision tasks as a unique one: detecting each object in an image and assigning to each pixel a binary label inside the corresponding bounding box.

C&Z Segmentation addresses the problem with a surprising different technique that deviates from the current norm of using proposal object candidates [79]. In semantic segmentation the need for rich information models that entangle some kind of notion from the different parts that constitute an object is exacerbated. To alleviate this issue we build on the use of the hierarchical model in [78] and explore the rich space of information of the Ultrametric Contour Map (UCM) in order to find the best possible semantic

segmentation of the given object. For this task, we exploit bounding boxes to facilitate the search. Hence, we simply hash the patches enclosed by the bounding boxes and retrieve closest nearest neighbors to the given objects, obtaining superior segmentations. Using this simple but effective technique we get the segmentation mask which is then refined using Hierarchical Section Pruning.

We start from a detector of bounding boxes and refine the object support, as in Hypercolumns [80]. We propose here a train-free similarity hashing alternative to their approach.

We present a simple yet effective module that leverages the need for a training step and can provide segmentations after any given detector. Our approach is to use a state-of-the-art region-based CNN detector [85] as prior step to guide the process of segmentation.

Outline: We begin next with a high-level description of the proposed method and develop further the idea to propose Hierarchical Section Hashing and Hierarchical Section Pruning in Section 4.1 and Section 4.3. Prior work follows in Section 4.2. We conclude with the evaluation metrics in Section 4.4 and a brief discussion in Section 4.5.

We start with a primer. C&Z Segmentation consists on the following main blocks:

- **Detection of Objects using Bounding Boxes.** We use a convolutional neural network [85] to detect all the objects in an image and generate the corresponding bounding boxes. We consider a detected object in an image as each output candidate thresholded by the class level score (benchmarks specifications in Section 4.4).

- **Hierarchy.** The image is presented as a tree of hierarchical regions based on the UCM [78].

- **Similarity Hashing.** We develop Hierarchical Section Hashing based on the LSH technique of [109].

- **Refinement of Regions.** The segmentations are refined by the use of Hierarchical Section Pruning.

- **Evaluation.** We evaluate the results on the PASCAL VOC 2012 Segmentation dataset [110] using the JACCARD index metric, which measures the average best overlap achieved by a segmentation mask for a ground truth object.

This work is inspired on how humans segment images: they first localize the objects they want to segment, they carefully inspect the object on the image by the use of their visual system and finally they choose the region that belongs to the body of that particular object. We believe that although the problem of detection has to be solved by the use of deep learning based on convolutional neural networks, in the same way that current breakthroughs have been attained in Generative Adversarial Networks (GAN) [1, 2, 32, 38, 93], the problem of segmenting those objects is of a different nature and can be best understood by the use of hashing. Furthermore, current research trends link concepts of Deep Learning to Kernel Methods, proposing a unifying theory of learning in [96–98].

Our main contributions are presented as follows:

- Novel approach to solve the task of segmentation by similarity hashing exploiting the detection of objects using bounding boxes.

- Use of hierarchical structures which are rich on semantic meaning instead of other current state-of-the-art techniques such as generation of proposal object candidates.

- No need of training data for the task of segmentation under the framework of detection using bounding boxes, that is train-free accurate segmentations.

- State-of-the-art results.

To our knowledge, we are the first to provide a segmentation based on hashing. This approach leverages the need to optimize over a high-dimensional space.

Despite the success of region proposal methods in detection, they have in turn arisen as the main computational bottleneck of these approaches. Yet unlike the latter, hierarchical structures derived from the UCM are in comparison inexpensive to compute and store. While we continue to use a very fast region-based convolutional neural network (R-CNN) to solve the task of detection, we propose to solve the problem of segmentation by exploring efficiently the space generated by a hierarchical image.

4.2 PRIOR WORK

Recent works [52, 111] present Object Detection and Segmentation as a single problem. The task requires to detect and segment every instance of a category in the image. Our work is however more closely related to the approach in Hypercolumns [80], where they go from bounding boxes to segmented masks. Our course of action is related in the sense that we propose an alternative that does not require a training step and can be used as an off-the-shelf high-quality segmenter.

For semantic segmentation [35, 51, 78, 79, 82–84], there has been several approaches where they guide the process of segmentation by the use of a prior detector [54, 55, 85–88]. Recently, this strategy has also presented state-of-the-art results in person detection and pose estimation [112]. Alternate procedures count on a human-on-the-loop [113]. Ongoing research on the matter uses Neural Architecture Search [20, 114–118] to design efficient architectures of neural networks for dense image prediction [119]. With the advent of present-day autonomous vehicles, the need to generate segmentations directly from the point cloud given by LIDAR [120–122], as well as detect 3D objects [89, 123–128], is also recently attracting lots of research efforts. Our segmenter starts rather than from raw pixels, [129] and [130], or bounding box proposals as in [131] and [111], from a set of hierarchical regions given by the UCM structure. Other techniques rely on superpixels e. g. [81]. This is a distinct tactic that works directly on a different representation.

4.3 SEGMENTATION C&Z

We delve into the details of the C&Z Segmentation construction, Figure 4.2.

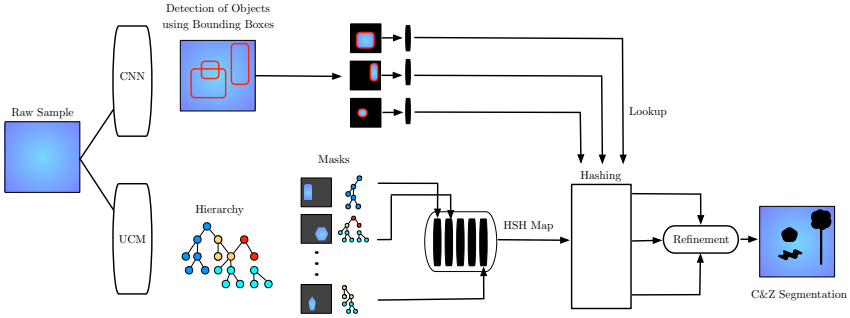

FIGURE 4.2: **C&Z Segmentation**. We construct a hierarchical image based on the UCM and 'train' the HSH map by hashing each region of the parent partition nodes. To retrieve a segmentation mask, we 'test' the HSH map by doing a lookup of the detected area enclosed by the bounding box, videlicet a fast search of approximate nearest neighbors on the hierarchical structure, and finally refine the result through HSP.

4.3.1 *Detection of Objects Using Bounding Boxes*

We begin by using the R-CNN object detector proposed by [85], which is in turn based on [54]. It introduces a Region Proposal Network (RPN) for the task of generating detection proposals and then solves the task of detection by the use of a FAST R-CNN detector. They train a CNN on ImageNet Classification and fine-tune the network on the VOC detection set. For our experiments, we use the network trained on VOC 2007 and 2012, and evaluate the results on the VOC 2012 evaluation set. We use the very deep VGG-16 model [132], Figure 4.3.

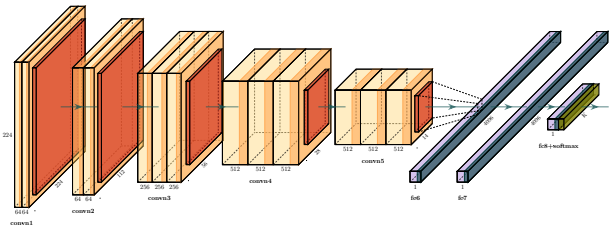

FIGURE 4.3: **VGG-16 Architecture**. VGG-16 model consists on an arrangement of convolutions, layers fully connected and softmax.

4.3.2 *Hierarchical Image*

We consider the representation of a hierarchical image described in [79]. Considering a segmentation of an image into regions that partition its domain $S = \{S_c\}_c$. A segmentation hierarchy is a family of partitions $\{S^*, S^1, \ldots, S^L\}$ such that: (1) S^* is the finest set of superpixels, (2) S^L is the complete domain, and (3) regions from coarse levels are unions of regions from fine levels.

4.3.3 *Hierarchical Section Hashing*

In this paper we introduce a novel segmentation algorithm that exploits bounding boxes to automatically select the best hierarchical region that segments the image. We introduce Hierarchical Section Hashing (HSH) which is in turn based on Locality-Sensitive Hashing (LSH). This algorithm helps us surpass the problem of computational complexity of the k-nearest neighbor rule and allows us to do a fast approximate neighbor search in the hierarchical structure of [78].

HSH can be summarized as follows:

- Detect bounding boxes on an image using a state-of-the-art convolutional neural network [85].
- Construct a hierarchical image by using the UCM and convey the result as a hierarchical region tree.
- Each hierarchical region is indexed by a number of tables of hashing using LSH and then constructing a HSH map.
- Each bounding box is hashed into the HSH map to retrieve the approximate nearest neighbor in sublinear time.

C&Z Segmentation has two main steps: first 'train' the HSH map with all the hierarchical regions of the image. Then 'test' the HSH map with all the detected bounding boxes to retrieve the approximate nearest neighbors that segment each of the objects in the image. The novelty of this approach is that it provides the best hierarchical region provided by the UCM structure

that segments the object image. C&Z exploits the detection of objects using bounding boxes because it relies on the correct detection of the object detector.

4.3.4 *Hierarchical Section Pruning*

The final piece is to refine the segmentations given by the HSH map by using what we call Hierarchical Section Pruning (HSP).

HSP procedure can be summarized as follows:

- Once a segmentation mask has been selected for all the objects in the given image, and their bounding boxes recomputed, the bounding box overlap ratio for all box pair combinations, according to the intersection over union criterium, is performed.

- Those masks that present overlap with other object masks on the same image are hierarchically unselected. We always proceed to unselect the low-level hierarchical regions, which by construction enclose a smaller region area and thus a single segmented object, from the high-level hierarchical region, which encloses more than one object and a bigger image area.

- Finally, isolated pixels on the mask are erased to preserve a single connected segmentation.

HSP is based on the fact that each segmentation mask represents a node on the hierarchical region tree constructed from the UCM. Therefore, hierarchical sections containing more than one object represent higher level nodes on the hierarchy. When HSP is applied, low-level hierarchical regions are unselected from the high-level hierarchical sections and therefore replaced by mid-low level sections on the same region tree structure that represents a single object or a smaller area of the image.

HSH and HSP Visual Examples can be seen in Figure 4.4.

	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	MBike	Person	Plant	Sheep	Sofa	Train	TV	Global
C&Z Segmentation (Instance Level)	45.4	27.5	55.9	44.2	42.0	43.2	41.3	66.3	31.4	57.2	42.3	63.3	43.8	43.6	40.9	40.6	57.2	51.2	48.0	54.1	45.2
C&Z Segmentation (Class Level)	33.3	18.5	48.1	37.5	40.7	45.1	39.4	59.9	23.3	51.0	43.3	60.4	39.8	43.1	34.6	37.2	51.0	47.0	53.6	54.2	43.1

TABLE 4.1: **VOC 2012 Validation Set**. Per-class and global JACCARD Index Metric at instance level.

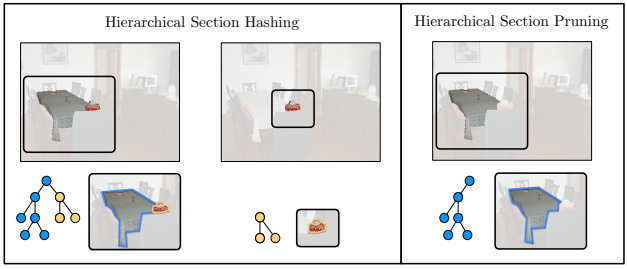

FIGURE 4.4: Left: HSH Visual Example. Right: HSP Visual Example.

C&Z Segmentation relies on the prior detection and therefore availability of bounding boxes for all the objects in a given image. The latter can be very useful as C&Z can be understood as a simple and effective technique to provide high-quality segmentations of still images after any available detector of bounding boxes. Likewise, you get train-free off-the-shelf accurate segmentations for any given method that detects bounding boxes.

4.3.5 *Locality Sensitive Hashing*

Our goal is to retrieve the k-nearest neighbors of a given hierarchical vector, which we call *image code*. In this setup we are limited by the curse of dimensionality and therefore using an exact search is inefficient. Our approach uses a technique of approximate nearest neighbors: Locality Sensitive Hashing (LSH).

A LSH function maps $x \rightarrow h(x)$ such that the similarity between $(\mathbf{x}, \mathbf{y})$ is preserved as

$$\left| \frac{d(h(\mathbf{x}), h(\mathbf{y}))}{D(\mathbf{x}, \mathbf{y})} - 1 \right| \leq \epsilon \tag{4.1}$$

which is not possible for all $D(\mathbf{x}, \mathbf{y})$ but available for instance for euclidean metrics.

We build on the LSH work of [109, 133–137]. LSH is a randomized hashing scheme, investigated with the primary goal of $\epsilon - R$ neighbor search. Its main constitutional block is a family of locality sensitive functions. We can define that a family $\mathcal{H}$ of functions $h : \mathcal{X} \to \{0, 1\}$ is (p_1, p_2, r, R)-sensitive if, for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$,

$$Pr_{h \sim U[\mathcal{H}]}(h(\mathbf{x}) = h(\mathbf{y}) \mid ||\mathbf{x} - \mathbf{y}|| \leq r) \geq p_1, \tag{4.2}$$

$$Pr_{h \sim U[\mathcal{H}]}(h(\mathbf{x}) = h(\mathbf{y}) \mid ||\mathbf{x} - \mathbf{y}|| \geq R) \leq p_2, \tag{4.3}$$

where these probabilities are chosen from a random choice of $h \in \mathcal{H}$.

Algorithm 1 gives a simple description of the LSH algorithm for the given case when the distance of interest is L_1, which is the one in use in C&Z Segmentation. The family $\mathcal{H}$ in this case contains axis-parallel stumps, which means the value of $h \in \mathcal{H}$ is generated by taking a simple dimension $d \in \{1, \ldots, \dim(\mathcal{X})\}$ and thresholding it with some T:

$$h^{LSH} = \begin{cases} 1 & \text{if } x_d \leq T, \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

A LSH function $g : \mathcal{X} \to \{0, 1\}^k$ is formed by independently k functions $h_1, \ldots, h_k \in \mathcal{H}$.

That is, we can understand that an example in our hierarchical partition $S_c \in \mathcal{S}$ provides a k-bit key

$$g(S_c) = [h_1(S_c), \ldots, h_k(S_c)]. \tag{4.5}$$

This process is repeated l times and produces l independently constructed functions of hashing $g_1, \ldots, g_l$. The available reference ('training') data S are indexed by each one of the l functions of hashing, producing l tables of hashing, namely each of the S_c hierarchical partitions generated by all the corresponding parents of the hierarchical tree.

Algorithm 1 LSH Algorithm [134]

Given: Dataset $X = [\mathbf{x}_1, \mathbf{x}_N], \mathbf{x}_c \in \mathbb{R}^{\dim(X)}$.
Given: Number of bits k, number of tables l.
Output: A set of

1: **for** $z = 1, \ldots, l$ **do**
2: **for** $c = 1, \ldots, k$ **do**
3: Randomly (uniformly) draw

$$d \in \{1, \ldots, \dim(\mathcal{X})\}.$$

4: Randomly (uniformly) draw

$$\min\{\mathbf{x}_{(d)}\} \leq v \leq \max\{\mathbf{x}_{(d)}\}.$$

5: Let h_c^z be the function $\mathcal{X} \to \{0, 1\}$ defined by

$$h_c^z(\mathbf{x}) = \begin{cases} 1 & \text{if } x_{(d)} \leq v, \\ 0 & \text{otherwise.} \end{cases}$$

6: The z-th LSH function is $g_z = [h_1^z, \ldots, h_k^z]$.

Once the LSH data structure has been built it can be used to perform a very efficient search for approximate neighbors in the following way. When a query S_0 arrives, we compute its key for each table of hashing z, and record the examples $C = \{S_1^l, \ldots, S_{n_l}^l\}$ resulting from the lookup with that key. In other words, we find the 'training' examples that fell in the same bucket of the l-th table of hashing to which S_0 would fall. These l lookup operations produce a set of candidate matches, $C = \cup_{z=1}^{l} C_z$. If this set is empty, the algorithm reports it and stops. Otherwise, the distances between candidate matches and S_0 are explicitly evaluated, and the examples that match the search criteria, which means that are closer to S_0 than $(1 + \epsilon)R$, are returned.

4.4 EVALUATION AND RESULTS

We extensively evaluate C&Z Segmentation on VOC 2012 validation set. Top detections from our algorithm can be seen in Figure 4.1.

4.4.1 *JACCARD Index Metric*

In Table 4.1 we show the results of the JACCARD Index Metric. This measure represents the average best overlap achieved by a candidate for a ground truth object.

C&Z Segmentation with Jaccard at instance level 45.24% and Jaccard at class level 43.05%. Recall at overlap 0.5 is 43.36%.

4.5 DISCUSSION

In this paper we introduce C&Z Segmentation, an algorithm to segment objects based on hashing that exploits the detection using bounding boxes. We show C&Z achieves compelling results and generates off-the-shelf accurate segmentations.

BIBLIOGRAPHY

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. Generative Adversarial Networks. *NIPS* **27** (2014).
2. Radford, A., Metz, L. & Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *ICLR* (2016).
3. Zhu, J., Park, T., Isola, P. & Efros, A. A. Unpaired Image-to-Image Translation using Cycle-consistent Adversarial Networks. *IEEE International Conference on Computer Vision* (2017).
4. Wu, J., Zhang, C., Xue, T., Freeman, W. T. & Tenenbaum, J. B. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-adversarial Modeling. *NIPS* **29** (2016).
5. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O. & Li, H. High-resolution Image Inpainting using Multi-scale Neural Patch Synthesis. *IEEE International Conference on Computer Vision* (2017).
6. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D. & Krishnan, D. Unsupervised Pixel-level Domain Adaptation with Generative Adversarial Networks. *IEEE International Conference on Computer Vision* (2017).
7. Li, C., Xu, K., Zhu, J. & Zhang, B. Triple Generative Adversarial Nets. *NIPS* **30** (2017).
8. Wang, T., Liu, M., Zhu, J., Tao, A., Kautz, J. & Catanzaro, B. High-resolution Image Synthesis and Semantic Manipulation with Conditional GANs. *IEEE International Conference on Computer Vision* (2018).
9. Wang, T., Liu, M., Zhu, J., Liu, G., Tao, A., Kautz, J. & Catanzaro, B. Video-to-video Synthesis. *NIPS* (2018).
10. Portenier, T., Hu, Q., Szabó, A., Bigdeli, S. A., Favaro, P. & Zwicker, M. FaceShop: Deep Sketch-based Face Image Editing. *ACM Transactions on Graphics (SIGGRAPH)* **37** (2018).
11. Lombardi, S., Saragih, J., Simon, T. & Sheikh, Y. Deep Appearance Models for Face Rendering. *ACM Transactions on Graphics (SIGGRAPH)* **37** (2018).

12. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X. & Huang, T. S. Generative Image Inpainting with Contextual Attention. *IEEE International Conference on Computer Vision* (2018).

13. Sankaranarayanan, S., Balaji, Y., Jain, A., Lim, S. N. & Chellappa, R. Learning from Synthetic Data: Addressing Domain Shift for Semantic Segmentation. *IEEE International Conference on Computer Vision* (2018).

14. Romero, A., Arbeláez, P., Gool, L. & Timofte, R. SMIT: Stochastic Multi-label Image-to-image Translation. *arXiv:1812.03704* (2018).

15. Chen, Y., Li, W., Chen, X. & Gool, L. Learning Semantic Segmentation from Synthetic Data: a Geometrically Guided Input-output Adaptation Approach. *IEEE International Conference on Computer Vision* (2019).

16. de Curtó, J. & Duvall, R. Cycle-consistent Generative Adversarial Networks for Neural Style Transfer Using Data from Chang'E-4. *arXiv:2011.11627* (2020).

17. Allan, M., Wong, U., Furlong, P. M., Rogg, A., McMichael, S., Welsh, T., Chen, I., Peters, S., Gerkey, B., Quigley, M., Shirley, M., Deans, M., Cannon, H. & Fong, T. Planetary Rover Simulation for Lunar Exploration Missions. *IEEE Aerospace Conference* (2019).

18. Wong, U., Nefian, A., Edwards, L., Buoyssounouse, X., Furlong, P. M., Deans, M. & Fong, T. POLAR (Polar Optical Lunar Analog Reconstruction) Stereo Dataset. *NASA Ames Research Center* (2017).

19. Vayugundla, M., Steidle, F., Smisek, M., Schuster, M. J., Bussmann, K. & Wedler, A. Datasets of Long Range Navigation Experiments in a Moon Analogue Environment on Mount Etna. *International Symposium on Robotics* (2018).

20. Zhang, Z. B., Zuo, W., Zeng, X. G., Gao, X. Y. & Ren, X. The Scientific Data and Its Archiving from Chang'E 4 Mission. *4th Planetary Data Workshop* (2019).

21. Ledergerber, A., Hamer, M. & D'Andrea, R. A Robot Self-localization System using One-way Ultra-wideband Communication. *IEEE International Conference on Intelligent Robots and Systems (IROS)* (2015).

22. Mueller, M. W., Hamer, M. & D'Andrea, R. Fusing Ultra-wideband Range Measurements with Accelerometers and Rate Gyroscopes for Quadrocopter State Estimation. *ICRA* (2015).

23. Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M. A. & Al-Khalifa, H. S. Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances. *Sensors* (2016).

24. Xu, H., Wang, L., Zhang, Y., Qiu, K. & Shen, S. Decentralized Visual-Inertial-UWB Fusion for Relative State Estimation of Aerial Swarm. *ICRA* (2020).

25. Kingma, D. P. & Welling, M. Auto-encoding Variational Bayes. *ICLR* (2014).

26. van den Oord, A., Kalchbrenner, N. & Kavukcuoglu, K. Pixel Recurrent Neural Networks. *ICML* (2016).

27. Odena, A., Olah, C. & Shlens, J. Conditional Image Synthesis with Auxiliary Classifier GANs. *ICML* (2017).

28. Antoniou, A., Storkey, A. & Edwards, H. Data Augmentation Generative Adversarial Networks. *ICLR* (2018).

29. Wang, X. & Gupta, A. Generative Image Modeling Using Style and Structure Adversarial Networks. *EUROPEAN Conference on Computer Vision* (2016).

30. Zhu, J., Krähenbühl, P., Shechtman, E. & Efros, A. A. Generative Visual Manipulation on the Natural Image Manifold. *EUROPEAN Conference on Computer Vision* (2016).

31. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X. & Metaxas, D. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. *IEEE International Conference on Computer Vision* (2017).

32. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. & Chen, X. Improved Techniques for Training GANs. *NIPS* (2016).

33. Mescheder, L., Nowozin, S. & Geiger, A. The Numerics of GANs. *NIPS* (2017).

34. Mescheder, L., Nowozin, S. & Geiger, A. Which Training Methods for GANs Do Actually Converge? *ICML* (2018).

35. Chen, Q. & Koltun, V. Photographic Image Synthesis with Cascaded Refinement Networks. *IEEE International Conference on Computer Vision* (2017).

36. Dosovitskiy, A. & Brox, T. Generating Images with Perceptual Similarity Metrics Based on Deep Networks. *NIPS* (2016).

37. Zhao, J., Mathieu, M. & LeCun, Y. Energy-based Generative Adversarial Networks. *ICLR* (2017).

38. Karras, T., Aila, T., Laine, S. & Lehtinen, J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *ICLR* (2018).

39. Wei, X., Gong, B., Liu, Z., Lu, W. & Wang, L. Improving the Improved Training of WASSERSTEIN GANs: a Consistency Term and Its Dual Effect. *ICLR* (2018).

40. Brock, A., Donahue, J. & Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *ICLR* (2019).

41. Song, Y. & Ermon, S. Generative Modeling by Estimating Gradients of the Data Distribution. *NIPS* (2019).

42. Song, Y. & Ermon, S. Improved Techniques for Training Score-based Generative Models. *NIPS* (2020).

43. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A. & Tran, D. Image Transformer. *ICML* (2018).

44. Park, T., Efros, A. A., Zhang, R. & Zhu, J. Contrastive Learning for Unpaired Image-to-Image Translation. *EUROPEAN Conference on Computer Vision* (2020).

45. Gatys, L. A., Ecker, A. S. & Bethge, M. Image Style Transfer Using Convolutional Neural Networks. *IEEE International Conference on Computer Vision* (2016).

46. Gatys, L. A., Bethge, M., Hertzmann, A. & Shechtman, E. Preserving Color in Neural Artistic Style Transfer. *arXiv:1606.05897* (2016).

47. Ulyanov, D., Lebedev, V., Vedaldi, A. & Lempitsky, V. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. *ICML* (2016).

48. Schneider, T., Dymczyk, M., Fehr, M., Egger, K., Lynen, S., Gilitschenski, I. & Siegwart, R. Maplab: an Open Framework for Research in Visual-inertial Mapping and Localization. *IEEE Robotics and Automation Letters* (2018).

49. Usenko, V., Demmel, N., Schubert, D., Stückler, J. & Cremers, D. Visual-inertial Mapping with Non-linear Factor Recovery. *IEEE Robotics and Automation Letters* (2019).

50. Badrinarayanan, V., Kendall, A. & Cipolla, R. SegNet: a Deep Convolutional Encoder-decoder Architecture for Image Segmentation. *IEEE International Conference on Computer Vision* (2015).

51. Chen, L., Papandreou, G., Schroff, F. & Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv:1706.05587* (2017).

52. He, K., Gkioxari, G., Dollár, P. & Girshick, R. MASK R-CNN. *IEEE International Conference on Computer Vision* (2017).

53. Bolme, D. S., Beveridge, J. R., Draper, B. A. & Lui, Y. M. Visual Object Tracking Using Adaptive Correlation Filters. *IEEE International Conference on Computer Vision* (2010).

54. Girshick, R. FAST R-CNN. *IEEE International Conference on Computer Vision* (2015).

55. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You Only Look Once: Unified, Real-time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition* (2016).

56. de Curtó, J. & Duvall, R. Vulcan Centaur: towards End-to-end Real-time Perception in Lunar Rovers. *arXiv:2011.15104* (2020).

57. Murthy, K., Saryazdi, S., Iyer, G. & Paull, L. gradslam: Dense SLAM Meets Automatic Differentiation. *ICRA* (2020).

58. Newcombe, R., Lovegrove, S. & Davison, A. DTAM: Dense Tracking and Mapping in Real-time. *IEEE International Conference on Computer Vision* (2011).

59. Engel, J., Schöps, T. & Cremers, D. LSD-SLAM: Large-scale Direct Monocular SLAM. *EUROPEAN Conference on Computer Vision* (2014).

60. Engel, J., Koltun, V. & Cremers, D. Direct Sparse Odometry. *T-PAMI* (2017).

61. Snavely, N., Seitz, S. M. & Szeliski, R. Photo Tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics (SIGGRAPH)* **25** (2006).

62. Agarwal, S., Snavely, N., Simon, I., Seitz, S. M. & Szeliski, R. Building Rome on a Day. *IEEE International Conference on Computer Vision* (2009).

63. Tateno, K., Tombari, F., Laina, I. & Navab, N. CNN-SLAM: Real-time Dense Monocular SLAM with Learned Depth Prediction. *IEEE Conference on Computer Vision and Pattern Recognition* (2017).

64. Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S. & Davison, A. CODESLAM - Learning a Compact, Optimisable Representation for Dense Visual SLAM. *IEEE Conference on Computer Vision and Pattern Recognition* (2018).

65. Z. Teed, J. D. DeepV2D: Video to Depth with Differentiable Structure from Motion. *ICLR* (2020).

66. Graham, B. & Novotny, D. RidgeSfM: Structure from Motion via Robust Pairwise Matching under Depth Uncertainty. *3DV* (2020).

67. Ono, Y., Trulls, E., Fua, P. & Yi, K. M. LF-Net: Learning Local Features from Images. *NIPS* (2018).

68. DeTone, D., Malisiewicz, T. & Rabinovich, A. SuperPoint: Self-supervised Interest Point Detection and Description. *IEEE Conference on Computer Vision and Pattern Recognition* (2018).

69. Hartley, R. & Zisserman, A. *Multiple View Geometry in Computer Vision* 2nd Edition (CAMBRIDGE University Press, 2004).

70. Tang, C. & Tan, P. BA-Net: Dense Bundle Adjustment Network. *ICLR* (2019).

71. Yang, N., von Stumberg, L., Wang, R. & Cremers, D. D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry. *IEEE Conference on Computer Vision and Pattern Recognition* (2020).

72. Godard, C., Mac Aodha, O. & Brostow, G. Unsupervised Monocular Depth Estimation with Left-right Consistency. *IEEE Conference on Computer Vision and Pattern Recognition* (2017).

73. Zhou, T., Brown, M., Snavely, N. & Lowe, D. Unsupervised Learning of Depth and Ego-motion from Video. *IEEE Conference on Computer Vision and Pattern Recognition* (2017).

74. Yin, Z. & Shi, J. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. *IEEE Conference on Computer Vision and Pattern Recognition* (2018).

75. Sünderhauf, N., Pham, T. T., Latif, Y., Milford, M. & Reid, I. Meaningful Maps With Object-oriented Semantic Mapping. *IEEE International Conference on Intelligent Robots and Systems (IROS)* (2017).

76. McCormac, J., Clark, R., Bloesch, M., Davison, A. & Leutenegger, S. Fusion++: Volumetric Object-level SLAM. *3DV* (2018).

77. Sucar, E., Wada, K. & Davison, A. NODESLAM: Neural Object Descriptors for Multi-view Shape Reconstruction. *3DV* (2020).

78. Arbeláez, P., Maire, M., Fowlkes, C. & Malik, J. Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions (TPAMI)* **33** (2011).

79. Arbeláez, P., Pont-Tuset, J., Barron, J. T., Marques, F. & Malik, J. Multiscale Combinatorial Grouping. *IEEE Conference on Computer Vision and Pattern Recognition* (2014).

80. Hariharan, B., Arbeláez, P., Girshick, R. & Malik, J. Hypercolumns for Object Segmentation and Fine-grained Localization. *IEEE Conference on Computer Vision and Pattern Recognition* (2015).

81. Mostajabi, M., Yadollahpour, P. & Shakhnarovich, G. Feedforward Semantic Segmentation with Zoom-out Features. *IEEE Conference on Computer Vision and Pattern Recognition* (2015).

82. Yu, F. & Koltun, V. Multi-scale Context Aggregation by Dilated Convolutions. *ICLR* (2016).

83. Chen, L., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. Encoder-decoder with Atrous Separable Convolution for Semantic Image Segmentation. *EUROPEAN Conference on Computer Vision* (2018).

84. Chen, L., Hermans, A., Papandreou, G., Schroff, F., Wang, P. & Adam, H. MaskLab: Instance Segmentation by Refining Object Detection with Semantic and Direction Features. *IEEE Conference on Computer Vision and Pattern Recognition* (2018).

85. Ren, S., He, K., Girshick, R. & Sun, J. FASTER R-CNN: Towards Real-time Object Detection with Region Proposal Networks. *NIPS* **28** (2015).

86. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. & Berg, A. C. SSD: Single Shot Multibox Detector. *EUROPEAN Conference on Computer Vision* (2016).

87. Redmon, J. & Farhadi, A. YOLO9000: Better, Faster, Stronger. *IEEE Conference on Computer Vision and Pattern Recognition* (2017).

88. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S. & Murphy, K. Speed/Accuracy Trade-offs for Modern Convolutional Object Detectors. *IEEE Conference on Computer Vision and Pattern Recognition* (2017).

89. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A. & Torralba, A. Semantic Understanding of Scenes through the ADE20K Dataset. *IJCV* (2018).

90. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B. & Belongie, S. Feature Pyramid Networks for Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition* (2017).

91. Xiao, T., Liu, Y., Zhou, B., Jiang, Y. & Sun, J. Unified Perceptual Parsing for Scene Understanding. *EUROPEAN Conference on Computer Vision* (2018).

92. Bao, W., Lai, W., Ma, C., Zhang, X., Gao, Z. & Yang, M. Depth-aware Video Frame Interpolation. *IEEE Conference on Computer Vision and Pattern Recognition* (2019).

93. Curtó, J. D., Zarza, I. C., Torre, F., King, I. & Lyu, M. R. High-resolution Deep Convolutional Generative Adversarial Networks. *arXiv:1711.06491* (2017).

94. Arjovsky, M. & Bottou, L. Towards Principled Methods for Training Generative Adversarial Networks. *ICLR* (2017).

95. Sønderby, C., Caballero, J., Theis, L., Shi, W. & Huszár, F. Amortised MAP Inference for Image Super-resolution. *ICLR* (2017).

96. Curtó, J. D., Zarza, I. C., Yang, F., Smola, A., Torre, F., Ngo, C. & Gool, L. McKernel: a Library for Approximate Kernel Expansions in Log-linear Time. *arXiv:1702.08159* (2017).

97. Curtó, J. D., Zarza, I. C., Kitani, K. & Lyu, R. Doctor of Crosswise: Reducing Over-parametrization in Neural Networks. *arXiv:1905.10324* (2017).

98. de Zarzà, I. *A Unifying Theory of Learning: DL Meets Kernel Methods* PhD thesis (ETH Zürich, 2021).

99. Xiong, X. & Torre, F. Supervised Descent Method and its Application to Face Alignment. *CVPR* (2013).

100. Klambauer, G., Unterthiner, T., Mayr, A. & Hochreiter, S. Self-normalizing Neural Networks. *NIPS* (2017).

101. Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ICML* (2015).

102. Arjovsky, M., Chintala, S. & Bottou, L. WASSERSTEIN Generative Adversarial Networks. *ICML* (2017).

103. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. Improved Training of WASSERSTEIN GANs. *NIPS* (2017).

104. Denton, E., Chintala, S., Szlam, A. & Fergus, R. Deep Generative Image Models Using a LAPLACIAN Pyramid of Adversarial Networks. *NIPS* **28** (2015).

105. Kingma, D. & Ba, J. Adam: a Method for Stochastic Optimization. *ICLR* (2015).

106. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. GANs Trained by a Two Time-scale Update Rule Converge to a Local NASH Equilibrium. *NIPS* **30** (2017).

107. Wu, J., Huang, Z., Thoma, J., Acharya, D. & Gool, L. WASSERSTEIN Divergence for GANs. *EUROPEAN Conference on Computer Vision* (2018).

108. Curtó, J. D., Zarza, I. C., Smola, A. & Gool, L. Segmentation of Objects by Hashing. *arXiv:1702.08160* (2017).

109. Charikar, M. Similarity Estimation Techniques from Rounding Algorithms. *STOC* (2002).

110. Everingham, M., Gool, L., Williams, C., Winn, J. & Zisserman, A. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV* **88** (2010).

111. Hariharan, B., Arbeláez, P., Girshick, R. & Malik, J. Simultaneous Detection and Segmentation. *EUROPEAN Conference on Computer Vision* (2014).

112. Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C. & Murphy, K. Towards Accurate Multi-person Pose Estimation in the Wild. *IEEE Conference on Computer Vision and Pattern Recognition* (2017).

113. Acuna, D., Ling, H., Kar, A. & Fidler, S. Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++. *IEEE Conference on Computer Vision and Pattern Recognition* (2018).

114. Zoph, B. & Le, Q. Neural Architecture Search with Reinforcement Learning. *ICLR* (2017).

115. Zoph, B., Vasudevan, V., Shlens, J. & Le, Q. V. Learning Transferable Architectures for Scalable Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition* (2018).

116. Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L., Fei-Fei, L., Yuille, A., Huang, J. & Murphy, K. Progressive Neural Architecture Search. *EUROPEAN Conference on Computer Vision* (2018).

117. Pham, H., Guan, M., Zoph, B., Le, Q. & Dean, J. Efficient Neural Architecture Search via Parameters Sharing. *ICML* (2018).

118. Li, L. & Talwalkar, A. Random Search and Reproducibility for Neural Architecture Search. *ICML* (2019).

119. Chen, L., Collins, M. D., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H. & Shlens, J. Searching for Efficient Multi-Scale Architectures for Dense Image Prediction. *NIPS* (2018).

120. Qi, C. R., Su, H., Mo, K. & Guibas, L. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition* (2017).

121. Qi, C. R., Yi, L., Su, H. & Guibas, L. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *NIPS* (2017).

122. Qi, C. R., Liu, W., Wu, C., Su, H. & Guibas, L. Frustum PointNets for 3D Object Detection from RGB-D Data. *IEEE Conference on Computer Vision and Pattern Recognition* (2018).

123. Chen, X., Ma, H., Wan, J., Li, B. & Xia, T. Multi-view 3D Object Detection Network for Autonomous Driving. *IEEE Conference on Computer Vision and Pattern Recognition* (2017).

124. Yang, B., Luo, W. & Urtasun, R. PIXOR: Real-time 3D Object Detection from Point Clouds. *IEEE Conference on Computer Vision and Pattern Recognition* (2018).

125. Luo, W., Yang, B. & Urtasun, R. Fast and Furious: Real Time End-to-end 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net. *IEEE Conference on Computer Vision and Pattern Recognition* (2018).

126. Liang, M., Yang, B., Wang, S. & Urtasun, R. Deep Continuous Fusion for Multi-sensor 3D Object Detection. *EUROPEAN Conference on Computer Vision* (2018).

127. Ku, J., Mozifian, M., Lee, J., Harakeh, A. & Waslander, S. Joint 3D Proposal Generation and Object Detection from View Aggregation. *IEEE International Conference on Intelligent Robots and Systems (IROS)* (2018).

128. Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J. & Beijbom, O. PointPillars: Fast Encoders for Object Detection from Point Clouds. *IEEE Conference on Computer Vision and Pattern Recognition* (2019).

129. Long, J., Shelhamer, E. & Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition* (2015).

130. Badrinarayanan, V., Kendall, A. & Cipolla, R. SegNet: a Deep Convolutional Encoder-decoder Architecture for Image Segmentation. *IEEE Transactions (TPAMI)* **39** (2017).

131. Girshick, R., Donahue, J., Darrell, T. & Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition* (2014).

132. Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks For Large-scale Image Recognition. *ICLR* (2015).

133. Indyk, P. & Motwani, R. Approximate Nearest Neighbors: towards Removing the Curse of Dimensionality. *STOC* (1998).

134. Gionis, A., Indyk, P. & Motwani, R. Similarity Search in High Dimensions via Hashing. *VLDB* (1999).

135. Shakhnarovich, G., Viola, P. & Darrell, T. Fast Pose Estimation with Parameter-sensitive Hashing. *IEEE International Conference on Computer Vision* (2003).

136. Shakhnarovich, G. *Learning Task-specific Similarity* PhD thesis (MIT, 2005).

137. Andoni, A., Indyk, P., Laarhoven, T., Razenshteyn, I. & Schmidt, L. Practical and Optimal LSH for Angular Distance. *NIPS* **1** (2015).

5

PUBLICATIONS

Articles:

1. de Curtò, J. & Duvall, R. Cycle-consistent Generative Adversarial Networks for Neural Style Transfer Using Data from Chang'E-4. *arXiv:2011.11627* (2020).
2. de Curtò, J. & Duvall, R. Vulcan Centaur: towards End-to-end Real-time Perception in Lunar Rovers. *arXiv:2011.15104* (2020).
3. Curtò, J. D., Zarzà, I. C., Torre, F., King, I. & Lyu, M. R. High-resolution Deep Convolutional Generative Adversarial Networks. *arXiv:1711.06491* (2017).
4. Curtò, J. D., Zarzà, I. C., Smola, A. & Gool, L. Segmentation of Objects by Hashing. *arXiv:1702.08160* (2017).
5. Curtò, J. D., Zarzà, I. C., Yang, F., Smola, A., Torre, F., Ngo, C. & Gool, L. McKernel: a Library for Approximate Kernel Expansions in Log-linear Time. *arXiv:1702.08159* (2017).
6. Curtò, J. D., Zarzà, I. C., Kitani, K., King, I. & Lyu, R. Doctor of Crosswise: Reducing Over-parametrization in Neural Networks. *arXiv:1905.10324* (2017).

6

CURRICULUM VITAE

GENERAL INFORMATION

J. de Curtò i DíAz.

E-mail: c@decurto.tw
Webpage: www.decurto.tw
Phone: +1 (412) 407-6797.

Nationality: Catalunya (Regne d'Espanya).

CAREER (SELECTED)

Iris Lunar Rover.
ETH Zürich. **Carnegie Mellon**. Pittsburgh.
BRAIN Team. Co-lead.
September 2020 - till now.

Founding member of a small research unit that works at the forefront of research in space exploration in the context of planetary rovers.

Iris Lunar Rover.
Carnegie Mellon. Pittsburgh.
Teleoperations. Localization and Environment Reconstruction Engineer.
September 2020 - December 2020.

Integral part of the team that is developing a SLAM and VIO solution for the rover that will be deployed on the Moon in 2021.

Distinctions: Award for most progress made. Hackhaton. 14-15 November 2020.

Universitat Rovira i Virgili. Tarragona.
Research Scholar. September 2018 - September 2020.
Group of Intelligent Robotics and Computer Vision.

The Chinese University of Hong Kong (CUHK). Hong Kong.
Research Assistant. October 2017 - February 2018.
Department of Computer Science and Engineering.

Carnegie Mellon. Pittsburgh.
Research Assistant. June 2017 - July 2017.
School of Computer Science. Robotics.

City University of Hong Kong. Hong Kong.
Senior Research Assistant. January 2017 - May 2017.
Department of Electrical Engineering.

Challenger Deep, Ltd. Hong Kong.
Research Director and Technical Lead. Co-founder. May 2016 - November 2017.
Top state-funded incubator program. Cyberport.

ETH Zürich (ETHZ). Zürich.
Graduate Research Assistant. April 2015 - September 2016.
Laboratory of Computer Vision.

City University of Hong Kong. Hong Kong.
Research Associate. July 2015 - August 2015.
Department of Computer Science.

Carnegie Mellon. Pittsburgh.
Research Associate I. June 2014 - August 2014.
School of Computer Science. Robotics.

EDUCATION (SELECTED)

ETH Zürich (ETHZ). Zürich.

Doctor of Science. April 2015 -.
Laboratory of Computer Vision.

Carnegie Mellon. Pittsburgh.

Master of Science. May 2014 - February 2015.
School of Computer Science. ML Department and Robotics.

Thesis: A Library for Fast Kernel Expansions with Applications to Computer Vision and Deep Learning.

Academic Distinctions:

- HKPFS 2014-2015
  Highly prestigious HK Fellowship to pursue PhD studies at HKUST during academic years 2015-2018. Department of Computer Science and Engineering. Declined.
- SINGA PhD Fellowship 2014.
  Highly prestigious international award to pursue PhD studies at NTU Singapore during academic years 2015-2019. School of Electrical and Electronic Engineering. Declined.

City University of Hong Kong. Hong Kong.

Master of Science. September 2013 - February 2015.
Department of Electrical Engineering.
GPA: 4.12 (0-4 scale).
Classification of Award: Distinction (1/+100).

Academic Distinctions:

- Top Achiever 2015.
  Award for being the first with regard to academic performance.
- MS Internship Sponsorship 2014.
  Award for top performing students. Robotics. Carnegie Mellon. Pittsburgh.
- MS Entrance Scholarship 2013/2014.
  Award given to the three most excellent students.

Universitat Autònoma de Barcelona. Cerdanyola del Vallès (Barcelona).

5-year Degree in Engineering of Telecommunication, Second Cycle. 2011 - 2013.
Specialization in Communications, Signal Processing and Microwave Engineering.
School of Engineering.

Thesis: Construction and Performance of Network Codes.
Grade: Excellent. First Class with Distinction.

Universitat Politècnica de Catalunya (UPC). Barcelona.

5-year Degree in Engineering of Telecommunication, First Cycle. 2006 - 2009.

University Entrance Examination.
Average Grade: 9.22/10. First Class with Distinction.

Academic Distinctions:

- First year scholarship for university studies. Ministry of Education.
  This award is given to the top nationwide first year university students.
- First year scholarship for university studies. Caixa Manresa.
  This award is given to the top university entrance examination average grades in the region of Catalunya.
- University scholarship for an outstanding academic performance. Technological Baccalaureate. Government of Salou. This award is given to the top students in each graduation year by the local authority.

Technological Baccalaureate. 2004 - 2006.
Average Grade: 9.7/10. First Class Degree and Honorary Scholarship.

Academic Distinctions:

- Outstanding Thesis of Research. Development and Design of a Virtual Shop in Visual Basic on the .NET Framework.
- Outstanding Curriculum.

PUBLICATIONS

De Curtò i DíAz and Duvall.
Vulcan Centaur: towards end-to-end real-time perception in lunar rovers.
arxiv.org/pdf/2011.15104

De Curtò i DíAz and Duvall.
Cycle-consistent Generative Adversarial Networks for Neural Style Transfer using data from Chang'E-4.
arxiv.org/pdf/2011.11627

Curtò, Zarzà, Kitani, King and Lyu.
Doctor of Crosswise: Reducing Over-parametrization in Neural Networks.
decurto.tw/c/doctor_of_crosswise.pdf

Curtò, Zarzà, Torre, King and Lyu.
High-resolution Deep Convolutional Generative Adversarial Networks.
decurto.tw/c/hdcgan.pdf

Curtò, Zarzà, Smola and Gool.
Segmentation of Objects by Hashing.
decurto.tw/c/c_and_z.pdf

Curtò, Zarzà, Yang, Smola, Torre, Ngo and Gool.
McKernel: A Library for Approximate Kernel Expansions in Log-linear Time.
decurto.tw/c/mckernel.pdf

De Curtò i DíAz, De Zarzà i Cubero and Vázquez.
Secure Network Coding: Overview and State-of-the-art.
Universitat Autònoma de Barcelona. Cerdanyola del Vallès (Barcelona). 2012.
https://hal.archives-ouvertes.fr/hal-03168605/document

De Curtò i DíAz, Moreno, Torrellas, Bofill and Muñoz.
Dear New Student: a Comparison between a Frontal and an Active Approach.
ALE 2007. Toulouse.

DISSERTATIONS

Master of Science.
A Library for Fast Kernel Expansions with Applications to Computer Vision and Deep Learning.
Supervisors: Smola, De la Torre and Ngo.
Carnegie Mellon. Pittsburgh. 2014.
decurto.tw/c/decurto.pdf
decurto.tw/c/slides_decurto.pdf

5-year Degree in Engineering of Telecommunication.
Construction and Performance of Network Codes.
Supervisor: Vázquez.
Universitat Autònoma de Barcelona. Cerdanyola del Vallès (Barcelona). 2013.
https://hal.archives-ouvertes.fr/tel-03167326/document
decurto.tw/c/slides_pfc_decurto.pdf

LANGUAGES

English -

TOEFL Internet Based test. 26-11-2016. **Score 114/120.**

First Certificate in English. December 2005. **Grade: A.**

WORK EXPERIENCE

CELLS ALBA Synchrotron Facility. Cerdanyola del Vallès (Barcelona).
Research Scientist. April 2010 - June 2010.

CELLS ALBA Synchrotron Facility. Cerdanyola del Vallès (Barcelona).
Internship. January 2010 - February 2010.

Universitat Politècnica de Catalunya (UPC). Barcelona.
Teaching Assistant. Departament de Teoria del Senyal i Comunicacions. September 2009 - December 2009.

CELLS ALBA Synchrotron Facility. Cerdanyola del Vallès (Barcelona).
Internship. July 2009 - September 2009.

Universitat Politècnica de Catalunya (UPC). Barcelona.
Teaching Assistant. Departament d'Arquitectura de Computadors. 2008 - 2009.

Universitat Politècnica de Catalunya (UPC). Barcelona.
Teaching Assistant. Departament d'Arquitectura de Computadors. 2006 - 2007.

CoRL. Virtual Conference. Formerly in Cambridge. 16/11 - 18/11. 2020.

Attendee.

ROS World. Virtual Conference. 12/11. 2020.

Attendee.

ICML 2020. Virtual Conference. Formerly in Vienna. 12/07 - 18/07. 2020.

Attendee.

Robotics: Science and Systems. Virtual Conference. Formerly in Corvallis. 12/07 - 16/07. 2020.

Attendee.

SIAM Conference on Imaging Science. Virtual Conference. Formerly in Toronto. 06/07 - 17/07. 2020.

Attendee.

IEEE International Symposium on Information Theory. Virtual Symposium. Formerly in Los Angeles. 21/06 - 26/06. 2020.

Attendee.

IEEE Conference on Computer Vision and Pattern Recognition. Virtual Conference. Formerly in Seattle. 14/06 - 19/06. 2020.

Attendee.

IEEE International Conference on Communications. Virtual Conference. Formerly in Dublin. 07/06 - 11/06. 2020.

Attendee.

IEEE International Conference on Robotics and Automation. Virtual Conference. Formerly in Paris. 31/05 - 04/06. 2020.

Attendee.

SIAM Conference on Mathematics of Data Science. Virtual Conference. Formerly in Cincinnati. 04/05 - 30/06. 2020.

Attendee.

IEEE International Conference on Acoustics, Speech, and Signal Processing. Virtual Conference. Formerly in Barcelona. 04/05 - 08/05. 2020.

Attendee.

ICLR 2020. Virtual Conference. Formerly in Addis Ababa. 26/04 - 01/05. 2020.

Attendee.

Oral presentation at the Social Virtual Event on Open source tools and practices in state-of-the-art DL research.

Slides with Q&A annotations: www.decurto.tw/c/iclr2020_DeCurto.pdf

First European Training School in Network Coding: Random Network Coding and Designs over $GF(q)$. IEEE Information Theory Society. Universitat Autònoma de Barcelona. Cerdanyola del Vallès (Barcelona). 04/02 - 08/02. 2013.

From designs over $GF(q)$ to applications of networking: a cross-road for mathematics, computer science and engineering.

Attendee and Volunteer.

ESOF 2008. Barcelona. 18/07 - 22/07. 2008.

Scientific Volunteer.

EXTRACURRICULAR ACTIVITIES

Program of Open Mentoring. Department of Computer Science. The University of Hong Kong. 2014 - 2018.

MS Class Representative, cohort 2013. Department of Electrical Engineering. City University of Hong Kong. 2013 - 2014.

Course in Investment and Financial Markets. Technical Analysis and Risk Management. Barcelona. 23 May 2011 - 26 May 2011.

Course in Investment and Financial Markets. Barcelona. 18 April 2011 - 21 April 2011.

Competition of Entrepreneurship. EMPRÈN UPC. 1st Edition. Universitat Politècnica de Catalunya (UPC). Finalist project awarded with honorable mention and 1000 euros. Barcelona. 14 March 2011 - 14 June 2011.

PROGRAMMING

C, C++, Java, MATLAB, Python, HTML, VHDL and Assembly.

SOFTWARE

LaTeX, Maple, PSpice and ADS.