

Procédure de test png2pretrain

Delengaigne Pivetaud Schlick

March 29, 2019

1 Protocole

Ce test permet de vérifier que les arguments donnés au programme sont bien gérés (2 arguments donnés qui correspondent bien à des dossiers accessibles par le programme ou que qu'il peut créer) et que l'architecture créée correspond à notre attente.

- Se rendre dans le dossier test
- Exécuter le script "test_png2pretrain.sh"
- Vérifier les résultats

2 Résultats

2.1 Terminal

Le terminal doit afficher les résultats des 6 tests du script, les résultats attendus sont les suivants:

- Erreur pour les 4 premiers tests
- Pas d'erreur pour les 2 derniers
- Création du [output dir] pour le test 5

```

(pfe env) razen@ThinkPad:~/Documents/MC-GAN/test$ ./test_png2pretrain.sh
--- Test png2pretrain ---
Test 1: Error: number of argument != 3 (2)
Usage : python png2pretrain.py [dir with png file] [output dir]
Test 2: Error: number of argument != 3 (4)
Usage : python png2pretrain.py [dir with png file] [output dir]
Test 3: Error: image dir is not a directory
Usage : python png2pretrain.py [dir with png file] [output dir]
Test 4: Error: output dir is not a directory
output dir 'blankfile' is not a directory
Usage : python png2pretrain.py [dir with png file] [output dir]
Test 5: Good: output dir do not exist
Create output directory : test-png2pretrain3/
create test-png2pretrain3/BASE/
create test-png2pretrain3/test/
create test-png2pretrain3/test_dict/
create test-png2pretrain3/train/
create test-png2pretrain3/val/
10 images in png dir
Train: 9
Test: 1
Val:1
Test 6: Good: output dir exist
create test-png2pretrain2/BASE/
create test-png2pretrain2/test/
create test-png2pretrain2/test_dict/
create test-png2pretrain2/train/
create test-png2pretrain2/val/
10 images in png dir
Train: 9
Test: 1
Val:1

```

Figure 1: Résultat attendu sur le terminal après exécution du script

2.2 Images et architecture

Les résultats se trouvent dans les dossiers "test-png2pretrain2" et "test-png2pretrain3". Ces dossiers doivent contenir l'architecture suivante:

- un dossier BASE ne contenant qu'une image Code-New-Roman.0.0.png
- un dossier test ne contenant qu'une image Aetherius.0.0.png
- un dossier test_dict contenant un fichier test_dict
- un dossier train contenant 10 images

- un dossier val contenant une copie d'une des images de train/ : Adiescode.0.0.png

Toutes les images présentent dans les dossiers doivent suivre la même convention que les images créées par le programme "font2png" soit:

- 7296 pixels de large et 64 pixels de haut
- contiennent 114 caractères alignés dans l'image
- de noms `FontName.0.0.png`

a b c d e f g h i j k l m n o p q r s t u v w x y z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 ! " # \$ % & ' () * + , - . / : ; < = > ? @ [\] _ ` { | } ~ ã ä å ç è é ê ë ì í î ï ð ñ ò ó ô õ ö ù ü ÿ Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ñ Ò Ó

Figure 2: Exemple de l'organisation des images (Acme-Regular.0.0.png)