PROCESSAMENT D'IMATGES

Laboratori 6 - Report

curs 2023-24

Universitat Pompeu Fabra Barcelona

Xavier Riera - 268446

Pau Noguera - 265340

Introducció:

L'objectiu d'aquest laboratori és, mitjançant l'ús d'una xarxa neuronal convolucional i un data set format per imatges de mascotes, segmentar una imatge en dues regions: la regió formada per la mascota i la regió formada pel fons de la imatge.

El primer pas és definir la xarxa neuronal. Donat que les imatges a processar estan formades per milers de píxels, seria necessària una xarxa amb milers de neurones i milers de paràmetres. Ja que els nostres ordinadors no tenen la capacitat de treballar amb una quantitat tan elevada de paràmetres, implementarem una xarxa convolucional que redueix significativament el nombre de paràmetres a optimitzar i, conseqüentment, la velocitat de processament de les imatges.

El primer pas és separar el data set (format per imatges d'animals) en *training dataset* i *validation dataset*. D'aquesta manera, estem separant el dataset en dos grups: un grup d'imatges per entrenar els paràmetres de la xarxa i un altre grup d'imatges per comprovar si la xarxa és capaç de realitzar prediccions amb imatges completament noves.

Una vegada separat el dataset, és necessari definir els hiperparàmetres amb els quals la xarxa "aprendrà" a optimitzar els paràmetres. Aquests hiperparàmetres determinen característiques de la xarxa que no són modificades a mesura que la xarxa és entrenada i, per tant, és necessari inicialitzar-los amb uns valors adequats i que maximitzin la precisió de la xarxa. En aquest laboratori, a més d'usar diferentes imatges d'Internet per veure si la xarxa és capaç de segmentar la imatge adientement, usarem diversos valors per als hiperparàmetres per tal de determinar empíricament quins són els valors més adequats.

Cal esmentar que cada una de les imatges del dataset té una imatge complementària anomenada *True Mask*. Aquesta imatge és el resultat ideal que volem que la xarxa sigui capaç d'aconseguir i han estat realitzades manualment. Per això, una vegada feta la predicció d'imatges del dataset, podem calcular quin és l'error de la predicció usant el *cross entropy error*. Malgrat, ja que les nostres imatges són d'Internet i no tenim una imatge que determini la predicció ideal que volem obtenir, no podem determinar numèricament l'error de la predicció i ho farem qualitativament.

Els hiperparàmetres que modificarem són: el nombre d'iteracions (epochs) que realitza l'algorisme per entrenar, el learning rate (valor que determina com es modifiquen els pesos una vegada es calcula el seu gradient; un learning rate més baix és més precís però requereix un major nombre d'iteracions i amb un learning rate més alt s'obtenen resultats menys precisos però amb menys iteracions) i steps per epoch (aquest valor determina amb quina quantitat de batchs s'entrena la xarxa a cada iteració; on els batchs són conjunts de datapoints que ens permeten entrenar la xarxa sample a sample).

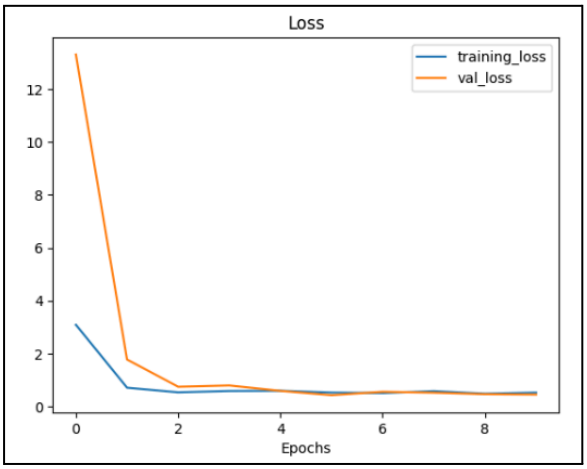
La imatge que usarem per veure si millorem el model és la següent:

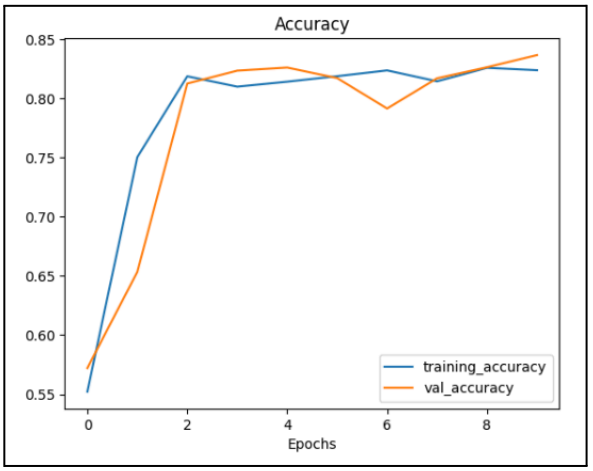
Num. epochs 10 →

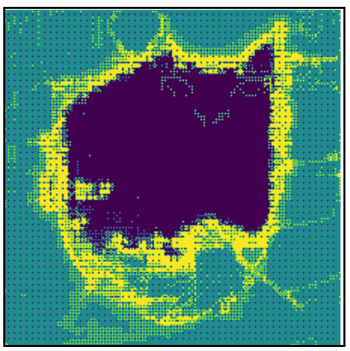
En primer lloc, hem realitzat sis casos mantenint el nombre d'epochs a 10 i variant la resta d'hiperparàmetres. En els tres primers casos, el nombre d'steps per epoch s'ha mantingut igual (115) i s'han usat tres learning rates diferents.

1. Per a un learning rate de 0,1 hem obtingut:

```
Epoch 1/10
115/115 [==============================] - 43s 236ms/step - loss: 3.0898 - accuracy: 0.5522 - val_loss: 13.3156 - val_accuracy: 0.5721
Epoch 2/10
115/115 [==============================] - 22s 190ms/step - loss: 0.7095 - accuracy: 0.7502 - val_loss: 1.7746 - val_accuracy: 0.6533
Epoch 3/10
115/115 [==============================] - 22s 189ms/step - loss: 0.5332 - accuracy: 0.8185 - val_loss: 0.7452 - val_accuracy: 0.8124
Epoch 4/10
115/115 [==============================] - 21s 187ms/step - loss: 0.5845 - accuracy: 0.8097 - val_loss: 0.7948 - val_accuracy: 0.8232
Epoch 5/10
115/115 [==============================] - 21s 186ms/step - loss: 0.5928 - accuracy: 0.8139 - val_loss: 0.5894 - val_accuracy: 0.8259
Epoch 6/10
115/115 [==============================] - 21s 184ms/step - loss: 0.5242 - accuracy: 0.8185 - val_loss: 0.4251 - val_accuracy: 0.8169
Epoch 7/10
115/115 [==============================] - 22s 188ms/step - loss: 0.5049 - accuracy: 0.8235 - val_loss: 0.5546 - val_accuracy: 0.7913
Epoch 8/10
115/115 [==============================] - 22s 188ms/step - loss: 0.5792 - accuracy: 0.8142 - val_loss: 0.5203 - val_accuracy: 0.8168
Epoch 9/10
115/115 [==============================] - 24s 209ms/step - loss: 0.4817 - accuracy: 0.8256 - val_loss: 0.4631 - val_accuracy: 0.8261
Epoch 10/10
115/115 [==============================] - 21s 187ms/step - loss: 0.5224 - accuracy: 0.8236 - val_loss: 0.4488 - val_accuracy: 0.8364
```

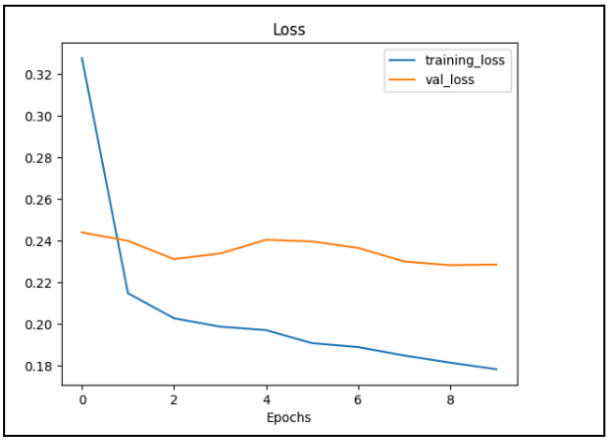


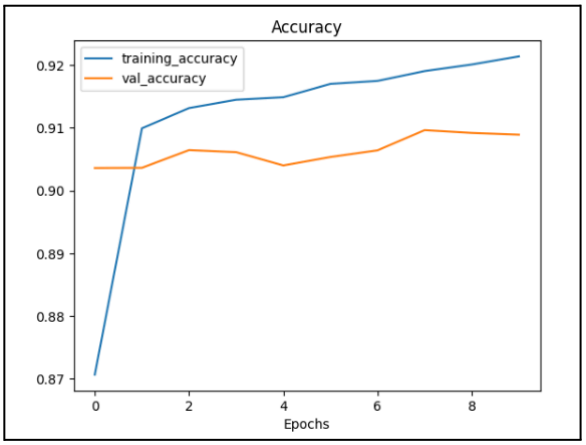


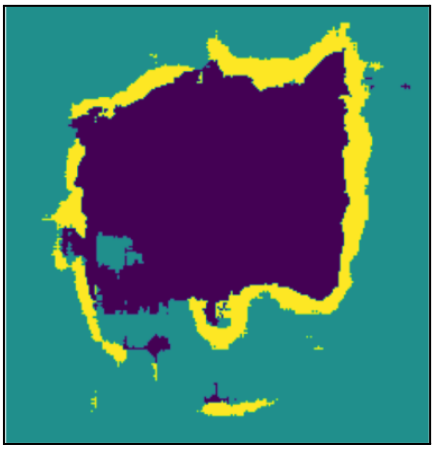


2. Per a un learning rate de 0,001 hem obtingut:

```
Epoch 1/10
115/115 [==============================] - 26s 187ms/step - loss: 0.3275 - accuracy: 0.8707 - val_loss: 0.2440 - val_accuracy: 0.9036
Epoch 2/10
115/115 [==============================] - 21s 182ms/step - loss: 0.2148 - accuracy: 0.9099 - val_loss: 0.2399 - val_accuracy: 0.9036
Epoch 3/10
115/115 [==============================] - 21s 185ms/step - loss: 0.2028 - accuracy: 0.9132 - val_loss: 0.2312 - val_accuracy: 0.9065
Epoch 4/10
115/115 [==============================] - 21s 186ms/step - loss: 0.1988 - accuracy: 0.9145 - val_loss: 0.2339 - val_accuracy: 0.9061
Epoch 5/10
115/115 [==============================] - 21s 182ms/step - loss: 0.1971 - accuracy: 0.9149 - val_loss: 0.2404 - val_accuracy: 0.9040
Epoch 6/10
115/115 [==============================] - 21s 180ms/step - loss: 0.1909 - accuracy: 0.9170 - val_loss: 0.2396 - val_accuracy: 0.9054
Epoch 7/10
115/115 [==============================] - 24s 205ms/step - loss: 0.1890 - accuracy: 0.9175 - val_loss: 0.2366 - val_accuracy: 0.9064
Epoch 8/10
115/115 [==============================] - 21s 184ms/step - loss: 0.1850 - accuracy: 0.9191 - val_loss: 0.2301 - val_accuracy: 0.9096
Epoch 9/10
115/115 [==============================] - 21s 187ms/step - loss: 0.1815 - accuracy: 0.9201 - val_loss: 0.2283 - val_accuracy: 0.9092
Epoch 10/10
115/115 [==============================] - 21s 183ms/step - loss: 0.1784 - accuracy: 0.9214 - val_loss: 0.2286 - val_accuracy: 0.9089
```

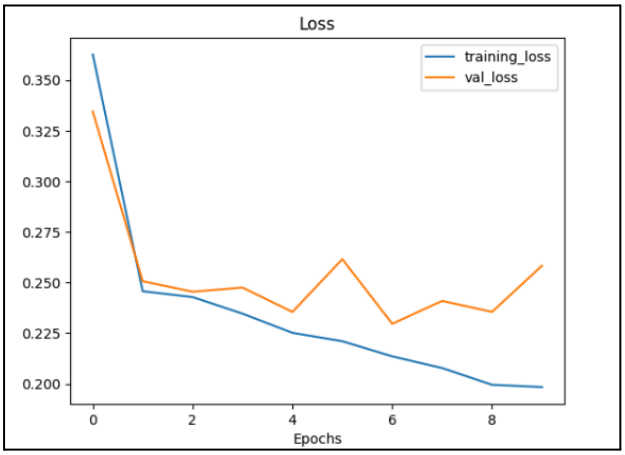


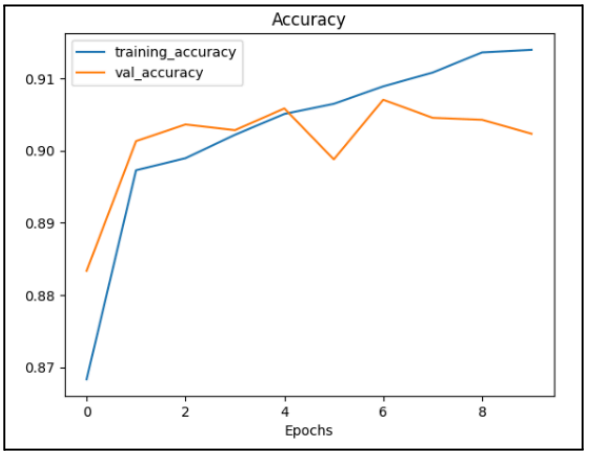


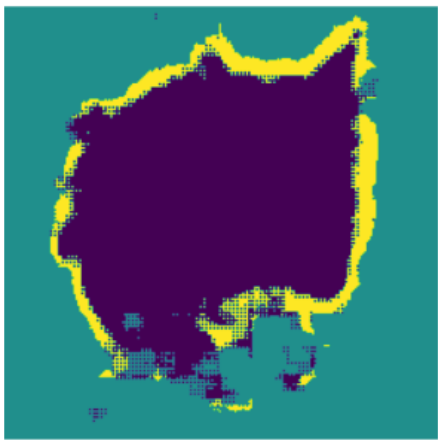


3. Per a un learning rate que decreix exponencialment hem obtingut:

```
Epoch 1/10
115/115 [==============================] - 26s 192ms/step - loss: 0.3626 - accuracy: 0.8683 - val_loss: 0.3345 - val_accuracy: 0.8833
Epoch 2/10
115/115 [==============================] - 21s 182ms/step - loss: 0.2457 - accuracy: 0.8973 - val_loss: 0.2506 - val_accuracy: 0.9013
Epoch 3/10
115/115 [==============================] - 21s 186ms/step - loss: 0.2428 - accuracy: 0.8989 - val_loss: 0.2454 - val_accuracy: 0.9036
Epoch 4/10
115/115 [==============================] - 20s 177ms/step - loss: 0.2346 - accuracy: 0.9022 - val_loss: 0.2475 - val_accuracy: 0.9028
Epoch 5/10
115/115 [==============================] - 21s 187ms/step - loss: 0.2252 - accuracy: 0.9050 - val_loss: 0.2355 - val_accuracy: 0.9058
Epoch 6/10
115/115 [==============================] - 20s 177ms/step - loss: 0.2210 - accuracy: 0.9065 - val_loss: 0.2615 - val_accuracy: 0.8988
Epoch 7/10
115/115 [==============================] - 24s 209ms/step - loss: 0.2135 - accuracy: 0.9089 - val_loss: 0.2296 - val_accuracy: 0.9070
Epoch 8/10
115/115 [==============================] - 21s 179ms/step - loss: 0.2078 - accuracy: 0.9108 - val_loss: 0.2409 - val_accuracy: 0.9045
Epoch 9/10
115/115 [==============================] - 22s 188ms/step - loss: 0.1995 - accuracy: 0.9136 - val_loss: 0.2355 - val_accuracy: 0.9042
Epoch 10/10
115/115 [==============================] - 21s 182ms/step - loss: 0.1983 - accuracy: 0.9139 - val_loss: 0.2583 - val_accuracy: 0.9023
```





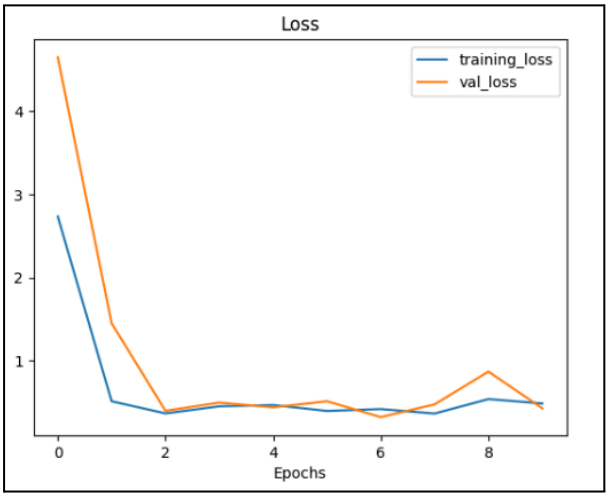


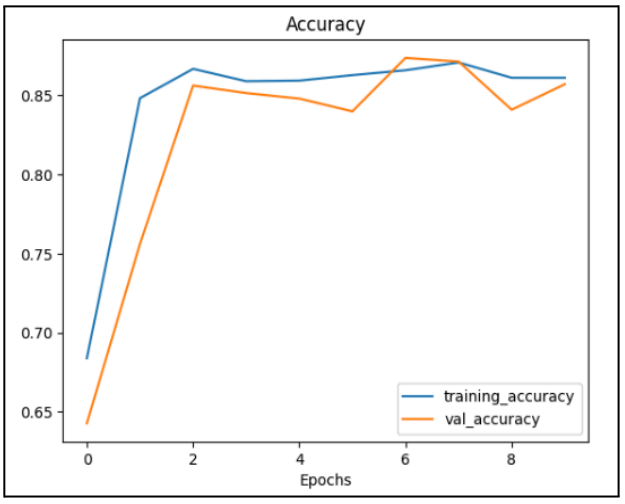
Com podem veure, en el primer cas (learning rate de 0,1) la xarxa ha acabat amb un error molt proper a 0, encara que la precisió ha estat la més baixa dels tres casos. A més, en els altres dos casos, l'accuracy del training data set és molt més elevada que la del validation set, indicant que la xarxa té un problema d'overfitting. No obstant, si miram les segmentacions obtingudes podem veure que la xarxa sí que és capaç de separar les mascotes del fons de la imatge.

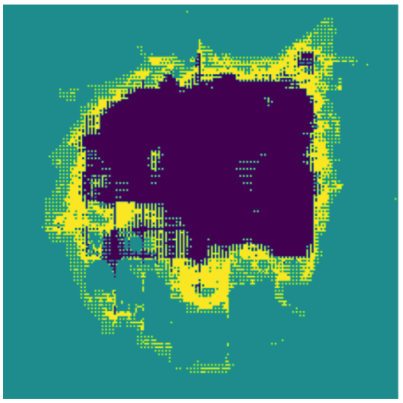
A continuació, modificarem l'hiperparàmetre d'steps per epoch i tornarem a realitzar els tres casos igual que abans:

4. Per a un learning rate de 0,1 hem obtingut:

```
Epoch 1/10
80/80 [==============================] - 40s 310ms/step - loss: 2.7360 - accuracy: 0.6840 - val_loss: 4.6504 - val_accuracy: 0.6428
Epoch 2/10
80/80 [==============================] - 16s 198ms/step - loss: 0.5141 - accuracy: 0.8482 - val_loss: 1.4518 - val_accuracy: 0.7562
Epoch 3/10
80/80 [==============================] - 15s 188ms/step - loss: 0.3677 - accuracy: 0.8668 - val_loss: 0.3992 - val_accuracy: 0.8562
Epoch 4/10
80/80 [==============================] - 16s 198ms/step - loss: 0.4536 - accuracy: 0.8590 - val_loss: 0.4978 - val_accuracy: 0.8515
Epoch 5/10
80/80 [==============================] - 15s 184ms/step - loss: 0.4686 - accuracy: 0.8593 - val_loss: 0.4409 - val_accuracy: 0.8480
Epoch 6/10
80/80 [==============================] - 15s 185ms/step - loss: 0.3974 - accuracy: 0.8628 - val_loss: 0.5146 - val_accuracy: 0.8399
Epoch 7/10
80/80 [==============================] - 16s 196ms/step - loss: 0.4198 - accuracy: 0.8659 - val_loss: 0.3238 - val_accuracy: 0.8737
Epoch 8/10
80/80 [==============================] - 19s 233ms/step - loss: 0.3663 - accuracy: 0.8708 - val_loss: 0.4767 - val_accuracy: 0.8714
Epoch 9/10
80/80 [==============================] - 16s 201ms/step - loss: 0.5401 - accuracy: 0.8611 - val_loss: 0.8707 - val_accuracy: 0.8409
Epoch 10/10
80/80 [==============================] - 16s 197ms/step - loss: 0.4875 - accuracy: 0.8611 - val_loss: 0.4298 - val_accuracy: 0.8571
```

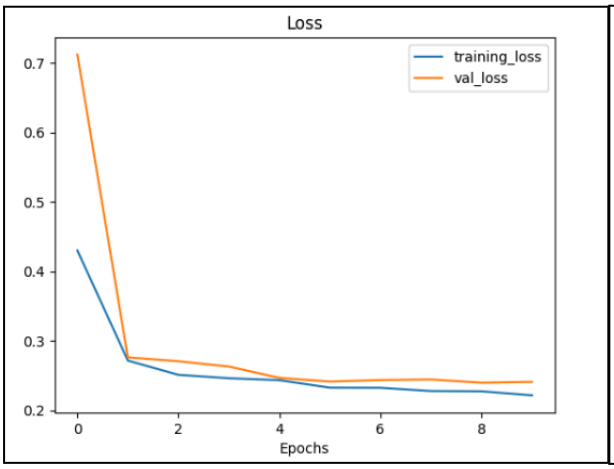


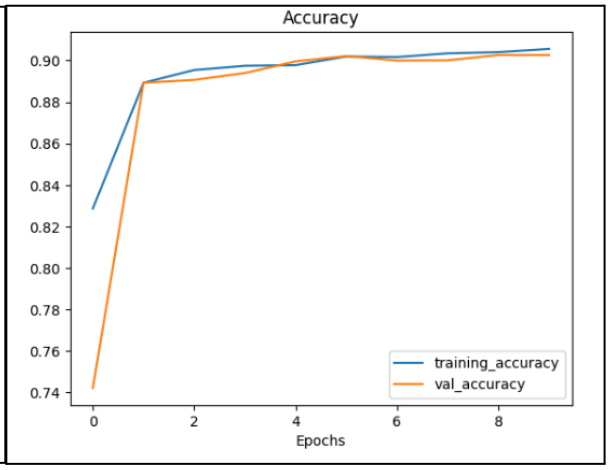


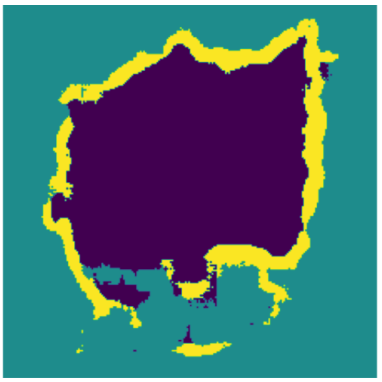


5. Per a un learning rate de 0,001 hem obtingut:

```
Epoch 1/10
80/80 [==============================] - 45s 275ms/step - loss: 0.4299 - accuracy: 0.8287 - val_loss: 0.7118 - val_accuracy: 0.7422
Epoch 2/10
80/80 [==============================] - 15s 190ms/step - loss: 0.2714 - accuracy: 0.8893 - val_loss: 0.2760 - val_accuracy: 0.8893
Epoch 3/10
80/80 [==============================] - 16s 203ms/step - loss: 0.2509 - accuracy: 0.8955 - val_loss: 0.2704 - val_accuracy: 0.8907
Epoch 4/10
80/80 [==============================] - 15s 192ms/step - loss: 0.2460 - accuracy: 0.8975 - val_loss: 0.2628 - val_accuracy: 0.8940
Epoch 5/10
80/80 [==============================] - 15s 193ms/step - loss: 0.2432 - accuracy: 0.8978 - val_loss: 0.2465 - val_accuracy: 0.8996
Epoch 6/10
80/80 [==============================] - 16s 205ms/step - loss: 0.2324 - accuracy: 0.9020 - val_loss: 0.2412 - val_accuracy: 0.9022
Epoch 7/10
80/80 [==============================] - 19s 234ms/step - loss: 0.2323 - accuracy: 0.9017 - val_loss: 0.2434 - val_accuracy: 0.9000
Epoch 8/10
80/80 [==============================] - 15s 191ms/step - loss: 0.2276 - accuracy: 0.9035 - val_loss: 0.2442 - val_accuracy: 0.9001
Epoch 9/10
80/80 [==============================] - 16s 203ms/step - loss: 0.2272 - accuracy: 0.9041 - val_loss: 0.2396 - val_accuracy: 0.9027
Epoch 10/10
80/80 [==============================] - 15s 191ms/step - loss: 0.2214 - accuracy: 0.9056 - val_loss: 0.2408 - val_accuracy: 0.9027
```

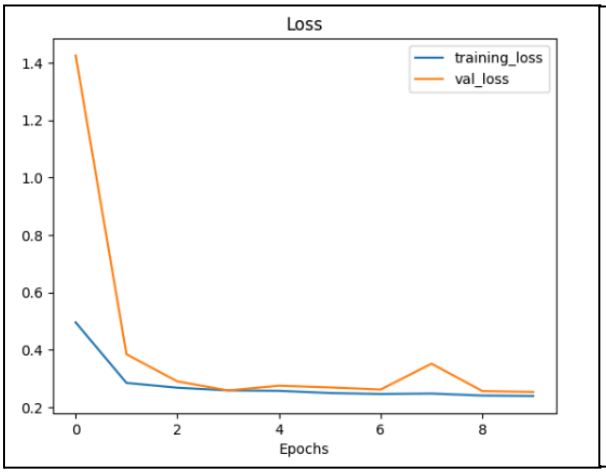


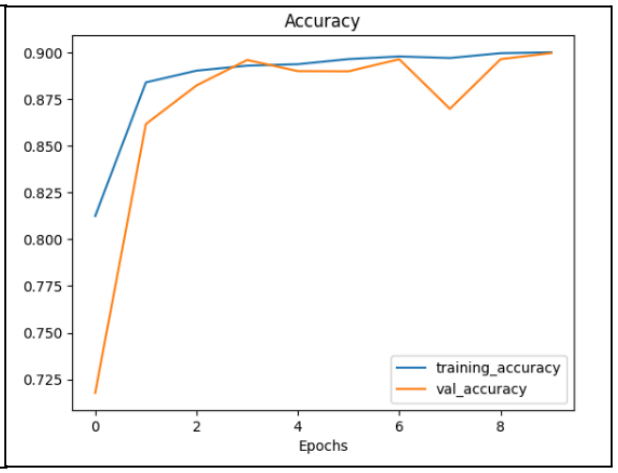


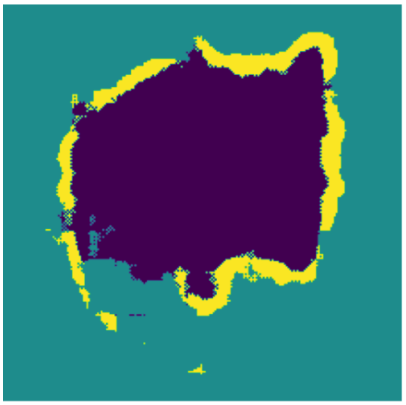


6. Per a un learning rate que decreix exponencialment hem obtingut:

```
Epoch 1/10
80/80 [==============================] - 40s 325ms/step - loss: 0.4949 - accuracy: 0.8125 - val_loss: 1.4247 - val_accuracy: 0.7178
Epoch 2/10
80/80 [==============================] - 20s 246ms/step - loss: 0.2840 - accuracy: 0.8840 - val_loss: 0.3840 - val_accuracy: 0.8617
Epoch 3/10
80/80 [==============================] - 17s 208ms/step - loss: 0.2675 - accuracy: 0.8903 - val_loss: 0.2894 - val_accuracy: 0.8824
Epoch 4/10
80/80 [==============================] - 16s 197ms/step - loss: 0.2582 - accuracy: 0.8930 - val_loss: 0.2580 - val_accuracy: 0.8960
Epoch 5/10
80/80 [==============================] - 17s 209ms/step - loss: 0.2566 - accuracy: 0.8937 - val_loss: 0.2745 - val_accuracy: 0.8900
Epoch 6/10
80/80 [==============================] - 17s 213ms/step - loss: 0.2490 - accuracy: 0.8965 - val_loss: 0.2685 - val_accuracy: 0.8899
Epoch 7/10
80/80 [==============================] - 16s 205ms/step - loss: 0.2456 - accuracy: 0.8979 - val_loss: 0.2612 - val_accuracy: 0.8964
Epoch 8/10
80/80 [==============================] - 16s 198ms/step - loss: 0.2472 - accuracy: 0.8971 - val_loss: 0.3512 - val_accuracy: 0.8699
Epoch 9/10
80/80 [==============================] - 16s 197ms/step - loss: 0.2401 - accuracy: 0.8997 - val_loss: 0.2559 - val_accuracy: 0.8964
Epoch 10/10
80/80 [==============================] - 17s 209ms/step - loss: 0.2386 - accuracy: 0.9001 - val_loss: 0.2526 - val_accuracy: 0.8997
```





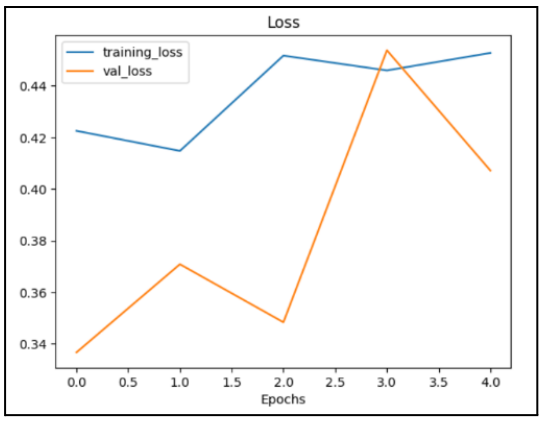


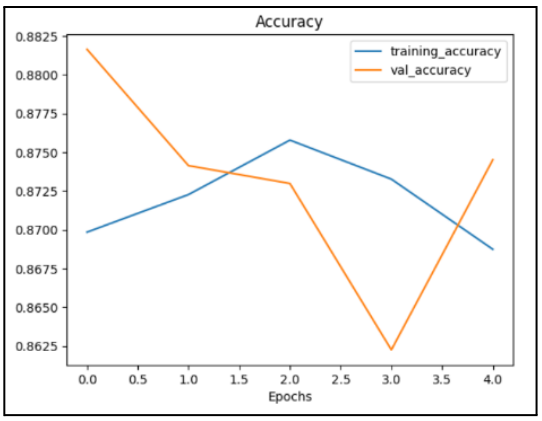
Num. epochs 5 →

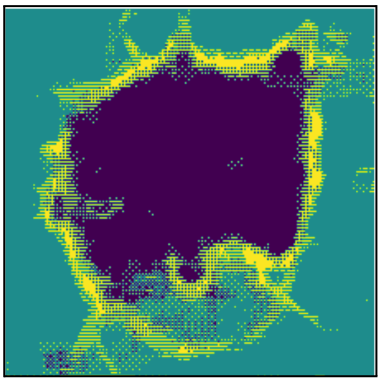
En segon lloc, per a sis exemples més, hem canviat el nombre de l'epoch a 5 i modifiquem els hiperparàmetres de la mateixa manera els sis casos anteriors. Comencem per un nombre de steps de 115 (pels tres primers) i variarem el learning rate en cada cas.

7. Per a un learning rate de 0,1 hem obtingut:

```
Epoch 1/5
115/115 [==============================] - 21s 178ms/step - loss: 0.4226 - accuracy: 0.8698 - val_loss: 0.3367 - val_accuracy: 0.8816
Epoch 2/5
115/115 [==============================] - 22s 193ms/step - loss: 0.4148 - accuracy: 0.8723 - val_loss: 0.3709 - val_accuracy: 0.8741
Epoch 3/5
115/115 [==============================] - 23s 196ms/step - loss: 0.4517 - accuracy: 0.8758 - val_loss: 0.3484 - val_accuracy: 0.8730
Epoch 4/5
115/115 [==============================] - 21s 187ms/step - loss: 0.4460 - accuracy: 0.8733 - val_loss: 0.4538 - val_accuracy: 0.8623
Epoch 5/5
115/115 [==============================] - 25s 214ms/step - loss: 0.4527 - accuracy: 0.8687 - val_loss: 0.4072 - val_accuracy: 0.8745
```

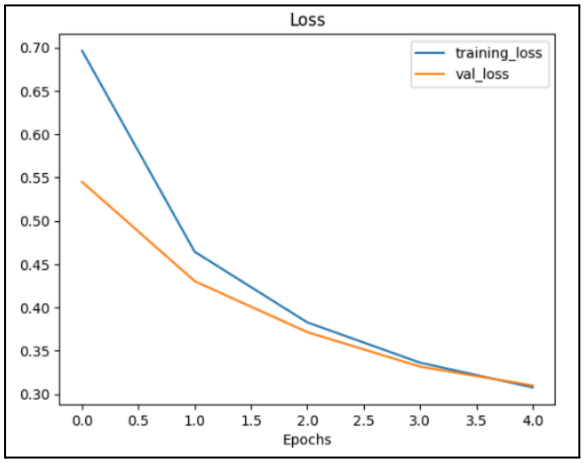


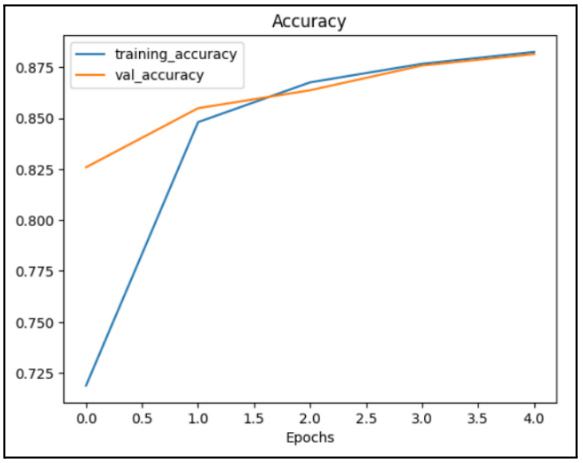


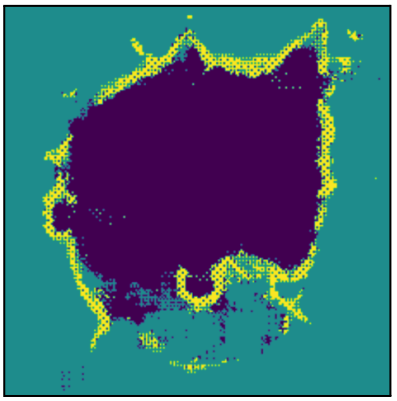


8. Per un learning rate de 0,001 obtenim:

```
Epoch 1/5
115/115 [==============================] - 30s 204ms/step - loss: 0.6963 - accuracy: 0.7187 - val_loss: 0.5449 - val_accuracy: 0.8259
Epoch 2/5
115/115 [==============================] - 22s 188ms/step - loss: 0.4642 - accuracy: 0.8481 - val_loss: 0.4303 - val_accuracy: 0.8549
Epoch 3/5
115/115 [==============================] - 25s 216ms/step - loss: 0.3828 - accuracy: 0.8676 - val_loss: 0.3715 - val_accuracy: 0.8637
Epoch 4/5
115/115 [==============================] - 22s 187ms/step - loss: 0.3363 - accuracy: 0.8767 - val_loss: 0.3318 - val_accuracy: 0.8759
Epoch 5/5
115/115 [==============================] - 22s 190ms/step - loss: 0.3078 - accuracy: 0.8825 - val_loss: 0.3100 - val_accuracy: 0.8815
```

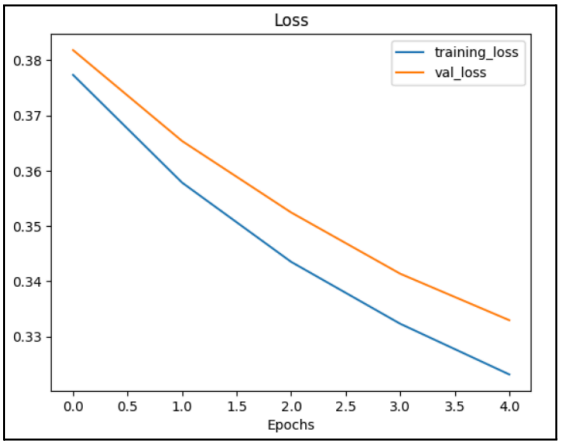


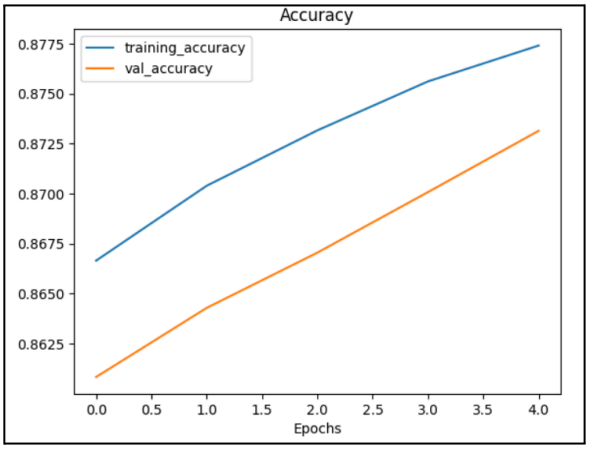


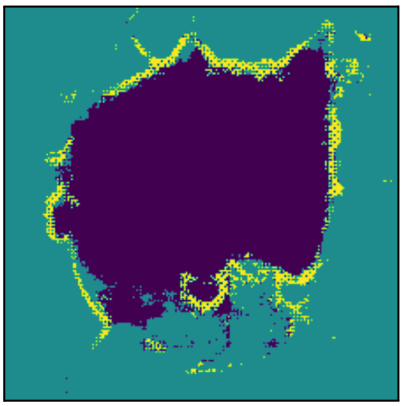


9. Per a un learning rate que decreix exponencialment hem obtingut:

```
Epoch 1/5
115/115 [==============================] - 23s 199ms/step - loss: 0.3773 - accuracy: 0.8666 - val_loss: 0.3818 - val_accuracy: 0.8608
Epoch 2/5
115/115 [==============================] - 23s 197ms/step - loss: 0.3578 - accuracy: 0.8704 - val_loss: 0.3654 - val_accuracy: 0.8643
Epoch 3/5
115/115 [==============================] - 22s 193ms/step - loss: 0.3435 - accuracy: 0.8732 - val_loss: 0.3524 - val_accuracy: 0.8670
Epoch 4/5
115/115 [==============================] - 21s 186ms/step - loss: 0.3323 - accuracy: 0.8756 - val_loss: 0.3413 - val_accuracy: 0.8701
Epoch 5/5
115/115 [==============================] - 22s 196ms/step - loss: 0.3231 - accuracy: 0.8774 - val_loss: 0.3329 - val_accuracy: 0.8731
```





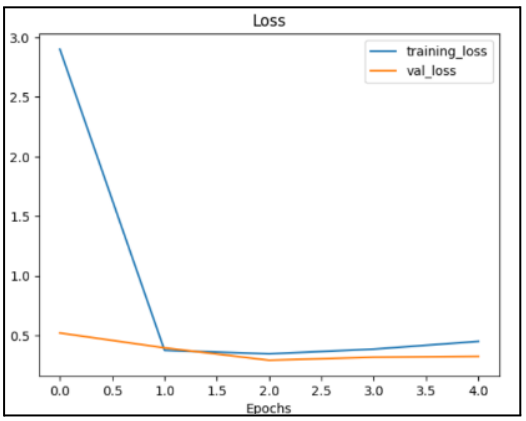


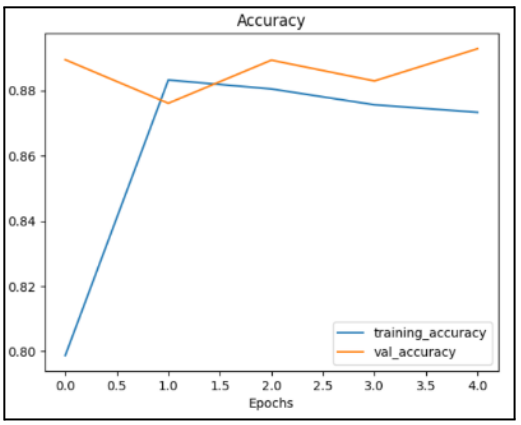
Com podem veure, en el primer cas (learning rate de 0,1) la xarxa ha tingut un error irregular i no descendent, probablement degut a les poques iteracions i el valor relativament alt del learning rate, el qual, no han permès que l'algortime es minimitzi i provocant que sigui la pitjor segmentació de tots els exemples. En els altres dos casos, l'error és decreixent i els dos resultats són molt similars, per tant, la xarxa sí que és capaç de separar les mascotes del fons de la imatge.

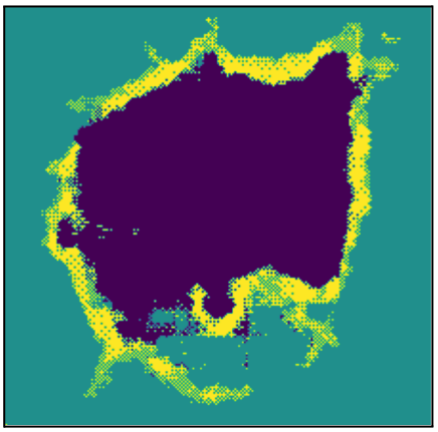
Finalment, modificarem l'hiperparàmetre d'steps per epoch (a 80) i ho tornarem a realitzar pels tres learning rates anteriors.

10. Per a un learning rate de 0,1:

```
Epoch 1/5
80/80 [==============================] - 23s 219ms/step - loss: 2.8992 - accuracy: 0.7987 - val_loss: 0.5201 - val_accuracy: 0.8895
Epoch 2/5
80/80 [==============================] - 16s 195ms/step - loss: 0.3746 - accuracy: 0.8833 - val_loss: 0.3957 - val_accuracy: 0.8761
Epoch 3/5
80/80 [==============================] - 15s 193ms/step - loss: 0.3452 - accuracy: 0.8805 - val_loss: 0.2916 - val_accuracy: 0.8893
Epoch 4/5
80/80 [==============================] - 16s 205ms/step - loss: 0.3847 - accuracy: 0.8756 - val_loss: 0.3173 - val_accuracy: 0.8830
Epoch 5/5
80/80 [==============================] - 19s 237ms/step - loss: 0.4498 - accuracy: 0.8733 - val_loss: 0.3241 - val_accuracy: 0.8928
```

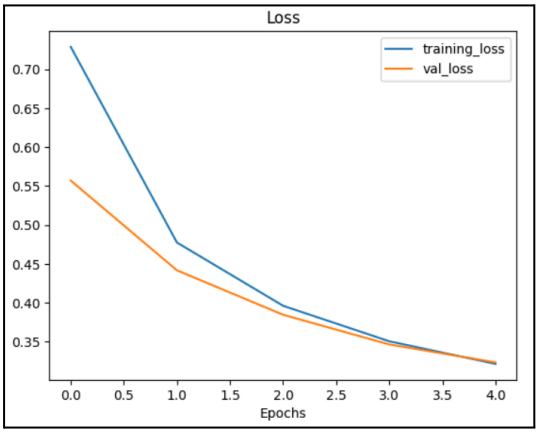


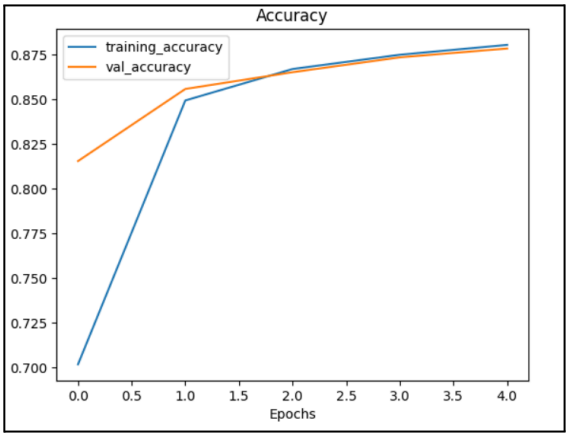


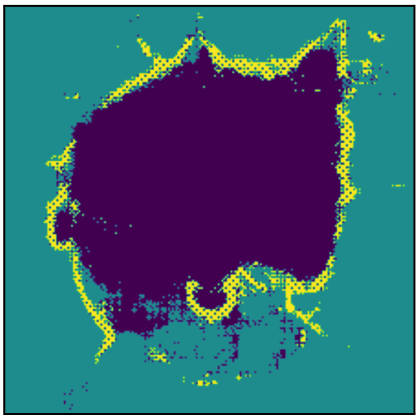


11. Per un learning rate de 0,001 obtenim:

```
Epoch 1/5
80/80 [==============================] - 20s 194ms/step - loss: 0.7287 - accuracy: 0.7016 - val_loss: 0.5570 - val_accuracy: 0.8153
Epoch 2/5
80/80 [==============================] - 15s 187ms/step - loss: 0.4774 - accuracy: 0.8492 - val_loss: 0.4417 - val_accuracy: 0.8557
Epoch 3/5
80/80 [==============================] - 18s 229ms/step - loss: 0.3962 - accuracy: 0.8668 - val_loss: 0.3847 - val_accuracy: 0.8650
Epoch 4/5
80/80 [==============================] - 16s 199ms/step - loss: 0.3506 - accuracy: 0.8748 - val_loss: 0.3465 - val_accuracy: 0.8734
Epoch 5/5
80/80 [==============================] - 15s 187ms/step - loss: 0.3216 - accuracy: 0.8803 - val_loss: 0.3235 - val_accuracy: 0.8783
```

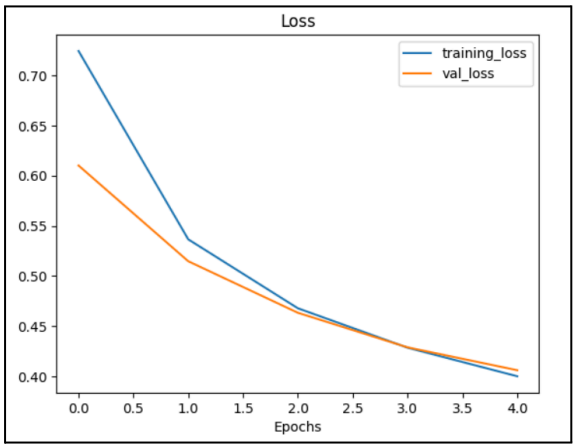


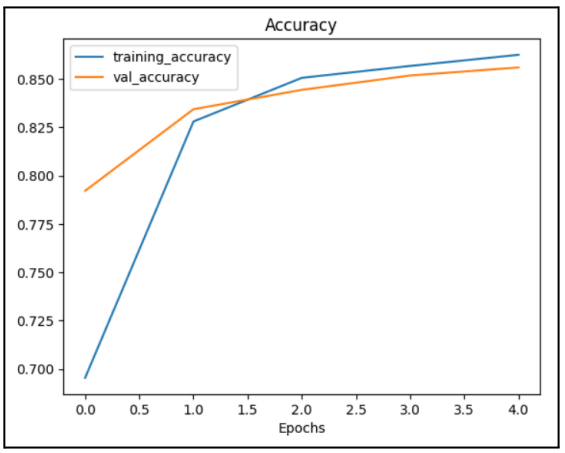


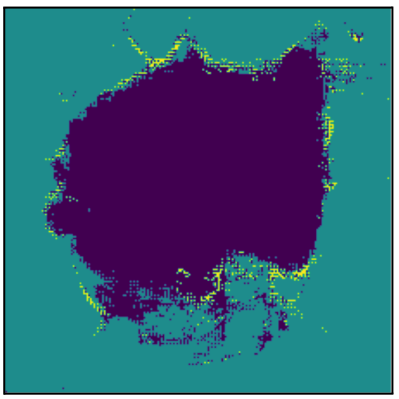


12. Per a un learning rate que decreix exponencialment hem obtingut:

```
Epoch 1/5
80/80 [==============================] - 21s 219ms/step - loss: 0.7244 - accuracy: 0.6954 - val_loss: 0.6102 - val_accuracy: 0.7921
Epoch 2/5
80/80 [==============================] - 17s 208ms/step - loss: 0.5366 - accuracy: 0.8280 - val_loss: 0.5147 - val_accuracy: 0.8344
Epoch 3/5
80/80 [==============================] - 15s 193ms/step - loss: 0.4678 - accuracy: 0.8506 - val_loss: 0.4633 - val_accuracy: 0.8444
Epoch 4/5
80/80 [==============================] - 15s 189ms/step - loss: 0.4285 - accuracy: 0.8568 - val_loss: 0.4289 - val_accuracy: 0.8518
Epoch 5/5
80/80 [==============================] - 16s 196ms/step - loss: 0.4000 - accuracy: 0.8625 - val_loss: 0.4061 - val_accuracy: 0.8560
```







Quan disminuim els steps per epoch a 80, obtenim resultats més difusos. On la xarxa confon en més regularitat la mascota amb altres objectes propers, en el nostre cas, la síndria. No obstant, aconsegueix delimitar els contorns amb més énfasis quan el learning rate és 0,1 o 0,001. En canvi, pel learning rate Exponential Decay no.

Conclusió:

Després d'haver variat els hiperparàmetres per comprovar com varien els resultats en funció dels mateix, podem concloure que les millors segmentacions s'han obtingut amb les xarxes neuronals entrenades amb 10 epochs. Encara que cap de les segmentacions obtingudes és capaç de separar a la perfecció la mascota de la resta de la imatge, en tots els casos es pot veure el contorn de la mascota. Malgrat, es pot veure com les segmentacions obtingudes amb la xarxa entrenada amb 5 epochs són pitjors que les entrenades amb 10 epochs; demostrant que, per a una tasca tan complexa com és la segmentació d'imatges, la xarxa requereix d'un nombre major d'iteracions per "aprendre" del training dataset.

Considerem que el segon i cinquè resultat obtingut han estat els millors. En ambdós casos s'ha usat un learning rate de 0,001 que permet un aprenentatge més precís i adient. La diferències entre aquests dos casos és el valor de l'hiperparàmetre d'steps per epoch. Podem veure que en el cinquè cas no apareixen tants de buits a la segmentació com en el segon cas i, per tant, podem inferir que un menor valor de steps per epoch resulta en una millor segmentació de la imatge.

Finalment, com a possibles millores de la pràctica, podriem entrenar la xarxa usant més varietat de valors per al learning rates així com provar diferents tipus d'optimitzadors i no només l'optimitzador Adam (com per exemple AdamW, SGD, Ftrl, etc)