

HW 01 – REPORT

소속 : 정보컴퓨터공학부

학번 : 201924437

이름 : 김윤하

1. 서론

1-1. 실습 목표

이번 과제 1의 실습 목표는 컴퓨터 비전의 기초적인 이해 및 파이썬을 활용한 기본적인 이미지 처리 기법의 실습입니다.

1-2. 컴퓨터 비전이란?

컴퓨터비전(Computer Vision)이란, 컴퓨터를 사용하여 인간의 시각적 인식 능력을 모방하기 위한 기술 분야입니다. 이미지나 비디오로부터 유용한 정보를 추출하고, 그를 해석하고 수학적 알고리즘을 이용해 연구하는 학문으로, 다양한 분야에 응용하는 것을 목표로 합니다. 다른 종류의 인공지능(AI)와 같이 컴퓨터 비전도 인간의 능력을 복제하는 작업을 자동화합니다.

IBM에서는 컴퓨터 비전을 다음과 같이 정의합니다.

“컴퓨터 비전은 인공지능(AI)의 한 분야로, 컴퓨터와 시스템을 통해 디지털 이미지, 비디오 및 기타 시각적 입력에서 의미 있는 정보를 추출한 다음 이러한 정보를 바탕으로 작업을 실행하고 추천할 수 있도록 합니다. AI를 통해 컴퓨터가 생각을 할 수 있다면 컴퓨터 비전을 통해서도 컴퓨터가 보고, 관찰하고 이해할 수 있습니다.

컴퓨터 비전은 이러한 구분을 수행하도록 머신을 훈련하지만 망막, 시신경, 시각 피질이 아닌 카메라, 데이터 및 알고리즘을 사용하여 훨씬 더 짧은 시간에 수행해야 합니다. 제품을 검사하거나 생산 자산을 관찰하도록 훈련된 시스템은 수천 개의 제품 또는 프로세스를 짧은 시간 동안 분석하여 감지할 수 없는 결함이나 문제를 찾아낼 수 있기 때문에 인간의 능력을 금방 앞지를 수 있습니다.

컴퓨터 비전은 에너지 및 유틸리티부터 제조, 자동차에 이르기까지 다양한 산업에서 사용되며, 시장은 계속 성장할 것으로 예상됩니다.”

1-3. 컴퓨터 비전의 작동 원리¹

컴퓨터 비전은 두 가지의 필수 기술이 사용됩니다. 일종의 머신 러닝인 딥 러닝과, 컨볼루션 신경망(CNN)입니다.

머신 러닝은 컴퓨터가 시각적 데이터의 컨텍스트에 따라 스스로 학습할 수 있도록 하는 알고리즘 모델을 사용합니다. 이러한 모델을 통해 충분한 데이터가 공급되면 컴퓨터가 데이터를 “보고”, 이미지를 서로 구별할 수 있도록 스스로 학습합니다. 알고리즘을 사용하면 누군가가 이미지를 인

¹ IBM : <https://www.ibm.com/kr-ko/topics/computer-vision>

식하도록 프로그래밍하는 대신 컴퓨터가 스스로 학습할 수 있습니다.

CNN은 이미지를 태그 또는 레이블이 지정된 픽셀로 분해하여 머신 러닝 또는 딥 러닝 모델이 “볼 수 있는” 형태로 만듭니다. 그런 다음 레이블을 사용하여 컨볼루션(세 번째 함수를 생성하는 두 함수에 대한 수학적 연산)을 수행하고 “보고 있는” 것에 대해 예측을 수행합니다. 신경망은 컨볼루션을 실행하는데, 예측이 실현되기 전까지 일련의 반복을 통해 예측의 정확성을 확인합니다. 그런 다음 인간과 유사한 방식으로 이미지를 인식하거나 보게 됩니다.

인간이 적당한 거리를 두고 이미지를 만드는 것처럼, CNN은 먼저 명확한 가장자리(hard edge)와 단순한 도형을 구분한 다음 예측을 반복하면서 정보를 채웁니다. CNN은 단일 이미지를 구분하는데 사용됩니다. 순환 신경망(RNN)은 비디오 애플리케이션에서 이와 유사한 방식으로 사용되는데, 컴퓨터가 여러 프레임으로 구성된 그림이 서로 어떻게 관련되어 있는지 이해하는 데 도움이 됩니다.

1-3. 실습 및 라이브러리 소개

실습을 통해 소스 이미지 파일(chipmunk.png)을 변환하여 4가지의 서로 다른 새로운 이미지로 재창출하여, 컴퓨터 비전에서 주로 활용하는 이미지 변환을 직접 수행해 보는 것이 목표입니다. 사용하게 될 라이브러리는 파이썬의 PIL과 Numpy의 오픈소스 라이브러리입니다.

Numpy는 ‘Numeric Python’의 약자로, 파이썬에서 수치 계산을 위해 사용되는 핵심 라이브러리입니다. 대규모 다차원 배열과 행렬 연산에 필요한 다양한 함수 및 메서드를 제공해 줍니다. 데이터 분석, 데이터 처리, 선형대수계산, 머신 러닝 등 다양한 분야에서 폭넓게 사용됩니다. Numpy는 고성능의 수치 계산을 위해 설계되었으며, 다차원 배열 객체 ndarray를 포함하여, 이를 다루기 위한 다양한 함수 및 도구들을 제공합니다. 벡터화 연산, 브로드캐스팅 기능 등을 통해, 복잡한 수치 연산을 효율적이고 간결하게 작성할 수 있게 합니다.

PIL (Python Imaging Library)는 파이썬에서 이미지 처리를 위한 라이브러리입니다. Pillow(PIL Fork)가 현재는 더 활발히 사용되며, 다양한 이미지 파일 포맷을 지원하고 이미지 생성, 수정, 저장 등 이미지 처리에 필요한 다양한 기능을 제공합니다. PIL 라이브러리를 사용하면 이미지 파일 형식 변환, 크롭핑, 회전, 색상 조정 등의 작업을 할 수 있습니다. 또한, 이미지에 대한 다양한 처리와 분석을 가능하게 하여 컴퓨터 비전 프로젝트나 머신 러닝 프로젝트에 유용하게 사용됩니다.

2. 본문

```
from PIL import Image
import numpy as np
```

먼저, 필요한 라이브러리인 PIL과 numpy를 import 합니다.

```
im = Image.open('chipmunk.png')
print (im.size, im.mode, im.format)
im.show()
```

그 다음, 원본 이미지를 'im'이라는 변수에 저장한 후, 이미지의 크기, 픽셀 포맷, 파일 포맷 등의 기본 정보를 출력합니다. 그 후 show() 함수를 통해 im의 이미지를 직접 볼 수 있습니다. 원본 이미지는 다음과 같습니다.



[그림 1] chipmunk.png

2-1. chipmunk_head 이미지 생성

```
im = im.convert('L')
im2 = im.crop((280,150,430,300))
im2.save('chipmunk_head.png','PNG')
```

이미지를 그레이스케일('L' → Luminance)로 변환합니다. 그 후 다람쥐(chipmunk)의 머리 부분을 잘라내기 위해 crop(280, 150, 430, 300)을 사용합니다. 머리 부분을 100 x 100 이미지로 잘라 'chipmunk_head.png'로 저장해 줍니다. 생성된 이미지는 다음과 같습니다.



[그림 2] chipmunk_head.png

2-2. chipmunk_head_bright 이미지 생성

```
im2_array = np.asarray(im2)
```

```

average = np.mean(im2_array)
im3_array = im2_array.copy()

for x in range(0,150):
    for y in range(0,150):
        im3_array[y,x] = min(im3_array[y,x] + 50, 255)

im3 = Image.fromarray(im3_array)
im3.save('chipmunk_head_bright.png','PNG')

```

앞서 크롭한 이미지인 'chipmunk_head(im2)'를 활용해 bright 이미지를 생성합니다. im2를 numpy 배열로 변환하여, intensity의 평균값을 계산해줍니다. 이후 copy() 함수를 사용하여 im3 배열에 저장해줍니다.

중첩 for문을 사용하여, 각 픽셀을 값을 순회하고 각각의 값에 50을 더해 (픽셀 값이 255 이상이면 255로 제한 : 클리핑) 이미지를 전반적으로 밝게 만들어 줍니다. 이후 조절된 numpy 배열을 다시 PIL 이미지 객체로 변환한 다음 'chipmunk_head_bright.png'로 저장해 줍니다. 생성된 이미지는 다음과 같습니다.



[그림 3] chipmunk_head_bright.png

2-3. chipmunk_head_dark 이미지 생성

```

im4_array = im2_array.copy()
im4_array = im4_array * 0.5

im4_array = im4_array.astype('uint8')
im4 = Image.fromarray(im4_array)
im4.save('chipmunk_head_dark.png','PNG')

```

다음은 다람쥐의 머리 부분을 반대로 어둡게 조절하는 작업을 시행합니다. 먼저, 크롭한 원본 이미지의 numpy 배열 복사본을 im4 배열에 만들어주고, 이 배열의 모든 값을 0.5배로 조절해 이미지를 전체적으로 어둡게 만듭니다.

이후 변형된 배열을 uint8의 타입으로 조정하고, 조정된 배열로부터 PIL 이미지를 새로 생성해 'chipmunk_head_dark.png'로 저장해 줍니다. 생성된 이미지는 다음과 같습니다.



[그림 4] chipmunk_head_dark.png

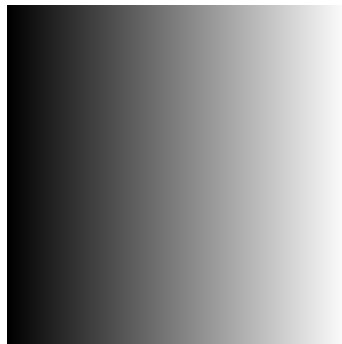
2-4. gradient 패턴 이미지 생성

```
grad = np.arange(0,256)
grad = np.tile(grad,[256,1])

im5 = Image.fromarray(grad.astype('uint8'))
im5.save('gradient.png','PNG')
```

마지막으로 임의의 그라데이션 테스트 패턴의 이미지를 생성하는 작업을 시행합니다. 먼저, 0부터 255까지의 값을 가지는 총 길이 256인 1차원 배열을 생성합니다. 이 배열을 256번 반복하여 2차원(256 x 256)의 그라데이션 패턴을 만듭니다.

이를 uint8 타입으로 조정하고, 이 2차원 배열을 이용해 PIL 이미지 객체를 생성해 'gradient.png'로 저장해 줍니다. 생성된 이미지는 다음과 같습니다.



[그림 5] gradient.png

3. 결론

3-1. 실습 결과

Numpy와 PIL 두 가지의 오픈소스 라이브러리를 사용하여 기본적인 이미지 처리를 실습할 수 있었습니다. 'chipmunk_head.png', 'chipmunk_head_bright.png', 'chipmunk_head_dark.png' 와 같이 기존 이미지인 'chipmunk.png'를 다양한 이미지 처리 기법을 통해 기존의 이미지를 변환하여 세 가지의 새로운 이미지를 생성할 수 있었고, 'gradient.png' 와 같이 이미지가 없어도 새로운 이미지

를 창출할 수 있었습니다.

3-2. 컴퓨터 비전 및 이미지 처리의 응용 분야

컴퓨터 비전과 이미지 처리 기술은 현대 기술의 중심축으로, 다양한 산업 분야에서 광범위하게 활용되고 있습니다. 이번 실험에서는 이미지 처리 기술에 중점을 두고 기본적인 처리 방법들을 배웠습니다. 다음은 컴퓨터 비전 및 이미지 처리가 응용될 수 있는 분야입니다.

- **컴퓨터 비전의 응용 분야**

- 제조업 : 고속 검사, 제품 품질 제어, 로봇 지도 등
- 의료 : 환자 진단 지원, 수술 시뮬레이션, 의료 이미지 분석 등
- 교통 : 자율 주행 차량, 교통 흐름 분석, 차량 번호판 인식 등
- 스포츠 : 선수 성능 분석, 게임 전략 개발, 팬 경험 향상 등

- **이미지 처리의 응용 분야**

- 자율 주행 차량 : 주변 환경 인식, 장애물 감지 및 회피, 차선 유지 등에 필요한 시각 정보 처리 등
- 보안 시스템 : 얼굴 인식을 통한 접근 제한, 감시 카메라를 통한 활동 감시 등
- 헬스케어 : 의료 영상 분석을 통한 질병 진단 및 치료 계획 등

이와 같은 다양한 분야에서의 응용을 통해 컴퓨터 비전과 이미지 처리 기술이 우리 삶을 어떻게 변화시키고 있는지를 확인할 수 있습니다. 앞으로도 기술 발전에 따라 새로운 응용 분야가 지속적으로 등장할 것으로 기대됩니다. 컴퓨터 비전과 이미지 처리의 발전은 무궁무진할 것입니다. 컴퓨터비전개론에서 이와 같은 분야에 대해 탐구하는 것이 우리의 궁극적인 목표입니다.

3-3. 마무리

앞으로 한 학기 동안 컴퓨터비전개론 수업을 통해 얻게 될 지식과 경험은 제 개인적인 역량으로의 발전뿐만 아니라, 빅데이터 및 인공지능 산업에서 중요한 역할을 하게 될 전문가로서 성장하는 데 큰 밑거름이 될 것이라 기대합니다. 특히, 이미지 처리를 활용한 생체 정보 인식 보안 시스템에 대한 관심이 큼니다.

이번 학기를 통해서, 이미지 처리와 컴퓨터 비전 기술의 기본적인 원리를 학습하고, 이를 실제에 적용하는 방법에 대한 실습을 하여 더 방대한 범위의 인공지능 기술 발전에 기여할 수 있는 역량을 키우고 싶습니다. 컴퓨터 비전과 이미지 처리 기술이 제공하는 무궁무진한 가능성을 탐구하여 전문성을 높여가고자 합니다.