TFG – Arquitectura de computadors i sistemes operatius

FITA#06: Entrenament del model CNN

Gestió del projecte a GitHub:

Branca del repositori:
https://github.com/UOC-Assignments/uoc.tfg.jbericat/tree/FITA%2306

Dashboard de seguiment de les tasques associades a la fita:
https://github.com/UOC-Assignments/uoc.tfg.jbericat/projects/7

Estudiant: *Jordi Bericat Ruz*

Professor col·laborador: *Daniel Rivas Barragan*

Semestre: *Tardor 2021/22 (Aula 1)*

Versió: *ESBORRANY_v6*

Índex

6 - Entrenament del model CNN

6.1 - Tasques de recerca i investigació

https://iq.opengenus.org/precision-recall-sensitivity-specificity/

6.1.1 – Fonaments de la detecció d'objectes amb aplicacions de visió per computador[1]

AIXÒ HA D'ANAR A LA FITA 6!

Per tal de definir els objectius de l'anàlisi d'imatges que realitzarà el model que es desenvoluparà durant aquesta secció, primer cal conèixer quins tipus d'aplicacions de visió per computador podem desenvolupar, en funció de si volem detectar un sol objecte, o bé un conjunt de múltiples objectes. Concretament, podem trobar els següents casos d'ús:

- **1) Un sol objecte**
    - **1.1) Classificació:** Es tracta del cas més senzill. L'entrenament del model té com a finalitat predir si un objecte es troba present o no en una imatge.
    - **1.2)Classificació i localització:** El mateix que la classificació, però en aquest cas es tracta d'emmarcar l'objecte detectat amb un *frame* o *bounding box*
- **2) Múltiples objectes**
    - **2.1) Detecció d'objectes:** En el cas que una imatge contingui múltiples instàncies d'objectes, es tracta de detectar-los tots ells, tot afegint un *bounding box* per a delimitar la identificació de cada objecte detectat. Es pot considerar una versió més complexa del cas de classificació i localització.
    - **2.2) Segmentació d'instàncies d'objecte:** Es tracta del cas que presenta un repte major d'implementació. A diferència de la detecció d'objectes, en aquest cas es tracta de delimitar la forma (*shape)* de l'objecte detectat (tècnica coneguda com a segmentació d'imatges).

La figura 6.x.x.x mostra un exemple de cadascun dels casos esmentats:

[1] https://medium.com/analytics-vidhya/guide-to-object-detection-using-pytorch-3925e29737b9

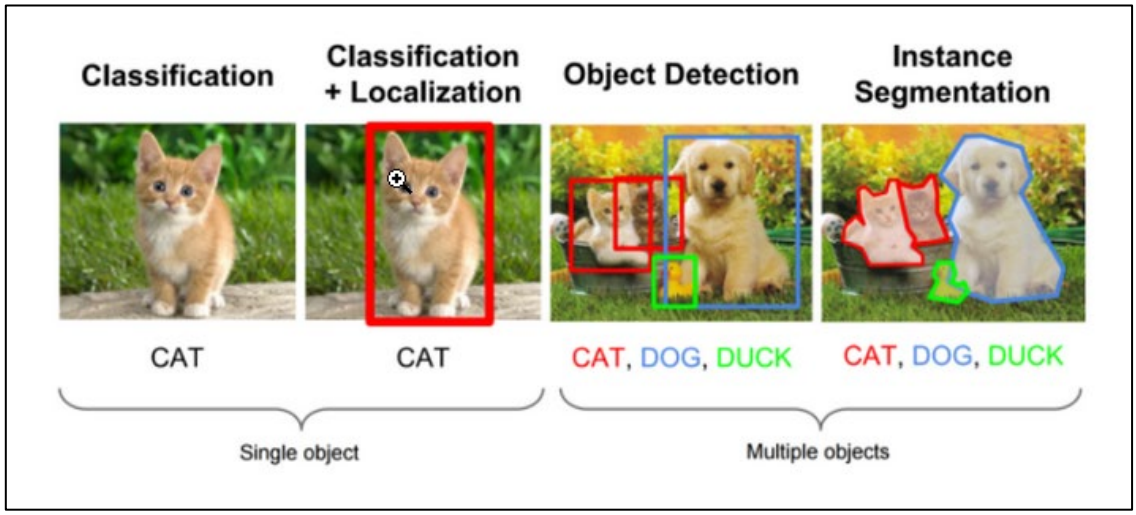


Figura 6.x.x.x – *Tècniques de visió per computador. Font: https://medium.com/analytics-vidhya/guide-to-object-detection-using-pytorch-3925e29737b9*

Fetes les breus consideracions anteriors, la idea inicial és la d'implementar el cas 1.2 (classificació + localització d'imatges). Tanmateix, primer es provarà a implementar el cas 1.1 (només classificació) donada la limitació temporal establerta per a la finalització d'aquest TFG, i si resta temps es realitzarà la implementació de la localització. En cas que no sigui possible, aquesta implementació queda oberta per a futurs exercicis de recerca per part d'altres investigadors que vulguin continuar treballant els conceptes presentats en aquest projecte.

6.1.2 – Identificació / definició de les mètriques de rendiment[2]

Definir les mètriques que s'utilitzaran per a mesurar el rendiment del model durant el seu entrenament és una tasca imprescindible per a poder monitoritzar si els canvis que es van realitzant en la configuració dels paràmetres a les diferents iteracions o *epoch* influeixen d'una manera o altra en la precisió (*accuracy*) de les prediccions realitzades pel model[3].

Per a establir doncs un marc en el qual puguem definir aquestes mètriques es representarà el rendiment del model DCNN mitjançant una **"taula de confusió"** (mètode de representació de mesures d'encert en la predicció utilitzat habitualment durant l'entrenament de models de ML i DL).

---

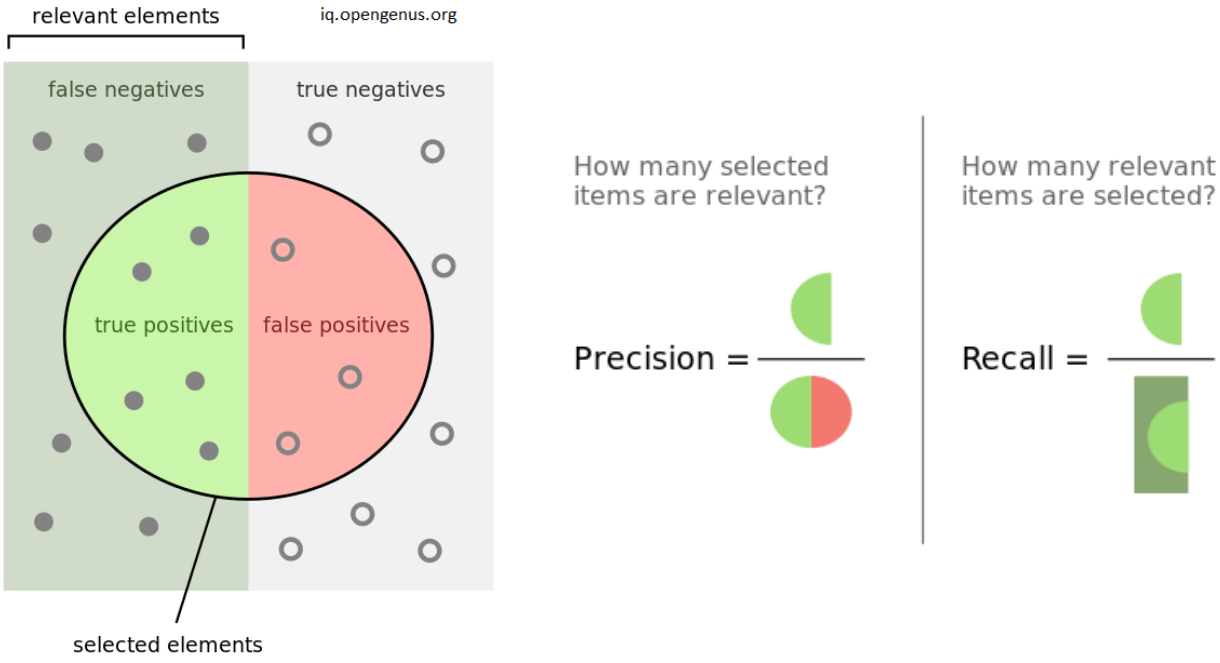
2 https://iq.opengenus.org/precision-recall-sensitivity-specificity/

3 Pàgina 148 del llibre de DL → "Defining the model evaluation metric is a necessary step because it will guide your approach to improving the System. Without clearly defined mètrics, it can be difficult to tell whether changed to a ML System result in progress or not".

Simplificant, l'objectiu al representar el rendiment del model mitjançant d'una taula de confusió és el de poder centrar la mesura de l'encert de les prediccions en funció de, o bé la proporció de falsos positius (falses deteccions d'incendis) o bé de la de falsos negatius (no detecció d'un incendi). Així en funció de la importància de que es produeixi un o altre cas en el projecte de DL que s'estigui desenvolupant, caldrà tenir més cura de controlar els falsos positius (mesurats mitjançant la *precisió),* o bé els falsos negatius (mesurats mitjançant la sensitivitat o *recall)*

iq.opengenus.org		Predicted Class	
		NO	YES
Actual Class	NO	True Negative (TN)	False Positive (FP)
	YES	False Negative (FN)	True Positive (TP)

Imatge obtinguda de: https://iq.opengenus.org/precision-recall-sensitivity-specificity/



Imatges obtingudes de: https://iq.opengenus.org/precision-recall-sensitivity-specificity/

Així doncs, fetes les consideracions anteriors, en aquest punt cal decidir quina de les possibles falses prediccions pot implicar "pitjors" conseqüències a l'hora de predir si una imatge conté o no característiques pròpies d'un incendi forestal (objectiu central de la PdC d'aquest TFG). Concretament, només es poden produir dos tipologies de falses prediccions:

- **Errada de tipus I: Fals positiu (*False Positive o FP*)**

  Un "fals positiu" significa que s'ha reconegut / identificat un incendi a una imatge però en realitat no ho és (p.e. en el cas de la PdC, es podria donar el cas que un cos o objecte que emeti calor es detecti com la característica d'un incendi, per exemple, una pila de compost orgànic)

- **Errada de tipus II: Fals negatiu (*False Negative o FN*)**

  Un fals negatiu significa que, havent-t'hi un incendi forestal a la imatge processada per l'algorisme d'aprenentatge profund, aquest no ha reconegut cap característica pròpia d'aquests.

Per tant, si tenim en compte que la detecció d'un incendi a les seves fases inicials és un factor crític que pot implicar la seva extinció abans que aquest causi danys a major escala, aleshores podem concloure que **tindrà pitjors conseqüències que es produeixi una predicció errònia de tipus II (FN)** que no pas de tipus I (FP). Podem determinar doncs que la mètrica de rendiment que ens permetrà identificar millor, o tenir una visió més precisa sobre la proporció d'incendis no detectats (FN) serà la basada en la **sensitivitat (*sensitivity* / *recall*)** o rati de positius vertaders (*True Positives o TP*). Dit d'una altra manera, la sensitivitat com a mètrica de rendiment ens permetrà saber amb millor exactitud que la precisió quantes vegades s'ha produït un FN durant cada iteració (*epoch*) d'entrenament del model, fet que ens ajudarà a fer un reajustament més precís dels paràmetres (pesos de les arestes del graf) i en conseqüència a obtenir un model final completament entrenat que mostri un rendiment més acurat (*model performance & accuracy*).

Podem calcular la sensitivitat de les dades mesurades amb la següent expressió:

$$sensitivitat = \frac{TP}{TP + FN}$$

Un cop realitzat l'estudi anterior per a determinar la millor manera de mesurar el rendiment del model a l'hora de resoldre el problema binari de classificar imatges en funció de si contenen o no característiques d'incendis, passem a ara a estudiar el cas en el què el resultat de resoldre el problema de classificació ha de ser la categorització d'un incendi en funció de les seves característiques, tal com es defineix a la següent taula[4]:

Categoria d'incendi (Intensitat x Mida)	Localitzat	**Moderadament estès**	**Molt estès**
Intensitat Baixa	A	**B**	**C**
Intensitat Moderada	**D**	**E**	**F**
Intensitat Alta	G	**H**	**I**

Abans però caldrà fer notar que, tenint en compte que s'està desenvolupant una PdC, per a simplificar considerarem les següents prioritats (essent "A" el nivell o categoria menys crítica i "I" el nivell o categoria més crítica) sense tenir en compte altres factors, com podria ser la proximitat a zones habitades:

```
A < B < C < D < E < F < G < H < I
```

En aquest nou escenari podem definir:

- **Errada de tipus I: Fals positiu (*False Positive o FP*)**

  Ara, un "fals positiu" significa que havent processat una imatge que conté un incendi de categoria inferior (o bé sense cap incendi), el model n'ha identificat un, i de categoria superior si la imatge en contenia característiques.

- **Errada de tipus II: Fals negatiu (*False Negative o FN*)**

  Ara, un fals negatiu significa que, havent processat una imatge que conté un incendi de categoria superior, el model n'ha identificat un de categoria inferior, o bé no n'ha identificat cap.

Observem que, de la mateixa manera que passava amb el problema de classificació binari, la mètrica més adequada serà la que ens permeti tenir una visió més acurada dels **falsos negatius**

[4] *En vermell es marquen aquelles classes que no han sigut implementades a la secció 7en aquest projecte (la causa ha sigut la necessitat de simplificar la PdC per tal que els objectius de la mateixa siguin assolibles)*

(FN) que es produeixin durant l'entrenament del model, és a dir, la **sensitivitat** de les mesures. Com a conseqüència o *side-effect* però, tindrem que amb aquesta mètrica no tindrem dades tant precises de la quantitat de falsos positius que es produeixin. Tanmateix, a la pràctica en el cas que ens ocupa això només podria suposar que en un moment donat algun dels drons de l'eixam intenti informar de / apagar un incendi inexistent, o bé que s'hi dediquin més recursos dels necessaris per a gestionar la situació, com a conseqüència d'un entrenament poc acurat del model. Així doncs, un cop realitzades les primeres mesures amb la mètrica seleccionada (sensitivitat o *recall*) contra el *dataset* d'explotació (veure secció 5.3), si s'observa un rendiment pobre en el cas de falsos positius el que caldrà fer serà tornar a fer l'entrenament del model utilitzant una mètrica menys conservadora, però que ens doni una visió més precisa dels falsos positius (sota risc de què es comencin a detectar casos de falsos negatius, cas en el que s'hauria d'avaluar si paga la pena el canvi de mesura de rendiment donades les fatals conseqüències que pot tenir **una sola** predicció errònia de tipus II en un incendi forestal). En aquest sentit, la mètrica basada en *F-Score*[5] és una possibilitat que cal tenir en compte i que s'estudiarà en funció dels resultats (*accuracy*) de les mesures basades en la sensitivitat (*recall*).

Podem calcular el *F-Score* de les dades mesurades de la següent manera:

$$r = sensitivitat = \frac{TP}{TP + FN}$$

$$p = \ precisió = \frac{TP}{TP + FP}$$

$$F - Score = \frac{2pr}{p + r}$$

5 Refs: Punt 4.1.4, pàg. 149 llibre DL

6.2 - Selecció del model i del dataset finals que s'utilitzaran per a la PdC

Ara que ja tenim tots els components necessaris per a realitzar la PdC, podem procedir a seleccionar les versions del model de xarxa neuronal i del dataset d'imatges que utilitzarem.

6.2.1 – Selecció del model

Pel que fa al model, de partida escollim la versió que té més capes, ja que la quantitat d'aquestes determina en gran mesura el rendiment del model (mentre mes capes, major promig d'encert en les prediccions). Per tant **) per a realitzar la PdC utilitzarem la versió 3.0 del model CNN (14 capes).**

6.2.2 – Selecció del dataset

Abans d'establir la versió del dataset serà necessari comprovar que el rendiment assolit amb la combinació de model + dataset escollits ens permet realitzar la prova en unes condicions en les que prediccions que es realitzin siguin correctes ni que sigui en la majoria de casos (aconseguir un 99% d'encert serà difícil donada la naturalesa de les imatges i altres factors, com s'ha comentat a la secció 4). Per, tant, **establirem la fita d'assolir al menys entre un 80% un 85% de rendiment a l'avaluar el model contra el subset de test**.

Dit això, tot seguit es procedeix a realitzar diferents execucions de l'script d'entrenament `/src/CNN/ytorch_training.py` (veure secció 5.3) tot variant les condicions de l'entorn i aplicant els canvis es creguin convenients a la naturalesa del dataset. **Un realitzades les accions que es descriuran a continuació, finalment s'acabarà seleccionant la versió 9.0**

6.2.2.1 – Dataset v4.0

En primer lloc tenim que entrenant el model v3.0 amb la versió del dataset v4.0[6] que presenta les següents característiques:

6 PATH DATASET v4.0

- Imatges de 650x512x3 (els canals *grayscale* estan clonats degut a limitacions del framework openCV[7] utilitzat a l'script `capture_ir_segmentation.py`, veure secció 4.x.x.x)
- **imatges llunyanes**
- **totes les classes** definides a la secció 4.x.x.x,)
- **Hiperparàmetres sense** optimitzar (s'ajusten de manera aleatòria per assaig-error sense realizar els càlculs definits a la secció 5.2.3

Donades les condicions anteriors tenim els següent resum de rendiments[8] obtinguts per a cada iteració d'entrenament (recordem que el rendiment es mesura contra el dataset de test al final de cada epoch):

```
The number of images in a training set is:  610
The number of images in a test set is:  210
The number of batches per epoch is:  61
The model will be running on cuda:0 device
For epoch 1 the test accuracy over the whole test set is 43 %
For epoch 2 the test accuracy over the whole test set is 38 %
For epoch 3 the test accuracy over the whole test set is 39 %
For epoch 4 the test accuracy over the whole test set is 32 %
For epoch 5 the test accuracy over the whole test set is 36 %
Finished Training
Real labels:  high-intensity-wildfire high-intensity-wildfire high-intensity-wildfire high-
intensity-wildfire high-intensity-wildfire high-intensity-wildfire high-intensity-wildfire high-
intensity-wildfire high-intensity-wildfire high-intensity-wildfire
Predicted:  no-wildfire low-intensity-wildfire  high-intensity-wildfire low-intensity-wildfire  no-
wildfire low-intensity-wildfire  no-wildfire low-intensity-wildfire  low-intensity-wildfire  no-
wildfire
```

Figura 6.x.x.x.a – Rendiment obtingut al entrenar el model v3 amb les imatges del dataset v4.0. S'observa un 90% d'errades en la predicció al fer un test final amb una mostra petita (10 imatges)

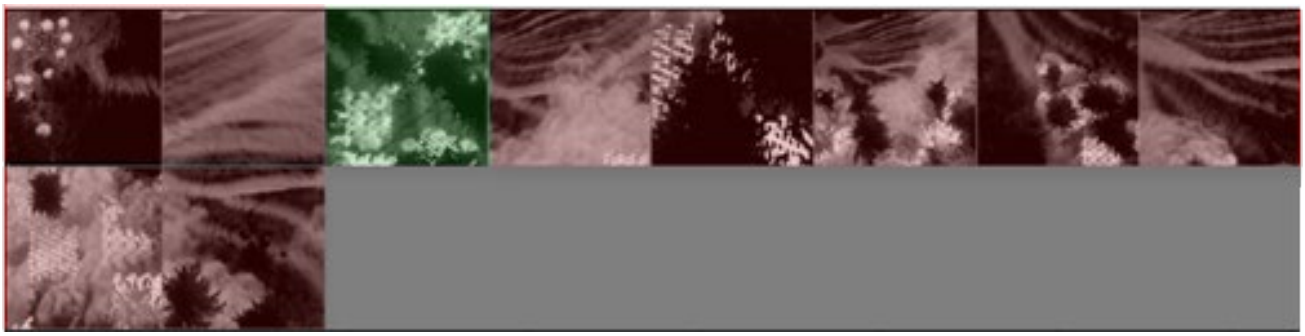


Figura 6.x.x.x.b – prediccions corresponent a la prova realitzada a la figura anterior.

---

7 In OpenCV (Python), why am I getting 3 channel images from a grayscale image? https://stackoverflow.com/questions/18870603/in-opencv-python-why-am-i-getting-3-channel-images-from-a-grayscale-image → "It seems that you have saved the image as BGR, however it is not true, it is just opencv, by default it reads the image with 3 channels, and in the case it is grayscale it copies its layer three times."

8 No s'ha conservat cap arxiu resum amb les estadístiques d'entrenament d'aquesta versió

A la figura Figura 6.x.x.x.a podem observar com el rendiment assolit és molt pobre, segurament degut a la pròpia naturalesa de les imatges:

- La mida original de les imatges hauria de ser quadrada (p.e. 512x512) i no 640x512, ja que d'aquesta manera al fer el `crop` potser estem convertint algunes imatges de la seva classe a la classe `no-wildfire` (és a dir, potser s'està retallant la part de la imatge que conté l'incendi)

- Caldria provar a generar un altre cop el dataset amb imatges molt més properes, en les que no s'hi trobin tantes instàncies de classe d'incendi en una sola imatge.

- Cal fer recerca per a realitzar els càlculs d'hiper-paràmetres[9]

6.2.2.2 – Dataset v5.0

Es decideix generar el dataset v5. tot aplicant el canvis esmentats a la secció 6.2.2.1 i comparar els resultats amb els del dataset v4.0. Un cop generada una nova versió s'aconsegueix millorar lleugerament el rendiment del model[10], tot i que encara segueix sense presentar un rendiment acceptable:

```
Type the dataset version you want to use to train the model (4 = v4 / 5 = v5): 5
The number of images in a training set is:  540
The number of images in a test set is:  220
The number of batches per epoch is:  54
The model will be running on cuda:0 device
For epoch 1 the test accuracy over the whole test set is 55 %
For epoch 2 the test accuracy over the whole test set is 45 %
For epoch 3 the test accuracy over the whole test set is 48 %
For epoch 4 the test accuracy over the whole test set is 55 %
For epoch 5 the test accuracy over the whole test set is 44 %
Finished Training
Real labels:  high-intensity-wildfire high-intensity-wildfire high-intensity-wildfire high-
intensity-wildfire high-intensity-wildfire high-intensity-wildfire high-intensity-wildfire
high-intensity-wildfire high-intensity-wildfire high-intensity-wildfire
Predicted:  low-intensity-wildfire  low-intensity-wildfire  no-wildfire high-intensity-
wildfire low-intensity-wildfire  high-intensity-wildfire low-intensity-wildfire  low-
intensity-wildfire  low-intensity-wildfire  low-intensity-wildfire
```

Figura 6.x.x.x – Rendiment obtingut al entrenar el model v3 amb les imatges del dataset v5.0. S'observa un 80% d'errades en la predicció al fer un test final amb una mostra petita (10 imatges), tot i que en general s'assoleixen resultats lleugerament millors que amb el dataset v4.0

9 En el moment de provar el dataset v4.0 encara no s'havia fet la recerca que va permetre els càlculs de la secció 5.x.x.x

10 No s'ha conservat cap arxiu resum amb les estadístiques d'entrenament d'aquesta versió

Com es pot observar a la figura 6.x.x.x, el rendiment obtingut segueix essent molt pobre. Es considera realitzar les següents accions per tal d'aconseguir la fita marca (80%-85% de rendiment):

- Aplicar els càlculs d'hiper-paràmetres realitzats a secció 5.x.x.x
- Implementar tècniques d'ampliació del dataset (veure secció 5.x.x.x)
- **No s'ha fet un "curat" manual de les imatges** del dataset (que recordem, s'han obtingut de manera automàtica mitjançant l'script `capture_ir_segmentation.py`), i n'hi ha bastants que no mostren clarament característiques de la classe etiquetada, o que són de mala qualitat. Això és degut a què és difícil calcular la distància per capturar imatges automàticament a l'entorn Unreal, terreny muntanyós on les alçades varien respecte de diferents punts de captura
- Després d'un anàlisi visual, s'ha detectat que **les imatges amb incendis de mitjana intensitat** (classe `medium-intensity-wildfires`) **mostren característiques massa semblants a les de la classe de baixa intensitat** `low-intensity`, segurament també indistingibles pel model (caldria de fer proves). Es tracta d'una errada de disseny a l'hora de preparar l'entorn per a prendre aquesta classe d'imatges en particular, caldrà pensar un *workaround*

6.2.2.3 – Dataset v6.0

De les consideracions fetes a la secció anterior, s'han completat els següents punts:

- Aplicar els càlculs d'hiper-paràmetres realitzats a secció 5.x.x.x
- Implementar tècniques d'ampliació del dataset (veure secció 5.x.x.x). En aquest cas només s'efectua l'ampliació (duplicació de mida) mitjançant el mirroring (`horitzontal-flip`)

Però no s'ha observat cap millora en el rendiment[11] (65%), mentre que el temps d'entrenament augmenta degut a l'increment d'imatges a processar.

[11] *No s'ha conservat cap arxiu resum amb les estadístiques d'entrenament d'aquesta versió*

6.2.2.4 – Dataset v7.0-beta

Donat que la versió v6.0 no ha presentat millores, es decideix conservar els canvis efectuats i a més aplicar la següent consideració de les fetes durant l'estudi de la versió 5 a la secció 6.2.2.2:

- Després d'un anàlisi visual, s'ha detectat que les imatges sense incendis (classe `medium-intensity-wildfires`) mostren característiques massa semblants a les de la classe `low-intensity`, segurament també indistingibles pel model (caldria de fer proves). Es tracta d'una errada de disseny a l'hora de preparar l'entorn per a prendre aquesta classe d'imatges en particular, caldrà pensar un *workaround*

Així doncs, s'efectua la prova d'eliminar el subset de classe `medium-intensity-wildfire` i en aquestes circumstàncies **s'assoleix un rendiment del 83%**, que entra dins els objectius mínims a assolir que ens havíem marcat:

```
For epoch 1 the test accuracy over the whole test set is 66 %
For epoch 2 the test accuracy over the whole test set is 80 %
For epoch 3 the test accuracy over the whole test set is 83 %
For epoch 4 the test accuracy over the whole test set is 74 %
For epoch 5 the test accuracy over the whole test set is 83 %

Real labels: high-intensity-wildfire high-intensity-wildfire high-intensity-
wildfire high-intensity-wildfire high-intensity-wildfire high-intensity-
wildfire high-intensity-wildfire high-intensity-wildfire high-intensity-
wildfire high-intensity-wildfire

Predicted: high-intensity-wildfire high-intensity-wildfire low-intensity-
wildfire high-intensity-wildfire high-intensity-wildfire high-intensity-
wildfire high-intensity-wildfire high-intensity-wildfire high-intensity-
wildfire high-intensity-wildfire
```

Figura 6.x.x.x.a – Rendiment obtingut al entrenar el model v3 amb les imatges del dataset v7.0. S'observa un 10% d'errades en la predicció al fer un test final amb una mostra petita (10 imatges)

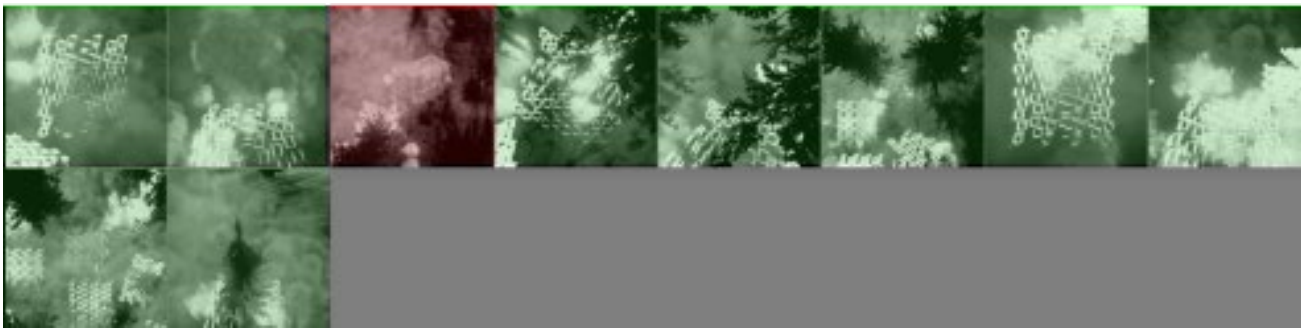


Figura 6.x.x.x.b – prediccions corresponent a la prova realitzada a la figura anterior.

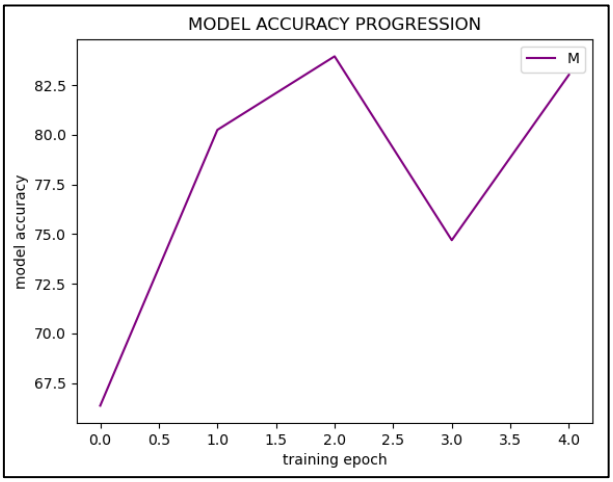


Figura 6.x.x.x.c - *Corba d'aprenentatge del model v3.0 utilitzant les dades del dataset v7.0-beta*

Per tant, es conclou que el problema del baix rendiment obtingut a les versions anteriors del dataset era la mala qualitat de la simulació de les imatges de classe `medium-intensity-wildfire` (s'hauria d'haver utilitzat un altre tips d'objecte `StaticMesh` de Unreal Engine que no s'assemblés tant als que s'utilitaven per a la classe `low-intensity-wildfire`). També cal observar que la simulació de visió tèrmica s'ha adaptat d'un altre projecte que tenia com a objectiu la detecció d'animals i no de foc. Aquest darrer fet ha sigut un *handicap* que ha "marcat" l'execució del projecte).

Per a provar de millorar encara més el dataset es proposa el següent:

- Implementar la divisió dels datasets d'entrenament i validació en el moment de carregar les imatges al model
- Realitzar diferents assajos variant paràmetres com:
  - Learning-rate
  - batch-size
  - nombre d'epochs

6.2.2.5 – Dataset v7.0

Un cop aplicades les millores proposades al punt anterior i fets els primers entrenaments de prova mb el nou algorisme, els resultats obtinguts[12] són els següents:

[12] *Es poden consultar els resums d'entrenament al següent arxiu del repositori → `/bin/CNN/cnn-training_XXXXXXXXXXXXX.tar.gz`*

```
*********************************************************************************
***                                                                           ***
***                      PoC's CNN Model Training Summary                     ***
***                                                                           ***
*********************************************************************************
***                                                                           ***
***                            Model version: v3.0                            ***
***                           Dataset version: v7.0                           ***
***                                                                           ***
*********************************************************************************

 TRAINING PARAMETERS:

 - Image size = (229 x 229 x 1)
 - Number of classes / labels = 3
 - Batch size = 128
 - Learning rate = 1e-05

*********************************************************************************

 DATASET TOTALS:

 - The number of images in a training set is:  768
 - The number of images in a validation set is:  256
 - The number of images in a test set is:  384
 - The number of batches per epoch is:  6

*********************************************************************************

 CNN BLUEPRINT:

 Network_v3(
   (conv1): Conv2d(1, 16, kernel_size=(9, 9), stride=(2, 2), padding=(2, 2))
   (bn1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   (conv2): Conv2d(16, 16, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2))
   (bn2): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   (pool): MaxPool2d(kernel_size=2, stride=1, padding=0, dilation=1, ceil_mode=False)
   (conv3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(2, 2))
   (bn3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   (conv4): Conv2d(32, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
   (bn4): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   (fc1): Linear(in_features=107648, out_features=3, bias=True)
)

*********************************************************************************

  MODEL TRAINING STATS:

 Model trained on cuda:0 device
 Training + validation time:  237.085171875  seconds

*********************************************************************************

 MODEL TESTING STATS:

 For epoch 1 the TEST accuracy over the whole TEST dataset is 55 %
 For epoch 2 the TEST accuracy over the whole TEST dataset is 54 %
 For epoch 3 the TEST accuracy over the whole TEST dataset is 53 %
 For epoch 4 the TEST accuracy over the whole TEST dataset is 59 %
 For epoch 5 the TEST accuracy over the whole TEST dataset is 64 %
 For epoch 6 the TEST accuracy over the whole TEST dataset is 66 %
 For epoch 7 the TEST accuracy over the whole TEST dataset is 72 %
 For epoch 8 the TEST accuracy over the whole TEST dataset is 75 %
 For epoch 9 the TEST accuracy over the whole TEST dataset is 80 %
 For epoch 10 the TEST accuracy over the whole TEST dataset is 82 %

Real labels:  high-intensity-wildfires high-intensity-wildfires no-wildfires high-intensity-wildfires
no-wildfires high-intensity-wildfires no-wildfires no-wildfires no-wildfires no-wildfires low-intensity-
wildfires no-wildfires high-intensity-wildfires no-wildfires low-intensity-wildfires low-intensity-
wildfires no-wildfires low-intensity-wildfires high-intensity-wildfires no-wildfires

Predicted:  low-intensity-wildfires low-intensity-wildfires no-wildfires high-intensity-wildfires no-
wildfires high-intensity-wildfires low-intensity-wildfires no-wildfires low-intensity-wildfires no-
wildfires low-intensity-wildfires no-wildfires high-intensity-wildfires no-wildfires low-intensity-
wildfires low-intensity-wildfires no-wildfires no-wildfires high-intensity-wildfires no-wildfires
```

Figura 6.x.x.x.a *– Rendiment obtingut al entrenar el model v3 amb les imatges del dataset v7.0. S'observa un 20% d'errades en la predicció al fer un test final amb una mostra petita (25 imatges)*

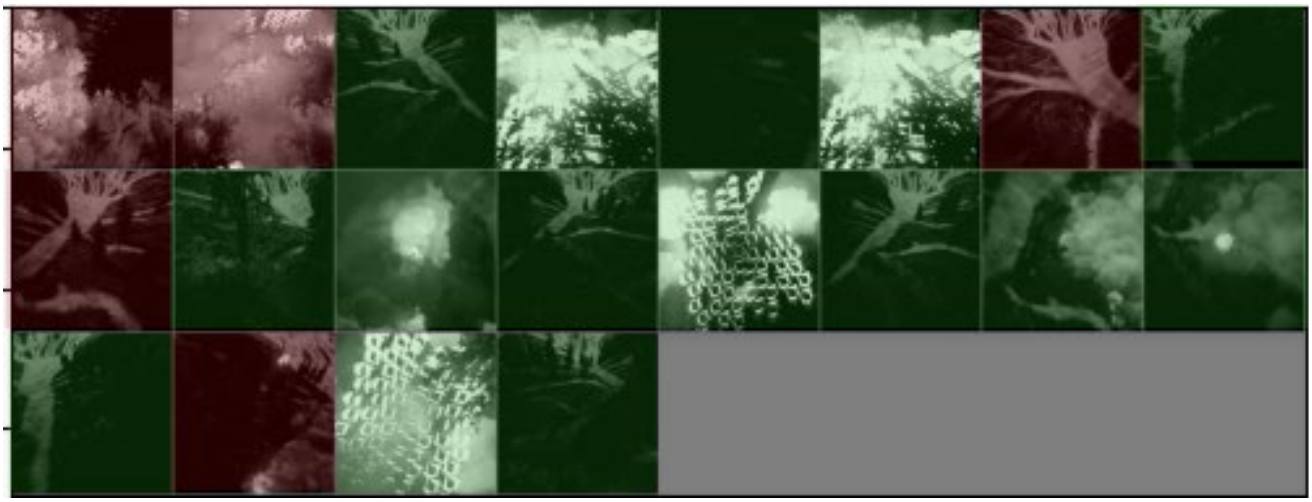


Figura 6.x.x.x.b *– prediccions corresponents a la prova realitzada a la figura anterior*

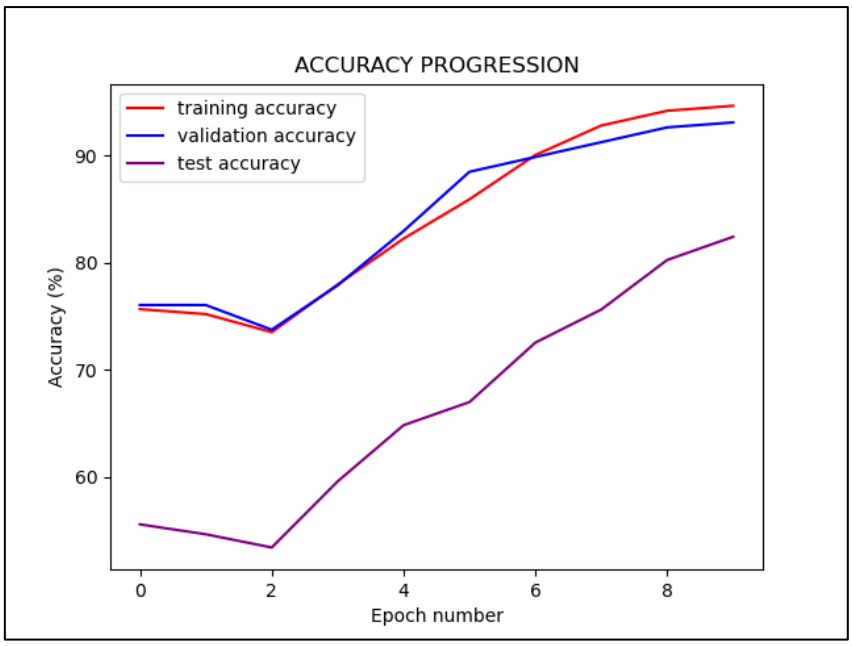


Figura 6.x.x.x.c - *Corba d'aprenentatge del model v3.0 utilitzant les dades del dataset v7.0*

6.2.2.6 – Dataset v8.0

Aquest dataset tractava d'implementar el mateix que la versió 9, excepte la fusió de les classes `medium-intensity-wildfires` i `low-intensity-wildfires`. Tanmateix, degut a les conclusions que es presentaran més endavant a la secció 6.x.x.x, finalment es descarta la seva implementació.

6.2.2.7 – Dataset v9.0

Als estudis realitzats per a versions anteriors del dataset s'ha determinat que la classe `medium-intensity-wildfires` s'havia de descartar degut a la seva alta semblança a les de `low-intensity-wildfires`, i s'han pres les mesures adients a la versió 7. Tanmateix, ara podem treure avantatge d'aquest *drawback* (així com aprofitar el temps invertit en crear les zones de l'entorn UE per aquesta classe) si unim totes les imatges d'aquestes dues classes en una de sola que podem anomenar també com a `low-intensity-wildfires`.

Ara, també s'observa que les imatges del dataset v4 (imatges llunyanes) i les del dataset v7 (imatges properes) es podrien fusionar per a experimentar amb les possibles reaccions del model. Així, d'una banda ampliem la quantitat d'imatges total del dataset i de l'altra millorem la eficiència en la detecció d'incendis per part dels drons quan s'executi la PdC, ja que aquests seran més capaços de realitzar la classificació des d'una distància major (és a dir, tindran un major rang d'acció).

finalment, una altra cosa que es podria fer per a millorar el rendiment és la d'aplicar encara més tècniques d'ampliació.

En resum, ara podem emprendre les següents accions:

- Unir classes `low-intensity-wildfires` i `medium-intensity-wildfires` en una de sola
- Unir imatges del dataset v4 (llarga distància) i del dataset v7 (curta distància)
- Aplicar més tècniques d'ampliació
- ~~Fer un curat manual de les imatges~~[13]

[13] *Aquesta opció es descarta degut a que implica una inversió elevada de temps i tampoc és imprescindible assolir nivells més alts de rendiments per a la realització de la PdC*

Un cop realitzades les anteriors accions obtindrem un **dataset final** amb les següents característiques (tenint en compte les ampliacions realitzades "en calent" durant l'entrenament del model);

- **Mida de les imatges:** 512 x 512 x 3
- **Nombre total d'imatges:**

Nombre total d'imatges al subset d'entrenament	2592
Nombre total d'imatges al subset de validació	864
Nombre total d'imatges al subset d'entrenament	1248
Nombre total d'imatges	**4704**

Figura 6.x.x.x – *Xifres del dataset v9.0*

- **Tipus d'imatges:** llarga distància + curta distància
- **Nombre de classes**: 3
  - `High-intensity-wildfires`
  - `low-intensity-wildfires`
  - `no-wildfires`
- **Configuració d'hyperparàmetres:** S'apliquen els càlculs efectuats a la secció 5.2.3

Un cop aplicades les millores proposades mes amunt i fets els primers entrenaments de prova amb el nou algorisme, els resultats obtinguts[14] són dels millors obtinguts fins al moment:

[14] *Es poden consultar els resums d'entrenament al següent arxiu del repositori → `/bin/CNN/cnn-training_20211230-102755.tar.gz`*

```
*********************************************************************************
***                                                                           ***
***                      PoC's CNN Model Training Summary                     ***
***                                                                           ***
*********************************************************************************
***                                                                           ***
***                            Model version: v3.0                            ***
***                           Dataset version: v9.0                           ***
***                                                                           ***
*********************************************************************************

 TRAINING PARAMETERS:

 - Image size = (229 x 229 x 1)
 - Number of classes / labels = 3
 - Batch size = 32
 - Learning rate = 0.0001

*********************************************************************************

 DATASET TOTALS:

 - The number of images in a training set is:  2592
 - The number of images in a validation set is:  864
 - The number of images in a test set is:  1248
 - The number of batches per epoch is:  81

*********************************************************************************

 CNN BLUEPRINT:

  Network_v3(
   (conv1): Conv2d(1, 16, kernel_size=(9, 9), stride=(2, 2), padding=(2, 2))
   (bn1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   (conv2): Conv2d(16, 16, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2))
   (bn2): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   (pool): MaxPool2d(kernel_size=2, stride=1, padding=0, dilation=1, ceil_mode=False)
   (conv3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(2, 2))
   (bn3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   (conv4): Conv2d(32, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
   (bn4): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   (fc1): Linear(in_features=107648, out_features=3, bias=True)
)

*********************************************************************************

 MODEL TRAINING STATS:

 - Model trained on cuda:0 device
 - Training time:  1504.925375  seconds

*********************************************************************************

 MODEL TESTING STATS:

   - For epoch 1 the TEST accuracy over the whole TEST dataset is 72 %
   - For epoch 2 the TEST accuracy over the whole TEST dataset is 78 %
   - For epoch 3 the TEST accuracy over the whole TEST dataset is 67 %
   - For epoch 4 the TEST accuracy over the whole TEST dataset is 78 %
   - For epoch 5 the TEST accuracy over the whole TEST dataset is 81 %
   - For epoch 6 the TEST accuracy over the whole TEST dataset is 79 %
   - For epoch 7 the TEST accuracy over the whole TEST dataset is 81 %
   - For epoch 8 the TEST accuracy over the whole TEST dataset is 77 %
   - For epoch 9 the TEST accuracy over the whole TEST dataset is 82 %
   - For epoch 10 the TEST accuracy over the whole TEST dataset is 81 %

*********************************************************************************
```

Figura 6.x.x.x.a *– Rendiment obtingut al entrenar el model v3 amb les imatges del dataset v7.0 (part 1)*

```
********************************************************************************

 FINAL PREDICTION TEST:

 - Model tested on cuda:0 device
 - Final test accuracy: 82 %

+--------------------------+--------------------------+
| Real Label               | Predicted Label          |
+==========================+==========================+
| low-intensity-wildfires  | no-wildfires             |
+--------------------------+--------------------------+
| no-wildfires             | no-wildfires             |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| no-wildfires             | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| no-wildfires             | no-wildfires             |
+--------------------------+--------------------------+
| high-intensity-wildfires | high-intensity-wildfires |
+--------------------------+--------------------------+
| high-intensity-wildfires | high-intensity-wildfires |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| high-intensity-wildfires | low-intensity-wildfires  |
+--------------------------+--------------------------+
| high-intensity-wildfires | high-intensity-wildfires |
+--------------------------+--------------------------+
| low-intensity-wildfires  | no-wildfires             |
+--------------------------+--------------------------+
| high-intensity-wildfires | high-intensity-wildfires |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| no-wildfires             | no-wildfires             |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| high-intensity-wildfires | low-intensity-wildfires  |
+--------------------------+--------------------------+
| high-intensity-wildfires | high-intensity-wildfires |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
```

Figura 6.x.x.x.b *– Rendiment obtingut al entrenar el model v3 amb les imatges del dataset v7.0 (Part 2)*
S'observa un 18% d'errades en la predicció al fer un test final amb una mostra petita (25 imatges)

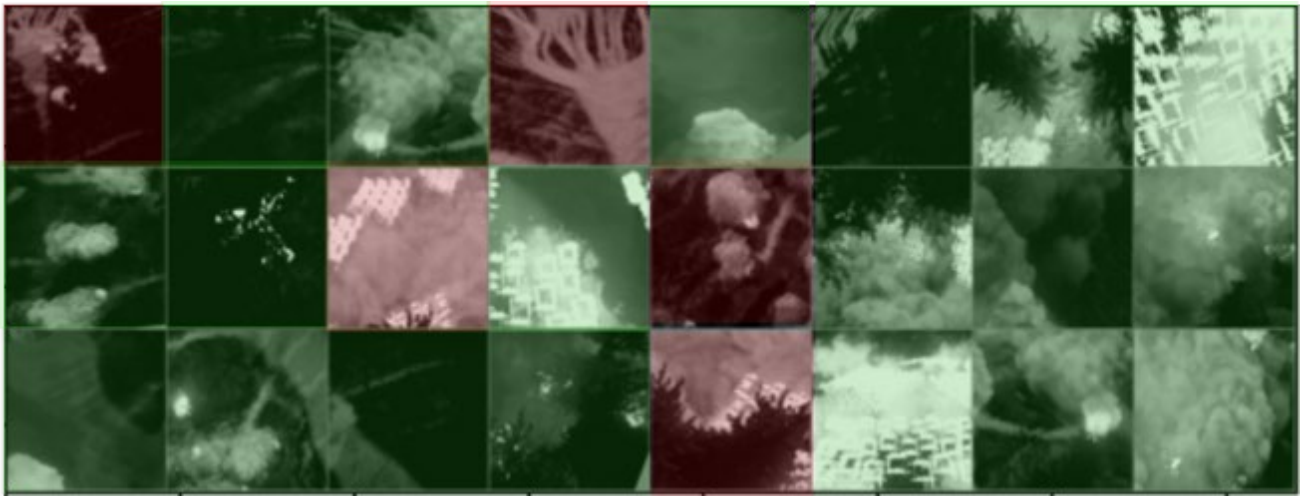


***Figura 6.x.x.x.c** – prediccions corresponents a la prova realitzada a la figura anterior*

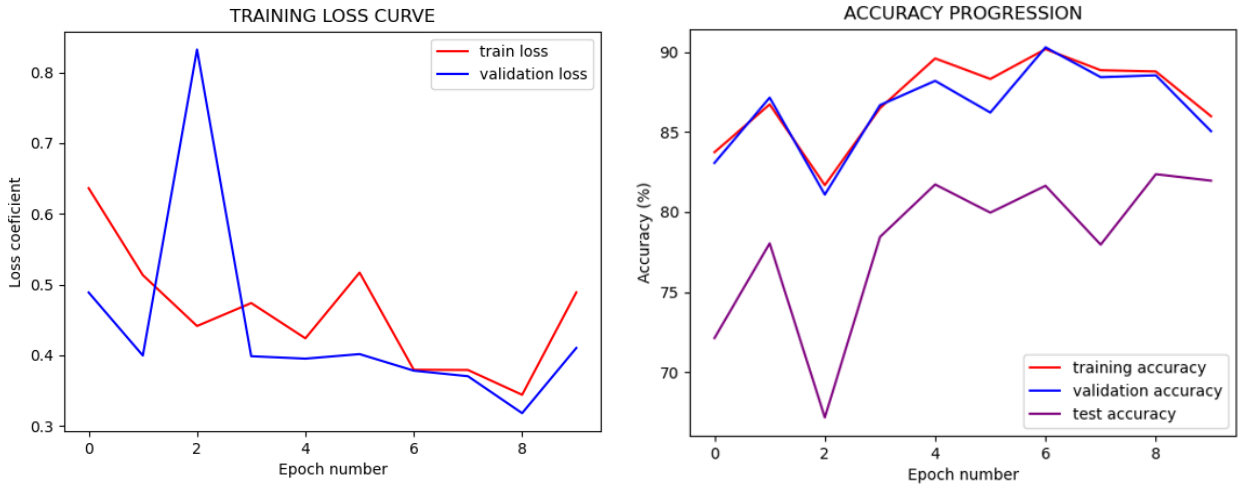


***Figura 6.x.x.x.d** – Gràfiques de pèrdua per als subset d'entrenament i validació (esquerra) i corbes d'aprenentatge amb els rendiments obtinguts a cada epoch sobre cadascun dels subsets de dades (dreta)*

6.3 – Entrenament del model amb el dataset seleccionat

Finalment, podem procedir a realitzar l'entrenament del model mitjançant l'script `pytorch_training.py`. Ara bé, cal tenir primer en compte que la **corba d'aprenentatge** de l'entrenament de prova realitzat a la secció 6.2.2.7 (figura 6.x.x.x) invita a pensar que, o bé la mida del `batch-size`, o bé la quantiat d'`epochs`, s'hauran de modificar abans de realitzar la proba, ja que no es veu una progressió creixent uniforme del rendiment (ans el contrari, a cada iteració els valors de rendiment pugen o baixen de manera aleatòria, és a dir, trobem un valor elevat de variança[15]). Això pot ser degut a què s'ha ampliat la mida del dataset fins a més de 4000 imatges, però es segueix alimentant el model amb lots de 32 imatges durant l'entrenament. Aquest fet redueix el grau d'aprofitament de la paral·lelització de recursos de GPU i en conseqüència la quantitat de càlculs realitzats i la quantitat de paràmetres reajustat a cada `epoch`. Tot i que cal observar que d'aquesta manera s'obtenen resultats amb menys epochs, s'ha de remarcar que aquests no són tan fiables degut a l'alta variança esmentada, fet que per exemple es manifesta en el moment de realitzar diferents proves de classificació contra el dataset de test amb una quantitat petita de mostres (la gran majoria d'experiments presenten diferents proporcions d'encert en les prediccions). Tanmateix, si ampliem la mida del batch a 256[16] permetem que la GPU efectuï més operacions en paral·lel, s'ajusten més paràmetres per `epoch` i en conseqüència obtenim un model consistent (amb una baixa **variança** en el rendiment obtingut al realitzar diferents prediccions contra el dataset de test, fet que implica obtenir el rendiment ofert pel model la majoria de vegades). Com a contrapartida però, tenim que ens caldrà realitzar més iteracions (`epochs`) per a aconseguir un model entrenat que ofereixi el millor rendiment possible, ja que al estar configurant tants paràmetres a l'hora haurem de fer més repeticions amb diferents arestes activades entre capes a cada iteració (això és el que és coneix com a convolucions), de manera que a cada epoch es configuraran paràmetres diferents (pesos d'aquestes arestes) als de la epoch anterior i el model acabarà podent detectar més característiques.

Fetes les consideracions anteriors, per a l'entrenament del model es procedeix a establir els següents canvis en la configuració de paràmetres d'entrenament:

- **Batch-size:** 256
- **Epochs:** 40

Un cop realitzat l'entrenament **obtenim un 86% de rendiment**[17] (els millors resultats fins al moment), fins i tot arribant a superar la fita marcada del 80% / 85%:

[15] *Explicar que és la variança en estadística blah blah blah*

[16] *S'ha provat amb 512 però la gestió de la memòria que fa pytorch no ho permet. Tanmateix, 256 és una mida que ja ens ofereix un bon resultat*

[17] Es poden consultar els resums d'entrenament al següent arxiu del repositori → `/bin/CNN/cnn-training_20211230-134145.tar.gz`

```
*************************************************************************
***                                                                   ***
***                    PoC's CNN Model Training Summary               ***
***                                                                   ***
*************************************************************************
***                                                                   ***
***                          Model version: v3.0                      ***
***                         Dataset version: v9.0                     ***
***                                                                   ***
*************************************************************************

 TRAINING PARAMETERS:

 - Image size = (229 x 229 x 1)
 - Number of classes / labels = 3
 - Batch size = 256
 - Learning rate = 0.0001

*************************************************************************

 DATASET TOTALS:

 - The number of images in a training set is:  2816
 - The number of images in a validation set is:  1024
 - The number of images in a test set is:  1280
 - The number of batches per epoch is:  11

*************************************************************************

 CNN BLUEPRINT:

  Network_v3(
   (conv1): Conv2d(1, 16, kernel_size=(9, 9), stride=(2, 2), padding=(2, 2))
   (bn1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   (conv2): Conv2d(16, 16, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2))
   (bn2): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   (pool): MaxPool2d(kernel_size=2, stride=1, padding=0, dilation=1, ceil_mode=False)
   (conv3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(2, 2))
   (bn3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   (conv4): Conv2d(32, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
   (bn4): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   (fc1): Linear(in_features=107648, out_features=3, bias=True)
)

*************************************************************************

 MODEL TRAINING STATS:

 - Model trained on cuda:0 device
 - Training time:  5257.3415  seconds

*************************************************************************

 MODEL TESTING STATS:

  - For epoch 1 the TEST accuracy over the whole TEST dataset is 50 %
  - For epoch 2 the TEST accuracy over the whole TEST dataset is 50 %
  - For epoch 3 the TEST accuracy over the whole TEST dataset is 54 %
  - For epoch 4 the TEST accuracy over the whole TEST dataset is 61 %
  - For epoch 5 the TEST accuracy over the whole TEST dataset is 71 %
  - For epoch 6 the TEST accuracy over the whole TEST dataset is 80 %
  - For epoch 7 the TEST accuracy over the whole TEST dataset is 80 %
  - For epoch 8 the TEST accuracy over the whole TEST dataset is 77 %
  - For epoch 9 the TEST accuracy over the whole TEST dataset is 77 %
  - For epoch 10 the TEST accuracy over the whole TEST dataset is 78 %
  - For epoch 11 the TEST accuracy over the whole TEST dataset is 79 %
  - For epoch 12 the TEST accuracy over the whole TEST dataset is 81 %
  - For epoch 13 the TEST accuracy over the whole TEST dataset is 83 %
  - For epoch 14 the TEST accuracy over the whole TEST dataset is 82 %
  - For epoch 15 the TEST accuracy over the whole TEST dataset is 80 %
  - For epoch 16 the TEST accuracy over the whole TEST dataset is 77 %
  - For epoch 17 the TEST accuracy over the whole TEST dataset is 75 %
  - For epoch 18 the TEST accuracy over the whole TEST dataset is 80 %
  - For epoch 19 the TEST accuracy over the whole TEST dataset is 80 %
  - For epoch 20 the TEST accuracy over the whole TEST dataset is 78 %
```

Figura 6.x.x.x.a *– Rendiment obtingut a l'entrenar el model v3.0 amb les imatges del dataset v9.0 utilitzant una mida de batch-size = 265 i 40 epoch (part 1)*

```
  - For epoch 21 the TEST accuracy over the whole TEST dataset is 81 %
  - For epoch 22 the TEST accuracy over the whole TEST dataset is 83 %
  - For epoch 23 the TEST accuracy over the whole TEST dataset is 84 %
  - For epoch 24 the TEST accuracy over the whole TEST dataset is 82 %
  - For epoch 25 the TEST accuracy over the whole TEST dataset is 78 %
  - For epoch 26 the TEST accuracy over the whole TEST dataset is 79 %
  - For epoch 27 the TEST accuracy over the whole TEST dataset is 84 %
  - For epoch 28 the TEST accuracy over the whole TEST dataset is 84 %
  - For epoch 29 the TEST accuracy over the whole TEST dataset is 83 %
  - For epoch 30 the TEST accuracy over the whole TEST dataset is 79 %
  - For epoch 31 the TEST accuracy over the whole TEST dataset is 82 %
  - For epoch 32 the TEST accuracy over the whole TEST dataset is 83 %
  - For epoch 33 the TEST accuracy over the whole TEST dataset is 82 %
  - For epoch 34 the TEST accuracy over the whole TEST dataset is 77 %
  - For epoch 35 the TEST accuracy over the whole TEST dataset is 82 %
  - For epoch 36 the TEST accuracy over the whole TEST dataset is 86 %
  - For epoch 37 the TEST accuracy over the whole TEST dataset is 86 %
  - For epoch 38 the TEST accuracy over the whole TEST dataset is 83 %
  - For epoch 39 the TEST accuracy over the whole TEST dataset is 82 %
  - For epoch 40 the TEST accuracy over the whole TEST dataset is 83 %

********************************************************************************

 FINAL PREDICTION TEST:

 - Model tested on cuda:0 device
 - Final test accuracy: 85 %

+--------------------------+--------------------------+
| Real Label               | Predicted Label          |
+==========================+==========================+
| low-intensity-wildfires  | high-intensity-wildfires |
+--------------------------+--------------------------+
| no-wildfires             | no-wildfires             |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| high-intensity-wildfires | high-intensity-wildfires |
+--------------------------+--------------------------+
| no-wildfires             | no-wildfires             |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| no-wildfires             | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| no-wildfires             | no-wildfires             |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| high-intensity-wildfires | high-intensity-wildfires |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| no-wildfires             | no-wildfires             |
+--------------------------+--------------------------+
| low-intensity-wildfires  | low-intensity-wildfires  |
+--------------------------+--------------------------+
| low-intensity-wildfires  | no-wildfires             |
+--------------------------+--------------------------+
| no-wildfires             | no-wildfires             |
+--------------------------+--------------------------+
| no-wildfires             | no-wildfires             |
+--------------------------+--------------------------+
| no-wildfires             | no-wildfires             |
+--------------------------+--------------------------+
```

Figura 6.x.x.x.b *– Rendiment obtingut al entrenar el model v3 amb les imatges del dataset v9.0 utilitzant una mida de batch-size = 265 i 40 epoch (Part 2). S'observa un 15% d'errades en la predicció al fer un test final amb una mostra petita (25 imatges)*

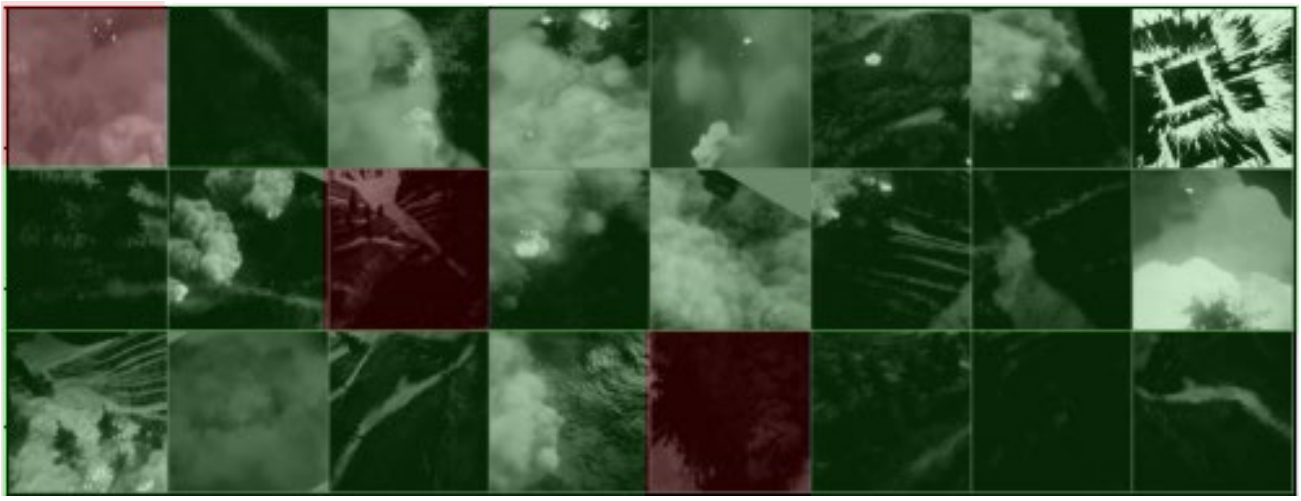


***Figura 6.x.x.x.c** – prediccions corresponents a la prova realitzada a la figura anterior*

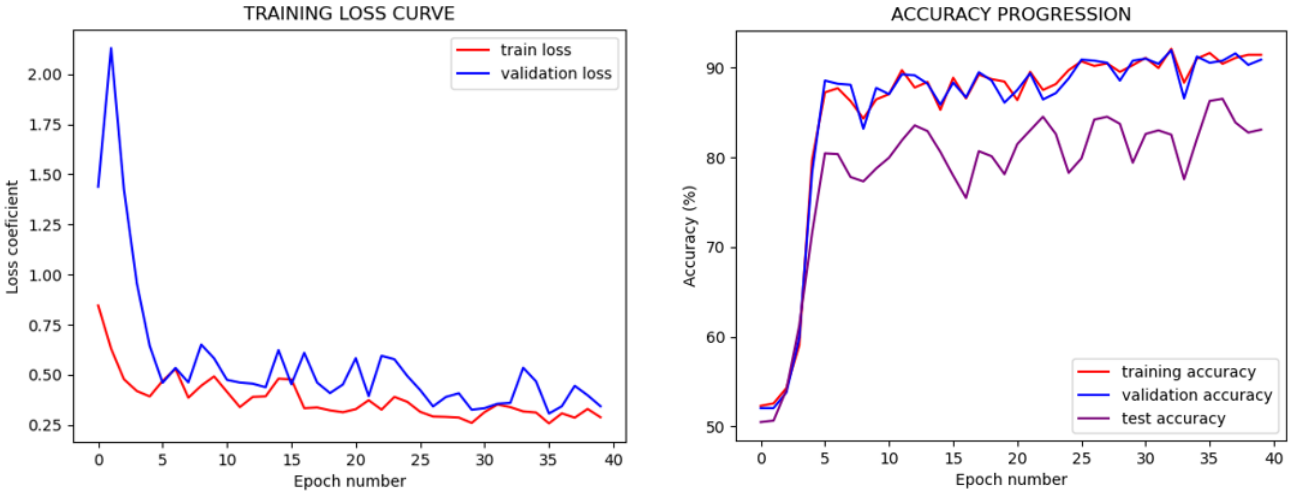


***Figura 6.x.x.x.d** – Gràfiques de pèrdua per als subset d'entrenament i validació (esquerra) i corbes d'aprenentatge amb els rendiments obtinguts a cada epoch sobre cadascun dels subsets de dades (dreta)*

6.4 - Avaluació del model i interpretació del seu rendiment (estudi del rati d'aprenentatge)

TO-DO

6.5 – ANNEX OPCIONAL: Millores en el rendiment (optimització de paràmetres)

TO-DO → posar aquí l'excel amb els resultats dels entrenaments (paràmetres-entrenaments.xlsx)