Drone recognition with Deep Learning

Julien Gérard

université
PARIS-SACLAY

Drone Recognition by Deep Learning

Sélection et Reconnaissance de Drones par Deep Learning

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 Science et Technologies
de l'Information et de la Communication (STIC)
Spécialité de doctorat : Informatique
Graduate School : Informatique et sciences du numérique,
Référent : CentraleSupélec

Thèse préparée dans le laboratoire interdisciplinaire des sciences du numérique (Université Paris-Saclay, CNRS), équipe GaLac (LRI), sous la direction de Joanna Tomasik, le co-encadrement de Christèle Morisseau, le co-encadrement de Arpad Rimmel et le co-encadrement de Gilles Vieillard

Thèse soutenue à Gif-sur-Yvette, le 16 février 2022, par

Julien Gérard

Composition du jury

Devan Sohier Professeur, UVSQ, Université Paris Saclay	Président
Tristan Cazenave Professeur, Université Paris-Dauphine	Rapporteur & Examinateur
Laurent Ferro-Famil Professeur, Université de Rennes 1	Rapporteur & Examinateur
Alexandre Benoit Maître de conférences, Université Savoie Mont-Blanc	Examinateur
Mihai Datcu Professeur, German Aerospace Centre DLR/CNAM	Examinateur
Olivier Schwander Maître de conférences, LIP6, Université Pierre et Marie Curie	Examinateur
Joanna Tomasik Professeur, CentraleSupélec, Université Paris Saclay	Directeur de thèse

THESE DE DOCTORAT
NNT : 2022UPASG002

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Sélection et Reconnaissance de Drones par Deep Learning
Mots clés : drones, micro-Doppler, réseaux de neurones, apprentissage profond

Résumé : Cette thèse étudie la reconnaissance de signaux radar micro-Doppler de drones par des méthodes d'apprentissage profond (Deep Learning). Le phénomène micro-Doppler est un ensemble de modulations de fréquence créé par les mouvements internes de la cible observée.

Tout d'abord, nous avons analysé les différentes données existantes : simulations, données collectées accessibles. Nous examinons alors les limites de ces données. Afin de s'en affranchir, nous avons effectué une campagne de mesure adaptée.

Une fois les données collectées, nous avons étudié l'impact des différents espaces de représentation afin de proposer à la communauté un format standard pour un usage Deep Learning.

Nous abordons alors un problème majeur en radar : le manque de données. Nous explorons alors la piste de l'augmentation de données par des GANs. Nous proposons une mesure de la qualité de ces algorithmes basés sur des critères d'utilités de la génération et non du réalisme de celle-ci. Avec cette mesure, nous avons observé une amélioration statistiquement significative des performances de classification grâce aux signaux générées par GAN.

Encouragé par ce résultat, nous implémentons des GANs plus avancés combinant vérité terrain et signaux réels. Nos expériences nous permettent alors d'atteindre les performances précédentes. Actuellement, nous identifions des axes de résolutions, que nous prévoyons de développer, pour les dépasser.

Title : Drone recognition by Deep Learning
Keywords : drone, micro-Doppler, neural network, Deep Learning

Abstract : We work on the recognition of radar micro-Doppler signals of drones thanks to Deep Learning tools. The micro-Doppler phenomenon consists in a frequency modulation set created by the intern movements of the observed target.

First, we analyze the different existing data : simulations, collected data available. We examine their limitations and carry out a measurement campaign to tackle them.

Once our data is collected, we study the impact of the different space representations to propose a standard format adapted for Deep Learning.

We continue our research on a major radar problem : the lack of data. Thus, we explore the data augmentation with GANs. We propose a measure of the quality of these algorithms based on utility criteria, and not on the realism of generated data. We observe a statistically significant improvement of classifications thanks to the signals generated by our GANs.

Encouraged by this result, we implement more advanced GANs conjugating ground truth and real data. As we identified possible resolution axes we currently develop them.

Table des matières

Physics, Acronyms - Notations

$\overrightarrow{x}$	An object $\overrightarrow{x}$ is a vector with x its associated norm ($x = \|\|\overrightarrow{x}\|\|$).
$\overrightarrow{u_x}$	For each vector $\overrightarrow{x}$, $\overrightarrow{u_x}$ is the associated unitary vector ($\overrightarrow{u_x} = \frac{\overrightarrow{x}}{\|\|\overrightarrow{x}\|\|}$).
$\overrightarrow{u_r}$, $\overrightarrow{u_\theta}$, $\overrightarrow{u_z}$	Cylindrical coordinate system.
blade	The flat rotating part of drone flying (radius). Do not confuse it with rotors. Each rotors may have several equally distributed blades.
c	Light speed ($c = 299\,792\,458\,m/s$).
CP	Cepstrum, details in Section 3.1.1.
CVD	Cadence Velocity Diagram, details in Section 3.1.1.
drone	Unless otherwise specified, a drone is a 'mini' drone as explained in [17].
f_0	Frequency of the emitted signal.
f_m	Rotation frequency of rotor m.
FT	Fourier Transform.
FFT	Fast Fourier Transform.
i	Unit imaginary number ($i^2 = -1$).
IFFT	Inverse Fast Fourier Transform.
$\overrightarrow{k}$	Propagation vector. In our case (back-scattering), $k = \frac{2\pi f_0}{c}$.
M	Rotor number of a target.
N	Blade number (radius) per rotor. A drone has in the majority of cases two blades per rotor, a wind turbine three.
PRF	Pulse Repetition Frequency.
$\overrightarrow{R_0}$	Vector from the radar to the target cell.
RCS	Radar Cross-Section.
RPM	Rotation Per Minute.
$s(t)$	Signal acquired (temporal).
$S(f)$	Fourier transforme of the acquired signal (frequential).
SG	Spectrogram, details in Section 3.1.1.
SNR	Signal Noise Ratio.
SPWVD	Smooth Pseudo Wigner-Ville Distribution. Details in Section 3.1.1.

STFT	Short-time Fourier Transform.
$\overrightarrow{v}$	Drone speed.
$\overrightarrow{v_r}$	Radial component of the drone speed ($\overrightarrow{v_r} = \overrightarrow{v}\overrightarrow{u_k}$).
WSP	Weighted Spectrum, details in Section 3.1.1.
α	Azimuth of the radar.
α_p	Yaw of the target.
β	Elevation of the radar.
β_p	Pitch of the target.
λ	Wavelength of the emitted signal.
π	Half-perimeter of the unitary circle (3.14...).
ϕ	Phase of the signal.
ω_x	For each frequency f_x, ω_x is the associated pulsation : $\omega_x = 2\pi f_x$.

Neural Network, Acronyms - Notations

ACGAN	Auxiliary Classifier Generative Adversarial Network [43].
cGAN	conditional Adversarial Generative Network [40]
CNN	Convolutional Neural Network.
D	Discriminator. A neural network that discriminate synthetic and real data.
DCGAN	Deep Convolutional Neural Network, details in [48].
fs-InfoGAN	full-supervised Informative Generative Adversarial Network.
G	Generator. A neural network that create data as output
GAN	Generative Adversarial Network [22].
generated data	Data created by a generative algorithm, also called synthetic data.
GPD	Generator update (back-propagation) per Discriminator update. This hyperparameter can be use to balance a GAN.
hyperparameter	See metaparameter.
InfoGAN	Informative Generative Adversarial Network [11].
memory GAN	An unwanted effect which can occur during a GAN training. In this situation, the GAN reproduces the examples of the training datasets without adding anything new.
metaparameter	Also called hyperparameter. A parameter fixed by the user to setup an algorithm such as the training setup of a neural network. It is fixed before learning and does not evolve thanks to the back-propagation contrary to the parameters corresponding to the weights of the network.
MLP	Multi-Layers Perceptron.
MNIST	The Modified National Institute of Standards and Technology dataset is composed of labeled numbers from zero to nine [36]. Considered as the "hello world" dataset of machine learning.
mode dropping	An unwanted effect which can occur during a GAN training. In this situation, the GAN creates only synthetic data of a part of the distribution.
N	Used to evaluate GAN algorithm, amount of classifier training with combined datasets. In all the results presented in this manuscript $N \geq 100$.
NN	Neural Network.
parameter	It corresponds to the current state of the network. Contrary to meta-parameters, it evolves across learning during the

	back-propagation. Thanks to this evolution, the network approximates gradually the desired function. It is also called weight.
K	Used to evaluate GAN algorithm, amount of generator training. Generally, $10 \leq K \leq 50$.
real data	Data collected from real-world measurement.
RGB	Red Blue Green.
simulated data	Data created by classic simulations : electromagnetic models as those defined in Section 2.1.2.
SVM	Support Vector Machine.
synthetic data	Data created by a generative algorithm. Also called generated data.
ss-InfoGAN	semi-supervised Informative Generative Adversarial Network [58].

1 - Introduction

For the protection of sensitive sites in a defense context, drones are one of the challenges of the last decade. Drones are unmanned aerial vehicles (UAVs). Their weights and dimensions vary, respectively, from tenths to several hundred kilograms and from tenths to several tens of meters. Their purpose depends on their dimension.

One drone category, the mini-drones [17], with dimensions from 0.5 to 1 meter and weight from one to ten kg, raises strong security threats. This category is big enough to carry out complex missions and small enough to be purchased easily by any individual. Besides, their performance and their load capacity continue to increase while their cost drops down.

A mini-drone can carry consequent and potentially dangerous payloads. For example, a DJI Phantom still flies with a payload of half a kilogram [46]. Moreover, the autonomy of such drones reaches one hour at a low altitude. For the handling aspect, the remote control needs few weeks of training but indicating way points for a planned trajectory is as easy as setting the GPS of a car. Due to these characteristics, the number of mini-drones increases in our everyday life which endangers public safety.

To protect sensitive sites, we have to recognize these commercial drones from a long distance (several kilometers) to evaluate the potential danger and set up an appropriate response. Currently, this problem is not solved which raises major concerns. In a study conducted in 2018 [45], the situation is summarized by the following sentence : *We have been extremely fortunate that there have been no terrorist attacks or loss of life to this date involving these commercially available drones*. For this reason, investments are made to propose technical solutions for recognizing the drone type from a long distance. This procedure is composed not only of the detection (the presence of the target) of the incoming target but also its identification (the drone classification) to evaluate the situation properly.

Numerous methods, based upon image (camera) or sound (microphone) principles have been proposed to reach this goal. However, as detailed in [17], the non-radar methods are inefficient even to detect mini-drones at few hundred meters, let aside to identify them.

The capacity of object perception by a radar, expressed as RCS (*Radar Cross Section*) is sufficiently high for commercial drones : a mini-drone can be detected from a distance of order of several kilometers. However, our problem is that the RCS alone does not solve the identification issue. In [50], the signal corresponding to three species of birds is compared to the signal of a mini-drone. In terms of RCS, the different radar signatures are comparable. Therefore, they do not differentiate drones from birds. Besides, the different models of drones cannot be distinguished with RCS either.

To identify a drone precisely, we need some supplementary information. The main candidate to play this role is the micro-Doppler effect. The micro-Doppler effect corresponds to frequency shifts created by internal movements of the target. We assume that these internal movements can be efficiently caught by the micro-Doppler effect and be sufficiently characteristic to identify the different targets.

Our goal is to study the classification of drones according to their radar profiles under the above-mentioned assumption, in other words, the identification of a detected target.

In order to investigate the classification possibilities, we used Deep Learning algorithms as their performance for image recognition has shown a high efficiency. First, we examined the possibilities to gather data in order to carry out our investigations. The micro-Doppler effect is captured by electromagnetic models which are at the heart of simulations. After the presentation of micro-Doppler effect on drones, we analyzed the theoretical and practical limits of simulations with the previously collected declassified NATO dataset provided by the working group SET245. Due to these limits, we continued with the details of the measurement campaign we carried out. We made preliminary observations on the collected data to validate its coherence and to highlight more precisely the problems reported in the literature. As we will explain, the latter justifies this study and the use of Deep Learning.

We kept up our work by determining a representation for micro-Doppler drone signals to obtain satisfactory classification results. Indeed, numerous formats are commonly used for the drone recognition problem. To choose the most adapted, we compared drone classification results according to each of them for the different use conditions we proposed. Conforming to the experiments conducted, the recommended format is a spectrum issued from long observations as its classification results are better for most criteria.

We continued with the data augmentation problem. Indeed, as the acquisition of a huge amount of data for radar signals is expensive and sometimes even impossible, we proposed to augment input data using GANs [22]. To begin with, we examined the most common GAN algorithms and their drawbacks such as their instability. Then, to conduct our experiments, we devised an evaluation method for data augmentation and a calibration method for GANs. Thanks to this study, we observed that the addition of generated data into the learning dataset leads to a significantly better quality of classification in comparison to the same training dataset alone. We emphasize that along with the interest of the radar community, our study is sufficiently general to be applied to other application domains.

Encouraged by this major result, we tried to improve it by generating more advanced synthetic micro-Doppler signals with the combination of Deep Learning algorithms and the use of pertinent ground truth. Indeed, as we had access to technical information such as the drone configuration during the measurement, we

tried to teach different GAN algorithms the relationship between the real signal and the drone state to make them generalize the observations by catching the physical laws beneath it. We experimented both the use of drone log information directly with an ACGAN [43] and the use of simulated signals with a pix2pix [27].

Our achievements in the field of drone recognition outlined above are described in this thesis, which is structured as follows.

In Chapter 2, we present the micro-Doppler effect with the electromagnetic models and their limitations together with an existing dataset to conclude with the measurement campaign we carried out.

Then, in Chapter 3, we determine a recommended format for micro-Doppler drone signals. We present the different formats studied and the neural network concept, followed by the use conditions we chose and our experimental protocol. We end with the results obtained with each format for the proposed use conditions.

Next, in Chapter 4, we treat the data augmentation problem with GAN algorithms. We start by depicting the GANs studied, followed by the evaluation method we developed and our experimental protocol. After a satisfactory calibration, we conclude with the results obtained.

Subsequently, in Chapter 5, we detail our investigations on data augmentation with the use of ground truth. We begin with the presentation of the ground truth, the domain generalization concept and, more specifically, the pix2pix algorithm. We follow with our experimental protocol to end with the results obtained.

The global conclusion of the main contributions of this PhD is provided in Chapter 6. Eventually, we summarize the most pertinent further research lines which our study as opened up.

2 - Creation of a Micro-Doppler Dataset

We introduce our first contribution to the drone identification problem. We explain the necessity to collect a large and realistic dataset which made us set up our own measurement campaign.

We start by explaining the micro-Doppler effect with electromagnetic models. These models are used in simulations which may be seen as a source of data, alternative to a real one. Simulated data is often used in the literature to study the micro-Doppler phenomenon. However, we are aware of the limitations of these models which leads to the necessity of producing large and realistic datasets. Eventually, we present the measurement campaign we carried out with its validation with the ground truth comparison.

2.1 . State of the Art

To introduce the electromagnetic micro-Doppler phenomenon, we settle the general radar equations, our measurement scenario, and the hypotheses arising from them. Then, we detail the micro-Doppler phenomenon.

The models described are based upon [39, 9, 10, 47]. As the micro-Doppler effect is not specific to drones, we outline that some of these articles do not deal with them. For instance, in the radar domain, the most anterior works concern mostly helicopters. However, as the micro-Doppler effect observed corresponds to rotating parts, those results remain relevant.

The notation used comes mostly from [10]. It is also summarized in the glossary (pages 6–7).

2.1.1 . Radar Signals and Doppler Effect

Scenario and Hypotheses

The scenario studied is the protection of a sensitive site against a drone menace. We consider that we have access to this site, the radar is thus set up there. The threat (a target) is going towards the radar.

The radar signal is transmitted in the environment towards a specific direction $\overrightarrow{u_k}_{\text{emission}}$. It is then reflected on objects (trees, drones, birds, cars, etc.) in different directions depending on the radar signal and the target type. The reflected signal from the direction $\overrightarrow{u_k}_{\text{reception}}$ is then acquired by the receptor.

We work under a set of hypotheses listed below :

Mono-static hypothesis The system is static and composed of one antenna which manages both emission (transmitter) and reception (receptor). We thus define $\overrightarrow{u_k} = \overrightarrow{u_k}_{\text{emission}} = -\overrightarrow{u_k}_{\text{reception}}$.

Far-field hypothesis The signal wavelength and the dimensions of the target are negligible compared to the radar-drone distance R_0. For this reason, the reflections from all parts of a target arrive from the same direction.

Direct trajectory hypothesis Only the direct signal reflection on the targets is considered by the receptor. Other target signal components due to several sequential reflections, denoted multipath (ground, wall, etc.) coming back to the receptor much later than the expected time and attenuated, are considered as a environmental noise even if they indirectly are a contribution from the target.

Reflector point hypothesis The targets are modeled by a set of reflector points. These points are not spatially resolved and do not interact. The signal of a combination of the reflector points is thus the sum of the signal of each point alone (**linearity hypothesis**). So, these points cannot hide each other. The reflection created by each point is equal along any direction (**isotopic hypothesis**). Their electromagnetic signatures denoted RCS (*Radar Cross Section*) are constant whatever their motion during the observation is.

These hypotheses induce the following restrictions :

— The defense of a sensitive site is not generally based on a single radar. More complex models based either on the collaboration of different systems or one large radar system with a set of transmitters and receptors (MIMO, *Multi Input Multi Output*) can be used to gather richer information.

— The electromagnetic models neglect certain physical phenomena. Some of these are observed on the signals collected during our measurement campaign and detailed in Section 2.3.2.

General Equation

The models are formulated for a sinusoidal waveform of the emitted signal. Similar results can be obtained for more complex emitted signals. For simplification, continuous equations are provided, in practice and for most radars, data is a sequence of discrete values regularly spaced.

We denote :

— c : light speed,

— λ : wavelength of the emitted signal,

— f_0 : carrier frequency of the emitted signal, $\lambda = \frac{c}{f_0}$,

— $k = \frac{2\pi f_0}{c}$: norm of the wavelength propagation vector, $\overrightarrow{k} = k\overrightarrow{u_k}$,

— $\overrightarrow{r(t)}$: radar-target distance vector,

— R_0 : initial radar-target distance,

— $s(t)$: received signal,

— A : amplitude of this received signal,

— t : time.

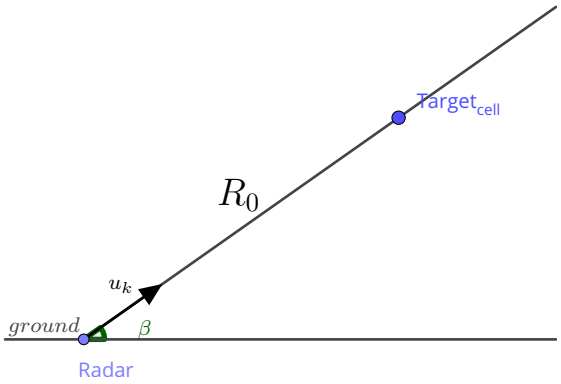


Figure 2.1 – Observation of a target at distance R_0 in the azimuthal plane.

$$\begin{aligned} s(t) &= Ae^{2i\pi f_0 t} e^{-ik\overrightarrow{r(t)}(\overrightarrow{u_k}_{\text{emission}} - \overrightarrow{u_k}_{\text{reception}})} \\ &= Ae^{2i\pi f_0 t} e^{-ik\overrightarrow{r(t)}(2\overrightarrow{u_k})}. \end{aligned} \tag{2.1}$$

The observation of the target is represented in Fig. 2.1. The $e^{2i\pi f_0 t}$ term comes from the carrier frequency. The reflection term is $e^{-ik\overrightarrow{r(t)}(2\overrightarrow{u_k})} = e^{-i\phi(t)}$ with

— $\phi(t) = k\overrightarrow{r(t)}(2\overrightarrow{u_k})$ which denotes the phase factor.

Doppler Phenomenon

The Doppler effect consists in a frequency shift due to the variation of the distance between the transmitter and the receptor during the observation time. In our case, the radar (the transmitter and the receptor together) is fixed. The trajectory of the signal is a round trip between the fixed radar and the target in motion. So, the Doppler effect appears due to the target motion between the consecutive signal acquisitions. An identical phenomenon would be observed if the radar moved and the target standed still. The crucial thing is the variation of the radar-target distance. Thanks to the mono-static hypothesis, a factor 2 occurs $(2\overrightarrow{u_k})$ as in Equation (2.1). We define :

— $f_D(t)$: frequency shift due to the Doppler effect :

$$f_D(t) = -\frac{1}{2\pi}\frac{d\phi(t)}{dt} = -\frac{2f_0}{c}\frac{d}{dt}\overrightarrow{r(t)}\overrightarrow{u_k}.$$

For a moving object, we denote :

— v_r : radial speed, $v_r = \overrightarrow{v}\overrightarrow{u_k}$,

— R_0 : initial radar-target distance vector,

— $\overrightarrow{r(t)}$: radar-target distance vector across time, $\overrightarrow{r(t)} = \overrightarrow{R_0} + t\overrightarrow{v}$

and thus the main Doppler effect equals to

$$f_D(t) = -\frac{2f_0 v_r}{c}.$$

Only the radial component v_r of the speed vector has an impact on the Doppler effect. Consequently, a target turning around the radar will not produce such a Doppler effect. Radars measuring only the main Doppler effect assume that v_r is large. This hypothesis is justified by the fact that the radar is located at the sensitive site to protect it. We expect targets to come toward this place which implies a strong radial speed.

We also point out that only the variation of the signal phase is interesting in our models. Thus, the amplitude terms A are secondary in the remainder and their formulæ will not be detailed.

2.1.2 . Micro-Doppler Phenomenon for Drones : Time Series

The micro-Doppler phenomenon is a specific kind of Doppler effect. The target is no longer considered as a single moving point but as an object with internal movements. Those movements induce their own Doppler effects. These phenomena, called micro-Doppler, are added to the main Doppler effect. The main part of the target, which is considered without internal movements, is called the target cell.

The strongest internal movements of a drone are produced by the blades of the rotors. As there is no standard convention, we call the rotor the whole turning part. A rotor is thus composed of a fixed center and several blades. One blade extremity is attached to the rotor center. Most drones have two blades per rotor.

For the simulation model presented below, other micro-Doppler effects such as vibrations (of the radar and the drone) are neglected even if they might impact the collected data [10]. They are considered as a part of the environmental noise.

The micro-Doppler effects can take other forms, depending on the nature of the target. Here, we introduce only the micro-Doppler effect for drones but similar models can be developed for other targets such as birds or humans. Even if it is not the signal of our interest, we highlight the use of micro-Doppler to identify

human gaits. All along with this manuscript, we cite articles using micro-Doppler for human gaits, particularly to study their analysis tools. At the origin, the micro-Doppler effect was observed for small internal movements, it thus corresponds to small (micro) variation around the Doppler effect. For drones, the micro-Doppler effect may often induce higher frequency shifts and amplitudes than the main Doppler effect due to high rotational speeds and the blade material. However, they are still called micro-Doppler as they are caused by small distance variations to the radar compared to the variations induced by the drone cell.

To understand our modeling, we use an incremental approach. We start by considering a single rotor composed of one blade represented by a single rotating point P to end with a complete drone composed of points.

Rotating Points : Simple Case

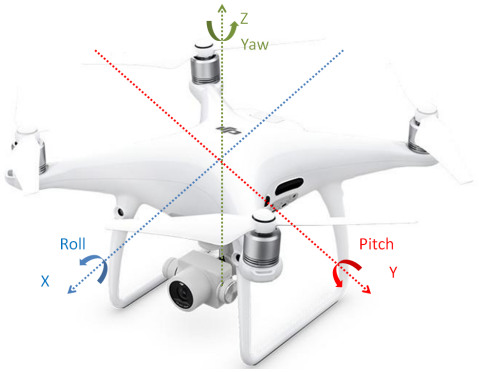


Figure 2.2 – Definition of yaw, pitch α_p, roll β_p in drone coordinates.

To begin with, we consider the influence of one point P rotating around its center $\text{Target}_{\text{cell}}$. The $\text{Target}_{\text{cell}}$ influence itself is not taken into account at the moment.

We also neglect for now the azimuth and elevation of the radar (a radar is well focused on the target) and the intern rotation axes of the target (roll, pitch, yaw ; see Figs. 2.2 and 2.3). These strong hypotheses result in the schema in Fig. 2.4.

This theoretical target is a moving rotating point P :

— l : radius of the intern movement,

— ω : rotation speed (rad/s) of the intern movement,

— θ_0 : initial phase of the intern movement,

— $\overrightarrow{v}$: main constant speed of the target,

— v_r : radial component of the speed vector $\overrightarrow{v}$,

— $\overrightarrow{r(t)}$: target-radar vector distance across time,

— R_0 : initial target-radar vector distance,

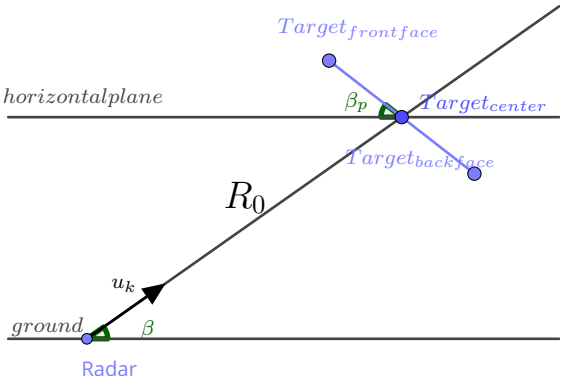


Figure 2.3 – Observation on the azimuth plane of a target with a non-zero observation angle between the rotation plane and the radar orientation.

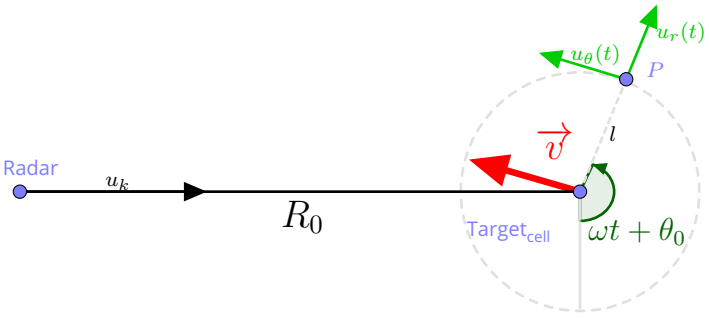


Figure 2.4 – Observation in the plane XY of one point P rotating, assuming that there is no angle between the rotation plane and the radar orientation.

— A : amplitude of the received signal.

If we consider polar coordinates $(\overrightarrow{u_r(t)}, \overrightarrow{u_\theta(t)})$ for the point P, we obtain the following equations :

$$\begin{aligned}\overrightarrow{r(t)} &= \overrightarrow{R_0} + t\overrightarrow{v} + l\overrightarrow{u_r(t)},\\ \overrightarrow{R_0}\overrightarrow{u_k} &= R_0,\\ l\overrightarrow{u_r(t)}\overrightarrow{u_k} &= l\sin(\omega t + \theta_0).\end{aligned}$$

The electromagnetic answer of a rotating point P is thus :

$$\begin{aligned}s(t) &= Ae^{2i\pi f_0 t}e^{-ik\overrightarrow{r(t)}(2\overrightarrow{u_k})},\\ &= Ae^{2i\pi f_0 t}e^{-ik(\overrightarrow{R_0}+t\overrightarrow{v}+l\overrightarrow{u_r(t)})(2\overrightarrow{u_k})},\\ &= Ae^{2i\pi f_0 t}e^{-2ik\overrightarrow{R_0}\overrightarrow{u_k}}e^{-2ikt\overrightarrow{v}\overrightarrow{u_k}}e^{-2ikl\overrightarrow{u_r(t)}\overrightarrow{u_k}},\\ &= Ae^{2i\pi f_0 t}e^{-2ik\overrightarrow{R_0}\overrightarrow{u_k}}e^{-2iktv_r}e^{-2ikl\sin(\omega t+\theta_0)}.\end{aligned}$$

The Doppler effect is defined by the factor e^{-2iktv_r}. The micro-Doppler effect for rotation is expressed by $e^{-2ikl\sin(\omega t+\theta_0)}$.

More generally, a target rotor is made up of several blades. These blades have the same shape and are angularly equally distributed. Let N be the number of blades. Each blade n has now its own initial phase shifted by $\frac{2n\pi}{N}$. We replace $\theta_0 \leftarrow \theta_0 + \frac{2n\pi}{N}, n \in [\![1; N]\!]$ and sum the contribution of all the blades to obtain :

$$s(t) = Ae^{2i\pi f_0 t}e^{-2ik\overrightarrow{R_0}\overrightarrow{u_k}}e^{-2iktv_r}\sum_{n=1}^{N}e^{-2ikl\sin(\omega t+\theta_0+\frac{2n\pi}{N})}.$$

If we gather the amplitude terms in the constant A (secondary terms) and use the following notations :

$$\begin{aligned}A &\leftarrow Ae^{-2ik\overrightarrow{R_0}\overrightarrow{u_k}},\\ \forall n \in [\![1; N]\!], \theta_n &= \theta_0 + \frac{2n\pi}{N},\\ B &= -2k,\\ &= -\frac{4\pi f_0}{c},\\ \omega_{v_r} &= 2\pi f_0 - \frac{2f_0}{c}v_r,\end{aligned}$$

we obtain the model of one rotor with N blades, each considered as a point expressed by :

$$s(t) = Ae^{i\omega_{v_r}t}\sum_{n=1}^{N}e^{iBl\sin(\omega t+\theta_n)}. \tag{2.2}$$

Rotating Points : Main Case

Now, we take into account the different angles :

— α : azimuth of the radar,

— β : elevation of the radar,

— α_p : yaw of the target,

— β_p : pitch of the target.

We assume that these angles are small : the radar aims at a stable target.

As detailed in [10] and summarized below, we re-obtain Equation (2.2) with a modified coefficient B. To keep short, the coordinates of $\overrightarrow{R_0}$ and $l\overrightarrow{u_r}$ become :

$$\begin{aligned}\overrightarrow{R_0} &= R_0\left(\cos(\alpha)\cos(\beta), \sin(\alpha)\sin(\beta), \sin(\beta)\right),\\ l\overrightarrow{u_r} &= l\left(\cos(\alpha_p)\cos(\beta_p), \sin(\alpha_p)\sin(\beta_p), \sin(\beta_p)\right).\end{aligned}$$

It implies a variation of the norm $r(t)$ of the vector $\overrightarrow{r(t)}$:

$$\begin{aligned}r(t) =&[(R_0\cos(\alpha)\cos(\beta) + l\cos(\alpha_p)\cos(\beta_p))^2\\ &+ (R_0\sin(\alpha)\cos(\beta) + l\sin(\alpha_p)\cos(\beta_p))^2\\ &+ (R_0\sin(\beta) + l\sin(\beta_p))^2]^{\frac{1}{2}},\\ =&\left(R_0^2 + l^2 + 2R_0 l\left(\cos(\alpha-\alpha_p)\cos(\beta)\cos(\beta_p) + \sin(\beta)\sin(\beta_p)\right)\right)^{\frac{1}{2}}.\end{aligned}$$

Thanks to the far field hypothesis, $l \ll R_0$. So the approximation $(1+x)^{\frac{1}{2}} \simeq 1 + \frac{x}{2}$, for small x, is valid :

$$r(t) = R_0 + l\left(\cos(\alpha-\alpha_p)\cos(\beta)\cos(\beta_p) + \sin(\beta)\sin(\beta_p)\right).$$

From this equation, we re-obtain Equation (2.2) with :

$$B = -\frac{4\pi f_0}{c}\left(\cos(\alpha-\alpha_p)\cos(\beta)\cos(\beta_p) + \sin(\beta)\sin(\beta_p)\right).$$

In this model, the radial velocity of the drone cell shifts the micro-Doppler frequency as an overset, Equation (2.2).

Blades in Motion

Now, we add the impact of a blade of length L which moves. We consider it as a 1-dimensional continuous line at a distance $l \in [0; L]$ from the rotor center

which is considered to be at the same position than the $\text{Target}_{\text{cell}}$. As we assume no interaction between the different points, we sum the contribution of all of them :

$$
\begin{aligned}
s(t) &= Ae^{i\omega_{v_r}t}\sum_{n=1}^{N}\int_{l=0}^{L} e^{iBl\sin(\omega t+\theta_n)}dl,\\
s(t) &= Ae^{i\omega_{v_r}t}\sum_{n=1}^{N}\left[\frac{e^{iBl\sin(\omega t+\theta_n)}}{B\sin(\omega t+\theta_n)}\right]_0^L,\\
s(t) &= Ae^{i\omega_{v_r}t}\sum_{n=1}^{N} e^{iB\frac{L}{2}\sin(\omega t+\theta_n)}\frac{e^{iB\frac{L}{2}\sin(\omega t+\theta_n)}-e^{-iB\frac{L}{2}\sin(\omega t+\theta_n)}}{B\sin(2\pi\omega t+\theta_n)},\\
s(t) &= Ae^{i\omega_{v_r}t}\sum_{n=1}^{N} e^{iB\frac{L}{2}2\sin(\omega t+\theta_n)}\frac{\sin(B\frac{L}{2}\sin(\omega t+\theta_n))}{B\sin(\omega t+\theta_n)}.
\end{aligned}
$$

The equation for a single rotors with several blades considered as 1-D lines is :

$$
\begin{aligned}
&A \leftarrow AL,\\
&s(t) = Ae^{i\omega_{v_r}t}\sum_{n=1}^{N} e^{iB\frac{L}{2}\sin(\omega t+\theta_n)}\text{sinc}\left(B\frac{L}{2}\sin(\omega t+\theta_n)\right).
\end{aligned}
\tag{2.3}
$$

Spinning Rotors

We add the contribution of the different rotors of the drone taking the Equation (2.3) as a starting point. We assume to have

— M : rotors,

— N : blades per rotor.

As before, thanks to the linearity hypothesis, we sum the contribution of the rotors. Thus, each rotor m turns at its own speed (Fig. 2.5). In the previous equations we replace :

— $\omega \leftarrow \omega_m$,

— $\theta_0 \leftarrow \theta_{0_m}$,

— $\theta_n \leftarrow \theta_{n_m} = \theta_{0_m} + \frac{2n\pi}{N}$.

As we work with the far-field hypothesis, all distances between the $\text{Target}_{\text{cell}}$ and the center of a rotor are neglected. The M rotors turn with the same center without any interaction.

The model of M rotors with N blades per rotors gives the following equation :

$$
s(t) = Ae^{i\omega_{v_r}t}\sum_{m=1}^{M}\sum_{n=1}^{N} e^{iB\frac{L}{2}\sin(\omega_m t+\theta_{n_m})}\text{sinc}\left(B\frac{L}{2}\sin(\omega_m t+\theta_{n_m})\right). \tag{2.4}
$$

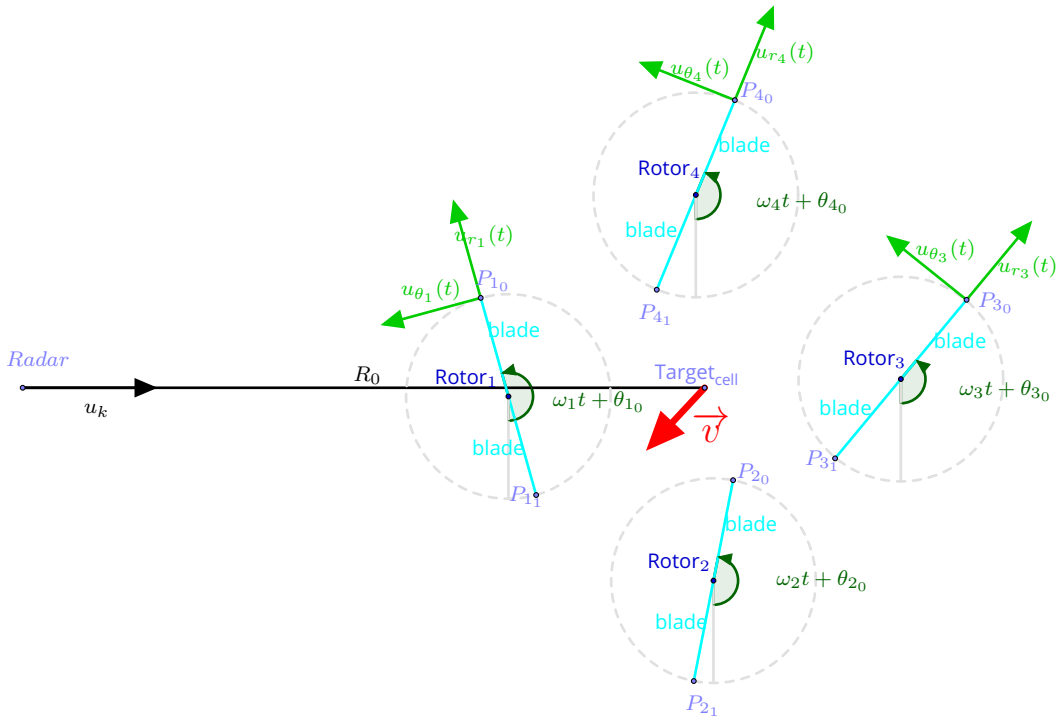


Figure 2.5 – Observation from above of one drone with 4 rotors. Each rotor having 2 blades. The scale of the objects is not respected. Due to the far field hypothesis, the rotor centers are considered at the same location as the target cell.

Drone cell

Now, we add to Equation (2.4) the influence of the drone cell which corresponds to the main part of the drone without internal movement. It is modeled with one reflector point $\text{Target}_{\text{cell}}$. Let :

— A_1 be the amplitude corresponding to the signal of the rotors,

— A_2 be the amplitude corresponding to the signal of the target cell.

We obtain the following model :

$$s(t) = e^{i\omega_{v_r} t} \left[\left(A_1 \sum_{m=1}^{M} \sum_{n=1}^{N} e^{iB\frac{L}{2}\sin(\omega_m t + \theta_{n_m})} \text{sinc}\left(B\frac{L}{2}\sin(\omega_m t + \theta_{n_m}) \right) \right) + A_2 \right]. \tag{2.5}$$

This equation allows us to observe that two constants, A_1, and A_2, are unknown. They are associated to the different RCSs of the target and can only be obtained by having access to at least a part of a radar signal. Other information (rotation speed, roll, pitch, etc.) are not accessible with radar measurements but with an access to the drone status across time. Such information is recorded by the drone in the drone log. To match better the simulation to the real signal, we have to consider that A, A_1, A_2 vary slowly across time. So, long trajectories are split into mall chunks. Then, the Equations (2.2) – (2.5) are used with constant amplitude terms for each chunks. These coefficients vary between the chunks only. A chunk duration is typically from 0.1 to 1 second.

In some studies, more and more complex simulations are proposed [47, 7]. For example, we can consider the radius of the rotor center L_1. Indeed, in practice, each rotor center is not an infinitesimal point. It contains a central circular part of a specific radius L_1. This part does not vary across time which implies a reduction of the rotating contribution of the blade. In our models, the computation is identical to the previous part but instead of integrating from 0 to L, we integrate only from L_1 to $L_2 = L$. Thus, the larger is L_1, the less is the signal.

$$s(t) = e^{i\omega_{v_r} t} \left[A_1 \sum_{m=1}^{M} \sum_{n=1}^{N} e^{iB\frac{L_1+L_2}{2}\sin(\omega_m t + \theta_{n_m})} \text{sinc}\left(B\frac{L_2 - L_1}{2}\sin(\omega_m t + \theta_{n_m}) \right) + A_2 \right]. \tag{2.6}$$

Another example, an Australian declassified document [47], takes into account simple 2-D shapes for the helicopter rotors. However, as we will detail in Section 2.2.1, the simulations have more important issues than the blade shapes such as the setup of realistic drone states (rotation speeds, roll, pitch, yaw, etc.) during the maneuvers. In Section 2.3.2, a brief comparison between simulations and the collected data is made to understand the electromagnetic limitations of these theoretic models.

2.1.3 . The Micro-Doppler Phenomenon for Drones : Frequency Analysis

Now, we present the impact of the micro-Doppler phenomenon from the frequency point of view with FT (*Fourier Transform*).

In this presentation, spectra are calculated only for an observation time longer than the rotation speed of the rotors. The resulting spectrograms (concatenation of spectra across time) are thus a *long-time* spectrograms compared to the *short-time* spectrograms detailed in the next chapter. The signal is thus considered stationary for each spectrum.

According to the Dirichlet theorem, the Fourier series of a continuous and $\frac{2\pi}{\omega_m}$-periodic function g converges pointwise toward g :

$$g(t) = \sum_{p=-\infty}^{\infty} c_p(g) e^{ip\omega_m t},$$

with,

$$c_p(g) = \frac{\omega_m}{2\pi} \int_{-\frac{\pi}{\omega_m}}^{\frac{\pi}{\omega_m}} g(t) e^{-ip\omega_m t} dt.$$

Rotating Points

We start our Fourier analysis with one rotor and one rotating point per rotor. Thus, the function g to study is $g_{l,m}(t) = Ae^{iBl\sin(\omega_m t+\theta_0)}$, Equation (2.2).

It implies :

$$\begin{aligned} g_{l,m}(t) &= Ae^{iBl\sin(\omega_m t+\theta_0)}, \\ s(t) &= e^{i\omega_{v_r} t} g_{l,m}(t), \\ c_p(g_{l,m}) &= \frac{\omega_m}{2\pi} \int_{\frac{-\pi}{\omega_m}}^{\frac{\pi}{\omega_m}} Ae^{i(Bl\sin(\omega_m t+\theta_0)-p\omega_m t)} dt. \end{aligned}$$

We introduce the Bessel integrals of order 1 $J_p(x)$, defined as follows :

$$\begin{aligned} \forall x \in \mathbb{C}, \forall p \in \mathbb{Z}, J_p(x) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i(x\sin(\tau)-p\tau)} d\tau, \\ c_p(g_{l,m}) &= AJ_p(Bl)\, e^{ip\theta_0}, \\ g_{l,m}(t) &= A \sum_{p=-\infty}^{\infty} J_p(Bl)\, e^{ip(\omega_m t+\theta_0)}, \\ s(t) &= A \sum_{p=-\infty}^{\infty} J_p(Bl)\, e^{ip\theta_0} e^{i(p\omega_m+\omega_{v_r})t}. \end{aligned}$$

The Fourier Transform $S(f)$ of the signal $s(t)$ is written :

$$\begin{aligned} F_{v_r} &= \frac{\omega_{v_r}}{2\pi}, \\ f_m &= \frac{\omega_m}{2\pi}, \\ s(t) &\overset{\mathsf{FT}}{\rightarrow} S(f), \\ e^{i(p\omega_m + 2\pi F_{v_r})t} &\overset{\mathsf{FT}}{\rightarrow} \delta(f - pf_m - F_{v_r}), \end{aligned}$$

$$S(f) = A \sum_{p=-\infty}^{\infty} J_p\left(Bl\right) e^{ip\theta_0} \delta(f - pf_m - F_{v_r}).$$

We observe that great values appear in the spectrum every f_m. Now, in the model of N rotating points, the function is $\frac{2\pi}{N\omega_m}$-periodic ($g_{l,m} \leftarrow g_{l,m,n}$) and we define $h_{l,m}(t)$:

$$\begin{aligned} h_{l,m}(t) &= \sum_{n=1}^{N} g_{l,m,n}(t), \\ &= \sum_{n=1}^{N} e^{iBl\sin\left(\omega_m + \theta_0 + n\frac{2\pi}{N}\right)}, \\ c_p(h_{l,m}) &= \frac{N\omega_m}{2\pi} \sum_{n=1}^{N} \int_{-\frac{\pi}{N\omega_m}}^{\frac{\pi}{N\omega_m}} e^{i\left(Bl\sin\left(\omega_m + \theta_{m_0} + n\frac{2\pi}{N}\right) - pN\omega_m t\right)} dt, \\ &= \frac{N}{2\pi} e^{ipN\theta_{m_0}} \sum_{n=1}^{N} \int_{-\frac{\pi}{N} + \theta_{m_0} + n\frac{2\pi}{N}}^{\frac{\pi}{N} + \theta_{m_0} + n\frac{2\pi}{N}} e^{i(Bl\sin u - pNu)} du, \\ &= \frac{N}{2\pi} e^{ipN\theta_{m_0}} \int_{\frac{\pi}{N} + \theta_{m_0}}^{\frac{(2N+1)\pi}{N} + \theta_{m_0}} e^{i(Bl\sin u - pNu)} du, \\ &= N J_{Np}(Bl) e^{ipN\theta_{m_0}}, \\ h_{l,m}(t) &= \sum_{p=-\infty}^{\infty} c_p(h_l) e^{iNp\omega_m t}, \\ &= \sum_{p=-\infty}^{\infty} N J_{Np}(Bl) e^{ipN\theta_{m_0}} e^{iNp\omega_m t}. \end{aligned} \tag{2.7}$$

By doing this, the Fourier transform of one rotor with N rotating points at the same distance l is obtained :

$$
\begin{aligned}
A &\leftarrow AN, \\
s(t) &= A \sum_{p=-\infty}^{\infty} J_{Np}\left(Bl\right) e^{iNp\theta_{m_0}} e^{i(pN\omega_m + \omega_{v_r})t},
\end{aligned}
$$

$$
S(f) = A \sum_{p=-\infty}^{\infty} J_{Np}\left(Bl\right) e^{iNp\theta_{m_0}} \delta(f - pNf_m - F_{v_r}). \tag{2.8}
$$

Now, micro-Doppler effects appear every Nf_m. A rotor of two blades at $5\,000$ RPM (*Revolution Per Minute*) has the same impact as a rotor of one blade at $10\,000$ RPM.

This simplified model is sometimes used in the literature with $l = \frac{L}{2}$, the median point of the blade to simulate a drone spectrum profile, expressed by the following equation :

$$
S(f) = A \sum_{p=-\infty}^{\infty} J_{Np}\left(B\frac{L}{2}\right) e^{ipN\theta_{m_0}} \delta(f - pNf_m - F_{v_r}).
$$

Blades in Motion

If we take into account the impact of a full blade of length L, the sinc() term from Equation (2.3) is included in the Equation (2.8). We define $h_m(t)$:

$$
\begin{aligned}
h_m(t) &= \int_{l=0}^{L} h_{l,m}(t) dt, \\
&= \sum_{n=1}^{N} \int_{l=0}^{L} g_{l,m,n}(t) dl, \\
&= \sum_{n=1}^{N} e^{iB\frac{L}{2}\sin(\omega_m t + \theta_{n_m})} \mathsf{sinc}\left(B\frac{L}{2}\sin(\omega_m t + \theta_{n_m})\right), \\
s(t) &= A e^{2i\pi F_{v_r} t} h_m(t).
\end{aligned}
$$

Next, the result obtained in Equation (2.7) is directly used :

$$
\begin{aligned}
c_p(h_m) &= \frac{N\omega_m}{2\pi} \sum_{n=1}^{N} \int_{-\frac{\pi}{N\omega_m}}^{\frac{\pi}{N\omega_m}} \int_0^L e^{i\left(Bl\sin\left(\omega_m+\theta_{m_0}+n\frac{2\pi}{N}\right)-pN\omega_m t\right)} dl dt, \\
&= \int_0^L \frac{N\omega_m}{2\pi} \sum_{n=1}^{N} \int_{-\frac{\pi}{N\omega_m}}^{\frac{\pi}{N\omega_m}} e^{i\left(Bl\sin\left(\omega_m+\theta_{m_0}+n\frac{2\pi}{N}\right)-pN\omega_m t\right)} dt dl, \\
&= \int_{l=0}^{L} c_p(h_{l,m}) dl, \\
&= N e^{ipN\theta_{m_0}} \int_0^L J_{Np}(Bl) dl.
\end{aligned}
$$

With the definition :

$$
\forall a \in \mathbb{N}, v_a = \int_0^L J_a(x) dx,
$$

and the Bessel formula [4] :

$$
\begin{aligned}
\forall a \in \mathbb{Z}, J_a(x) &= J_{a+2}(x) + 2J'_{a+1}(x), \\
\forall x, J_a(x) &\overset{a\to+\infty}{\to} 0,
\end{aligned}
$$

we obtain by integration :

$$
\begin{aligned}
v_a &= v_{a+2} + 2J_a(L), \\
v_a &= \lim_{k\to+\infty} v_{a+2k} + 2\sum_{k=0}^{+\infty} J_{a+2k+1}(L),
\end{aligned}
$$

which yields :

$$
\begin{aligned}
c_p(h_m) &= 2N e^{ipN\theta_{m_0}} \sum_{k=0}^{+\infty} J_{Np+2k+1}(BL), \\
h_m(t) &= \sum_{p=-\infty}^{+\infty} c_p(h) e^{ip\omega t}.
\end{aligned}
$$

We insert this result into the model with several blades and one rotor given by Equation (2.3) to obtain its Fourier transform :

$$
\begin{aligned}
s(t) &= A \sum_{p=-\infty}^{+\infty} e^{ipN\theta_{m_0}} \sum_{k=0}^{+\infty} J_{Np+2k+1}(BL) e^{i(pN\omega_m+\omega_{v_r})t}, \\
S(f) &= A \sum_{p=-\infty}^{+\infty} e^{ipN\theta_{m_0}} \sum_{k=0}^{+\infty} J_{Np+2k+1}(BL) \delta(f - pNf - F_{v_r}).
\end{aligned} \tag{2.9}
$$

The simulated example in Fig. 2.6 illustrates Equation (2.9). This simulation is made with one rotor with two blades ($N = 2, M = 1$) at a RPM 5000 r/min : $\omega_1 = 2\pi\frac{\text{RPM}}{60}$. The gap between two lines is $133 H_z = N\frac{\omega_1}{2\pi}$.

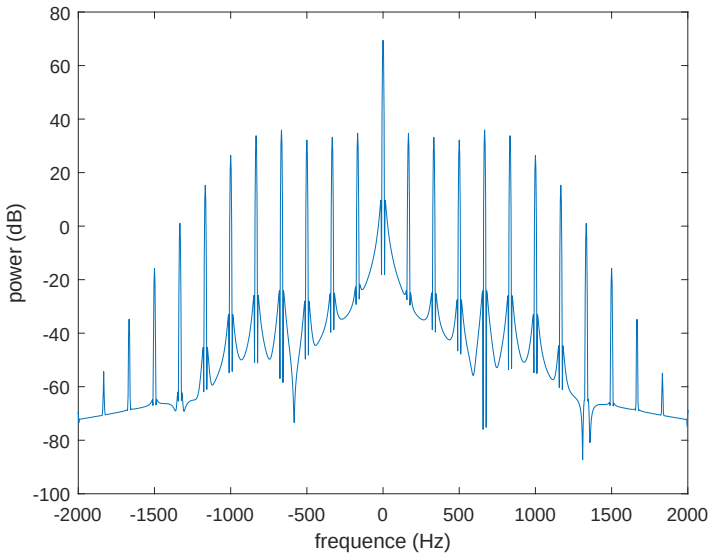


Figure 2.6 – Simulation of micro-Doppler profile. The spectrum of one rotor with two blades turning at $5\,000$ RPM. The time window (300 ms) is relatively long to the period of rotation.

Spinning Rotors

We place ourselves in the case of M rotors. The linearity of the models and of FT allow us to sum directly the contribution of each rotor m :

$$s(t) = A \sum_{m=1}^{M} \sum_{p=-\infty}^{+\infty} e^{ipN\theta_{m_0}} \sum_{k=0}^{+\infty} J_{Np+2k+1}(BL) e^{i(pN\omega_m + \omega_{v_r})t},$$

$$S(f) = A \sum_{m=1}^{M} \sum_{p=-\infty}^{+\infty} e^{ipN\theta_{m_0}} \sum_{k=0}^{+\infty} J_{Np+2k+1}(BL) \delta(f - pNf_m - F_{v_r}).$$

Drones Cell

Equation (2.5), with the drone cell added, rewritten in the frequency domain is :

$$
\begin{aligned}
s(t) =& e^{i\omega_{v_r} t} \left(A_1 \sum_{m=0}^{M} \sum_{n=1}^{N} e^{iB\frac{L}{2}\sin(\omega_m t + \theta_{n_m})} \text{sinc}\left(B\frac{L}{2}\sin(\omega_m t + \theta_{n_m}) \right) \right) \\
&+ A_2 e^{i\omega_{v_r} t}, \\
S(f) =& A_1 \sum_{m=1}^{M} \sum_{p=-\infty}^{+\infty} e^{ipN\theta_{m_0}} \sum_{k=0}^{+\infty} J_{Np+2k+1}(BL)\delta(f - pNf_m - F_{v_r}) \\
&+ A_2 \delta\left(f - F_{v_r}\right). \qquad (2.10)
\end{aligned}
$$

To conclude, we consider the radius of the rotor center L_1, obtaining :

$$
\begin{aligned}
S(f) = A_1 \left[\sum_{m=1}^{M} \sum_{p=-\infty}^{+\infty} e^{ipN\theta_{m_0}} \sum_{k=0}^{+\infty} \left(J_{Np+2k+1}(BL_2) - J_{Np+2k+1}(BL_1) \right) \delta(f - pNf_m - F_{v_r}) \right] \\
+ A_2 \delta\left(f - F_{v_r}\right). \qquad (2.11)
\end{aligned}
$$

2.2 . Incompleteness on the State of the Art

We present the limitations of the models which lead us to carry out our own measurement campaign. We start with the limitations of the simulations given above and continue with the practical limitations of the only real dataset available.

2.2.1 . Simulations

From the electromagnetic models commented in the previous section, we can obtain simulated radar profiles. As we explained, even more complex models exist to be as precise as possible. However, the quality of a simulation depends on the quality of the weakest link of the entire protocol. This weak link corresponds to the information needed to set up simulations.

The protocol to obtain simulated signals is depicted in Fig. 2.7. First of all, we observe that these models need information about electromagnetic characteristics : RCS values of the target and signal shape. Access to those values is not a challenge. For example, the RCS of a drone depends in practice on the observation angle but it can be obtained in an anechoic chamber as in [17] or during a measurement campaign. This procedure suffers, however, from a major problem, the lack of information about mechanic characteristics : the evolution of the drone state.

The micro-Doppler effect characterizes the maneuvers of the target. As we consider that these maneuvers are distinct enough to discriminate the different targets, we indirectly consider that the micro-Doppler characterizes the target. Thus, we need realistic drone states for reliable simulations. Nonetheless, these drone states are hard to obtain. The lack of drone states is a main obstacle to set up simulations.

This problem is often put aside in existing studies which leads to major differences between simulations and real signals. For example, in [7, 12] the rotor

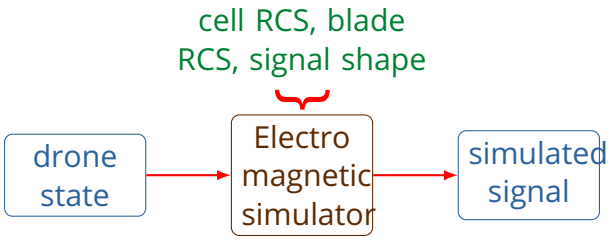


Figure 2.7 – Diagram for simulation process fed with real drone states.

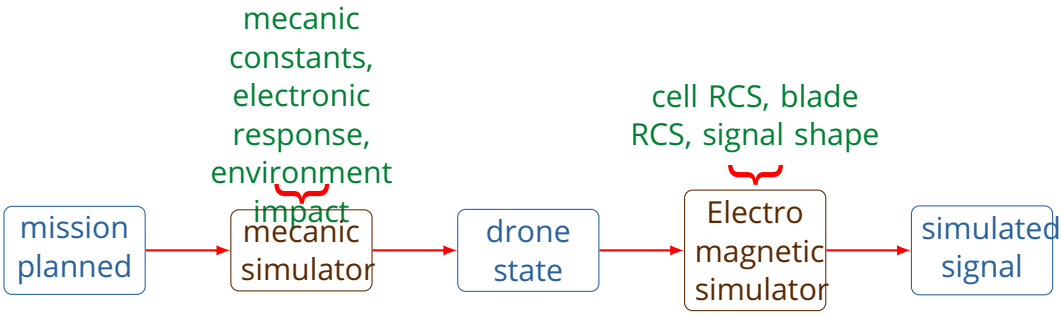


Figure 2.8 – Diagram for simulation process without real drone states available.

speeds are arbitrarily chosen. In [7], the different rotors turn at the same speed for each observation of 250 ms.

The mechanic behind the drone behavior is complex and depends on numerous parameters. However, comprehension of this behavior is essential as it results in considerable differences between real signals and simulations. For example, quadcopter drones can fly in two modes that we will call "$+$" and "$\times$" (speed vector going from left to right). In the $\times$ mode, the positions of the rotors relative to the drone speed direction are two front and two back, while in the $+$ mode two rotors are in the middle, one front and one back. Moreover, outdoor environmental influence (wind, etc.) is unpredictable and the drone flying commands have to compensate it constantly. The drone is mostly considered as being in a constant unstable equilibrium.

The first possibility is to use the drone logs of the measured drones. A log corresponds to the information given by the drone while flying. It may contain many pertinent materials going from basic information such as the GPS position to more complex ones (rotor speeds or even wind direction and strength). These values are not available in the studies of the literature.

So, a second possibility is to use a drone mechanical behavior simulator to obtain the drone states for different maneuvers. Thus, the protocol for simulation process of Fig. 2.7 becomes the one in Fig. 2.8. Such a simulator is not used in

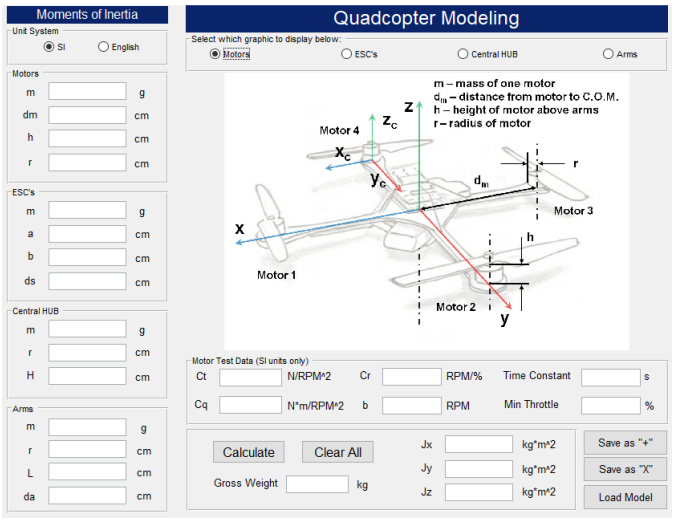


Figure 2.9 – Mechanic parameters needed for mechanic simulation.

the radar literature but exists in other scientific communities. We cite for example a free open-source mechanic simulator [23]. This simulator is proposed for nano-drones [17] which are too small for our application but could probably be adapted for our purpose.

In this new protocol, the flight simulator gives realistic states at a short refreshing time : every ten or hundreds of milliseconds. Between each mechanic state, the drone is considered stationary and the electromagnetic simulator gives the signal answer at a high sampling time : tens or hundreds of microseconds ($\frac{1}{PRF}$). However, a large set of mechanic parameters are needed (Fig. 2.9). These parameters are hard to acquire for the vast majority of drones. Besides, other limitations of this model can be observed. For example, the environmental influence (wind, etc.) is neglected despite its considerable impact. The resulting signal is not realistic with most rotors turning at a similar speed with low variations. As we will detail in the next subsection, a real drone signal implies many rotor variations.

To conclude, simulations provide a good understanding of the physical phenomenon. They suffer, however, from theoretical and practical weaknesses. In addition to the simplification implied by the hypothesis needed for electromagnetic simulations (about SER parameters, rotors not hiding each other, etc.), we emphasize the lack of mechanic simulators for radar studies. To overcome these difficulties, real datasets need to be collected.

2.2.2 . Existing Dataset

The acquisition of radar signals is difficult for various reasons. First, the majority of radars is not designed for targets as small as a mini-drone. Second, those calibrated to such targets do not measure the micro-Doppler effect. Indeed,

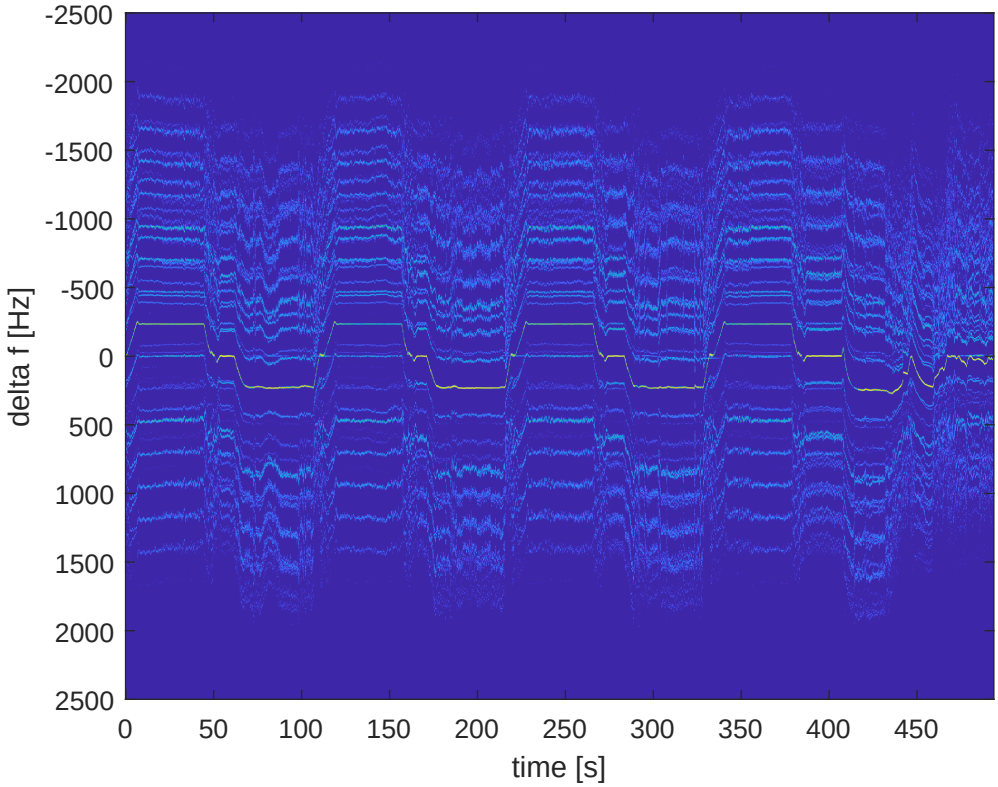


Figure 2.10 – Long-time spectrogram, example on real data from the NATO dataset.

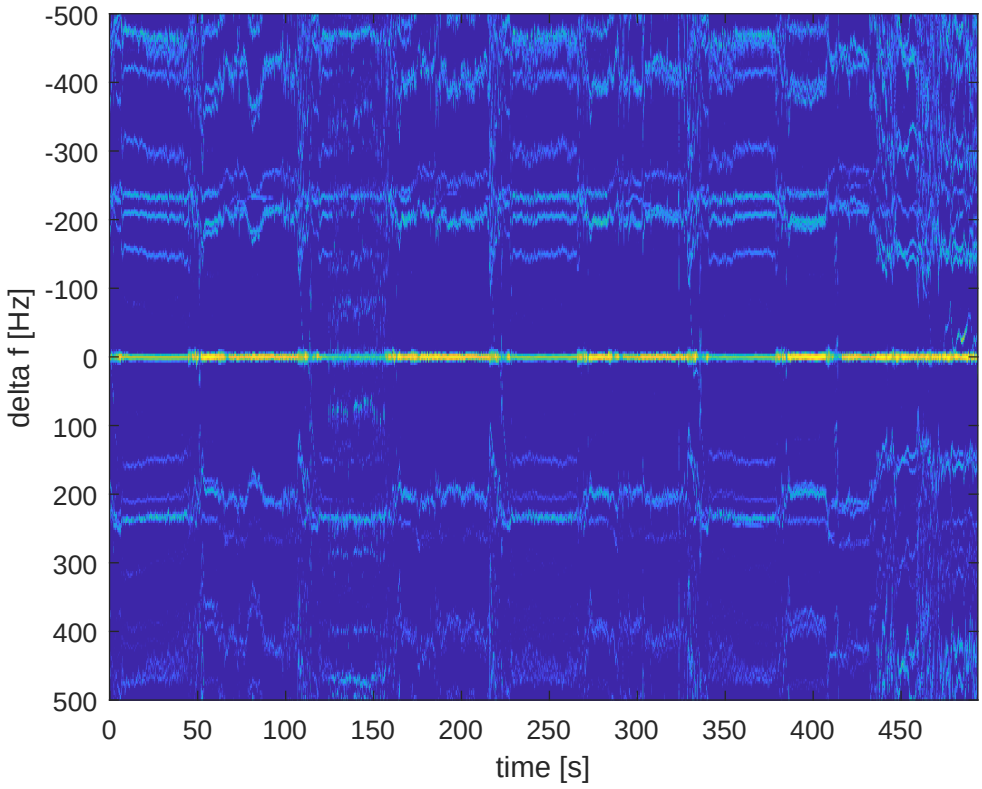


Figure 2.11 – Long-time spectrogram, example on real data from the NATO dataset, main Doppler effect set to zero, a zoom on the y-axis from Fig. 2.10.

they measure the main Doppler effect to detect a potential incoming target without identification. Thus, most of the micro-Doppler measurements are performed with prototype radars, still under research, not in industrial production. Third, some targets like birds are only opportunity targets and do not cooperate.

Fortunately, the NATO working group SET-245 declassified a few years ago one dataset which was used in [6]. To the best of our knowledge, it is the only dataset available with those properties [1]. This set contains data of ten different drones with the measure recording lengths which vary from 7 s to 672 s. As that measurement campaign was carried out to obtain reference profiles, the measurements were made in perfect conditions. They were not designed to cover a variety of real situations. Each drone is measured only once on a simple trajectory mostly composed of uniform rectilinear movements going either forwards or backward.

In [6], drone classification methods are compared thanks to this dataset. However, its limits raise concerns. For example, two drones with only a few seconds of recording cannot be exploited as they are not identified at all. Moreover, as each drone is recorded only once, it creates environmental biases. There is no way of knowing whether the classifier recognizes the target or the environment (meteorological condition, etc.).

Such a phenomenon, denoted *Irrelevant Correlations* [52], is common in machine learning. For example, in [49], a bias is willingly introduced for the classification between wolves and huskies. The authors selected only wolves images in a snow environment and husky images in another environment. In this case, they observe that regardless of the fur color, pose, or face of the dog, the classifier tends to consider the white background as the main recognition feature for wolves.

In the radar context, this problem is always present albeit hidden. The datasets may look large with hundreds or even thousands of profiles per class but all profiles correspond to consecutive instants of one or several trajectories with poor diversity of the drone behavior.

The measurements have a very good resolution thanks to PRF of the radar, PRF $= 25\,000$ Hz, a very high value compared to a classic radar. The measurements were made at different frequencies (from 1 to 20 GHz), but we only consider the S-band (3 GHz) as this band is frequently used. Reasoning remains the same regardless of the band.

The higher the band is, the lower the wavelength is, which implies a better resolution. However, the attenuation due to the radar-target distance increases with the augmentation of the emitted frequency. Some recent radars work with higher frequencies to improve the resolution but the majority of the accessible radars work in the S-band.

1. We would like to thank Air Force Research Laboratory (AFRL) for sharing the experimental data of this study. Data provided through NATO SET-245.

Fig. 2.10, presents a long-time spectrogram obtained with a time window 0.3 seconds (a frequency zoom from -2500 Hz to 2500 Hz). The target is a DJI Phantom, one of the most popular mini-drones, often used in measurement campaigns. The target trajectory can be deduced from the main Doppler effect corresponding to the strongest signal (yellow) in the center. Each plateau corresponds to a rectilinear uniform movement. When the target is going toward the radar, the frequency shift is negative. Around this main shift, the micro-Doppler lines are observed. The main Doppler effect can be considered as an offset and is removed for the target identification as in Fig. 2.11.

Even if the trajectory is composed of basic movements (succession of rectilinear uniform ones), the resulting micro-Doppler lines are quite unstable across time. From time to time, a widespread profile is observed (for example in Fig. 2.10 at 51 s). It reflects abrupt changes of the trajectory which correspond probably to U-turns but the lack of the ground truth (the exact trajectory is not available) prevents us from verifying this hypothesis or any other hypotheses we could make by observing this signal.

Most of the time, only two or three micro-Doppler lines are visible even though this signal corresponds to a quad-copter. In absence of the ground truth, we do not know whether the missing lines are due to rotors hiding each other or to the confusion created by rotors going at an identical speed.

Some signals are also observed at unexpected frequencies, for example between $100\,s$ to $150\,s$ at frequencies below 100 Hz. It might be due to very low rotor speeds, target vibrations, environment (clutter, etc.), or confusion with a second target (a bird following the drone). Indeed, if the distance resolution is lower than the distance between several targets, the signals of all targets are coherent and their isolation is a hard task.

Generally speaking, in datasets available in the literature, the ground truth is quite poor (no GPS data, no rotor speeds, etc.) which does not allow us to formulate any interpretation of the analysis of the observations.

The human interpretation of micro-Doppler signals is difficult, even for long observations such as depicted in Figs. 2.10 and 2.11. In the field, interpretations become even more difficult for shorter times as real targets do not remain measurable for a time long enough : capturing signals longer than ten to hundreds of milliseconds is problematic.

From the observations of available datasets, the following statements are deduced :

— The micro-Doppler effect observed for commercial drones gives diverse profiles, difficult to interpret by humans.

— Real data is poor, with only one trajectory per drone and no complex maneuvers. The lack of real data leads to biases which imply too high performance for recognition algorithms, compromising any analysis.

— The lack of the ground truth restricts the analysis of the data, for example, its comparison with associated simulations is impossible. There is a need for annotated data, especially about flight conditions as annotations may impact the micro-Doppler profiles considerably.

2.3 . Measurements : Dataset Creation

To cope with the problems mentioned above, we carried out a measurement campaign with the ONERA radar MEDYCIS.

2.3.1 . Measurement Campaign

The measurements were performed [2] in a single location during two weeks in April 2019 under different meteorological conditions with one target at a time. Fig. 2.12 represents some pictures taken during the campaign. Five drones (DJI S1000-A2, grand Spyder, Gryphon, DJI Phantom 4 Pro, DJI Mavic Pro) and several birds (peregrine falcon, lanner falcon, gyrfalcon, and opportunity targets as well) were involved.

The measurements were taken in similar conditions for all drones. Drones flew on trajectories differing by speed, altitude, pattern (up/down, rectilinear/circular movement, stationary/ rotation around its center, free fly) to gather a large variety of data. The drones were kept in "$\times$" mode with a fixed front and back face (the drone must make a 180° turn to switch its position). The camera of the drone, recording the scene in front of it, was left static. All the drones had two blades per rotor. Only classic commercial drones were used and no stealth material nor stealth shape were considered neither. We did not equip our drones with anything which would improve the signal quality. The drones allowing the autopilot were both measured on autopilot and humanly piloted mode.

Concerning birds, falcons were chosen because they can stay in the measured area with trajectories comparable to those designed for drones. Some opportunity birds were also measured during the campaign and added to the database. Unfortunately, the amount of data collected for birds was small. Indeed, falcons are not as controllable as a drone and we did not have any control on opportunity targets.

Table 2.1 lists the different trajectories recorded. Each drone trajectory lasts roughly fifteen minutes except for DJI S1000-A2 which had a lower autonomy (10 minutes). As the MEDYCIS radar system measures constantly the whole range, opportunity birds were also measured during some of these trajectories at other locations than the drone. The signals corresponding to each target were distant enough to avoid any interaction.

2. We thank Jean-Paul Marcelin and Jean-François Petex for their work on the data collecting campaign with the ONERA measurement system MEDYCIS.

Target	Date	hour (GPS)	drone log
Grand Spyder	04/04/2019	14h06m	yes
Grand Spyder	04/04/2019	14h29m	yes
Grand Spyder	09/04/2019	11h33m	yes
Grand Spyder	09/04/2019	11h48m	yes
Grand Spyder	09/04/2019	12h01m	yes
Grand Spyder	09/04/2019	12h20m	yes
Grand Spyder	10/04/2019	15h21m	yes
Grand Spyder	10/04/2019	15h36m	yes
Grand Spyder	12/04/2019	12h54m	yes
Gryphon	04/04/2019	09h08m	yes
Gryphon	04/04/2019	09h40m	yes
Gryphon	09/04/2019	12h39m	yes
Gryphon	09/04/2019	12h58m	yes
Gryphon	09/04/2019	13h40m	yes
Gryphon	12/04/2019	13h11m	yes
Gryphon	12/04/2019	13h35m	yes
DJI Phantom	04/04/2019	10h06m	yes
DJI Phantom	05/04/2019	14h26m	yes
DJI Phantom	05/04/2019	14h46m	yes
DJI Phantom	09/04/2019	14h03m	yes
DJI Phantom	09/04/2019	14h46m	yes
DJI Mavic	04/04/2019	13h18m	no
DJI Mavic	04/04/2019	13h38m	yes
DJI Mavic	08/04/2019	13h47m	no
DJI Mavic	08/04/2019	14h07m	no
DJI Mavic	10/04/2019	09h40m	no
DJI Mavic	10/04/2019	13h02m	yes
DJI S1000-A2	05/04/2019	08h04m	yes
DJI S1000-A2	05/04/2019	08h22m	yes
DJI S1000-A2	05/04/2019	08h36m	yes
DJI S1000-A2	05/04/2019	08h51m	yes
DJI S1000-A2	08/04/2019	12h50m	yes
DJI S1000-A2	08/04/2019	13h07m	yes
DJI S1000-A2	08/04/2019	13h22m	yes
Peregrine 1	06/04/2019	08h13m	yes
Peregrine 2	06/04/2019	09h23m	no
Peregrine 2	12/04/2019	14h34m	no
Peregrine 2	12/04/2019	14h50m	no
Lanner	06/04/2019	09h47m	yes
Lanner	06/04/2019	10h04m	yes
Gyrfalcon	06/04/2019	10h30m	no
Gyrfalcon	06/04/2019	10h47m	no

Table 2.1 – List of trajectories carried out

Figure 2.12 – Pictures took during the measurement campaign. The author of this thesis is depicted in the left of the top left picture.

Planning, Choices, and Expected Impact

Measurement System : Details

We choose for the waveform of the MEDYCIS radar signal, a linear chirp at S-band with a bandwidth of $300\,\text{MHz}$ (from 2.85 to $3.15\,\text{GHz}$) with PRF $= 10\,\text{kHz}$ and a peak power of 50 Watts on horizontal polarization (used both in emission and reception).

We restrict our measurement distance to $1.4\,\text{km} \pm 250\,\text{m}$ imposed by the legislation but we could theoretically measure the entire reception time (Fig. 2.13) so from $d_1 = \frac{T_{\text{pulse}} c}{2} = 750\,\text{m}$ to $d_2 = \frac{T_{\text{inter-pulse}} c}{2} = 15\,\text{km}$. In terms of altitude, we were also legislatively restricted to $150\,\text{m}$ maximum. The bandwith of the emitted signal is Band $= 300\,\text{MHz}$ in the MEDYCIS case, the distance ambiguity is $\frac{c}{2\text{Band}} = 0.5\,\text{m}$

Because of the direct trajectory hypothesis, targets being far appear at a shorter distance by folding effect due to our PRF. For example, the signal reflected on a target at $24\,\text{km}$ takes $\tau = 160\,\mu s$ to come back to the radar. For this reason, its

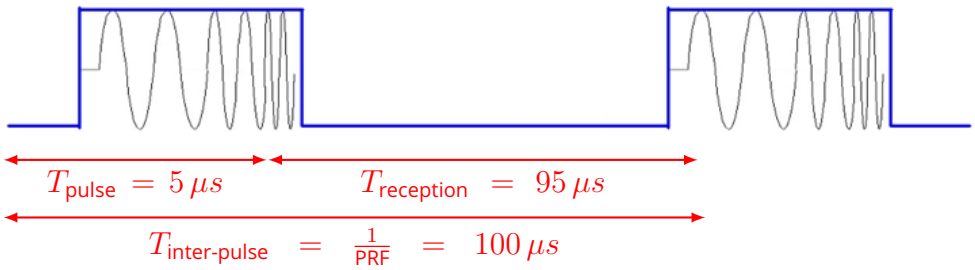


Figure 2.13 – Signal sent (MEDYCIS values).

signal is superimposed with the response of the second pulse sent at $\tau_2 = 100\,\mu s$. The signal is interpreted as the signal of a smaller target at a shorter distance $d = c\frac{(\tau - \tau_2)}{2} = 9\,\text{km}$. In our measurement scenario, such a situation would occur for a target at a very high altitude such as a plane. However, this situation did not happen.

Concerning the polarization, a linear polarization has been chosen. This polarization ensures a high energy of the reflected signal as rotor blades are horizontal but, in return, the ground contribution is reinforced. We chose this polarization because it is commonly used in micro-Doppler studies.

The Doppler ambiguity is $\Delta f_{max} = \frac{\text{PRF}}{2}$. If we consider a target at radial speed $v = 100\,\text{km/h}$, it implies a main Doppler shift of $\Delta f_D = \frac{2 f_0 v}{c} = 556\,\text{Hz}$ with c signifying the light speed. In addition, the maximum micro-Doppler shift induced by a rotor of radius L is $\Delta f_{\mu\text{-}D} = \frac{8\pi L f_0}{c}\frac{\text{RPM}}{60}$ [39]. For typical RPM $= 5\text{–}6 \cdot 10^3$ r/min and $L = 10\,\text{cm}$, the shift $\Delta f_{\mu\text{-}D}$ is 2–3 kHz. Thus, a PRF ($\Delta f_{max} > \Delta f_D + \Delta f_{\mu\text{-}D}$) large enough is an efficient means to avoid aliasing and to capture the micro-Doppler effect entirely. We choose the highest PRF possible for the MEDYCIS radar ($10\,\text{kHz}$) to reach this goal.

2.3.2 . Calibration and Preliminary Analysis

The data gathered should be post-processed to make it exploitable by machine learning algorithms. The post-treatment created the distance profiles and calibrated the data. The validation of the measurement campaign outcome was made with the "cleaned" data.

Distance Profile and Calibration

The signal response for each distance along the measured range is obtained by a correlation between the received and emitted signals (distance compressing). This provides the distance profiles, displayed in Fig. 2.15. In this image, the x-axis is the target-radar distance while the y-axis corresponds to the time (from top to bottom). The measured target is a DJI Phantom 4.

For the part of the trajectory depicted, the drone evolves between $1350\,\text{m}$, and $1400\,\text{m}$. From 1100 and $1150\,\text{m}$ we observe a dense bush. As we did the majority

of measurements with a sky background, we opted to discard these few seconds of such measures. However, we highlight the fact that observations with a large clutter may occur in practice and further studies should be done in this area.

In addition to the trajectories in Table 2.1, a specific drone called CALIBRI (Fig. 2.14) was also measured for the calibration purpose. This drone carries a specific crown which gives it a smooth shape. This crown hides the drone rotors to mask their micro-Doppler effect. It also gives a constant RCS to the target : the same observation is made regardless of the orientation. This constant RCS is known. As this drone is very specific, the data collected with it are not used for any other purpose than calibration. The same calibration can be made for data collected in the future for comparison. Studies gathering data from different measurement campaigns are a potential research topic issued from this measurement campaign.

Figure 2.14 – Calibri drone : a specific drone carrying a crown for RCS calibration.

Tracking

We collected distance profiles like in Fig. 2.15 of a several teraoctet. Fortunately, the interesting part is much smaller as the drone part corresponds to several distance cases only. To make the case selection (tracking) we used the GPS data of the drone (perfect tracking) except for opportunity birds and for data without logs where we used basic tracking algorithms. Each distance case has a dimension of $0.27\,\mathrm{m}$. We select the three strongest cases for each drone and compute the mean over it. We used this method for several reasons. First of all, this mean ensures the tracking to avoid losing any information with a distance resolution close to the size of the targets of our interest. Second, some radars have a distance resolution larger than the MEDYCIS system. By decreasing this resolution we put ourselves in more general conditions. For larger targets, this distance resolution may be used directly.

The ground truth is recorded by a drone and stored in its logs files. To have the same information for each drone we selected a part available for any drone (position, pitch, roll, yaw, rotor speeds), pertinent according to the electromagnetic models. We highlight that such information is difficult to obtain as the raw data associated differ for each drone. We thus had to apply specific procedures for each drone. For example, for some drones, only the control voltage of each rotor is available. Thus, we had to convert it to the rotor speed.

We also emphasize the importance of the ground truth to confront it with any hypothesis made for a measurement signal. To the best of our knowledge, this campaign is the first measurement campaign with annotated information such as rotor speeds. Unfortunately, due to practical issues during the measurement campaign, we did not gather some information for few trajectories of the DJI Mavic, Table 2.1.

Validation : Comparison to Ground Truth

To conclude this measurement campaign, we compared the data collected with the ground truth (drone logs). This comparison also allows us to illustrate certain limitations of the electromagnetic models.

Figs. 2.16 and 2.17 present two long-time spectrograms (60 and $40\,$s of observation) obtained by a concatenation of spectrum of the trajectory given in Fig. 2.18.

In these figures, the signal is compared to the theoretical rotor impacts available thanks to the drone log file. The real signal is in the background (a high amplitude in black) and the location of the theoretical effect of each rotor is in the foreground. We point out that the rotor speeds do not vary much during the $300\,$ms period. The relatively stationary hypothesis is respected for the observation time of this order.

In Fig. 2.16, the target speed impact (the main Doppler effect) is set to zero. The signal corresponding to it is significant and assimilated to the radar cross-section of the drone cell.

We also observe other amplitudes at various frequencies matching roughly to the expected micro-Doppler rotor lines. However, the radar measurement is probably more precise than the drone log, even if the latter is considered as a ground truth. The small misalignment observed results from the drone log imprecision. We assimilated the corresponding signal to the radar cross-section of the drone rotors.

The rotor RCS values vary considerably across time, some rotors becoming even invisible at certain moments. For example, in Fig. 2.17, the left front rotor disappears at around $385\,$s. The trajectory in Fig. 2.18, allows us to assume that this rotor is hidden by the others due to the drone maneuver (a turn). The right front rotor, however, remains visible, probably due to the drone orientation or roll. Sometimes, a rotor is only visible at even amplitude order, it is probably due to

rotors hiding each other : only one turn over two is visible. Such unpredictable phenomena are not taken into account in the models detailed in Section 2.1.2.

Another limitation of models lies in the asymmetry of the micro-Doppler line amplitudes between negative and positive Δ_f. It is caused by the slope of the rotor blade. Therefore, for a drone flying toward the radar, the blades appearing to the radar correspond to the negative values while the blades disappearing (half a turn later) correspond to the positive ones. The two sides are not equally visible.

We conclude that the electromagnetic simulation corresponds roughly to the signal when we are using the associated real drone log and can be assimilated to the global content of the signal. The measured signal also contains a realistic style due to unpredictable physical phenomena. This realistic style contains information on the target as well. We assume that these phenomena are characteristic for the different targets and can be caught by Deep Learning recognition methods.

2.3.3 . Conclusion

The preliminary analysis allows us to validate our measurement campaign. We managed to avoid the limitations reported in the state of the art and to produce reliable data. We thus ensure the quality of data to feed Deep Learning tools in our further experiments.

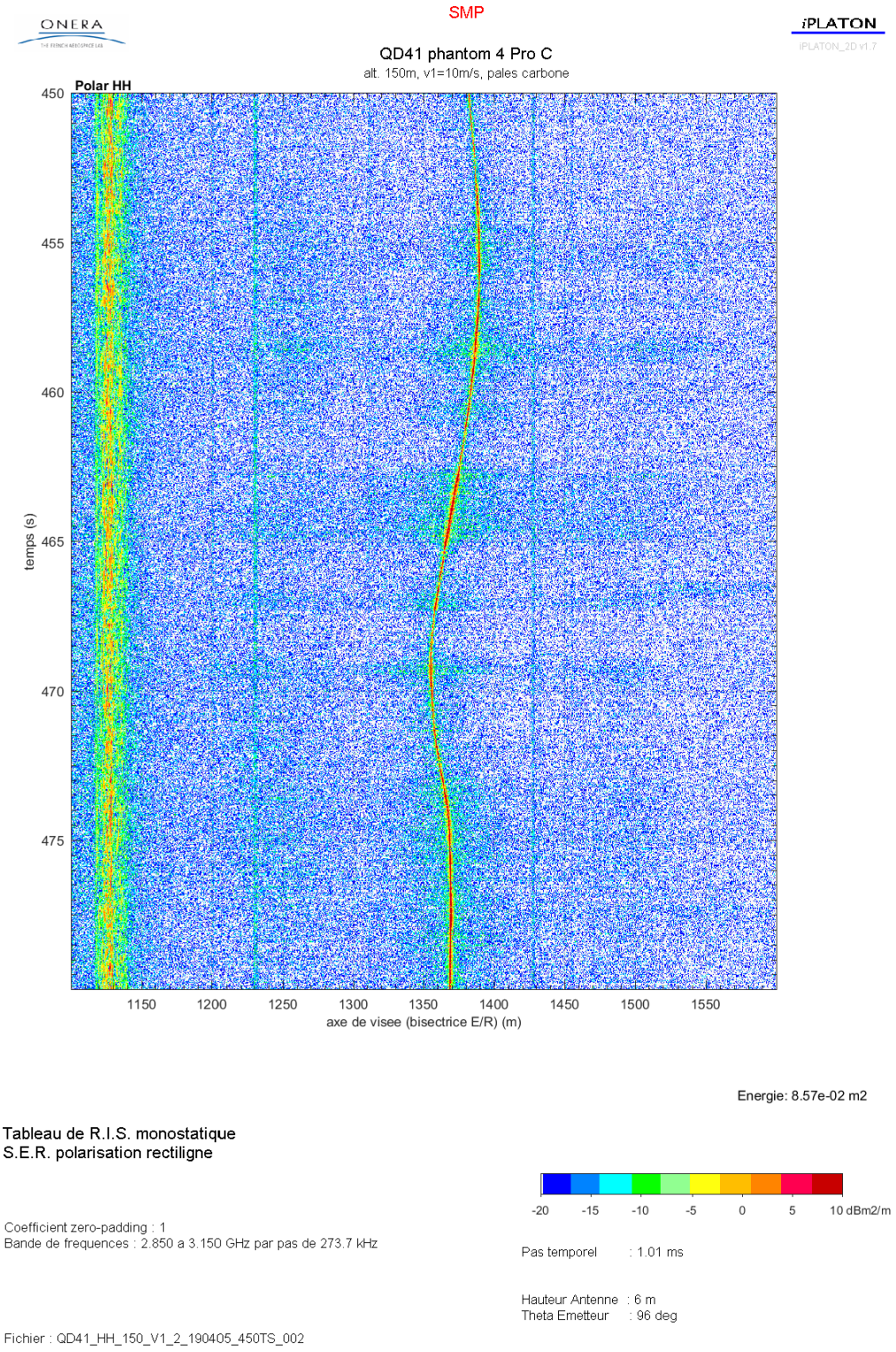


Figure 2.15 – Distance profile of a DJI Phantom, with a bush at 1150 m. A spectrogram of this trajectory is given in Fig. 2.16.

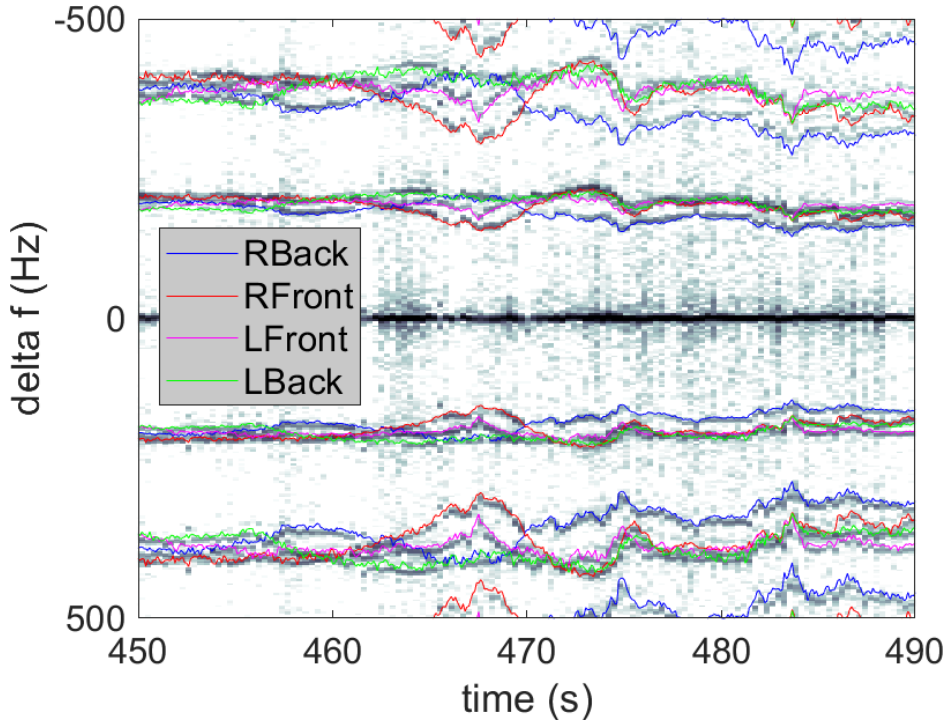


Figure 2.16 – Comparison between theoretical data and real observation. The foreground lines correspond to theoretical impact of rotors. The background is a concatenation of spectra to obtain a long-time spectrogram. The coordinate of this trajectory is in Fig. 2.18. The spectra data displayed in Fig. 3.2 corresponds to one column of this signal.

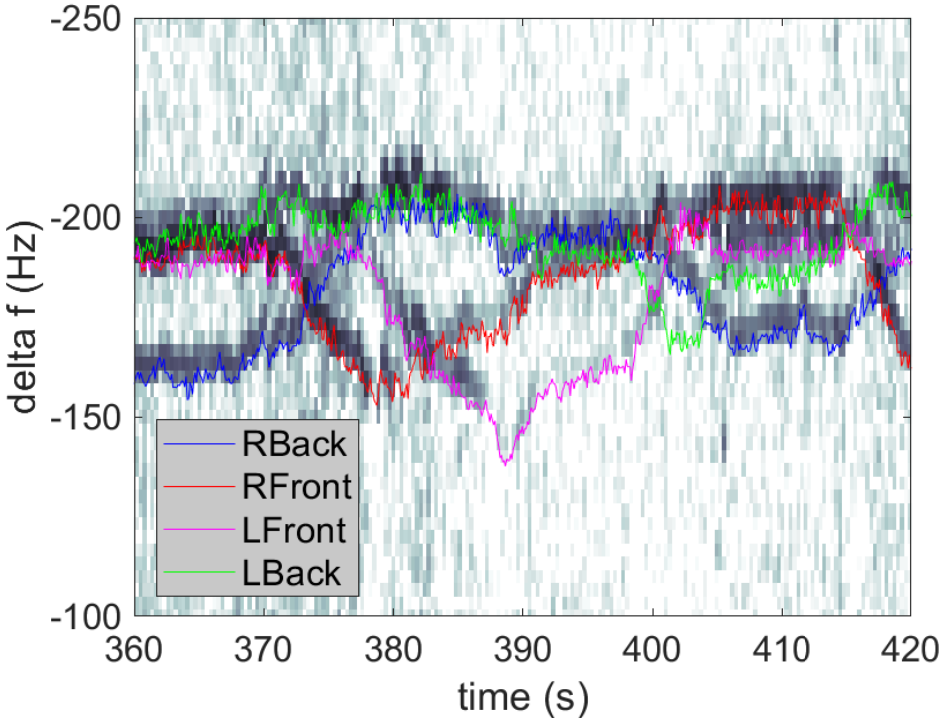


Figure 2.17 – Another comparison between theoretical data and real observation. The foreground lines correspond to theoretical impact of rotors. The background is a concatenation of spectra to obtain a long-time spectrogram. The coordinate of this trajectory is in Fig. 2.18.

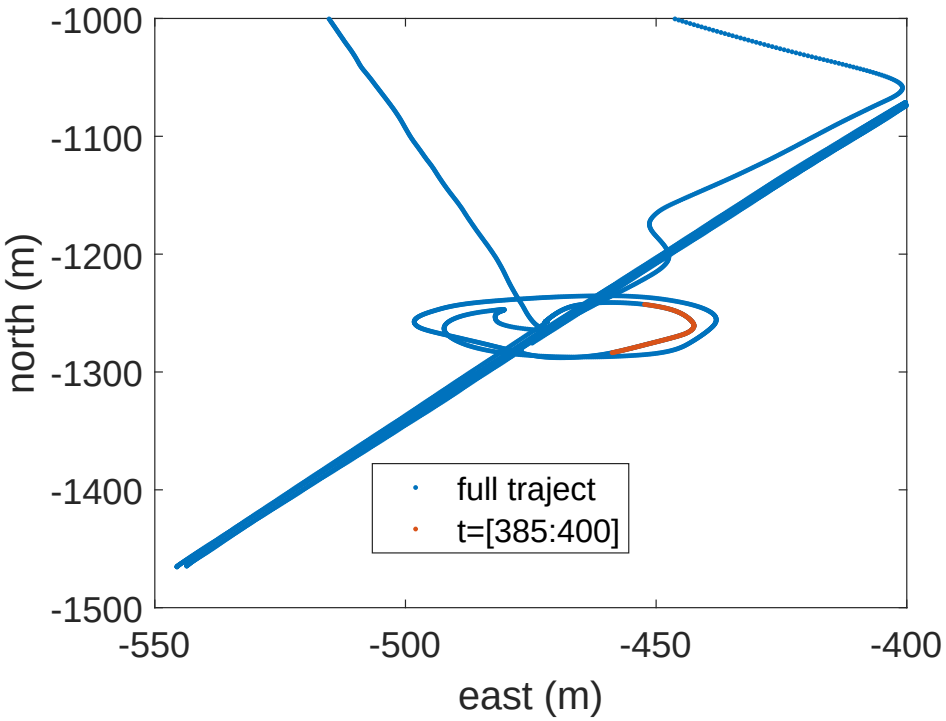


Figure 2.18 – Trajectory of the signals of Figs. 2.16 and 2.17.

3 - Space Representation of Micro-Doppler Signal for Drone Classification

With the data collected during the measurement campaign described in the previous chapter, we begin our drone recognition experiments based on micro-Doppler signals. Many formats are used by the scientific community to represent the micro-Doppler data. This diversity implies major difficulties to compare the different experiments reported in the literature as each study uses a particular format. Besides, as detailed in Section 3.1.3, the studies vary on many parameters, in terms of classifier techniques and in terms of data diversity. These differences also compromise the comparison of their results.

We identify the best space representation of micro-Doppler signals for drone recognition. To obtain this goal, we propose different use conditions as comparison criteria. Then, we evaluate the classification performance of each format in line with the function of its capacity to reach the highest classification rate with a given neural network on the different use conditions we identified. According to the experiments conducted, the recommended format is a spectrum issued from a long observation as its classification results are better for most use conditions.

3.1 . State of the Art

To begin with, we present the most common formats, followed by the concept of neural networks for classification. Eventually, we detail the most pertinent experiments present in the literature and their limitations justifying our study.

3.1.1 . Space Representations for Micro-Doppler Signal

We have selected the five most relevant space representations for the micro-Doppler analysis according to the literature : time signal after range compression $x(t)$, weighted spectrum of the signal *WSP*, cepstrum *CP*, spectrogram *SG*, and cadence velocity diagram *CVD*. They are defined below with continuous equations. We explain the reasons for this selection on two other common formats that we discarded.

All the selected formats but one are based upon FT (*Fourier Transform*). We did not observe any improvement with the use of zero-padding and thus did not apply it in the remainder. For all time windows h defined below, we chose Kaiser windows. We also investigated rectangular windows and did not observe any significant differences with Kaiser windows.

From the theoretical models formulated in Section 2.1, we have expectations for each format depending on whether the drone variations (rotor speeds, orientation,

etc.) are slow compared to the observation length. We denote targets respecting this hypothesis as relatively stationary targets. To explain each format in detail, the simulated signal corresponding to a target with one rotor is presented. This rotor is composed of two blades rotating at the constant speed of $5\,000$ RPM (*Revolution Per Minute*). The RCS (*Radar Cross Section*) of the blades and target cell is set to one (Fig 3.1a). An example of each format from our measurement campaign is also shown in Fig. 3.2 and compared with simulations in Section 3.4.6.

Time Signal After Range Compression - Raw Data

We denote as a time signal, $x(t)$, the complex signal corresponding to the target position, obtained with range compression. It is discretized at $\frac{1}{\text{PRF}}$ (*Pulse Repetition Frequency*). The other formats are produced by processing $x(t)$.

In simulation (Fig. 3.1b), we observe a high peak every 6 milliseconds, resulting from the position of the two blades flashing in the radar direction. We call them the blade flashes. Indeed as the rotor speed is 5 000 RPM, one rotor turns every twelve milliseconds and a blade flash occurs every half turn (two blades per rotor).

Spectrum

The weighted spectrum, *WSP*, of a signal $x(t)$ is its FFT (*Fast Fourier Transform*) on a time window h :

$$\text{WSP}(x(t))(f) = \text{FFT}\left(h(t)x(t)\right)(f). \tag{3.1}$$

The WSP format offers a good frequency resolution ($\frac{\text{PRF}}{N}$ for an N point signal) but it has no time resolution. For relatively stationary drones, peaks corresponding to multiples of twice the rotor speed (two blades per rotor) occur.

To remove potential biases, the main Doppler effect due to target speed is set to zero. To equilibrate WSP, all frequency shifts lower/higher than ± 4 kHz are removed. Thus, even if, as explained in Section 2.3, only two speeds were recorded for each drone, we expect that the training dataset would be as rich as if the same drones were moving at different speeds in the test dataset.

In Fig. 3.1c the simulated signal is in the WSP format. The main peak at zero frequency represents the main Doppler effect and thus the drone cell RCS, while the secondary peaks are associated with the rotor (blade flashes). The frequency gap between the main peak and the first secondary peak is $\frac{\omega_m}{N} = 167$ Hz.

Cepstrum

The cepstrum, *CP*, of a signal $x(t)$ is defined with the Inverse Fast Fourier Transform (*IFFT*) :

$$\mathrm{CP}(x(t))(\tau) = \mathrm{IFFT}\left(\log\left(|\mathrm{WSP}(x(t))(f)|^2\right)\right)(\tau). \tag{3.2}$$

Introduced in [5], CP is similar to WSP. The quefrencies τ, equivalent to frequencies f, highlight the signal echoes, especially when they are close one to another : the quefrency resolution is $\frac{1}{\mathrm{PRF}}$. For relatively stationary drones, peaks due to the rotors may be observed, as for WSP. As a cepstrum is similar to a spectrum, the vocabulary for this format is based upon the vocabulary for a spectrum [5].

This format, originally designed to bring out echoes for human visualization, is based upon the modulus information of WSP. In human visualization, when we compare CP with WSP, we also use only its modulus (Fig. 3.1). Recent studies [65], using only the modulus for complex signals in the Deep Learning context, have shown that the loss of information due to the modulus may deteriorate the neural network accuracy.

The CP format transforms the convolution in the time domain into addition in the quefrency domain. The authors [5] concluded that CP resists better to noise than WSP.

For the simulation (Fig. 3.1), every 6 ms, a blade flash is visible. The drone cell influence is measurable at 0 quefrency.

Spectrogram

The spectrogram, *SG*, is obtained from $x(t)$ with Short Time Fourier Transform (*STFT*) :

$$\mathrm{SG}\left(x(t)\right)(\tau, f) = \int_t h(t-\tau)x(t)e^{-2i\pi ft}dt. \tag{3.3}$$

This format is based upon a trade-off between the time and frequency resolutions. The short time window h allows one to concatenate profiles to obtain a time resolution, contrary to WSP. However, the frequency resolution is reduced. For an h window on M points, the time resolution is $\frac{M}{\mathrm{PRF}}$ and the frequency resolution is $\frac{\mathrm{PRF}}{M}$. We chose an h window of 2.5 ms.

As for WSP, the main Doppler effect is set to zero and all frequencies higher/lower than ± 4 kHz are suppressed.

For relatively stationary drones, we expect to see blade flashes. This effect can be put in light by window overlaps. For this reason, the overlap may improve the classifier quality even though it does not provide us with more information, increases the data size, and makes learning longer. After several experiments, we used an overlap of 60%, the lowest value giving the best classification results.

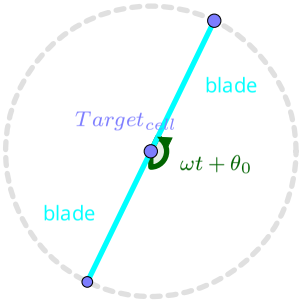
The spectrogram can be set with many h windows. Thus, if we concatenate spectra, we also obtain a spectrogram. To avoid any confusion, a spectrogram obtained in such a manner is called in this manuscript a long-time spectrogram. Such a format cannot be compared to the short-time spectrogram as its observation

time would be longer than 300 ms. Contrary to the long-time spectrogram, the currently defined spectrogram has an observation time shorter than the revolution time of the rotors of the studied drones.

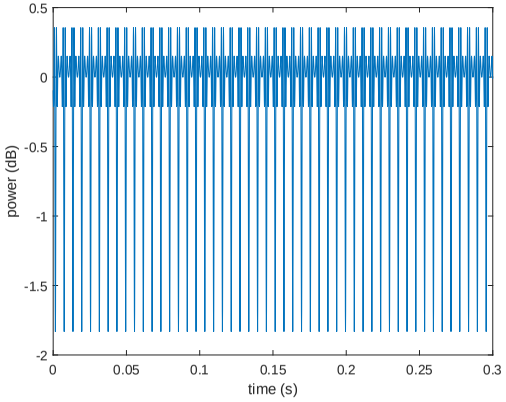
The spectrum can be seen as an extreme case of this one and intermediary formats of any time window h could be created. We assume that for other time windows shorter than the revolution time, the results would be similar to the results of the format defined here. Whereas, for a time window longer than the revolution time, the results would be similar to the results of the spectrum format. The other extreme with no time window h corresponds to $x(t)$ format.

We choose a time window h of 2.5 ms, a common value set before any classification test. It has not been optimized to avoid any over-fitting on the test dataset.

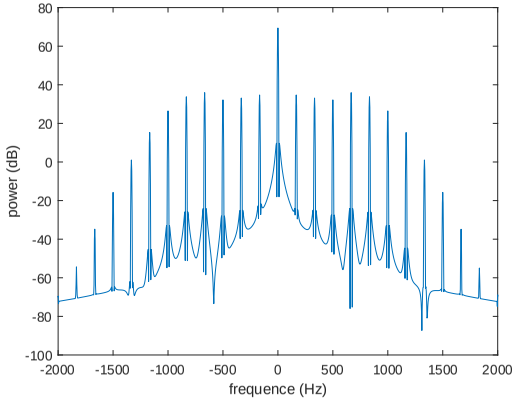
In a simulated signal in this format (Fig. 3.1e), the blade flashes occurring every 167 Hz are less visible due to the influence of the drone cell which hides the central frequencies of the SG profile. Indeed, contrary to WSP where the drone cell influence is thin in the center of the profile, for this format the drone cell is larger. It can be shortened by increasing the time window which would make this format closer to WSP but also decrease the time resolution. The visibility of the micro-Doppler effects for this format depends strongly on RCS of the drone cell.



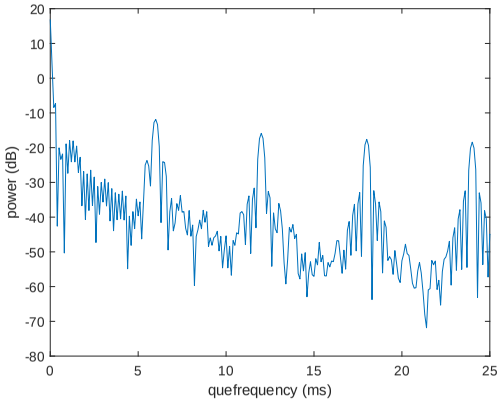
a - Simulation
Rotor at 5 000 RPM



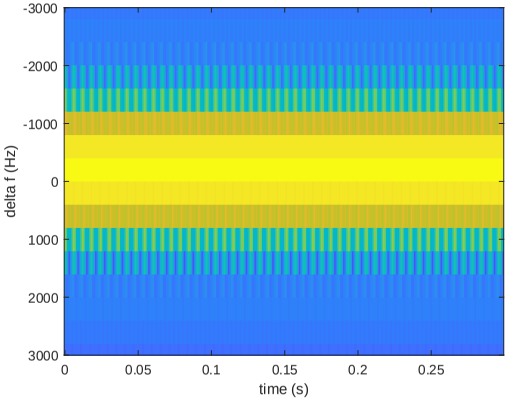
b - $x(t)$
Time signal



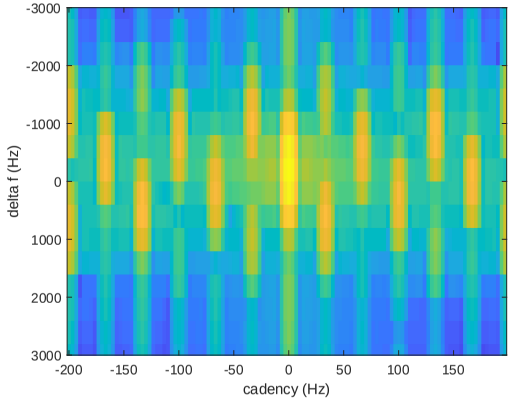
c - WSP
Spectrum



d - CP
Cepstrum



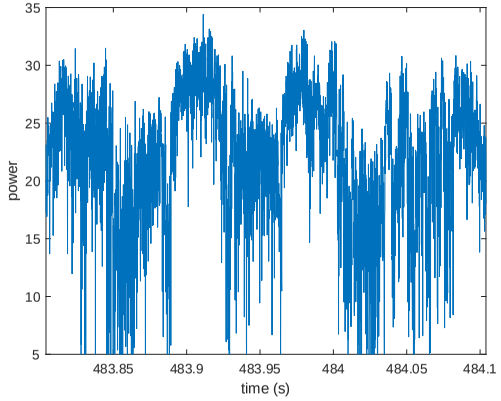
e - SG
Spectrogram



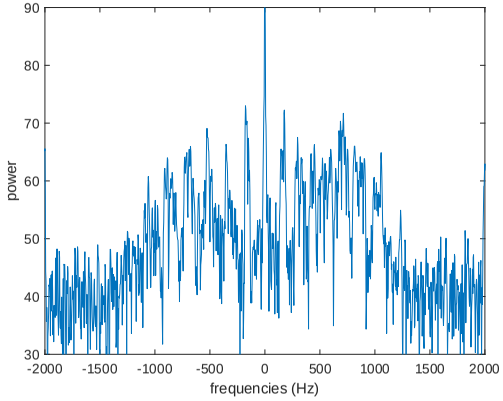
f - CVD
Cadence Velocity Diagram

Figure 3.1 – Simulated signature of a rotor at 5 000 RPM observed during 300 ms expressed in the different formats. Data is displayed in decibel (dB), zooms are added for better visualization.

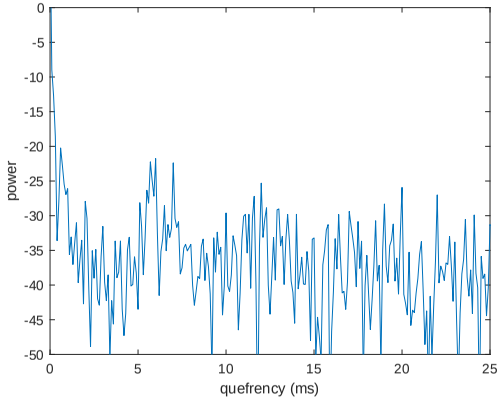
a - Drone
DJI Phantom 4 Pro



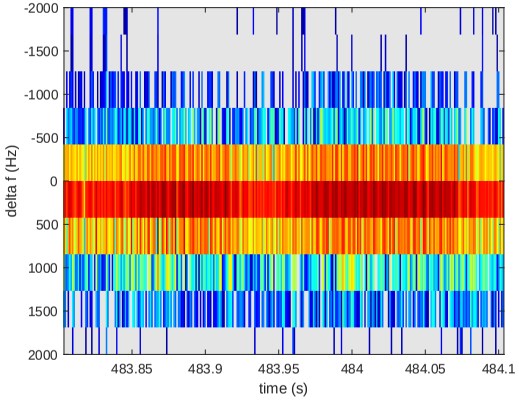
b - $x(t)$
Time signal



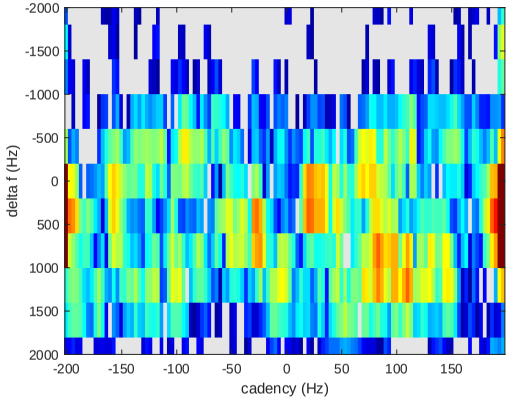
c - WSP
Spectrum



d - CP
Cepstrum



e - SG
Spectrogram



f - CVD
Cadence Velocity Diagram

Figure 3.2 – Signature of a DJI Phantom 4 Pro observed during 300 ms with the different formats. Data is displayed in dB, zooms are added for better visualization. Data depicted corresponds to a random part of one trajectory (from 483.8 to 484.1 seconds in Fig. 2.16.

Cadence Velocity Diagram

The Cadence Velocity Diagram, *CVD*, is produced by performing FFT for all SG frequencies :

$$\begin{array}{rcl} \mathrm{CVD}\left(x(t)\right)(\nu, f) & = & \mathrm{WSP}\left(\mathrm{SG}\left(x(t)\right)(\tau, f)\right)(\nu) \\ & = & \int_\tau h(\tau)\mathrm{SG}\left(x(t)\right)(\tau, f)e^{-2i\pi\nu\tau}d\tau. \end{array} \quad (3.4)$$

This format highlights the frequency periodicity better than SG upon which it is based. It is particularly appropriate for substantially long observations (at least several seconds).

For CVD on N points, produced from SG made with an h window of M points, the cadence resolution is $\frac{M\mathrm{PRF}}{N}$ while its frequency resolution is $\frac{\mathrm{PRF}}{M}$, identical as the SG one. This format is thus convenient for a long observation as it requires a great number of profiles to execute the second FFT.

The interpretation of this format is not obvious. In the example produced from a simulated signal (Fig. 3.1f), we see the influence of the drone cell at the zero cadency. This influence is reduced for a long observation. For a longer observation, we can increase the length of the time window h to avoid aliasing. As the sampling frequency of the second FFT is $200\,\mathrm{Hz}$, we are under the Nyquist frequency and so aliasing can be observed. For example, the signal measured at $33\,\mathrm{Hz}$ is, in reality, the fifth-order at $166.7 \cdot 5 = 833\,\mathrm{Hz}$. For a scenario with a long observation, if the relatively stationary hypothesis still holds, this format is easier to interpret.

Other Transformations

Numerous transformations can be made on radar signals. Apart from the aforementioned formats, other transformations such as Cepstrogram and Cohen transformation might be taken into consideration. We present discarded formats to explain the reason of their rejection.

Cepstrogram is for a cepstrum the equivalent of what is a spectrogram for a spectrum. It consists in a trade-off between time and quefrency resolutions. We did not select this format as we expected similar results for a spectrogram. Moreover, this format is rarely used in literature.

However, Cohen transformations frequently appear in the literature, most particularly the SPWVD format (*Smooth Pseudo Wigner-Ville Distribution*) [60, 10, 32], which consists in bilinear transformations :

$$C(\tau, f) = \int_u \int_t \int_\theta \Phi(\theta, t) x\left(u, +\frac{t}{2}\right) x^*\left(u + \frac{t}{2}\right) e^{-i(\theta\tau - 2\pi f t + \theta u)}\, du\, dt\, d\theta \quad (3.5)$$

These bilinear transformations avoid choosing a time window length. Thus, no trade-off has to be taken between the frequency and time resolutions as if we were computing the Short Time Fourier Transform with all the possible time window

lengths, instead of having to select an adapted one. As detailed in [10], SPWVD is based upon auto-correlations and a kernel $\Phi(\theta, \tau)$ defines a specific transformation chosen. An example of the SPWVD format is given in Fig. 3.3. The peak at 167 Hz explained above for the spectrum is also observed for the same reason.

Bilinear transformations have a higher joint time-frequency resolution than any linear transformation. The kernel Φ is chosen to reduce the common bilinear transformation problem of high cross-interference terms while preserving as much as possible the resolution. We did not use these formats for classification purposes due to one principal reason : the size of the resulting data is too large.

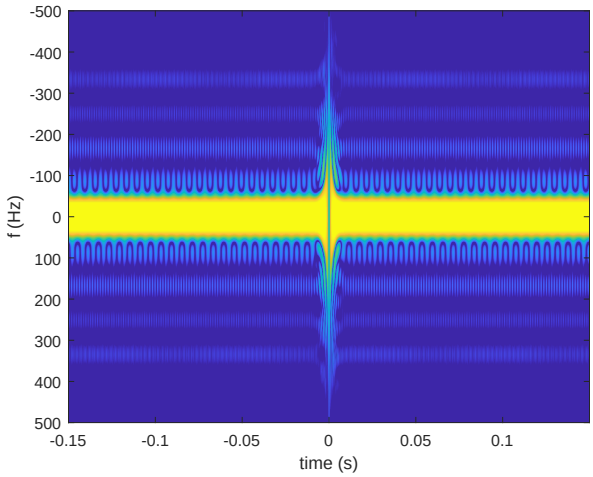


Figure 3.3 – Simulated signature of a rotor at 5 000 RPM observed during 300 ms with the SPWVD format. Data are displayed in dB, zooms are added for better visualization.

More specifically, let assume a signal composed of n points. Then the size of the data after a linear transformation is a linear function of n. For example, for a spectrogram with an overlap of $a\%$ and a zero-padding $\times b$, assuming that the high frequencies are kept, the resulting data has a dimension :

$$\frac{n}{1 - \frac{a}{100}} b. \tag{3.6}$$

Without zero-padding and without overlaps larger than 90%, the size of the data is at most ten times greater than the raw data size. For a bilinear transformation, the size of the resulting data is a function of n^2 increasing substantially the size. Such an amount of data becomes difficult to process or store. The only way to have data of a reasonable size would be to reduce the resolution by decimating (both in the mathematical and the usual sense) the signal. We gave up this solution because we did not find any acceptable compromise.

In addition to the data size problem, this format is hard to compute and takes processing time which raises concerns for any potential on-board system. For these simple but inevitable reasons, this format was not used.

We highlight that despite these drawbacks, some articles continue to promote this format without using it for classification, studying only the resulting output on a few examples. We do not think that any research could go further with the SPWVD format without resolving these issues.

3.1.2 . Neural Network for Classification

The real radar signal is complex to interpret as the electromagnetic models have limitations. We assume that Deep Learning recognition methods can efficiently tackle this issue. To begin, we present briefly neural networks, outlining their global behavior and illustrating it with architecture examples.

Preliminaries

A neural network (*NN*) can be seen as a black box function receiving an input x and producing, in a computationally reasonable time, an output $f(x)$, Fig. 3.4. This output depends on the parameters, also called weights, of the network. Neural networks are used to approximate functions known only on a specific set of examples, called a training dataset. Parameters of a network must not be confused with hyperparameters (also called meta-parameters). Parameters evolve during learning thanks to the back-propagation process explained below, while hyperparameters define a fixed setup of the network such as the number of iterations for the training or the layer number of the network.

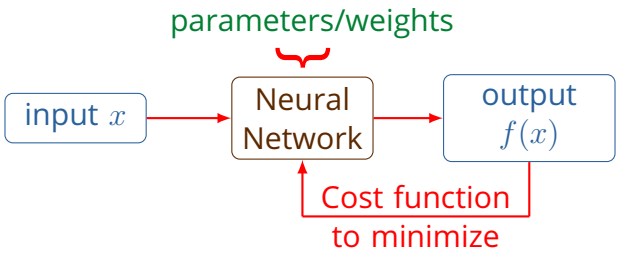


Figure 3.4 – Schema of a neural network.

The algorithm is composed of three phases :

Step 1, Initialization The NN is initialized with random parameters and for any input x, it produces a random output $f(x)$ which is certainly not the desired result.

Step 2, Learning/Training For each input x of the training dataset, a cost function is computed to evaluate the quality of the output $f(x)$ produced by the NN. The NN adapts its parameters by a descent algorithm to reduce this cost.

Step 3, Test The NN parameters are fixed. The network is ready for unseen examples (testing datasets) to assess its quality.

A cost function called also, an error, or a loss measures the difference between the obtained result $f(x)$ and the desired result y. By a descent algorithm, the NN modifies its weight to reduce this cost as the objective is to produce a result $f(x)$ closer to y. After the weight modification at an iteration, the NN corresponds to a new function f. With the same input x, the new output $f(x)$ would give a value closer to the desired result y. However, in a stochastic descent, a new x is injected into the network at each iteration. We assume that the desired function f can be obtained for the entire distribution of x.

For example, for the MNIST classification (Fig 3.5), the output $f(x)$ should give the digit represented on the image. The input is a 28×28 gray image. The output is a vector of size ten. The highest output is considered as the digit recognized by the NN. In Fig. 3.5, the NN sees 1 (highest output) but should see 4. Thanks to a human annotation, as x comes from the training dataset, the label y of x is accessible. The labels are information on data. For MNIST, the label "[0 ;0 ;0 ;0 ;1 ;0 ;0 ;0 ;0 ;0]" corresponds to a four while the label "[1 ;0 ;0 ;0 ;0 ;0 ;0 ;0 ;0 ;0]" corresponds to a zero. In this example, the components of the network output are limited between zero and one. In classification problems, the output is often also normalized to have components corresponding to probabilities : between zero and one and with a sum equals to one.

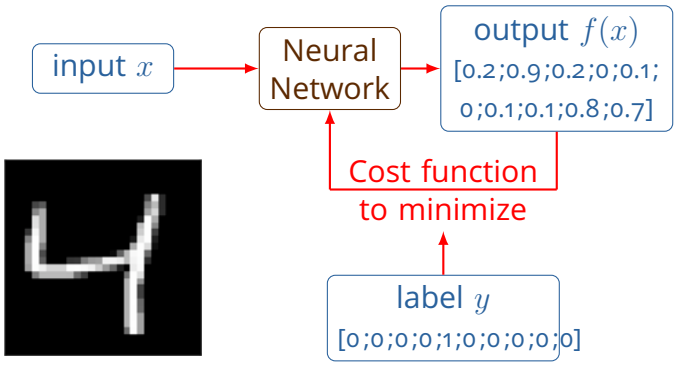


Figure 3.5 – Classification example MNIST dataset

In a more general case, at each iteration, a subset of the training dataset called a batch is taken. The descent algorithm is made on the mean cost of the batch. Then at the next iteration, a new batch composed of different data is taken. When the entire training dataset has been used, the first epoch has been completed. The

data is shuffled and new batches can be drawn for the next epoch. This approach implies the following equation :

$$|d| \cdot \text{epoch} = |\text{batch}| \cdot \text{iteration}, \tag{3.7}$$

where d is the training dataset.

There are many existing architectures of neural networks and new architectures appear. These architectures have to respect the following conditions :

— For any input x, the result $f(x)$ has to be computed in a short time (fast forward direction).

— For any input x, the derivative of the cost of $f(x)$ according to any weights of the NN has to be computed in a reasonable time by using back-propagation.

We detail two types of NN : full-connected and convolutional. The notation and diagrams are mostly based on the online book [42].

Full-connected Neural Network

A fully connected network also called dense network or MLP (*Multi-Layers Perceptron*) is a network composed of a succession of neuron layers. The number of layers L is the depth of the network. The neurons of the first layer receive the input data $x = a^0$, while the output is retrieved from the last layer, $f(x) = a^L$. The input of each neuron of any layer is the output of the neurons of the previous layer. We start with the concept of a neuron.

Let consider a neuron j on the layer l, it receives its input a^{l-1} from the layer $l-1$. Naturally, a^l is neuron's output, injected into the neurons of the next layer $l+1$. The neuron is composed of two parts, Fig. 3.6. The weights of this neuron are the ω_j^l.

This neuron computes a linear combination of entries $z_j^l = \sum_{k=0}^{N_{l-1}} a_k^{l-1}\omega_{jk}^l$, where N_{l-1} denotes the number of neurons of the layer $l-1$. To have affine and not only homogeneous linear functions, the constant input $a_0^{l-1} = 1$ is added.

Next, the neuron performs an activation function σ to produce a non-linear output $a_j^l = \sigma\left(z_j^l\right)$. The activation function and its derivative must be easy to compute. The non-linear operation prevents a fully connected network from reducing to a single neuron.

The combination of the different layers leads to a complicated function. A full-connected network example is given in Fig. 3.7. This network has a depth of two. It takes 2-dimensional vectors for input (a_1^0, a_2^0), and output (a_1^2, a_2^2) :

$$\begin{aligned} &\mathbf{R}^2 \to \mathbf{R}^2 \\ &(a_1^0, a_2^0) \to \left(\sigma\left(\sum_{k=0}^{2}\sigma\left(\sum_{r=0}^{3} a_r^0\omega_{kr}^0\right)\omega_{1k}^1\right), \sigma\left(\sum_{k=0}^{2}\sigma\left(\sum_{r=0}^{3} a_r^0\omega_{kr}^0\right)\omega_{2k}^1\right)\right). \end{aligned}$$

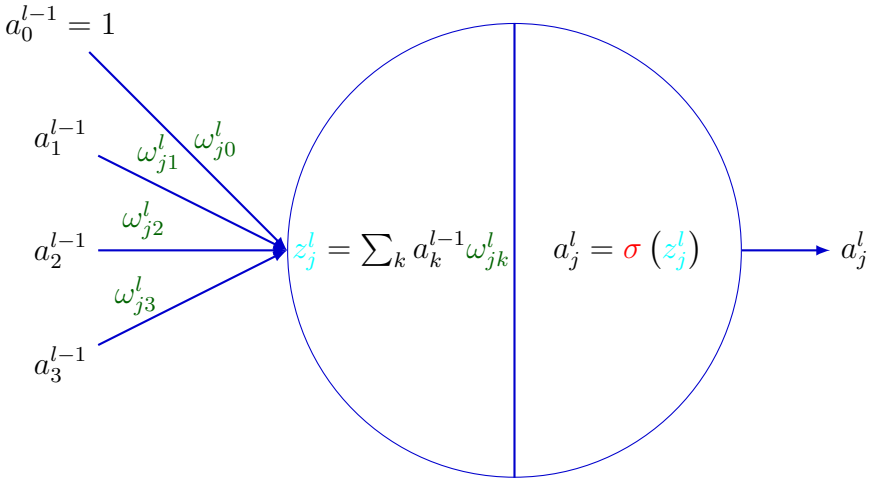


Figure 3.6 – Neuron j of layer l with σ the activation function.

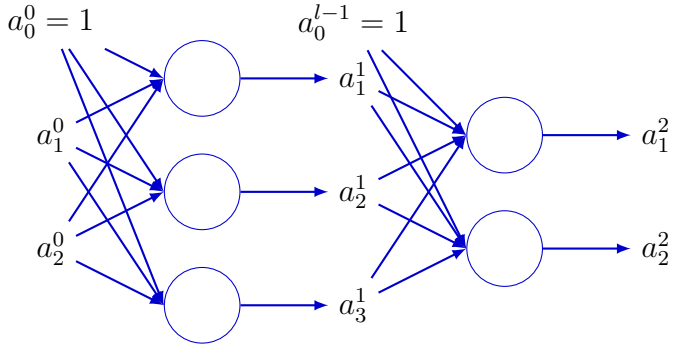


Figure 3.7 – Full-connected network example.

The full-connected network will be a base to introduce the back-propagation, the main component of the learning phase. The back-propagation needs a computable cost function C on all the inputs of the training dataset. Our goal is to minimize this cost. For example, for recognition tasks, the cost function usually compares the output produced $f(x)$ with the label y. As x comes from the training dataset, its associated label y is known and the cost function can be computed.

To reduce this cost, we need to compute $\frac{\partial C}{\partial \omega_{jk}^l}(x)$ for each input k of each neuron j of each layer l.

To begin with, we observe that :

$$\frac{\partial C}{\partial \omega_{jk}^l} = \frac{\partial C}{\partial z_j^l} a_k^{l-1}.$$

The value a_k^{l-1} is given, so computing $\delta_j^l = \frac{\partial C}{\partial z_j^l}$ simplifies the process with the absence of the dependence upon k.

Coefficients δ_j^l are obtained by back-propagation. For the last layer ($l = L$), these coefficients can be computed due to the fact that C can be computed. Then, for any other layer l, δ_j^l can be obtained thanks to δ_j^{l+1}. The equations are presented below :

$$l = L, \delta_j^L = \frac{\partial C}{\partial a_j^L}\frac{\partial a_j^L}{\partial z_j^L} = \frac{\partial C}{\partial f(x)_j}\sigma'(z_j^L)$$
$$l < L, \delta_j^l = \sum_k \frac{\partial C}{\partial z_k^{l+1}}\frac{\partial z_k^{l+1}}{\partial z_k^l} = \sum_k \delta_k^{l+1}\omega_{kj}^{l+1}\sigma'(z_j^l).$$

From these formulæ, we can already remark some typical NN problems such as the vanishing gradient. Indeed, as the coefficients of layer l are computed from the linear combination of coefficients of layer $l+1$ and as these values are often close to zero, they tend to vanish for the first layers. Slow changes on the initial layers compromise their learning.

Convolution Neural Network

The full-connected architecture is simple and may reach a reasonable performance. It suffers, however, from several disadvantages such as a complete translation dependency. If we shift the original data by one column, the fully connected neural network produces a different result, despite the fact that the new input is very similar to the original one. We depict a more advanced architecture : CNN (*Convolutional Neural Network*).

In CNN, each neuron is seen as a filter looking for a specific pattern of small size (typically 3×3, 5×5 pixels). This pattern is searched all along with the data by convolution. The result is denoted as a feature map. Each layer is composed of several feature maps. For example, for images, the first layer is composed of three feature maps (Red, Green, and Blue channels). From these three pieces of information, each filter searches the presence of its pattern. So, a 5×5 filter at the first layer is in reality a $5 \times 5 \times 3$ filter as the research is made along all channels.

The output of the first layer is thus composed of a concatenation of different feature maps corresponding to simple patterns from the data. Then, the next convolutional layer looks for patterns on these feature maps corresponding to more advanced patterns from the original data (patterns of patterns).

Details on the principle or the resulting equations can be found in the online book [42].

GoogLeNet

The GoogLeNet [59] is a 22-layer CNN. It won the ImageNet Large-Scale Visual Recognition Challenge in 2014 (ILSVRC14).

The GoogLeNet is based on specific convolutional layers called inception layers. The Fig. 3.8 reproduces its global topology. Here, we lay out only the main concepts such as inception layers.

Inception layers are based on the concatenation of filters of different sizes at the same layer : 1x1, 3x3, 5x5. This diversity has been developed to help the network to catch different kinds of features in the data. The largest convolutions (3x3, 5x5) are preceded by a 1x1 convolution to reduce the shape and thus reduce the computational time. This diversity should help a CNN to treat various data distributions.

GoogLeNet is composed of several classifiers. It was originally designed to combat the vanishing gradient problem (Section 3.1.2). Nevertheless, according to [59], the effect of the auxiliary classifiers are minor and introduce additional weighting to set up : the weighting of the different loss terms in the global loss.

Despite its original goal, GoogLeNet is often successfully used in practice for other purposes and it has become a standard neural network. According to Google Scholar, the article [59] has been quoted about $30\,000$ times in six years. Among the numerous articles, we spotlight micro-Doppler classification articles [30, 57].

In [30], the authors use GoogLeNet to compare the performance of different signal representation : the SG format, the CVD format, and a designed format constituted of the combination of SG and CVD. This combination was performed by simply juxtaposing the two representation. They adapted their formats to match the input shape of GoogLeNet. The modulus of the signal was repetitively put into the three RGB channels.

In [57], the micro-Doppler signals studied correspond to human movements. The goal is the recognition of the gait of each individual. The signals are obtained at different frequencies and environmental conditions but the analysis tools remain pertinent for us. The full network of GoogLeNet was not directly used. The authors of [57] designed their own network inspired by GoogLeNet, using notably the concept of inception layers to create a smaller network.

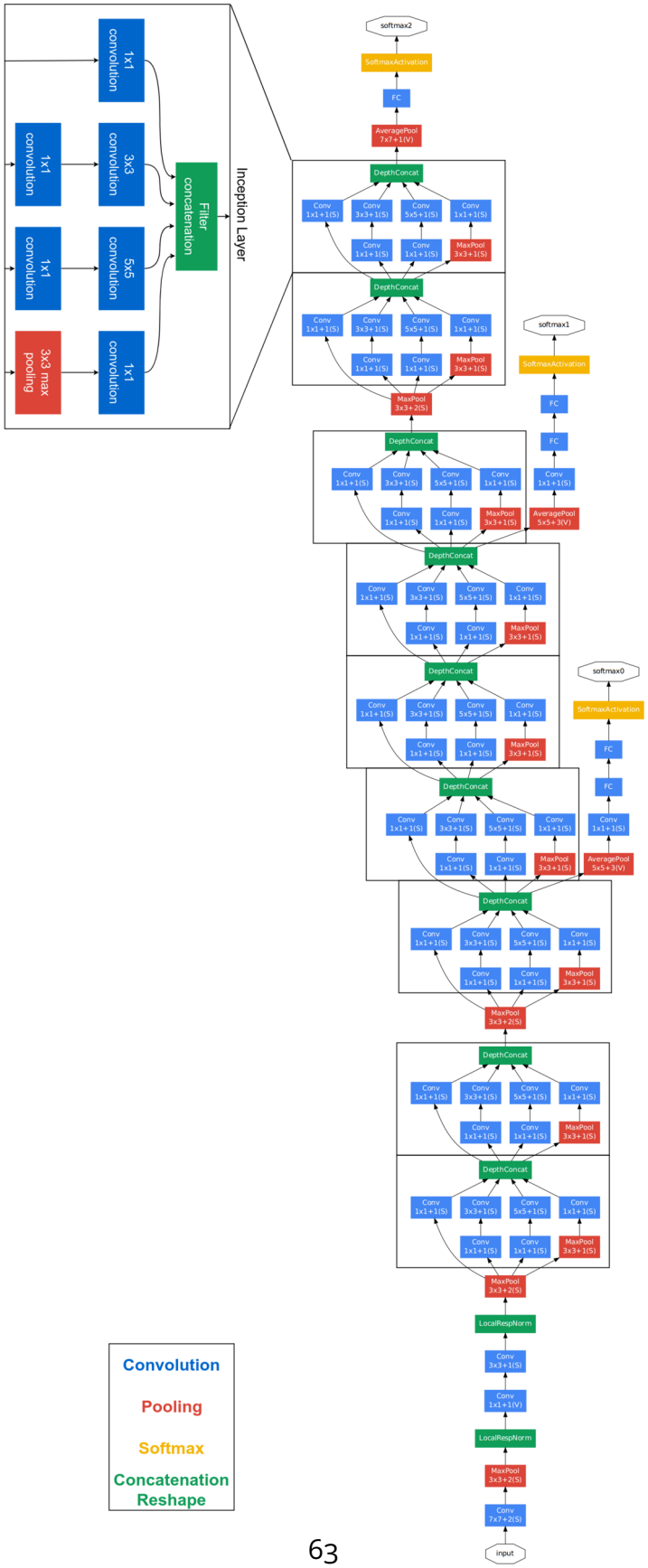


Figure 3.8 – GoogLeNet architecture taken from [59]. The two auxiliary classifiers softmax1 and softmax0 are optional and were not used in our study.

3.1.3 . Classification of Micro-Doppler Signals

Numerous experiments show the pertinence of micro-Doppler signal for human movement discrimination [31, 56, 26, 61, 57, 57] and drones [19, 41, 21, 56, 41, 7, 30, 46, 12]. As already said, the experiments differ considerably in classifier methods and data diversity.

Classifier Methods

For the micro-Doppler classification, Bayes methods [19, 41] or SVM *Support Vector Machine* [41, 21, 56] has been discussed recently. The latest progress in machine learning, especially CNN, has revealed a new line to study this phenomenon as the neural network extracts the features itself. Studies show the advantages of neural networks in the case of micro-Doppler profiles for drone classification [7, 30, 46, 12, 6] or human movements [31, 56, 26, 61, 57].

These studies vary in terms of classifier algorithms and statistics used for the comparison. For example, the classification results presented in [6] are averages over only five runs, without considering the large confidence intervals. Generally speaking, the result interpretations of the literature are compromised by the lack of statistically robust tools.

Data Diversity

In terms of diversity of data, we observe major differences across studies : simulated or real data, set of drones, various trajectories, classification goals, measurement conditions (radar, range, observation time, environment, etc.). Hence, these differences should have an impact on the results.

Despite the limitation of simulations, detailed in Chapter 2, simulations are often used in studies [21, 7, 12] because of the difficulty to acquire real data. As said in Section 2.2.1, these studies do not exploit the log files of the drones [7, 12]. Thus, the parameters, such as the evolution of rotor speed, needed to be set in electromagnetic simulations are arbitrarily chosen. The risk is that they are unrealistic compared to the real behavior of drones. We thus have to stay cautious about the outcome of these studies. In [21], only bird signals are simulated since acquiring bird signals is even harder than acquiring drone ones, let aside bird logs.

Each measurement campaign was made with its own set of drones (at most 7 drones). Among the different drones measured, we observe that the DJI Phantom, the largest-selling drone in the market, is mostly used [46, 30, 19, 21, 50, 17].

As detailed in Chapter 2, the signal depends mainly on the motion of the drone. The majority of studies composed of real data consider only simple movements such as hovering flies [46, 30, 19, 21] or rectilinear movements [46, 50].

The classification goal also differs between studies. Many of them classify drones by group depending on their characteristic such as the rotor number

[7, 12, 41] while others have more sophisticated objectives such as a classification of the same drone with different payloads [46].

One of the main limitations of studies is the radar-target distance. The majority of studies are made at distances lower than 150 m [21, 41, 19, 30, 50]. This scenario is far from the classic scenario of several kilometers [17].

3.2 . Use Conditions

The performance of classification in function of the input format is compared according to different criteria. We describe the use conditions we identified to establish the criteria. The first case is taken as a reference case. The following ones are robustness to noise and short observation duration. The last two use conditions, corresponding to the facility of training of the CNN and the human visualization, have their own dedicated evaluation.

We aim at observing the impact of the format chosen upon the quality of the CNN classification. We are interested not only in the classification rate under every condition but also in the gap between the reference and the current use condition (indicated in the Δ column in the tables in Section 3.4).

3.2.1 . Reference Case

The reference case consists in classifying drones into 5 classes using data we collected. The reference observation time is 300 ms. It corresponds to the minimum duration needed to differentiate two rotors within 100 RPM for CP and WSP under the hypothesis of relatively stationary targets.

The reference SNR (*Signal Noise Ratio*) is between 30 and 50 dB, depending on the drone and the fragment of its trajectory (altitude, speed, orientation of the drone, etc.).

3.2.2 . Robustness to Noise

The reference case provides us with a signal with a good SNR for each target because our measurement campaign was mostly done with a sky background. To explore more difficult scenarii, we assess which format allows a satisfactory classification regardless of the SNR deterioration.

To achieve this goal, we added a white Gaussian noise to the reference data. The noise level is absolute and thus independent of the drone type. Consequently, it has a different impact on each signal section according to its SNR. The resulting dataset SNR is between 10 and 30 dB.

3.2.3 . Short Observation

The radar observation time of a non-cooperative target is difficult to estimate. Indeed, depending on the scenario, the radar may track any potential target for a quite long observation time (even for few seconds or more) or be in a surveillance mode of a large area (azimuth angles). In the latter case, the radar tends to turn

on itself to check for every azimuth and thus is not able to observe at a precise azimuth for a long time (40 ms or less). Other more complex systems use several radars to increase the observation time.

We chose a short observation time to study the performance of the surveillance model described. However, we note that it would also have been interesting to study longer observation time. As the data is made with a split of the trajectories into chunks of equal size, the number of data pieces depends on the observation length. We made this choice to avoid being forced to work with very small datasets which would have compromised the result interpretation.

Some formats are conceived to be more adapted to shorter/longer observations. For this use condition, the observation time is shortened in comparison to the reference case to assess observation duration robustness. The shorter observation length chosen is 36 ms which corresponds to three turns for a rotor at 5 000 RPM.

3.2.4 . Discrimination Drones/Birds

We distinguish drones from birds to determine which format is more adapted to this classification goal.

For this use condition, we gathered targets into two categories : birds and drones of all five types. As this dataset is different from the one used for the reference case, we limit ourselves to a result discussion.

3.2.5 . Facility of Training

Certain formats change data dimension or shape. These modifications impact substantially the time needed to execute each CNN iteration. They may also improve the contrast between data and thus reduce the number of iterations needed to reach the maximum accuracy value.

To evaluate formats according to this criterion, the time required for one back-propagation is computed. We also determine the time needed to reach 95% of the maximum accuracy.

3.2.6 . Human Visualization

The use of a format for human readability is not our main interest. However, as it can be an important criterion in other studies, we present a subjective evaluation of the facility to interpret each format. We compare observations from simulated signal (Figs. 3.1 and 3.10) with the observation with real signal (Figs. 3.2, 2.16, and 3.9).

3.3 . Experimental Protocol

3.3.1 . Measurement Campaign

The data used comes from the measurement campaign. The different targets were measured in various configurations. The different trajectory patterns (up/down, rectilinear/circular movement, stationary/rotation around its center,

and free-fly) are present in each trajectory and both in the training and the testing dataset.

3.3.2 . Deep Network Configuration

The classification results are obtained with GoogLeNet (Section 3.1.2). The network has been chosen because it has been used in related works [30, 57]. As this network has been used in numerous studies on many other domains, we assumed it to be standard enough to have generalizable results. We insist on the fact that our goal is to compare the performance of the different formats and not to obtain the best classification results possible. To simplify the problem and avoid any over-fitting we removed the auxiliary classifiers as they have a minor impact.

Each trajectory is split into chunks of duration equal to the observation time chosen. Each chunk is represented by all formats under study and sent as an independent input to the network. The training and testing sets contain trajectories collected on different days to ensure data independence. There are 36 730 trajectory chunks in the learning set and 4 670 in the testing one. All chunks correspond to nearly three and half hours of accumulated measurements.

We stress that without day separation between the testing and training set, the classification accuracy reaches more than 98% in all configurations (even 100% for $x(t)$), much too optimistic. We strongly recommend that experiments should be conducted with the day separation. As we were restrained to collect the data at one location, we expect poorer results for a training and testing sets conducted at different locations.

We use a batch size of 256 chunks and fix a dropout of 0.6 on the first fully connected layer. In every configuration, we run 50 learnings of 25 000 iterations. The test accuracy is assessed every 200 iterations. The classification results presented correspond to the maximum of the mean accuracy and its associated confidence interval at 95%. The time/step values are measured on the same computer (GPU card : NVIDIA Tesla P100-PCIe-12GB).

For unidimensional (1-D) data, the network architecture is adapted to be 1-D (filters : $5 \times 5 \rightarrow 5 \times 1$, etc.). The real and imaginary components of each format are injected into the CNN as two color channels. Each data x is normalized by the formula :

$$\frac{x - m}{M - m}. \tag{3.8}$$

with M the maximum and m the minimum modulus on the training dataset. All data has thus a modulus between zero and one without removing energy information. Other normalizations we tried, based on removing the energy information (computing M and m of each signal, for example), resulted in performance degradation.

The same datasets are used in all cases except for the discrimination between birds and drones. Because of the small quantity of bird data gathered, we had to

severely reduce the amount of drone data to have balanced datasets. We remind that as for drones, testing and training datasets are composed of data using the same drones and the same birds, measured on different days.

3.4 . Results

We present the classification results of each format under the different use conditions.

3.4.1 . Reference

Format	Reference [%]
$x(t)$	75.8 $\pm$ 1.55
WSP	**98.1 $\pm$ 0.09**
CP	97.4 $\pm$ 0.11
SG	92.6 $\pm$ 0.32
CVD	94.5 $\pm$ 0.17

Table 3.1 – Reference use condition.

The network performs significantly better for the WSP and CP formats than the others (at least 3 points more). These two formats are based upon a good frequency / quefrency resolution despite the absence of time resolution. The CP results are 0.7 points lower than WSP. The differences might be explained by the loss of information while taking the modulus.

The network reaches 94.5% of accuracy for the CVD format which is 1.9 points above the SG performance despite the short observation. The second FFT made on SG has a positive impact on the classification performance.

Feeding the network with $x(t)$ produces the worst result (75.8%). Other formats project the information contained in $x(t)$ into another space. Such an operation makes the differences between drones more contrasted.

3.4.2 . Robustness to Noise

Format	Reference [%]	Noise added [%]	Δ
$x(t)$	75.8 $\pm$ 1.55	72.6 $\pm$ 1.50	3.2
WSP	98.1 $\pm$ 0.09	**92.7 $\pm$ 0.15**	**5.4**
CP	97.4 $\pm$ 0.11	80.9 $\pm$ 0.30	16.5
SG	92.6 $\pm$ 0.32	73.4 $\pm$ 0.42	19.2
CVD	94.5 $\pm$ 0.17	79.2 $\pm$ 0.26	15.3

Table 3.2 – Robustness to noise.

Our CNN obtains the best results with noisy data when using the WSP format (92.7%). This result is more than 10 points higher than for any other format. The WSP format also resists the best to noise as its performance is only 5.4 points less than for the reference. In contrast, losses observed in classification with the use of other formats after noise injection are about 15 points.

The classification with the CP and CVD formats has similar results, around 80%. Surprisingly, the results with CP, based as WPS on long integration, are not better than the WSP one. This is astonishing because CP was created to resist better than WSP to noise [5].

Similarly to the reference, the classification performance for SG is relatively low (73.4%) and worse than for CVD. The gap between the results obtained with these two formats has even increased from 1.9% to 5.8%. A possible explanation is the noise subduction by the second FFT in CVD.

The classification with the $x(t)$ format is poor.

3.4.3 . Short Observation Duration

Format	Reference [%]	Short signal [%]	Δ
$x(t)$	75.8 $\pm$ 1.50	67.1 $\pm$ 1.01	8.7
WSP	98.1 $\pm$ 0.09	93.7 $\pm$ 0.12	4.4
CP	97.4 $\pm$ 0.11	**94.0 $\pm$ 0.12**	**3.4**
SG	92.6 $\pm$ 0.32	87.3 $\pm$ 0.23	5.3
CVD	94.5 $\pm$ 0.17	79.3 $\pm$ 0.26	15.2

Table 3.3 – Short observation.

Although CP and WSP have comparable classification values (around 94%), CP withstands better than WSP to the reduction of observation time (3.4% instead of 4.4% in Δ column). Shortening the observation time diminishes the number of peaks produced by the rotors. The remaining peaks, however, are sufficient for the CNN to discriminate the drones.

Once again, the classification with SG is low compared to WSP and CP with only 87.3% of success rate and a degradation of 5.3 points due to the short observation.

The CVD classification performance decreases considerably (three times as much as for any other format). It confirms the statement from Section 3.1.1 : this format is made for long observations. For short ones, the second FFT is performed on an insufficient profile number and produces thus a poor resolution in terms of frequency cadence. For observations longer than the reference case, we expect that its performance might increase.

For each format, shortening the observation time decreases the network performance ($x(t)$ too low to be considered). Despite that, the short observation duration is sufficient to capture several rotor speeds, not all the characteristics

needed to identify the target are present. We believe that the dynamic evolution of the drone is used to classify. In a real scenario, the observation time depends in particular on the environment (if the drone can be hidden by obstacles). Thus, the latter has a strong impact on the results.

3.4.4 . Discrimination Drones/Birds

Format	Drones/Birds [%]
$x(t)$	75.7 ± 2.25
WSP	$\mathbf{97.3 \pm 0.36}$
CP	96.7 ± 0.25
SG	86.8 ± 0.65
CVD	94.2 ± 0.44

Table 3.4 – Discrimination drones/birds.

A small amount of bird data only was collected, we thus have to be cautious about this criterion.

Contrary to what we could expect, the neural network does not see this problem as a trivial one. The radar signals of birds have a great variability depending on the birds' behavior (flapping its wings, letting be carried by the wind, etc.). Thus, the bird data distribution is hard to be fully determined by a neural network. Further works should be done in this area, with a larger database.

For this criterion, the classification with the WSP format is significantly better than the other tested formats. It is closely followed by CP. The CVD has a good performance : only 3 points below the WSP format.

The neural network performance with $x(t)$ as input format is once again poor compared to other formats. We conjecture that the RCS is not strong enough to fully differentiate drones from birds. The results with SG as input are surprisingly low. As before, the classification accuracy is better with CVD than SG but the gap is large (8 points).

3.4.5 . Facility of Training

Format	Dimension	Time/Step [s]	Time [s]
$x(t)$	$1 \times 3\,000 \times 2$	0.39	79
WSP	$1 \times 2\,400 \times 2$	0.32	449
CP	$1 \times 3\,000 \times 2$	0.39	394
SG	$20 \times 300 \times 2$	0.18	425
CVD	$20 \times 120 \times 2$	0.10	163

Table 3.5 – Facility of training.

The time per step and the time to reach 95% of the maximum accuracy value are given in seconds.

The time per step for 1-D data is longer than for 2-D data of the same size because the 2×2 max-pool is replaced by a 2×1 one and, consequently, 1-D data is less reduced on deeper levels of the network. However, the CNN does not necessarily need more iterations to reach 95% of its maximum accuracy. The time to reach this maximum may be twenty times longer. Thus each training takes between several minutes and three hours.

Except for the fast training with CVD, the other formats have similar training speeds. WSP is the slowest format. For $x(t)$, only a few steps are needed to reach its maximum value, probably because this value is smaller. The time is therefore significantly shorter than the other formats.

3.4.6 . Human Visualization

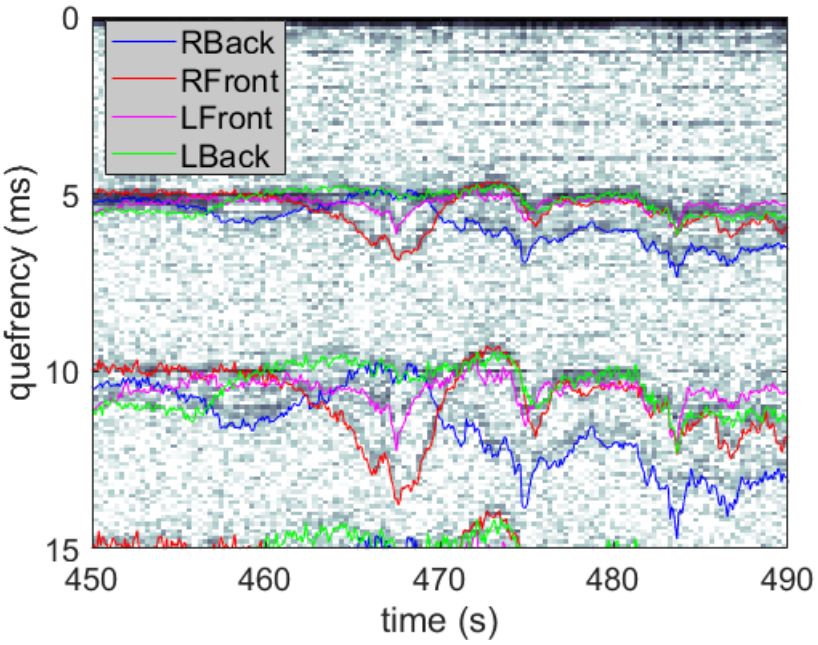


Figure 3.9 – Comparison between theorical data and real observation for a long-time cepstrogram. The foreground lines correspond to theoretical impact of rotors. The background is a concatenation of WSP to obtain a long-time spectrogram. The coordinate of this trajectory is present in Fig. 2.18. The CP data displayed in Fig. 3.2 corresponds to one column of this signal.

The subjective evaluation of a micro-Doppler signature is not our main goal. However, as it is frequently used in other studies, we do some comments taking Fig. 3.2 (simulated signal) and Fig. 3.1 (real signal) as examples.

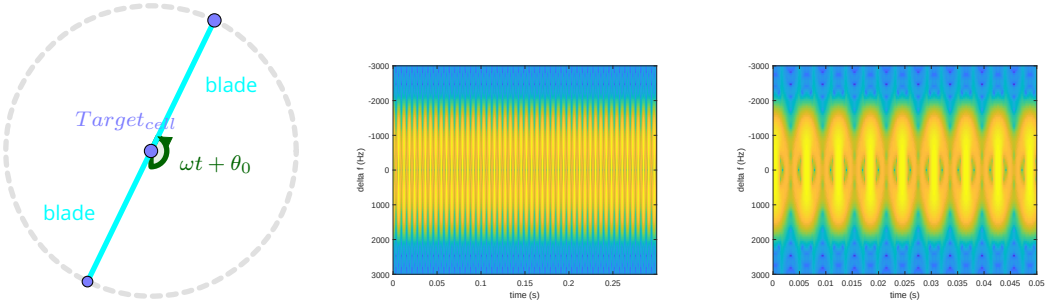
The human interpretation of $x(t)$ for multi-rotor drones is problematic. As pointed out in [17], small variations of the observation angle imply significant RCS variations. We observe similar results in terms of the main energy level in both simulated and real signals. In addition to the different periods where rotors hide each other, this format is strongly influenced by noise.

For WSP, the main Doppler effect, due to the target speed, is reduced by the long integration strategy. For relatively stationary drones, peaks produced by the rotors are observed. With long observation time, WSP profiles can be concatenated to obtain a long-time spectrogram allowing us to more complex study (Section 2.3.2). We highlight the fact that the rotor speeds are quite stable during 0.3 ms. Thus, the relatively stationary hypothesis is respected which makes WSP easy to interpret. It is not excluded that this stationarity might be linked to the good CNN performance of this format. The frequency resolution allows us to distinct rotors within 100 RPM. Each rotor produces lines at different orders (every f_m). For the first order (between $\pm$ 150 and 250 Hz), we distinguish three different rotors. We can use higher orders to distinguish rotors more precisely but we risk confusing high rotors speeds from the previous order or low ones from the following. We stress that the signal corresponding to one rotor spreads on several frequencies. Thus, the real frequency resolution is not as thin as 100 RPM.

The CP format is similar to WSP. It also allows one to measure the speed of the different visible rotors. For long observations, one concatenates the consecutive cepstrums to a long-time cepstrogram to observe the evolution of rotor speed. Fig. 3.9 represents the long-time spectrogram for the same trajectory as Fig. 2.16.

The SG format was designed to highlight blade flashes. This format is often used in the literature on simulated signals with a strong overlap and zero-padding. Thus, figures such as Fig. 3.10 are common in literature. Such figures are obtained with huge overlap (95%) and zero-padding ($\times 8$). It turns a $20 \times 300 \times 2$ image to a $160 \times 6000 \times 2$ one, Equation (3.6). Such a size creates a serious computing time issue, especially in real-time with an on-board system. Besides, in real signals, the format is difficult to interpret without adjusting specific parameters like the size of h or the overlap for each target. The main Doppler effect is significant (a large band around the zero frequency in Fig. 3.2), contrary to WSP. This band hides partly the blade flashes. A considerable part of the blade flashes is also hidden in the noise. Moreover, the different rotors turn at different speeds with different original phases. The coherent sum of their contributions is difficult to interpret even for simulated signals. Thus, this format is more adapted to targets with only one main internal movement like helicopters [8]. Moreover, contrary to helicopters, the drone blade lengths are similar to the wavelength λ of our signal, $\left(\lambda = \frac{c}{f_0} = 10\,\text{cm}\right)$. Radars with a higher frequency f_0 would produce this format with better quality. A higher PRF would give a better frequency resolution and could also have a strong impact, especially in the SNR after a FT.

The CVD format is more convenient for longer observations. This format does not lend itself easily to visual analysis. We still observe a strong signal at ± 200 Hz which corresponds to the blade (Fig. 3.2). Unfortunately, we also observe that the signal is very diverse for longer observation time (Figs. 2.16 and 3.9) so this format should also not be easy to interpret for longer observations with real signals.



Simulation
Rotor at 5 000 RPM

SG
Full picture

SG
Zoom on time axis

Figure 3.10 – Representation of SG format for a simulated signal of 1 rotor at 5 000 RPM without any cell. The representation is made with a strong overlap ($95\,\%$) and zero-padding ($\times 8$).

3.5 . Result Interpretation

Table 3.6 summarizes our results. This qualitative evaluation is based on a trade-off between absolute classification and the Δ column. For each case, we assess one of the grade —, -, 0, +, ++ (from very bad to very good, respectively).

A WSP input produces better classification results than all other formats for the reference case. Moreover, the classification performance decreases significantly less with a weak SNR than for other formats. WSP is also robust to shortening of the observation time, only CP being more resilient. We therefore recommend using the WSP format to classify drones according to micro-Doppler characteristics, particularly for data with a poor SNR.

The network results with the CP format are close to those obtained with WSP for the reference case while resisting considerably less than it to SNR degradation. CP is, however, more robust to shorter observations. When classifying data with a good SNR and a short observation, we recommend comparing the results obtained with CP and WSP.

The networks fed with CVD, SG or $x(t)$ inputs are always outperformed by the ones with WSP, we thus do not recommend them.

We remind that the training and testing sets should be collected on different days as the random data repartition might improve the network performance artificially. More generally speaking, the laboratory-made measurement campaign

Format	Use conditions					
	A	**B**	**C**	**D**	**E**	**F**
$x(t)$	—	—	—	—	++	—
WSP	**++**	**++**	**+**	++	**-**	**++**
CP	++	-	++	++	0	++
SG	+	—	0	-	0	-
CVD	+	-	—	+	+	-

Table 3.6 –
Qualitative analysis of the CNN performance
— very bad to ++ very good

A Reference

B Robustness to noise

C Short observation

D Discrimination drones/birds

E Facility of training

F Human visualization

suffers from biases resulting in training and testing datasets too close and thus results higher than results expected in the field. In reality, the diversity of possible data (number of different drones, birds, locations, meteorological conditions, etc.) is high. So the available data, the training dataset is really small compared to the real test dataset.

The different formats contain at best as much information as the $x(t)$ format while giving better performance. Input formats have thus an influence on the performance despite the extractions made by a CNN. It might be due to the small amount of data and the environmental bias inherent to radar measurements.

The results on the discrimination between drones and birds also emphasize the strength of the WSP and CP formats. However, due to the insufficient data quantity, we remain cautious about the interpretation of these results. As the drone/bird classification does not seem trivial, we recommend further studies with more bird data collected. We remind that the prevalence of birds is much higher than drones in a real environment so a good recognition system should work to obtain very high results to avoid false alarms. It is worth noticing that a recognition of drone classes is successful, even for drones equipped with the same number of rotors. So we might assume that this problem can be solved with further studies.

Our study concerns the drone recognition application only. It would be interesting to see whether our conclusions would help in other micro-Doppler applications such as human movement recognition.

We also observe that all formats we studied are not specifically designed for classification with a CNN. An interesting research line would be to produce a micro-Doppler signature format dedicated for CNN, outperforming all-purpose formats.

From the experiment developed, we chose the WSP format for the next chapters. Due to the difficulty of the collection of birds data, we put aside this case.

4 - Data Augmentation

We already know that with an appropriate format and neural networks, we obtain a correct classification accuracy in a big data context (36 730 chunks in the learning dataset, 4 670 in the testing one, Section 3.4). However, in the radar context, gathering data is expensive. Even though we collected a large dataset, in practice the amount of data to train the classifier never covers sufficiently all the configurations met on the field. Now, we put ourselves in the situation where training datasets are much smaller than testing datasets. Our goal is to tackle this issue with data augmentation by GAN algorithms [22].

It is impossible to collect data representing the whole diversity of the field configurations. So, as we cannot increase the testing datasets with newly collected data, we reduce the training dataset to have this large difference between the training and testing datasets. We keep the same testing dataset as in Chapter 3 and work with reduced learning datasets. In this scenario, the testing dataset represents configurations which were not measured on the field. It allows us to be in the required scenario and to make proper evaluations. The goal is to maintain a good performance despite the reduction of the training dataset size. To fulfill our goal, we re-augment the training dataset with data synthesized by GAN algorithms.

4.1 . State of the Art

We explain the concept of data augmentation in general and in the GAN context in particular. Next, we perform a comparison of the different GAN algorithms studied.

4.1.1 . Data Augmentation

The strength of neural networks such as a CNN is to find features to classify the dataset themselves. This method offers a high performance of classification but can also raise an over-fitting which happens when the features chosen by the CNN are too specific to the training dataset, preventing it from classifying correctly the testing one. Over-fitting may be reduced when training datasets are large enough. Consequently, the lack of representative enough datasets is one of the main limitations of Deep Learning algorithms.

To address this issue, the data augmentation methods bring a potential solution.

Principle

Data augmentation consists in creating data, which we call synthetic data, to enrich the original training dataset, composed of real data. The concept is based

upon the assumption that the synthetic data contains relevant information. Thanks to the information added, the CNN is better trained and thus achieves a higher classification rate on the testing dataset.

A possible manner to increase the volume of training data is its generation from simulations of physical models (Section 2.1) which have been a major branch of study in radar applications, particularly in the case of micro-Doppler profiles for drones and human movements [10, 21, 12, 7, 25]. As said earlier, however, the data produced by classic simulations is based on strong hypotheses [39] and may differ much from real data (Section 2.3.2).

In [10, 21], a comparison between real data and electromagnetic simulation is made with visual interpretations. In [12, 7], neural networks are used to classify micro-Doppler profiles. As their authors did not have access to real data, they used simulations only. In [25], a neural network is used to remove the environmental noise from the profiles. To test their network they use two datasets : a real one and a simulated one, without mixing them. The simulations are used only as a tool for preliminary results.

Another possibility is producing new data by applying certain transformations to real data. For image recognition, numerous transformations such as rotation, scale, mirror are commonly used [66, 14, 35, 20]. For example, the rotated picture of a chair remains a chair. So from one chair image, one can create several appropriate chair images by applying rotations on the original one. In this way, the classifier is informed that the data is rotation-free. This strategy requires a prior identification of the appropriate transformations. As such transformations are not identified for micro-Doppler profiles, we cannot use them.

In recent years, other generation algorithms based on neural networks have been developed such as GANs [22] and Variational Auto-Encoders [33]. In these techniques, the aforementioned transformation is created by the generative network and does not require any prior information.

Originally designed for the image generation, recent works suggest that the GAN algorithms could be used for data augmentation in various domains [20] among them micro-Doppler signals for both human movements [25, 16, 2, 1, 15] and drones [13].

Limitations of GAN applications

Many articles discuss the potential of GANs without evaluating the utility of the generated images. The interest of the latent space produced by GANs on human micro-Doppler signatures for future classification goals is introduced in [15]. The possibility of denoising human micro-Doppler signals thanks to GANs is reported in [25]. The capacity of GANs to extract the drone signature from a signal heavily polluted by a wind turbine is studied in [13].

Some research works deal with the classification improvement thanks to GANs via data augmentation. For instance, such an improvement is observed on liver lesion classification [20] and human micro-Doppler signatures [16, 1, 2]. However, these works were made with one GAN trained on one dataset with a single classification run. As observed in [2], heavy variations occur for the same GAN setup without even mentioning the inherent variations during the classification. Besides, the variety of evaluation methods used in these studies compromise the comparison of their results. According to [38], a more systematic and objective evaluation procedure is required to evaluate the GAN algorithms efficiently.

We carried out a data augmentation study with GAN algorithms. The next section treats the different GAN algorithms studied. Then, we propose an evaluation method resilient against the instability of GANs while being sufficiently general to be applied to any kind of data.

4.1.2 . GANs

GAN Algorithm

A GAN [22] captures a data distribution out of a set of instances and produces new data according to it. It is based on the confrontation of two neural networks. One of them is the generator G which produces synthetic data taking a random vector $z \sim P_z$ as an input. Its outcome should be realistic enough to dupe the other network, the discriminator D, which attempts to distinguish synthetic data from real data, $x \sim P_x$. The networks G and D learn simultaneously with adversarial goals. The GAN diagram is given in Fig. 4.1. The min-max strategy is described in Equation (4.1) and the associated loss functions in Equation (4.2) :

$$\mathsf{min}_G \mathsf{max}_D V(D,G) = E_{x\sim P_X}[\log(D(x))] + E_{z\sim \mathsf{noise}}[\log(1-D(G(z)))], \quad (4.1)$$

$$\begin{aligned} \mathsf{Loss}_G^{GAN} &= E_{z\sim P_z}[\log(D(G(z)))], \\ \mathsf{Loss}_D^{GAN} &= E_{x\sim P_X}[\log(D(x))] + E_{z\sim P_z}[\log(1-D(G(z)))]. \end{aligned} \quad (4.2)$$

The two networks are dependent. The discriminator is used to avoid a direct comparison between the real data and the synthetic one. After the learning phase, only the generator is used. The discriminator was implemented for the learning of the generator only.

Undesired GANs

Among the different undesired phenomena which may occur during a GAN training, the memory GAN effect is one of the most common. A memory GAN is

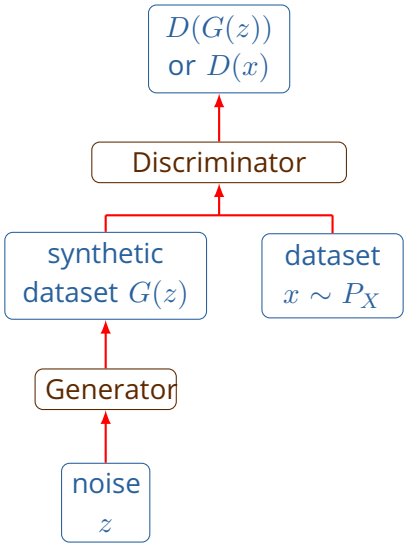


Figure 4.1 – GAN scheme

a trivial unwanted solution to the Equations (4.2). A GAN reaching such a state mimics the examples of the training dataset without producing anything new. This phenomenon is problematic : as it is an optimal solution to our equations, the minimization of the cost function might lead to it.

The strategy used to avoid this drawback is based upon several tricks. First of all, the generator does not have direct access to the real data. The information passes through the discriminator. The latter is conceived to catch only the important patterns on the real data and propagates only this information to the generator.

The network architecture generalizes as much as possible the example seen to avoid specialization. Several methods not detailed here are used such as pooling, batch normalization, drop-out. Besides, the generative quality can be assessed periodically to check a potential memory GAN effect. However, to evade this phenomenon, criteria based only upon the realism of the synthetic data are not relevant. The synthetic data is perfect in terms of realism but completely useless.

In addition to the problematic case of a memory GAN, other issues may occur. When GANs appeared in the literature, the process was very unstable with only a few runs over many leading toward an appropriate direction. For example, if the balance between the discriminator and generator capacities is compromised a mode dropping effect may happen.

A mode dropping effect consists in a generator that produces only synthetic data of a part of the distribution. This takes place when a discriminator is too weak for this part. For example, for a digit recognition, if the discriminator is too bad to recognize the real digits 1, the generator may decide that the production of 1s

exclusively is a better strategy to dupe the discriminator. The latter may change its strategy and consider that refusing all 1 and accepting everything else without any deeper analysis would be an interesting solution. Then, the generator may reply by producing only the digit 4 and so on.

DCGAN

In [48], different GAN setups are evaluated to obtain stable GANs with specific choices of some parameters and architecture. These specific setup choices stabilized considerably GANs to obtain good-enough DCGAN (*Deep Convolutional GAN*). In the literature, a DCGAN is a GAN respecting these specific choices.

We summarize here the main DCGAN guidelines [48] :

— Use ADAM optimizer [34] set with a learning rate of $2e^{-4}$ and a β_1 of 0.5 instead of respectively $1e^{-3}$ and 0.9 for usual classification problems.

— Replace each pooling layer by a stride.

— Use the batch normalization.

— Remove fully connected hidden layers for deeper architectures.

— Use the ReLU (*Rectified Linear Unit*) activation in the generator for all layers except for the output, which uses bounded function such as $\tanh$.

— Use the leaky ReLU (lReLU) activation in the discriminator for all layers.

We point out that from our experiments we conclude that the batch normalization and Adam optimizer setup are the most important criteria. Unless stated otherwise, all the networks used in this study and the studies from the literature respect at least those two criteria. For example, we kept in some cases one fully connected hidden layer at the end for drop-out despite the computational cost.

4.1.3 . GAN Extension with Prior Knowledge

As the classic GAN algorithm does not benefit from any prior knowledge, it cannot use specific information about the distribution. We lay out in this section certain GAN architectures allowing a prior-knowledge. As a prior-knowledge is widely used for many different purposes we explain this term. For simplification, we consider two kinds of prior-knowledge.

The first one, the general prior-knowledge, is information on the whole distributions. For example, for the MNIST dataset, a prior knowledge would be that there are ten digits. We still consider that an algorithm which uses such general information as unsupervised.

The second one, the specific prior-knowledge, is information on specific examples of the dataset. It is obviously more powerful than general prior-knowledge. The specific prior-knowledge may be expressed by labels or annotations. A label is the class of an object. For example, the image number

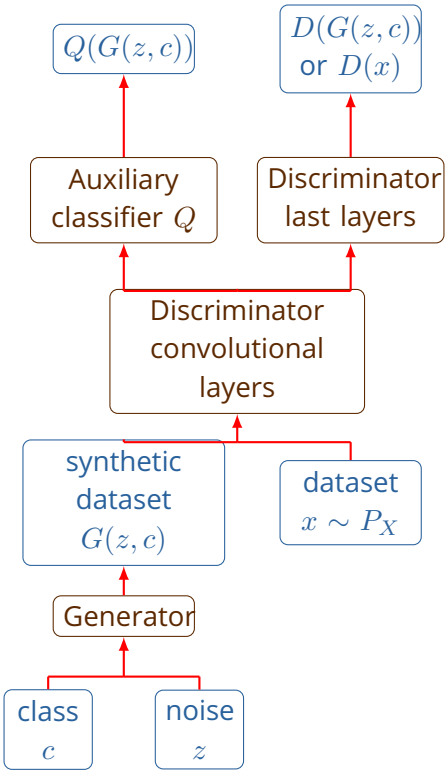


Figure 4.2 – InfoGAN scheme. For real data x, Q is inactive.

$35\,476$ of the MNIST training dataset is labeled by six. An annotation is a complementary information on a specific data which often takes continuous values (the thickness of the digit, etc.). We consider algorithms using labels or annotation as supervised.

InfoGAN Algorithm

InfoGAN [11] is an unsupervised algorithm based upon the general prior knowledge about the distribution. For the MNIST dataset, InfoGAN knows that the data are divided into ten classes but does not see any label.

The generator must produce data respecting an annotation or a label c despite not having any example. There is no guaranty that the labels given by the network are similar to the ones made by a human or, at least, understandable by humans.

The architecture of InfoGAN is schematized in Fig. 4.2. To use labels c, the cost function is modified accordingly by adding the similarity between c and the generated image represented as the information : $I(c, G(z,c))$. The maximization of the information $I(c, G(z,c))$ is indirectly obtained [3] via the maximization of the divergence of Kullback-Leibler, the right-hand term in the following equation :

$$I(c, G(z,c)) \geq E_{g\sim G(z,c)}\left(E_{c'\sim P(c|g)}\left(\log\left(Q\left(c'|g\right)\right)\right)\right) + H(c) = L_I(G,Q). \quad (4.3)$$

The loss function modifications are given below, parameter λ is a weighting factor :

$$\begin{aligned} \text{Loss}_{D,Q}^{\text{InfoGAN}} &= \text{Loss}_D^{\text{GAN}} - \lambda L_I(c,c'), \\ \text{Loss}_G^{\text{InfoGAN}} &= \text{Loss}_G^{\text{GAN}} - \lambda L_I(c,c'). \end{aligned} \quad (4.4)$$

The informative and discriminative parts share the convolutional part which is seen as a feature extractor, Fig. 4.2. It is made to reduce the computation time. Moreover, according to [11], the learning is faster than for a classic DCGAN. Thus, the informative part is "for free".

The prior-knowledge given to the InfoGAN is the choice of the dimension of c and its distribution law. For example for MNIST, we can choose c to be a vector of ten components with one equal to 1 and nine equal to 0. By doing so, we inform the network that the dataset is composed of ten disjoint classes. We then give a probability for each component to be the chosen one (value 1). For example, if it is on tenth for each, we inform the network that the dataset is uniformly distributed.

The InfoGAN algorithm can be summarized with the following concepts :

— The generator is being told that a part c of the random input vector (z, c) has physical meaningful information. The generator tries to give signification to this part : to maximize the mutual information between c and the synthetic data produce $G(z, c)$, Equation (4.3), although it is just a random vector.

— The choice of the dimension and probability law of c (general prior-knowledge) orientate the network to give the desired signification to c.

— At the end of learning, the verification is made whether the mutual information maximization worked, i.e. if c has an impact on the generation. If possible, c might be compared to known annotations/labels.

We illustrate the behavior of an InfoGAN with two examples from [11]. To begin, the InfoGAN captures the digit information about the rotation and width of the number (Fig. 4.3) without any label related to it, which is quite impressive. Moreover, it generalizes these concepts.

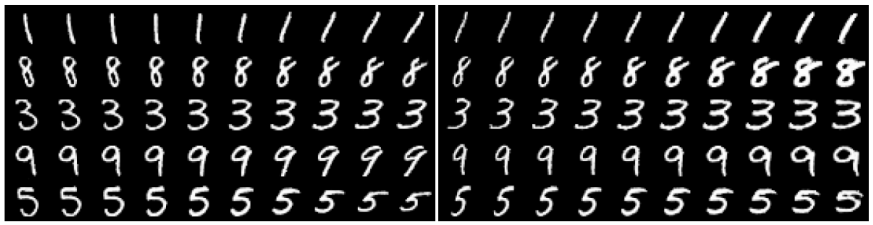
In this example, the network was trained with c_2 and c_3 (respectively for rotation and width) between -1 and 1 with an uniform probability. For the element of the training dataset, the generator creates the thinnest digits for $c_3 = -1$ and the thickest digits for $c_3 = 1$ relatively to the known examples (the training dataset). At the end of learning the authors give to c_2 and c_3 greater values : between -2 and 2. It compels the network to rotate/thick the digit more than it has ever seen during the learning. The authors observed that the network

followed their requests and managed the generalization capacity. The second example (Fig. 4.4) gives similar results on a more challenging dataset : 3-D chairs.

One of the crucial points is that the majority of the evaluations in [11] is human interpretations. The authors consider that the generator chose the rotation and the width for c_2 and c_3 but can only show some examples to prove this assertion. However, as c_1 was a vector of ten components to capture the label, an evaluation on the test dataset as a classifier was conducted. At the end of training, a human associates each component of c_1 to the probable digit captured. Indeed, there is absolutely no reason that the first component captured the digit 0 and so on. The evaluation achieves an error rate of 5% on the MNIST dataset.

According to [11], the λ coefficient may be essential to capture the correct annotations. Optimized λ values for each continuous latent might be needed while setting this coefficient to one is sufficient for a categorical latent.

The cost function related to a continuous latent is a classic norm such as the Euclidean norm while the cost function associated to a categorical latent is the cross-entropy.



(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation) (d) Varying c_3 from -2 to 2 on InfoGAN (Width)

Figure 4.3 – Example of InfoGAN test on MNIST, taken from [11]

(a) Rotation (b) Width

Figure 4.4 – Example of InfoGAN test on 3-D chairs dataset, taken from [11].

ACGAN Algorithm

An ACGAN [43] offers the possibility to control the class c of the synthetic output thanks to its auxiliary classifier Q. In addition to their adversarial goals

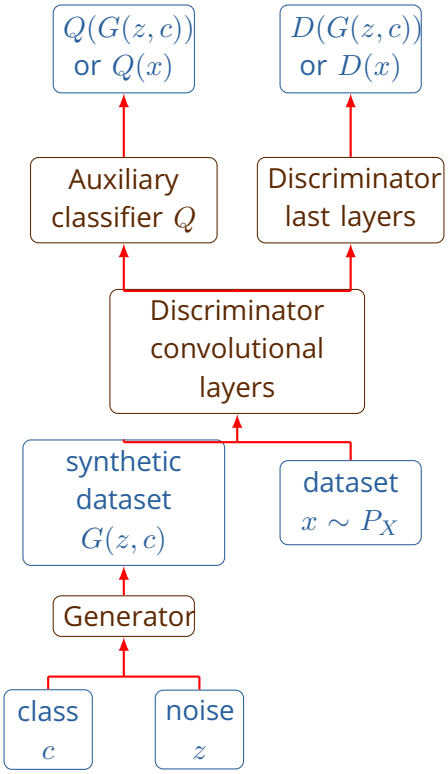


Figure 4.5 – ACGAN scheme, the label of x is known.

to create realistic samples, G and D help each other to produce synthetic data of the appropriate class by retro-propagating the error on Q. The latter uses the convolutional layers of the discriminator. The ACGAN diagram is given in Fig. 4.5 and the equations are below :

$$\begin{aligned} \text{Loss}_G^{\text{ACGAN}} &= \text{Loss}_G^{\text{GAN}} + E_{g\sim G(z,c)}[E_{c'\sim P(c|g)}[\log(Q(c'|g))]], \\ \text{Loss}_{D,Q}^{\text{ACGAN}} &= \text{Loss}_D^{\text{GAN}} + E_{x\sim P_X}[E_{c'\sim P(c|x)}[\log(Q(c'|x))]] \\ &\quad + E_{g\sim G(z,c)}[E_{c'\sim P(c|g)}[\log(Q(c'|g))]]. \end{aligned} \tag{4.5}$$

ss-InfoGAN Algorithm

ss-InfoGAN [58] is an extension of InfoGAN, based upon the combination of unsupervised labels/annotations and supervised ones. The c vector (Figs. 4.2 and 4.5) is split into two parts, an unsupervised one c_{us} and a semi-supervised one c_{ss}. The c_{us} is treated as for InfoGAN with the same cost function. For c_{ss}, some labels/annotations of real data are available. They are used to direct the c components associated to the desired labels/annotations, which leads to two loss terms $L_{I_{\text{ss}}}(G,Q)$, $L_{I_{\text{ss}}}(G,Q)$ for real data and $L^2_{I_{\text{ss}}}(G,Q)$ for generated data corresponding to $L_{I_{\text{ss}}}(G,Q)$ of InfoGAN :

$$I(c_{\text{ss}}, X) >= E_{x\sim P_X}\left(E_{c'_{\text{ss}}\sim P(c_{\text{ss}}|x)}\left(\log Q\left(c'_{\text{ss}}|x\right)\right)\right) + H(c_{\text{ss}}) = L^1_{I_{\text{ss}}}(Q),$$
$$I(c_{\text{ss}}, G(z, c_{\text{ss}})) >= E_{x\sim G(z,c_{\text{ss}})}\left(E_{c'_{\text{ss}}\sim P(c_{\text{ss}}|x)}\left(\log Q\left(c'_{\text{ss}}|x\right)\right)\right) + H(c_{\text{ss}}) = L^2_{I_{\text{ss}}}(G, Q). \tag{4.6}$$

The cost functions is defined as :

$$\begin{aligned}\text{Loss}^{\text{ss-InfoGAN}}_{D,Q} &= \text{Loss}^{\text{GAN}}_{D} - \lambda_1 L_{I_{\text{us}}}(c_{\text{us}}, c'_{\text{us}}) - \lambda_2 L^1_{I_{\text{ss}}}(c_{\text{ss}}, c'_{\text{ss}}),\\ \text{Loss}^{\text{ss-InfoGAN}}_{G} &= \text{Loss}^{\text{GAN}}_{D} - \lambda_1 L_{I_{\text{us}}}(c_{\text{us}}, c'_{\text{us}}) - \lambda_2 L^2_{I_{\text{ss}}}(c_{\text{ss}}, c'_{\text{ss}}).\end{aligned} \tag{4.7}$$

Contrary to what we could expect, $L^2_{I_{ss}}(G, Q)$ is not used for the learning of Q but only for G. It is probably due to avoid the real labels to be drowned in the labels corresponding to synthetic data. Indeed, as we are in a semi-supervised context, only a small quantity of labels of real data are known. So, to be sure that this information is well learned by Q, only $L^1_{I_{ss}}(G, Q)$ appears in the loss function.

fs-InfoGAN Algorithm

The ss-InfoGAN algorithm is conceived for partly labeled datasets. The part of labeled data in the datasets from [58] varies from 0.8% to 10%. Nevertheless, it can still be used for fully labeled datasets. We denote full-supervised InfoGAN (*fs-InfoGAN*), ss-InfoGAN used in this context :

$$\begin{aligned}\text{Loss}^{\text{ACGAN}}_{D,Q} &= \text{Loss}^{\text{ss-InfoGAN}}_{D,Q} + L^2_{I_{SS}}\\ \text{Loss}^{\text{ACGAN}}_{G} &= \text{Loss}^{\text{ss-InfoGAN}}_{G}\end{aligned} \tag{4.8}$$

ACGAN and fs-InfoGAN are similar. From Equations (4.5) and (4.7), we conclude that only one term distinguishes an ACGAN from an fs-InfoGAN without any annotations added.

The ACGAN was originally proposed for labels only, not for annotations. In our opinion, annotations could enrich an ACGAN with only the supplementary λ coefficients. We specify that we choose the notation fs-InfoGAN to avoid any confusion with ACGAN.

4.1.4 . GANs Comparison

The discussed GAN algorithms could be executed by any type of neural network architecture (convolutional, etc.). The difference between them arises from the loss function. To simplify the formulæ and highlight the differences, we denote :

G : term of the Generator,

D : term of the Discriminator (classification included for prior-knowledged GANs),

To keep short, when the term is identical for both networks, the letter G/D is not written,

d : discrimination term (real or synthetic image),

c : classification term (the real/synthetic image is correctly classified),

f : synthetic data term,

r : real data term.

For a fully-supervised case without any continuous information, we define the following terms :

— D_d_f : $E_{z\sim\text{noise}}[\log(1-D(G(z)))]$: cost of the discriminator to properly discriminate synthetic data,

— G_d_f : $\text{Loss}_G^{GAN} = E_{z\sim\text{noise}}[\log(D(G(z)))]$: cost of the generator to trick the discriminator by producing realistic-enough synthetic data,

— D_d_r : $E_{z\sim\text{noise}}[\log(D(G(z)))]$: cost of the discriminator to discriminate real data properly,

— c_r : $L^1_{I_{SS}}$: cost of the auxiliary classifier to classify real data precisely,

— c_f : $L^2_{I_{SS}}$: cost of the auxiliary classifier to classify synthetic data precisely.

Thanks to this notation, the loss functions corresponding to each GAN described earlier become :

$$\begin{aligned}
\text{Loss}_G^{\text{GAN}} &= G_d_f, \\
\text{Loss}_D^{\text{GAN}} &= D_d_f + D_d_r; \\
\\
\text{Loss}_G^{\text{InfoGAN}} &= G_d_f + c_f, \\
\text{Loss}_{D,Q}^{\text{InfoGAN}} &= D_d_f + D_d_r + c_f; \\
\\
\text{Loss}_G^{\text{ss-InfoGAN}} &= G_d_f + c_f, \\
\text{Loss}_{D,Q}^{\text{ss-InfoGAN}} &= D_d_f + D_d_r + c_r; \\
\\
\text{Loss}_G^{\text{ACGAN}} &= G_d_f + c_f, \\
\text{Loss}_{D,Q}^{\text{ACGAN}} &= D_d_f + D_d_r + c_r + c_f.
\end{aligned} \tag{4.9}$$

4.1.5 . GAN Algorithm Choice

As data should be augmented for each class, there are two directions we could take. First, we make different GANs for each class and gather the resulting data. Second, we implement one GAN algorithm which manages labels to generate the different classes. The two following arguments made us choose the second direction :

— a single generator producing all data simplifies the procedure : only one GAN to setup, one GAN to train and one GAN to evaluate,

— an auxiliary classifier Q should prevent from trespassing between class domains, leading the generator towards an appropriate direction. In the case of a wrong setup GAN containing an auxiliary classifier, the loss terms c_f and c_r allow us to detect this trespassing. The technique designed for this early detection will be given in Section 4.4.4.

4.2 . Evaluation Method of the Data Augmentation

The assessment of the synthetic data quality by the human interpretation is the most frequently reported in the existing literature. As we wish to measure the utility of the synthetic signals for automatic classification, we developed a rigorous measurement. This measurement gives us an objective evaluation, avoiding the subjective visual interpretation of micro-Doppler signals.

First, we introduce the evaluation of the original dataset d, composed of real data only. This quality is expressed in terms of its capacity to train a CNN. Second, we propose the evaluation of the generator G of an ACGAN by its efficiency to produce samples $G(d)$ increasing this training capacity. Eventually, we describe the procedure of finding an appropriate ACGAN setup A. As this method is general, A can be replaced by any generative algorithm. Combining these three axes of assessment, we evaluate the data augmentation method with ACGAN for a given real dataset.

4.2.1 . Dataset Evaluation

The quality of a real dataset d is evaluated by its efficiency to train a CNN (denoted classifier network C) to classify a test dataset t. In each evaluation in the remainder, the same C (GoogLeNet, Section 3.1.2) and the same t are used.

We run N training of C and compute their mean result. The classification results in the tables correspond to the maximum over iterations of the mean accuracy and its confidence interval at 95%.

We denote $c_n[d](i)$ the accuracy rate achieved by C at iteration i, $i \in [\![1 : I]\!]$ of training $n \in [\![1 : N]\!]$ with d as training dataset. The procedure is illustrated in Fig. 4.6. The formula below defines the quality of a dataset d :

$$\text{eval}(d) = \max_{i \in [\![1:I]\!]} \left\{ \frac{1}{N} \sum_{n=1}^{N} c_n[d](i) \right\}. \tag{4.10}$$

4.2.2 . Generator Evaluation

As our goal is to improve the classification performance by data augmentation, we do not examine the realism of synthetic data. To evaluate an ACGAN, we evaluate the combined dataset of the real dataset d and the ACGAN-generated

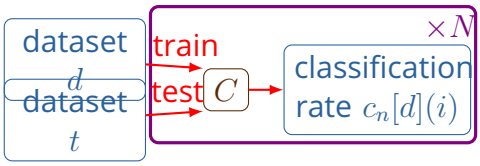


Figure 4.6 – Evaluation of a dataset d, the test dataset t is fixed.

one $G(d, c)$ as done in [20, 1, 2]. Contrary to [16], which classifies only with $G(d)$, we are not interested in the quality of the GANs to reproduce the entire distribution.

We take as a criterion the average classification rate. We start by creating a large synthetic dataset $G(d)$. Next, for each run n of C, synthetic samples $G^n(d)$ are uniformly selected to create the combined dataset $d + G^n(d)$. Unless specified otherwise, d and $G^n(d)$ have the same size. As detailed later (Section 4.4.7), we did not observe any gain with larger proportions of synthetic data during the calibration. Fig. 4.7 schematizes the protocol used.

The evaluation of a generator G for d is obtained from :

$$\text{eval}_d(G) = \max_{i \in [\![1:I]\!]} \left\{ \frac{1}{N} \sum_{n=1}^{N} c_n \left[d + G^n(d) \right](i) \right\}. \tag{4.11}$$

This evaluation tool helps us to achieve our main goal, creating useful generators.

Definition 4.1. *A generator G is **useful** if $\text{eval}_d(G)$ given by Equation (4.11) is significantly greater than $\text{eval}(d)$ from Equation (4.10). Put differently, G is useful if it produces data improving the classification performance.*

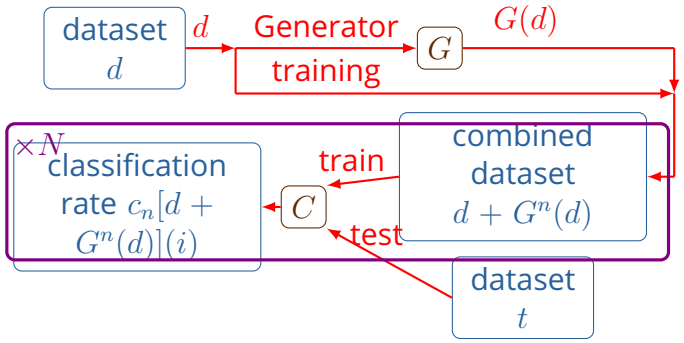


Figure 4.7 – Evaluation of a generator G for datasets d and t fixed.

4.2.3 . Generation Algorithm Evaluation

In order to evaluate our generation algorithm A, we determine the quality of the generation on a set of K generators $(G_1 ... G_K)$. The protocol used is schematized

in Fig. 4.8. The criterion we chose for A, denoted $\text{eval}_d(A)$, is the gran mean (the mean of the means) :

$$\text{eval}_d(A) = \frac{1}{K}\sum_{k=1}^{K} \text{eval}_d(G_k). \tag{4.12}$$

Our goal is to obtain a data augmentation method that creates useful generators with a reasonable probability. We assume to have enough resources to generate several generators and pick the best one. To evaluate the data augmentation method, the two following criteria are chosen.

The first one is the gran mean of the top quartile q_1 classification rates denoted $\text{eval}_d^{q_1}(A)$:

$$\text{eval}_d^{q_1}(A) = \underset{G \in q_1}{\text{mean}} \left\{\text{eval}_d(G)\right\}. \tag{4.13}$$

The second one, denoted $\%_d^{>}(A)$, is the percentage of the useful generators obtained. For each gran mean, the associated standard error at 95% obtained from the exact pooled variance [53] is also computed.

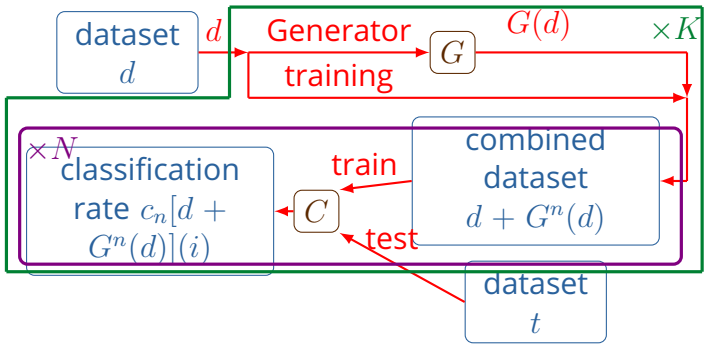


Figure 4.8 – Evaluation of the data augmentation for d and t fixed.

4.3 . Experimental Protocol

To start our experiments, we choose one reduced dataset adapted to the observation of the data augmentation on the drone recognition problem. Then, we describe the architecture of the chosen GAN. We continue in the next section with the first results obtained and the calibration made to reach the best performance, Section 4.5.

4.3.1 . Reduction of Training Dataset

We start by reducing substantially the size of the training dataset. We randomly select a small dataset d_0 of size $|d_0| = 25$ (5 elements per class) with an evaluation of $\text{eval}(d_0) = \mathbf{58.2 \pm 0.29}\%$.

4.3.2 . GAN Architecture Chosen

Our considerations led us to choose the ACGAN and fs-InfoGAN algorithms which differ only in a single term in their loss function. The cost of the classification of synthetic images by the auxiliary classifier c_f, introduced into the ACGAN loss function, Equation (4.9) makes its learning process more stable than the one of the fs-InfoGAN. However, c_f might confuse Q and lead it to consider non-existent patterns. Indeed, if at the beginning of learning, the synthetic example of a class contains a non-realistic specific pattern, Q might learn it and consider it as an aid to distinguish classes despite the absence of this pattern in the real dataset. The argument in favor of our ultimate choice, ACGAN, will be explained in detail in Section 4.4.7.

We detail below the architecture of our GANs. The concepts used are introduced in Section 3.1.2. To have a more in-depth description, we advise the online book [42].

The generator is a network composed of two fully connected layers (FC) followed by six deconvolution layers (deconv). The chosen discriminator is a mirrored network with six convolution layers (conv) followed by two dense layers. We did several tests with deeper and shallower networks but it seems that other meta-parameters such as the DCGAN conditions have a stronger impact than the network architecture.

The first full-connected layers of the generator are used to gather properly the noise input z and the informative vector c as advised in [11]. We assume that keeping the same depth for the generator and the discriminator provides a better balance between the two networks. The vector c is a categorical latent corresponding to the five drone classes.

We follow the DCGAN rules and take as activation functions ReLU functions for the generator and lReLU functions for the discriminator. For the last layers, a bounded activation function is suggested to accelerate the learning. In [48] the $\tanh$ is suggested. We chose the widely used sigmoid expecting similar results with $\tanh$.

As the number of epochs in GAN training is often large, we used label shifting on real data to avoid discriminator over-fitting. Instead of real labels equal to 1, we shifted the value to 0.9. In preliminary tests, we observed a considerable impact on the GAN stability thanks to this parameter.

The cost function implemented is the cross-entropy for both the auxiliary classifier Q and the discrimination part. We did not observe any statistically significant gain thanks to the loss weighting of the loss terms, Equation (4.9). This absence of gain is in accordance with the results from [11].

After preliminary experiments, we observed that a short batch and a great iteration number were giving satisfactory results. We took a batch of 4 for small training datasets ($|d| = 25$) and make it evolve with the dataset size. We

layer	D and Q	G
input	$1 \times 2\,400 \times 2$ Radar Image	69, $\lvert z\rvert = 64$, $\lvert c\rvert = 5$
layer$_1$	1×4 conv 32 lReLU, stride 2	FC 1 024 units ReLU, batchnorm
layer$_2$	1×4 conv 32 lReLU, stride 2	FC 19 200 units ReLU, batchnorm
layer$_3$	1×4 conv 64 lReLU, stride 2	1×4 deconv 128 ReLU, stride 2, batchnorm
layer$_4$	1×4 conv 64 lReLU, stride 2	1×4 deconv 64 ReLU, stride 2, batchnorm
layer$_5$	1×4 conv 128 lReLU, stride 2	1×4 deconv 64 ReLU, stride 2, batchnorm
layer$_6$	1×4 conv 128 lReLU, stride 2	1×4 deconv 32 ReLU, stride 2, batchnorm
layer$_7$	FC 1 024 units lReLU for D FC 128 units lReLU for Q	1×4 deconv 32 ReLU, stride 2, batchnorm
layer$_8$	FC 1 units sigmoid for D FC 5 units sigmoid for Q	1×4 deconv 2 sigmoid, stride 2

Table 4.1 – Topology of the neural networks.

observed that 10 000 iterations were sufficient. As explained later, we chose a different stopping criterion than an arbitrary iteration number.

Table 4.1, specifies the network topology. We remind that our data is uni-dimensional. To have a similar notation as the image recognition community, we denote filters in 2-D with one dimension set to 1. The data is also composed of two channels corresponding to real and imaginary components of the complex data.

4.4 . Main Setup Calibration

We present the procedure used to calibrate our GAN algorithm in order to create useful generators.

4.4.1 . Preliminary Results

When we started our data augmentation experiments, we observed a poor performance. We struggled to generate credible data. Augmenting the dataset d_0 with an evaluation of $58.2 \pm 0.29\,\%$, resulted in an evaluation four points lower for the combined dataset. Besides, the performance tended to decrease with a larger proportion of synthetic data. Table 4.2 illustrates these preliminary experiments with a generator G_{first}.

eval(d_0)	eval$_{d_0}(G_{\text{first}})$ synthetic proportion		
	0.5	0.67	0.8
58.2 ± 0.29	54.5 ± 0.71	53.2 ± 0.70	51.7 ± 0.64

Table 4.2 – Preliminary results example ($K = 1$, $N = 100$) obtained before calibration.

4.4.2 . Necessity of Calibration

The calibration of a neural network is always a tricky task. This difficulty comes from several characteristics. First of all, the black-box scheme of neural networks makes them hard to understand. The lack of any standard measurement to compare properly the different algorithms increases the difficulty to obtain efficient solutions.

Furthermore, among neural networks, GANs are probably the hardest to be understood as they suffer from strong instability issues [2], due to the unstable equilibrium between the generator and the discriminator.

The GAN setup is composed of numerous meta-parameters which need to be determined for each problem. However, the calibration has to remain general enough to avoid an over-fitting on a specific dataset. Unfortunately, it is impossible to verify all combinations with our evaluation metrics, $\text{eval}_d^{q_1}(A)$, because the computing time for each simulation is long and the random influence forces us to compute mean results over a large number of trainings.

4.4.3 . Protocol

To tackle the calibration issue, we used a model based on indicators, hypotheses, and their associated solutions. This model can be used for any GAN calibration. The hypotheses give possible explanations for the behavior of our GAN. Our indicators are values that can be found with a moderate computational effort. They inform us more deeply about the GAN state. We specify the expected values of these indicators in an operational condition. Thanks to these indicators, the plausibility of the different hypotheses can be assessed, once a hypothesis is validated, we suggest an adequate solution.

To avoid over-fitting, the calibration is made with the randomly selected dataset d_0. Results with other training bases are given in the next section. These results were performed with the same calibration.

4.4.4 . Indicators

The literature lists numerous indicators [44], let aside self-made ones. We only describe the indicators we studied. We start with a self-made one : the trivial generators, and continue with more classic indicators : the observation of error terms, the FID (*Frechet Inception Distance* [24]), and the tournament win rate.

Impact of Trivial Generators

To interpret the score of a generator, we compare it, not only with the evaluation of the original dataset alone but also with the evaluation of trivial generators. We identified several trivial generators we denoted : wrong generator, noise generator, and perfect generator.

Definition 4.2. *A **wrong generator** is a generator creating perfect data in terms of realism but with an incorrect label. It is considered the worst generator because*

it traps the most efficiently the classifier. The wrong generator is created with real data and a random label among all possible labels.

Definition 4.3. *A **perfect generator** is a generator creating perfect data both in terms of realism and label. It is considered the best generator. It gives an idea of the maximum possible gain reachable with data augmentation.*

Definition 4.4. *A **noise generator** is a generator creating random outputs without any link to the data distribution. For small quantity of synthetic data, this generator has no impact on the classification because the classifier does not take it into account. If a large part of the combined dataset is its product, the noisy data dominates the real data and the classifier performance degrades. One interesting aspect is the amount of synthetic data needed to observe this degradation.*

In an operational condition, the generation algorithm evaluations are below the perfect generator and above the noise one.

Loss Terms

We observe each error term individually, Equation (4.9) D_d_f, G_d_f, D_d_r, c_r, c_f. Those terms inform us about the realism of the synthetic data (generator-discriminator balance) and of the capacity to create synthetic data with a correct label. Details of their interpretation are given in the next section. In an operational condition, these different loss terms are unstable but have similar amplitudes during training.

Certain hyperparameters such as label shifting or weighting have an impact upon the value of these loss terms despite identical network output.

FID

FID [24] (*Frechet Inception Distance*) is a distance between two distributions, in our case, between those of the real and synthetic datasets. It comes from the Frechet distance of the second to last layer of inception-v3 network [55] :

$$\text{Frechet}(A, B) = \|\mu_A - \mu_B\|_2^2 + \text{Tr}\left(\Sigma_A + \Sigma_B - 2\left(\Sigma_A \Sigma_B\right)^{\frac{1}{2}}\right). \tag{4.14}$$

FID gives an estimation of the realism of the generated images but no information about correct labeling. In an operational condition, the FID value should decrease during training. Since its introduction in 2017 [24], FID has been used in a large number of studies. We highlight that this criterion is often the only one used to evaluate generative methods.

In [24], the consistency of the FID measurement is shown for various perturbations (Gaussian noise, Gaussian blur, implanted black rectangles, swirled

images, salt and pepper noise, and CelebA dataset contaminated by ImageNet images). Namely, this score is shown as consistent with the human judgment on several examples.

In [38], FID is the measurement used to compare the realism of the generated images by different GANs algorithms. The authors concluded from experiments that FID detected some undesired phenomena such as the mode dropping but remains, as the majority of measures, inefficient to detect memory GANs.

As inception-v3 has been designed for RGB images, we need three-channel data to compute FID. To feed those three channels, we combine the real and imaginary components together with the modulus of the signal. We repeat each 1-D column to create a 2-D image for inception-v3. We note that even if FID is faster to compute than our evaluation method (Section 4.2.3), it may still take a non-negligible time compared to a GAN training.

Tournament Win Rate

TWR [44] (*Tournament Win Rate*) is a measure of the balance between a generator G (or a set of generators) and a discriminator D (or a set of discriminators) thanks to a set of examples. To compute it, we use the discriminator outputs of a synthetic dataset and of a real one.

Each data resulting in a discriminator output above 0.5 is considered as real and that below 0.5 is considered as synthetic. When the discriminator is right, we give it a point. We then compute the mean score of the discriminator to obtain the tournament win rate for the discriminator (the complementary value corresponds to the tournament win rate for the generator). Ideally, in an operational condition, TWR should have a value of 0.5, indicating a good balance.

We did not use this indicator in our work because it has several drawbacks. First, in our experiments, TWR trends towards the extreme values 0 and 1, skipping intermediate values which might be explained by the size of the dataset. The threshold of 0.5 seems arbitrary and it deprives us information contained in a $D(x)$ value. For those reasons, we opted to use directly the mean loss values instead of the TWR ones.

4.4.5 . Hypotheses

We enumerate hypotheses resulting from GAN calibration problems. Unfortunately, those hypotheses are not mutually exclusive, some of them may occur simultaneously making the diagnostic harder.

Non-Representative Data Hypothesis

According to this hypothesis, the generated data is not realistic and different from each other. Under this assumption, a small amount of generated data should have no impact on the evaluation. Indeed, the classifier should just choose two

different methods to classify the images. It should find random patterns for the synthetic data and the proper pattern for the real data.

If this hypothesis is verified, the generator will have similar performances as the noise generator.

There is no evident solution for this case.

Non-Realistic Common Pattern Hypothesis

Under this hypothesis, the generated data contains a pattern that does not exist in real data like high values at the border of the data. This pattern is common for all classes and should not be used by the classifier.

If this hypothesis is verified, the generator will have a similar performance as the noise generator. Moreover, the discriminator should detect the synthetic data easily, $G\_d\_f \gg D\_d\_f$.

There is no evident solution for this case either, except finding the not realistic pattern to prevent it.

Non-Realistic Class-specific Pattern Hypothesis

If this hypothesis holds, the data is not realistic because it contains a class-specific pattern. For some GAN algorithms such as an ACGAN, this problem may occur when the two goals tear apart. To be able to classify properly the generated data, the generator produces a clear difference between the generated data of each class. Unfortunately, this difference does not exist in the reality.

If this hypothesis is verified, the generator should have a worse evaluation than the real dataset alone. False significant patterns are added, tricking the classifier. We should observe a performance worse than the noise generator. Moreover, the discriminator should easily detect the synthetic data, $G\_d\_f \gg D\_d\_f$. In addition to that, the classification error of the synthetic data c_f should be really low.

In this case, we need to reinforce the generator. We can assign weight to the different loss terms to increase the influence of G_d_f and reduce the influence of c_f. The learning rates or the architecture of the networks can also be modified to strengthen the generator.

Class Confusion Pattern Hypothesis

Now, realistic data is generated but it is wrongly classified. It is the opposite problem to the previous hypothesis. The evaluation of the generator should be much lower than the one of the dataset alone.

The significant patterns become ambiguous due to their presence in different classes. We should observe a similar performance as the wrong generator. The

Generator	synthetic proportion			
	0.5	0.67	0.8	0.89
perfect	61.9 ± 0.98	64.0 ± 1.02	70.5 ± 0.67	75.8 ± 0.51
noise	59.0 ± 1.11	57.3 ± 1.04	43.1 ± 2.36	25.4 ± 1.20
wrong	37.0 ± 1.52	31.1 ± 1.17	26.5 ± 0.90	24.5 ± 0.97

Table 4.3 – Performance for trivial generators for d_0 dataset (eval$(d_0) = 58.2 \pm 0.29$).

generator should often dupe the discriminator in terms of realism, $G\_d\_f \ll D\_d\_f$, but the classification error of the synthetic data c_f should be high.

In this case, we need to reinforce the discriminator. We can assign weight to the different loss terms to decrease the influence of G_d_f and augment the influence of c_f. The learning rates or the architecture of the networks can also be modified to strengthen the discriminator.

Incomplete Representation Hypothesis

We suppose that realistic and correctly labeled data is generated. Unfortunately for at least some classes, only a part of the distribution is generated. It implies that in the combined dataset distribution is incorrect with only the real data in the neglected part and too much synthetic data in the rest. Unless it is possible to identify such parts with, for example, labels there is no straightforward method to verify this hypothesis.

4.4.6 . Analysis and Correction

In our experiments, we evaluate each generator with at least 100 classifications, $N \geq 100$. The GAN training number, K, varies to limit the computing time needed and is specified for each setup. To begin with, we give the impact of trivial generators, Fig. 4.9 and Table 4.3.

In Fig. 4.9, the reference corresponds to the evaluation of d_0 : 58.2 ± 0.29. The interesting area is below the perfect generator and above the reference value. We thus know the limits on the classification quality. It is also important to take into account that for a small quantity of synthetic data added, the noise generator has a similar performance as the reference one. So, without statistically significant evaluation criteria, the noise generator may look as good for data augmentation. We start by establishing the diagnosis to obtain better performance than in Section 4.4.1. Unless specified differently, the proportion of synthetic data is 0.5 and the generation algorithm used is the ACGAN. The impact of this proportion is given later, together with the comparison between ACGAN and fs-InfoGAN.

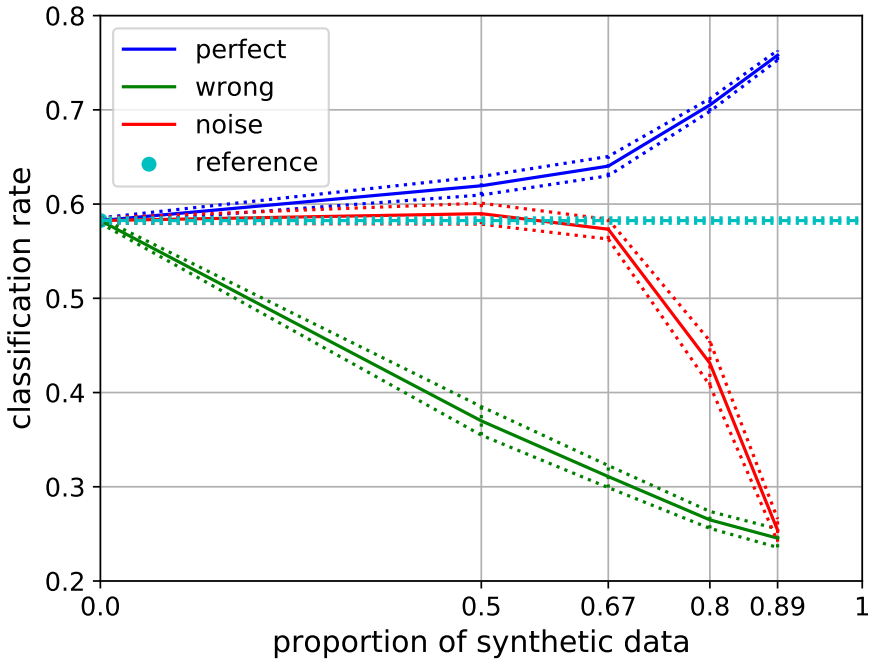


Figure 4.9 – Example of classification measure on d_0 dataset. The values displayed are also gathered in Table 4.3.

Differential Diagnosis

We use the calibration protocol to diagnose our GAN.

We observe that our generator G_{first} has a lower evaluation than the noise generator for low synthetic proportion but overcomes it for higher proportions (values re-given in Table 4.4). The synthetic data contains information and we are not under the *Non-representative data* hypothesis. As the performance of our generator is much higher than the wrong generator, we are probably not under the *Class confusion pattern* hypothesis either.

Now, we observe the different loss terms, Fig. 4.10, and draw the conclusion that G_d_f is very high compared to the other loss terms. The generator performs

Generator	synthetic proportion		
	0.5	0.67	0.8
noise	59.0 ± 1.11	57.3 ± 1.04	43.1 ± 2.36
G_{first}	54.5 ± 0.71	53.2 ± 0.70	51.7 ± 0.64

Table 4.4 – Comparison between a preliminary test and the noise generator, eval$(d_0) = 58.2 \pm 0.29$.

much worse than the discriminator. Besides, the classifications losses c_r and c_f are low. The *Non-realistic class-specific pattern* hypothesis is the most probable hypothesis.

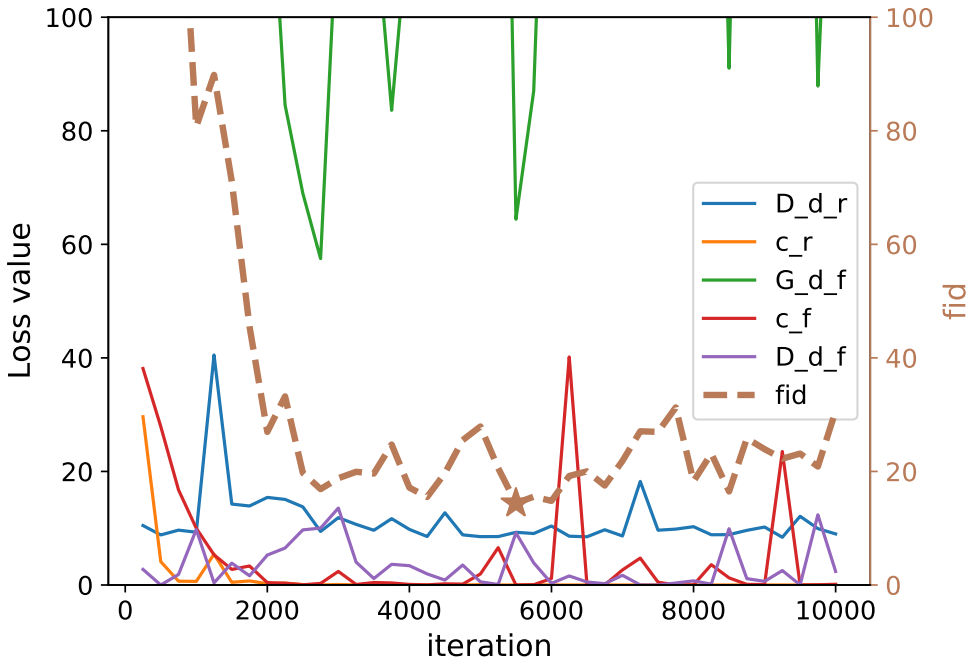


Figure 4.10 – Example of run before the calibration : G_{first}. Left-hand side y-axis shows the losses, the right-hand one shows the FID.

The correction consists in improving the generator. We modified one parameter we call GPD. GPD, *Generator training Per Discriminator training*, corresponds to the number of retro-propagations (updates) of the generator for each discriminator update. By default, its value is one. We need to increase this value.

In Table 4.5, we present the results obtained for different GPD values. We highlight that only one GAN training ($K = 1$) was performed and remain cautious in our interpretations. In addition to the evaluations with a proportion of 0.5 synthetic data, we give the minimum FID reached and the loss values at the iteration corresponding to this minimum.

Table 4.5 shows that increasing GPD leads to higher evaluations. We also observe that the G_d_f loss term decreases thanks to this correction, as expected. Having G_d_f, D_d_f, and D_d_r at a similar level is a necessary condition for good evaluations. We can also observe that c_r and c_f re-increase slightly in some cases. As the data becomes more realistic, the class-specific pattern, which was problematic, disappeared and the class recognition for synthetic images is

GPD	FID_{min}	D_d_f	G_d_f	D_d_r	c_r	c_f	$\text{eval}_{d_0}(G)$
1	14	9	64	9	$2e^{-3}$	$2e^{-3}$	54.5 ± 0.71
2	7	6	97	9	0.1	0.1	54.7 ± 0.62
4	10	22	14	14	0.1	0.1	60.9 ± 0.74
8	12	15	22	17	5.7	0.4	58.2 ± 0.52
16	13	13	23	22	0.5	0.4	57.7 ± 0.76
32	26	14	21	19	39.7	40.1	46.5 ± 1.47

Table 4.5 – Preliminary observation to calibrate our GAN ($K = 1$, $N = 100$). We remind that $\text{eval}(d_0) = 58.2 \pm 0.29$.

not trivial anymore. At the same time, as the discriminator learns less often, the auxiliary classifier also learns less.

All those facts indicate that we correctly identified the hypothesis explaining the behavior of our GAN : the *Non-realistic class-specific pattern hypothesis*. We will now apply an adequate correction by conducting statistically significant experiments.

GPD Correction

From the diagnosis established above, we concluded that GPD is a key parameter to calibrate our GANs. To find the most convenient value of GPD, we apply the evaluation procedure depicted in Fig. 4.8. The chosen criterion is the gran mean of the classification accuracy of the combined dataset. The obtained results are gathered in Table 4.6. We observe that the highest performance is obtained when GPD is equal to 8. We will therefore use this value while being aware that values of 4, 6, and 10 produced similar results.

Contrary to other works where the discriminator often learns more frequently than the generator [38, 29], our generator needs a strong aid to compete with the discriminator. It seems that for high GPDs (14-16), the generator performance re-increases. This result is surprising and needs further studies.

GPD seems to be a good means to balance the two networks. Numerous similar corrections can also be done to strengthen the generator such as a higher learning rate than the discriminator and further studies could be done in this area.

4.4.7 . Other Calibrations

After finding a manner to produce useful generators, we detail other research lines we explored. We start with the stopping criterion of the GAN learning to continue with the comparison of ACGAN and fs-InfoGAN. Eventually, we justify our choice on the synthetic data proportion.

GPD	K	$\text{eval}_{d_0}(A)[\%]$	$\text{eval}^{q_1}_{d_0}(A)[\%]$	$\%^{>}_{d}(A)$
1	20	55.3 ± 1.14	58.3 ± 1.80	10.0
2	20	56.0 ± 1.13	58.1 ± 1.74	0.0
4	40	56.6 ± 0.98	58.9 ± 1.41	7.5
6	40	56.8 ± 0.96	58.7 ± 1.39	5.0
8	40	$\mathbf{57.5 \pm 1.00}$	59.9 ± 1.46	20.0
10	20	57.1 ± 1.19	60.0 ± 1.92	15.0
12	40	55.5 ± 0.98	57.7 ± 1.37	2.5
14	40	56.2 ± 0.99	59.1 ± 1.41	10.0
16	39	56.8 ± 0.98	59.4 ± 1.43	12.8

Table 4.6 – $\text{eval}_{d_0}(A)$ in function of GPD; $\text{eval}(d_0) = 58.2 \pm 0.29$.

Stopping Criterion

Another problem for the GAN is the stabilization of the results as they tend to differ significantly across generator training. As the confrontation of the two networks is unstable, there is no clear stopping criterion. This confrontation can be balanced at some points to become unbalanced later. Among the potential stopping criteria, we suggested the iteration producing the lowest FID. This criterion suffers from some weaknesses but we did not find any better. Now, we detail two interesting trials for illustrative purposes.

The first example is made with GPD of 32. FID and the different loss terms are indicated in Fig. 4.11 and the values at the key iterations are in Table 4.7. The first five iterations chosen correspond to the five lowest FID values. The sixth one corresponds to an interesting moment with a large difference between D_d_f and G_d_f.

iteration	FID	D_d_f	G_d_f	D_d_r	c_r	c_f	$\text{eval}_{d_0}(G)$
1 250	26	14	21	19	40	40	46.5 ± 1.47
6 500	28	19	16	16	5	1	56.8 ± 0.53
5 000	29	16	18	19	11	2	55.6 ± 0.55
6 750	34	13	23	23	4	1	55.6 ± 0.59
5 500	34	14	21	23	11	1	55.6 ± 0.54
9 250	146	32	9	12	8	1	58.1 ± 0.74

Table 4.7 – Example 1, unsatisfactory results despite a low FID; $\text{eval}(d_0) = 58.2 \pm 0.29$.

As we see, the moment where FID is the lowest produces a very bad generator. It is due to the high c_r and c_f loss values. This fact clearly indicates the *Class confusion pattern* hypothesis. In this example, the best iteration is 9 250 with the high gap between D_d_f and G_d_f while D_d_r, c_r and c_f are low. The

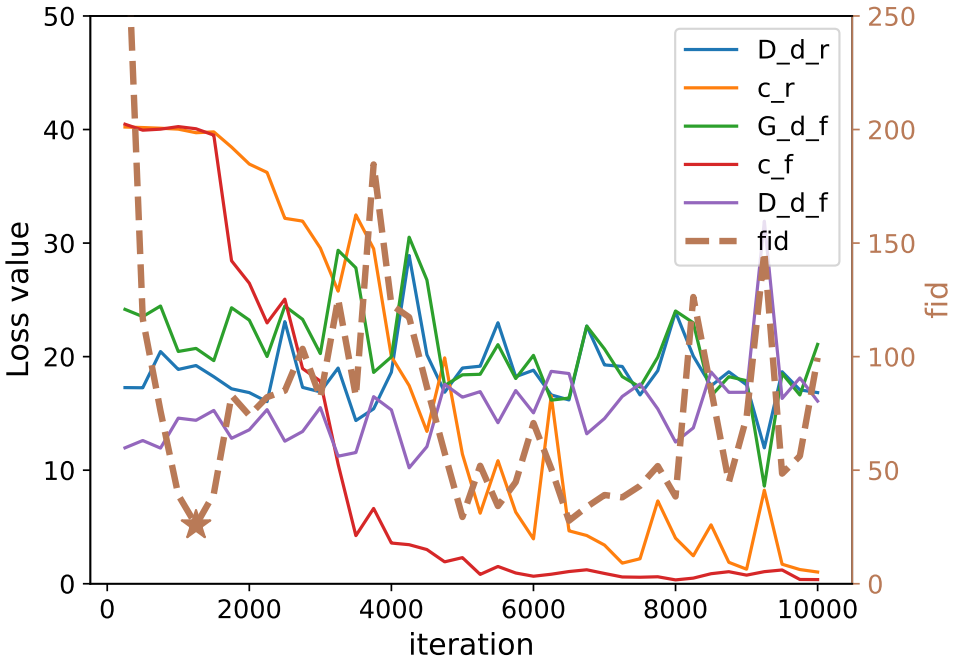


Figure 4.11 – Example 1, unsatisfactory results despite a low FID. Left-hand side y-axis shows the losses, the right-hand one shows the FID.

$9\,250$ iteration gives a generator that might look unbalanced because it beats the discriminator $D\_d\_f > G\_d\_f$ but the discriminator remains good on real data : D_d_r low.

In the second example, with GPD equal to 4, we observe a different result, summarized in Table 4.8 and Fig. 4.12. The best results are given when FID is low. In both examples, good results are obtained when c_r and c_f are low but this condition is insufficient (example 2, iteration $4\,000$). We also observe a significant result variation even for iterations close to each other (example 2, iterations $9\,000$, $9\,250$, $9\,500$).

We made numerous trials without finding any clear stopping criterion. These unproductive trials are not presented. Eventually, we selected the instant where FID is minimum as it was leading to pertinent results in most cases. This condition is based on the assumption that FID-realistic data is better than data obtained at an arbitrary iteration. However, further work should be done in this critical area.

ACGAN vs. fs-InfoGAN

We made several experiments with similar setups to compare the performance of ACGAN and fs-InfoGAN. The results are gathered in Table 4.9. We take note

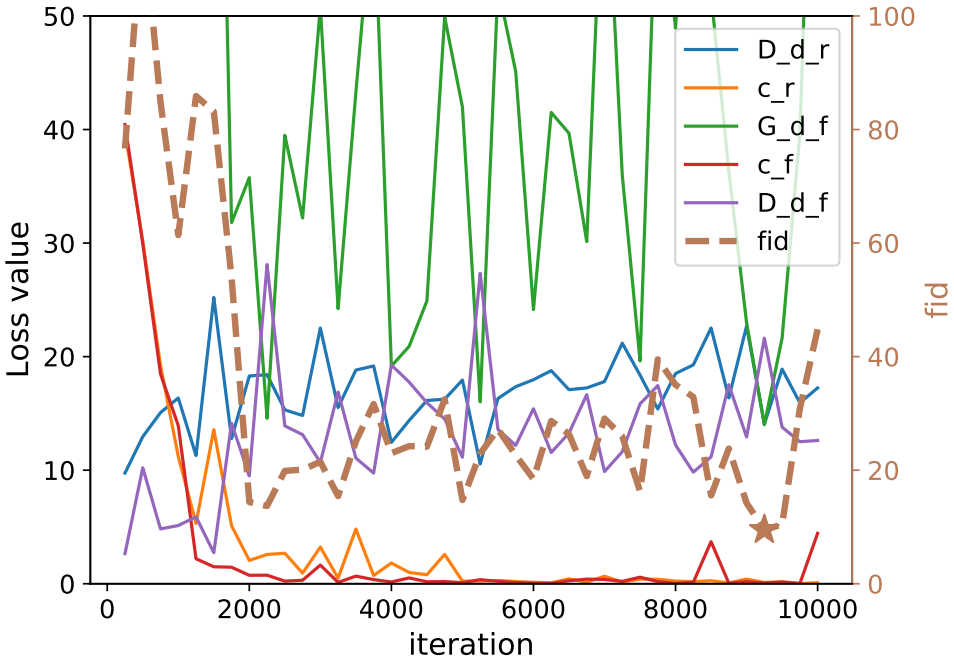


Figure 4.12 – Example 2, good evaluations for a low FID. Left-hand side y-axis shows the losses, the right-hand one shows FID.

iteration	FID	D_d_f	G_d_f	D_d_r	c_r	c_f	$\text{eval}_{d_0}(G)$
9 250	10	22	14	14	0.1	0.1	60.9 ± 0.74
9 500	10	14	22	19	0.2	0.2	59.8 ± 0.56
2 250	14	28	15	18	2.6	0.8	56.6 ± 0.55
9 000	14	13	23	23	0.4	0.2	57.7 ± 0.55
2 000	14	10	36	18	2.1	0.7	56.1 ± 0.55
4 000	23	27	16	11	0.2	0.4	56.2 ± 0.61

Table 4.8 – Example 2, good evaluations for a low FID; $\text{eval}(d_0) = 58.2 \pm 0.29$.

that the results are very similar but slightly better for ACGAN. We will therefore use this algorithm later on.

Proportion of Synthetic Data

Our combined datasets are composed of a half of the synthetic data and a half of real data. During the preliminary experiments, we did not observe any gain by taking a larger synthetic data proportion. More specifically, the generators

GPD	Algorithm A	K	eval$_{d_0}(A)$	eval$^{q_1}_{d_0}(A)$	$\%^{>}_{d}(A)$
8	ACGAN	40	$\mathbf{57.5 \pm 1.00}$	$\mathbf{59.9 \pm 1.46}$	**20.0**
	fs-InfoGAN	20	55.7 ± 1.14	57.6 ± 1.73	0.0
4	ACGAN	40	56.5 ± 0.97	58.9 ± 1.41	7.5
	fs-InfoGAN	20	56.4 ± 1.14	58.7 ± 1.79	10.0

Table 4.9 – Comparison between ACGAN and fs-InfoGAN with two adapted GPD values.

with good performance for this proportion maintain their performance for higher proportion while the performance observed for the bad ones continues to decrease with higher proportions.

We describe three examples of bad generators and three examples of good generators, illustrated in Table 4.10.

A proportion of 0.5 for synthetic data is sufficient to observe the usefulness of a generator. As a larger proportion just enlarge the training dataset without any improvement, we choose to keep the proportion of 0.5 in the further experiments.

Generator	synthetic data proportion		
	0.5	0.67	0.8
perfect	61.9 ± 0.98	64.0 ± 1.02	70.5 ± 0.67
noise	59.0 ± 1.11	57.3 ± 1.04	43.1 ± 2.36
wrong	37.0 ± 1.52	31.1 ± 1.17	26.5 ± 0.90
ACGAN$_{\text{bad}_0}$	55.5 ± 0.62	53.7 ± 0.71	51.4 ± 0.73
ACGAN$_{\text{bad}_1}$	46.5 ± 1.47	41.7 ± 1.32	40.1 ± 1.49
ACGAN$_{\text{bad}_2}$	56.4 ± 0.72	53.3 ± 0.95	50.7 ± 0.82
ACGAN$_{\text{good}_0}$	60.9 ± 0.74	60.6 ± 0.53	60.1 ± 0.66
ACGAN$_{\text{good}_1}$	60.3 ± 0.83	59.8 ± 0.70	60.1 ± 0.52
ACGAN$_{\text{good}_2}$	60.3 ± 0.71	60.6 ± 0.55	60.2 ± 0.54

Table 4.10 – Examples of generator evaluation with higher proportion of synthetic data; eval$(d_0) = 58.2 \pm 0.29$.

4.4.8 . Conclusion

The calibration procedure allows us to select the values of GAN parameters suitable for our purpose.

The key parameter to calibrate our GANs is GPD. We invested computing resources to find that the value 8 was a good choice.

We choose the FID stopping criterion, even if it is imperfect. We selected the ACGAN algorithm as its performance is slightly better than fs-InfoGAN. We

also fix the proportion of synthetic data to 0.5 as we did not observe any better performance with higher proportions.

In addition to the experiments presented above, other secondary experimentations were performed. We did not observe any other improvements with specific loss weighting. We also remarked that the label shifting did not need to be modified.

4.5 . Results

We conducted experiments on different training datasets with the networks calibrated according to our method. As data augmentation is more efficient when dealing with a small amount of real data, we first focus on small datasets. In the next step, we increase the size of the dataset to determine to which point the results remain acceptable. The classification gains, represented by the difference of the two classification rates $\text{eval}_d^{q_1}(A)$ are given in the Δ column of the tables summarizing our results.

4.5.1 . Small Real Datasets

To observe the impact of the chosen reduced dataset on the data augmentation, we start with 30 arbitrarily selected reduced datasets of size $|d| = 25$ (5 samples for each of the five drone classes). Their evaluations go from $44.9\,\% \pm 0.37$ to $62.3\,\% \pm 0.28$ with a gran mean evaluation of $54.5\,\% \pm 1.12$. We then apply the data augmentation on five datasets with an evaluation regularly spaced within the interval. The evaluation for the datasets is presented Table 4.11 and Fig. 4.13 with $K = 20$ generators.

$\text{eval}(d)[\%]$	$\text{eval}_d^{q_1}(A)[\%]$	Δ	$\%_d^{>}(A)$
44.9 ± 0.37	54.9 ± 2.15	10.0	50.0
50.5 ± 0.30	57.1 ± 1.84	6.6	55.0
54.3 ± 0.26	60.2 ± 1.96	5.9	80.0
58.2 ± 0.29	59.8 ± 1.85	1.6	20.0
62.3 ± 0.28	62.7 ± 2.10	0.4	5.0

Table 4.11 – Augmentation gain for real datasets d, $|d| = 25$.

For the best dataset under study (the last line of Table 4.11), the gain observed is small : only one generator is useful. It seems difficult for the generators to improve or even not to degrade a dataset with a too high evaluation compared to the median case. However, for the other four datasets under study, at least twenty percent of the generators are useful. Furthermore, $\text{eval}_d^{q_1}(\text{ACGAN})$ is significantly better than the evaluation of the real dataset for three of them with gains between 5.9 and 10.0 points. This suggests that we can obtain with a reasonable probability a useful generator for the majority of datasets evaluated.

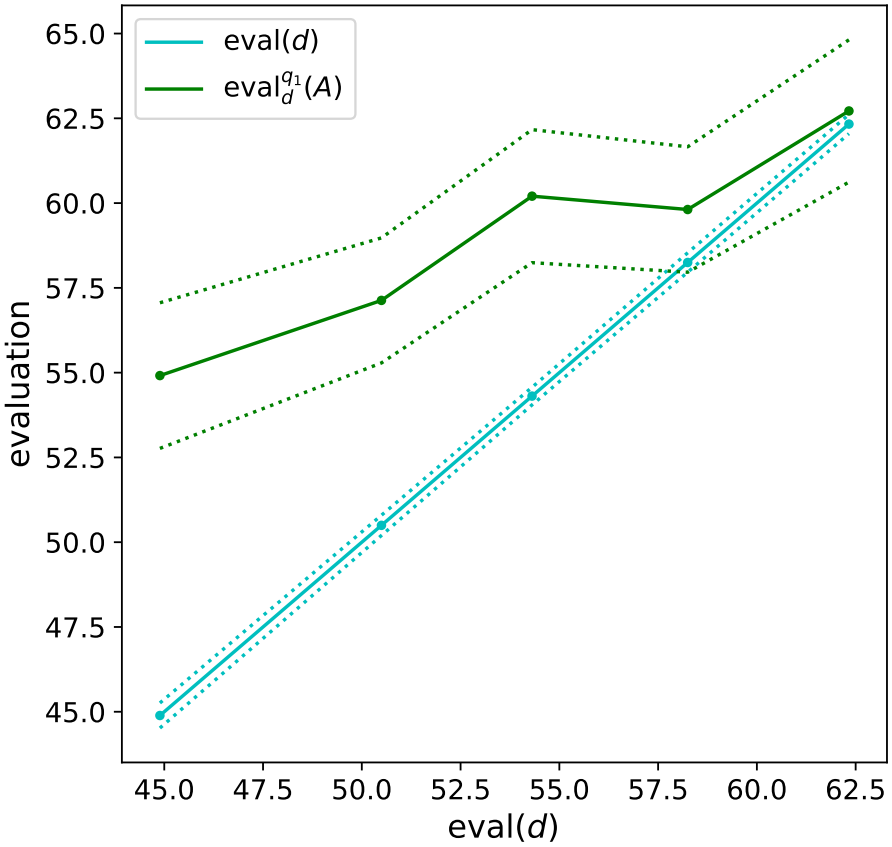


Figure 4.13 – Data augmentation gain for $|d| = 25$.

4.5.2 . Larger Real Datasets

We choose larger real datasets to observe the impact of their size on the performance of our data augmentation method. The constraint $|d| = |G(d)|$ is preserved

Due to the computing burden, data augmentation is applied on only one real dataset per size. We choose it by evaluating twenty datasets and picking one with the median evaluation. We assume that a deeper study with several datasets per size would give similar results.

Fig. 4.14 and Table 4.12 present the evaluation for each dataset size. We trained $K = 50$ ACGANs for each real dataset.

For larger datasets ($200 \leq |d| \leq 400$), the gain observed with our algorithm is low. The main criterion, $\text{eval}_d^{q_1}(A)$, gives no significant improvement but we still find useful generators. They are just less frequent than 25%. For the largest size, we evaluate that we have 16% of useful generators.

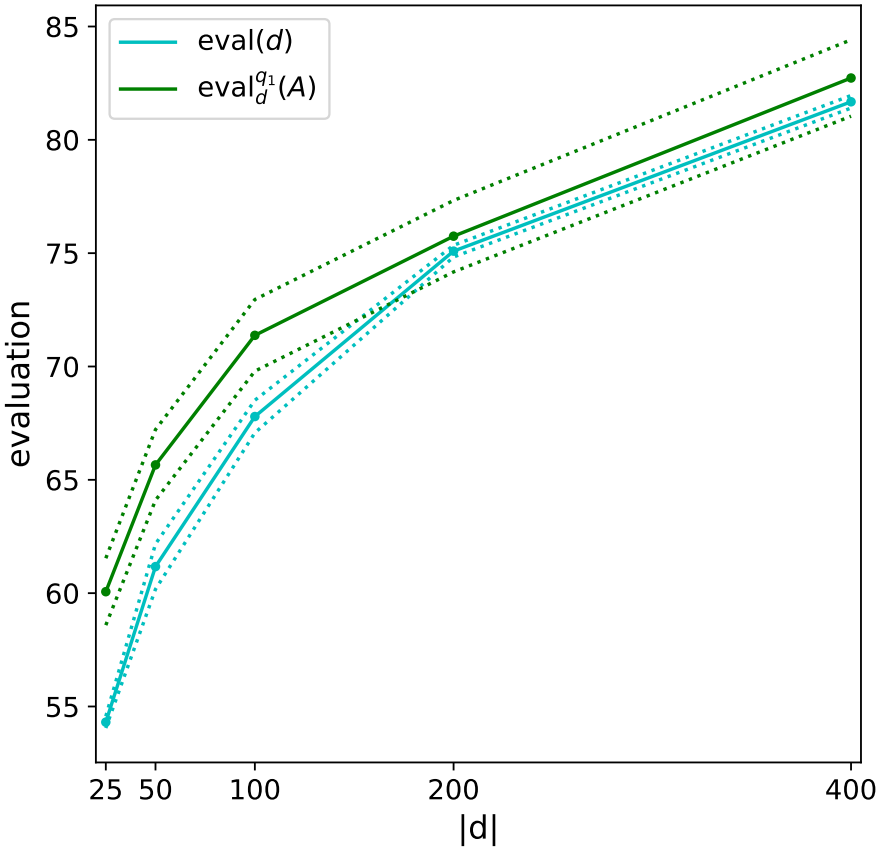


Figure 4.14 – Data augmentation gain in function of the reduced real dataset size $|d|$.

However, for smaller datasets ($25 \leq |d| \leq 100$), the gain measured is high. For the first criterion, $\text{eval}_d^{q_1}(A)$, a statistically significant gain of at least 3.6 points is observed. The number of useful generators overcomes 36% reaching nearly three-quarters of the generators for the smallest size which confirms the results from Table 4.11. Our data augmentation method significantly improves the original datasets, particularly small ones.

4.6 . Conclusion and Further Works

In radar applications, and most particularly in the case of micro-Doppler profiles, the data is expensive and always insufficient compared to the diversity of situations occurring in reality. For this reason, we studied data augmentation methods to tackle efficiently the problem of lack of available real data. We

$\|d\|$	eval$(d)[\%]$	eval$_d^{q_1}(A)[\%]$	Δ	$\%_d^>(A)$
25	54.3 ± 0.26	60.1 ± 1.48	5.8	74.0
50	61.2 ± 1.00	65.7 ± 1.56	4.5	36.0
100	67.8 ± 0.72	71.4 ± 1.58	3.6	40.0
200	75.1 ± 0.26	75.7 ± 1.58	0.6	8.0
400	81.7 ± 0.28	82.7 ± 1.68	1.0	16.0

Table 4.12 – Data augmentation gain in function of the reduced real dataset size $|d|$.

choose the GAN algorithms due to their potential. After introducing the different GANs, we developed an evaluation method based on the utility, not on the realism of the synthetic data. We establish a calibration protocol to identify the key parameters to setup our GANs. Eventually, we present the results obtained.

Even though the ACGAN algorithm studied accesses the same information as our classifier, it is still able to produce synthetic data which adds relevant information to an original dataset. Using a finely tuned ACGAN, we obtain generators producing micro-Doppler signals which significantly improve the classification of drones. The classification gain is prominent for small datasets and tends to decrease for larger ones. However, it is still possible to significantly improve the classification of larger real datasets under the condition of deploying computational resources to find an appropriate ACGAN.

We believe that similar results could be obtained in many other applications, efficiently tackling the inherent problem of too small datasets issued from expensive measurements. Besides, thanks to the evaluation measure we propose, other generative algorithms can be assessed to go further. Among the potential improvements, we highlight that establishing a stopping criterion different than FID to cope with the GAN instability seems a main research line.

5 - Data Augmentation with Ground Truth

We evaluated the pertinence of Deep Learning generative methods for classification improvement under the classical context of lack of data. In radar applications, a common approach to have more data without a costly measurement campaign is its generation with simulations. In Chapter 2, we saw that with the access to the drone states (drone logs) on real trajectories across time, these simulations contain pertinent information. The ground truth from the log files provides us with complementary information. The log information can be injected into the networks as it is. Another possibility is to compute radar profiles with simulations fed with these logs. In this chapter, we try to use this information to improve the data augmentation performance.

This new scenario is similar to the scenario of the previous chapter. We still have a small amount of real data (training) but we add complementary information in the form of real drone logs or simulated datasets obtained thanks to the drone logs. This scenario corresponds to the access of many trajectories without any radar involved or the existence of a mechanic simulator providing drone logs. In the remainder, all the drone logs come from real trajectories.

We assess the capacities of generative methods to produce useful synthetic data with complementary information. We keep the evaluation method from Chapter 4. However, now the objective is not only to obtain a significant gain relatively to the real dataset alone but also to beat the generative algorithms without complementary information.

We add the ground truth directly to the ACGAN from Chapter 4 and continue our experimentations by using the simulation data with a generative algorithm adapted presented below : pix2pix.

After presenting in detail the two forms of complementary information, we continue with the state of the art in domain generalization followed by the pix2pix algorithm. Then, we introduce our experimental protocol accompanied by the results.

5.1 . Ground Truth Representations

We aim at augmenting data with real data and complementary information (prior-knowledge). We detail the two forms of complementary information we have : drone logs and simulated data.

5.1.1 . Ground Truth

A drone records its parameters (rotor speeds, position, roll, pitch, yaw, etc.) during a flight. This information is a specific prior-knowledge as we know the drone state related to each radar data. Only the common information available of each drone is used (rotor speed, roll, pitch, yaw) but further works focused on drones giving more information (wind, etc.) would be a relevant research line. As the drones studied have 4, 6, or 8 rotors, we decided to represent the rotor speed values with an 8 elements vector. For a quadcopter, the four remaining components are set to zero.

5.1.2 . Simulation

The electromagnetic model used to obtain our simulated signal corresponds to Equations (2.5) – (2.10). We neglected the impact of the rotor radius by setting $L_1 = 0$.

The RCS values are obtained directly from real measurements of the training dataset (Chapter 3). We restrain our access to real data assuming at the same time that the RCS value distribution is available. This scenario is credible because the RCS values are usually obtained from measurements in an anechoic chamber.

These RCS values correspond to couples of cell and blade RCS values, $(\mathrm{RCS}_{\mathsf{cell}}, \mathrm{RCS}_{\mathsf{blade}})$ from the real profiles. We take the energy level from the WSP profiles. To obtain the cell RCS value, we collected at each instant the corresponding values in the WSP distribution. To obtain the blade RCS values, we computed at each instant a mean over the values corresponding to each blade. We consider that each rotor is equally visible with a constant RCS corresponding to the mean value observed. This hypothesis has been chosen because the phenomenon of rotors hiding each other is hardly predictable.

We consider that we know the distribution of these couples $(\mathrm{RCS}_{\mathsf{cell}}, \mathrm{RCS}_{\mathsf{blade}})$ without knowing which value corresponds to which maneuver as if we were measuring it inside an anechoic chamber. A random couple $(\mathrm{RCS}_{\mathsf{cell}}, \mathrm{RCS}_{\mathsf{blade}})$ is thus used to compute each simulated profile. Experiments carried out with more or less information on RCS values were made with similar results. For example, we also considered the hypothesis that these RCS values were not coming from direct measurement but only from a Gaussian approximation law.

The resulting simulated signal is normalized according to Equation (3.8). The maximum M and minimum m are obtained from the real training dataset and thus vary slightly depending on the training dataset used. For clarity, we call simulated data the data obtained by electromagnetic models while we continue to call synthetic data the one created by a neural network.

5.2 . State of the Art

In the majority of micro-Doppler studies using simulations and real data, the authors do not mix them (Section 4.1.1). The simulations are used in absence of real data or as a preliminary validation for further studies with real data.

In other applications, however, research works combining several domains to improve classification accuracy have been performed. We present the global concept of this field of study denoted *domain generalization*. We explain in detail one particular algorithm, pix2pix, that we use later.

5.2.1 . Domain Generalization

Principle

Only one domain is studied in most recognition problems, for example, the classification of objects in photographs. In reality, however, the same object can be represented in different domains (photograph, painting, sketch, etc.). The study of the relationships between different domains is called domain generalization [68].

The domain generalization methods can be used with similar or completely different domains. The likeness of the different domains depends on human interpretations. The simulated data and the real data of the same drone may be seen as two far domains. Two near domains would be the same drone measured in a different environment (weather conditions, locations, etc.). Generally speaking, domain generalization deals with the similarity between the training and testing datasets, including the definition of this similarity. It can be seen as a deeper study to fight over-fitting. If we show to a person a photograph of an elephant, he will understand the global concept of an elephant and not over-fit to recognize elephants in photographs only.

The domain generalization is used to benefit from easy-to-access domains (source domains) to classify data on hard-to-access domains (target domains). In our case, we have an easier-to-access source domain (drone flight recordings or simulated data) and we wish to use them to help the recognition of real radar signals.

We can split domain generalization methods into three categories : meta-learning [18], domain alignment [37], and data augmentation [68].

Meta-learning, also called "learning to learn", consists in algorithms which solve many different problems. The idea is to take advantage of a finely tuned algorithm adapted to several common problems to solve a new problem. The algorithm should be able to use these solved problems as a prior-knowledge about the world to learn a method to solve a new problem with only a few examples and learning steps. Such an algorithm is conceived for several solved problems and does not use directly the previous domains nor the link between the previous domains and the new one. It does not correspond to our situation.

Domain alignment consists in finding a feature space in common for the different domains. The assumption is that this common feature space is a conceptual space that would generalize the concept of the studied object. It

implies that this space would be automatically adapted to new domains. For example, in [37] an adversarial auto-encoder is used. The goal is to use the residual space as a conceptual space adapted to all domains simultaneously, to align all domains to a general one. As we only have two domains (real and simulated data), it does not seem to be appropriate to our case either.

Data augmentation consists in generating new data containing pertinent information. Thanks to the added information, we hope to reach a higher classification rate. In this context, the generated data can be obtained by using data from different source domains. For example, in [68] the algorithm used is like an ACGAN fed with data from different domains. The discriminator is replaced by a domain classifier with an output label of the size of the available domain. The generator learns to dupe the domain classifier by generating synthetic data which looks like it does not belong to any source domain. The auxiliary classifier Q, called in [68] the label classifier, helps the generator to maintain the class of synthetic data.

5.2.2 . Pix2pix Algorithm

The satisfactory performance of pix2pix [27], one of the Image to Image algorithms, makes it the most popular. As the generator of the pix2pix algorithm is a U-net, we start by presenting U-nets before introducing pix2pix. We continue with potential upgrades of the pix2pix algorithm.

U-net

A U-net [51] is a specific neural network architecture used to transform images from one domain to another (originally from images to segmentation maps). This architecture is based upon the encoder-decoder architecture enriched with specific links between layers as explained in the seminal article [51] reproduced here in Fig. 5.1. Such an architecture begins with a reduction (the encoder part), continues with a low dimension size in the central part (the residual part) to ends with a re-augmentation of the feature vector dimension (the decoder part). It allows one to treat a large number of feature maps while keeping the computing cost reasonable.

During the encoding process, the information coming from the input is contracted to capture only the most pertinent features. Then, during the residual process, the compressed information is analyzed to be eventually expanded and thus re-put in the correct dimension in the decoding part. Furthermore, some links (copy and crop arrows in Fig. 5.1) are introduced to associate specific layers of the encoder and the decoder. These links accelerate the learning process by tackling the well-known vanishing gradient problem. We highlight that in the decoding part, the layers are concatenated before the activation function. Thus, the same convolutional layer is used with two different activation functions (leaky relu for the encoder and relu for the decoder).

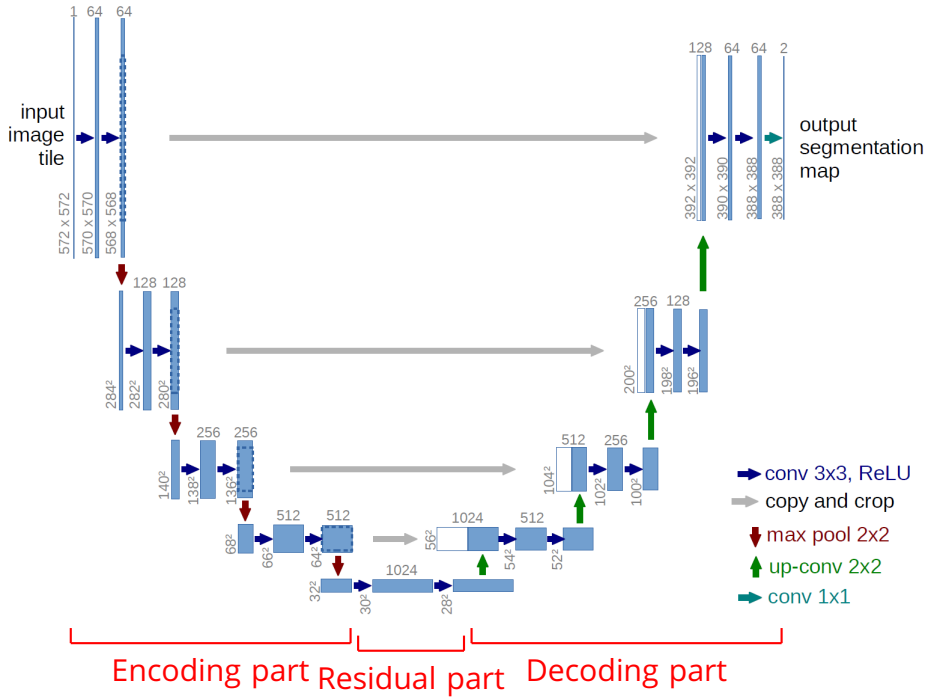


Figure 5.1 – U-net architecture example. This figure comes from [51].

Algorithm

The idea of the pix2pix algorithm is to make the generator learn the relationship between an input domain and an output domain. The generator, based on a U-net architecture, learns this relationship thanks to a cost function combining the cGAN (*conditional Generative Adversarial Network*) [40] cost and a classic pixel to pixel costs.

A cGAN is a specific GAN (Fig 5.2). It constraints the generator to produce a synthetic image which is realistic according to the input. Thus, instead of a noise input vector z, the generator input is a comprehensible element from a source domain. The association between the source and target domains is not created by an auxiliary classifier Q as in ACGAN/InfoGAN.

Instead, this link is obtained by modifying the input of the discriminator D which takes as an input the concatenation of the synthetic data $G(z)$ or the real data x and the source domain image z associated. The realism of $G(z)$ depends thus on the associated z. This approach is adapted to a high-dimension source domain (for example, of the same dimension as the target one) when the ACGAN approach is more adapted to a short feature vector.

In the discriminator, the source and target domains are gathered. For generated data, it corresponds to the data of the source domain z and the synthetic data produced from it $G(z)$ and for real data, it corresponds to the data of the source

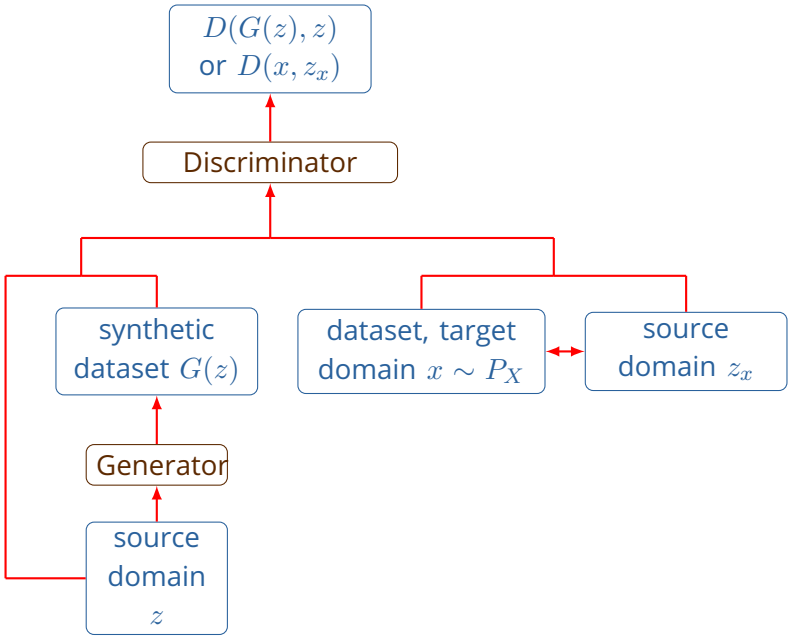


Figure 5.2 – cGAN scheme.

domain linked with the real data. The discriminator tries to recognize realistic data from the target domain corresponding to the source domain. The domain gathering is made in pix2pix by channel concatenation which is particularly convenient to source and target domains of the same dimension. In the majority of situations, the relationship between the two domains is precise : the bottom of the z image of the source domain corresponds to the bottom of $G(z)/x$ image of the target one. The loss of a cGAN is identical to losses of an original GAN with the exception of the modification of the discriminator input :

$$\begin{aligned} D(x) &\leftarrow D(x, z_x), \\ D(G(z)) &\leftarrow D(G(z), z), \\ \text{Loss}_G^{cGAN} &= E_{z\sim P_z}[\log(D(G(z), z))], \\ \text{Loss}_D^{cGAN} &= E_{x\sim P_X}[\log(D(x, z_x))] + E_{z\sim P_{z_x}}[\log(1 - D(G(z), z))]. \end{aligned}$$

The pix2pix algorithm is a cGAN with another loss term added. As for each z during training, the associated desired image y of the target domain is known, y and $G(z)$ are also directly compared according to the Manhattan norm. This norm is considered as more efficient than the Euclidean one according to several articles including [27]. A reason of this difference might be that the Euclidean norm strengthens excessively large errors between the produced and expected results, neglecting small ones. The loss formulæare :

$$\begin{aligned} \text{Loss}_G^{pix2pix} &= \text{Loss}_G^{cGAN} + \|y - G(z)\|_1 , \\ \text{Loss}_D^{pix2pix} &= \text{Loss}_D^{cGAN}. \end{aligned} \tag{5.1}$$

We detail the most relevant pix2pix characteristics.

It uses instance normalization instead of batch normalization [62]. It does not use any full-connected layer at the end of the discriminator (PatchGAN). So, the discriminator output is not a number between 0 (synthetic data recognized) and 1 (real data recognized) but a 2-D vector (a feature map). Each element of this vector represents a specific part of the original image and values 0 for $G(z)$ or 1 for x.

The pix2pix algorithm is considered as a GAN with only a "minor stochasticity" which prevents it from capturing "the full entropy of the conditional distribution" since the input vector is deterministic and the random influence is present exclusively inside the network as a constant dropout in some layers. Except for segmentation map purposes, where the desired output is known, the pix2pix algorithm is only evaluated with realism criteria (FID, human interpretation) for the generation of synthetic data.

Potential Upgrades of Pix2pix

Due to the popularity and the efficiency of the pix2pix algorithm, numerous studies [63, 64, 67, 69] were carried out to develop its derivatives. We detail some of them [63, 67].

A first example is the cycleGAN [69]. A cycleGAN deals with two domains but an element of a domain is not linked to a specific data of the second domain. The purpose of the algorithm is to re-create this link. The algorithm is based upon weaker hypotheses on the domains and has thus lower results. In our drone application, this link is known. So, we did not use the cycleGAN algorithm and will not detail it.

In [63], a controller is added to the algorithm to reinforce its stability and improve its performance. The controller checks if the synthetic image corresponds entirely to the associated source image by trying to reproduce the latter given the synthetic image only. The same concept is also used in cycleGAN but here the association is known. The evaluation of this method is not a classification improvement on specific datasets but the quality of the generated image for target domains composed of segmentation maps. The results show an encouraging performance. However, as the evaluation is not made with robust statistics (5 runs only), the gain of this method compared to pix2pix cannot be confirmed and does not seem significant.

In [67], the pix2pix algorithm is used for de-raining images. De-raining images correspond to filter out the rain impact (raindrop obstruction, deformation, and

background blurring). The idea is to obtain the same picture without the rainy environment. The algorithm is slightly modified, mainly its loss function. Thus, the non-GAN loss term is a Manhattan norm comparison not only between the ground truth (a clean image) and the reality (a rainy image) but also inside the generator at a feature level between different layers output of the generator. To compute this cost, the ground truth is also sent into the generator during the learning. Other modifications are made, for example, the DCGAN conditions are not fully respected, $\beta_1 = 0.9$ and not 0.5 for ADAM optimizer. The improvement from this modification is measured thanks to visual criteria : SSIM (*Structural SIMilarity*) and PSNR (*Peak Signal to Noise Ratio*) which are once again not statistically significant (1 run only).

From these three examples, we conclude that the majority of studies consists in adapting, improving, or using pix2pix directly. The more recent algorithms for similar problems are based on pix2pix which thus remains the principal approach. As no statistically significant improvement has been shown, we use the original pix2pix algorithm in our experiments.

5.3 . Experimental Protocol

Our objective is to use the source domains (drone logs / simulated data) to force the generative algorithm to generalize the distribution of the target domain (real data) with synthetic data. We keep the same test dataset as before and continue to use reduced training datasets.

As for some trajectories, the log files were not collected, we had to discard them to work in a full supervised mode. Despite the removal of training datasets for which the drone log is missing, we manage to preserve a similar evaluation with our new training datasets. We did not observe a significant impact after discarding these trajectories during preliminary trials. The new selected dataset d_1 has a size of 25 and an evaluation of 55.6 ± 1.02 which is close to the median case.

The ground truth of the training trajectories of Chapter 3 is considered as available for the neural network contrary to the ground truth of the testing one

We have two categories of log information :

— log files corresponding to radar profiles available : radar profiles in the training dataset.

— log files corresponding to non-available radar profiles. These log files comes from real trajectories without radars involved.

We use log information in three manners :

— l_g : log used as an input for the generator during generator training,

— l_c : log used as an input for the generator after its training for the data augmentation purpose,

— l_d : log used for the training of the auxiliary classifier Q corresponding to real data.

Fig. 5.3 illustrates the different manners to inject logs into an ACGAN or fs-InfoGAN. These manners can be extended to the other algorithm such as pix2pix.

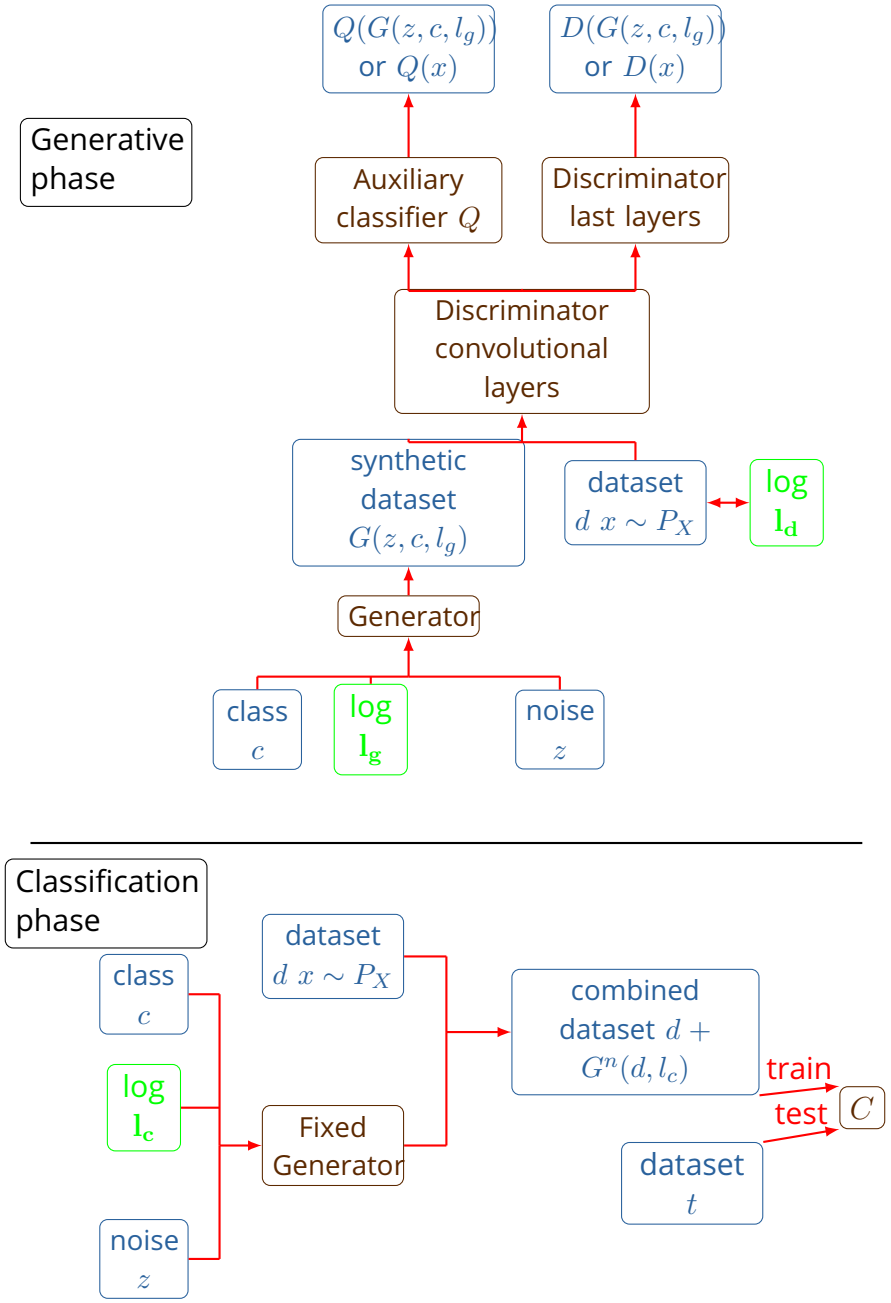


Figure 5.3 – Different manners to use the drone logs for the data augmentation procedure.

For the auxiliary classifier training with l_d, only the log corresponding to real profiles can be used. This is not the case for the two other manners l_g and l_c which can also contain logs from unavailable radar profiles. Furthermore, it seems logical to use the logs of available profiles in l_g. If during training the generator produces only synthetic data differing from the real data, it may disturb the balance between the two networks, thus, $l_d \subset l_g$. Besides, logs corresponding to non-available data in l_c may be used to push the generator outside its comfort zone.

As the log information is captured from the internal components of the drone and the real data is obtained from the radar measurement, they do not have the same acquisition rate. The different values even of the same drone logs have in some cases different acquisition rates. As this rate is high compared to the observation length (300 ms), we re-sampled the drone logs to have one of them per profile. This operation was possible because the rotor speeds do not vary too much during the observation length (the stationary hypothesis). The re-sampling was done by linear interpolations.

The logs are normalized by the same formula that the real datasets, Equation (3.8), with M the maximum, and m minimum modulus present in the log dataset. The rotor speeds are normalized with a unique M and m while the other log information (roll, pitch, yaw) are normalized with their own extrema.

5.4 . Results

To begin with, we discuss the preliminary result obtained with an identical setup to the one which allowed us to get the best results without drone logs (Section 4.5).

As the performance is worse than expected we continue with an analysis of the potential explanations of this fact. However, as we were not able to solve this issue, we detail several unsuccessful attempts, outline other investigated trials, and potential explanations for this situation.

5.4.1 . Preliminary Results

We start with a simple case, similar to the setup of the one giving the best results without drone logs. This choice is justified partly by the fact that the informative part of the network is assumed to be "for free" [11] because the network should do this new job without an increase of its depth or any other modification.

We impose that $l_d = l_g = l_c$ and use only the rotor speeds as the ground truth vector. This simplification is justified by the electromagnetic models which show that the rotor speed is the predominant factor.

The results are given in Table 5.1. In order to assess their quality, this table also contains the results from Chapter 4 without logs.

We observe that the new method increases the classification rate in comparison to the case with no data augmentation. Nevertheless, the data injected does not allow us to improve the performance compared to the previous case (lower Δ and

drone logs	eval$(d)[\%]$	eval$_d^{q_1}(A)[\%]$	Δ	$\%_d^{\geq}(A)$
without	54.3 ± 0.26	60.2 ± 1.96	5.9	80
with	55.6 ± 1.02	59.4 ± 1.51	3.8	46

Table 5.1 – Comparison between results from data without logs and data enriched with logs.

$\%_d^{\geq}(A)$). We analyze this case enumerating possible explanations which should guide us later.

5.4.2 . Analysis

A half of our generators are useful which is an impressive result by itself. However, we aim to improve the results obtained without logs. We hypothesize that the tools developed for the calibration (Section 4.4) may be insufficient. We start by detailing the most plausible explanations of this disappointing performance to pursue with a summary of other potential hypotheses. Thereafter, we carry out the resolution of some of these hypotheses.

Loss Balance Hypothesis

The first reasonable possibility is that the problem comes from the loss function. As we have different objectives expressed through the different loss terms, the combination of them may lead to a local minimum. In Chapter 4, our main calibration issue was a loss problem with $G\_d\_f \gg D\_d\_f$. As we keep a GPD of 8, we did not observe an imbalance between G_d_f, D_d_f, and D_d_r. However, the loss corresponding to the continuous latent is computed with the Manhattan norm. It thus has no reason to give directly a similar order of magnitude than the other loss terms which are cross-entropy terms.

Generalization Difficulty Hypothesis

Another problem is the diversity of signals. We ask the network to understand both the realism of the micro-Doppler profiles and the relationship between the profiles and the drone logs with small datasets. This algorithm is not conceived for a dataset as small as five elements per class.

We assume that our objective is too ambitious. This hypothesis is justified by the fact that the algorithm continues to have a good GAN behavior : high Δ and eval$_d^{q_1}(A)$. It simply gives up the new objective of associating the drone log and the profiles.

Simulation Representation Hypothesis

Another hypothesis is that we ask too much with too little. Perhaps, the space representation of the complementary information is too simple. The network not only has to understand that the high peaks in the signal correspond to the rotor speed values but it must also randomly hide some rotors and produce a realistic environmental noise.

Another representation of the complementary information may aid the network to fulfill its objective.

Other Hypotheses

To complete our list of hypotheses, we investigated numerous other possibilities.

A first conjecture would be the great diversity of the requested tasks. As the data is diverse both in terms of energy level (RCS) and rotor type (number, blade length, etc.), it results that the same ground truth vector has a different interpretation for each class. It might confuse the generator.

A second supposition concerns the ground truth precision. We consider that each profile is independent with one ground truth per profile. We do not know the previous flight instants leading to this situation.

For each acquired signal, we have the drone log corresponding to a given moment. However, it may potentially not be the most pertinent moment : an offset between the drone log value or simply the fact that the state of the drone preceding a measured instant impacts it. So, considering a concatenation of drone logs might help the network to catch more efficiently the association between log and signal leading to better synthetic signals.

We now detail the resolution axes of our main hypotheses. To reduce computing cost, evaluations presented below are mostly made with a number of ACGAN training K equals 10. For the classification runs, we keep $N = 100$. Computing resources are invested in the most promising results ($10 \leq K \leq 100$) depending on the indicators from Section 4.4.4. For this reason, the confidence interval at $95\,\%$ of the different runs varies considerably.

5.4.3 . Loss Balance

An efficient solution to the loss balance problem may be an appropriate weighting. By decreasing/increasing a part of the global objective, we try to restore the balance. Besides, in the seminal article of InfoGAN, the weighting optimization of continuous latent is indicated as primordial, even if, as we observed in the previous chapter, the categorical loss weighting has no impact.

We present in Table 5.2 the experiment conducted for the continuous latent weighting λ_l (l for logs). The weighting of the categorical latent λ_c (c for class) is set to 1.

λ_l	eval$(d_1)[\%]$	eval$_d^{q_1}(A)[\%]$	Δ	$\%_d^{>}(A)$
0.01	55.6 ± 1.02	60.6 ± 2.17	5.0	60
0.1	55.6 ± 1.02	59.5 ± 1.54	3.9	38
1	55.6 ± 1.02	59.4 ± 1.51	3.8	46
10	55.6 ± 1.02	51.3 ± 8.57	-4.3	0

Table 5.2 – Data augmentation gain in function of the weighting of continuous label. The weighting of categorical label λ_c is set to 1.

We do not observe any significant gain despite weighting optimization even if the loss terms are more balanced. The experiment with $\lambda_l = 0.01$ is slightly better than those with $\lambda_l = 1$ but the gap is not significant. We also tried different weighting of λ_l for other λ_c values with no improvement either.

As for many hyperparameters, the optimization of λ_l has a small impact on the performance as long as λ_l is in a coherent range.

We thus discard this hypothesis. For the following experiments, the presented results correspond to a weighting of $\lambda_c = 1$ (we also tried $\lambda_l = 0.01$).

5.4.4 . Larger Datasets

We work with larger datasets to avoid the insufficient size problem. This hypothesis is tricky because we want to work with test datasets much larger than the training ones. We may think that datasets of size 200 or 400 would give the network enough examples to find the association between the ground truth and the signal.

In addition, for the dataset d_1 the mean evaluation of the perfect generator (Section 4.4.4) is $eval_{d_1}(\text{perfect}) = 60.7 \pm 0.30$, a value close to our evaluation eval$_d^{q_1}(A) = 59.4 \pm 1.51$, even if it corresponds to the best quartile only. As it is very unlikely that we can perform better than the perfect generator, we assume that we will need larger datasets in order to observe an improvement.

We took a fully labeled dataset of size 200 with evaluation of 74.7 ± 0.48. We chose this size because it was the smallest size for which the data augmentation gain was minor (see the previous chapter for details). The results with and without drone logs are given in Table 5.3. We observe no improvement and even a degradation as we did not obtain any useful generator.

drone logs	eval$(d)[\%]$	eval$_d^{q_1}(A)[\%]$	Δ	$\%_d^{>}(A)$
without	75.1 ± 0.26	75.7 ± 1.58	0.6	8
with	74.7 ± 0.48	72.9 ± 1.52	-1.8	0

Table 5.3 – Comparison of the best data augmentation case with and without drone logs for a larger dataset.

We present one case with a larger dataset among the different trials we carried out. It seems that in any case, a method giving an interesting performance for very small datasets is confronted with a strong decrease for larger datasets. It compromises the hypothesis of a dataset too small even if we can still argue that we are far from a big data context. We formulate several interpretations of this result.

First, we can consider that the performance of the dataset alone is too high so any improvement is hard to obtain. Second, we might consider that the generalization capability of the generated methods is not sufficiently powerful even in the case of drone logs, because the role of the drone log c, pushing the generator outside its comfort zone is not fulfilled. Third, we can still consider that the training dataset is too small as it should contain thousands of profiles.

5.4.5 . Simulation Format

As the drone logs failed, we can use other complementary information. For each drone log, the simulated signal associated may be computed. This simulation is complementary information conceived to be closer to the real signal as it also is a micro-Doppler profile. This fact should help the generator.

In this case, the generator would generate realistic data by using the simulated data and adding a realistic style. This problem is a domain transfer from two spaces (the simulated and the real data) of the same dimension.

We do not use directly the ACGAN algorithm for two reasons. First, it would need a big auxiliary classifier (2400 components for the last layer) resulting in an unacceptable computing time. Second, we have two domains (simulation and real) with the same dimension and metrics, so the architecture of the pix2pix algorithm is more appropriate.

We describe one pix2pix trial in which the generator has 8 layers for the encoder and decoder parts and the discriminator is composed of five layers. The dataset d_1 is used and the results are given in Table 5.4.

drone logs	eval$(d)[\%]$	eval$_d^{q_1}(A)[\%]$	Δ	$\%_d^{>}(A)$
without	54.3 ± 0.26	60.2 ± 1.96	5.9	80
with	55.6 ± 1.02	58.5 ± 1.53	2.9	38

Table 5.4 – Comparison between results between the ACGAN without drone logs and the pix2pix with simulation.

The preliminary observation of the pix2pix behavior was promising. Indeed, pix2pix trials produce interesting indicator values in terms of FID or a loss balance. The mean FID over $K = 50$ generators was 7.8, dropping down to 3.7 in the best run which is low compared to our other results. For example, the calibration experimentation commented in Section 4.4.6, contains useful generators with FID values higher than 10. However, according to Table 5.4, the new generator obtained

is less useful than the one we obtain earlier. The pix2pix algorithm seems to be efficient to produce realistic signals but they are useless for our purposes. We advance several interpretations of this phenomenon.

The location of the peaks is primordial in micro-Doppler profiles whereas in images this criterion is secondary. As the FID measurement is based upon the features obtained by convolutional layers, we assume that the precise peak position has not a large impact on the FID value making it less pertinent for our case. Moreover, the pix2pix algorithm is known to stay in its comfort zone.

5.4.6 . Other Hypotheses Investigated

Several other hypotheses were also investigated. We present some of them briefly.

— At the origin, the objective was to benefit from the fact that l_g and l_c are larger than l_d. We detailed above the experiments with $l_g = l_d = l_c$ but we also tried larger sets to force the network to generalize the distribution. We first increase l_c by adding logs from unavailable data close to the logs of l_d according to Euclidean distance to continue with farther logs.

— As the performance of our ACGAN without logs was almost as good as the performance of a perfect generator, we assume that a larger proportion of synthetic data in the combined dataset would be an interesting research line. The drone logs could force the network to generalize the distribution and make it reach a higher performance. However, we did not observe any improvement with this method. The useful generators maintain their performance while the useless ones degrade it.

— Another possibility is a dataset problem because the dataset used for calibration was too specific, preventing us from making any improvement. We tried the protocol on other datasets and we attempted a semi-supervised approach with ss-InfoGAN to directly use the dataset d_0 of the previous chapter. These attempts were not successful either.

— Another possibility is the representation of the drone. We first sorted the rotor speed vector. The problem is thus simplified as the rotor location is removed. We also investigated an extension of the drone logs with roll, pitch, yaw, altitude to verify whether another latent has an impact. The performance seemed a little higher but still not as high as we would wish. Other norms to compare the loss terms associated with the ground truth such as an Euclidean norm were also taking into account. All of these attempts failed.

— We tried once again to compare the algorithms fs-InfoGAN and ACGAN. One should not forget that the ACGAN was originally designed for categorical latent. Besides, as a continuous latent seems harder to catch, the loss term c_f in the discriminator loss could be a reasonable

explanation. The experiments made us observe a similar performance for our main criteria, $\text{eval}_d^{q_1}(A)$, but a higher performance for the secondary ones, $\text{eval}_d(A)$, $\%_d^{>}(A)$, for fs-InfoGAN which is thus more stable than ACGAN with a higher percentage of useful generators. So fs-InfoGAN is slightly better but insufficient to overcome the performance of the previous chapter.

— Numerous other unsuccessful pix2pix trials were conducted. We conjecture that the main identified criteria, the micro-Doppler peaks, were not recognized as the primordial feature by the generator. The generator probably focused its attention mostly on the environmental noise. We added Gaussian noise levels to blur this part and thus insist on the peaks. First, we tried to add different noise values to the simulation at each iteration without success. Second, we added the noise outcomes to the real signal. The idea was to prevent the discriminator from paying too much attention to these secondary values. It led us to more balanced algorithms (`G_d_f`, `D_d_f` and `D_d_r`) but the poorer quality of the synthetic signals due to the blur.

The highest performance for small datasets was obtained for an fs-InfoGAN with rotor speeds sorted and weighting coefficients equals to one, Table 5.5, however, it remains poor compared to the one obtained without logs (Chapter 4).

drone logs	$\text{eval}(d)[\%]$	$\text{eval}_d^{q_1}(A)[\%]$	Δ	$\%_{d_1}^{>}(A)$	$\text{eval}_{d_1}(A)$
without	54.5 ± 0.94	60.2 ± 1.17	5.7	65	57.6 ± 0.88
with	55.6 ± 1.02	60.4 ± 0.88	4.8	72	57.7 ± 0.67

Table 5.5 – Comparison of the best data augmentation with and without drone logs for a small dataset.

5.5 . Conclusion

We thoroughly designed several propositions to incorporate information from drone log files with respect to the electromagnetic models and human observations. Despite our efforts, we did not find any manner to exploit the ground truth and outperform the data augmentation without the log information. We identified several potential explanations. We could not validate certain hypotheses because their potential was relatively weak compared to their complexity or simply because we do not have the technical means to do this.

If the problem comes from the difficulty induced by the diversity of the data, it can be solved by simplifying it with a generation algorithm per drone. This hypothesis can be carried out but we discard it as it implies a larger complexity despite relatively low expectations. Indeed, in Chapter 4, the preliminary runs with

a GAN per drone did not give good results and the vanishing of the classification loss terms, c_f, c_r, could lead to trespassing between the classes during the generalization.

The second not investigated other hypotheses is the ground truth precision as detailed in Section 5.4.2. It can be solved with 2-D images because the previous instant leading to this position may help the generator. The datasets, the evaluations, and the networks would have to be different which would make impossible the comparison with the previous results. For example, the datasets available would not be random elements corresponding to the real case of short instants captured with non-cooperative targets but rather consecutive instants, strongly dependent. For this reason, we did not investigate this lead.

Another possibility is to work with greater datasets, containing thousands of micro-Doppler profiles. This approach would need a tremendous computing time both for the generation and the classification. We would not be in the same scenario of a small available dataset compared to the reality (the test dataset).

Another hypothesis could be that the information present in the simulation outcome cannot bring classification improvement although it was created with a real drone log. If it was the case, more precise electromagnetic models would be needed. This hypothesis suffers from the fact that the drone log itself does not bring any improvement. So the generative algorithms used are not efficient enough to catch the information from the drone log. They could take it from the simulated data but the latter is too basic to be pertinent. Consequently, this hypothesis is not the most probable.

The generative algorithms we studied are not originally conceived to produce useful generators but to produce realistic images. It is thus difficult to affirm whether this research line has a potential that we could have observed with longer studies. The need for robust evaluation methods is crucial and we hope that the method we proposed will be adopted by the scientific community. Relevant evaluation methods should lead to the design of smart algorithms aiming directly at maximization of classification quality.

At last but not at least, we point out that the objective of this chapter was ambitious as we have already observed a significant classification gain without logs.

6 - Conclusion

To conclude this study, we remind our main contributions on drone recognition with micro-Doppler signals. Thanks to the Deep Learning tools we investigated the classification accuracy under different conditions. A large part of our work can be applied in other Deep Learning applications such as the data augmentation. The possible lines of research which emerge from our work will be presented in the next chapter.

6.1 . Space Representation for Micro-Doppler Signals

We highlighted the influence of the space representation on the classification quality. Even if the Deep Learning tools are conceived to create their own feature space, we observed that the classic pre-treatment tools used in radar applications have a statistically significant impact upon the classification accuracy.

Among the most common formats, we identified the WSP format as the one giving the best accuracy for the standard classifier GoogLeNet. As we did not take any advantage of any particularity of GoogLeNet, we assume that this result is generalizable for any other neural network. Another format, the CP format, also attracted our attention. However, we remind that our main goal is not the results themselves but the evaluating method of the different formats.

One of the interpretations of the different reactions of the classifier on the input format is the presence of biases in the data, probably due to its insufficient quantity. As data is difficult to collect, especially for opportunity targets, this status quo will be maintained. The creation of new formats is a reasonable solution to improve classification performance. The evaluation method we proposed is adapted to assess the usefulness of any emergent format.

In particular, we warn about the data distribution problem for the training and testing sets. Biases, such as data measured the same day in training and testing for micro-Doppler applications, impact the results leading to unreliable evaluations.

6.2 . Data Augmentation for Micro-Doppler Signals

To tackle the lack of data available, we studied GAN algorithms to augment the data quantity. Our techniques are general enough to be applied in any domain as they do not need any radar-specific knowledge.

Our first contribution in this subject is the development of a new evaluation method for generation algorithms. This evaluation is based on the utility for the further classification, not the realism of the synthetic data.

Besides, we set up a calibration method to fight against the GAN instability and find a good GAN tuning with a reasonable computing effort. This calibration makes

GAN generate synthetic data which improves the classification rate. This result leads us to our second contribution : we significantly improved the classification rates thanks to synthetic data obtained by a GAN algorithm.

This result encourages data augmentation approaches with GANs. Of course, new algorithms dedicated to pursuing directly the evaluations based on the utility of the synthetic data should be better than the algorithms studied as they were originally designed to promote the realism.

6.3 . Conjugation between Ground Truth and Micro-Doppler Signal

To improve the data augmentation performance we studied the combination of the ground truth and the real signals in a Deep Learning approach. Even if we were not able to overpass the performance obtained without this ground truth, we still observe a data augmentation gain. However, the investigations conducted allow us to capture potential explanations and to eliminate numerous hypotheses.

7 - Further Research

We identified numerous pertinent research lines that we could not explore within this PhD thesis. Some of them were not explored due to a lack of time, others due to a lack of technical means. We summarize these lines of research and provide the reader with our opinion on their potential.

7.1 . Radar Limits

Even if we collected the largest dataset of micro-Doppler signals, the measurements gathered would be insufficient to cover the diversity of real situations. Enriched by our experience, we formulated guidelines for future measurement campaigns which are commented below.

7.1.1 . Environmental Conditions

Measurements are made under basic conditions to have the best observation possible of the micro-Doppler effect. During our experiments, we set ourselves out of a comfort zone by collecting a large diversity of data (trajectories, meteorological conditions, etc.). However, due to the cost and legislation issues, we did not collect data in many pertinent on-the-field scenarii requested for an operational defense system. We illustrate this statement with some pertinent examples.

The future data collections should include measurements under a large clutter with very low SNR obtained for example at lower altitudes. The study of the degradation of the micro-Doppler classification becomes primordial to observe the limitations of the methods. We added a Gaussian noise to be in similar conditions but the environmental influence is not that simple.

Data was collected in one location which may lead to a bias. We observed the impact of the day separation between training and testing sets. Similar experiments should be made with location separation to measure an eventual degradation of the classification quality.

Generally speaking, one of the main limitations in radar applications is the measurement of consecutive instants of a drone trajectory. Ideally, numerous short trajectories under various conditions should be performed, even if such a measurement campaign is harder to carry out.

7.1.2 . Target Measured

We conducted our studies with five commercial drones of different sizes and configurations (rotor amount, etc.). Larger studies with more drones collected might be interesting but should result in a similar performance if the amount of data per drone is large enough. We thus do not think that it would be primordial

in itself from a research point of view. It may however become more beneficial under several conditions.

We assumed that the drones were commercial, non-stealth. In practice, the development of measurement capacities should lead to the development of countermeasures. For example, we also made measurements with plastic blades instead of carbon blades. We did not discuss the results as we did not observe a significant impact of such measurements during our short observations : the RCS response is just slightly reduced for plastic blades and so is the classifier performance. However, other scenarii often applied for planes could be adapted for drones.

We restricted ourselves to the measurement of one target at a time but drones may also fly in formation, close to each other (a drone swarm). In such configurations, the drones could confuse the radar which would capture one large target. Drones could also be hidden by other targets such as birds. Currently, the research on drone recognition is based on simple measurements. Nonetheless, such research lines, denoted-counter-countermeasure, tackling efficiently such scenarii should be developed in a near future.

One of the recognition capacities to improve in the short term is the discrimination between birds and drones. We had difficulties capturing bird data during our measurement campaign. We think that the method we developed should be efficient for birds if a large enough dataset was collected which is not trivial.

In addition to the potential biases, we highlight that we were limited to only one drone per class. It would be interesting to measure different drones of the same class to observe if we can identify each drone individually.

7.1.3 . Radar System

As we had access to one laboratory radar only, we could not assess more complex systems. We think that Deep Learning methods may be adapted to systems based on several measurement tools. For example, the signal measured by different systems could be gathered for the input of a single classifier network.

Such methods could combine different signal shapes to cover a larger variety of targets. It could even be used for different systems combining, for example, radar and acoustic or optical systems. Even if a radar system is the only one able to spot a target at a long distance, we imagine a combination of systems at different ranges to cover a broader scope of situations.

7.2 . Space Representation

We determined a micro-Doppler format giving the best recognition rates for the use conditions we studied. From this conclusion, we identify several research directions.

7.2.1 . Time Window

We observe that the WSP format was giving the best performance while the SG format was giving low results. However, the WSP format can be seen as an extreme case of spectrograms with the longest time window h, so, it may exist a better intermediate solution. We did not explore this idea to avoid over-fitting. Nonetheless, we think that for a sharper analysis, the hyperparameter $|h|$ corresponding to the length of the time window is an essential hyperparameter.

7.2.2 . Combination of Space Representations

We compared each space representation individually. However, a combination of the different spaces might lead to better results. Such an operation is possible but can be problematic for formats of different dimensions. In [30], the combination of two specific formats was analyzed. Profiles in different formats were simply concatenated without any transition. Despite this simplicity, the results look promising.

A deeper study with statistical tools might give better results for the combination of several formats.

7.2.3 . Deep Format

The radar formats studied were proposed before the rising of Deep Learning methods. They were originally designed to other analysis tools but now they are used successfully as classifier input. A format dedicated to Deep Learning tools might boost the classifier network.

7.3 . Unknown Targets - no Targets

We limited our study to known targets. In reality, it is impossible to have a measurement of all drones and birds. Thus, studies focusing on unknown targets (some of the targets only in the testing datasets) are an interesting research line. We suggest for such a use two potential Deep Learning approaches : Deep One Class classification methods [54] and Introspective Classification Networks [28].

7.4 . Data Augmentation Algorithm

The statistically significant gains observed thanks to GAN algorithms, gives us ideas for further works.

7.4.1 . GAN Stability

We observed the GAN instability. To balance the generator and discriminator networks, we opted for a modification of the training step of the generator for each discriminator training. However, other studies prefer to modify the learning rates or to use other tools such as the depth of the two networks. Such setups are based on key hyperparameters not often specified in the articles, sometimes only visible

in a GitHub project. Deeper studies to solve this balance issue might lead to higher performance.

7.4.2 . Stopping Criterion

We observed that the stopping iteration chosen to generate the synthetic data has a strong influence on the performance. A stopping criterion more appropriate than FID should be proposed.

We remarked that the balance of the loss values (G_d_f, D_d_f and D_d_r) is primordial but we were not able to determine an accurate criterion. To establish relevant correlations, evaluations should be made at numerous iterations of the GAN training, which requires a substantial computation power.

7.4.3 . Other Data Augmentation Methods

Except for some pix2pix trials, we worked with pure GANs methods for the data augmentation. Other methods such as the auto-encoder could be used. A comparison of the data augmentation performance of other methods with the evaluation we proposed is a research line that could redirect the research. Tools with a poorer performance in terms of realism might have, however, a higher potential in terms of the utility of the synthetic data.

7.5 . Conjugation of Real Data and Ground Truth

Even if we did not observe any improvement with the combination of real signals and drone logs, this research line preserves a certain potential as this complementary observation is pertinent according to the models. Deeper studies might lead to an interesting outcome. We listed potential explanations on our results in Section 5.5. We think that our work should help researches planned in this area.

8 - Annexe : Synthèse

Dans un contexte de protection de sites sensibles, la reconnaissance des drones commerciaux est l'un des enjeux majeurs actuels. Ces drones sont suffisamment performants pour effectuer des missions complexes potentiellement nuisibles, tout en étant assez petits pour être accessible à tous (maniabilité, prix) et échapper aux moyens de surveillances actuels. De nouveaux systèmes doivent donc être mis en place. Un axe de recherche privilégié pour ces systèmes se base sur le phénomène radar micro-Doppler.

Le phénomène Doppler est un décalage fréquentiel du signal réfléchi par une cible en mouvement. Accompagnant ce décalage, le phénomène micro-Doppler consiste en un ensemble de modulations créé par les mouvements internes de la cible, notamment les rotations des pales. Nous étudions le potentiel du micro-Doppler avec des outils Deep Learning pour résoudre notre problème de reconnaissance de drones.

Nous avons alors besoin de données. Une première piste s'oriente vers des modélisations électromagnétiques du micro-Doppler. En la développant, nous observons ses limites qui plaident à disposer de données réelles. Cependant, ces données sont rares et incomplètes. Nous réalisons donc une campagne de mesures.

Une fois les données collectées, nous nous retrouvons face à un nouvel obstacle : leur représentation. Il n'existe ainsi pas d'espace de représentation standard de données utilisé pour classifier des drones, ce qui compromet la comparaison des études. Nous observons alors l'impact de ces différents formats sur la classification de nos signaux avec des réseaux de neurones. À la suite des expériences conduites, nous proposons à la communauté un format standard possédant de très bonnes performances pour les différentes conditions d'utilisations étudiées : le format WSP (*Weighted Spectrum Format*).

Nous pouvons alors aborder un problème majeur en radar : le manque de mesures représentatives de la diversité des situations réelles. Notre objectif est de maintenir nos performances sur de petites bases de données. Nous examinons des algorithmes d'augmentation de données conçus à partir de GAN (*Generative Adversarial Network*). Nous proposons en parallèle une nouvelle mesure de la qualité de ces algorithmes basée sur l'utilité et non seulement sur le réalisme des données générées. Cette mesure est suffisamment générale pour être applicable dans de nombreux autres domaines de recherche. Nous développons une procédure de calibration des GANs afin d'identifier à moindre coût les hyper-paramètres clefs. Grâce à celle-ci nous mettons en place un GAN obtenant une augmentation de la classification des drones statistiquement significative grâce aux signaux qu'il génère.

Encouragés par ce résultat, nous implémentons alors des GANs plus avancés. Nous nous basons sur la conjugaison des signaux radar et des informations accessibles sur ces signaux (vérité terrain). Nos expériences nous permettent alors d'atteindre les performances précédentes. Actuellement, nous identifions des axes de résolutions, que nous prévoyons de développer, pour les dépasser.

Bibliographie

[1] I. Alnujaim et al. Generative adversarial networks for classification of micro-Doppler signatures of human activity. *Proc. IEEE, GRSL*, 2019.

[2] I. Alnujaim et al. Generative adversarial networks to augment micro-Doppler signatures for the classification of human activity. In *Proc. IEEE, IGARSS*, 2019.

[3] D. Barber et al. The IM algorithm : a variational approach to information maximization. In *Proc. NeurIPS*, 2003.

[4] Bessel function. Bessel function — Wikipedia, the free encyclopedia, 2018. [Online ; accessed 28-January-2019].

[5] B. P. Bogert et al. The quefrency alanysis of time series for echoes. *Time series analysis*, 1963.

[6] D. Brooks et al. A Hermitian positive definite neural network for micro-doppler complex covariance processing. 2019.

[7] D. A. Brooks et al. Temporal deep learning for drone micro-Doppler classification. In *Proc. IEEE, IRS*, 2018.

[8] R. Chen et al. Extracting radar micro-Doppler signatures of helicopter rotating rotor blades using k-band radars. In *Proc. SPIE*, 2014.

[9] V. C. Chen. Analysis of radar micro-Doppler with time-frequency transform. In *Proc. IEEE, SSAP*, 2000.

[10] V. C. Chen et al. Micro-Doppler effect in radar : phenomenon, model, and simulation study. *Proc. IEEE, TAES*, 2006.

[11] X. Chen et al. InfoGAN : Interpretable representation learning by information maximizing generative adversarial nets. In *Proc. NIPS*, 2016.

[12] B. Choi et al. Classification of drone type using deep convolutional neural networks based on micro-Doppler simulation. In *Proc. IEEE, ISAP*, 2018.

[13] L. Cifola et al. Target/clutter disentanglement using deep adversarial training on micro-Doppler signatures. In *Proc. IEEE, EuRAD*, 2019.

[14] D. C. Cireșan et al. Deep, big, simple neural nets for handwritten digit recognition. *Proc. Neural Computation*, 2010.

[15] H. G. Doherty et al. Unsupervised learning using generative adversarial networks on micro-Doppler spectrograms. In *Proc. EuRAD*, 2019.

[16] B. Erol et al. GAN-based synthetic radar micro-Doppler augmentations for improved human activity recognition. In *Proc. IEEE, RadarConf*, 2019.

[17] J. Farlik et al. Radar cross section and detection of small unmanned aerial vehicles. In *Proc. IEEE, ME*, 2016.

[18] C. Finn et al. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. ICML*, 2017.

[19] F. Fioranelli et al. Classification of loaded/unloaded micro-drones using multistatic radar. *Electronics Letters*, 2015.

[20] M. Frid-Adar et al. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, 2018.

[21] L. Fuhrmann et al. Micro-Doppler analysis and classification of UAVs at Ka band. In *Proc. IEEE, IRS*, 2017.

[22] I. Goodfellow et al. Generative adversarial nets. In *Proc. NIPS*, 2014.

[23] D. Hartman et al. *Quadcopter dynamic modeling and simulation (Quad-Sim v1.00)*. 2014.

[24] M. Heusel et al. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Proc. NIPS*, 2017.

[25] D. Huang et al. Micro-Doppler spectrogram denoising based on generative adversarial network. In *Proc. IEEE, EuMC*, 2018.

[26] K. Ishak et al. Human motion training data generation for radar based deep learning applications. In *Proc. IEEE, ICMIM*, 2018.

[27] P. Isola et al. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE, CVPR*, 2017.

[28] L. Jin et al. Introspective classification with convolutional nets. *Proc. NIPS*, 2017.

[29] A. Jolicoeur-Martineau. The relativistic discriminator : a key element missing from standard GAN. *arXiv preprint arXiv :1807.00734*, 2018.

[30] B. Kim et al. Drone classification using convolutional neural networks with merged Doppler images. *Proc. IEEE, GRSL*, 2017.

[31] Y. Kim et al. Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks. *Proc. IEEE, GRSL*, 2016.

[32] Y. Kim et al. Extraction of micro-Doppler characteristics of drones using high-resolution time-frequency transforms. *Proc. MOTL*, 2018.

[33] D. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv :1312.6114*, 2013.

[34] D. P. Kingma et al. Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.

[35] A. Krizhevsky et al. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012.

[36] Y. LeCun et al. Learning algorithms for classification : a comparison on handwritten digit recognition. *Neural Networks : the statistical mechanics perspective*, 1995.

[37] H. Li et al. Domain generalization with adversarial feature learning. In *Proc. IEEE, CVPR*, 2018.

[38] M. Lucic et al. Are GANs created equal ? A large-scale study. *arXiv :1711.10337*, 2017.

[39] J. Martin et al. Analysis of the theoretical radar return signal from aircraft propeller blades. In *Proc. IEEE, Radar Conference*, 1990.

[40] M. Mirza et al. Conditional generative adversarial nets. *arXiv :1411.1784*, 2014.

[41] P. Molchanov et al. Classification of small UAVs and birds by micro-Doppler signatures. *Proc. IJMWT*, 2014.

[42] M. A. Nielsen. *Neural networks and deep learning*, volume 25. 2015.

[43] A. Odena et al. Conditional image synthesis with auxiliary classifier GANs. In *Proc. ICML*, 2017.

[44] C. Olsson et al. Skill rating for generative models. *arXiv preprint arXiv :1808.04888*, 2018.

[45] J. S. Patel et al. Review of radar classification and RCS characterisation techniques for small UAVs ordrones. *Proc. IET, RSN*, 2018.

[46] J. S. Patel et al. Multi-time frequency analysis and classification of a micro-drone carrying payloads using multistatic radar. *The Journal of Engineering*, 2019.

[47] Melino R. et al. *Modelling Helicopter Radar Backscatter*. ADF, DST, Electronic Warfare and Radar Division, 2011.

[48] A. Radford et al. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv :1511.06434*, 2015.

[49] M. T. Ribeiro et al. "Why should I trust you ?" Explaining the predictions of any classifier. In *Proc. ACM, SIGKDD*, 2016.

[50] M. Ritchie et al. Monostatic and bistatic radar measurements of birds and micro-drone. In *Proc. IEEE, RadarConf*, 2016.

[51] O. Ronneberger et al. U-net : convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, 2015.

[52] D. Roselli et al. Managing bias in AI. In *Proc. WWWC*, 2019.

[53] J. W. Rudmin. Calculating the exact pooled variance. *arXiv preprint arXiv :1007.1012*, 2010.

[54] L. Ruff et al. Deep one-class classification. In *Proc. ICML*, 2018.

[55] T. Salimans et al. Improved techniques for training GANs. *arXiv preprint arXiv :1606.03498*, 2016.

[56] M. S. Seyfioğlu et al. Deep learning of micro-Doppler features for aided and unaided gait recognition. In *Proc. IEEE, RadarConf*, 2017.

[57] Y. Shao et al. Deep learning methods for personnel recognition based on micro-Doppler features. In *Proc. ICSPC*, 2017.

[58] A. Spurr et al. Guiding InfoGAN with semi-supervision. In *Proc. Joint ECML and MLKDD*, 2017.

[59] C. Szegedy et al. Going deeper with convolutions. In *Proc. IEEE, CVPR*, 2015.

[60] R. Tan et al. Improved micro-Doppler features extraction using smoothed-pseudo Wigner-Ville distribution. In *Proc. IEEE, TENCON*, 2016.

[61] R. P. Trommel et al. Multi-target human gait classification using deep convolutional neural networks on micro-Doppler spectrograms. In *Proc. IEEE, EuRAD*, 2016.

[62] D. Ulyanov et al. Instance normalization : The missing ingredient for fast stylization. *arXiv preprint arXiv :1607.08022*, 2016.

[63] X. Wang et al. Enhancing pix2pix for remote sensing image classification. In *Proc. IEEE, ICPR*, 2018.

[64] Z. Wang et al. Thermal to visible facial image translation using generative adversarial networks. *Proc. IEEE, SPL*, 2018.

[65] D. S. Williamson et al. Complex ratio masking for monaural speech separation. *Proc. IEEE, TASLP*, 2016.

[66] S. C. Wong et al. Understanding data augmentation for classification : when to warp ? In *Proc. IEEE, DICTA*, 2016.

[67] P. Xiang et al. Single-image de-raining with feature-supervised generative adversarial network. *Proc. IEEE, SPL*, 2019.

[68] K. Zhou et al. Deep domain-adversarial image generation for domain generalisation. In *Proc. AAAI*, 2020.

[69] J.-Y. Zhu et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. IEEE, ICCV*, 2017.

Acknowledgments

First and foremost, I would like to express gratitude to my supervisors Prof. Joanna Tomasik, Dr. Arpad Rimmel, Christèle Morisseau and Gilles Vieillard. We managed to efficiently work together during this thesis despite the numerous technical difficulties we have been confronted especially with the measurement campaign and the covid situation. Besides these difficulties we were able to use the diversities of our knowledge and skills to handle this PhD. I have learned a lot during the many discussions and debates we had.

I would like to thank Tristan Cazenave and Laurent Ferro-Famil for their very insightful comments in the reviews of this thesis. I also express my gratitude to the other members of the jury for attending my Ph.D. defense : Alexandre Benoit, Devan Sohier, Mihai Datcu and Olivier Schwander. The discussions we had during the defense thesis were deep and interesting.

I am grateful to Paul Legrand for the work we made during its internship about GANs. I am also grateful for the help of Julien Léal during the begining of my thesis on the electromagnetic simulations. I also want to thanks every people who took part of the measurement campaign.

Eventually, I would like to thank my family, my friends and my colleagues both from the LRI and the ONERA. I will not forget the great support I received, especially from my mother Sylvia, my father Didier.