

군복 이미지 기반 이진 분류 모델

한국군 vs 북한군

202144046
김민재



문제 정의



입력 변수

구분	내용
입력 데이터	이미지 (JPG/PNG 등)
이미지 크기	(64, 64, 3)
채널 수	3채널 (RGB)
입력 변수 개수	$64 \times 64 \times 3 = 12,288$ 개의 숫자
입력 값 형태	np.array로 (배치 수, 64, 64, 3) 형태



출력 변수



출력 값	의미	모델 최종 예측
0.00 ~ 0.49	한국군 일 확률 높음	→ 0 (한국군)
0.50 ~ 1.00	북한군 일 확률 높음	→ 1 (북한군)

데이터 개수

항목	내용
총 데이터 개수	직접 구한 학습 이미지 개수 (40장), 테스트 이미지 개수 (10장)
클래스 수	2개 (한국군 rok(20장), 북한군 dprk(20장))
속성(Feature)	이미지 픽셀 값 (64×64 크기, RGB 3 채널 → 총 12,288 개 입력 값)
속성 타입	실수형 (float32), 0~1로 정규화 됨
출력(타겟)	이진 클래스 (0=한국군, 1=북한군)
Train/Test 분할 비율	80% 훈련, 20% 테스트, 그 외 외부 테스트 데이터 10장



전처리





전 처리 과정



01 이미지 크기 **통일** (64 x 64)

02 픽셀 값 **정규화** (0~255 -> 0~1)

03 데이터 **무작위** 셔플



데이터 증강

증강 옵션	설정 값	효과
rescale	1./255	픽셀값 정규화
rotation_range	90	최대 90도 회전
brightness_range	[0.8, 1.0]	밝기 조절
width_shift_range	0.2	수평 이동
height_shift_range	0.2	수직 이동
zoom_range	[0.8, 1.2]	확대/축소



하이퍼 파라미터



최종 하이퍼 파라미터

항목	설정 값	설명
이미지 크기	64 x 64 x 3	RGB 이미지
Conv2D layer 수	3개	32, 64, 128
MaxPooling	3개	MaxPooling2D(2)
Dense layer	1개 + 출력층	256 노드, Dropout(0.4)
출력층	Dense(1, activation='sigmoid')	이진 분류용
배치 사이즈	16	데이터 증강 시 설정됨
에포크	30	학습 반복 횟수



그 외 옵션



항목	설정 값	설명
Optimizer	Adam	- 작은 데이터셋에서도 안정적 학습 가능
손실 함수 (Loss)	binary_crossentropy	- 정답이 0 또는 1일 때 확률 기반 예측 오류 계산
Metric	accuracy	- 정확도 기준으로 학습 성능을 판단





최종 성능





테스트 과정



- 01 기본 코드 : 성능 60%**
- 02 Conv2D 레이어 추가 : 성능 70%**
- 03 Input 데이터 크기 증가 (64->128) : 성능 70%**



테스트 코드 01

01 기본 코드 : 성능 60%

```
model = Sequential()  
model.add(Conv2D(32, (3,3), activation='relu', input_shape=(64,64,3)))  
model.add(MaxPooling2D(2))  
model.add(Conv2D(64, (3,3), activation='relu'))  
model.add(MaxPooling2D(2))  
model.add(Flatten())  
model.add(Dense(256, activation='relu'))  
model.add(Dropout(0.4))  
model.add(Dense(1, activation='sigmoid'))
```

테스트 코드 02

02 Conv2D 레이어 추가 : 성능 70%

```
model = Sequential()  
model.add(Conv2D(32, (3,3), activation='relu', input_shape=(64,64,3)))  
model.add(MaxPooling2D(2))  
model.add(Conv2D(64, (3,3), activation='relu'))  
model.add(MaxPooling2D(2))  
model.add(Conv2D(128, (3,3), activation='relu')) # 추가된 부분  
model.add(MaxPooling2D(2))  
model.add(Flatten())  
model.add(Dense(256, activation='relu'))  
model.add(Dropout(0.4))  
model.add(Dense(1, activation='sigmoid'))
```

테스트 코드 03

03 Input 데이터 크기 증가 (64->128) : 성능 70%

```
img = cv2.resize(img, (64, 64))  
-> img = cv2.resize(img, (128, 128))
```

```
model.add(Conv2D(32, (3,3), activation='relu', input_shape=(64,64,3)))  
-> model.add(Conv2D(32, (3,3), activation='relu', input_shape=(128,128,3)))
```

```
img_resized = cv2.resize(img, (64, 64)) / 255.0  
-> img_resized = cv2.resize(img, (128, 128)) / 255.0
```

```
img_input = img_resized.reshape(-1, 64, 64, 3)  
-> img_input = img_resized.reshape(-1, 128, 128, 3)
```


최종 선정

01 ~~기본 코드 : 성능 60%~~

02 **Conv2D 레이어 추가 : 성능 70%**

03 ~~Input 데이터 크기 증가 (64→128) : 성능 70%~~



보고서 결과

≡ 군복 이미지 분류 CNN 모델 ≡

군복 이미지를 활용하여 한국군(ROK) vs 북한군(DPRK)을 구분하는 **이진 분류 모델** 구현

초기 모델 : Conv2D × 2, Dense × 1, 입력 크기 64×64

기본 테스트 정확도 : **60.00%** (데이터 부족 및 클래스 불균형 원인)

성능 개선을 위한 실험

- ImageDataGenerator로 데이터 증강 수행
- Conv 레이어 수를 3개로 확장
- Dense 뉴런 256으로 증가
- 입력 이미지 크기를 128×128로 조정

최종 결과 및 향후 계획

- 테스트 정확도 70.00% 달성
- 모델 구조 및 전처리 기법의 중요성 확인
- 향후 Transfer Learning 도입 및 데이터셋 보강 계획