



# Detecció visual de bucles en fons marins mitjançant xarxes neuronals

**Jordi Gálvez Santos**

Màster Universitari d'Enginyeria Informàtica  
Intel·ligència Artificial

Consultor: **Antonio Burguera Burguera**

Professor responsable de l'assignatura: **Carles Ventura Royo**

Data Lliurament: 02/06/2020



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>Detecció visual de bucles en fons marins mitjançant xarxes neuronals</i>
<b>Nom de l'autor:</b>	<i>Jordi Gálvez Santos</i>
<b>Nom del consultor/a:</b>	<i>Antonio Burguera Burguera</i>
<b>Nom del PRA:</b>	<i>Carles Ventura Royo</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>06/2020</i>
<b>Titulació o programa:</b>	<i>Màster universitari d'Enginyeria Informàtica</i>
<b>Àrea del Treball Final:</b>	<i>Intel·ligència Artificial</i>
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	<i>SLAM, Computer vision, Deep Learning</i>
<b>Resum del Treball</b>	
<p>Un vehicle autònom disposat en un entorn desconegut s'ha de poder localitzar i desplaçar sense supervisió. SLAM pretén resoldre aquest problema. Un dels punts importants de SLAM és identificar si el robot ha passat abans per un punt donat. Això rep el nom de detecció de bucles.</p> <p>Existeixen diferents mecanismes per resoldre la detecció del bucle. Una de les variants és emprar tècniques d'aprenentatge computacional (<i>machine learning</i>) per comparar imatges.</p> <p>El problema que pretén resoldre aquest treball és crear una Xarxa Neuronal per a la detecció de bucles en fons marins. La detecció de bucles es du a terme sobre imatges del fons marí obtingudes per robots.</p> <p>Les imatges d'entrada de la xarxa es modificaran creant una imatge sintètica fruit d'aplicar transformacions a la imatge original, originant un parell d'imatges relacionades. Es calcularà un descriptor global (HOG) d'una de les imatges implicades en el bucle i s'entrenarà la xarxa per tal que produeixi un descriptor semblant a partir de l'altra imatge implicada en el bucle. D'aquesta forma la xarxa aprendrà a generar descriptors similars davant de parelles d'imatges implicades en bucles.</p> <p>La resolució del problema s'ha implementat amb les llibreries keras en Python. S'ha construït una xarxa neuronal que s'ha avaluat i afinat per oferir millors resultats.</p> <p>El resultat ha estat una Xarxa Neuronal viable per al problema proposat. Com a treballs futurs cal destacar comparar el sistema creat amb altres sistemes.</p>	

## **Abstract**

An autonomous vehicle placed in a not known environment must be able to be located and moved without supervision. SLAM wants to resolve this problem. Identify when a robot has previously passed through a given point is called loop closure and is an important step on SLAM methodology.

There are various mechanisms for resolve loop closure problem. Compare images with machine learning is one of these techniques.

The purpose of this work is to create a Neural Network to detect loop closures on seabed. The loop closure detection is performed by images of seabed taken by robots.

The input images in Neural Network will be transformed to create a synthetic image, generating a pair of images (the original and the altered). A global descriptor (HOG) will be calculated on one of these loop images and the Network will be trained with the other image to find a similar descriptor. With this training, the Neural Network will learn to generate similar descriptors from pairs of images involved in loop closures

On the resolution of this problem we will use Keras on Python. We built a Neural Network that has been evaluated and then refined to provide better results.

The result of this work has been a viable Neural Network for the proposed problem. As a future work will be compare this Neural Network with other loop closure techniques.

# Índex

1. Introducció.....	1
1.1. Estructura de la memòria.....	1
2. Planificació .....	2
2.1. Gestió del risc.....	4
3. Localització en robòtica mòbil .....	8
3.1. Definint el problema.....	8
3.2. Solucionant el problema .....	9
3.3. Dibuixant el mon .....	10
4. Xarxes Neuronal.....	12
4.1. Neurona.....	12
4.2. Capes de neurones .....	14
4.3. Xarxes Neuronal Convolucionals .....	15
4.4. Capes d'agrupació.....	19
5. Disseny de la xarxa neuronal .....	20
5.1. Algoritme de tancament de bucle per N. Merril i G. Huang .....	20
5.2. Detecció de bucles en fons marins .....	24
5.3. Arquitectura del model.....	25
6. Implementació de la xarxa neuronal.....	28
6.1. Diagrama de classes .....	28
6.2. Dades d'entrada .....	28
6.3. Model.....	31
6.4. Xarxa Neuronal.....	32
6.5. Generador de dades.....	32
6.6. Avaluació de la xarxa.....	33
6.7. Tractament de dades.....	33
7. Resultats experimentals .....	35
7.1. Introducció .....	35
7.2. Criteris per l'avaluació de la xarxa neuronal.....	35
7.3. Cerca d'hiperparàmetres .....	37
7.4. Resum dels escenaris.....	54
7.5. Avaluació qualitativa .....	56
8. Conclusions.....	59
8.1. Planificació final .....	59
8.2. Treballs futurs.....	62
Glossari .....	63
Annex: Eines i llibreries emprades .....	64
Bibliografia.....	65

## Llista d'il·lustracions

IL·LUSTRACIÓ 1 DIAGRAMA DE GANTT DEL PROJECTE SENCER.....	4
IL·LUSTRACIÓ 2 DIAGRAMA DE GANTT DE LES TASQUES T02.....	4
IL·LUSTRACIÓ 3 DIAGRAMA DE GANTT DE LES TASQUES T03.....	4
IL·LUSTRACIÓ 4 DIAGRAMA EXEMPLE DEL DESPLAÇAMENT D'UN ROBOT I ELS PUNTS DE REFERÈNCIA, FONT [1].	9
IL·LUSTRACIÓ 5 IDENTIFICACIÓ D'UN BUCLE EN EL DESPLAÇAMENT D'UN ROBOT.....	10
IL·LUSTRACIÓ 6 ESTRUCTURA BÀSICA D'UNA NEURONA BASADA EN LA DOCUMENTACIÓ DE [8].	12
IL·LUSTRACIÓ 7 GRÀFICA DE LA FUNCIÓ SIGMOIDE.....	13
IL·LUSTRACIÓ 8 GRÀFICA DE LA FUNCIÓ RELU.....	13
IL·LUSTRACIÓ 9 CLASSIFICACIÓ AMB UN ÚNIC PERCEPTRÓ.....	14
IL·LUSTRACIÓ 10 EXEMPLE D'UNA CAPA AMB DEU NEURONES.....	14
IL·LUSTRACIÓ 11 EXEMPLE D'UNA XARXA NEURONAL AMB CAPES OCULTES.....	15
IL·LUSTRACIÓ 12 EXEMPLE D'UNA FINESTRA MÒBIL. IMATGE ORIGINAL DE PXHERE.....	16
IL·LUSTRACIÓ 13 REPRESENTACIÓ D'UNA IMATGE EN VALORS BINARIS.....	16
IL·LUSTRACIÓ 14 REPRESENTACIÓ D'UN FILTRE D'UNA PILOTA.....	16
IL·LUSTRACIÓ 15 EXEMPLE DE CÀLCUL D'UNA CONVOLUCIÓ.....	17
IL·LUSTRACIÓ 16 EXEMPLES D'UNA FINESTRA MÒBIL AMB DESPLAÇAMENT 2 I 3.....	17
IL·LUSTRACIÓ 17 EXEMPLE D'AFEGIR FARCIMENT A UNA MATRIU.....	18
IL·LUSTRACIÓ 18 EXEMPLE D'APLICACIÓ D'UNA CAPA D'AGRUPACIÓ.....	19
IL·LUSTRACIÓ 19 RESULTAT D'APLICAR UNA CAPA D'AGRUPACIÓ, [14].	19
IL·LUSTRACIÓ 20 TRACTAMENT DE DADES PER A L'ENTRENAMENT DE L'ALGORITME [7]. IMATGE CREDTA AMB RECURSOS DE [27], [30] I <a href="https://fontawesome.com">HTTPS://FONTAWESOME.COM</a> .....	21
IL·LUSTRACIÓ 21 EXEMPLE DE VISIÓ D'UN ROBOT EN MOVIMENT. RECURSOS USATS DE [28].	21
IL·LUSTRACIÓ 22 EXEMPLE DE MOVIMENT DE ROBOT GIRANT. RECURSOS USATS DE [28].	22
IL·LUSTRACIÓ 23 GENERACIÓ D'UNA IMATGE SINTÈTICA A PARTIR D'UNA IMATGE REAL. RECURSOS USATS DE [31].	22
IL·LUSTRACIÓ 24 VECTOR HOG D'UNA IMATGE. RECURS [31].	22
IL·LUSTRACIÓ 25 HISTOGRAMA DE LA IL·LUSTRACIÓ 24. GRÀFICA OBTINGUDA AMB [27].	23
IL·LUSTRACIÓ 26 REPRESENTACIÓ DEL PROCÉS DE TRACTAMENT DE LES DADES D'ENTRADA. RECURSOS DE [31].	23
IL·LUSTRACIÓ 27 VISTA LATERAL D'UN ROBOT AQUÀTIC I TERRESTRE, RECURSOS DE GRABCAD COMMUNITY.....	24
IL·LUSTRACIÓ 28 BUCLE EN UN FONTS MARÍ.....	24
IL·LUSTRACIÓ 29 EXEMPLE DE DUES IMATGES QUE NO FORMEN UN BUCLE.....	25
IL·LUSTRACIÓ 30 ARQUITECTURA DE LA XARXA NEURONAL.....	27
IL·LUSTRACIÓ 31 DIAGRAMA DE CLASSES.....	28
IL·LUSTRACIÓ 32 ESTRUCTURA DELS DIRECTORIS I FITXERS DEL CONJUNT DE DADES.....	29
IL·LUSTRACIÓ 33 EXEMPLE DE L'ESTRUCTURA DELS FITXERS D'INFORMACIÓ DEL CONJUNT DE DADES.....	30
IL·LUSTRACIÓ 34 IMATGE ORIGINAL (ESQUERRA) I TRANSFORMACIÓ APLICADA (DRETA). HA SOFERT TRANSLACIÓ I ROTACIÓ.....	34
IL·LUSTRACIÓ 35 IMATGE ORIGINAL (ESQUERRA) I TRANSFORMACIÓ APLICADA (DRETA). HA SOFERT TRANSLACIÓ, ROTACIÓ I CANVI DEL NIVELL GAMMA.....	34
IL·LUSTRACIÓ 36 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 0 AMB 10 ÈPOQUES.....	37
IL·LUSTRACIÓ 37 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 0 AMB 30 ÈPOQUES.....	37
IL·LUSTRACIÓ 38 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 0 AMB 50 ÈPOQUES.....	38
IL·LUSTRACIÓ 39 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 0 AMB 200 ÈPOQUES.....	38
IL·LUSTRACIÓ 40 PERCENTATGE D'ENCERT RESPECTE LA MIDA DEL CONJUNT D'IMATGES DE L'ESCENARI 0, AMB 30 ÈPOQUES.....	38
IL·LUSTRACIÓ 41 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 1.....	39
IL·LUSTRACIÓ 42 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 2 AMB PÈRDUA MSLE.....	40
IL·LUSTRACIÓ 43 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 2 AMB PÈRDUA MAE.....	40
IL·LUSTRACIÓ 44 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 2 AMB PÈRDUA MSE.....	41
IL·LUSTRACIÓ 45 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 2 AMB PÈRDUA BCE.....	41
IL·LUSTRACIÓ 46 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 2 AMB PÈRDUA KLD.....	41
IL·LUSTRACIÓ 47 REPRESENTACIÓ DELS VALORS TPF I FPR DE L'ESCENARI 2 I BASE.....	42
IL·LUSTRACIÓ 48 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 3 AMB OPTIMITZADOR SGD.....	43
IL·LUSTRACIÓ 49 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 3 AMB OPTIMITZADOR ADAGRAD.....	43
IL·LUSTRACIÓ 50 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 3 AMB OPTIMITZADOR ADADELTA.....	43
IL·LUSTRACIÓ 51 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 3 AMB OPTIMITZADOR ADAM.....	44
IL·LUSTRACIÓ 52 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 3 AMB OPTIMITZADOR NADAM.....	44
IL·LUSTRACIÓ 53 REPRESENTACIÓ DELS VALORS TPF I FPR DE L'ESCENARI 3 I BASE.....	45
IL·LUSTRACIÓ 54 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 4 AMB LRN.....	45
IL·LUSTRACIÓ 55 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 5 AMB TANH.....	46
IL·LUSTRACIÓ 56 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 5 AMB SIGMOIDE.....	46
IL·LUSTRACIÓ 57 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 5 AMB LEAKYRELU.....	47
IL·LUSTRACIÓ 58 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 5 AMB SWISH.....	47
IL·LUSTRACIÓ 59 REPRESENTACIÓ DELS VALORS TPF I FPR DE L'ESCENARI 5 I BASE.....	48
IL·LUSTRACIÓ 60 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 6 AMB SOFTMAX.....	48
IL·LUSTRACIÓ 61 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 6 AMB TANH.....	49

IL·LUSTRACIÓ 62 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 6 AMB LINEAL.....	49
IL·LUSTRACIÓ 63 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 9 PEL CAS BIAS_ZEROS.....	53
IL·LUSTRACIÓ 64 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 9 PEL CAS KGLOROT_BZEROS.....	53
IL·LUSTRACIÓ 65 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 9 PEL CAS KGLOROT_BNORMAL.....	53
IL·LUSTRACIÓ 66 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 9 PEL CAS HE_GLOROT.....	54
IL·LUSTRACIÓ 67 EVOLUCIÓ DE L'ENTRENAMENT DE L'ESCENARI 9 PEL CAS HE_GLOROT_BZEROS.....	54
IL·LUSTRACIÓ 68 IMATGE QUERY.....	56
IL·LUSTRACIÓ 69 IMATGE QUE LA XARXA NEURONAL DETECTA COM A BUCLE.....	56
IL·LUSTRACIÓ 70 COMPOSICIÓ DE LA IMATGE QUERY AMB LA DETECTADA QUE FORMA UN BUCLE.....	57
IL·LUSTRACIÓ 71 IMATGE DETECTADA COM A BUCLE PER LA XARXA PERÒ REALMENT NO HO ÉS.....	57
IL·LUSTRACIÓ 72 COMPOSICIÓ DEL BUCLE NO DETECTAT PER LA XARXA NEURONAL.....	58
IL·LUSTRACIÓ 74 QUERY NO DETECTADA PER LA XARXA.....	58
IL·LUSTRACIÓ 73 IMATGE QUE FORMA UN BUCLE REAL NO DETECTAT PER LA XARXA.....	58
IL·LUSTRACIÓ 75 DIAGRAMA DE GANTT A FINAL DE PROJECTE.....	61

## Llista de taules

TAULA 1 NIVELL DEL RISC.....	5
TAULA 2 TAULA DE RISCOS.....	6
TAULA 3 MESURES CORRECTORES.....	7
TAULA 4 CONJUNT DE CAPES DE LA CODIFICACIÓ.....	26
TAULA 5 CONFIGURACIÓ DE LES CAPES DE DESCODIFICACIÓ.....	26
TAULA 6 MATRIU DE CONFUSIÓ DE L'ESCENARI BASE.....	39
TAULA 7 AVALUACIÓ DE L'ESCENARI 0.....	39
TAULA 8 RESULTATS DE L'AVUACIÓ DE L'ESCENARI 1 I COMPARACIÓ AMB L'ESCENARI BASE.....	40
TAULA 9 COMPARACIÓ DELS RESULTATS DE L'ESCENARI 2 I DE L'ESCENARI BASE.....	42
TAULA 10 COMPARACIÓ DELS RESULTATS DE L'ESCENARI 3 I DE L'ESCENARI BASE.....	44
TAULA 11 COMPARACIÓ DELS RESULTATS DE L'ESCENARI 4 I DE L'ESCENARI BASE (AMB OPTIMITZADOR SGD).....	46
TAULA 12 COMPARACIÓ DELS RESULTATS DE L'ESCENARI 5 I DE L'ESCENARI BASE (AMB RELU).....	47
TAULA 13 COMPARACIÓ DELS RESULTATS DE L'ESCENARI 6 I DE L'ESCENARI BASE (AMB SIGMOIDE).....	49
TAULA 14 RESULTATS DE L'ESCENARI BASE.....	50
TAULA 15 COMPARACIÓ DELS RESULTATS DE L'ESCENARI 7, PART I.....	50
TAULA 16 COMPARACIÓ DELS RESULTATS DE L'ESCENARI 7, PART II.....	51
TAULA 17 COMPARACIÓ DELS RESULTATS DE L'ESCENARI 8, PART I.....	51
TAULA 18 COMPARACIÓ DELS RESULTATS DE L'ESCENARI 8, PART II.....	52
TAULA 19 COMPARACIÓ DELS RESULTATS DE L'ESCENARI 8, PART III.....	52
TAULA 20 COMPARACIÓ DELS RESULTATS DE L'ESCENARI 9 I DE L'ESCENARI BASE (NORMAL).....	54
TAULA 21 COMPARACIÓ DE L'OPTIMITZADOR SGD I ADAGRAD.....	55





# 1. Introducció

A dia d'avui la conducció autònoma és un tema rellevant on múltiples empreses, grups científics o ens públics inverteixen importants recursos per intentar superar les barreres actuals i ser els primers en tenir sistemes fiables. A les cases es poden trobar robots netejadors que recorren i escanegen les llars per estalviar feina als humans. Fins i tot hi ha substituïts metàl·lics de gossos d'atura, o una aproximació a aquests.

La base d'aquests sistemes és semblant: un robot ha de prendre decisions per desplaçar-se. A partir d'aquest punt de partida hi ha múltiples camins. Les tècniques de *Simultaneous Localization and Mapping* o SLAM, que es detallen al capítol 3, pretenen ajudar a resoldre una part d'aquesta problemàtica.

L'objectiu principal d'aquest projecte és, amb l'ús d'una xarxa neuronal, identificar quan el vehicle ha passat per un lloc en el que ja havia estat prèviament. Aquest procés, que rep el nom de detecció de bucle, és una part important en el procés que defineix SLAM. En concret es treballarà sobre les dades proporcionades per un robot submarí mentre analitzava visualment el fons marí.

Per aconseguir aquest objectiu, es pretén analitzar l'estat de l'art actual en la identificació d'imatges emprant xarxes neuronals. Es dissenyarà i implementarà una xarxa neuronal i es validarà el seu funcionament amb un conjunt de dades. Tot seguit s'afinarà per millorar la detecció.

Els productes generats en aquest projecte seran:

- Una xarxa neuronal per identificar bucles. Aquesta xarxa neuronal estarà formada per diversos arxius de codi font.
- Aquest document amb la memòria del projecte.

## 1.1. Estructura de la memòria

Aquest treball està dividit en diferents capítols. Els primers, 3 i 4, introdueixen el concepte de localització d'un robot i fan una introducció al món de les xarxes neuronals.

El capítol 5 es dissenya la Xarxa Neuronal. Primer, a la secció 5.1, presenta un treball on detallen una proposta de resolució al problema de la detecció de tancament bucles. A continuació, en la secció 5.2 analitza l'especificitat del problema presentat al ser el fons marí l'entorn d'estudi. Per acabar, a l'apartat 5.3, defineix l'arquitectura del model.

Al capítol 6 es documenta la implementació de la Xarxa Neuronal.

Finalment al capítol 7 s'avalua i afina la xarxa neuronal per proporcionar millors resultats.

## 2. Planificació

Per a l'estimació d'aquet treball es considerarà la disponibilitat aproximada de 15 hores setmanals, tenint en compte que els principals dies de feina seran caps de setmana i festius. Aquesta disponibilitat disminuirà a partir de febrer del 2020 al haver de compartir hores amb el desenvolupament d'una altra assignatura del màster. La planificació s'ha desglossat en cinc etapes que es detallen a continuació.

A les conclusions, secció 8.1, s'analitzarà l'evolució d'aquesta planificació.

De forma general, el projecte s'ha estimat segons la següent taula:

### T00: Projecte sencer

Data inici:	<b>18/09/2019</b>	Data Fi:	<b>05/05/20</b>	Hores estimades:	<b>351h</b>
Descripció:	Elaboració completa del projecte de TFM.				

El projecte es pot dividir en cinc gran grups de tasques. A més d'aquestes tasques s'ha de tenir en compte el treball relacionat amb les entregues de les PACs de l'assignatura que no es troben indicades en aquestes tasques.

### T01: Selecció TFM

Data inici:	<b>18/09/2019</b>	Data Fi:	<b>30/09/2019</b>	Hores estimades:	<b>30h</b>
Descripció:	En aquesta tasca s'avaluen les diferents propostes de TFM i s'acorda amb el director el contingut del mateix.				

### T02: Adquirint coneixement

Data inici:	<b>01/10/19</b>	Data Fi:	<b>29/10/19</b>	Hores estimades:	<b>62h</b>
Descripció:	Aquesta tasca permet adquirir coneixements tècnics per resoldre la problemàtica que planteja el TFM. Aquest coneixement inicial s'anirà ampliant durant el desenvolupament.				

### T03: Resolució del problema

Data inici:	<b>30/10/19</b>	Data Fi:	<b>24/03/20</b>	Hores estimades:	<b>189h</b>
Descripció:	Aquesta és la tasca principal del TFM. Inclou, entre altres, la creació del sistema per resoldre el problema plantejat pel TFM.				

### T04: Comparació amb altres sistemes

Data inici:	<b>25/03/20</b>	Data Fi:	<b>14/04/19</b>	Hores estimades:	<b>30h</b>
Descripció:	El sistema creat s'ha de comparar amb altres sistemes i determinar possibles punts de millora.				

### T05: Documentació

Data inici:	<b>01/10/19</b>	Data Fi:	<b>05/05/20</b>	Hores estimades:	<b>40h</b>
Descripció:	Aquesta serà una tasca de fons durant tot el projecte. Hi haurà un procés final d'adequació de la documentació, que tindrà dos lliurables: una memòria i una presentació.				

Degut a la complexitat de les tasques T02 Adquirint coneixement i T03 Resolució del problema aquestes es subdivideixen en les següents tasques:

### T02-01: SLAM

Data inici:	<b>01/10/19</b>	Data Fi:	<b>29/10/19</b>	Hores estimades:	<b>15h</b>
Descripció:	Permet incorporar coneixements sobre la gestió de moviments en robots. Concretament les tècniques SLAM i V-SLAM.				

### T02-02: IA

Data inici:	<b>01/10/19</b>	Data Fi:	<b>29/10/19</b>	Hores estimades:	<b>40h</b>
Descripció:	Aprofundir en el coneixement de sistemes avançats de IA, com poden ser les xarxes neuronals o <i>Deep Learning</i> .				

### T02-03: Estructura de dades

Data inici:	<b>01/10/19</b>	Data Fi:	<b>14/10/19</b>	Hores estimades:	<b>7h</b>
Descripció:	Conèixer les estructures de dades pròpies per al treball.				

### T03-01: Anàlisi del problema

Data inici:	<b>30/10/19</b>	Data Fi:	<b>05/11/19</b>	Hores estimades:	<b>13h</b>
Descripció:	Anàlisi de la problemàtica a resoldre i estudi inicials per determinar els camins següents.				

### T03-02: Disseny

Data inici:	<b>06/11/19</b>	Data Fi:	<b>19/11/19</b>	Hores estimades:	<b>26h</b>
Descripció:	Disseny de la solució que s'implementarà. S'ha de tenir en compte la informació del anàlisi. S'hauria de definir possibles alternatives.				

### T03-03: Implementació

Data inici:	<b>20/11/19</b>	Data Fi:	<b>18/02/20</b>	Hores estimades:	<b>105h</b>
Descripció:	Implementació del disseny.				

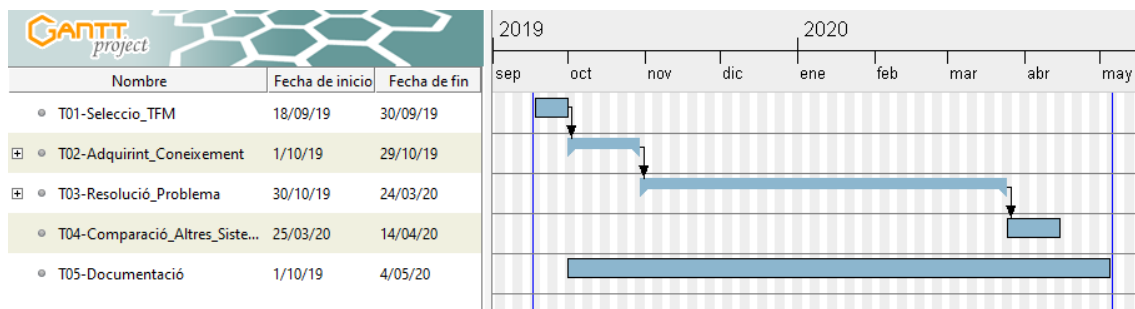
### T03-04: Disseny d'experiments

Data inici:	<b>19/02/20</b>	Data Fi:	<b>03/03/20</b>	Hores estimades:	<b>20h</b>
Descripció:	Establir diferents experiments per poder avaluar la implementació.				

### T03-05: Experimentació

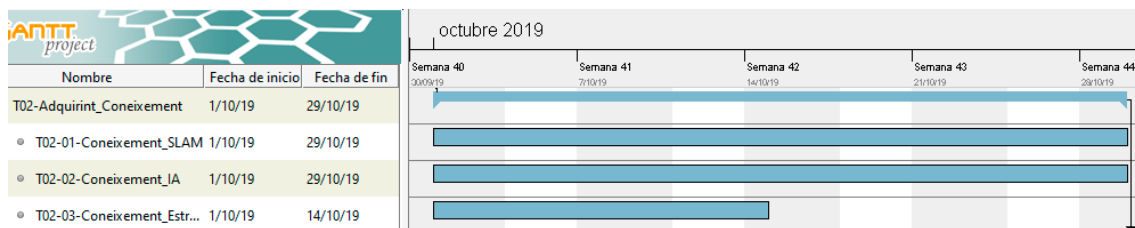
Data inici:	<b>04/03/20</b>	Data Fi:	<b>24/03/20</b>	Hores estimades:	<b>25h</b>
Descripció:	Realitzar proves per avaluar el rendiment de la solució.				

Aquestes tasques es poden representar en els diagrames de Gantt de les II·lustració 1, II·lustració 2 i II·lustració 3.



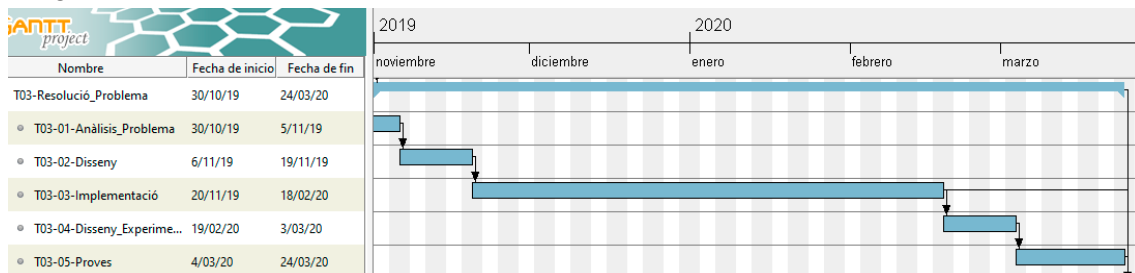
II·lustració 1 Diagrama de Gantt del projecte sencer.

Diagrama de Gantt de les tasques T02:



II·lustració 2 Diagrama de Gantt de les tasques T02.

Diagrama de Gantt de les tasques T03:



II·lustració 3 Diagrama de Gantt de les tasques T03.

## 2.1. Gestió del risc

Els principals riscos detectats a l'inici del projecte són:

- Que el projecte no estigui enllestit a la data de finalització esperada.
- No comprensió de la problemàtica a resoldre.
- No comprensió de les eines o tècniques a emprar en el projecte.
- No obtenir cap xarxa neuronal viable.
- Que les xarxes neuronals implementades no millorin la resolució del problema.
- Possibles problemes en la experimentació amb altres sistemes.

Per identificar els riscos del projecte es fa us de la metodologia NIST 800-30<sup>1</sup>. En aquesta metodologia, l'impacte pot ser classificat en:

- Baix. Afecta poc al negoci.

<sup>1</sup> <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf>

- Mitjà. Afecta de manera important al negoci, però pot continuar-lo.
- Alt. Afecta greument al negoci.

Els criteris per a l'avaluació de la probabilitat que s'estableixen en aquest projecte són:

- Baixa. És poc probable que es produeixi.
- Mitjana. És possible que es produeixin aquest risc.
- Alta. Pot produir-se de forma habitual.

Es pot definir el nivell de risc segons la combinació del seu impacte i probabilitat, tenint com a referència la matriu de la Taula 1. A la Taula 2 es poden trobar els riscos associats al projecte.

*Taula 1 Nivell del risc.*

		Impacte		
		Baix	Mitjà	Alt
Probabilitat	Alta	Baix	Mitjà	Alt
	Mitjana	Baix	Mitjà	Mitjà
	Baixa	Baix	Baix	Baix

Taula 2 Taula de riscos.

<b>Codi</b>	<b>Nom</b>	<b>Causa</b>	<b>Descripció</b>	<b>Conseqüència</b>	<b>Probabilitat</b>	<b>Impacte</b>	<b>Nivell</b>
<b>R01</b>	No finalitzat a temps	Mala planificació, problemes externs, ...	Donat el poc temps disponible o la poca experiència en la matèria pot provocar que es necessiti més temps	No poder entregar el projecte en la data estimada	Mitjana	Alt	Mitjà
<b>R02</b>	No comprensió de la problemàtica	Poca experiència en la matèria	Al ser el desenvolupador un neòfit en la matèria pot ocasionar una sèrie de dubtes	Prendre decisions incorrectes o allargar el desenvolupament	Mitjana	Mitjà	Mitjà
<b>R03</b>	No comprensió d'eines o tècniques	Falta de coneixement	Si el desenvolupador té poca experiència pot tenir punts on no pugui avançar	Prendre decisions incorrectes o allargar el desenvolupament	Mitjana	Mitjà	Mitjà
<b>R04</b>	No obtenció xarxa neuronal vàlida	Errors de càlculs o decisions incorrectes	Les xarxes neuronals obtingudes no serveixen per resoldre la problemàtica establerta	No poder entregar el projecte en les dates establertes o canviar la funcionalitat final	Mitjana	Alt	Mitjà
<b>R05</b>	No millorar resultats existents	Errors de càlculs	La xarxa neuronal obtinguda no millora altres tècniques o xarxes	El projecte s'entrega, però no s'assoleixen els objectius desitjats	Alta	Baix	Baix
<b>R06</b>	Problemes en la experimentació	Falta de temps o dificultats en la preparació de les proves	A l'hora de comparar amb altres sistemes, no es pugui realitzar per falta de temps o dificultats en la implementació d'aquests sistemes	Que no es pugui comparar la xarxa neural amb altres sistemes	Mitjana	Mitjà	Mitjà

A la Taula 3 es troben les possibles mesures correctores acordades pels riscos identificats.

*Taula 3 Mesures correctores.*

<b>Codi Acció</b>	<b>Codi Risc</b>	<b>Acció</b>	<b>Tipus</b>	<b>Risc residual</b>
<b>A1R01</b>	R01	Acordar amb director del TFM i amb el tutor la continuació del projecte en el següent curs	Mitigadora	Mitjà
<b>A2R01</b>	R01	Acordar amb el tutor un canvi de TFM	Mitigadora	Mitjà
<b>A1R02</b>	R02	Buscar assessorament del director del TFM	Correctora	Baix
<b>A1R03</b>	R03	Buscar assessorament del director del TFM	Correctora	Baix
<b>A1R04</b>	R04	Al detectar desviacions a l'esperat, consultar al director del TFM	Correctora	Baix
<b>A1R05</b>	R05	Anotar els resultats obtinguts i detallar la investigació comesa	Mitigadora	Baix
<b>A1R06</b>	R06	Comparar el resultat amb dades ja publicades	Mitigadora	Baix

### 3. Localització en robòtica mòbil

Per obtenir una xarxa neuronal per a la detecció de bucles en fons marins, abans s'ha de comprendre què significa un bucle en el procés de localització d'un vehicle autònom. Aquest capítol explica com un robot mòbil pot determinar la seva posició en un entorn desconegut. Aquesta tasca s'anomena localització. Es definirà què és *Simultaneous Localization and Mapping* o *SLAM* i un component bàsic d'aquest, la detecció de bucles. Per acabar es descriurà breument una possible solució per a la detecció de bucles.

En l'àmbit de la robòtica, la localització i construcció de mapes simultanis o *SLAM* [1] intenta aconseguir que un robot mòbil ubicat en una posició desconeguda en un entorn no conegut construeixi un mapa consistent del seu entorn mentre, simultàniament, determina la seva ubicació en aquest mapa. Aquest algoritme permet, per exemple, que un robot pugui realitzar un recorregut i tornar al punt de tornada de forma autònoma.

Els primers treballs que plantegen aquest problema són de 1986, en la *IEEE Robotics and Automation Conference* i posteriorment en un treball de *Smith, R., Self, M. i Cheeseman, P.* [2]. A dia d'avui aquest problema s'ha formulat i solucionat teòricament i s'han implementat múltiples solucions en diversos entorns: interiors, exteriors, subaquàtics, aeris, etcètera.

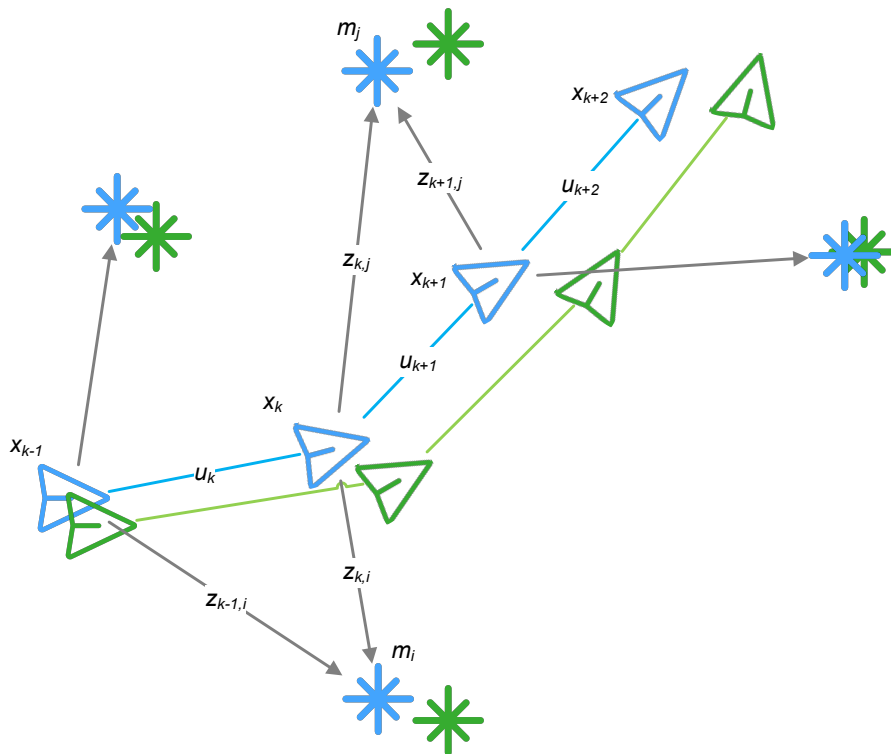
#### 3.1. Definint el problema

Simplificant el problema plantejat, es pot tenir com a base un robot que, donada una posició fixa en un moment de temps  $k-1$  donat, fa una observació de l'entorn que el rodeja amb els seus sensors. D'aquesta mesura extreu una estimació de la posició d'una sèrie de punts de referència i la relació entre aquests i la posició del robot. El robot es mou a una nova posició, en un nou moment  $k$  de temps, on pren una altra mesura. Llavors analitza les relacions entre els punts de referència anteriors, els nous punts de referència i la nova posició del robot. La informació que emmagatzema [1] és:

- $x_k$ : informació de la posició i orientació del vehicle.
- $u_k$ : el vector de moviment que col·loca al robot en la nova posició en el moment  $k$ . A un nivell purament teòric,  $u_k$  representa la consigna de moviment proporcionada a priori al robot. Ara bé, la majoria d'implementacions ho substitueixen per observacions a posteriori del moviment anomenades *dead reckoning* o, molt sovint, odometria.
- $m_i$ : un vector per descriure la localització del  $i$ -èsim punt de referència.
- $z_{ik}$ : observació feta del robot a un punt  $i$ -èsim de referència en un moment  $k$ .
- $X_{0:k} = \{x_0, x_1, \dots, x_k\} = \{X_{0:k-1}, x_k\}$ : la història de posicions del robot.
- $U_{0:k} = \{u_0, u_1, \dots, u_k\} = \{U_{0:k-1}, u_k\}$ : la història de moviments del robot.
- $Z_{0:k} = \{z_0, z_1, \dots, z_k\} = \{Z_{0:k-1}, z_k\}$ : el conjunt de tots els punts de referència.

És important destacar que les dades recollides són una estimació i poden diferir dels valors reals, tal com es mostra a la Il·lustració 4. En aquesta imatge es pot observar el desplaçament real (en verd) i estimat (en blau) d'un robot (triangle) i els punts de referència identificats (estels). Tota aquesta informació s'ha de emmagatzemar i processar iterativament. Aquest procés té un elevat cost computacional.





Il·lustració 4 Diagrama exemple del desplaçament d'un robot i els punts de referència, font [1].

En un model probabilístic [1] es pot definir el model d'observació com aquell que descriu la probabilitat de prendre una observació  $z_k$  quan les posicions del vehicle i del punt de referència són conegudes,  $P(z_k|x_k, m)$ . Podem definir com model de moviment per al robot una distribució de probabilitat en les transicions d'estats  $P(x_k|x_{k-1}, u_k)$ . Això permet que l'estat  $x_k$  depengui del precedent  $x_{k-1}$ .

A mesura que el robot es va desplaçant, aquest pot observar en diferents posicions el mateix punt de referència  $m_j$ . L'estimació d'aquest s'actualitzarà en tots els valors previs on s'hagi visualitzat  $m_j$ . Per exemple, a la Il·lustració 4 es pot observar que en la posició  $x_k$  el robot identifica  $m_j$  i que a la posició  $x_{k+1}$  el robot identifica el mateix punt de referència  $m_j$ . També cal destacar que en  $x_{k+1}$  s'afegeix un nou punt de referència, que es relacionarà amb el punt anterior  $m_j$ . Al final de tot aquest procés s'obté una xarxa de punts de referència relacionats entre ells. La precisió d'aquestes estimacions serà més elevada com més observacions s'hagin pres i s'hagin pogut associar a observacions anteriors.

### 3.2. Solucionant el problema

Un dels principals reptes per a la solució del problema és la representació dels dos models definits, el d'observació i el de moviment. Aquestes representacions poden ser en format d'un model espai d'estats amb soroll Gaussià (com es planteja en la proposta amb filtre de Kalman extès *EKF-SLAM*<sup>2</sup>), un conjunt de mostres amb una distribució no Gaussiana (plantejat a *FastSLAM* [3]) o altres.

Com s'ha comentat amb anterioritat, una nova observació obliga a recalculer les relacions entre els diferents components. Un punt important en la resolució del problema és com reduir l'elevat cost que comporta aquesta situació. Hi han diferents aproximacions [4]: dispersió, actualitzacions particionades, mètodes de submapeig, etcètera.

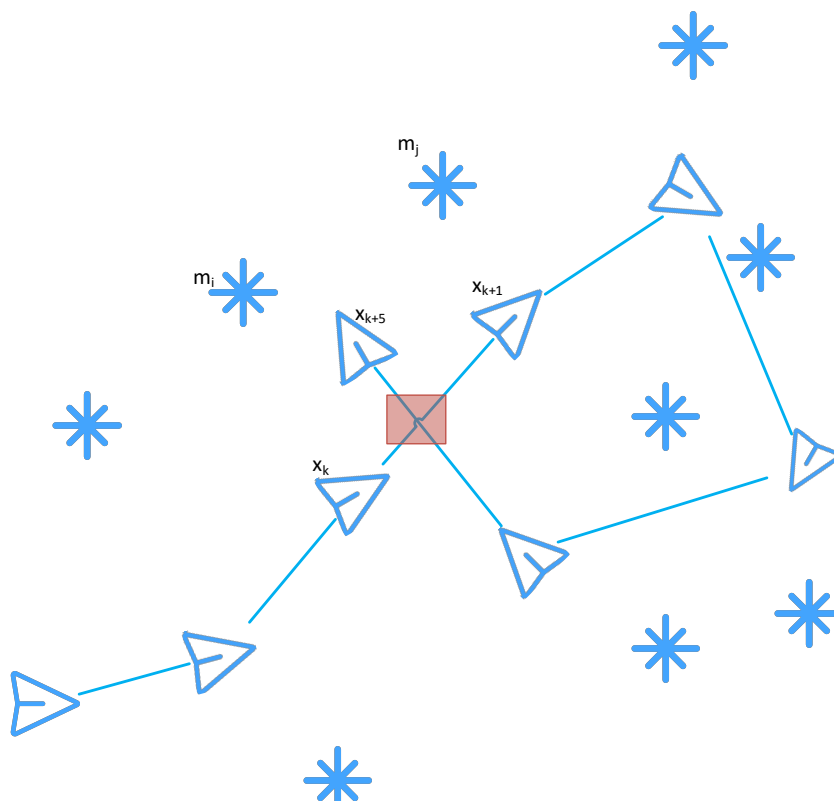
<sup>2</sup> [http://www.iri.upc.edu/people/jsola/JoanSola/objectes/curs\\_SLAM/SLAM2D/SLAM%20course.pdf](http://www.iri.upc.edu/people/jsola/JoanSola/objectes/curs_SLAM/SLAM2D/SLAM%20course.pdf)

### 3.3. Dibuixant el mon

En els inicis de SLAM [4] es pensava que es podia modelar l'entorn del robot com a un conjunt de punts de referència discrets, descrivint-los com a primitives geomètriques. Aquesta premissa no és compleix del tot en entorns més complexes, com pot ser el fons marí. Les observacions es poden fer amb làser, sonar o càmera, entre altres sensors. Quan es tracta d'un sensor visual es parla de V-SLAM.

Per a poder implementar un V-SLAM complet [5] es necessita que aquest implementi una sèrie mòduls: transmissió de dades del sensor visual, odòmetre visual, optimització, detecció de bucles o *loop closure*, i mapeig.

La detecció del tancament de bucles és un aspecte important en SLAM. Es pot definir [6] com que un vehicle visita un lloc pel que ha passat prèviament, tal com es pot identificar a la Il·lustració 5. En aquesta imatge, el robot (triangle) es desplaça identificant punts de referència (estels). En vermell el tancament de bucle detectat. El tancament de bucle s'ha detectat en la nova posició ( $x_{k+\delta}$ ) on el robot ha identificat dos punts de referència ( $m_i$  i  $m_j$ ) que ja s'havien detectat posicions anteriors ( $x_k$  i  $x_{k+1}$ ). Una detecció de bucles precisa i eficient redueix l'error acumulat i permet construir un mapa més consistent [6]. En canvi, un reconeixement incorrecte d'un tancament de bucle pot provocar errors en el procés d'optimització i fer irrecuperable la cartografia i defectes de mapeig [7]. En el cas contrari, no detectar correctament un bucle, no afecta de forma tan negativa al procés ja que no afegeix informació errònia a l'optimitzador.



Il·lustració 5 Identificació d'un bucle en el desplaçament d'un robot.

La detecció de bucles es troba lligada al sensor o sensors que fan servir. En el cas concret de V-SLAM, la detecció de bucles es farà comparant imatges. Si una imatge conté part d'una escena d'una imatge anterior es pot considerar un bucle.

El procés de detecció de bucles acostuma a tenir dues fases [6]: extracció de característiques invariants de la imatge i una estimació de similitud d'aquestes característiques amb altres imatges candidates per tancar el bucle.

Un model habitual en la detecció de bucles és el Bossa de Paraules o *Bag-Of-Words* (*BoW*). Aquest model [8] agrupa descriptors de característiques en un diccionari per agrupacions *k-means*. Aquest diccionari genera un seguit de paraules. La repetició d'aquestes paraules pot indicar un bucle. En aquest tipus de model, les característiques són seleccionades manualment. Això provoca que l'algoritme no sigui eficient i que no aprofiti totes les possibles característiques d'una imatge, que pot produir una baixa precisió.

Amb aprenentatge profund, i concretament amb xarxes de tipus convolucionals, es pot millorar el procés d'extracció de característiques de les imatges i trobar possibles bucles de forma més eficient i amb una precisió major.

## 4. Xarxes Neuronals

En aquest projecte es busca la detecció de bucles que es realitzarà sobre imatges de fons marins en el procés de localització d'un robot aquàtic. Al capítol 3 s'ha explicat com d'important es la detecció de bucles de forma general per a SLAM. Al final d'aquell capítol s'indicava com la intel·ligència artificial, i concretament les xarxes neuronals, poden ajudar en aquest procés. Al capítol 5 es detallarà un estudi de *N. Merril* i *G. Huang* [9] on defineixen una arquitectura concreta per detectar bucles d'àmbit general. Aquest estudi fa us de diferents tipus d'estructures que s'intenten definir en aquest capítol.

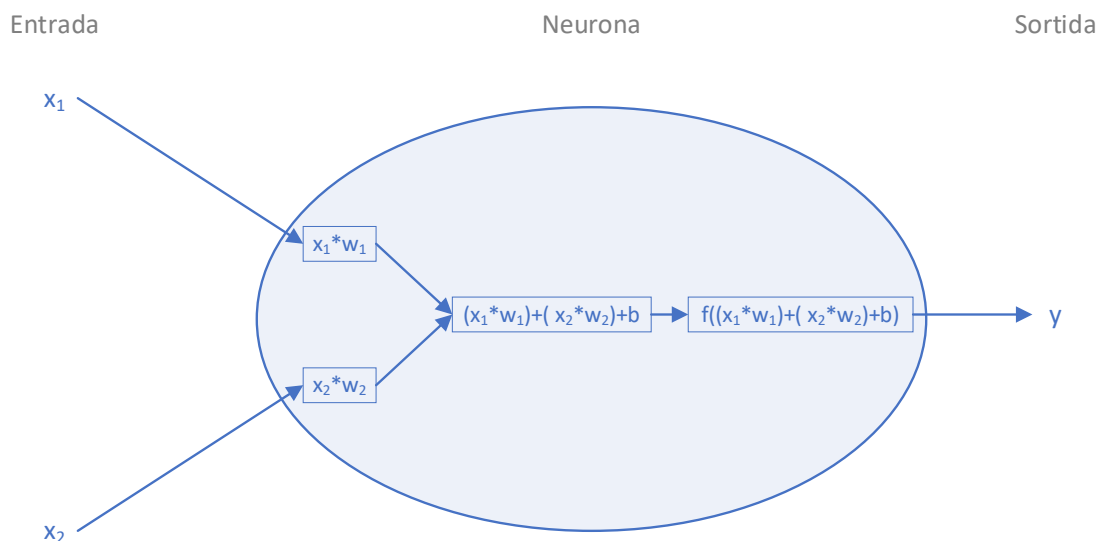
### 4.1. Neurona

Dins de la intel·ligència artificial podem considerar una neurona [10] com una entitat que rep unes entrades, du a terme unes operacions matemàtiques sobre les entrades i genera una sortida. Aquestes operacions matemàtiques es podrien simplificar com:

1. Cada entrada ( $x_i$ ) es multiplica per un pes ( $w_i$ ):  $x_i \rightarrow x_i * w_i$
2. Es sumen tots els resultats del punt anterior (on  $n$  és el número d'elements d'entrada) i s'afegeix un biaix ( $b$ ):  $b + \sum_{i=1}^n x_i * w_i$
3. El resultat ( $y$ ) és l'aplicació d'una funció d'activació ( $f$ ) sobre el resultat del punt anterior:  $y = f(b + \sum_{i=1}^n x_i * w_i)$

La Il·lustració 6 ofereix una interpretació d'aquesta neurona. Amb aquesta fórmula, i tenint un conjunt de dades d'entrada,  $X$  on  $x_i \in X$ , es pot entrenar aquesta neurona assignant diferents valors als pesos ( $w_i$ ), biaixos ( $b$ ) i funcions d'activacions per obtenir un valor  $y$  esperat. Un cop entrenada la neurona, per tant definits els valor de  $w_i$ ,  $b$  i la funció d'activació donada, es pot predir el resultat d'un valor d'entrada  $x_j \notin X$ .

Aquesta arquitectura es coneix com Perceptró i va ser ideada al 1957 per Frank Rosenblatt [11]. Aquest treball partia d'un estudi previ de Warren S. McCulloch i Walter Pitts del 1943 [12].



Il·lustració 6 Estructura bàsica d'una neurona basada en la documentació de [8].

Una funció d'activació es pot interpretar com aquella que determina si la neurona s'ha d'activar o no. Per exemple, si s'està classificant imatges on es poden veure gats o no, el resultat de la funció pot indicar que a la imatge apareix o no un gat. Això es podria

interpretar com que un valor proper a 1 és que té un gat, i un valor proper a 0 és que no, per exemple. El resultat de la funció d'activació acostuma a estar normalitzat entre 0 i 1 o -1 i 1.

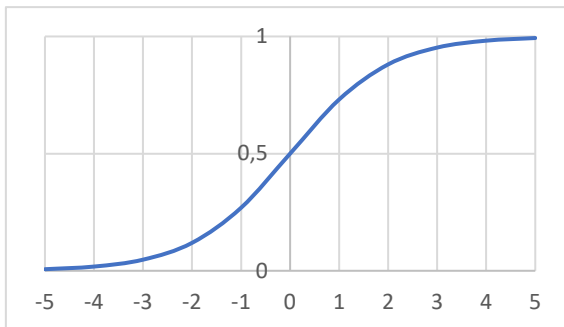
Abans d'aplicar la funció d'activació es té una equació de l'estil  $b + (x_i * w_i)$ . Aquesta funció, que és de tipus polinòmica, és una funció lineal<sup>3</sup>. La funció d'activació permet afegir no linealitat a la sortida de la neurona [13].

Existeixen diferents tipus de funcions d'activació. Dues de les més habituals són la Sigmoide (o Logística) i la ReLU.

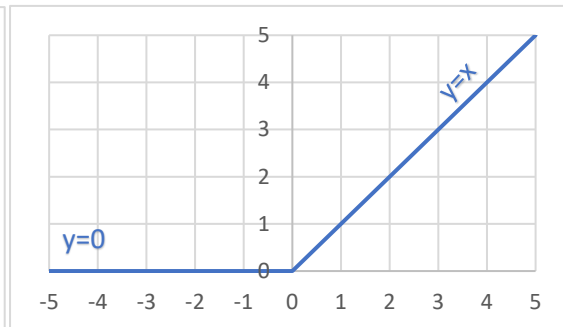
La funció d'activació Sigmoide té com a equació  $y = \frac{1}{1+e^{-x}}$  i té la forma indicada a la Il·lustració 7.

Amb la funció Sigmoide, valors d'entrada molt negatius tendeixen a 0. En canvi, els valors positius grans tendeixen a 1. Aquesta funció [14] permet indicar percentatges de pertinença a una determinada classe.

*Rectifier Linear Unit* o ReLU és un tipus de funció d'activació definida per l'equació  $y = \max(0, x)$  i té la forma indicada a Il·lustració 8.



Il·lustració 7 Gràfica de la funció Sigmoide.



Il·lustració 8 Gràfica de la funció ReLU.

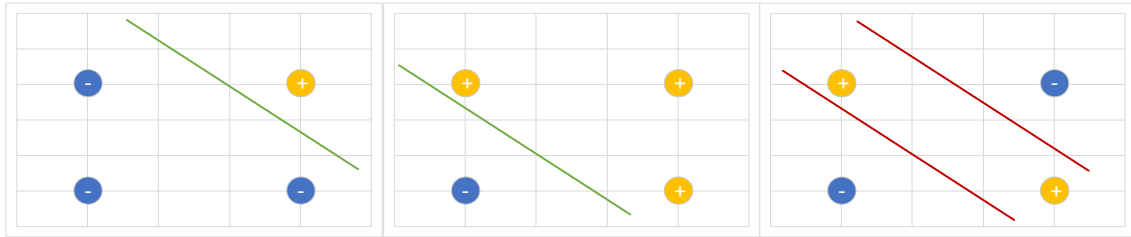
La funció ReLU s'empra de forma habitual a les xarxes convolucionals. Aquesta funció d'activació té diferents avantatges [15]:

- Té un cost baix per calcular el resultat, el que permet ser més eficient a l'entrenar les xarxes neuronals.
- Convergeix més ràpidament que altres funcions, el que elimina el problema al calcular la baixada del gradient.
- Per valors negatius no s'activa. Això permet que sigui més ràpida, ja que ocasiona que la xarxa tingui menys elements que calcular. Pot fer que sigui més precisa en determinats models.

<sup>3</sup> Es pot trobar més informació a [https://en.wikipedia.org/wiki/Linear\\_function\\_\(calculus\)](https://en.wikipedia.org/wiki/Linear_function_(calculus)).

## 4.2. Capes de neurones

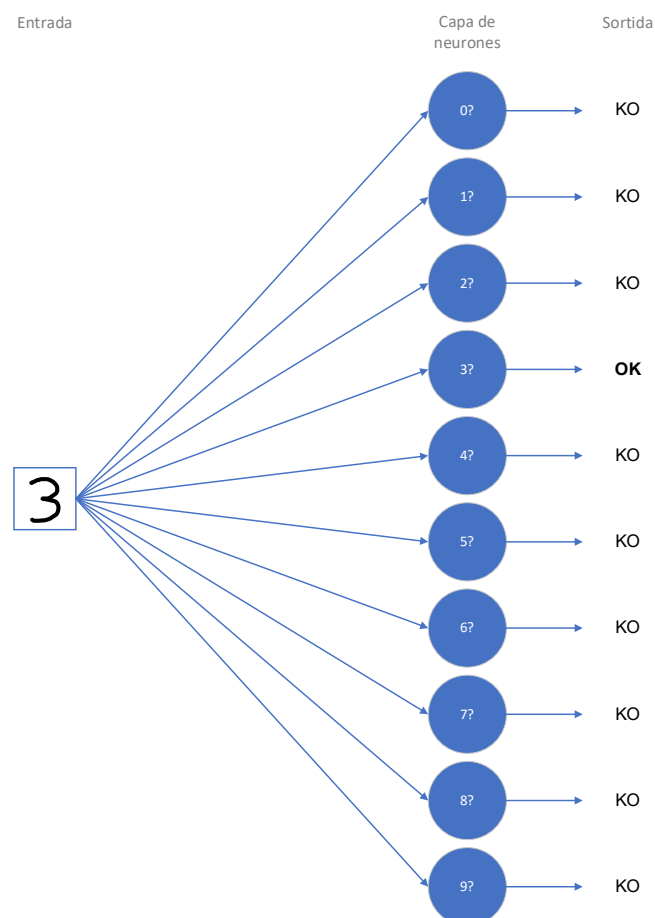
Marvin Minsky i Seymour Papert al 1969 publiquen un estudi en el que demostren una de les limitacions més importants dels perceptrons: no poden aprendre les funcions XOR ja que no són separables linealment [16]. A la Il·lustració 9 es pot observar com amb un



Il·lustració 9 Classificació amb un únic perceptró.

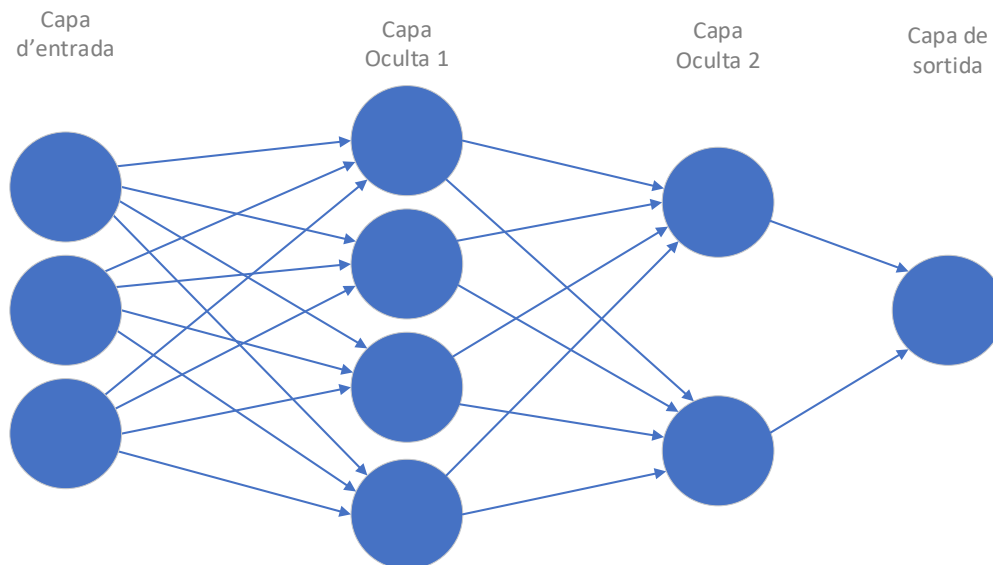
perceptró es pot classificar un *AND* (esquerra) i un *OR* (mig), però no es pot classificar un *XOR* (dreta).

Amb una neurona es pot classificar sí una determinada entrada pertany o no a una classe concreta. Per exemple, si com a entrada es disposa un conjunt d'imatges que representen números escrits per persones, es pot entrenar una neurona per a indicar si una entrada concreta és o no un "1". Si s'agrupen neurones formant una capa, per exemple de deu neurones, aquestes es poden entrenar [16] per a que cadascuna determini si és o no un determinat número, del "0" al "9". Un exemple es pot trobar a la Il·lustració 10.



Il·lustració 10 Exemple d'una capa amb deu neurones.

Diferents grups de capes es poden agrupar formant estructures més complexes, originant una xarxa neuronal. Aquestes xarxes tindran una capa d'entrada i una capa de sortida. Entre aquestes es poden afegir capes ocultes. Aquest ús de capes ocultes proporcionen més capacitat d'abstracció o jerarquització al model. Per exemple [17] si



Il·lustració 11 Exemple d'una xarxa neuronal amb capes ocultes.

a l'entrada hi ha una imatge i a la sortida es vol identificar si aquesta imatge representa un gos, la primera capa d'entrada tindria cadascun dels píxels que componen la imatge. La següent capa (primera capa oculta) podria identificar els llinars dels objectes. La següent capa (oculta) identificaria les formes. A continuació una capa permet trobar característiques. Finalment a la sortida hi ha una etiqueta per indicar si a la imatge hi ha representat un gos. A la Il·lustració 11 es troba un exemple d'aquest model.

Aquesta arquitectura, una capa d'entrada, una o varies capes ocultes i una de sortida, rep el nom de *Multi-Layer Perceptron* o *MLP*. Quan aquest model està compost per varies capes ocultes es parla d'Aprenentatge Profund o *Deep Learning* [18].

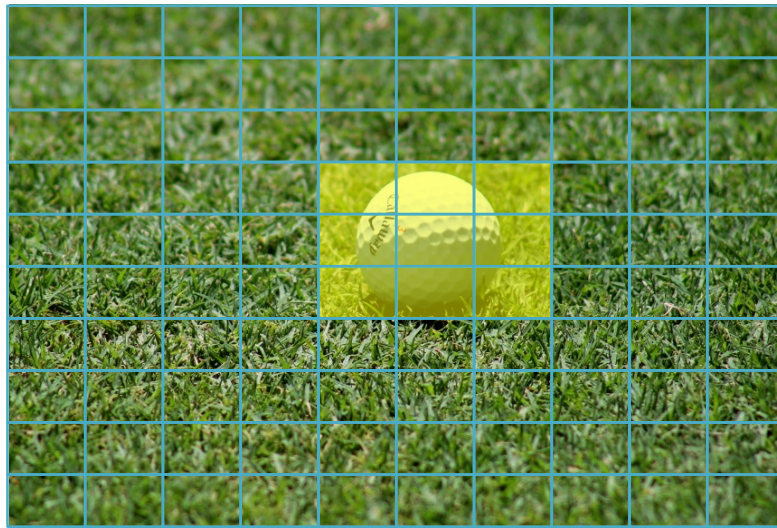
### 4.3. Xarxes Neuronals Convolucionals

Un tipus de xarxa neuronal àmpliament usat en el tractament d'imatges són les xarxes neuronals convolucionals. Es pot interpretar una imatge com un conjunt de píxels. Cadascun d'aquests es pot considerar com una característica per al model. Com és comentat en un exemple de [19], una imatge de 100 píxels horitzontals i 100 píxels verticals equival a 10.000 característiques en una xarxa neuronal.

La idea inicial de les xarxes convolucionals [19] és que els píxels propers entre si contenen informació relacionada. La informació rellevant es troba als llocs de la imatge on es representen els contorns o llinars dels objectes que apareixen en aquesta. En una imatge d'un camp de golf es pot distingir una pilota de la gespa: tota la zona verda pot representar la gespa i una esfera sobre el fons verd pot representar una pilota.

Continuant amb aquest exemple, la pilota només estarà en una part local de la imatge. Si subdividim en seccions rectangulars la imatge original, la pilota només estarà en una o varies seccions, depenent de la mida de la finestra. Si es recorren totes les seccions, una o diverses d'aquestes representarà el contorn de la pilota. Aquest procés rep el nom de finestra mòbil. A la Il·lustració 12 hi ha un exemple figurat d'una finestra mòbil.

Amb aquesta informació es pot obtenir un filtre que determini si una secció representa a una pilota. Aquest filtre tindrà una mida de  $h$  píxels verticals i  $w$  píxels horitzontals, també anomenats *kernel size*.



Il·lustració 12 Exemple d'una finestra mòbil. Imatge original de pxhere.

Per aproximar-se al comportament d'una xarxa neuronal, la Il·lustració 12 es pot reinterpretar com una matriu de dades on la gespa equival al valor 0 i la pilota al valor 1. El resultat es pot observar en la Il·lustració 13.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Il·lustració 13 Representació d'una imatge en valors binaris.

Per identificar una entitat en una imatge es necessita un filtre que representi aquella entitat. En el cas d'exemple, seria necessari un filtre per identificar una pilota. La representació d'aquest filtre el es troba a Il·lustració 14, on els valors 1 equivalen al llindar de la pilota. Aquest filtre representat té una mida de  $3 \times 3$ . Aquest filtre o màscara és el que la xarxa neuronal aprèn durant l'entrenament.

0	1	0
1	0	1
0	1	0

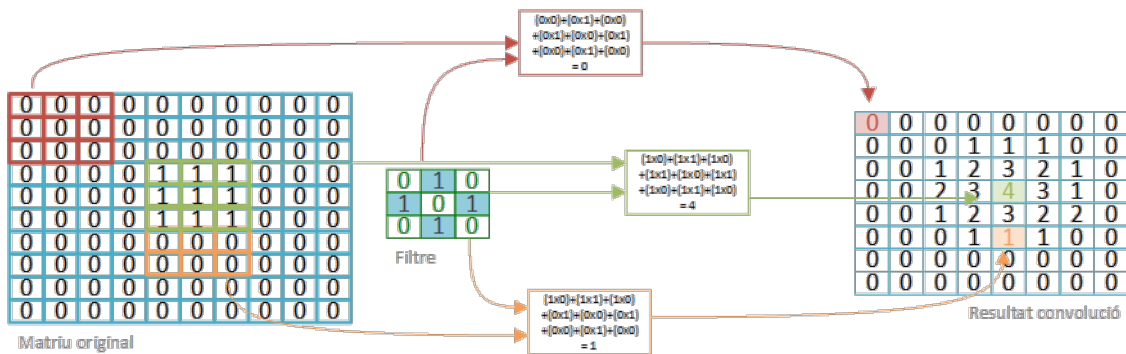
Il·lustració 14 Representació d'un filtre d'una pilota.



Aplicant el procés de convolució d'aquest filtre sobre la imatge anterior es pot determinar que la imatge conté una pilota. Matemàticament [14] la convolució de matrius de dues dimensions es representa a la següent equació:

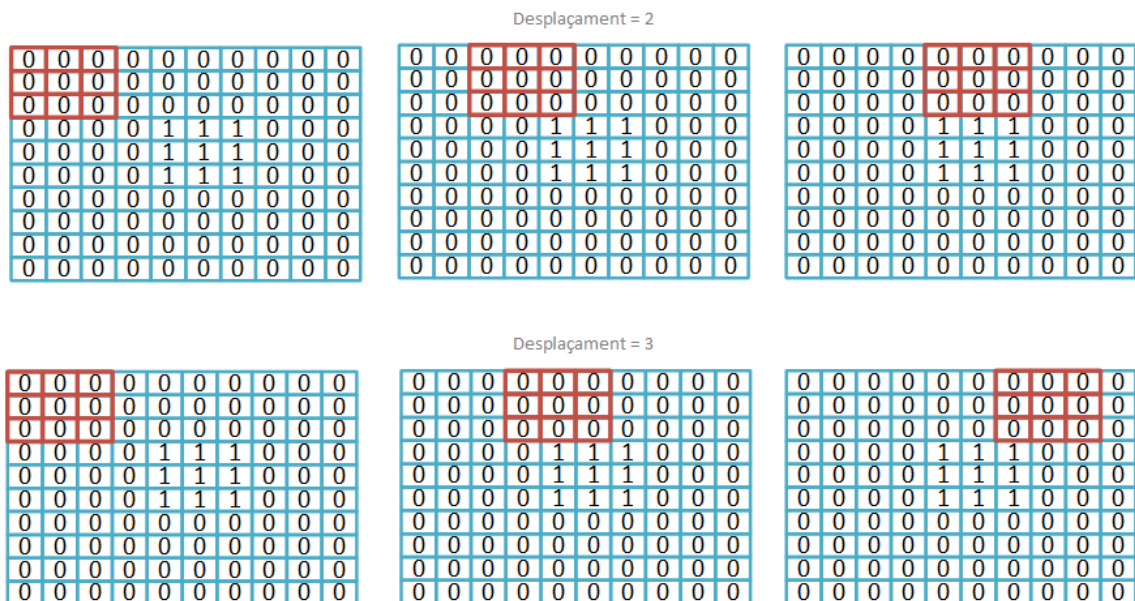
$$Y = X * W \rightarrow Y[i, j] = \sum_{k_1=-\infty}^{+\infty} \sum_{k_2=-\infty}^{+\infty} X[i - k_1, j - k_2]W[k_1, k_2]$$

D'aquesta equació,  $X$  és una matriu de mida  $n_1 \times n_2$  que representa la imatge original.  $W$  és la matriu del filtre amb una mida  $m_1 \times m_2$ , on  $m_1 \leq n_1$  i  $m_2 \leq n_2$ .



Il·lustració 15 Exemple de càlcul d'una convolució.

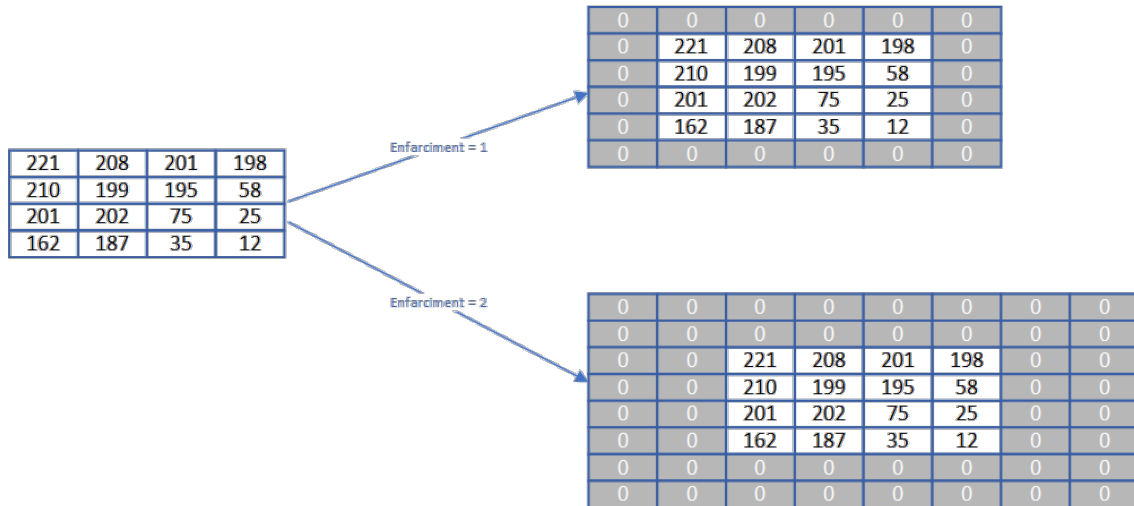
A la Il·lustració 15 es pot observar un exemple del càlcul de convolució aplicat a una matriu. En aquest cas el càlcul s'inicia per la cantonada superior esquerra, en vermell. Aquest càlcul es representa amb una submatriu de mida igual a la del filtre. El següent càlcul seria moure la finestra mòbil a la següent posició. Aquesta nova posició varia segons el desplaçament seleccionat, en el cas que fos 1, la finestra mòbil es desplaça una columna cap a la dreta. A l'arribar a l'extrem dret de la matriu el següent càlcul, segons el desplaçament indicat, podria ser tornar a l'extrem esquerra però una fila inferior a l'actual. Aquest procés es produeix iterativament fins recórrer tota la matriu original. Aquest procés genera com a resultat una nova matriu.



Il·lustració 16 Exemples d'una finestra mòbil amb desplaçament 2 i 3.

En aquest exemple la finestra mòbil es desplaça en cada càlcul un espai. Aquest paràmetre s'anomena desplaçament o *stride* i pot prendre valors superiors a 1. A la Il·lustració 16 es pot observar un exemple amb dos desplaçaments diferents.

En aquest procediment de convolució es pot afegir farciment o *padding* a la matriu original. Aquest procés consisteix en afegir noves files o columnes en els extrems de la matriu, i emplenar aquestes noves cel·les amb zeros. Serveix [19] per escanejar millor els llindars de la imatge original. Un exemple de farciment d'una matriu es pot visualitzar a la Il·lustració 17.



Il·lustració 17 Exemple d'afegir farciment a una matriu.

En l'exemple d'aquesta secció, el farciment no és útil ja que la pilota és al centre de la imatge i no es necessiten escanejar els llindars. També cal recordar que en aquest exemple s'ha definit la gespa com 0 i la pilota com 1, però això no serà extrapolable a la realitat.

Amb aquestes dades, es pot calcular la mida que tindrà la matriu resultant de la convolució amb les següents fórmules [20]:

$$numCol = \left\lceil \frac{W - F + 2P}{S} \right\rceil + 1$$

$$numFil = \left\lceil \frac{H - F + 2P}{S} \right\rceil + 1$$

A on *numCol* és el número de columnes de la matriu resultant, *numFil* és el número de files de la matriu resultant, *W* és el número de columnes de la matriu original, *H* és el número de files de la matriu original, *F* és la mida del filtre, *P* és el farciment afegit i *S* és el desplaçament del filtre.

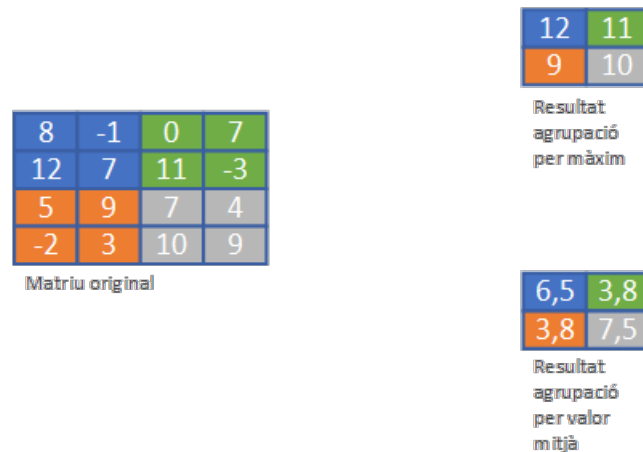
Com s'ha vist anteriorment, el píxel és la unitat bàsica d'una imatge. I aquesta es pot interpretar com una matriu de píxels. Aquesta informació es pot descompondre i treballar amb tres matrius, una per cada canal de color primari, si la imatge és en color. També es pot transformar per a que sigui un únic canal en escala de grisos.

El procés que s'ha fet fins ara és el d'aplicar una capa de convolució a una matriu. Amb aquesta s'ha pogut interpretar que els llindars d'un pilota es troben en la imatge original. Afegint una capa a continuació [18] es podrien identificar altres elements. Una altra capa més podria identificar el tipus de pilota i així arribar a un procés més abstracte i complex.

#### 4.4. Capes d'agrupació

El procés de convolució origina una matriu d'una mida gran. Per millorar l'eficiència i reduir la mida de les xarxes neuronals, després d'una capa de convolució s'acostuma a posar una capa de tipus reducció, submostreig o *pooling*.

Aquest tipus de capa fa servir una finestra mòbil, amb una mida i un pas concret, per recórrer la capa anterior i generar una matriu resultant que, depenent del tipus, agafarà el valor màxim o mitjà de la finestra mòbil. Un exemple es pot trobar a la Il·lustració 18.

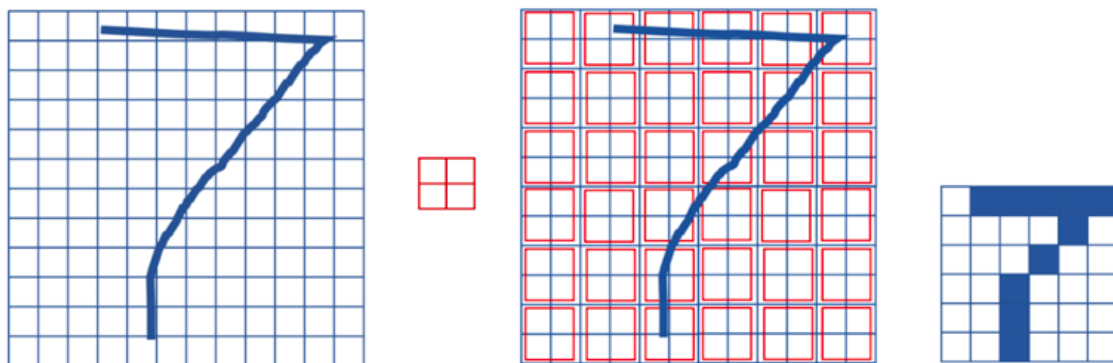


Il·lustració 18 Exemple d'aplicació d'una capa d'agrupació.

L'ús de les capes d'agrupació [19] [21] principalment té dos objectius:

- Millorar la precisió. Això es produeix per que és més resistent al soroll de les dades d'entrada.
- Redueix el número de paràmetres. Aquest punt comporta dos avantatges: aconseguir una millor velocitat d'entrenament i reduir el grau de sobreajust.

Un exemple del resultat d'aquesta capa es pot trobar a la Il·lustració 19.



Il·lustració 19 Resultat d'aplicar una capa d'agrupació, [14].

## 5. Disseny de la xarxa neuronal

Aquest capítol especifica el disseny de la xarxa neuronal per a la detecció de bucles en fons marins. Primerament detalla el treball de Merrill i Huang on construeixen una xarxa per a la detecció de bucles de forma general. A continuació repassa la diferència que implica la visió del fons marins respecte la visió terrestre de l'algoritme de Merrill i Huang. Posteriorment identifica quan es pot interpretar un bucle. Per acabar defineix l'arquitectura de la capa que s'implementarà en aquest projecte.

### 5.1. Algoritme de tancament de bucle per N. Merrill i G. Huang

En aquesta secció es detalla l'algoritme per a la detecció de bucles sense supervisió ideat per *N. Merrill* i *G. Huang* [9]. Es centra en aspectes fonamentals que serveixen com a base de molts algorismes d'estil similar: preparació de les dades d'entrada, generació d'imatges sintètiques, ...

La intenció dels autors d'aquest algoritme es crear un mètode per a la detecció del tancament de bucles que sigui compacte i fiable. L'objectiu és que pugui ser executat en temps real per robots que no hagin de tenir gran potència de càlcul. A més, l'algoritme proporcionat és sense supervisió, el que permet reaprofitar-lo i no haver d'etiquetar les imatges prèviament.

Per crear aquest algoritme els autors usen, entre altres eines, la creació d'imatges sintètiques a partir de les imatges reals, emulant el moviment del robot. També utilitzen la informació de la geometria i la il·luminació que proporciona l'histograma de gradients orientats o *Histogram of Oriented Gradients* (HOG).

L'arquitectura de la xarxa neuronal serà de tipus convolucional autocodificadora amb eliminació de soroll o *Deep Convolutional Denoising Autoencoder*.

#### Tractament de les dades d'entrada de l'entrenament

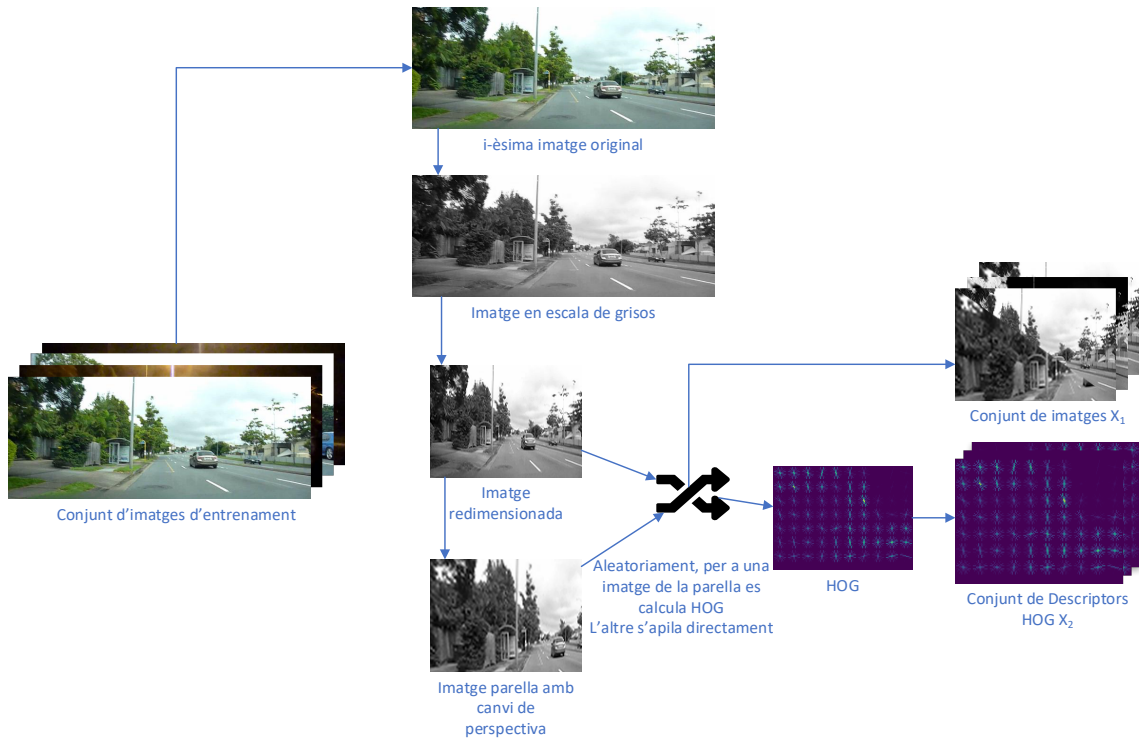
El primer pas de l'algoritme és el tractament de les dades per a l'entrenament de la xarxa neuronal. Donat un conjunt de mida  $N$  d'imatges d'entrada per a l'entrenament, per cadascuna de les imatges es realitzen els següents passos:

1. Es converteix a escala de grisos.
2. Es redimensiona a  $120 \times 160$  píxels.
3. Es crea una imatge parella aplicant una modificació de la perspectiva.
4. Es selecciona de forma aleatòria una de les dues imatges. Es calcula el vector de descriptors HOG de mida 3648 d'aquesta imatge.
5. El descriptor HOG d'aquesta parella s'apila al conjunt  $X_2$ .
6. L'altra imatge s'apila al conjunt  $X_1$ .

Al final d'aquest procés iteratiu s'obtenen:

- Conjunt d'imatges  $X_1$  de mida  $N \times 120 \times 160$ .
- Conjunt  $X_2$  de vectors de descriptors HOG de mida  $N \times D$ , on  $D$  és la dimensió del vector de descriptors HOG, en aquest cas 3648.

A la Il·lustració 20 es resumeix aquest procés. En aquesta imatge es pot observar un conjunt d'entrada, a l'esquerra, i dos conjunts de sortida, a la dreta. Al centre es pot observar, de forma seqüencial, els passos pel tractament de les dades d'entrada.



Il·lustració 20 Tractament de dades per a l'entrenament de l'algoritme [7]. Imatge creada amb recursos de [27], [30] i <https://fontawesome.com>.

El primer pas d'aquest tractament és transformar la imatge original a escala de grisos. Aquesta transformació permet reduir l'ús de memòria ja que en comptes de tenir tres matrius, una per cada canal de color, es té només una. Al reduir la memòria l'algoritme serà més adient en la identificació del bucles en temps real.

El següent pas és redimensionar la imatge. L'entrenament de les xarxes neuronals que treballin amb imatges com a dades d'entrada s'han de redimensionar per tenir totes les mateixes mides [19].

A continuació l'algoritme genera una imatge sintètica. Aquesta és una transformació projectiva aleatòria a partir de la imatge d'entrada. Aquest pas pretén emular el moviment del robot i possibles canvis del seu punt de vista.

A la Il·lustració 21 es pot observar un exemple de moviment d'un robot. En aquest cas, un possible punt de referència, en requadre verd, pateix una augmentació respecte des de la posició inicial a la final. A la Il·lustració 22 es pot observar com la visió d'un robot ha girat cap a l'esquerra, provocant una transformació diferent.

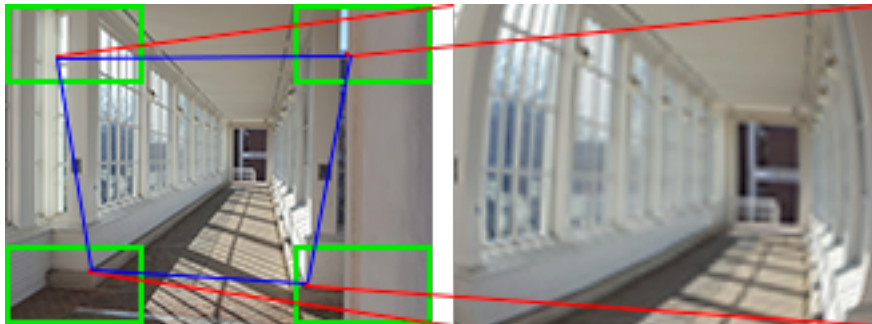


Il·lustració 21 Exemple de visió d'un robot en moviment. Recursos usats de [28].



Il·lustració 22 Exemple de moviment de robot girant. Recursos usats de [28].

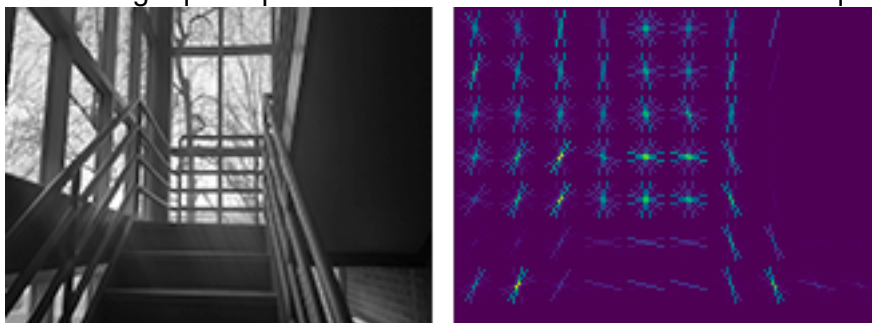
Per emular el moviment del robot i generar noves imatges sintètiques, es seleccionen quatre punts aleatoris dins d'uns marges propers a cada cantonada de la imatge original i fa que aquests siguin els quatre nous punts de les cantonades de la nova imatge. Per a que la distorsió no sigui extrema, aquests quatre punts es seleccionen de la secció entre les cantonades originals més  $Alçada/4$  i  $Amplada/4$ . A la Il·lustració 23 es pot observar aquest procés on els requadres verds són les seccions d'on es triaran els nous extrems de la nova imatge i el requadre blau és la nova imatge generada.



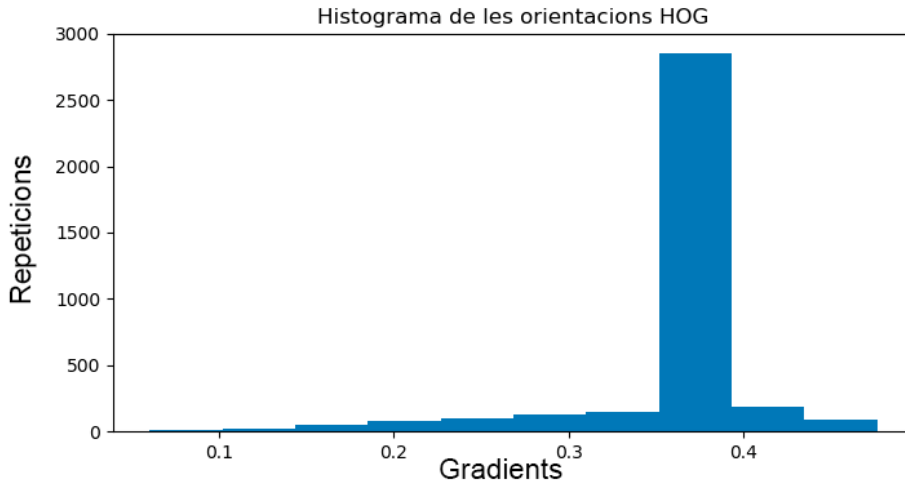
Il·lustració 23 Generació d'una imatge sintètica a partir d'una imatge real. Recursos usats de [31].

En el punt actual es té la imatge original, modificada la dimensió i el nombre canals de colors, i la generada sintèticament a partir d'aquesta. Aquestes dues imatges és desaran en dos conjunts diferents de forma aleatòria. S'introdueix aquesta selecció aleatòria per evitar un possible biaix a l'entrenament que es podria produir si els conjunts de sortida fossin uniformes. Una de les dues imatges s'emmagatzema al conjunt de sortida  $X_1$ .

De l'altra imatge s'obté l'histograma de gradients orientats (HOG) i aquest es desa al conjunt  $X_2$ . Aquest vector HOG proporciona informació geomètrica de la imatge que facilita la comparació entre dues imatges. HOG no aporta robustesa en la detecció amb canvis de perspectiva però sí que aporta informació quan hi ha una variació en la il·luminació. A la Il·lustració 24 es pot observar la representació gràfica d'un vector HOG i a la Il·lustració 25 l'histograma d'aquest vector. L'histograma mostra els diferents gradients de la imatge que capturen informació del contorn i la silueta d'aquesta.

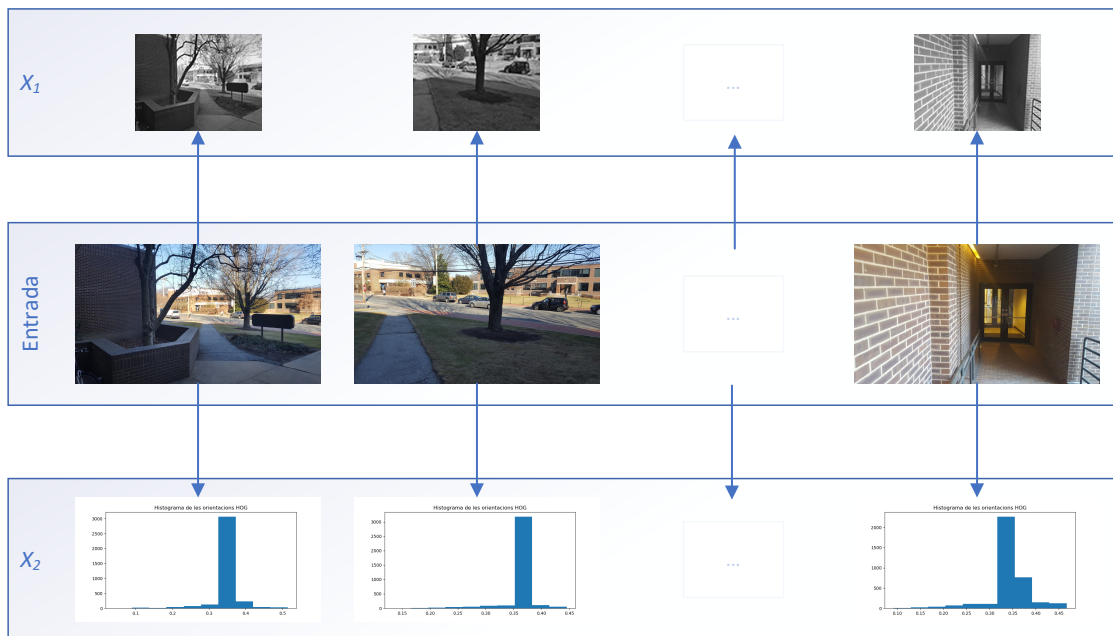


Il·lustració 24 Vector HOG d'una imatge. Recurs [31].



Il·lustració 25 Histograma de la Il·lustració 24. Gràfica obtinguda amb [27].

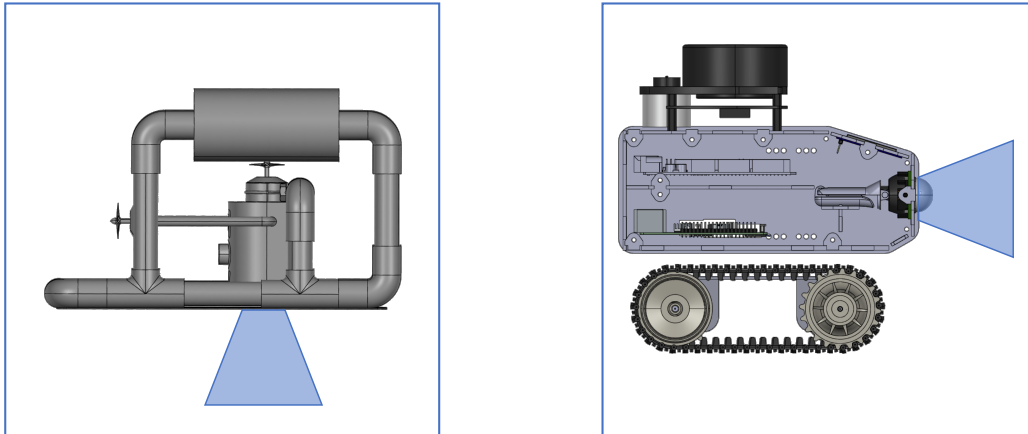
Al final del procés de tractament de dades s'haurà processat de forma iterativa un conjunt d'entrada i generat dos conjunts de sortida: un amb imatges i l'altre amb vectors HOG. La xarxa neuronal utilitzarà el descriptor global proporcionat per HOG en l'entrenament. En aquest procés s'haurà generat una imatge sintètica amb transformacions perspectives respecte l'original. La Il·lustració 26 representa aquest procés.



Il·lustració 26 Representació del procés de tractament de les dades d'entrada. Recursos de [31].

## 5.2. Detecció de bucles en fons marins

Aquest treball agafa com a base l'algoritme ideat per *N. Merril* i *G. Huang* [9]. El tractament de les dades d'entrada que realitzen a l'algoritme de [9] està pensat per a robots terrestres on la visió de la càmera és cap endavant, poden girar cap als costats o canviar i inclinar-se cap al cel o cap al terra. En el cas de robots aquàtics en els que es centra aquest treball la càmera mira cap al fons marí. A la Il·lustració 27 es pot observar la visió de la càmera interpretada pel trapezi blau.

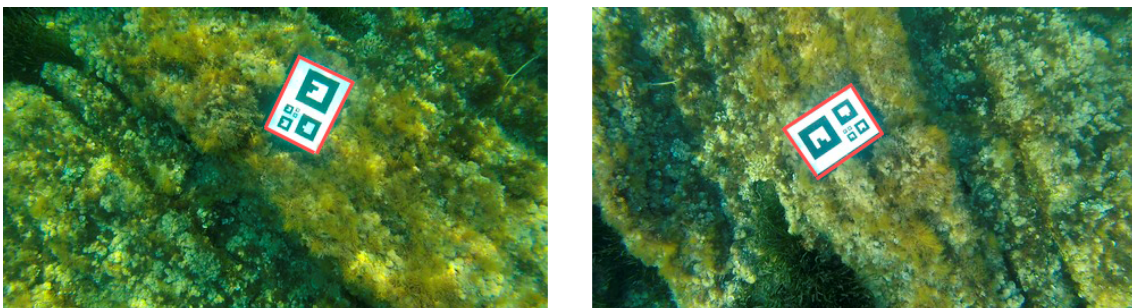


Il·lustració 27 Vista lateral d'un robot aquàtic i terrestre, recursos de GrabCAD Community.

La diferenciació entre el moviment i posició de la càmera dels dos tipus de robots provoca que l'algoritme per a fons marins sigui diferent respecte al plantejat per *Merril* i *Huang* en la generació d'imatges sintètiques. A causa que la càmera mira cap al terra i segueix un pla paral·lel a aquest (sense inclinacions), es poden aplicar translacions, rotacions, canvis d'escala i canvis d'il·luminació. El tractament de les imatges serà semblant al del treball de Merril i Huang, però només aplicant translacions, rotacions, canvis d'escala i d'il·luminació.

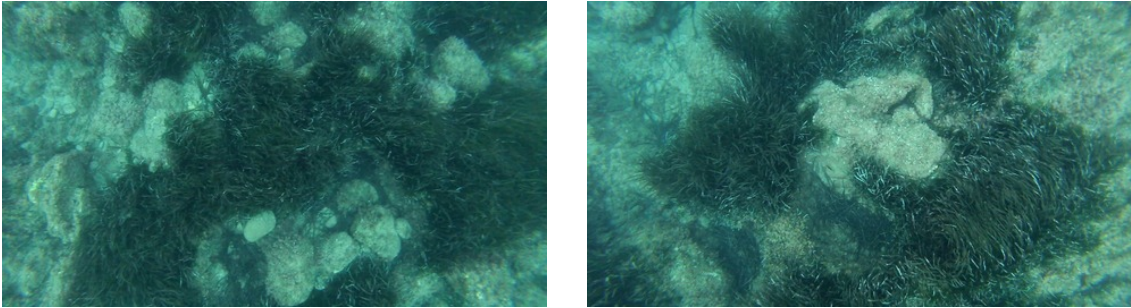
### Identificació de bucle

Un bucle, un vehicle passa per una posició ja observada anteriorment, s'identifica en el cas d'aquest projecte quan dues imatges del fons marí presenten elements comuns de l'escena. L'habitual en aquests casos és que hi hagi una diferència d'escala i/o translació i/o il·luminació, com s'identifica a la Il·lustració 28. En canvi, a la Il·lustració 29 es poden observar dues imatges que no tenen elements comuns, tot i ser semblants, aquestes no



Il·lustració 28 Bucle en un fons marí.





Il·lustració 29 Exemple de dues imatges que no formen un bucle.

formen un bucle. Es pot suposar que dues imatges es troben en un bucle si la descripció global de la imatge que ha predit la xarxa neuronal per dues imatges són semblants. La xarxa s'entrenarà de manera semblant a la proposta de Merrill i Huang, produir un descriptor d'una imatge semblant al descriptor HOG de la imatge parella. Però per tal de generar bucles sintètics no es realitzarà la transformació de perspectiva descrita sinó els canvis indicats per a imatges submarines. La detecció d'un bucle es pot donar segons la següent equació:

$$|f(I_i) - f(I_j)| \sim 0$$

En aquesta equació,  $f(I_i)$  i  $f(I_j)$  són les prediccions de dues imatges. Es pot analitzar un conjunt de dades concret i definir un llindar  $\delta$  per al que si la diferència entre dues imatges és més petit que aquest valor indica que és un bucle, això es representa a:

$$|f(I_i) - f(I_j)| < \delta$$

$$|f(I_i) - f(I_j)| \geq \delta$$

En aquestes equacions es representa quan es detecta un bucle (la diferència entre la predicció de les dues imatges és menor que  $\delta$ ) i quan no (la diferència entre les dues prediccions és igual o major a  $\delta$ ).

En una execució *online* de l'algoritme, la detecció de bucles es faria seleccionant determinats fotogrames clau, per exemple un de cada set, per poder avaluar-ho. Si la consulta d'aquest fotograma clau està per sota del llindar establert és candidat a bucle. Si tres consultes consecutives donen un resultat positiu respecte els sis fotogrames posteriors al fotograma clau identificat anteriorment es confirma com a bucle. Aquests valors són estimacions ja que s'ha de tenir en compte que hi ha factors variables que poden modificar-los, com pot ser la velocitat de moviment del robot.

### 5.3. Arquitectura del model

Aquesta secció detalla l'estructura del model de la xarxa neuronal, tenint com a base la de la xarxa de *N. Merrill* i *G. Huang* [9].

En l'entrenament de la xarxa neuronal, aquesta intentarà reconstruir el conjunt  $X_2$ , vector de descriptors HOG, a partir del conjunt  $X_1$ , imatges. La xarxa neuronal primerament té una fase de codificació i una posterior de descodificació. Al final es calcula la funció de pèrdua.

### Codificació

La fase de codificació està formada per tres conjunts de capes: les dues primeres de convolució i agrupació i la tercera només de convolució.

A la Taula 4 es troba la configuració de les capes de codificació de la xarxa neuronal. A continuació de la tercera s'afegeix una capa, *Flat1*, per aplanar el contingut de la capa anterior i obtenir un vector que serà el conjunt de dades d'entrada

Taula 4 Conjunt de capes de la codificació.

		Conjunt 1 (Conv1)	Conjunt 2 (Conv2)	Conjunt 3 (Conv3)
Convulsió	Entrada	$X_1$	Conv1	Conv2
	Mida del filtre	5x5	4x4	3x3
	Desplaçament	2	1	1
	Número filtres	64	128	4
	Farciment	4	2	0
	Inicialització pesos <sup>4</sup>	Gaussiana, $\sigma=0,01$		
	Inicialització biaixos	Gaussiana, $\sigma=0,01$		
	Funció activació	ReLU		
Reducció	Mida del filtre	3x3		
	Desplaçament	2		
	Mètode d'agrupació	Màxim		
Normalització	Tipus	LRN		
	Canals sumar	5		
	Escala	$1 \times 10^{-4}$		
	Exponent	0,75		

### Descodificació

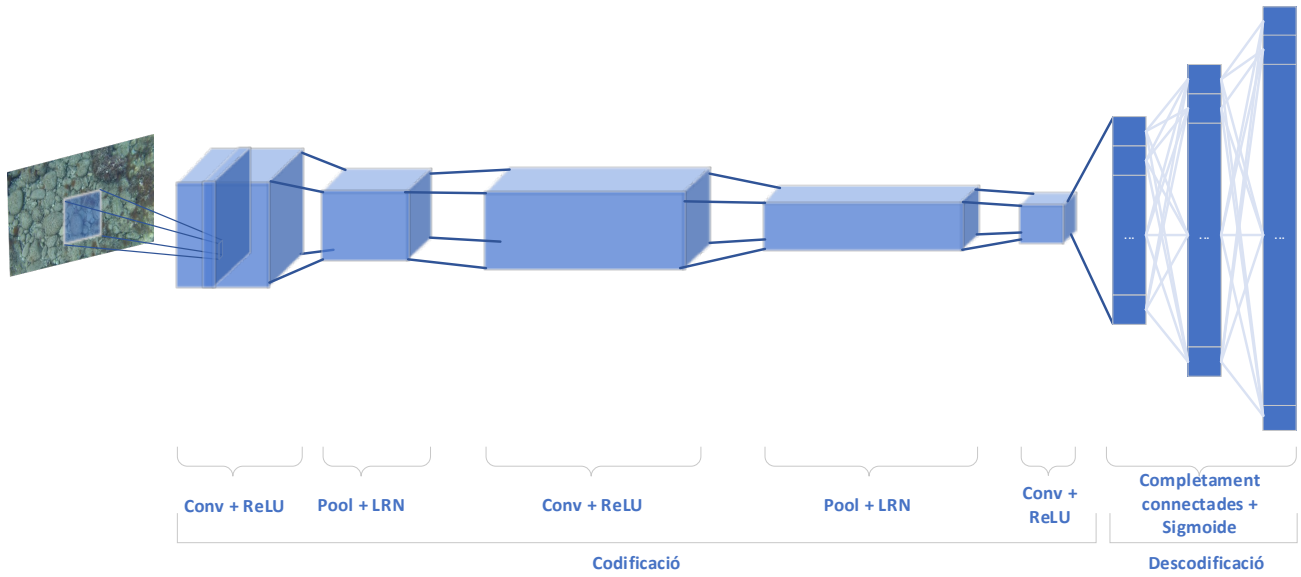
El procés de descodificació consta de tres capes completament connectades, la configuració d'aquestes es troba a la Taula 5.

Taula 5 Configuració de les capes de descodificació.

	Completament connectada (Conn1)	Completament connectada (Conn2)	Completament connectada (Conn3)
Entrada	Flat1	Conn1	Conn2
Número filtres	1064	2048	3648
Inicialització pesos	Gaussiana, $\sigma=0,01$	Gaussiana, $\sigma=0,01$	Gaussiana, $\sigma=0,01$
Inicialització biaixos	Gaussiana, $\sigma=0,01$	Gaussiana, $\sigma=0,01$	Gaussiana, $\sigma=0,01$
Funció activació	Sigmoide	Sigmoide	Sigmoide

<sup>4</sup> Es pot trobar més informació de la inicialització de pesos i biaixos al bloc de *Jihong Ju* [29].

L'arquitectura final del model, composta per una fase de codificació i una posterior de descodificació, es representa a la Il·lustració 30.



Il·lustració 30 Arquitectura de la xarxa neuronal.

### Funció de pèrdua

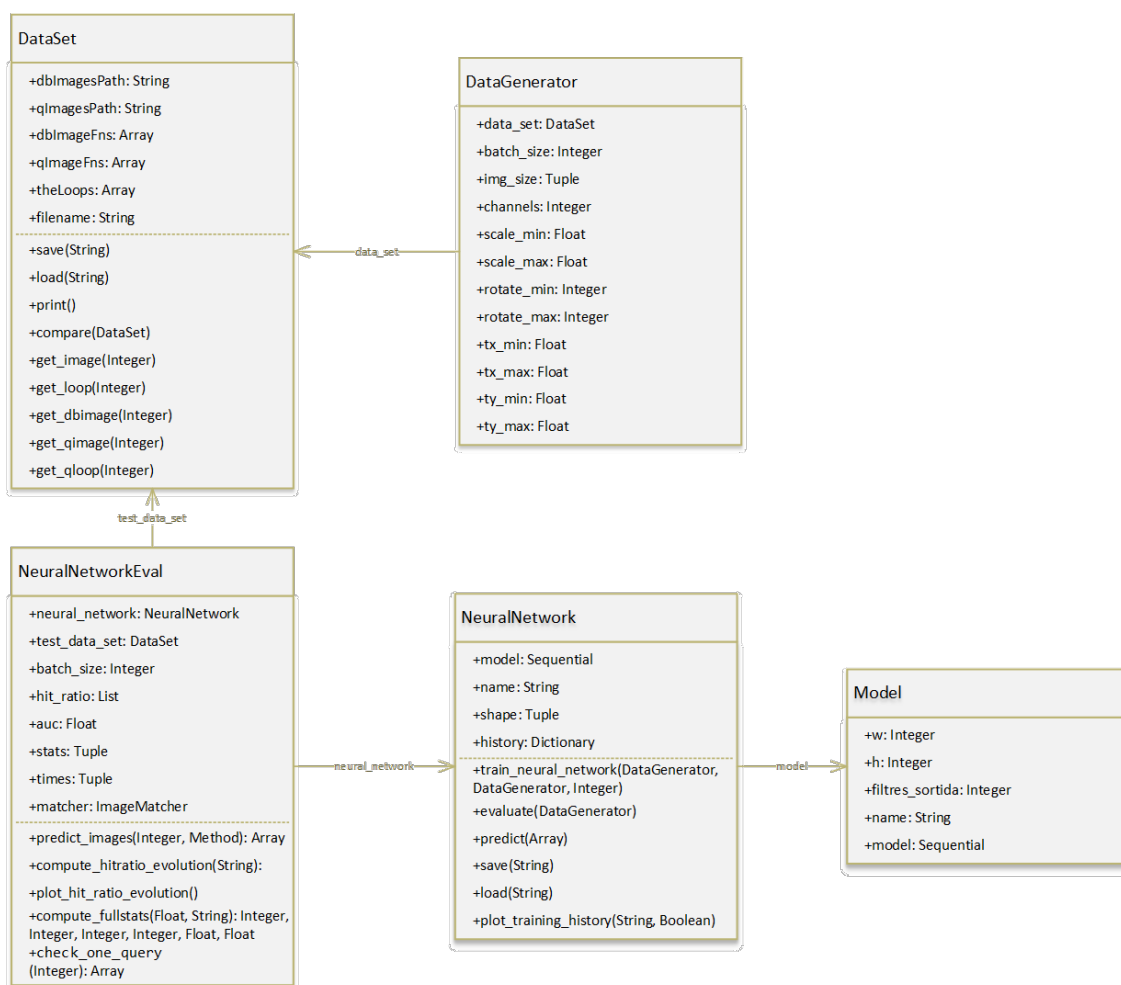
La funció de pèrdua en aquest model és la suma dels quadrats de les diferències, o distància euclidiana o norma L2 de  $Conn3$  i el conjunt  $X_2$  de descriptors HOG.

## 6. Implementació de la xarxa neuronal

Aquest capítol detalla diversos aspectes relacionats amb la implementació de la xarxa. Primer es defineixen les classes creades i els seus usos. S'especifiquen les classes que contenen la xarxa neuronal i el model, i altres classes de gestió del conjunt de dades o altres utilitats. També s'identifica l'origen del conjunt de dades emprat en aquest projecte així com la seva estructura.

### 6.1. Diagrama de classes

A l'il·lustració 31 es mostra el diagrama de classes de l'algorisme. Aquest diagrama mostra les principals classes omertent algunes altres d'utilitats.



Il·lustració 31 Diagrama de classes.

Al diagrama de classes es pot observar com *Model* i *NeuralNetwork* són la base de l'arquitectura de la xarxa neuronal. Les classes *DataSet* i *DataGenerator* són classes que permetran treballar amb dades. Finalment la classe *NeuralNetworkEval* permet avaluar la xarxa neuronal. *Model* i *NeuralNetwork* són classes independents per permetre canviar l'estructura de la xarxa neuronal amb facilitat sense modificar les operacions.

### 6.2. Dades d'entrada

Part del codi font usat en aquest projecte, en especial la dedicada a la del conjunt de dades i de l'avaluació de la xarxa neuronal, i el conjunt de dades, procedeixen del grup

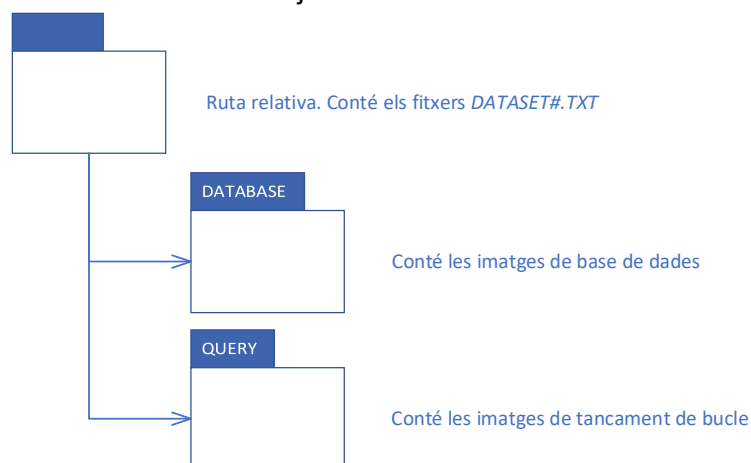
de Sistemes, Robòtica i Visió de la Universitat de les Illes Balears<sup>5</sup>. El conjunt d'imatges ofereix tres grups de dades en format RGB. L'origen son àrees costaneres de Mallorca. Algunes imatges s'han capturat a partir d'una seqüència de vídeo d'una càmera GoPro i les altres imatges s'han obtingut per un robot Sparus II AUV<sup>6</sup> amb càmeres Point Greys CM3-U3-31S4<sup>7</sup> i Manta G283<sup>8</sup>. El conjunt de dades està format pels següent components:

- Conjunt d'imatges que representa la base de dades amb el que entrenarà la xarxa neuronal.
- Conjunt d'imatges on cadascuna d'aquestes és un bucle amb com a mínim una imatge del conjunt anterior. Aquestes s'anomenaran *query*.
- Conjunt de bucles entre imatges.

La informació s'emmagatzema en fitxers anomenats *DATASET#.TXT*, on # és el número del conjunt de dades. Dins d'aquest fitxer es troben els següents camps, separats pel caràcter #:

- *dblImagesPath*: ruta relativa al directori que conté les imatges de la base de dades.
- *qlImagesPath*: ruta relativa a la carpeta que conté les imatges que formen part de bucles.
- *dblImageFns*: separats per comes, els noms dels fitxers de la base de dades.
- *qlImageFns*: separats per comes, els noms dels arxius que són *query*.
- *theLoops*: separats per comes, els índexs que representen bucles dels anteriors camps. La imatge *databaseImage[index[2\*i]]* forma un bucle amb la imatge *queryImage[index[2\*i+1]]*, on *i* va des de 0 fins a la mida de *theLoops*.

Un exemple d'aquesta estructura es pot trobar a la Il·lustració 32. En aquesta imatge es pot identificar com s'emmagatzema la informació d'un bucle concret. A la Il·lustració 33 es troba l'estructura del fitxers del conjunt de dades.



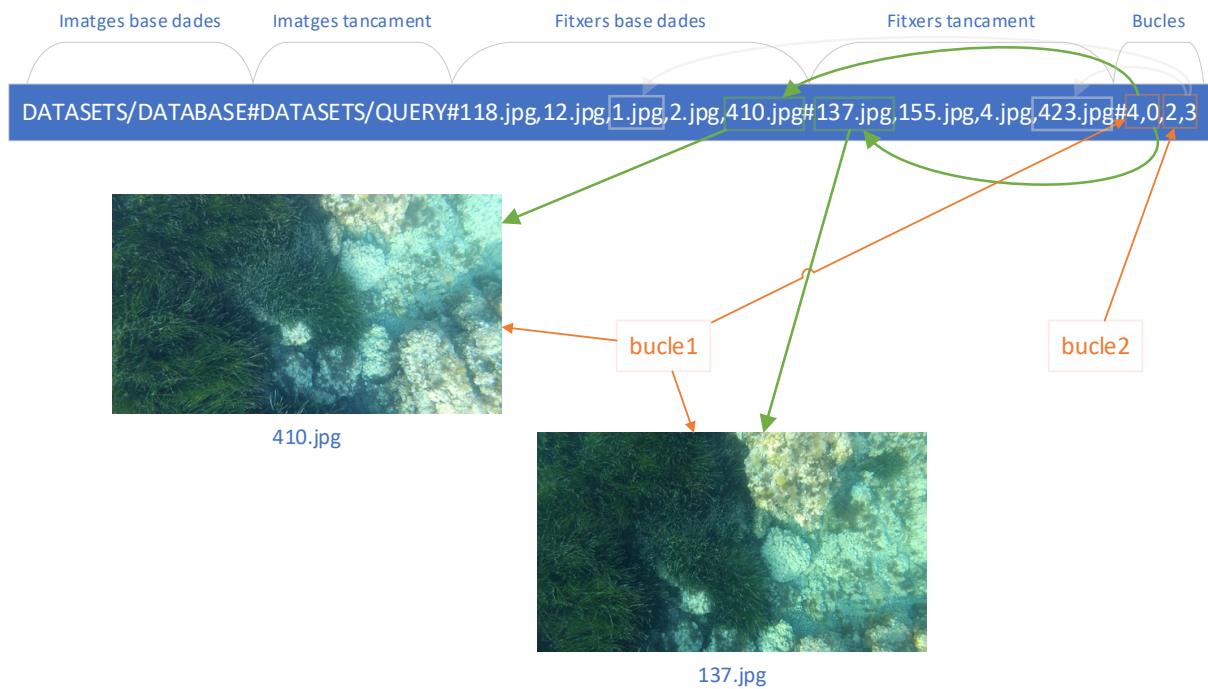
Il·lustració 32 Estructura dels directoris i fitxers del conjunt de dades.

<sup>5</sup> <http://srv.uib.es/>

<sup>6</sup> <http://iquarobotics.com/sparus-ii-auv>

<sup>7</sup> <https://www.flir.com/products/chameleon3-usb3/>

<sup>8</sup> <https://www.alliedvision.com/en/products/cameras/detail/Manta/G-283.html>



Il·lustració 33 Exemple de l'estructura dels fitxers d'informació del conjunt de dades.

L'arxiu *dataset.py* proporciona la classe *DataSet* que ofereix un conjunt d'utilitats per treballar amb l'estructura de dades. Aquestes utilitats són:

- *save(fileName)*: desa informació al fitxer indicat a *fileName*.
- *load(fileName)*: carrega dades del fitxer *fileName*.
- *print*: imprimeix informació sobre el *dataset*.
- *compare(dataset2)*: compara el *dataset* original amb el *dataset2* passat com a paràmetre.
- *get\_image(index)*: proporciona una imatge del *dataset* en format *skimage*<sup>9</sup> (alçada, amplada, canals). Els canals es troben en ordre RGB. La imatge que proporciona pot ser del conjunt *database* o del *query* i depèn de l'índex proporcionat. Si és igual o inferior a la mida del conjunt d'imatge de la base de dades, la imatge serà d'aquest conjunt. Si en canvi és superior, la imatge serà del tipus de tancament de bucle.
- *get\_loop(index)*: obté les dues imatges del bucle indicat per *index*. La primera imatge, *dbImage*, correspon a la imatge, en format *skimage*, de la base de dades. La segona imatge, *qImage*, correspon, en format *skimage*, al bucle.
- *get\_dbimage(index)*: retorna la imatge, en format *skimage*, de la base de dades que es troba a la posició indicada per *index*.
- *get\_qimage(index)*: retorna la imatge, en format *skimage*, que són bucle i es troba a la posició indicada per *index*.

<sup>9</sup> *skimage* és una llibreria per al tractament d'imatges. Podem trobar més informació a <https://scikit-image.org/>

- `get_qloop(index)`: retorna una llista amb les posicions que ocupen les imatges de la base de dades que pertanyen a un bucle que tenen com tancament la imatge `query` indicada per `index`.

### 6.3. Model

Aquesta classe proporciona el model de la xarxa neuronal, agafant com a base *Sequential*<sup>10</sup> de Keras. Aquest model és un conjunt de capes lineal i segueix una estructura similar a la indicada per *Merril* i *Huang* [9]. Aquest model serà la base amb el que s'avaluarà la xarxa neuronal.

La classe presenta els següent atributs:

- `w`: amplada de la matriu d'entrada al model.
- `h`: alçada de la matriu d'entrada al model.
- `filtres_sortida`: mida dels elements que tindrà el resultat de la xarxa neuronal.
- `name`: nom del model. El principal us serà per desar el model.
- `model`: objecte *Sequential* de *Keras* que emmagatzema el model.

La codificació del model es codifica com s'indica a continuació:

```
# INICI CODIFICACIO:

# 1er conjunt de capes
self.model.add(ZeroPadding2D(padding=(4, 4), input_shape=(h, w, 1)))
self.model.add(
    Conv2D(64, kernel_size=5, strides=2, padding='valid',
           kernel_initializer=keras.initializers.RandomNormal(stddev=0.01),
           bias_initializer=keras.initializers.RandomNormal(stddev=0.01),
           activation='relu'))
self.model.add(MaxPooling2D(pool_size=3, strides=2))
self.model.add(BatchNormalization())

# 2on conjunt de capes
self.model.add(ZeroPadding2D(padding=2))
self.model.add(
    Conv2D(128, kernel_size=4, strides=1, padding='valid',
           kernel_initializer=keras.initializers.RandomNormal(stddev=0.01),
           bias_initializer=keras.initializers.RandomNormal(stddev=0.01),
           activation='relu'))
self.model.add(MaxPooling2D(pool_size=3, strides=2))
self.model.add(BatchNormalization())

# 3er conjunt de capes
self.model.add(
    Conv2D(4, kernel_size=3, strides=1, padding='valid',
           kernel_initializer=keras.initializers.RandomNormal(stddev=0.01),
           bias_initializer=keras.initializers.RandomNormal(stddev=0.01),
           activation='relu'))

self.model.add(Flatten())

# FI CODIFICACIO ---

# INICI DESCODIFICACIO:

# 1ra capa completament connectada
self.model.add(
    Dense(1064, activation='sigmoid',
          kernel_initializer=keras.initializers.RandomNormal(stddev=0.01),
          bias_initializer=keras.initializers.RandomNormal(stddev=0.01)))
```

<sup>10</sup> Més informació sobre la API de *Keras* i en concret sobre *Sequential* es pot trobar a <https://keras.io/getting-started/sequential-model-guide/>

```
# 2na capa completament connectada
self.model.add(
    Dense(2048, activation='sigmoid',
         kernel_initializer=keras.initializers.RandomNormal(stddev=0.01),
         bias_initializer=keras.initializers.RandomNormal(stddev=0.01)))

# 3ra capa completament connectada
self.model.add(
    Dense(filtres_sortida, activation='sigmoid',
         kernel_initializer=keras.initializers.RandomNormal(stddev=0.01),
         bias_initializer=keras.initializers.RandomNormal(stddev=0.01)))

# FI DESCODIFICACIO---
```

## 6.4. Xarxa Neuronal

La classe *NeuralNetwork* engloba les operacions sobre la xarxa neuronal. L'atribut principal és de tipus *Sequential*, definit a la classe *Model*. Fa us de la API de *Keras* per oferir diferents operacions sobre el model.

Els atributs d'aquesta classe són:

- *model*: estructura de la xarxa neuronal de tipus *Sequential* de *Keras*.
- *name*: nom de la xarxa neuronal. Coincideix amb el nom del Model.
- *shape*: mida de l'entrada de la xarxa.
- *history*: història de l'entrenament de la xarxa.

Les operacions sobre la xarxa neuronal que ens ofereix aquesta classe són:

- *train\_neural\_network(dades\_entrenament, dades\_validacio, epoques)*: entrena la xarxa neuronal amb les dades que li proporciona el generador *dades\_entrenament*. Per validar el model usa els valors de *dades\_validacio*. El número d'èpoques que s'entrenarà la xarxa ve donat per l'argument *epoques*. Aquest mètode actualitza el camp *history* de la classe.
- *evaluate(dades\_test)*: avalua la xarxa neuronal amb les dades que obté del generador *dades\_test*.
- *predict(entrada)*: retorna la predicció de l'entrada respecte el model de la xarxa neuronal.
- *save(filename)*: desa el model al fitxer indicat a *filename*. Si *filename* no té valor, seleccionarà el valor que tingui l'atribut *name* de la classe.
- *load(filename)*: carrega la xarxa neuronal emmagatzemada al fitxer indicat per *filename*.
- *plot\_training\_history(títol, mostrar\_info)*: dibuixa la gràfica de l'entrenament, comparant pèrdua amb èpoques.

## 6.5. Generador de dades

La classe *DataGenerator* hereta de *Sequence* i proporciona un mecanisme per poder treballar amb la xarxa neuronal en un procés per lots. La metodologia aplicada serveix per a que la xarxa no processi tot el conjunt de dades sencer i treballi amb conjunt més petits. Aquests conjunts d'informació s'hauran modificat segons el que s'ha indicat en els capítols anteriors.



Aquesta classe té un conjunt d'atributs per a la gestió del procés de lots, com poden ser *batch\_size* o *data\_set*. Altres atributs, com *scale\_min*, *rotate\_max* o *tx\_max*, seran utilitzades per a les transformacions del conjunt d'imatges.

La classe *DataGenerator* ha de implementar dos mètodes: `__len__` i `__getitem__`. El primer dona com a resultat el número de lots totals a processar. El segon retornarà la informació amb la que s'entrenarà la xarxa neuronal. En aquest cas el resultat d'aquest mètode són dos *arrays*, el primer conté una imatge i el segon el HOG d'aquesta imatge. La imatge o el HOG hauran patit modificacions, tal com s'ha explicat en capítols anteriors.

## 6.6. Avaluació de la xarxa

La classe *NeuralNetworkEval* proporciona mecanismes per avaluar una xarxa neuronal. Aquesta classe té els següents atributs:

- *neural\_network*: xarxa neuronal que s'avaluarà.
- *test\_data\_set*: conjunt de dades per a l'execució de les proves.
- *batch\_size*: mida del lot en el procés.
- *hit\_ratio*: ràtio d'encert.
- *auc*: mitjana d'encert del conjunt de dades.
- *stats*: estadístiques de l'avaluació. Inclou els *true positive* (tp), *true negative* (tn), *false positive* (fp) i *false negative* (fn).
- *times*: emmagatzema el temps que triga en calcular les distàncies entre les imatges i el temps que triga en calcular la matriu de confusió.
- *matcher*: classe que permetrà indicar si dues imatges presenten encavalcament.

Les operacions que proporciona aquesta classe són:

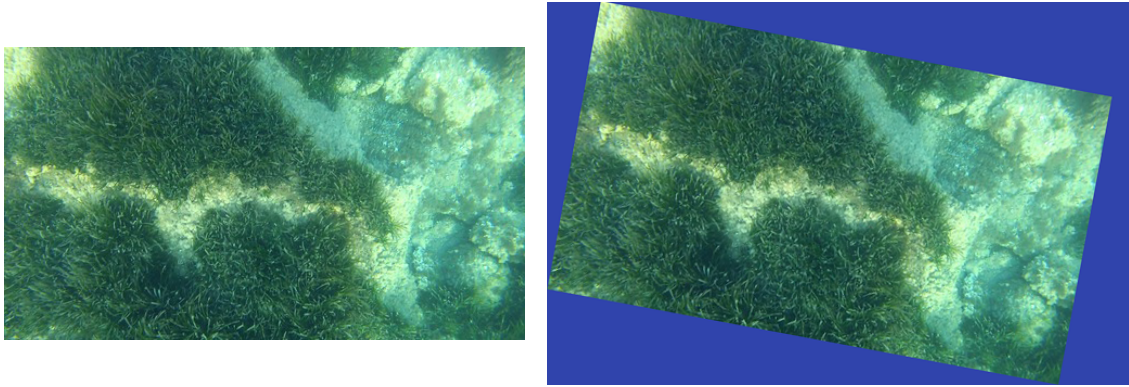
- *predict\_images(numero\_imatges, conjunt\_imatges)*: realitza prediccions sobre un conjunt d'imatges.
- *compute\_hitratio\_evolution(distancia)*: analitza el grau d'encert de la xarxa neuronal i actualitza els atributs *hit\_ratio* i *auc*.
- *plot\_hit\_ratio\_evolution()*: dibuixa una gràfica amb el percentatge d'encert sobre el conjunt de dades.
- *compute\_fullstats()*: calcula la matriu de confusió, guardant les dades als atributs *stats* i *times*.
- *check\_one\_query(query\_index)*: donada la imatge de bucle indicada a *query\_index*, retorna la posició trobada per la xarxa neuronal segons la predicció d'aquesta imatge i el corresponent bucle indicada al *dataset*.

## 6.7. Tractament de dades

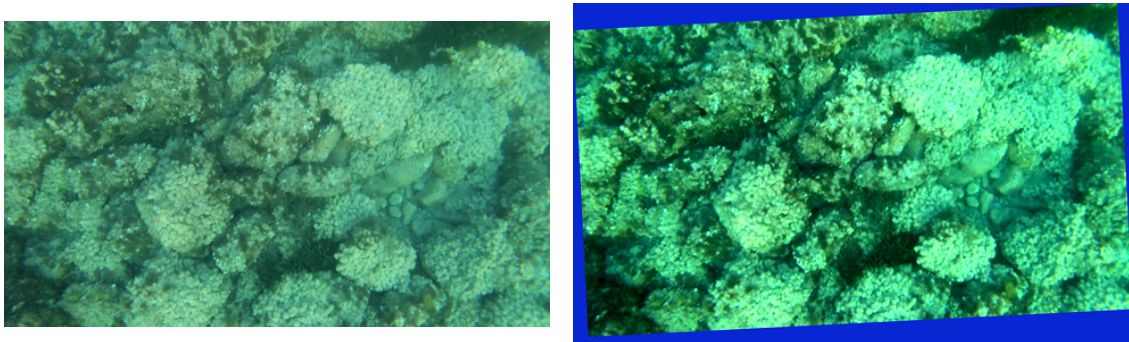
La classe *data\_generator\_utils* proporciona mètodes d'utilitats per a la classe *DataGenerator*. El mètode *generate\_pair\_images* tracta la imatge d'entrada i retorna la imatge processada i el vector HOG corresponent. La implementació d'aquestes modificacions es poden fer amb llibreries de tercers, com *skimage*, o amb una implementació pròpia. Cada imatge serà modificada de forma diferent a les anteriors.

Aleatòriament es modificaran els valors per a la translació, la rotació i el valor gamma de la imatge.

La implementació pròpia d'aquest algoritme parteix d'uns valors mínims i màxims que poden agafar les transformacions, definits com a paràmetres. Llavors, de forma aleatòria es decideix quins canvis pateix (translació, rotació i/o valor gamma) i aleatòriament es determinen els valors d'aquestes modificacions dins dels límits indicats anteriorment. Alguns exemples d'aquestes modificacions es troben en Il·lustració 34 i Il·lustració 35.



*Il·lustració 34 Imatge original (esquerra) i transformació aplicada (dreta). Ha sofert translació i rotació.*



*Il·lustració 35 Imatge original (esquerra) i transformació aplicada (dreta). Ha sofert translació, rotació i canvi del nivell gamma.*

## 7. Resultats experimentals

Aquest capítol analitza els resultats de la xarxa neuronal creada, on l'èxit o no depèn de si es detecten correctament bucles en les imatges de fons marins. Posteriorment, i donat aquest punt d'origen, es modifiquen els hiperparàmetres, arquitectura de la xarxa o el tractament de les dades prèvies per detectar possibles millores en els resultats.

### 7.1. Introducció

En l'avaluació de la xarxa neuronal, el conjunt de proves s'agrupen en escenaris, on cada escenari treballa sobre un punt específic de la xarxa o hiperparàmetre. Per cada escenari es calcularà la gràfica de pèrdua respecte el número d'èpoques en el conjunt d'entrenament i de validació. Si el valor de l'Error Absolut Mitjà (*mae*) ofereix informació rellevant s'afegirà a la gràfica.

En el primer escenari, que s'anomenarà escenari 0 o base, s'avalua la xarxa neuronal en el seu estat original. En aquest escenari es determinarà el número d'èpoques d'entrenament que seguirà la resta d'escenaris.

A l'escenari 2 s'estudiaran diferents funcions de pèrdua. L'elecció de la funció de pèrdua és important [22] ja que les xarxes neuronals aprenen a ajustar els seus paràmetres gràcies als valors retornats per aquesta funció. Segons la natura del problema a resoldre es preferible seleccionar funcions de pèrdua de regressió (MSLE, MAE o MSE), funcions de classificació binària (BCE) o de classificació multi classe (KLD). Cadascuna d'aquestes poden tenir avantatges o inconvenients sobre la resta [23].

A l'escenari 3 s'ha avaluat diferents optimitzadors. Aquests permeten [24] optimitzar els valors dels paràmetres per a reduir l'error de la xarxa. Els optimitzadors avaluats seran: RMSProp (de l'escenari base), SGD, Adagrad, Adadelta, Adam i Nadam.

L'escenari 4 analitza dues formes de normalitzar la xarxa, *Local Response Normalization* i *Batch Normalization*. Amb la normalització el que es pretén és compensar el comportament de les funcions d'activació ja que algunes no tenen un rang tancat [25].

Els escenaris 5 i 6 analitzen diverses funcions d'activació al conjunt de capes de codificació i descodificació, respectivament.

Els escenaris 7 i 8 modifiquen l'arquitectura de la xarxa. Tant a la part de codificació com de descodificació s'estudien diferents casos: es modifiquen les mides de les capes actuals, s'elimina alguna o s'afegeixen noves capes.

El darrer escenari compara diferents inicialitzacions de biaixos o nuclis. Uns valors adequats permeten obtenir millors característiques i convergir abans [26].

Finalment s'analitza el model resultant de les proves i s'avaluen diferents exemples dels resultats.

### 7.2. Criteris per l'avaluació de la xarxa neuronal

Els mecanismes per avaluar la xarxa neuronal són:

- Un algoritme que donat les imatges *query* i *database* del conjunt de dades aconsegueix la predicció d'aquestes amb la xarxa neuronal. Posteriorment per cada imatge *database*, de forma incremental en el conjunt, analitza si en aquelles imatges existeix algun bucle, indicat per *query*, i la xarxa neuronal l'ha detectat. Com a resultat d'aquest algoritme s'obtenen dades que es poden graficar. El percentatge de l'eix *X* en aquest gràfic ve determinat pel número d'imatges

*database* del conjunt. El valor *hit ratio* defineix el percentatge d'imatges que formen un bucle i són detectades com a tal en aquella porció concreta del conjunt *database* [27]. Com a valor per avaluar la viabilitat de la xarxa s'utilitza el percentatge de l'àrea total que resta per sota de la corba (indicat per AUC), com més gran és aquest valor millor serà la precisió que tindrà la xarxa neuronal. Aquest valor s'obté com la mitjana de tots els *hit ratio*. També es determina quan s'arriba al 80% d'encert com un paràmetre per avaluar la xarxa neuronal. Com més petit sigui aquest valor millor serà la xarxa.

- Es calcula la matriu de confusió, obtenint:
  - Vertaders positius o TP: s'han identificat com imatges que són un bucle i realment sí formen part d'aquest bucle.
  - Vertaders negatius o TN: s'han predit com imatges que no formen part d'un bucle i s'han identificat correctament.
  - Falsos positius o FP: s'han detectat que pertanyen a un bucle però realment no és correcte.
  - Falsos negatius o FN: la xarxa neuronal considera que no són bucle, però realment sí que ho són.

Amb aquests quatre valors (TP, TN, FP i FN) es procedeix a calcular les següents mètriques:

- Exactitud: rati de correctesa en la predicció respecte el total de les observacions. Com més gran aquest valor millor.

$$Exactitud = \frac{tp + tn}{tp + fp + fn + tn}$$

- Precisió: rati dels positius correctament identificats respecte el total de positius predits. Una alta precisió indicarà que hi ha un baix rati de falsos positius. Com més gran aquest valor millor.

$$Precisió = \frac{tp}{tp + fp}$$

- Sensibilitat o TPR: rati dels positius correctament identificats respecte al conjunt de positius reals. Com més gran aquest valor millor.

$$TPR = \frac{tp}{tp + fn}$$

- Rati de falsos positius o FPR: rati de prediccions positives incorrectes entre el total de negatius. Com més petit aquest valor millor.

$$FPR = \frac{fp}{fp + tn}$$

- Valor F1: combina els valors de la precisió i sensibilitat. Com més gran aquest valor millor.

$$Valor F1 = 2 \times \frac{Sensibilitat \times Precisió}{Sensibilitat + Precisió}$$

- Qualitativament, amb una funció que donat un bucle identificat comparará aquesta imatge amb les imatges del conjunt de dades. El resultat l'ordenarà per identificar, segons la xarxa neuronal, quines són més properes per ser bucle. A continuació identificarà la posició en la que es troba la imatge que realment és el bucle dins de la llista ordenada proporcionada per la xarxa neuronal. Com més petit sigui aquest valor més encertat serà el resultat. Si el valor està per sota o igual a 5 es considerarà un encert. La llista s'inicia en 0. El bucle identificat sempre serà el mateix en tots els escenaris. Aquesta prova s'indica com *ValidaBucle* dins dels escenaris.

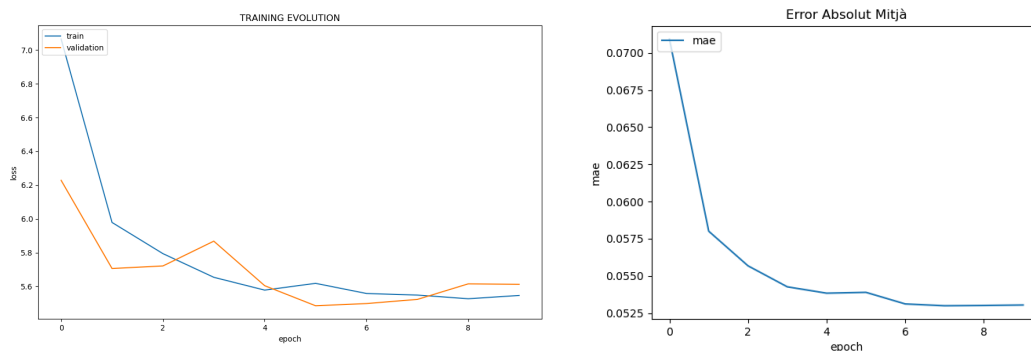
### 7.3. Cerca d'hiperparàmetres

En aquesta secció s'avalua la xarxa neuronal per identificar els hiperparàmetres que milloren els resultats originals.

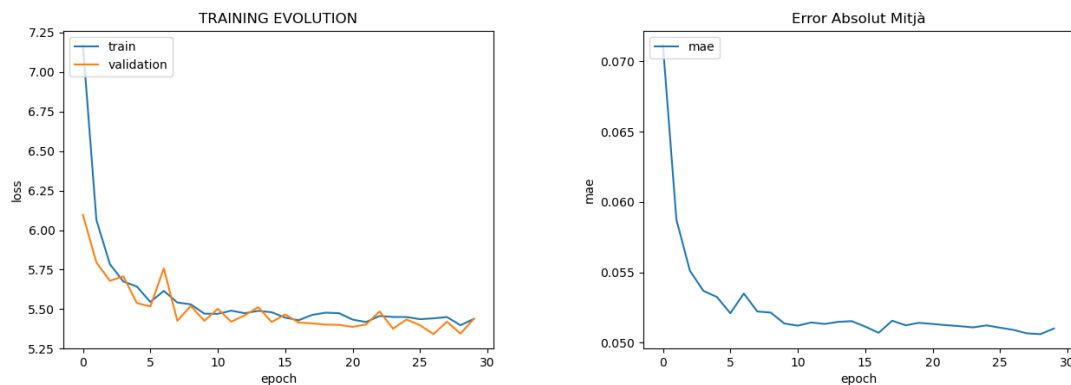
#### Escenari 0

L'escenari 0 o base correspon a la implementació comentada en capítols anteriors. Amb aquest primer escenari es determinarà el número d'èpoques adequat per fer els càlculs en la resta d'escenaris.

A continuació es mostren diversos entrenaments amb diferents èpoques per analitzar el valor òptim per calcular els següents escenaris. Les Il·lustració 36, Il·lustració 37, Il·lustració 38 i Il·lustració 39 mostren el valor de pèrdua dels conjunts d'entrenament i validació per cada època de l'entrenament, en proves amb 10, 30, 50 i 200 èpoques respectivament. La gràfica de la dreta mostra el valor de l'Error Absolut Mitjà.

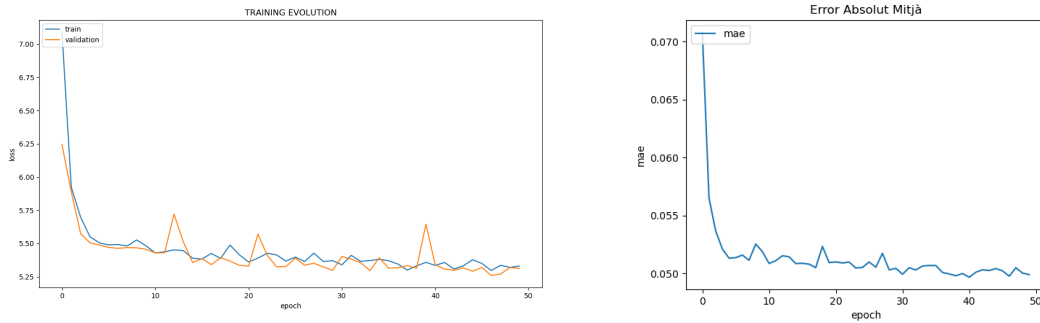


Il·lustració 36 Evolució de l'entrenament de l'escenari 0 amb 10 èpoques.

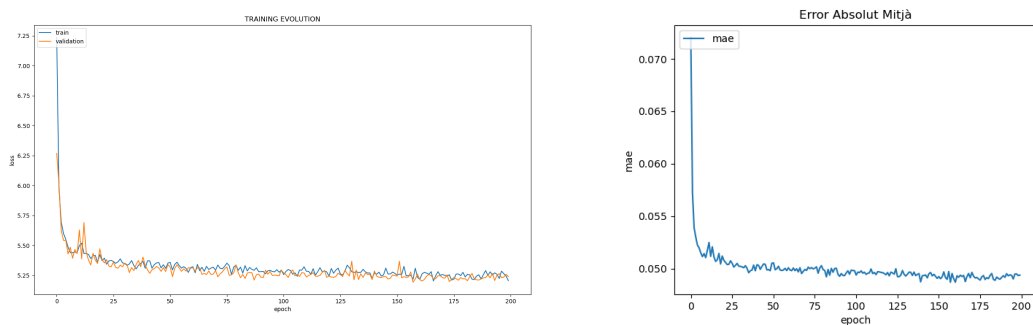


Il·lustració 37 Evolució de l'entrenament de l'escenari 0 amb 30 èpoques.

## Detecció visual de bucles en fons marins mitjançant xarxes neuronals

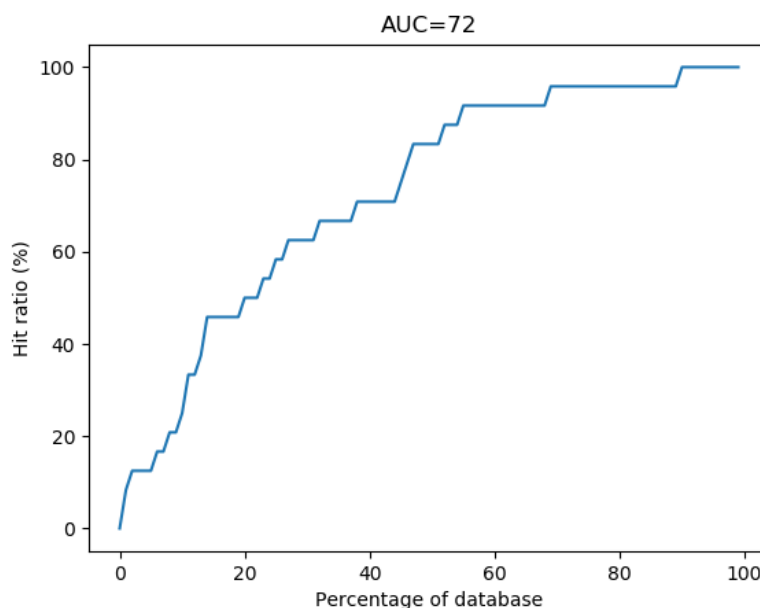


Il·lustració 38 Evolució de l'entrenament de l'escenari 0 amb 50 èpoques.



Il·lustració 39 Evolució de l'entrenament de l'escenari 0 amb 200 èpoques.

Es pot observar com fins a la quarta època el valor de la pèrdua té una pendent de baixada força pronunciada. A partir d'aquesta època va baixant de forma menys pronunciada fins a la època 10. A partir de llavors, tot i que continuar reduint-se, no varia significativament. Les gràfiques de l'Error Absolut Mitjà confirmen l'observació feta amb els valors de pèrdua. En l'entrenament dels següents escenaris, i pel següent punt d'aquest escenari, la xarxa neuronal s'entrenarà amb el valor de referència de 30 èpoques.



Il·lustració 40 Percentatge d'encert respecte la mida del conjunt d'imatges de l'escenari 0, amb 30 èpoques.

Es pot observar a la Il·lustració 40 com el percentatge d'encert va creixent i a partir del 43% està per sobre del 80% d'encert.

A la Taula 6 es troba la matriu de confusió per aquest escenari amb un entrenament de 30 èpoques. En aquesta taula es pot observar que el nombre d'imatges que no pertanyen a un bucle és significativament superior a les imatges que pertanyen a un bucle. Aquest fet fa que el valor de F1 tingui més rellevància al haver-hi tanta diferència entre les classes. A la Taula 7 es troben els resultats de les proves d'aquest escenari.

Taula 6 Matriu de confusió de l'escenari base.

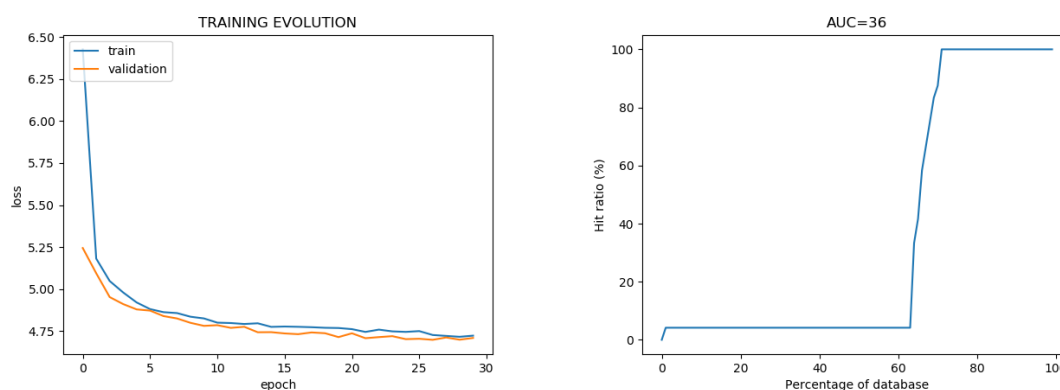
		Condicció correcta	
		Positiva	Negativa
Condicció predita	Positiva	tp: 6	fp: 10
	Negativa	fn: 20	tn: 5820

Taula 7 Avaluació de l'escenari 0.

Escenari 0	
Mitjana encert	76%
80% encert	43%
TP	6
FP	10
FN	20
TN	5820
Exactitud	0,99488
Precisió	0,37500
TPR	0,23077
FPR	0,00172
Valor F1	0,28571
ValidaBucle	0

### Escenari 1: tractament de dades

Aquest escenari varia el tractament de dades. Els valors assignats a les transformacions difereixen de l'escenari anterior i la transformació segueix una altra metodologia. L'arquitectura de la xarxa neuronal és la mateixa que l'escenari anterior. S'entrena el model i s'obtenen les gràfiques de pèrdua i d'encert indicades a Il·lustració 41.



Il·lustració 41 Evolució de l'entrenament de l'escenari 1.

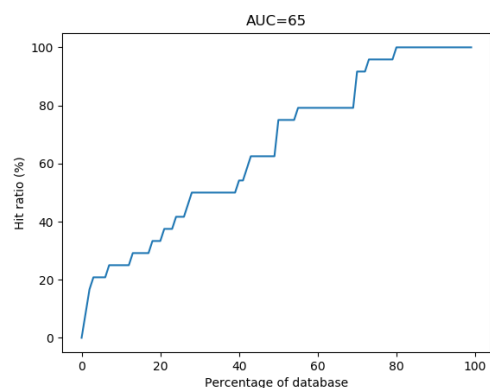
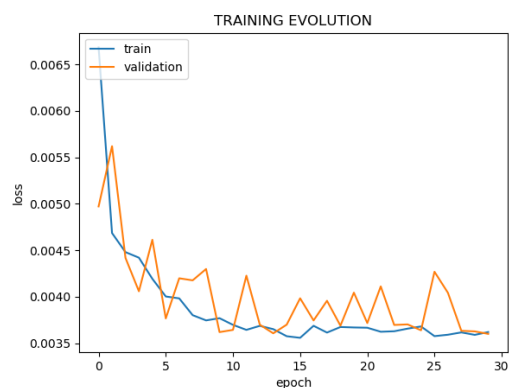
A la Taula 8 es pot comparar els resultats obtinguts en aquest escenari respecte l'escenari 0. Es consideren millors resultats els obtinguts a l'escenari base, per tant els escenaris següents tindran el 0 com a tractament de dades base.

Taula 8 Resultats de l'avaluació de l'escenari 1 i comparació amb l'escenari base.

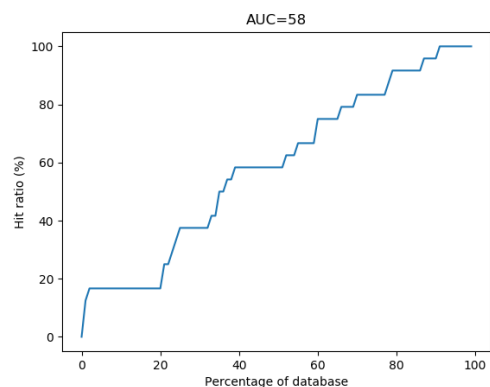
	Escenari 0	Escenari 1
<b>Mitjana encert</b>	76%	36%
<b>80% encert</b>	43%	72%
<b>TP</b>	6	0
<b>FP</b>	10	4
<b>FN</b>	20	26
<b>TN</b>	5820	5826
<b>Exactitud</b>	0,99488	0,99488
<b>Precisió</b>	0,37500	0,00000
<b>TPR</b>	0,23077	0,00000
<b>FPR</b>	0,00172	0,00069
<b>Valor F1</b>	0,28571	-
<b>ValidaBucle</b>	0	163

### Escenari 2: funcions de pèrdua

Aquest escenari té com a base l'escenari 0 i analitza diferents funcions de pèrdua sense modificar l'arquitectura de la xarxa. La funció de pèrdua de l'escenari base és la suma dels quadrats de les diferències. Les funcions de pèrdua que s'avaluen són Error Logarítmic Quadràtic Mitjà (*msle*), Error Absolut Mitjà (*mae*), Error Quadràtic Mitjà (*mse*), Entropia Creuada Binària (*bce*) i Divergència de Kullback-Leibler (*kld*). L'evolució de l'entrenament es pot observar a Il·lustració 42, Il·lustració 43, Il·lustració 44, Il·lustració 45 i Il·lustració 46.

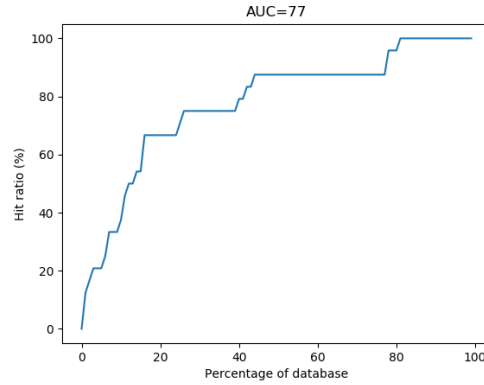
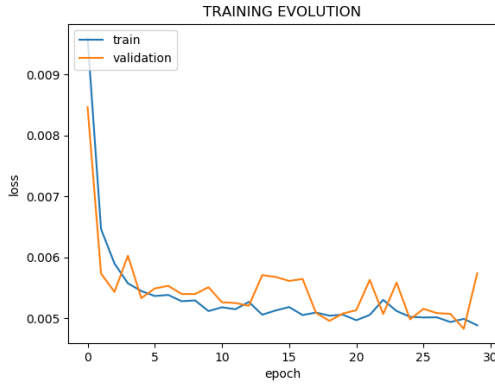


Il·lustració 42 Evolució de l'entrenament de l'escenari 2 amb pèrdua *msle*.

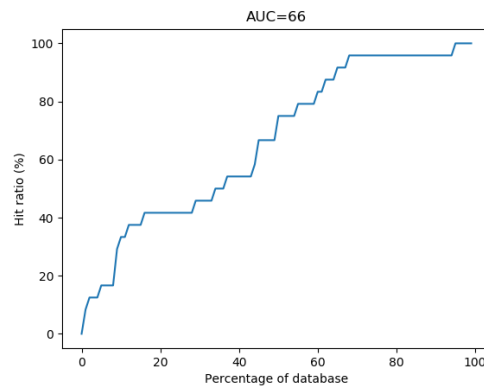
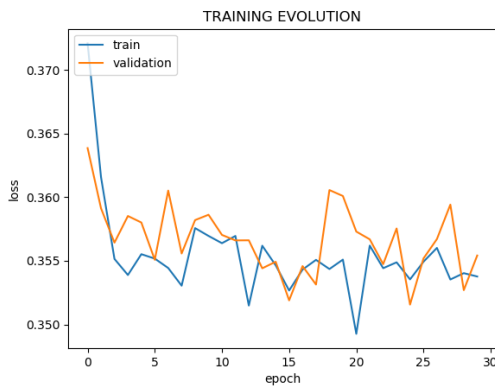


Il·lustració 43 Evolució de l'entrenament de l'escenari 2 amb pèrdua *mae*.

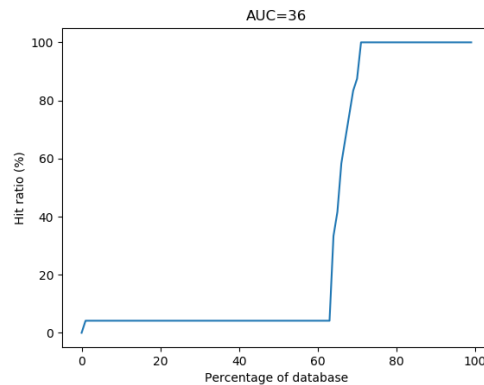
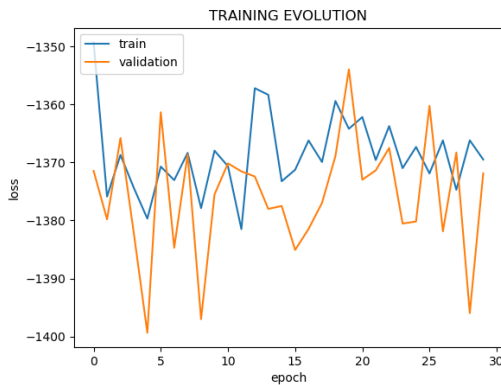




Il·lustració 44 Evolució de l'entrenament de l'escenari 2 amb pèrdua mse.



Il·lustració 45 Evolució de l'entrenament de l'escenari 2 amb pèrdua bce.

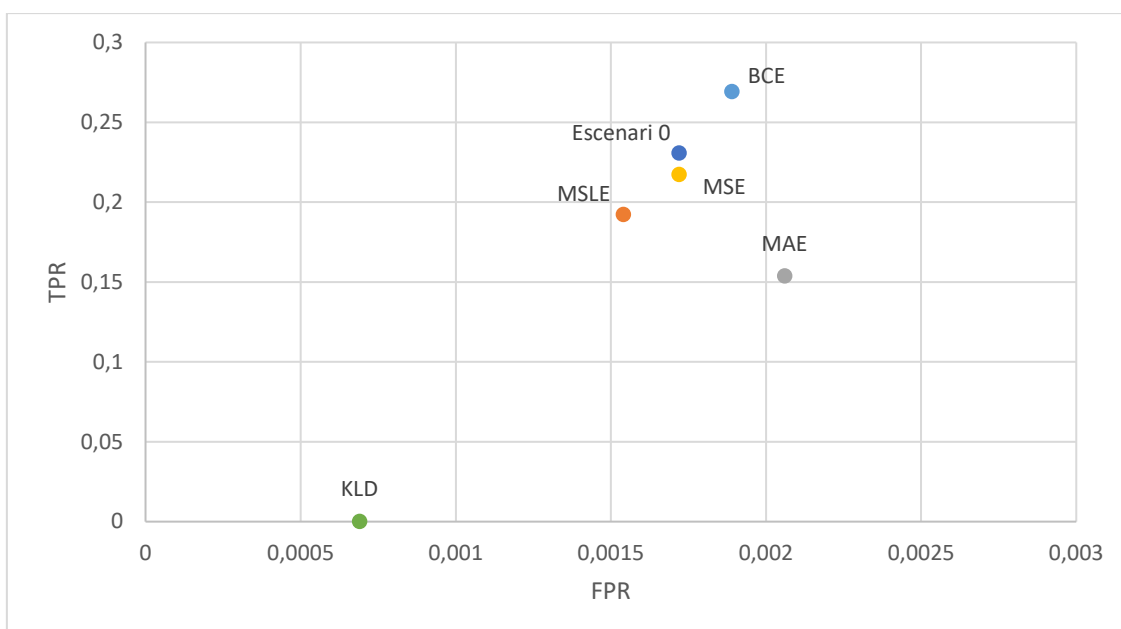


Il·lustració 46 Evolució de l'entrenament de l'escenari 2 amb pèrdua kld.

La Taula 9 mostra els resultats d'aquest escenari i els compara amb l'escenari base. A la Il·lustració 47 es troba la representació dels valors TPR respecte FPR. En aquesta gràfica els valors superiors de l'eix Y són millors que els inferiors. De l'eix X, els valors més pròxims a 0 són millors. En el cas d'aquest projecte, una valor FPR petit implica tenir un risc menor de detecció errònia de bucles amb la gravetat que això implica.

Taula 9 Comparació dels resultats de l'escenari 2 i de l'escenari base.

	Escenari 0	MSLE	MAE	MSE	BCE	KLD
<b>Mitjana encert</b>	76%	65%	58%	77%	66%	36%
<b>80% encert</b>	43%	48%	66%	42%	58%	74%
<b>TP</b>	6	5	4	5	7	0
<b>FP</b>	10	9	12	10	11	4
<b>FN</b>	20	21	22	18	19	26
<b>TN</b>	5820	5821	5818	5820	5819	5826
<b>Exactitud</b>	0,99488	0,99488	0,99419	0,99522	0,99488	0,99488
<b>Precisió</b>	0,37500	0,35714	0,25000	0,33333	0,38889	0,00000
<b>TPR</b>	0,23077	0,19231	0,15385	0,21739	0,26923	0,00000
<b>FPR</b>	0,00172	0,00154	0,00206	0,00172	0,00189	0,00069
<b>Valor F1</b>	0,28571	0,25000	0,19048	0,26316	0,31818	-
<b>ValidaBucle</b>	0	0	2	3	148	163



Il·lustració 47 Representació dels valors TPF i FPR de l'escenari 2 i base<sup>11</sup>.

Amb l'anàlisi dels valors proporcionats per la matriu de confusió es pot observar que el cas *bce* ofereix bons resultats. Millora lleugerament les dades de l'escenari 0, però a la prova d'un bucle donat (*ValidaBucle*) no obté bons resultats. La mitjana d'encert per aquest cas està per sota de l'escenari base.

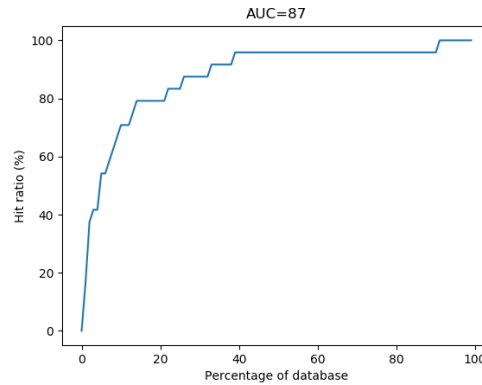
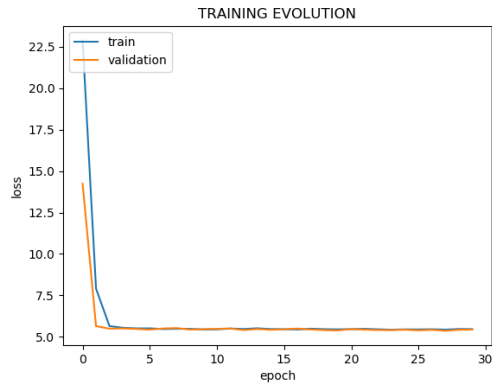
La següent millor alternativa d'aquest escenari és *mse*. Aquest cas obté bons resultats en la prova qualitativa, però els valors que obté es troben lleugerament per sota de l'escenari 0.

Analitzant aquestes dades es considera adient continuar usant la funció de pèrdua de l'escenari base.

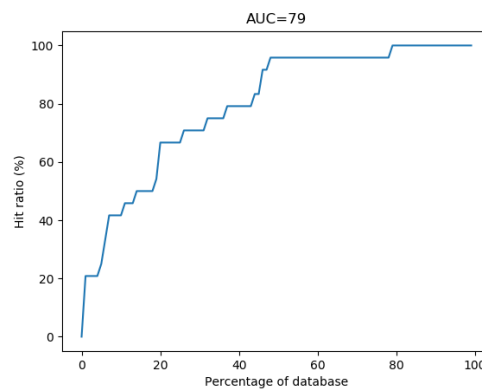
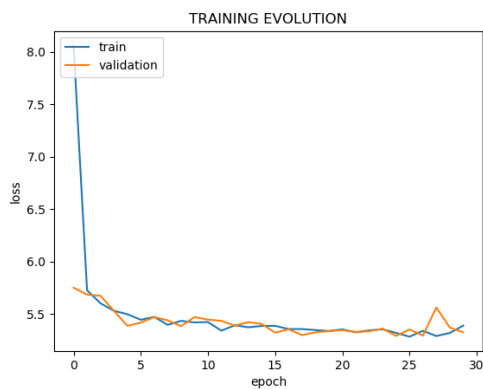
<sup>11</sup> L'escala del eixos es mostren en un interval més reduït per facilitar la visualització. L'escala de l'eix X és 100 vegades menor que la de l'eix Y.

### Escenari 3: optimitzadors

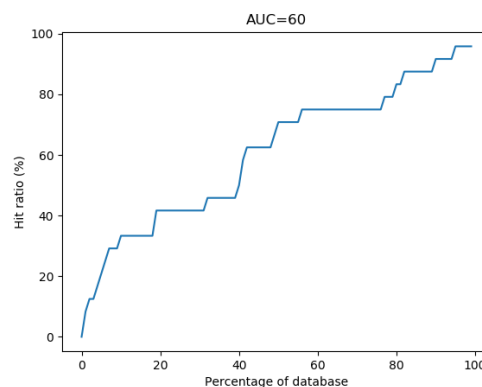
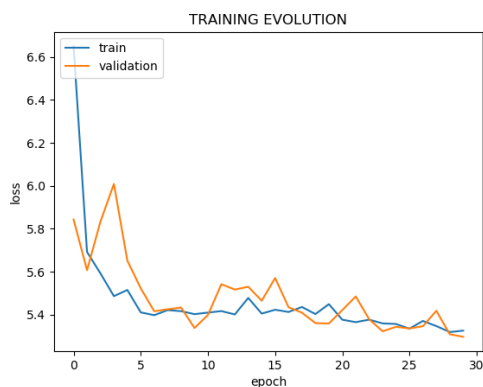
En aquest escenari s'avaluen diferents optimitzadors en la xarxa neuronal. La base de la que partim és l'escenari 0 que té com a optimitzador RMSProp. Els optimitzadors analitzats són Descens de gradient estocàstic o SGD (II·lustració 48), Adagrad (II·lustració 49), Adadelta (II·lustració 50), Adam (II·lustració 51) i Nadam (II·lustració 52).



II·lustració 48 Evolució de l'entrenament de l'escenari 3 amb optimitzador SGD.

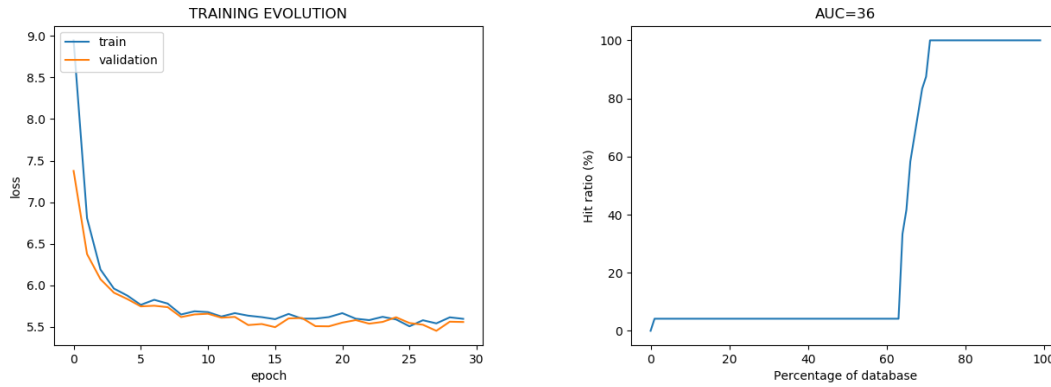


II·lustració 49 Evolució de l'entrenament de l'escenari 3 amb optimitzador Adagrad.

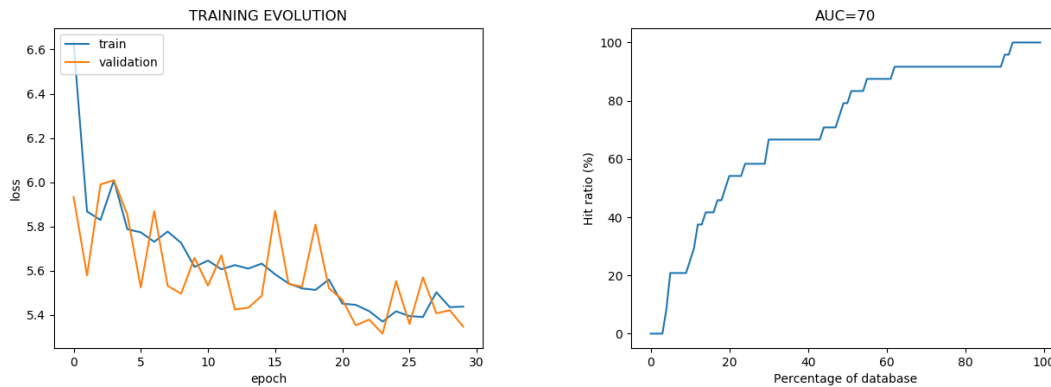


II·lustració 50 Evolució de l'entrenament de l'escenari 3 amb optimitzador Adadelta.

Detecció visual de bucles en fons marins mitjançant xarxes neuronals



Il·lustració 51 Evolució de l'entrenament de l'escenari 3 amb optimitzador Adam.

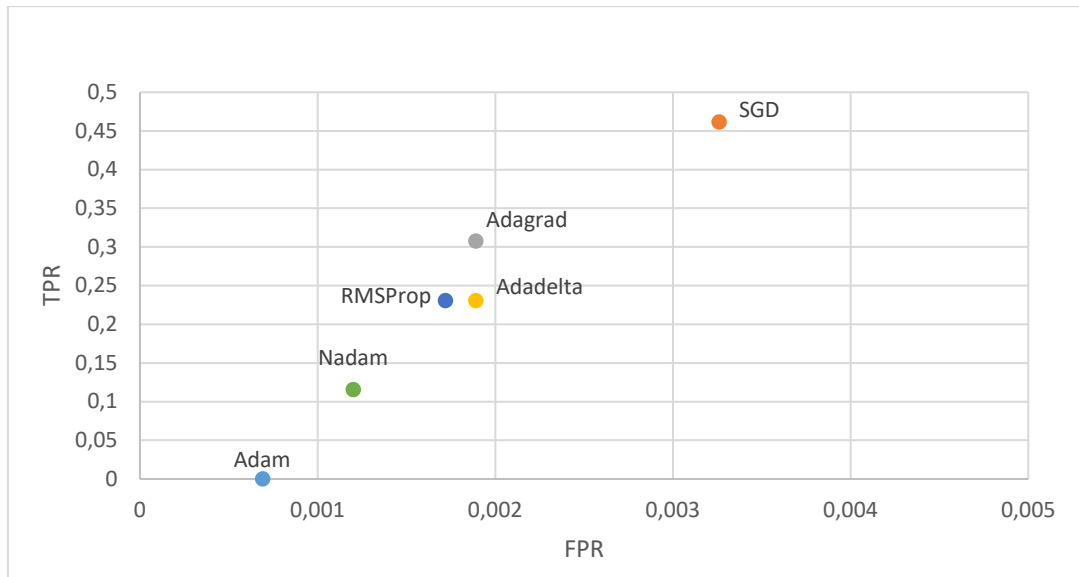


Il·lustració 52 Evolució de l'entrenament de l'escenari 3 amb optimitzador Nadam.

A la Taula 10 es troben els resultats de l'avaluació d'aquest escenari i la comparació amb l'escenari base, que té com optimitzador RMSProp. A la Il·lustració 53 es proporciona la gràfica dels valors TPR i FPR d'aquest escenari.

Taula 10 Comparació dels resultats de l'escenari 3 i de l'escenari base.

	RMSProp	SGD	Adagrad	Adadelta	Adam	Nadam
<b>Mitjana encert</b>	76%	87%	79%	60%	36%	70%
<b>80% encert</b>	43%	18%	39%	82%	73%	54%
<b>TP</b>	6	12	8	6	0	3
<b>FP</b>	10	19	11	11	4	7
<b>FN</b>	20	14	18	20	26	23
<b>TN</b>	5820	5811	5819	5819	5826	5823
<b>Exactitud</b>	0,99488	0,99436	0,99505	0,99471	0,99488	0,99488
<b>Precisió</b>	0,37500	0,38710	0,42105	0,35294	0,00000	0,30000
<b>TPR</b>	0,23077	0,46154	0,30769	0,23077	0,00000	0,11538
<b>FPR</b>	0,00172	0,00326	0,00189	0,00189	0,00069	0,00120
<b>Valor F1</b>	0,28571	0,42105	0,35556	0,27907	-	0,16667
<b>ValidaBucle</b>	0	2	1	241	163	7

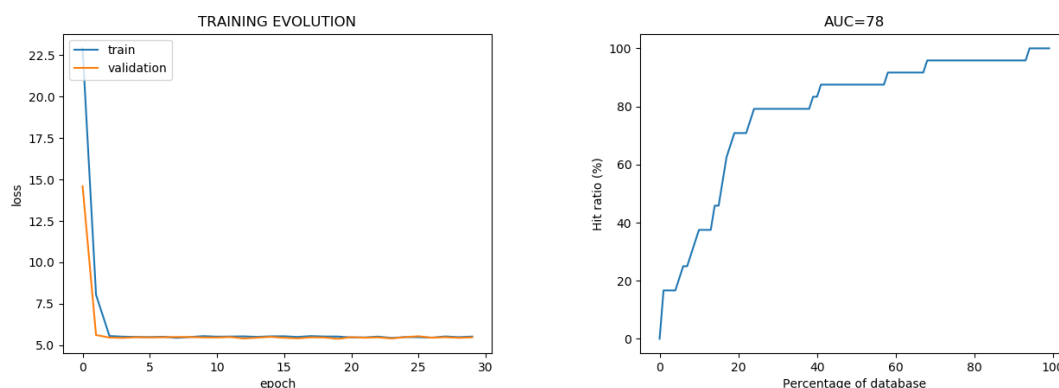


Il·lustració 53 Representació dels valors TPF i FPR de l'escenari 3 i base<sup>12</sup>.

En l'anàlisi d'aquests resultats destaca l'optimitzador SGD per sobre dels altres i ofereix en general millors resultats que l'optimitzador base RMSProp. Els següents escenaris tindran com optimitzador SGD, buscant maximitzar els resultats. Malgrat que sembli la millor opció, a l'apartat 7.4 detallarem una condició important en la resolució del problema.

#### Escenari 4: Batch Normalization

En el treball de Merrill N. i Huang G. [9], la capa de codificació té una capa *Local Response Normalization* o *LRN*. A Keras no es troba disponible aquest tipus de capa i per tots els escenaris anteriors s'ha utilitzat una capa *Batch Normalization* en el seu lloc. Aquest escenari empra una capa *LRN*<sup>13</sup> en comptes de la capa *Batch Normalization*. La Il·lustració 54 mostra l'evolució de l'entrenament.



Il·lustració 54 Evolució de l'entrenament de l'escenari 4 amb LRN.

A la Taula 11 es troba la comparació d'aquest escenari amb l'escenari base (SGD i *Batch Normalization*). Avaluant aquests resultats es pot observar que ofereix millors resultats *Batch Normalization* i serà la que tindran els següents escenaris.

<sup>12</sup> L'escala del eixos es mostren en un interval més reduït per facilitar la visualització. L'escala de l'eix X és 100 vegades menor que la de l'eix Y.

<sup>13</sup> Es recupera codi d'una versió anterior de Keras que implementava aquest tipus de capa: <https://github.com/keras-team/keras/commit/82353da4dc66bc702a74c6c233f3e16b7682f9e6#diff-e25e0456915e86c190fa1b4973c14433>

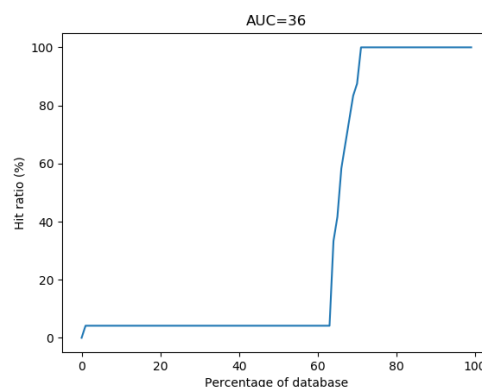
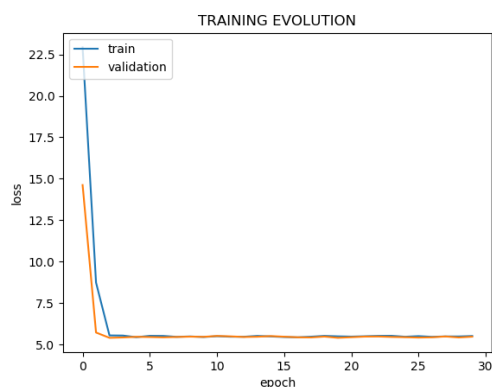
Taula 11 Comparació dels resultats de l'escenari 4 i de l'escenari base (amb optimitzador SGD).

	Batch Normalization	LRN
Mitjana encert	87%	78%
80% encert	18%	25%
TP	12	7
FP	19	14
FN	14	19
TN	5811	5816
Exactitud	0,99436	0,99436
Precisió	0,38710	0,33333
TPR	0,46154	0,26923
FPR	0,00326	0,00240
Valor F1	0,42105	0,29787
ValidaBucle	2	31

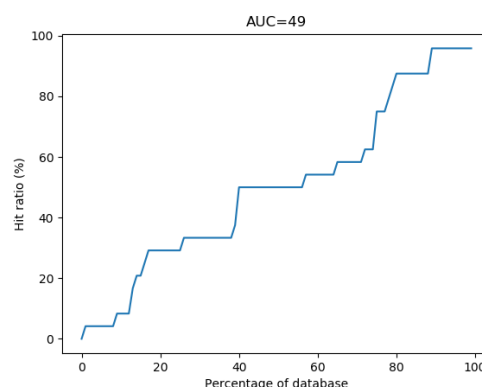
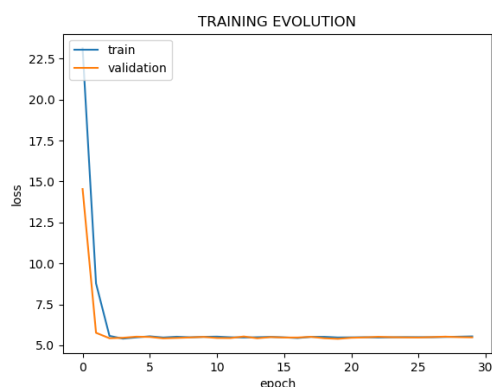
### Escenari 5: funcions activació codificació

En aquest escenari s'avaluen les funcions d'activació de les capes de convolució. En les proves anteriors aquesta funció d'activació era la Unitat Lineal Rectificada o ReLU.

Per aquest escenari s'avaluaran les funcions d'activació (per a les capes de convolució) Tangent Hiperbòlica o tanh (Il·lustració 55), Sigmoide (Il·lustració 56), Leaky ReLU (Il·lustració 57) i Swish<sup>14</sup> (Il·lustració 58).

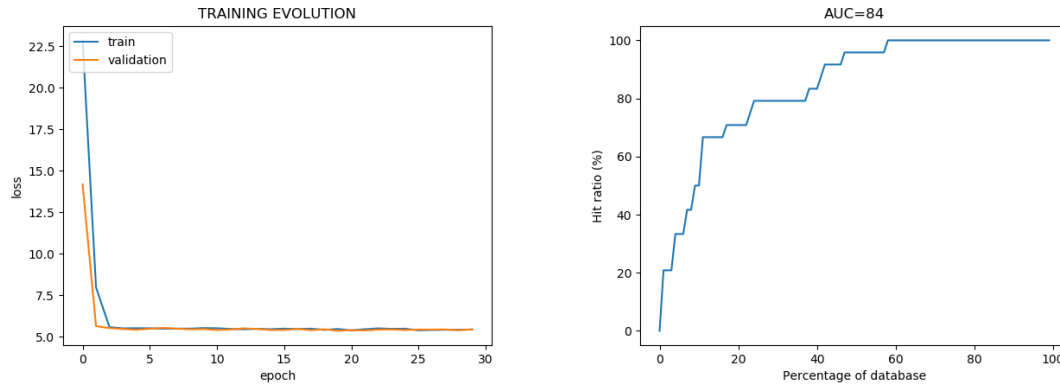


Il·lustració 55 Evolució de l'entrenament de l'escenari 5 amb tanh.

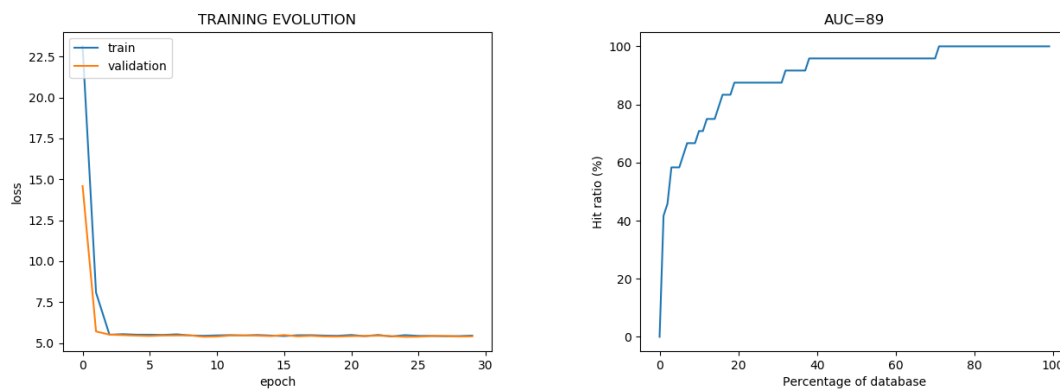


Il·lustració 56 Evolució de l'entrenament de l'escenari 5 amb Sigmoide.

<sup>14</sup> Per a la implementació Swish s'ha fet us de <https://lukasbrausch.wordpress.com/2019/10/29/adding-the-swish-activation-function-to-your-keras-model/>.



Il·lustració 57 Evolució de l'entrenament de l'escenari 5 amb LEakyReLU.

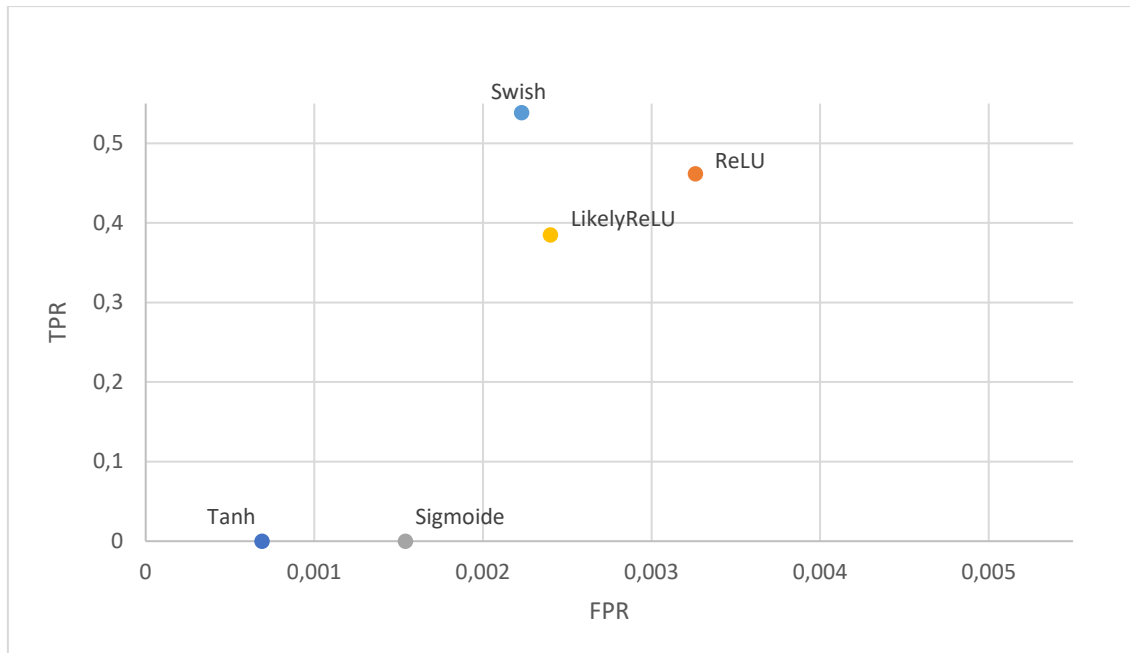


Il·lustració 58 Evolució de l'entrenament de l'escenari 5 amb Swish.

A la Taula 12 es troben els resultats de les proves fetes en aquest escenari i es comparen amb l'escenari base (ReLU). A la Il·lustració 59 es pot trobar la gràfica amb els valors TPR i FPR d'aquest escenari.

Taula 12 Comparació dels resultats de l'escenari 5 i de l'escenari base (amb ReLU).

	ReLU	Tanh	Sigmoide	L-ReLU	Swish
<b>Mitjana encert</b>	87%	36%	50%	84%	90%
<b>80% encert</b>	18%	68%	81%	24%	16%
<b>TP</b>	12	0	0	10	14
<b>FP</b>	19	4	9	14	13
<b>FN</b>	14	26	26	16	12
<b>TN</b>	5811	5826	5821	5816	5817
<b>Exactitud</b>	0,99436	0,99488	0,99402	0,99488	0,99573
<b>Precisió</b>	0,38710	0,00000	0,00000	0,41667	0,51852
<b>TPR</b>	0,46154	0,00000	0,00000	0,38462	0,53846
<b>FPR</b>	0,00326	0,00069	0,00154	0,00240	0,00223
<b>Valor F1</b>	0,42105	-	-	0,40000	0,52830
<b>ValidaBucle</b>	2	163	31	0	1

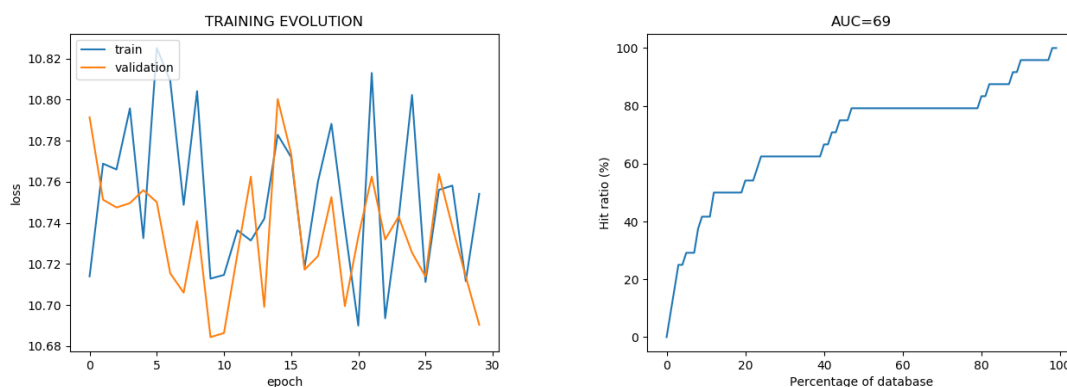


Il·lustració 59 Representació dels valors TPF i FPR de l'escenari 5 i base<sup>15</sup>.

Analizant aquests resultats es pot observar que la funció d'activació de l'escenari base, ReLU, té un comportament similar al de la funció Likely ReLU, però la funció d'activació Swish ofereix millors resultats en tots els paràmetres estudiats. En els escenaris següents s'emprarà aquesta funció d'activació per a les capes de convolució.

### Escenari 6: funcions activació descodificació

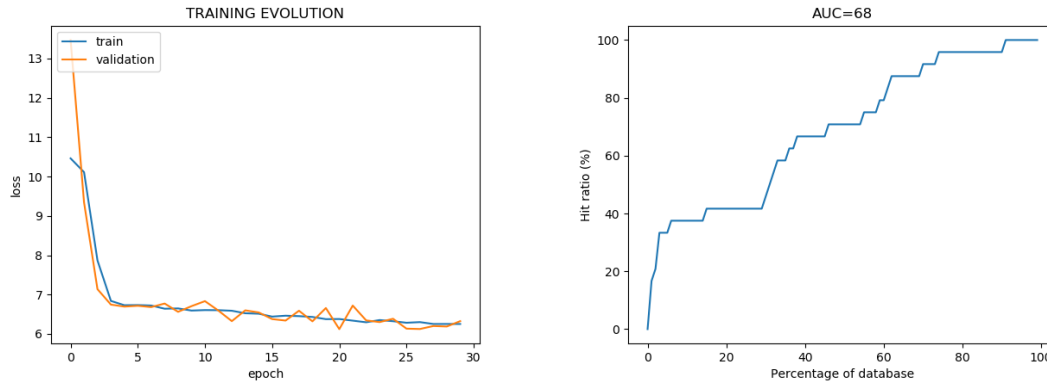
En aquest escenari s'analiza com afecta el canvi de la funció d'activació a les capes de descodificació. Fins a aquest moment la funció d'activació d'aquestes capes era la Sigmoid. En aquest cas s'avaluaran Softmax (Il·lustració 60), Tangencial Hiperbòlica o tanh (Il·lustració 61) i Lineal (Il·lustració 62).



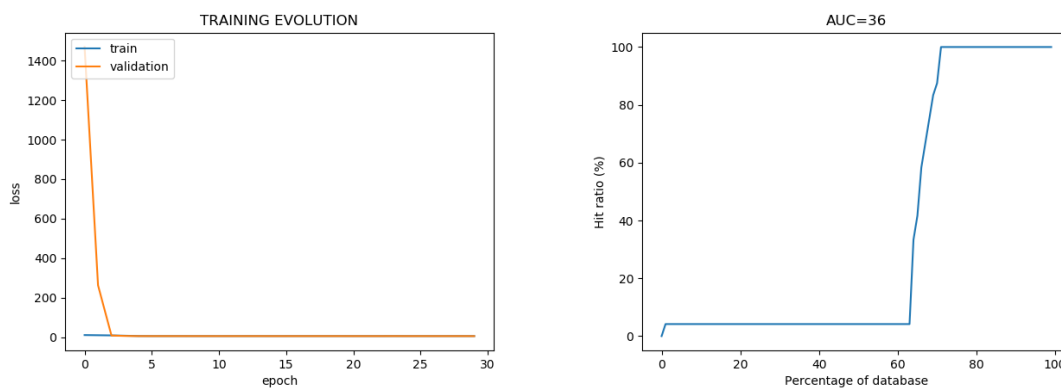
Il·lustració 60 Evolució de l'entrenament de l'escenari 6 amb Softmax.

<sup>15</sup> L'escala del eixos es mostrem en un interval més reduït per facilitar la visualització. L'escala de l'eix X és 100 vegades menor que la de l'eix Y.





Il·lustració 61 Evolució de l'entrenament de l'escenari 6 amb tanh.



Il·lustració 62 Evolució de l'entrenament de l'escenari 6 amb Lineal.

A la Taula 13 es poden trobar els resultats d'aquest escenari i els de l'escenari base, amb funció d'activació Sigmoide. Es pot observar com l'escenari base ofereix millors resultats, per tant no es modifica aquesta funció d'activació que serà la mateixa en els següents escenaris.

Taula 13 Comparació dels resultats de l'escenari 6 i de l'escenari base (amb Sigmoide).

	Sigmoide	Softmax	Tanh	Lineal
<b>Mitjana encert</b>	90%	69%	68%	36%
<b>80% encert</b>	16%	47%	60%	69%
<b>TP</b>	14	7	7	0
<b>FP</b>	13	11	11	4
<b>FN</b>	12	19	19	26
<b>TN</b>	5817	5819	5819	5826
<b>Exactitud</b>	0,99573	0,99488	0,99488	0,99488
<b>Precisió</b>	0,51852	0,38889	0,38889	0,00000
<b>TPR</b>	0,53846	0,26923	0,26923	0,00000
<b>FPR</b>	0,00223	0,00189	0,00189	0,00069
<b>Valor F1</b>	0,52830	0,31818	0,31818	-
<b>ValidaBucle</b>	1	106	1	163

### Escenari 7: estructura codificació

En aquest escenari es modifica l'estructura de les capes de codificació de la xarxa. Originalment aquesta part de la xarxa està composta, principalment, per tres capes convolucionals amb una mida de 64, 128 i 4 respectivament. Els resultats de l'escenari base es troben a la Taula 14.

Taula 14 Resultats de l'escenari base.

Base(A64B128C4)	
<b>Mitjana encert</b>	90%
<b>80% encert</b>	16%
<b>TP</b>	14
<b>FP</b>	13
<b>FN</b>	12
<b>TN</b>	5817
<b>Exactitud</b>	0,99573
<b>Precisió</b>	0,51852
<b>TPR</b>	0,53846
<b>FPR</b>	0,00223
<b>Valor F1</b>	0,52830
<b>ValidaBucle</b>	1
<b>Mida</b>	152

En els resultats s'afegeix una nova fila per indicar la mida en Megabytes de la xarxa en un fitxer. A les taules s'indica amb una *A* si la modificació és a la primera capa, *B* per a la segona i *C* per a la tercera. El número que s'indica a continuació de la capa és la mida que tindrà en les proves. Si no figura la capa vol dir que manté la mida original. Si s'indica una *X* significa que la capa s'elimina. Si s'afegeix una nova capa s'indica amb *D* a la posició on s'afegeixi, seguit de la seva mida.

Taula 15 Comparació dels resultats de l'escenari 7, part I.

	A256	A128	A32	B256	C8	C16	C32	C64
<b>Mitjana encert</b>	68%	78%	86%	82%	87%	82%	73%	87%
<b>80% encert</b>	59%	43%	21%	31%	19%	24%	44%	16%
<b>TP</b>	7	10	13	10	13	11	7	13
<b>FP</b>	11	13	15	12	19	14	20	12
<b>FN</b>	19	16	13	16	13	15	19	13
<b>TN</b>	5819	5817	5815	5818	5811	5816	5810	5818
<b>Exac.</b>	0,99488	0,99505	0,99522	0,99522	0,99454	0,99505	0,99334	0,99573
<b>Precis.</b>	0,38889	0,43478	0,46429	0,45455	0,40625	0,44000	0,25926	0,52000
<b>TPR</b>	0,26923	0,38462	0,50000	0,38462	0,50000	0,42308	0,26923	0,50000
<b>FPR</b>	0,00189	0,00223	0,00257	0,00206	0,00326	0,00240	0,00343	0,00206
<b>F1</b>	0,31818	0,40816	0,48148	0,41667	0,44828	0,43137	0,26415	0,50980
<b>Valida Bucle</b>	1	1	0	0	0	2	2	2
<b>Mida</b>	155	153	151	153	187	258	399	683

Taula 16 Comparació dels resultats de l'escenari 7, part II.

	A32C34	A128B2 56C16	A512B2 56C128	A128Bx C32	A128Bx C64	A64B12 8D32C4	A64B12 8D64C4	A64B12 8D8C4
<b>Mitjana encert</b>	82%	82%	74%	73%	71%	61%	80%	79%
<b>80% encert</b>	44%	40%	50%	58%	57%	60%	33%	31%
<b>TP</b>	11	9	6	9	8	5	8	7
<b>FP</b>	15	12	14	14	11	10	12	12
<b>FN</b>	15	17	20	18	18	21	18	19
<b>TN</b>	5815	5818	5816	5816	5819	5820	5818	5818
<b>Exac.</b>	0,99488	0,99505	0,99419	0,99454	0,99505	0,99471	0,99488	0,99471
<b>Precis.</b>	0,42308	0,42857	0,30000	0,39130	0,42105	0,33333	0,40000	0,36842
<b>TPR</b>	0,42308	0,34615	0,23077	0,33333	0,30769	0,19231	0,30769	0,26923
<b>FPR</b>	0,00257	0,00206	0,00240	0,00240	0,00189	0,00172	0,00206	0,00206
<b>F1</b>	0,42308	0,38298	0,26087	0,36000	0,35556	0,24390	0,34783	0,31111
<b>Valida Bucle</b>	2	1	3	15	1	238	5	74
<b>Mida</b>	682	255	1.200	1.250	2.400	121	122	121

A les Taula 15 i Taula 16 es poden trobar els resultats d'aquest escenari. Avaluant aquests resultats es pot considerar que deixar la configuració original de les capes de convolució és la millor opció.

### Escenari 8: estructura descodificació

Aquest escenari analitza modificacions en l'arquitectura de descodificació. Aquesta part està formada per tres capes completament connectades (Dense). A l'escenari base la primera capa té una mida de 1064, la segona 2048 i la tercera té una mida definida pels filtres de sortida.

Aquest escenari variarà les mides d'aquestes capes (les dues primeres), o eliminarà existents o afegirà altres capes. Per indicar la capa que es modifica s'indica amb A si es la primera i B la segona. A continuació un número indica la mida de la prova. Es marca amb X si s'ha eliminat aquella capa i amb una D si s'afegeix una capa.

Taula 17 Comparació dels resultats de l'escenari 8, part I.

	A2048	A4096	A532	B4096	B8192	B1024	Ax	Bx
<b>Mitjana encert</b>	81%	72%	77%	87%	78%	87%	86%	83%
<b>80% encert</b>	43%	51%	40%	20%	50%	24%	22%	20%
<b>TP</b>	9	5	8	12	10	14	11	10
<b>FP</b>	14	13	14	16	14	9	15	14
<b>FN</b>	17	21	18	14	16	12	15	16
<b>TN</b>	5816	5817	5816	5814	5816	5821	5815	5816
<b>Exac.</b>	0,99471	0,99419	0,99454	0,99488	0,99488	0,99641	0,99488	0,99488
<b>Precis.</b>	0,39130	0,27778	0,36364	0,42857	0,41667	0,60870	0,42308	0,41667
<b>TPR</b>	0,34615	0,19231	0,30769	0,46154	0,38462	0,53846	0,42308	0,38462
<b>FPR</b>	0,00240	0,00223	0,00240	0,00274	0,00240	0,00154	0,00257	0,00240
<b>F1</b>	0,36735	0,22727	0,33333	0,44444	0,40000	0,57143	0,42308	0,40000
<b>Valida Bucle</b>	2	38	3	2	2	1	2	3
<b>Mida</b>	195	294	122	260	486	92	168	88

Taula 18 Comparació dels resultats de l'escenari 8, part II.

	A2048B1024	A2048B4096	A512B4096	A512B1024	AD4096B	AD512B
<b>Mitjana encert</b>	82%	84%	82%	82%	87%	74%
<b>80% encert</b>	28%	30%	42%	33%	20%	56%
<b>TP</b>	12	10	13	7	12	9
<b>FP</b>	12	17	14	11	15	11
<b>FN</b>	14	16	13	19	14	17
<b>TN</b>	5818	5813	5816	5819	5815	5819
<b>Exactitud</b>	0,99556	0,99436	0,99539	0,99488	0,99505	0,99522
<b>Precisió</b>	0,50000	0,37037	0,48148	0,38889	0,44444	0,45000
<b>TPR</b>	0,46154	0,38462	0,50000	0,26923	0,46154	0,34615
<b>FPR</b>	0,00206	0,00292	0,00240	0,00189	0,00257	0,00189
<b>Valor F1</b>	0,48000	0,37736	0,49057	0,31818	0,45283	0,39130
<b>Valida Bucle</b>	0	2	1	0	2	1
<b>Mida</b>	131	323	225	68	235	147

A les Taula 17 i Taula 18 es poden trobar els resultats de l'escenari 8. A la Taula 19 es troba una comparació dels millors resultats d'aquest escenari comparant-los amb l'escenari base. En aquesta taula es pot observar que els valors són molt semblants. En percentatge d'encert l'escenari base és millor, però el cas B1024 té una millor precisió i valor F1. També destaca que la mida del cas B1024 és aproximadament un 60% la mida del cas Base. Amb aquestes dades es considera adient canviar per l'opció B1024 (la mida de la segona capa serà de 1024) en els futurs escenaris.

Taula 19 Comparació dels resultats de l'escenari 8, part III.

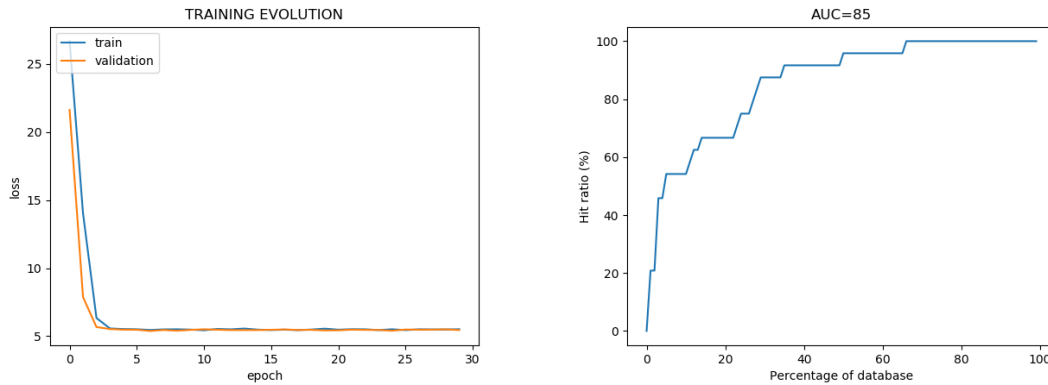
	Base(A1064B2048)	B4096	B1024	AD4096B
<b>Mitjana encert</b>	90%	87%	87%	87%
<b>80% encert</b>	16%	20%	24%	20%
<b>TP</b>	14	12	14	12
<b>FP</b>	13	16	9	15
<b>FN</b>	12	14	12	14
<b>TN</b>	5817	5814	5821	5815
<b>Exactitud</b>	0,99573	0,99488	0,99641	0,99505
<b>Precisió</b>	0,51852	0,42857	0,60870	0,44444
<b>TPR</b>	0,53846	0,46154	0,53846	0,46154
<b>FPR</b>	0,00223	0,00274	0,00154	0,00257
<b>Valor F1</b>	0,52830	0,44444	0,57143	0,45283
<b>ValidaBucle</b>	1	2	1	2
<b>Mida</b>	152	260	92	235

### Escenari 9: inicialitzacions

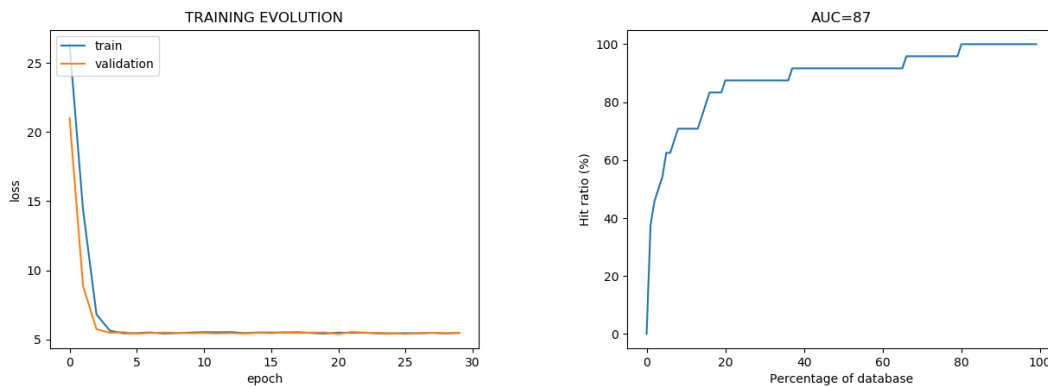
Aquest escenari analitza diferents inicialitzacions de biaixos o nucli (*kernel*) de les capes. A l'escenari base totes les capes tenen com a inicialització una distribució Normal. Les proves d'aquest escenari són:

- **bias\_zeros**: canviar les inicialitzacions del biaix a zeros (Il·lustració 63).
- **Kglorot\_Bzeros**: usar pel nucli Glorot uniforme i pels biaixos zeros (Il·lustració 64).

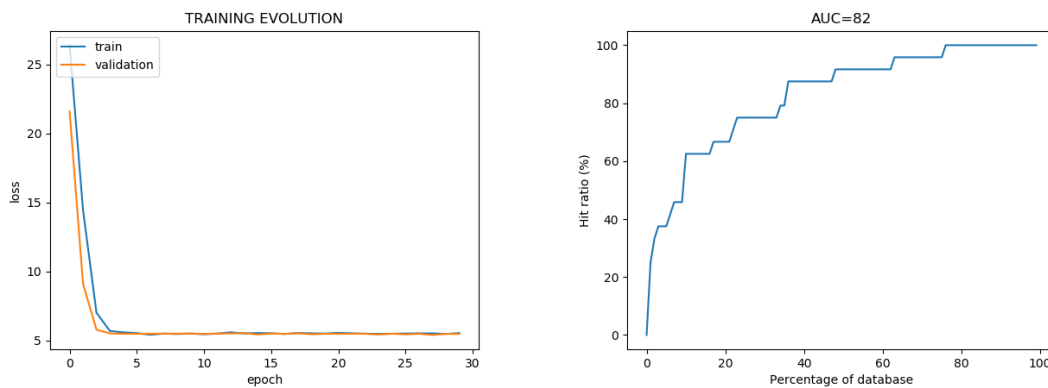
- **Kglorot\_Bnormal**: emprar pel nucli Glorot uniforme i pels biaixos distribució Normal (Il·lustració 65).
- **he\_glorot**: Normal He per als nuclis de les capes convolucionals i Glorot per als nuclis de les capes completament connectades (Il·lustració 66).
- **he\_glorot\_Bzeros**: Normal He per als nuclis de les capes convolucionals, Glorot per als nuclis de les capes completament connectades i zeros per als biaixos (Il·lustració 67).



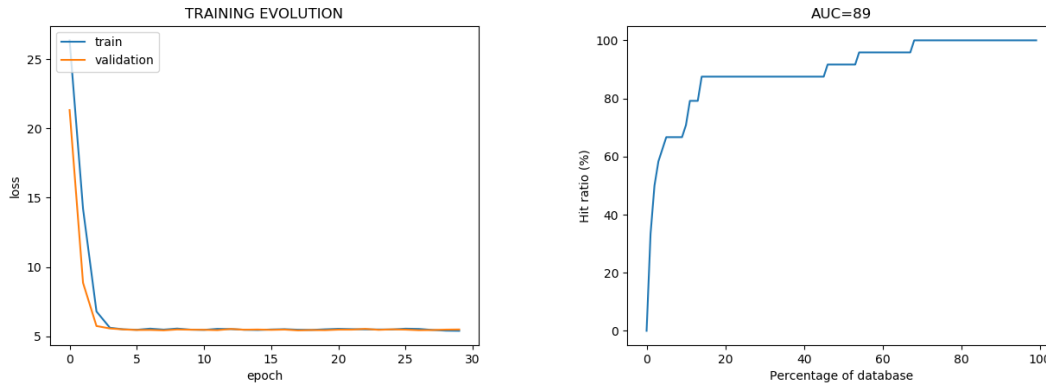
Il·lustració 63 Evolució de l'entrenament de l'escenari 9 pel cas bias\_zeros.



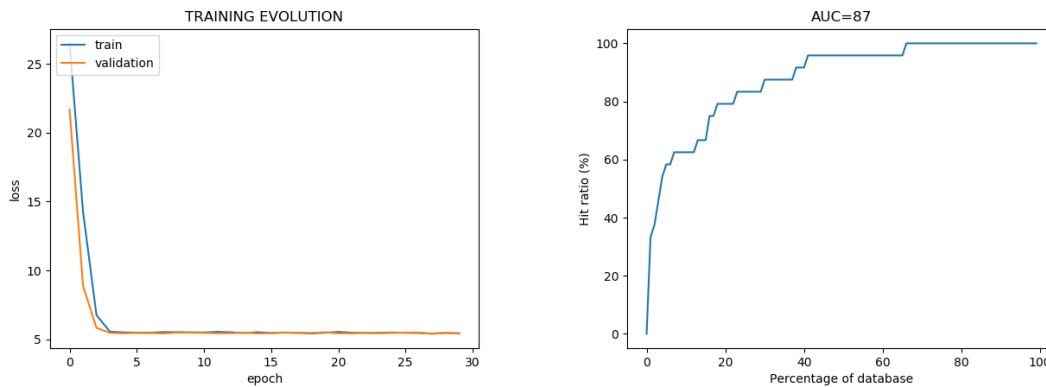
Il·lustració 64 Evolució de l'entrenament de l'escenari 9 pel cas Kglorot\_Bzeros.



Il·lustració 65 Evolució de l'entrenament de l'escenari 9 pel cas Kglorot\_Bnormal.



Il·lustració 66 Evolució de l'entrenament de l'escenari 9 pel cas he\_glorot.



Il·lustració 67 Evolució de l'entrenament de l'escenari 9 pel cas he\_glorot\_Bzeros.

A la Taula 20 es troben els resultats d'aquest escenari i es comparen amb l'escenari base, amb distribució Normal. L'opció **he\_glorot** ofereix bons resultats, però en general els valors de la distribució Normal són millors.

Taula 20 Comparació dels resultats de l'escenari 9 i de l'escenari base (Normal).

	Normal	bias_zeros	Kglorot _Bzeros	Kglorot _Bnormal	he_glorot	he_glorot _Bzeros
<b>Mitjana encert</b>	87%	85%	88%	75%	89%	80%
<b>80% encert</b>	24%	27%	15%	35%	13%	22%
<b>TP</b>	14	11	14	11	14	12
<b>FP</b>	9	9	19	17	13	18
<b>FN</b>	12	15	12	15	12	14
<b>TN</b>	5821	5821	5811	5813	5817	5812
<b>Exactitud</b>	0,99641	0,99590	0,99471	0,99454	0,99573	0,99454
<b>Precisió</b>	0,60870	0,55000	0,42424	0,39286	0,51852	0,40000
<b>TPR</b>	0,53846	0,42308	0,53846	0,42308	0,53846	0,46154
<b>FPR</b>	0,00154	0,00154	0,00326	0,00292	0,00223	0,00309
<b>Valor F1</b>	0,57143	0,47826	0,47458	0,40741	0,52830	0,42857
<b>ValidaBucle</b>	1	0	1	2	2	1

#### 7.4. Resum dels escenaris

Al llarg d'aquests escenaris s'han executat diferents proves buscant millorar els valors obtinguts de la xarxa neuronal. Amb els resultats de l'escenari 2, on s'han estudiat diferents funcions de pèrdua, els valors indiquen que la funció de pèrdua original, indicada al capítol 0, obté millors efectes que les altres opcions.

En l'escenari 3, on s'han avaluat diferents optimitzadors, els resultats obtinguts, de forma general, determinen que l'optimitzador SGD ofereix una millor solució que l'original RMSProp. Si en comptes de mirar el problema de forma general s'avalua de forma concreta, el problema de la detecció de bucles en fons marins, en un procés de detecció de bucle un fals positiu pot ser catastròfic per a l'algoritme SLAM i per aquest motiu el valor de FPR és important. Tot i ser valors del FPR molt petits en tots els casos analitzats, la diferència entre SGD i el segon millor, Adagrad, respecte FPR és que és gairebé un 60% més petit Adagrad que SGD, oferint millors resultats en aquesta mètrica.

Al llarg dels altres escenaris les proves han intentant maximitzar el resultat general. Al final d'aquesta secció es comparará el resultat obtingut usant els optimitzadors SGD i Adagrad sobre el mateix model de xarxa.

A l'escenari 4, on s'han avaluat dues formes de normalitzar la xarxa, la normalització original, *Batch Normalization*, ofereix millors resultats.

Les funcions d'activació estudiades als escenaris 5 i 6 mostren que en la part de codificació, que originalment feia us de ReLU (utilitzades habitualment en les capes de tipus convolucionals [28]), Swish millora els seus resultats. Aquest fet succeeix de forma general [29] [30]. A la part de la descodificació no s'ha millorat els resultats obtinguts per la funció Sigmoide original.

En la modificació de l'arquitectura del model, avaluada als escenaris 7 i 8, les modificacions en la part de codificació no milloren els resultats respecte l'original. En canvi, en la part de descodificació els canvis analitzats donen uns resultats similars, destacant un cas que millora lleugerament l'original i redueix la mida del fitxer, que permetrà un millor us de l'algoritme en ús *online*.

Les proves executades a l'escenari 9, comparant diferents inicialitzacions de biaixos o nuclis, no milloren l'original.

Després d'analitzar totes les modificacions comentades anteriorment, es procedeix a comparar el model final actual, amb optimitzador SGD, i el model amb optimitzador Adagrad.

Taula 21 Comparació de l'optimitzador SGD i Adagrad.

	SGD	Adagrad
<b>Mitjana encert</b>	87%	67%
<b>80% encert</b>	24%	56%
<b>TP</b>	14	5
<b>FP</b>	9	7
<b>FN</b>	12	21
<b>TN</b>	5821	5823
<b>Exactitud</b>	0,99641	0,99522
<b>Precisió</b>	0,60870	0,41667
<b>TPR</b>	0,53846	0,19231
<b>FPR</b>	0,00154	0,00120
<b>Valor F1</b>	0,57143	0,26316
<b>ValidaBucle</b>	1	30

La Taula 21 ofereix els resultats d'aquesta avaluació. Indiquen que la millor opció és mantenir l'optimitzador SGD, concretament els valors de TPR i F1 són significativament superiors en SGD. En canvi el valor de FPR son similars en tots dos casos.

### 7.5. Avaluació qualitativa

En aquesta secció s'avalua el comportament de la xarxa neuronal amb exemples concrets del conjunt de dades número 3.

Aquest conjunt de dades indica, entre altres, que l'arxiu *query* "153\_614.jpg" (Il·lustració 68) forma un bucle amb la imatge "153\_623.jpg" (Il·lustració 69), i la xarxa neuronal detecta correctament aquest bucle (TP). La superposició d'aquest bucle es pot observar a la Il·lustració 70. En canvi la xarxa neuronal identifica com a bucle el fitxer "673.jpg" (Il·lustració 71), amb el mateix arxiu *query*, però la informació del conjunt de dades ens confirma que no és correcte aquesta detecció (FP).

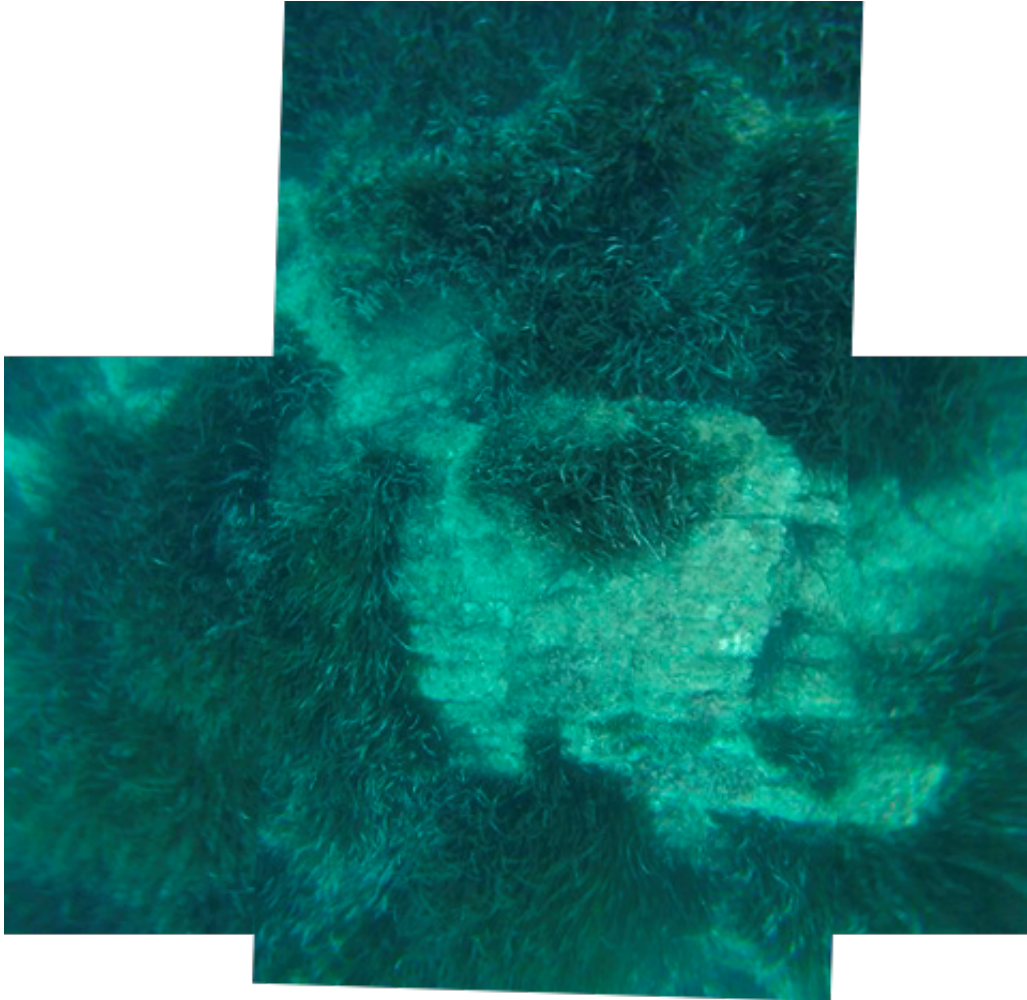


*Il·lustració 68 Imatge query.*



*Il·lustració 69 Imatge que la xarxa neuronal detecta com a bucle.*





*Il·lustració 70 Composició de la imatge query amb la detectada que forma un bucle.*

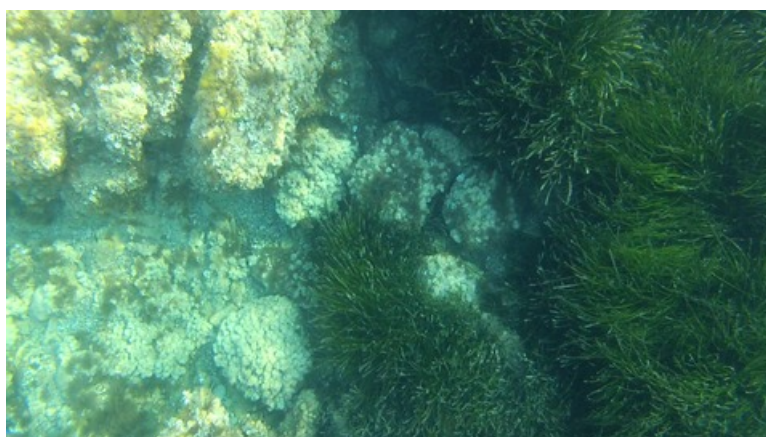


*Il·lustració 71 Imatge detectada com a bucle per la xarxa però realment no ho és.*

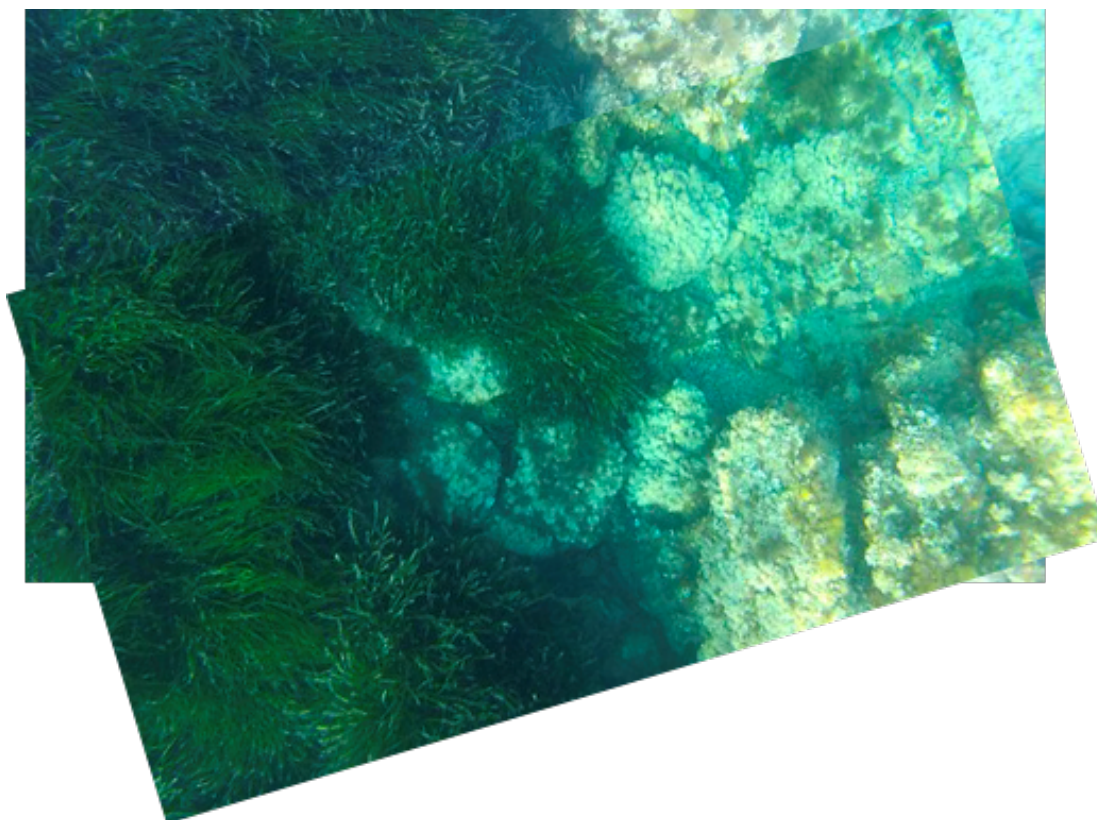
Un altre exemple d'imatge *query* és "00510\_n.jpg" (Il·lustració 72). Aquest fitxer, segons les dades del conjunt, forma un bucle amb l'arxiu "556\_n.jpg" (Il·lustració 73), però la xarxa neuronal no ho detecta com a bucle (FN). A la Il·lustració 74 es pot trobar la composició d'aquest bucle no detectat per la xarxa neuronal.



*Il·lustració 72 Composició del bucle no detectat per la xarxa neuronal.*



*Il·lustració 74 Imatge que forma un bucle real no detectat per la xarxa.*



*Il·lustració 73 Query no detectada per la xarxa.*

## 8. Conclusions

L'elaboració d'aquest projecte ha assolit gran part de les fites establertes a l'inici del mateix. El principal repte era aconseguir una xarxa neuronal que identifiqués possibles bucles en fons marins amb imatges i, segons les proves executades, s'ha obtingut una xarxa neuronal viable per a aquest propòsit.

L'estructura implementada, on s'ha separat el model de la xarxa neuronal, i la posterior metodologia de proves per escenaris, ha donat agilitat a l'hora de la cerca dels hiperparàmetres i de configuracions arquitectòniques diferents. De forma alternativa, la cerca dels hiperparàmetres podria haver estat amb un pas de llistes amb valors que podria haver automatitzat aquest procés. La part arquitectònica hauria sigut més complexa de treballar d'una forma similar.

L'autor ha creat una xarxa neuronal amb eines d'alt nivell (keras) que permet identificar bucles en imatges. A més ha adquirit coneixements en general de SLAM i *Deep Learning*, i en particular de capes convolucionals, generació de dades sintètiques, funcions d'activació i avaluació de xarxes neuronals entre altres.

El desenvolupador considera insuficients els coneixements matemàtics dels que disposa per comprendre i implementar xarxes neuronals a baix nivell. Tanmateix creu que eines com keras faciliten el desenvolupament i permeten l'aproximació al *Deep Learning* sense un alt coneixement en matemàtiques, però sí són necessàries algunes *percepcions*<sup>16</sup> dels efectes de les decisions que es prenen en el transcurs de la creació d'una xarxa neuronal.

### 8.1. Planificació final

La planificació final del projecte excedeix l'estimat inicialment. A continuació es detallen els principals blocs i la seva evolució.

Evolució global del projecte:

#### T00: Projecte sencer

Data inici:	<b>18/09/19</b>	Data Fi:	<b>05/05/20</b>	Hores estimades:	<b>351h</b>
Descripció:	Elaboració completa del projecte de TFM.				
% assolit:	100%	Nova Data Inici:		Nova Data Fi:	<b>02/06/20</b>
Hores:	Reals:	<b>428h</b>	$\Delta$ hores	+77h	$\Delta$ % +22%

Evolució de les tasques principals del projecte:

#### T01: Selecció TFM

Data inici:	<b>18/09/19</b>	Data Fi:	<b>30/09/19</b>	Hores estimades:	<b>30h</b>
Descripció:	En aquesta tasca s'avaluen les diferents propostes de TFM i s'acorda amb el director el contingut del mateix.				
% assolit:	100%	Nova Data Inici:		Nova Data Fi:	
Hores:	Reals:	<b>20h</b>	$\Delta$ hores	-10h	$\Delta$ % -33%

<sup>16</sup> Entenent com *percepció* el comportament que pot tenir una decisió o una altra sobre la xarxa neuronal, per exemple per que aplicar una capa d'agrupació després d'una capa de convolució.

## T02: Adquirint coneixement

Data inici:	<b>01/10/19</b>	Data Fi:	<b>29/10/19</b>	Hores estimades:	<b>62h</b>
Descripció:	Aquesta tasca permet adquirir coneixements tècnics per resoldre la problemàtica que planteja el TFM. Aquest coneixement inicial s'anirà ampliant durant el desenvolupament.				
% assolit:	100%	Nova Data Inici:		Nova Data Fi:	<b>12/05/20</b>
Hores:	Reals:	<b>85h</b>	$\Delta$ hores	+23h	$\Delta$ % +37%

## T03: Resolució del problema

Data inici:	<b>30/10/19</b>	Data Fi:	<b>24/03/20</b>	Hores estimades:	<b>189h</b>
Descripció:	Aquesta és la tasca principal del TFM. Inclou, entre altres, la creació del sistema per resoldre el problema plantejat pel TFM.				
% assolit:	100%	Nova Data Inici:		Nova Data Fi:	<b>02/05/20</b>
Hores:	Reals:	<b>223h</b>	$\Delta$ hores	+34h	$\Delta$ % +18%

## T04: Comparació amb altres sistemes

Data inici:	<b>25/03/19</b>	Data Fi:	<b>14/04/20</b>	Hores estimades:	<b>30h</b>
Descripció:	El sistema creat s'ha de comparar amb altres sistemes i determinar possibles punts de millora.				
% assolit:	0%	Nova Data Inici:		Nova Data Fi:	
Hores:	Reals:		$\Delta$ hores		$\Delta$ %
<b>Per assegurar la viabilitat i qualitat del projecte, aquesta tasca no es realitza.</b>					

## T05: Documentació

Data inici:	<b>01/10/19</b>	Data Fi:	<b>05/05/20</b>	Hores estimades:	<b>40h</b>
Descripció:	Aquesta serà una tasca de fons durant tot el projecte. Hi haurà un procés final d'adequació de la documentació, que tindrà dos lliurables: una memòria i una presentació.				
% assolit:	100%	Nova Data Inici:		Nova Data Fi:	<b>02/06/20</b>
Hores:	Reals:	<b>100h</b>	$\Delta$ hores	+60h	$\Delta$ % +150%

A continuació es detallen les tasques pels principals blocs del desenvolupament:

### T02-01: SLAM

Data inici:	<b>01/10/19</b>	Data Fi:	<b>29/10/19</b>	Hores estimades:	<b>15h</b>
Descripció:	Permet incorporar coneixements sobre la gestió de moviments en robots. Concretament les tècniques SLAM i V-SLAM.				
% assolit:	100%	Nova Data Inici:		Nova Data Fi:	<b>12/05/20</b>
Hores:	Reals:	<b>17h</b>	$\Delta$ hores	+2h	$\Delta$ % +13%

### T02-02: IA

Data inici:	<b>01/10/19</b>	Data Fi:	<b>29/10/19</b>	Hores estimades:	<b>40h</b>
Descripció:	Aprofundir en el coneixement de sistemes avançats de IA, com poden ser les xarxes neuronals o <i>Deep Learning</i> .				
% assolit:	100%	Nova Data Inici:		Nova Data Fi:	
Hores:	Reals:	<b>60h</b>	$\Delta$ hores	+20h	$\Delta$ % +50%

### T02-03: Estructura de dades

Data inici:	<b>01/10/19</b>	Data Fi:	<b>14/10/19</b>	Hores estimades:	<b>7h</b>	
Descripció:	Conèixer les estructures de dades pròpies per al treball.					
% assolit:	100%	Nova Data Inici:		Nova Data Fi:		
Hores:	Reals:	<b>8h</b>	Δhores	+1h	Δ%	+14%

### T03-01: Anàlisi del problema

Data inici:	<b>30/10/19</b>	Data Fi:	<b>05/11/19</b>	Hores estimades:	<b>13h</b>	
Descripció:	Anàlisi de la problemàtica a resoldre i estudi inicials per determinar els camins següents					
% assolit:	100%	Nova Data Inici:		Nova Data Fi:		
Hores:	Reals:	<b>13h</b>	Δhores	0h	Δ%	0%

### T03-02: Disseny

Data inici:	<b>06/11/19</b>	Data Fi:	<b>19/11/19</b>	Hores estimades:	<b>26h</b>	
Descripció:	Disseny de la solució que s'implementarà. S'ha de tenir en compte la informació del anàlisi. S'hauria de definir possibles alternatives.					
% assolit:	100%	Nova Data Inici:		Nova Data Fi:		
Hores:	Reals:	<b>30h</b>	Δhores	+4h	Δ%	+15%

### T03-03: Implementació

Data inici:	<b>20/11/19</b>	Data Fi:	<b>18/02/20</b>	Hores estimades:	<b>105h</b>	
Descripció:	Implementació del disseny.					
% assolit:	100%	Nova Data Inici:		Nova Data Fi:		
Hores:	Reals:	<b>110h</b>	Δhores	+5h	Δ%	+5%

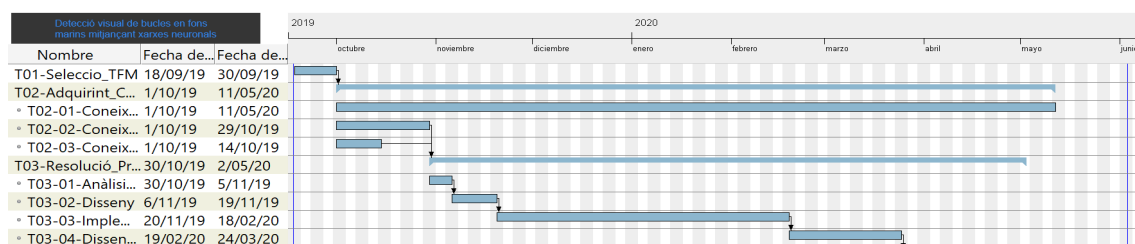
### T03-04: Disseny d'experiments

Data inici:	<b>19/02/20</b>	Data Fi:	<b>03/03/20</b>	Hores estimades:	<b>20h</b>	
Descripció:	Establir diferents experiments per poder avaluar la implementació.					
% assolit:	100%	Nova Data Inici:		Nova Data Fi:	<b>24/03/20</b>	
Hores:	Reals:	<b>25h</b>	Δhores	+5h	Δ%	+25%

### T03-05: Experimentació

Data inici:	<b>04/03/20</b>	Data Fi:	<b>24/03/20</b>	Hores estimades:	<b>25h</b>	
Descripció:	Realitzar proves per avaluar el rendiment de la solució.					
% assolit:	100%	Nova Data Inici:	<b>25/03/20</b>	Nova Data Fi:	<b>02/05/20</b>	
Hores:	Reals:	<b>45h</b>	Δhores	+20h	Δ%	+80%

El diagrama de Gantt del projecte final, sense comptar la presentació, es troba a la Il·lustració 75.



Il·lustració 75 Diagrama de Gantt a final de projecte.

Analitzant les dades es troben greus desviacions del temps estimat respecte el acomplert. Els principals motius són:

- L'estimació de la tasca de documentació va ser massa optimista.
- En el procés d'adquirir coneixement, l'autor del projecte no es va centrar en la part central del problema, fet que va provocar que necessites més temps.
- A l'hora d'executar els experiments, la configuració de l'equip de proves va provocar un augment del temps necessari per dur a terme els entrenaments.
- Raons personals han dificultat en alguns moments del projecte la plena dedicació.

La principal mesura correctora per assegurar la viabilitat i qualitat del projecte ha estat suprimir el bloc **T04: Comparació amb altres sistemes**. Aquest bloc permetia avaluar la xarxa neuronal respecte altres sistemes. Alternativament aquest punt es deixar com a treballs futurs.

## 8.2. Treballs futurs

A partir del treball presentat es poden suggerir diferents treballs futurs:

- Elaborar i avaluar altres afinaments de la xarxa. Per exemple:
  - Canviar mides de la imatge original i de les característiques finals.
  - Treballar amb tres canals de colors en comptes de la transformació a escala de grisos.
  - Altres modificacions en la generació d'imatges sintètiques.
  - Canvis a l'arquitectura del model.
- Us d'altres conjunts de dades d'imatges de fons marins, com poden ser:
  - El conjunt Aqualoc<sup>17</sup>.
  - El conjunt del *Girona underwater vision and robotics*<sup>18</sup>.
  - *ACFR Marine Robotics Program*<sup>19</sup>.
- Anàlisi d'altres tipus de xarxes per resoldre aquest problema, com per exemple NeRF<sup>20</sup>, xarxes neuronals siameses<sup>21</sup> o xarxes generatives antagòniques<sup>22</sup>.
- Comparació amb altres mètodes de detecció de bucles i quins efectes tenen sobre aquests l'aplicar la identificació d'imatges que la xarxa neuronal d'aquest estudi proporciona.

---

<sup>17</sup> <http://www.lirmm.fr/aqualoc/>

<sup>18</sup> <https://cirs.udg.edu/caves-dataset/>

<sup>19</sup> <http://marine.acfr.usyd.edu.au/>

<sup>20</sup> <http://www.matthewtancik.com/nerf>

<sup>21</sup> <https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf>

<sup>22</sup> <https://pathmind.com/wiki/generative-adversarial-network-gan>

## Glossari

**Bucle:** En l'àmbit d'aquest projecte defineix quan un robot passa per una posició per la que ja havia passat prèviament.

**Conjunt de dades:** Agrupació de dades que serveixen per entrenar, validar i testejar les xarxes neuronals. En aquest projecte seran imatges de fons marí.

**Database:** Relacionat amb el conjunt de dades, són imatges que ha capturat el robot del fons marí i poden o no formar un bucle.

**Entrenament:** Procés en el que una Xarxa Neuronal adapta els valors interns. Donada una entrada i una sortida esperada, el procés d'entrenament afina els valors per a que la sortida real sigui el més semblant a la sortida esperada.

**Època:** El procés d'entrenament d'una Xarxa Neuronal és un procés iteratiu. El número d'èpoques d'entrenament és el número de cicles que entrena la xarxa.

**Imatge sintètica:** Imatge generada a partir d'una altra. Aquesta nova imatge, en el cas d'aquest treball, pateix translacions, rotacions, canvis d'escala i canvis d'il·luminació, respecte la imatge original.

**Punt de referència:** Part d'una escena visitada per un robot que l'algoritme SLAM emmagatzema per poder tenir referència dels llocs visitats.

**Query:** Relacionat amb el conjunt de dades, són imatges que ha capturat el robot del fons marí i formen un bucle amb alguna imatge de Database.

**SLAM:** Localització i construcció de mapes simultanis. Intenta aconseguir que un robot mòbil ubicat en una posició desconeguda en un entorn no conegut construeixi un mapa consistent del seu entorn mentre, simultàniament, determina la seva ubicació en aquest mapa.

## Annex: Eines i llibreries emprades

Durant aquest projecte s'ha fet us de diferents eines:

- GanttProject
- Microsoft Excel
- Microsoft OneNote
- Microsoft Visio
- Microsoft Word
- Notepad++
- PyCharm

En el desenvolupament s'han utilitzat diferents llibreries, entre altres:

- python 3.6.7
- cv2: 3.4.2
- imutils 0.5.3
- keras 2.2.4
- matplotlib 3.0.3
- numpy 1.14.6
- scipy 1.2.1
- skimage: 0.14.1
- plaidml 0.7.0 (opcional)

Serveis utilitzats:

- DeepL, <https://www.deepl.com/>
- Diccionari Grup Enciclopèdia Catalana, <http://www.diccionari.cat/>
- Diccionari de sinònims Softcatalà, <https://www.softcatala.org/diccionari-de-sinonims/>
- Gitlab, <https://gitlab.com/>
- IEEEExplore, <https://ieeexplore.ieee.org/Xplore/home.jsp>
- OPTImot, <https://aplicacions.llengua.gencat.cat/llc/AppJava/index.html>
- termcat, <https://www.termcat.cat/>



## Bibliografía

- [1] H. Durrant-Whyte i T. Bailey, «Simultaneous Localization and Mapping: Part I,» *IEEE Robotics & Automation Magazine*, vol. 13, núm. 2, pp. 99-110, Juny 2006.
- [2] R. Smith, M. Self i P. Cheeseman, «A stochastic map for uncertain spatial relationships,» de *Proceedings of the 4th International Symposium on Robotics Research*, 1988.
- [3] M. Montemerlo, S. Thrun, D. Roller i B. Wegbreit, «FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges.,» de *IJCAI International Joint Conference on Artificial Intelligence*, 2003.
- [4] T. Bailey i H. Durrant-Whyte, «Simultaneous Localization and Mapping (SLAM): Part II,» *IEEE Robotics & Automation Magazine*, vol. 13, núm. 3, pp. 108-117, 2006.
- [5] J. Lai, Z. Liu i J. Lin, «Loop Closure Detection for Visual SLAM Systems Using Various CNN algorithms Contrasts,» de *Chinese Automation Congress (CAC)*, Hangzhou, 2019.
- [6] S. Wang, X. Lv, X. Liu i D. Ye, «Compressed Holistic ConvNet Representations for Detecting Loop Closures in Dynamic Environments,» *IEEE Access*, vol. 8, pp. 60552-60574, 2020.
- [7] C. Yu, Z. Liu, X.-J. Liu, F. Qiao, Y. Wang, F. Xie, Q. Wei i Y. Yang, «A DenseNet feature-based loop closure method for visual SLAM system\*,» de *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dali, 2019.
- [8] Y. Chen, L. Zuo, C. Zhang, F. Liu i Y. Wu, «Performance Evaluation of Deep Neural Networks in Detecting Loop Closure of Visual SLAM,» de *11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou, 2019.
- [9] N. Merrill i G. Huang, *Lightweight Unsupervised Deep Loop Closure*, Pittsburgh, PA, 2018.
- [10] V. Zhou, «Machine Learning for Beginners: An Introduction to Neural Networks,» 03 març 2019. [En línia]. Available: <https://victorzhou.com/blog/intro-to-neural-networks/>. [Últim accés: 03 gener 2020].
- [11] F. Rosenblatt, «Perceptron Simulation Experiments,» *Proceedings of the IRE*, vol. 48, núm. 3, pp. 301-309, 1960.
- [12] W. S. McCulloch i W. Pitts, «A logical calculus of the ideas immanent in nervous activity,» *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943.
- [13] S. Tiwari, «Activation functions in Neural Networks,» [En línia]. Available: <https://www.geeksforgeeks.org/activation-functions-neural-networks/>. [Últim accés: 03 gener 2020].
- [14] S. Raschka i V. Mirjalili, *Python Machine Learning*, Barcelona: Marcombo, 2019.
- [15] «ReLU,» TinyMind, [En línia]. Available: <https://www.tinywind.com/learn/terms/relu>. [Últim accés: 04 gener 2020].
- [16] A. Kurenkov, «A 'Brief' History of Neural Nets and Deep Learning,» 24 decembre 2015. [En línia]. Available: <https://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/>. [Últim accés: 04 gener 2020].
- [17] «¿Qué es el Deep Learning?,» SmartPanel, [En línia]. Available: <https://www.smartpanel.com/que-es-deep-learning/>. [Últim accés: 05 gener 2020].
- [18] J. Torres, «DEEP LEARNING,» [En línia]. Available: <https://torres.ai/deep-learning-inteligencia-artificial-keras/>.
- [19] A. Burkov, *The hundred-page machine learning book*, 2019.
- [20] L. Araujo dos Santos, «Artificial Intelligence,» [En línia]. Available: <https://leonardoaraujosantos.gitbooks.io/artificial-intelligence>.
- [21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama i T. Darrell, «Caffe: Convolutional Architecture for Fast Feature Embedding,» *Berkeley Artificial Intelligence Research*, 2014. [En línia]. Available: [http://caffe.berkeleyvision.org/doxygen/classcaffe\\_1\\_1EuclideanLossLayer.html#details](http://caffe.berkeleyvision.org/doxygen/classcaffe_1_1EuclideanLossLayer.html#details).
- [22] J. Brownlee, «How to Choose Loss Functions When Training Deep Learning Neural Networks,» *Machine Learning Mastery*, 30 Gener 2019. [En línia]. Available: <https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/>. [Últim accés: 20 Març 2020].
- [23] sitio big data, «Aprendizaje automatico y las Metricas de regresión,» 27 Agost 2018. [En línia]. Available: <https://sitiobigdata.com/2018/08/27/machine-learning-metricas-regresion-mse/#>. [Últim accés: 4 Abril 2020].
- [24] J. Martínez, «¿Qué es un Optimizador y Para Qué Se Usa en Deep Learning?,» *DataSmarts*, [En línia]. Available: <https://datasmarts.net/es/que-es-un-optimizador-y-para-que-se-usa-en-deep-learning/>. [Últim accés: 4 Abril 2020].

- [25] A. Anwar, «Difference between Local Response Normalization and Batch Normalization,» Towards Data Science, 19 Juny 2019. [En línia]. Available: <https://towardsdatascience.com/difference-between-local-response-normalization-and-batch-normalization-272308c034ac>. [Últim accés: 13 Abril 2020].
- [26] S. J. Pawan, «Visualizing Various Filter Initializers in Keras,» Medium, 17 Març 2019. [En línia]. Available: <https://blog.goodaudience.com/visualizing-various-filter-initializers-in-keras-ca14c996db22>. [Últim accés: 2 Maig 2020].
- [27] A. Burguera Burguera i F. Bonin-Font, «Towards Visual Loop Detection in Underwater Robotics using a Deep Neural Network,» de *International Conference on Computer Vision Theory and Applications (VISAPP)*, Malta, 2020.
- [28] S. Sharma, «Activation Functions in Neural Networks,» towards data science, 6 Setembre 2017. [En línia]. Available: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>. [Últim accés: 25 Abril 2020].
- [29] MissingLink, «7 Types of Neural Network Activation Functions: How to Choose?,» [En línia]. Available: <https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/>. [Últim accés: 25 Abril 2020].
- [30] N. Ollis, «Implementing Swish Activation Function in Keras,» Big Nerd Ranch, 17 Setembre 2019. [En línia]. Available: <https://www.bignerdranch.com/blog/implementing-swish-activation-function-in-keras/>. [Últim accés: 25 Abril 2020].
- [31] scikit-image development team, «Histogram of Oriented Gradients,» [En línia]. Available: [https://scikit-image.org/docs/dev/auto\\_examples/features\\_detection/plot\\_hog.html](https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_hog.html). [Últim accés: 23 Novembre 2019].
- [32] glovera, «Day and Night with Lateral Pose Change Datasets,» Robotics@QUT, 14 Febrer 2014. [En línia]. Available: <https://wiki.qut.edu.au/display/cyphy/Day+and+Night+with+Lateral+Pose+Change+Datasets>. [Últim accés: 24 Novembre 2019].
- [33] J. Ju, «Jihong Ju Blog,» 10 maig 2017. [En línia]. Available: <https://jihongju.github.io/2017/05/10/caffe-filler/>. [Últim accés: 14 decembre 2019].
- [34] O. Lam, «Alderley Day/Night Dataset,» Robotics@QUT, 15 Abril 2019. [En línia]. Available: <https://wiki.qut.edu.au/pages/viewpage.action?pageId=181178395>. [Últim accés: 23 Novembre 2019].
- [35] N. Merrill, «Convolutional Autoencoder for Loop Closure,» 30 Abril 2018. [En línia]. Available: [https://github.com/rpng/calc/blob/master/TrainAndTest/test\\_data/CampusLoopDataset.tar.gz](https://github.com/rpng/calc/blob/master/TrainAndTest/test_data/CampusLoopDataset.tar.gz). [Últim accés: 16 Novembre 2019].