

Level 2. How to customize AI engines



<http://mindslab.ai>

1. CurrencyNet 구조

1. CurrencyNet Preprocessing

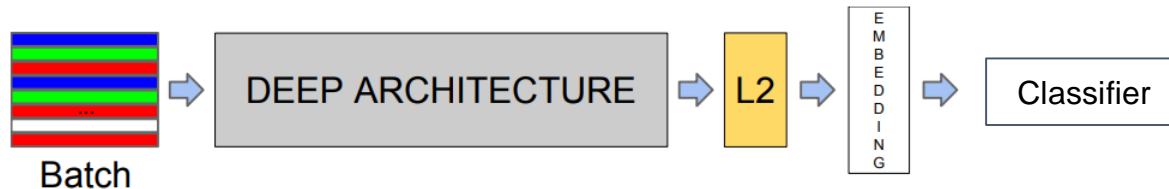
1. CurrencyNet Train

1. CurrencyNet Inference



CurrencyNet 이란

- GoogLeNet 설계팀에서 발표한 Inception 구조를 가진 화폐 사전학습 모델
- Deep Learning -> Support Vector Machine의 two-step 학습 방법을 사용
(이미지 feature를 계산/분류 해주는 두가지 형태의 모델을 사용)
- 대량의 화폐 이미지 데이터를 활용해 사전학습 수행



CurrencyNet source

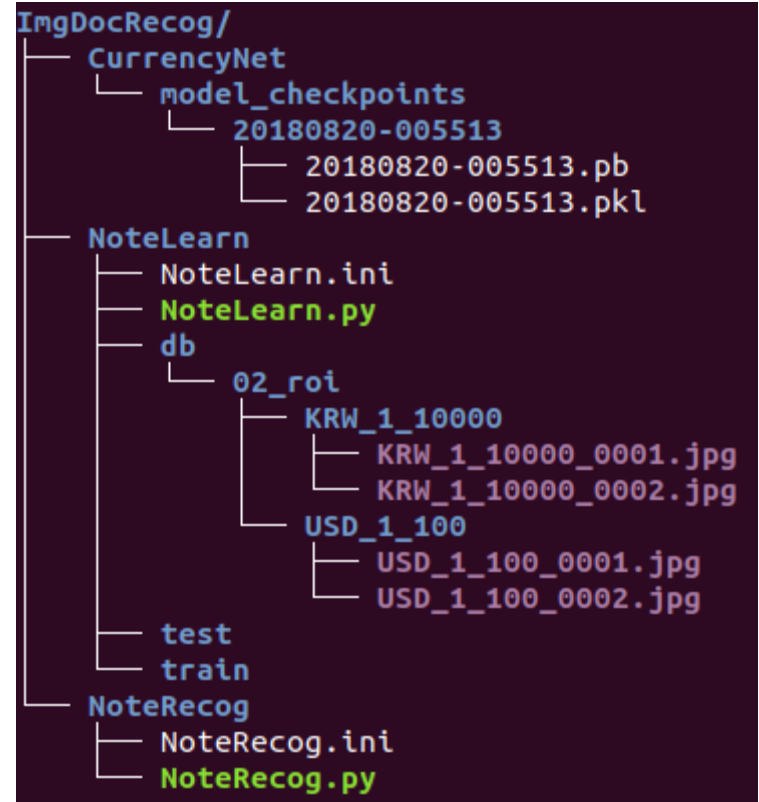
CurrencyNet source tree

Source tree

- model checkpoint: 화폐 학습 모델
 - └ .pb파일: CNN 모델(feature extractor)
 - └ .pkl파일: SVM 모델(feature classifier)
- db/02_roi: 전처리된 화폐 데이터
(클래스 별 N개 샘플 데이터)
- train: 학습 데이터 샘플
- test: 테스트 데이터 샘플

핵심 Command

- NoteLearn.py: CurrencyNet 학습 명령어
- NoteRecog.py: CurrencyNet 실행 명령어
(학습한 모델 테스트)



* CurrencyNet이 동작하기 위한 기본 폴더 구조

CurrencyNet 학습 및 실행 Process

- 기본적으로 아래와 같이 3단계로 진행하며, 성능 안정화가 될 때까지 반복 작업

데이터 수집 및 전처리 (Data Gathering & Preprocessing)

- ✓ 다양한 환경의 화폐 데이터 수집
- ✓ 이미지 전처리 수행(화폐 영역 crop)
- ✓ Data input format 통일화

CurrencyNet 학습 (CurrencyNet Train)

- ✓ CNN 모델 학습(Feature extractor)
- ✓ SVM 모델 학습(Feature classifier)

실행 및 테스트 (CurrencyNet Inference)

- ✓ 모델 학습 결과 확인
- ✓ 다양한 데이터 테스트를 통한 모델 평가

1. CurrencyNet 구조

1. CurrencyNet Preprocessing

1. CurrencyNet Train

1. CurrencyNet Inference

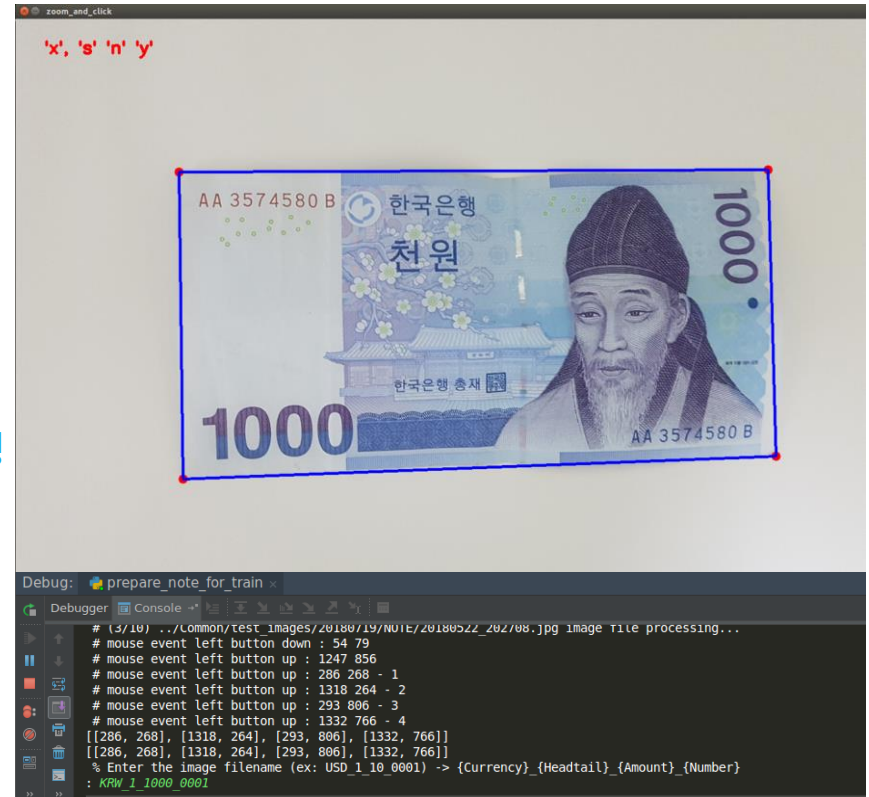


데이터 전처리

- 이미지 cropping & labeling

- 이미지에서 화폐 영역을 추출
- 추출된 화폐에 대한 label 정보 입력

* 학습 데이터 준비를 위해 수동으로 작업 진행



* 이미지 cropping & labeling의 예

데이터 전처리

- 이미지 augmentation & resizing

- 다양한 옵션의 augmentation 수행
- 이미지 최적화를 위해 resizing 수행

* 해당 내용은 'NoteLearn.py'에서 자동으로 수행됨

- 다음의 폴더 tree구조로 학습될 이미지 준비

- 폴더명은 학습될 이미지의 답으로 사용됨

```
- db/02_roi
  KRW_1_10000/
    KRW_1_10000_0001.jpg
    KRW_1_10000_0002.jpg
    ...
  USD_1_100/
    USD_1_100_0001.jpg
    ...
```



* 이미지 augmentation & resizing이 적용된 영상의 예

1. CurrencyNet 구조

1. CurrencyNet Preprocessing

1. CurrencyNet Train

1. CurrencyNet Inference



Note_Learn.py 명령 옵션 (1/3)

- 학습 범위 정하기

```
parser.add_argument('--mode', default='FULL_TRAINING', choices=['FULL_TRAINING', 'FEATURE_EXTRACTOR', 'CLASSIFIER'])
parser.add_argument('--new_dirs', help='New dirs to be trained')
```

- Specify Training mode:
 - `--mode FULL_TRAINING` (default): feature extractor와 classifier를 모두 학습. `.pb`, `.pkl` 파일을 생성함
 - `--mode CLASSIFIER`: 이미 있는 feature extractor(`.pb` 파일)를 사용하여 classifier만 학습. `.pkl` 파일을 생성한다. ini file에 pretrained_pb에 pb 파일이 들어있는 디렉토리를 명시함
 - `--mode FEATURE_EXTRACTOR`: feature extractor만 학습. `.pb`파일을 생성함
- 새로 추가할 이미지가 있는 폴더 경로를 comma로 나누어 argument로 설정
 - Default is `--new_dirs=""`
 - `--new_dirs='/path/to/new/dir1/,/path/to/new/dir2'`

Note_Learn.py 명령 옵션 (2/3)

- 학습 범위 정하기

```
parser.add_argument('--currency', default='ALL', help='Currency to be trained')
parser.add_argument('--headandtail', default='ALL', choices=['1', '0', 'ALL'], help='Headandtail to be trained')
parser.add_argument('--num_images', default=999999, type=int, help='Number of images to be used per note')
```

- 학습할 currency(지폐 종류)를 comma로 나누어 설정
 - --Default is `--currency='ALL'`
 - ex) KRW, USD, JPY, CNY -> `--currency='KRW,USD,JPY,CNY'`
- 학습할 지폐의 앞/뒷면을 설정
 - Default is `--headandtail='ALL'`
 - `--headandtail='1'`: 앞면만 학습
 - `--headandtail='0'`: 뒷면만 학습
- 지폐별 학습 이미지 수를 설정
 - Default is `--num_images=999999`
 - `--num_images=500000`

Note_Learn.py 명령 옵션 (3/3)

- 학습 범위 정하기(Optional)

- 새로운 학습 데이터에 대해 image augmentation 적용시 사용할 명령 옵션

```
parser.add_argument('--aug_num', default=10, type=int)  
parser.add_argument('--roi_num', default=10, type=int)
```

- 새로운 이미지가 있는 경우, 각 이미지 당 augmentation을 몇 번 할지 설정
 - --Default: `--aug_num=10`
- 새로운 이미지가 있는 경우, 위에서 augment된 이미지의 roi를 몇 번 잡을 지 설정
 - --Default: `--roi_num=10`
- 위의 옵션을 지정하면, 새로운 이미지 한 장당 $\text{aug_num} * \text{roi_num}$ 수 만큼 resize된 이미지가 `02_roi`에 저장됨

학습 command 예시

- 새로운 이미지들을 KRW,PHP,EUR 앞면에 대해 feature extractor(pb 파일) 학습시
 - `python NoteLearn.py --mode FEATURE_EXTRACTOR \`
`--new_dirs='/path/to/new/dir1,/path/to/new/dir2' \`
`--currency='KRW,PHP,EUR' --headandtail='ALL' \`
`--num_images=500000`
- Feature classifier(pkl 파일)만 학습시
 - `python NoteLearn.py --mode CLASSIFIER`

1. CurrencyNet 구조

1. CurrencyNet Preprocessing

1. CurrencyNet Train

1. CurrencyNet Inference



Note_Recog.py 명령 옵션

```
parser.add_argument("--img_path", required=True, help="Path for the image to be detected")  
parser.add_argument("--ini", default='./NoteRecog.ini', help="NoteRecog ini filename")
```

- 인식할 이미지가 저장되어 있는 경로 설정
 - `--new_dirs=/path/to/dir/USD_1_1_000001.jpg`
- 학습된 모델의 경로가 저장된 ini 파일내의 모델의 경로 설정
 - ex) model = `../CurrencyNet/model_checkpoints/20180820-005513/20180820-005513.pb`
 - ex) classifier = `../CurrencyNet/model_checkpoints/20180820-005513/20180820-005513.pkl`
- 명령을 수행하여 인식 결과 확인
 - currency : '국가명', headandtail : '앞(1)/뒤(0)', amount : '금액', confidence_level : 인식 신뢰

```
{'currency': 'USD', 'headandtail': '1', 'amount': '1', 'confidence_level': 0.9723970210381262}
```

관련 자료



- <https://arxiv.org/abs/1503.03832>
- https://norman3.github.io/papers/docs/google_inception.html
- <https://github.com/davidsandberg/facenet>
- <https://poddeeplearning.readthedocs.io/ko/latest/CNN/Inception.v4/>

감사합니다.

MINDs Lab. AI Platform Company