

Term Project

-- Classification of place images --

목표

장소 이미지를 입력 받아 자동으로 campus / farm / train station으로 분류하는 convolutional neural network (CNN) 개발

제공되는 파일 목록

1. TrainData.zip : Labeled images for training, 총 9000개 (class별 3000개)
2. Train_PlaceClassification_reference.ipynb : Reference code for training
3. Test_PlaceClassification_reference.ipynb : Reference code for testing
4. TestData_sample.zip : Unlabeled images, 총 30개 (Testing 코드 작동 확인용)

Coding instructions

1. 'Train_PlaceClassification_reference.ipynb'를 참고하여, generalization error를 최소화할 수 있는 CNN을 자유롭게 구성하여 파라미터를 최적화 하시오. 전체 훈련용 데이터를 사용하여 모델을 학습 시키고, training 결과 (즉, weight matrices & bias vector)를 torch.save 함수를 이용하여 출력 파일 이름은 'Model_TeamXX' 형식으로 저장하십시오.
 - 모델 저장 reference code : `torch.save(model, '저장할 모델 디렉토리(.../Model_TeamXX)')`
2. 'Test_PlaceClassification_reference.ipynb'를 참고하여, 저장된 모델을 load 하고, TestData_sample (30개의 unlabeled images) 에 대한 예측 결과를 출력하는 Test 코드를 완성하십시오.
 - 제공하는 'TestData_sample.zip' 는 코드의 작동 여부를 확인하는데 사용될뿐이며, 제출된 이 코드를 이용하여 추후 3000 개의 test images 에 대해서 예측 정확도를 평가함.
 - Training에 사용했던 모델, 데이터 전처리 과정을 동일하게 코딩 해야함을 유의 하시오.
3. 추가 규칙
 - Transfer learning 불가
 - 제공되는 reference 코드 외에 github 등 웹에서 검색 가능한 코드는 출처를 명시하여 사용 가능하되, pytorch 기반이어야 함

제출해야할 파일 목록

1. Training에 사용한 코드. 파일명 : 'Train_PlaceClassification_TeamXX.ipynb'
2. Testing에 사용한 코드. 파일명 : 'Test_PlaceClassification_TeamXX.ipynb'
3. Trained model 파일 (torch.save 함수를 이용하여 저장된 결과). 파일명 : 'Model_TeamXX'
4. Report : PDF 형식으로 표지 제외 7페이지 이하
 - 1) 최종 모델 도출 과정 및 결과 기술
 - 2) 최종 모델의 구조를 보여주는 block diagram을 반드시 기입

* 코드는 기입하지 마시오.

Grading policy

1. Prediction accuracy: 60%

3000장의 test set에 대한 예측 정확도 평가
2. Report quality: 30%
 - 1) 명확성 (clarity of presentation)
 - 2) 논리성 (logical thinking)
 - 3) 창의성 (originality)
3. Submission: 10%

안내된 형식, 파일명 준수 여부