

Rapport TP Classement d'images - CNN

Par **Donatien YETO**

Dans ce document nous allons expliquer la structure de notre solution à travers les différentes étapes clés puis parler brièvement des résultats obtenus.

Notre projet est constitué de 5 notebooks constituant les différentes étapes de notre résolution.

1. 00-View-and-process-data

Pour commencer, nous avons lu les images fournies (fonction `read_images`). A chaque étiquette se trouvant dans `etiquettes.txt` nous lisons les images et leur associons un entier unique comme numéro de classe dans `y`. Nous conservons aussi dans `labels` l'association `class_name` \leftrightarrow `class_number`.

En visualisant les images, on remarque qu'elles ne sont pas de même dimension.



Ce qui nous amène à faire un prétraitement des images. La fonction `myutils/dataset.py : images_enhancement` nous a permis de les redimensionner et surtout de les convertir comme par exemple en blanc-noir.



Les images traitées sont alors stockées dans un nouveau dossier (`output_dir`) pour être utilisées dans la suite.

2. 01-Simple-model

Avec **scale=0.5** dans 00-View-and-process-data, nous avons prétraité la moitié des images dans **./data-0.5-scale**.

Nous avons fait un premier modèle avec les images blanc noir (**set-24x24-L.h5**)

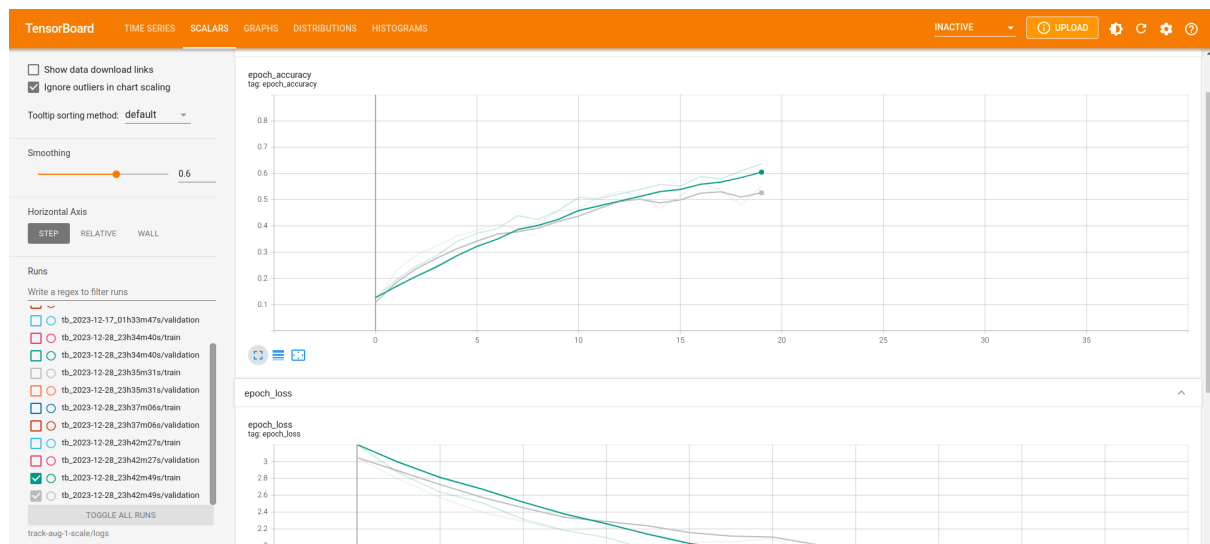
Nous avons obtenu une précision de **43%**.

3. 02-Data-augmentation

Ici nous avons utilisé ***keras.preprocessing.image.ImageDataGenerator***. Elle permet de générer des lots d'images augmentées en temps réel pendant l'entraînement du modèle. Elle est souvent utilisée pour augmenter la taille du jeu de données d'entraînement en appliquant diverses transformations aux images.

En combinaison avec **set-24x24-RGB.h5** et **epochs=20**, nous obtenons **55%** de précision.

Ici nous avons aussi intégré les **callbacks TensorBoard** pour visualiser l'apprentissage de notre modèle.



4. 03-Full-convolution

Nous avons mis en place ici notre protocole expérimental complet pour la sélection de meilleur modèle.

La sélection se base sur:

- les couches du modèle

Nous avons défini 4 modèles CNN différents en jouant sur le nombre et type de couche, les hyperparamètre comme nombre de filters, kernel_size, padding (**myutils/models.py**)

- le dataset d'entraînement
Le prétraitement d'images a permis d'obtenir différents formats d'images au choix ('set-24x24-L-LHE', 'set-24x24-RGB', 'set-48x48-L', 'set-48x48-RGB', 'set-48x48-RGB-HE', ...)
- avec ou sans augmentation de données
Nous avons aussi joué sur les hyperparamètres du générateur au cas où on en utilise.
- et parfois d'autre paramètres comme patch_size, epochs

Notre fonction **myutils/experimental_protocol.py** :

experimental_protocol_multi_run fait plusieurs run avec différentes possibilités de combinaisons pour fournir un rapport.

Voici une partie du rapport d'un multi run.

	Dataset	DatasetSize	Model	Datagen	Duration	Accuracy
0	set-24x24-L-LHE	7.341827	model_v1	none	59.205537	68.862277
1	set-24x24-L-LHE	7.341827	model_v1	generator_v1	60.687897	72.155690
2	set-24x24-L-LHE	7.341827	model_v1	generator_v2	1227.575542	72.455090
3	set-24x24-L-LHE	7.341827	model_v2	none	68.514913	70.359284
4	set-24x24-L-LHE	7.341827	model_v2	generator_v1	72.608461	73.652697
5	set-24x24-L-LHE	7.341827	model_v2	generator_v2	76.406427	71.856290
6	set-24x24-L-LHE	7.341827	model_v3	none	88.987368	64.670658
7	set-24x24-L-LHE	7.341827	model_v3	generator_v1	86.990403	68.562877
8	set-24x24-L-LHE	7.341827	model_v3	generator_v2	81.306295	70.359284
9	set-24x24-RGB	21.984406	model_v1	none	57.100912	73.952097
10	set-24x24-RGB	21.984406	model_v1	generator_v1	58.751402	79.041916
11	set-24x24-RGB	21.984406	model_v1	generator_v2	62.758004	78.143710
12	set-24x24-RGB	21.984406	model_v2	none	67.224905	73.353291
13	set-24x24-RGB	21.984406	model_v2	generator_v1	77.948421	78.143710
14	set-24x24-RGB	21.984406	model_v2	generator_v2	75.170601	79.041916
15	set-24x24-RGB	21.984406	model_v3	none	89.085841	58.083832
16	set-24x24-RGB	21.984406	model_v3	generator_v1	89.141974	69.760478
17	set-24x24-RGB	21.984406	model_v3	generator_v2	91.916154	71.856290
18	set-48x48-L	29.305695	model_v1	none	202.363285	72.155690
19	set-48x48-L	29.305695	model_v1	generator_v1	160.629947	77.844310
20	set-48x48-L	29.305695	model_v1	generator_v2	159.367326	79.640716
21	set-48x48-L	29.305695	model_v2	none	178.405525	70.958084
22	set-48x48-L	29.305695	model_v2	generator_v1	183.123963	77.844310
23	set-48x48-L	29.305695	model_v2	generator_v2	176.789216	76.946110
24	set-48x48-L	29.305695	model_v3	none	204.010438	66.467065
25	set-48x48-L	29.305695	model_v3	generator_v1	201.305293	75.149703
26	set-48x48-L	29.305695	model_v3	generator_v2	200.948354	77.544910
27	set-48x48-RGB	87.876007	model_v1	none	161.164087	75.449103
28	set-48x48-RGB	87.876007	model_v1	generator_v1	161.013065	82.035929
29	set-48x48-RGB	87.876007	model_v1	generator_v2	166.704486	80.838323
30	set-48x48-RGB	87.876007	model_v2	none	173.669242	73.652697
31	set-48x48-RGB	87.876007	model_v2	generator_v1	181.940956	79.341316

On peut voir la combinaison donnant le meilleur score : **82%** .

	Dataset	DatasetSize	Model	Datagen	Duration	Accuracy
28	set-48x48-RGB	87.876007	model_v1	generator_v1	161.013065	82.035929

5. 04-Test-best-model

Nous avons fait quelques prédictions avec le meilleur modèle. Les résultats sont satisfaisants.



ordinateur



parapluie



elephant



bateau



kangourou



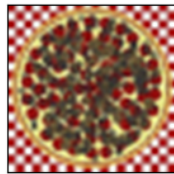
helicoptere



voilier



parapluie (tortue)



pizza



kangourou



parapluie



chaise

Une erreur sur 12 :)

Voici la matrice de confusion.

