

「지능화 캡스톤 프로젝트」 설계

## OpenCV 활용

주요 명령어 사용법, 영상파일 입출력 및 변화 등

2022. 03. 23

20-4 조

2020254005 김성웅

2020254011 윤재웅

# 목 차



# 목 차

---

목 표 1. CNN 의 구성 및 단계

목 표 2. 주요 명령어 및 사용법

목 표 3. 영상파일 입출력 및 변환 결과 확인

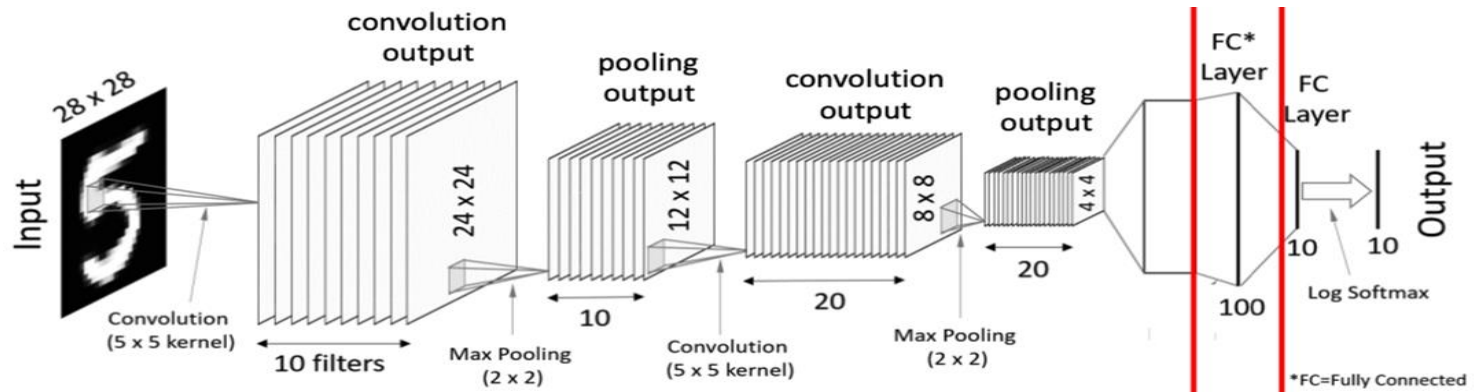
# CNN의 구성 및 단계

---

# 1. CNN 의 구성 및 단계

## 목 표 1. CNN 의 구성 및 단계

- **CNN**(Convolutional neural network)
- 주로 **Matrix 데이터**나 **이미지 데이터**에 대하여 **특징을 추출**하는데 용이
- 자동으로 **피쳐(Feature)**를 학습



출처 : <http://taewan.kim/post/cnn/>

- o Layer 1 : Convolution
- o Layer 2 : Pooling
- o Layer 3 : Convolution
- o Layer 4 : Pooling
- o Layer 5 : FC(Fully-Connected)

# 주요 명령어 및 사용법

---

## 2. 주요 명령어 및 사용법

---

### 목 표 2. 주요 명령어 및 사용법

1. *Open CV 사용 선언*
2. *Imread*
3. *Shape*
4. *channel*
5. *이미지 변환 및 화면 출력*
6. *이미지 저장*
7. *픽셀 변환*

## 2. 주요 명령어 및 사용법

### 목 표 2. 주요 명령어 및 사용법

#### 1. *Open CV* 사용 선언

○ *import cv2*

*OpenCV (Open Source Computer Vision Library)*

*import cv2*로 사용 선언이 가능하며 상호작용, 물체 인식, 안면 인식, 로봇틱스, 제스처 인식 등 컴퓨터 비전 관련 프로그래밍을 도와준다

○ *import random*

```
1
2  #1. Open CV 사용 선언
3  import cv2
4
5  #2. imread : 괄호안의 경로 이미지를 불러온다.
6  image = cv2.imread('image_F16.png')
```

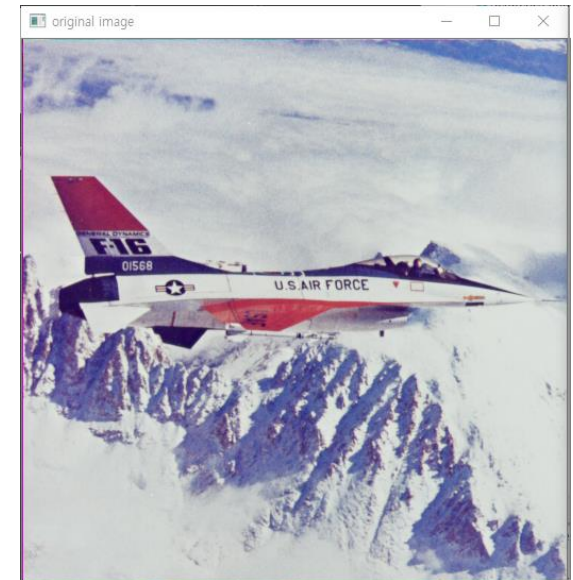
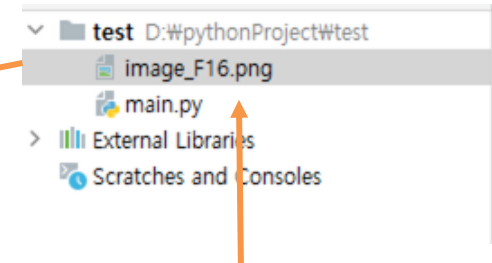


## 2. 주요 명령어 및 사용법

### 목 표 2. 주요 명령어 및 사용법

#### 2. *imread* : 괄호안의 경로 이미지 호출

```
5 #2. imread : 괄호안의 경로 이미지를 불러온다.  
6 image = cv2.imread('image_F16.png')
```



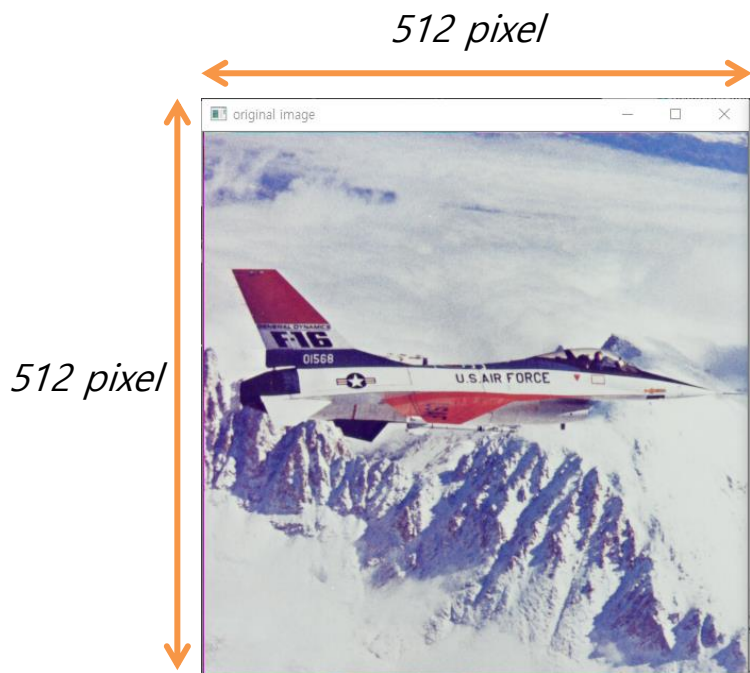
*Image\_F16.png*

## 2. 주요 명령어 및 사용법

### 목 표 2. 주요 명령어 및 사용법

#### 3. *shape* : 이미지의 정보를 호출

- 가로 픽셀수 : 이미지의 가로 픽셀수를 카운트
- 세로 픽셀수 : 이미지의 세로 픽셀수를 카운트



```
8  #3. shape : 이미지의 정보를 가져온다.  
9  #가로 픽셀수  
10 image_width = image.shape[1]  
11 #세로 픽셀수  
12 image_height = image.shape[0]  
13  
14  
15  
16 print(f"Width : {image_width}",)  
17 print(f"Height : {image_height}")  
18
```



실행 결과

```
C:\Users\user\AppData\Local\Programs  
Width : 512  
Height : 512  
  
Process finished with exit code 0
```

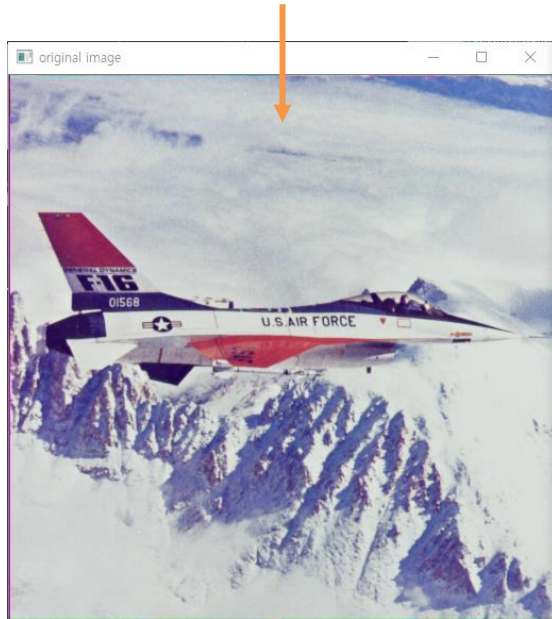
## 2. 주요 명령어 및 사용법

### 목 표 2. 주요 명령어 및 사용법

#### 4. *channel* : 채널의 수

○ (컬러 이미지는 B, G, R 세 개의 채널, 흑백 이미지는 한 개의 채널)

Blue, Green, Red로 이루어진 컬러 이미지



```
8  #3. shape : 이미지의 정보를 가져온다.  
9  #가로 픽셀수  
10 image_width = image.shape[1]  
11 #세로 픽셀수  
12 image_height = image.shape[0]  
13 #채널(컬러 이미지는 Blue, Red, Green 세개의 채널, 흑백 이미지는 1개 채널)의 숫자  
14 image_channel_count = image.shape[2]  
15  
16 print(f"Width : {image_width}",)  
17 print(f"Height : {image_height}")  
18 print(f"Channel : {image_channel_count}")
```



실행 결과

```
C:\Users\user\AppData\Local\Programs  
Width : 512  
Height : 512  
Channel : 3  
  
Process finished with exit code 0
```

## 2. 주요 명령어 및 사용법

### 목 표 2. 주요 명령어 및 사용법

#### 5. 이미지 변환 및 화면 출력

- o `imshow("제목",출력이미지)` : `cv2.imshow("original image",image)`
- o `cvtColor(원본이미지,변환형식)` : `gray_image = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)`

```
20 #4. 이미지 변환 및 화면 출력
21 #imshow("제목", 출력이미지)
22 cv2.imshow("original image",image)
23 #cvtColor(원본이미지, 변환형식)
24 gray_image = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
25 hsv_image = cv2.cvtColor(image,cv2.COLOR_BGR2HSV)
26 cv2.imshow("gray image",gray_image)
27 cv2.imshow("hsv image",hsv_image)
```

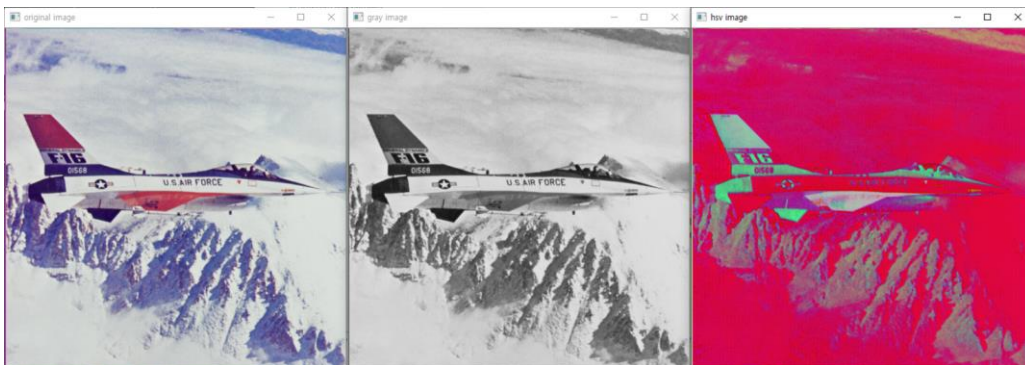


실행 결과

원본

Gray변환

hsv 색상변환



## 2. 주요 명령어 및 사용법

### 목 표 2. 주요 명령어 및 사용법

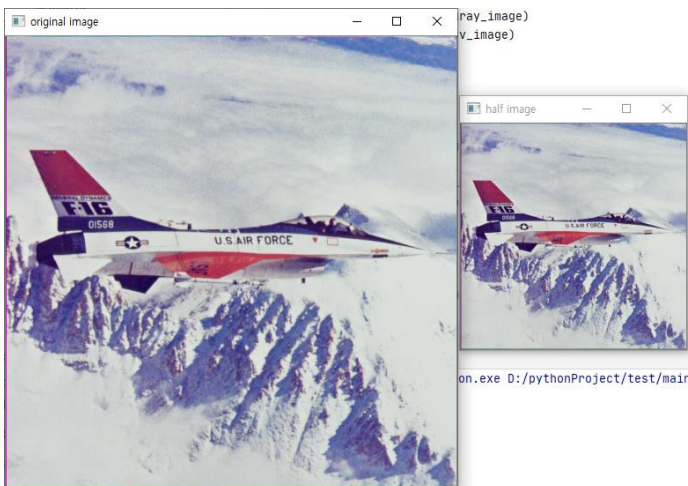
#### 5. 이미지 변환 및 화면 출력

o *resize(변경할 이미지, 크기) : half\_image = cv2.resize(image, (256,256))*

```
31  #5. 이미지 크기 변환
32  cv2.imshow("original image",image)
33  #resize(변경할 이미지, 크기)
34  half_image = cv2.resize(image, (256,256))
35  cv2.imshow("half image",half_image)
36  cv2.waitKey(0)
```



실행 결과



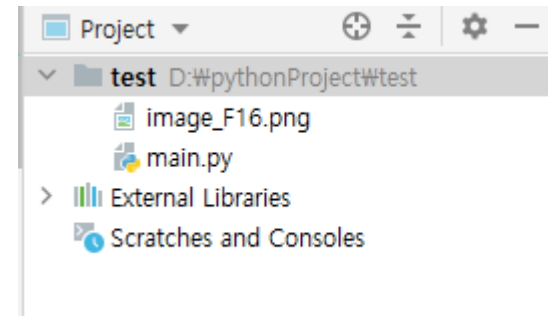
## 2. 주요 명령어 및 사용법

### 목 표 2. 주요 명령어 및 사용법

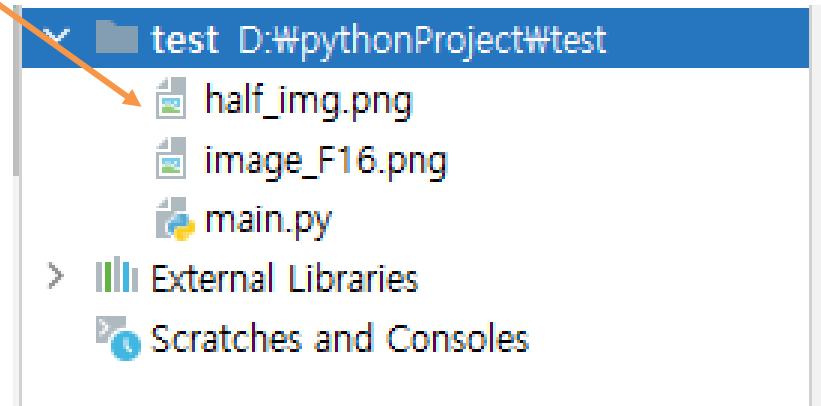
#### 6. 이미지 저장

○ `cv2.imwrite('half_img.png', half_image)`

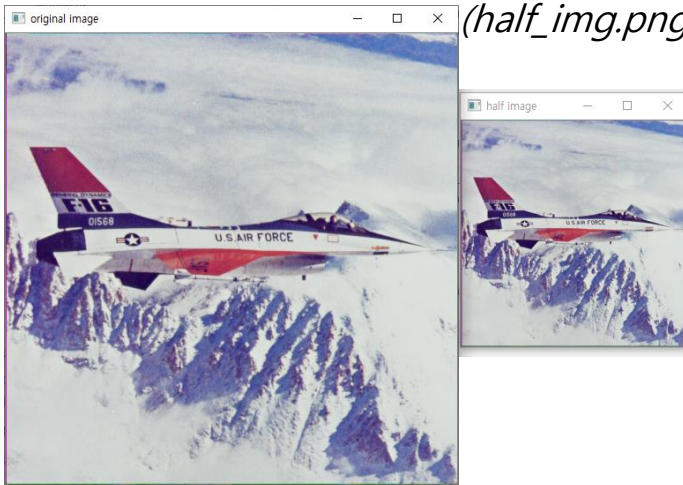
```
39 # 📌 이미지 저장
40 cv2.imwrite('half_img.png', half_image)
```



실행 결과



원본(image\_F16.png)      변환이미지  
(half\_img.png)





## 2. 주요 명령어 및 사용법

### 목 표 2. 주요 명령어 및 사용법

#### 7. 픽셀 변환

- 원본 이미지의 256이하의 세로 픽셀을 파란색으로 변경 :

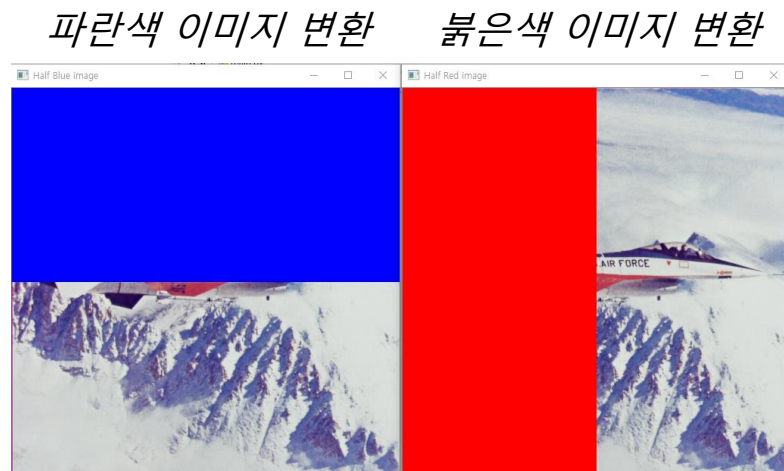
`blue_img[:256] = (255,0,0) #BGR`

- 원본 이미지의 256이하의 가로 픽셀을 빨간색으로 변경 :

`red_img = image.copy()`

```
43 #7. 픽셀 변환
44 blue_img = image.copy()
45 # 원본 이미지의 256 이하의 세로 픽셀을 파란색으로 변경
46 blue_img[:256] = (255,0,0) #BGR
47 cv2.imshow("Half Blue image",blue_img)
48
49 red_img = image.copy()
50 # 원본 이미지의 256 이하의 가로 픽셀을 빨간색으로 변경
51 red_img[:, :256] = (0,0,255) #BGR
52 cv2.imshow("Half Red image",red_img)
```

실행 결과



## 2. 주요 명령어 및 사용법

### 목 표 2. 주요 명령어 및 사용법

#### 7. 픽셀 변환

- 원본 이미지에서 랜덤한 픽셀을 빨간색으로 변경 :

`rand_img = image.copy()`

- 랜덤 함수는 `import random` 하여 사용

#1. Open CV 사용 선언

`import random`

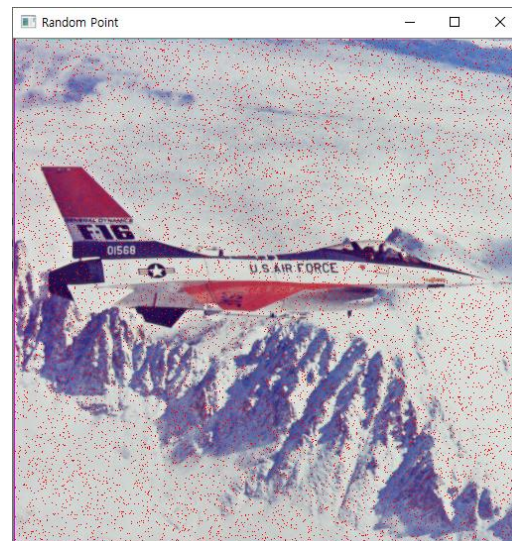
`import cv2`

```
54 #원본 이미지에서 랜덤한 픽셀을 빨간색으로 변경
55 rand_img = image.copy()
56 w, h = rand_img.shape[1], rand_img.shape[0]
57 for i in range(10000):
58     rand_w = random.randrange(w)
59     rand_h = random.randrange(h)
60     rand_img[rand_w,rand_h] = (0,0,255)
61
62 cv2.imshow("Random Point",rand_img)
```

실행 결과



랜덤한 픽셀 빨간색으로 이미지 변환



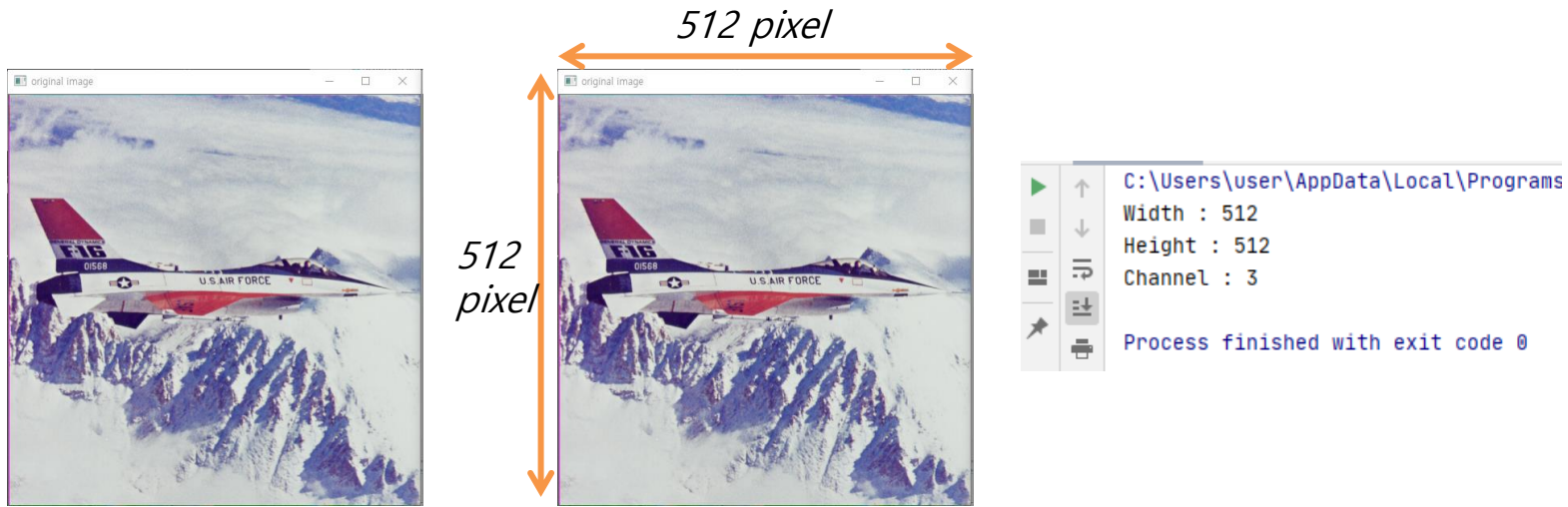


# 영상파일 입출력 및 변환 결과



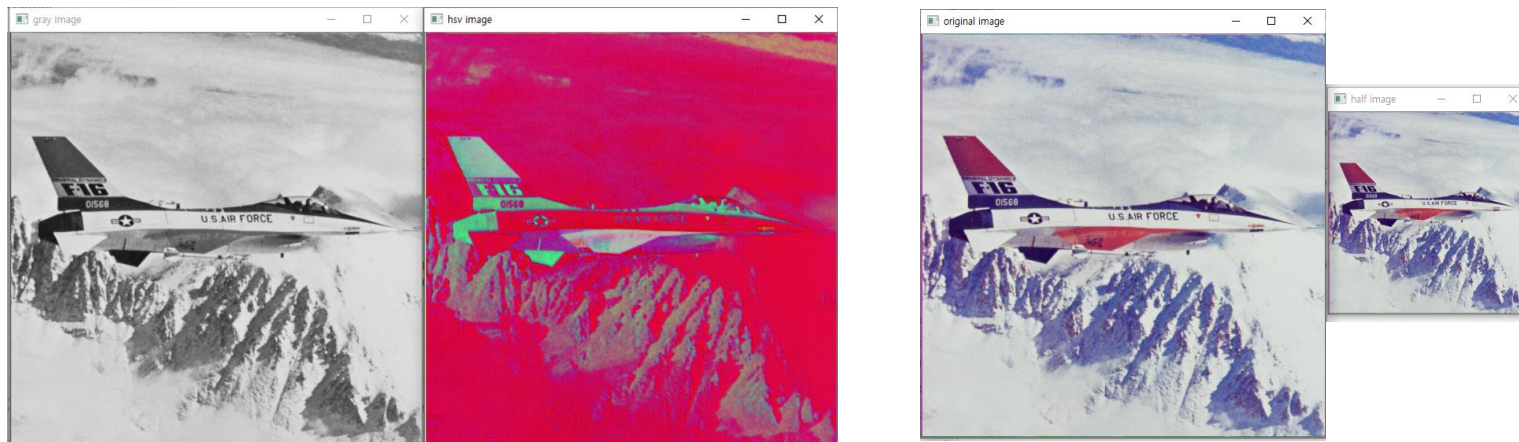
# 3. 영상파일 입출력 및 변환 결과

## 목 표 3. 영상파일 입출력 및 변환 결과



원본

픽셀 및 채널 수 카운트

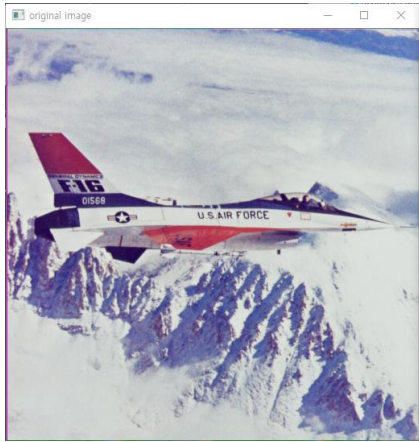


흑백 및 hsv 색 변환

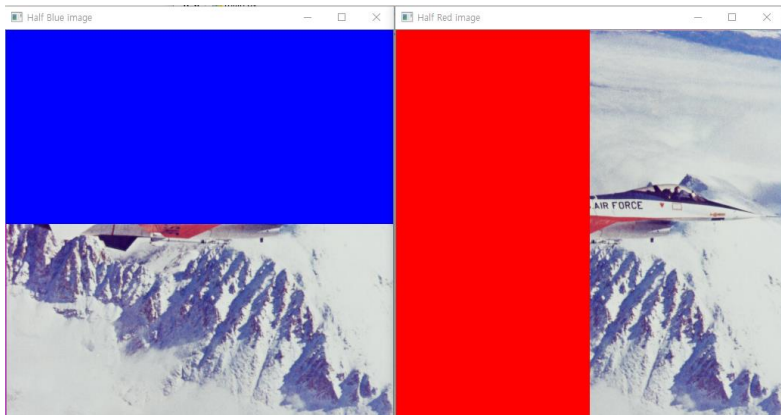
Half 사이즈 변환

# 3. 영상파일 입출력 및 변환 결과

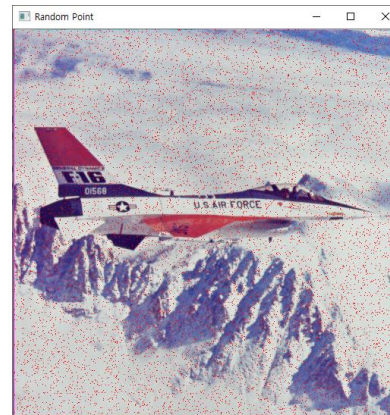
## 목 표 3. 영상파일 입출력 및 변환 결과



원본



파란색 및 빨간색 색상 변환



랜덤 픽셀 빨간색 색상 변환

감사합니다

