**BMP Combination with Mask**

모코코 - 김태현, 강형준, 신민성, 이도형

**0. Requirement**

C++, MFC, visual studio, python, pycharm

**1. Purpose**

BMP Image의 Header에 대해 학습한다.

Image Segmentation으로 auto Mask 생성 함수를 구현한다.

**2. Design**

2-1. BMP Image, background 두 장을 loading 한다.

8 비트와 24 비트 BMP Image를 흑백과 컬러로 load 할 수 있다. (DIB, V3)

8 비트 BMP Header에는 팔레트가 존재한다.

2-2. BMP Image에서 얼굴 부분을 masking 한다.

2-2-0. 자동 마스크 생성

2-2-1. 데이터 수집

Human Parsing-Dataset을 수집.

(https://github.com/lemondan/HumanParsing-Dataset)

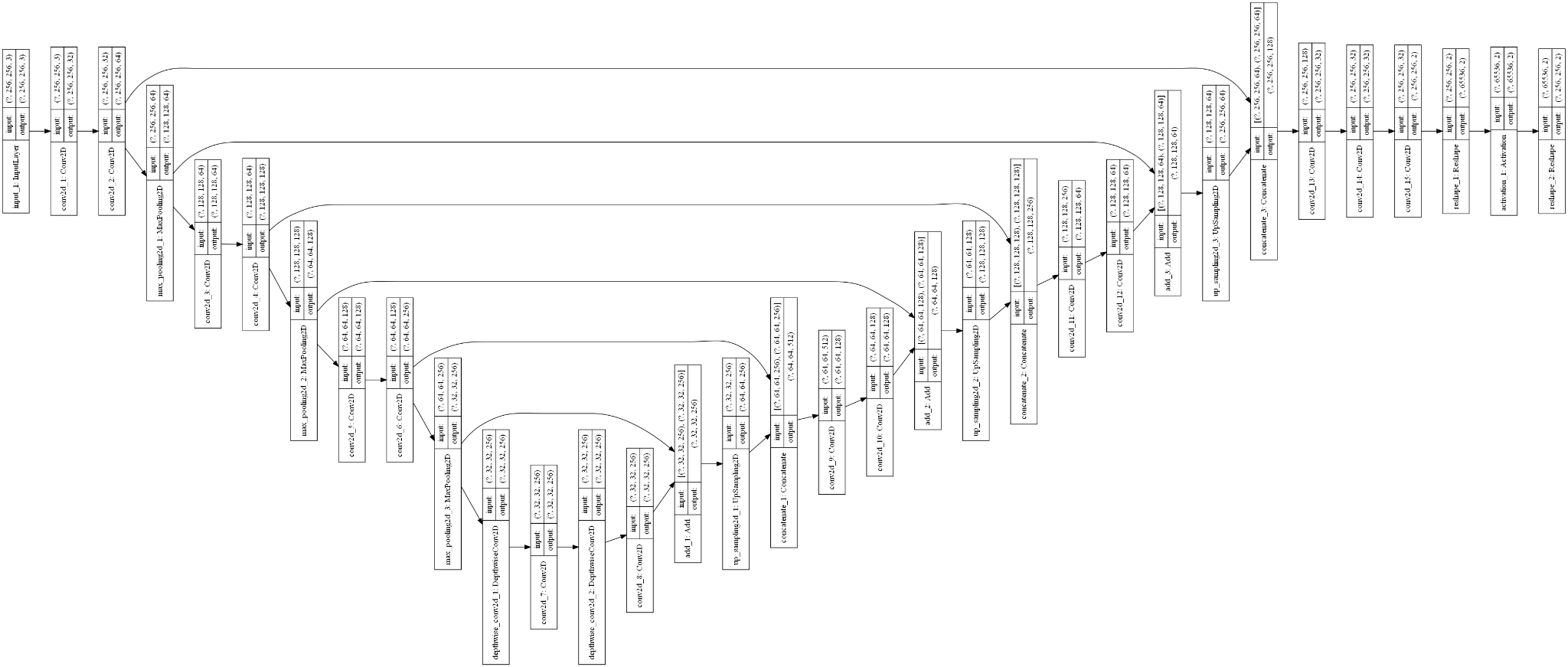
2-2-2. 데이터 전처리

각 이미지 데이터를 스케일에 맞게 (256,256) resize 한 후 2채널로 분리.

2-2-3. 모델 학습

학습 데이터를 정규화(1./255)한 후, 그 중 훈련 데이터를 밝기(0.7,1.3)로 증강.

Image Segmentation을 하기 위해 U-net 모델을 적용.

****

https://github.com/leedohyeong/KSA\_MAKING\_MASK/blob/main/model.png

con2d : 15, Max\_pool : 3, Up\_sampling : 3, Concatenate : 3, Depthwise: 2

loss = 'categorical\_crossentropy', optimizer = 'adam’, epochs = 100,

BATCH\_SIZE = 16, callback 함수를 활용해 학습률을 조정.

학습 결과 0.9853의 정확도를 보임.

모델을 h5 형태로 저장.

2-2-4. 모델 Test

OpenCV로 webcam을 활용해 입력 이미지를 저장.

저장된 모델에 입력 이미지를 reshape(256,256,3)한 후 예측 이미지 도출.

예측 이미지 중 1번 채널만 사용하여 background 이미지에 합성 이미지 파일 생성.

합성된 BMP 파일을 MFC로 불러들여 결과 확인.

2-3. Masking한 BMP Image를 BMP background 와 combination 한다.

2-4. 결과 BMP Image를 save 한다.

**3. Conclusion**

MFC코드 (visual studio)

마스크 구현코드 (pycharm)

실행영상파일 (mp4)

이미지파일 (bmp)

****

<원본 이미지 파일> <배경 이미지 파일> <합성 이미지 파일>

**4. BMP(Bitmap) File**

비트맵 디지털 그림을 저장하는 그림 파일 포맷.

RLE, JPEG, PNG의 압축형식을 지원하지만 대부분 파일 압축을 하지 않아 파일 크기가 크다.

압축을 풀 필요가 없기 때문에 속도가 빠르다.

단순하고 특허에 자유로우며 ZIP과 같은 무손실 데이터 압축 알고리즘으로 상당 용량을 압축할 수 있다.

4-1. BMP File 종류

4-1-1. DIB(Device Independent Bitmap)

DIB는 장치에 독립적인 비트맵 방식으로 DDB 정보에 색상 테이블과 해상도 정보 등의 추가 정보를 갖고 있기 때문에 활용도가 넓고 호환성이 좋다.

4-1-2. DDB(Device Dependent Bitmap)

DDB는 장치에 종속적인 비트맵 방식으로 출력 장치에 의존적이기 때문에 제한된 환경에서 사용한다.

4-2. BMP Color Depth

|  |  |  |
| --- | --- | --- |
| 비트 수 | 색상 수 | 구분 |
| 1 bit | 2 색 | 흑백 |
| 2 bit | 4 색 | 팔레트 |
| 3 bit | 8 색 |
| 4 bit | 16 색 |
| 8 bit | 256 색 |
| 16 bit | 65,536 색 | 하이컬러 Highcolor (R:G:B = 5:5:5) |
| 24 bit | 16,777,216 색 | 트루컬러 Truecolor (R:G:B = 8:8:8) |
| 32 bit | 16,777,216 색 + 8 bit 알파채널 | 트루컬러 + 알파채널 |

4-3. BMP File Format

|  |  |
| --- | --- |
| File Header | BMP 파일 식별 정보 |
| Image Header | 이미지에 대한 정보 |
| Color Palette | 색상 정보 |
| Pixel Data | 픽셀 데이터 |

4-3-1. File Header (BITMAPFILEHEADER)

File Header는 BITMAPFILEHEADER라는 구조체에 정의되어 있다.

현재 파일 포맷이 BMP인지 확인할 수 있는 정보가 있다.

|  |  |  |
| --- | --- | --- |
| 오프셋 | 크기 | 목적 |
| 0 | 2 | BMP 파일 식별. B(0x42), M(0x4D) 을 확인할 수 있다. |
| 2 | 4 | 헤더를 포함한 BMP 파일의 전체 크기. little-endian으로 표기. |
| 6 | 2 | 준비. 그림을 만드는 데 쓰인 응용 프로그램에 따라 달라진다. |
| 8 | 2 | 준비. 그림을 만드는 데 쓰인 응용 프로그램에 따라 달라진다. |
| 10 | 4 | 비트맵 데이터가 시작되는 위치. 오프셋(Offset) |

4-3-2. Image Header

Image Header는 BITMAPFILEHEADER 구조체 다음에 위치한다.

4-3-2-1. BMP Image Header 종류

|  |  |  |  |
| --- | --- | --- | --- |
| 크기 | 헤더 | 식별자 | GDI 지원 |
| 40 | 윈도 V3 | BITMAPINFOHEADER | 윈도우 3.0 이후의 모든 윈도 버전 |
| 12 | OS/2 V1 | BITMAPCOREHEADER | OS/2 및 윈도우 3.0 이후의 모든 윈도 버전 |
| 64 | OS/2 V2 |  |  |
| 108 | 윈도 V4 | BITMAPV4HEADER | 윈도 95/NT4 이후의 모든 윈도 버전 |
| 124 | 윈도 V5 | BITMAPV5HEADER | 윈도 98/2000 이상 |

호환성을 이유로 대부분 DIB 헤더를 사용한다.

OS/2가 물러나고 일반적으로 V3 헤더를 사용한다.

4-3-2-2. BMP Image Header (DIB, V3) (BITMAPINFOHEADER)

|  |  |  |
| --- | --- | --- |
| 오프셋 | 크기 | 목적 |
| 14 | 4 | 이 헤더의 크기 (40 바이트) |
| 18 | 4 | 비트맵 가로 (단위는 화소, signed integer) |
| 22 | 4 | 비트맵 세로 (단위는 화소, signed integer) |
| 26 | 2 | 사용하는 색 판(color plane)의 수. 1로 설정해야 한다. |
| 28 | 2 | 한 화소에 들어가는 비트 수, 그림의 색 깊이. 보통 값은 1, 4, 8, 16, 24, 32 |
| 30 | 4 | 압축 방식. 가능한 값에 대한 목록은 다음 표를 참조. |
| 34 | 4 | 그림 크기. 압축되지 않은 비트맵 데이터의 크기(아래 참조)이며,  파일 크기와 혼동하지 말 것. |
| 38 | 4 | 그림의 가로 해상도. (미터 당 화소, signed integer) |
| 42 | 4 | 그림의 세로 해상도. (미터 당 화소, signed integer) |
| 46 | 4 | 색 팔레트의 색 수, 또는 0에서 기본값 2^n. |
| 50 | 4 | 중요한 색의 수. 모든 색이 중요할 경우 0. 일반적으로 무시. |

4-3-2-3. 압축 방식 종류

압축 방식(바이트 #30-33)은 다음의 값을 가진다.

BI\_RGB 비트맵의 경우 그림 크기 필드는 0이 될 수 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| 값 | 식별자 | 압축 방식 | 비고 |
| 0 | BI\_RGB | 없음 | 가장 일반적이다. |
| 1 | BI\_RLE8 | RLE 8비트/화소 | 8비트/화소 비트맵에만 사용할 수 있다. |
| 2 | BI\_RLE4 | RLE 4비트/화소 | 4비트/화소 비트맵에만 사용할 수 있다. |
| 3 | BI\_BITFIELDS | 비트 필드 | 16, 32비트/화소 비트맵에만 사용할 수 있다. |
| 4 | BI\_JPEG | JPEG | 비트맵은 JPEG 이미지를 포함한다. |
| 5 | BI\_PNG | PNG | 비트맵은 PNG 이미지를 포함한다. |

4-3-3. Color Palette

BMP 파일의 비트수에 따라 세 가지(RGBQUAD, RGBTRIPLE, Mask) 포맷이 있다.

8 비트 이하에서는 RGB 값을 나타내기 위해 컬러맵(Color Palette)을 사용한다.

윈도우 BMP 파일은 RGBQUAD 구조체를 사용한다.

OS/2에서는 RGBTRIPLE 구조체를 사용한다.

4-3-3-1. RGBQUAD 구조체

|  |  |  |
| --- | --- | --- |
| Field | Size | Description |
| rgbBlue | 1 | 파란색 값 |
| rgbGreen | 1 | 빨간색 값 |
| rgbRed | 1 | 녹색 값 |
| rgbReserved | 1 | 항상 0 값 |

4-3-3-2. RGBTRIPLE 구조체

|  |  |  |
| --- | --- | --- |
| Field | Size | Description |
| rgbBlue | 1 | 파란색 값 |
| rgbGreen | 1 | 빨간색 값 |
| rgbRed | 1 | 녹색 값 |

4-3-3-3. Mask

컬러맵을 사용하지 않고 비트 마스크를 사용한다.

16, 24, 32 비트 이미지의 경우 팔레트 자체 용량이 커지기 때문이다.

|  |  |
| --- | --- |
| RED | 00 0000000000 0000000000 1111111111 |
| GREEN | 00 0000000000 1111111111 0000000000 |
| BLUE | 00 1111111111 0000000000 0000000000 |

4-3-4. Pixel Data

컬러 팔레트나 비트 마스크 다음에 위치한다.

1, 4 비트 이미지 : 2개, 8개의 필드로 나뉘고, 각각 컬러 팔레트 값을 나타낸다.

8 비트 이미지 : 줄단위 각 픽셀은 1 바이트이고, 각각 컬러 팔레트 값을 나타낸다.

16 비트 이미지 : 각 픽셀은 2 바이트 정수형으로 나타낸다.

BI\_RGB - 각 컬러값은 5 비트씩 사용하고, 최상위 비트는 사용하지 않는다.

BI\_BITMAP - 3개의 4 바이트 비트 마스크가 각 컬러 요소에 사용된다.

컬러 순서는 red, blue, green.

24 비트 이미지 : 각 픽셀은 blue, green, red의 연속된 바이트열로 나타낸다.

픽셀의 순서가 반대.

32 비트 이미지 : 각 픽셀은 4 바이트의 정수형으로 나타낸다.

BI\_RGB - blue, green, red 순서로 각 8 비트 값을 나타낸다.

24 비트와 유사하며 8 비트는 알파채널이다.

BI\_BITFIELD - 3개의 4 바이트 비트 마스크를 사용한다.

비트 마스크 순서는 red, blue, green.