Expected Results are achieved as seen in the files “alexnet-pet-images.txt”, “resnet-pet-images.txt” and “vgg-pet-images.txt”. See screenshots for all 3 with the results and another one for expected results (from lesson “Final Results”).

# Une image contenant texte, capture d’écran, Police, nombre Description générée automatiquement

Figure 1. Expected Results.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 2. Results for ALEXNET.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 3. Results for RESNET.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 4. Results for VGG.

In terms of timing, the obtained results are.

|  |  |
| --- | --- |
| **CNN Model Architecture** | **Time (hh:mm:ss)** |
| ALEXNET | 00:00:12 |
| RESNET | 00:00:17 |
| VGG | 00:00:49 |

Considering uniquely the results, the VGG architecture is the best.

Considering also the timing performances, the RESNET architecture is good enough. It takes almost 3 times less to execute than VGG and it is not so bad in terms of performances for identification.

Choosing the best will also need to consider the size of the images to analyse and the possibility of manually check discrepancies in labels, for example.

When checking the results for the uplodaded images we can see two points.

1. In terms of timings the tendency is the same, T(ALEXNET) < T(RESNET) < T(VGG).
2. Identification and classification is clearly better for RESNET as seen in the images below, being this one the best considering also the timing.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 5. Results for ALEXNET.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 6. Results for RESNET.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 7. Results for VGG