

Nenadzirana klasifikacija slika

Korisnička dokumentacija

Zagreb, Svibanj 2017.

Sadržaj

1	Uvod	2
2	Upoznavanje s radnim direktorijem	3
2.1	Numerički poddirektoriji u direktoriju CNN	3
2.2	Numerički poddirektoriji u direktoriju Results	3
3	Pregled potrebnih programskih alata i paketa	4
4	Pregled programskih skripti	5
5	Literatura	7

1 Uvod

Ovaj dokument predstavlja korisničku dokumentaciju vezanu uz realizaciju rješenja problema nenadzirane klasifikacije slika u svrhu natjecanja *Mozgalo*. Najprije ćemo ukratko čitatelja upoznati s radnim direktorijem, a nakon toga ćemo dati pregled programskih alata i paketa potrebnih za uspješno pokretanje svih priloženih skripti. Posebno ćemo istaknuti alate koji nisu nužni za pokretanje pojedinih programa, ali podosta utječu na efikasnost njihovog izvršavanja. U zadnjem poglavlju ćemo detaljno objasniti pokretanje svake skripte s ulaznim parametrima i očekivane izlazne rezultate. Također, osvrnut ćemo se i na neke varijabilne dijelove programskog koda.

2 Upoznavanje s radnim direktorijem

Radni direktorij sastoji se od sljedećih poddirektorija:

1. *Augmented*. Sadrži augmentirane slike načinjene od slika iz foldera *Images*.
2. *CNN*. Sadrži dva poddirektorija iste strukture koji odgovaraju dvjema neuronskim mrežama. Svaki od njih sadrži direktorije numeričkog imena koji odgovaraju broju pomoćnih klasa na kojima treniramo neuronsku mrežu. Više o tome se može pronaći u sljedećem potpoglavlju.
 - (a) *Big* : 100, 200, 500
 - (b) *Small* : 100, 200, 500
3. *Code*. Sadrži programski kod pisan u *Python*-u.
4. *Images*. Originalni dataset koji želimo klasterirati.
5. *Papers*. Literatura vezana uz rješenje problema.
6. *Results*. Direktorij slične strukture kao *CNN* koji sadrži značajke dobivene neuronskom mrežom, rezultate klasteriranja zapisane u *.csv* datotekama kao i direktorije s klasteriranim slikama.
7. *Test*. Sadrži reskalirane slike iz direktorija *Images*.

2.1 Numerički poddirektoriji u direktoriju CNN

Numerički folderi su iste strukture pa radi jednostavnosti u nastavku opisujemo samo sadržaj foldera 100 unutar foldera *Small*. On sadrži sljedeće foldere i datoteke:

1. *models*. Tu je smješten model *manje* neuronske mreže istreniran na 100 pomoćnih klasa.
2. *train* i *validate*. Direktoriji koji sadrže po 100 poddirektorija, tako da svaki poddirektorij sadrži augmentacije točno jedne slike. Služe za treniranje, odnosno validaciju neuronske mreže. Važno je napomenuti da *train* i *validate* nemaju zajedničkih slika.
3. *mylabel.txt*. Tekstualna datoteka koja u svakom retku sadrži ime jedne pomoćne klase.
4. *train-00000-of-00001* i *validation-00000-of-00001*. Binarne datoteke koje sadrže podatke pogodne za obradu u *TensorFlow*-u.

2.2 Numerički poddirektoriji u direktoriju Results

Numerički folderi su iste strukture pa radi jednostavnosti u nastavku opisujemo samo sadržaj foldera 100 unutar foldera *Small*. On sadrži sljedeće foldere i datoteke:

1. *features.txt*. Tekstualna datoteka koja sadrži podatke o naučenim značajkama.
2. *results.csv*. Tekstualna datoteka koja sadrži rezultate klasteriranja.
3. *results-yyyy-mm-dd hh:mm:ss*. Direktoriji s klasterima.

3 Pregled potrebnih programskih alata i paketa

Sve programske skripte napisane su u programskom jeziku *Python* [1] i testirane na verziji 3.5. Od korištenih programskih paketa u *Python*-u istaknut ćemo sljedeće:

1. *Imgaug*. Biblioteka za augmentaciju slika. Više informacija se može pronaći u [2]. Važno je napomenuti da je ova biblioteka ovisna o biblioteci *OpenCV* [3].
2. *Matplotlib*. Koristi se za crtanje dvodimenzionalnih grafova [4].
3. *NumPy*. Koristi se za znanstveno računanje [5].
4. *SciPy*. Koristi se za znanstveno računanje. Između ostalog sadrži biblioteke *Matplotlib* i *NumPy* [6].
5. *Scikit – learn*. Paket za strojno učenje [7].

Za treniranje konvolucijskih neuronskih mreža koristili smo *TensorFlow* [8]. Poželjno je konfigurirati *TensorFlow* da radi uz pomoć *Nvidia* grafičke kartice kako bi izračunavanja završila u razumnom vremenu. U tom slučaju je potrebno imati instalirati pakete *CUDA Toolkit* 8.0, *cuDNN* v5.1 i *libcupti-dev*, kako je navedeno u [9].

4 Pregled programskih skripti

Za svaku programsku skriptu ćemo dati kratak opis onoga što radi, istaknuti neke važnije parametre koji se mogu mijenjati, odrediti izlazne podatke i po potrebi identificirati moguće greške prilikom pokretanja. Posebno ističemo datoteku *variables.py* koja sadrži sve važnije varijable koje se koriste u ostalim programima i nije namijenjena za izvođenje. Sve skripte koje su namijenjene za izvođenje jednostavno se pokreću naredbom *python3 ime_skripte.py*.

1. *variables.py*. Većina ostalih skripti se poziva na nju. Skripta sadrži sljedeće varijable:

- (a) *main_dir*. Put prema glavnom direktoriju.
- (b) *aug_dir*, *cnn_dir*, *code_dir*, *images_dir*, *papers_dir*, *results_dir*, *test_dir*. Putevi prema sljedećim folderima: *Augmented*, *CNN*, *Code*, *Images*, *Papers*, *Results*, *Test*.
- (c) *num_surr*. Broj pomoćnih klasa.
- (d) *num_aug*. Broj augmentacija po slici.
- (e) *patch_size*, *height*, *width*. Veličine pravokutnog dijela slike na kojem je objekt od interesa.
- (f) *cnn_size*. Veličina neuronske mreže. Može biti *'Big'* i *'Small'*.
- (g) *cnn_current*. Put prema direktoriju koji se odnosi na mrežu koju trenutno treniramo.
- (h) *cnn_models*. Put prema folderu koji sadrži istrenirani model.
- (i) *cnn_model*. Put prema istreniranom modelu.
- (j) *train_dir*, *validate_dir*. Putevi prema slikama za treniranje, odnosno validaciju.
- (k) *labels_file*. Put prema datoteci s oznakama za treniranje i validaciju.
- (l) *results_current*. Put prema direktoriju koji se odnosi na rezultate treniranja.
- (m) *features_file*. Put prema datoteci sa značajkama.
- (n) *results_file*. Put prema rezultatima zapisanim u tekstualnoj datoteci.
- (o) *train_steps*. Broj koraka treniranja mreže.
- (p) *learning_rate*. Parametar koji se odnosi na brzinu učenja.
- (q) *batch_size*. Broj slika koje se koriste u danom koraku za ažuriranje težina u neuronskoj mreži.
- (r) *num_clusters*. Broj klastera za klasteriranje.
- (s) *visual_file*. Put prema slikama koje želimo prikazati u pravokutnoj mreži.

2. *big-CNN.py* i *small-CNN.py*. Skripte koje sadrže definiciju velike, odnosno male neuronske mreže u *TensorFlow*-u. Nisu namijenjene za izvođenje, ali se koriste kao pomoćne skripte u skriptama *big_train.py*, *big_getFeatures.py*, *small_train.py* i *small_getFeatures.py*.

3. *big_train.py* i *small_train.py*. Služe za treniranje velike, odnosno male neuronske mreže na podacima iz *train_dir*. Programi će svakih nekoliko koraka ispisati točnosti na podacima za treniranje, odnosno validaciju. Istrenirani model će se spremiti u *cnn_model*.

4. *big_getFeatures.py* i *small_getFeatures.py*. Služe za dobivanje značajki pomoću slika iz *test_dir* koje se spremaju u datoteku *features_file*. Moguća greška uključuje postojanje direktorija kojeg program želi stvoriti.
5. *makeAugmentations.py*. Program služi za stvaranje $num_surr \cdot num_aug$ augmentiranih slika koje se spremaju u folder *aug_dir*. Moguća greška uključuje postojanje direktorija kojeg program želi stvoriti.
6. *makeLabels.py*. Program stvara direktorije *train_dir* i *validate_dir* koje popunjava slikama iz *aug_dir* te datoteku *labels_file*. Moguća greška uključuje postojanje direktorija kojeg program želi stvoriti.
7. *makeTest.py*. Ova jednostavna skripta stvara direktorij *test_dir* pomoću slika iz *images_dir*. Moguća greška uključuje postojanje direktorija kojeg program želi stvoriti.
8. *organize.py*. Sadrži funkciju *organizeIntoClusters()* koja pomoću datoteke *results_file* kreira foldere s klasteriranim slikama.
9. *paint.py*. Pomoću slika iz *visual_file* stvara pravokutnu mrežu slika.
10. *load_train_image.py*. Sadrži funkciju *getImage()* koja čita jednu sliku iz *TFRecords* zapisa.
11. *build_image_data.py*. Skripta koja slike iz foldera *train_dir* i *validate_dir* stavlja u *TFRecords* format. Skripta je preuzeta sa stranice [10].
12. *kmeans.py* i *kmeans2.py*. Programi koriste algoritam $k - sredina$ na značajkama *features_file* da bi klasificirali slike. Rezultati su spremljeni u *results_file*.

5 Literatura

- [1] Python Software Foundation, <https://www.python.org/>
- [2] Python Software Foundation, <https://pypi.python.org/pypi/imgaug/0.2.1>
- [3] OpenCV, <https://en.wikipedia.org/wiki/OpenCV>
- [4] Matplotlib, <https://matplotlib.org/>
- [5] NumPy, <http://www.numpy.org/>
- [6] SciPy, <https://www.scipy.org/>
- [7] Scikit-learn, <http://scikit-learn.org/stable/>
- [8] TensorFlow, <https://www.tensorflow.org/>
- [9] TensorFlow (instalacija), <https://www.tensorflow.org/install/>
- [10] Build_image_data.py,
https://raw.githubusercontent.com/tensorflow/models/master/inception/inception/data/build_image_data.py