

Lab: Part 1

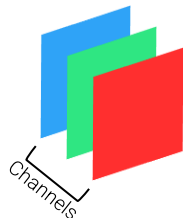
Image Manipulation in Python



2023/03/29 ~ 2023/04/09

Background

- BGR Format



- Blur



- BGRA Format



- Blend

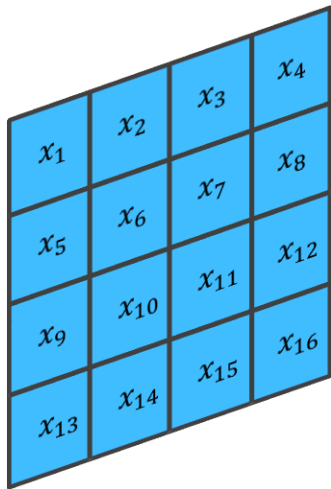
- Merge



- Overlay

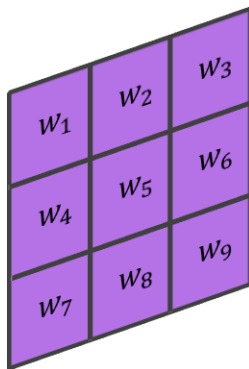


Image Blur



x_1	x_2	x_3	x_4
x_5	x_6	x_7	x_8
x_9	x_{10}	x_{11}	x_{12}
x_{13}	x_{14}	x_{15}	x_{16}

Input B Channel

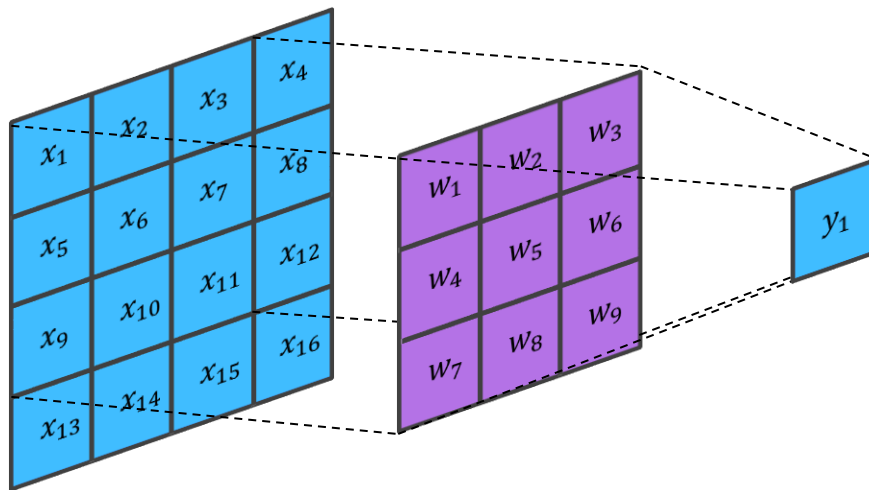


w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

3×3 Kernel

Image Blur

- Convolution



Input B Channel

3×3 Kernel

Output B Channel

$$\begin{aligned} y_1 = & x_1 w_1 + x_2 w_2 + x_3 w_3 \\ & + x_5 w_4 + x_6 w_5 + x_7 w_6 \\ & + x_9 w_7 + x_{10} w_8 + x_{11} w_9 \end{aligned}$$

Image Blur

- Convolution

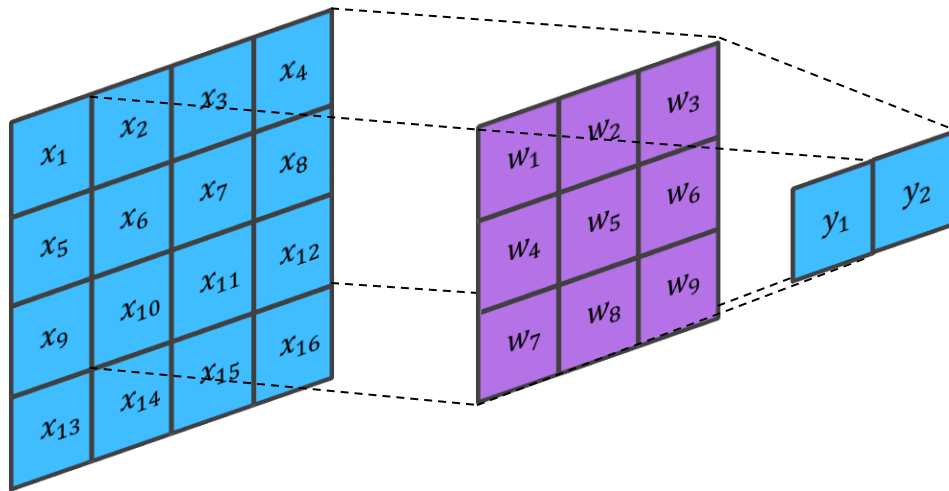


Image Blur

- Convolution

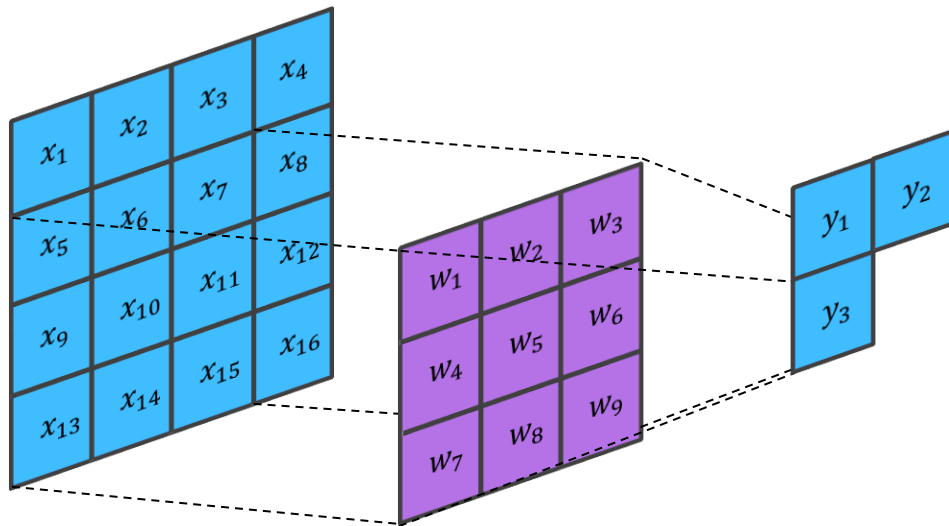


Image Blur

- Convolution

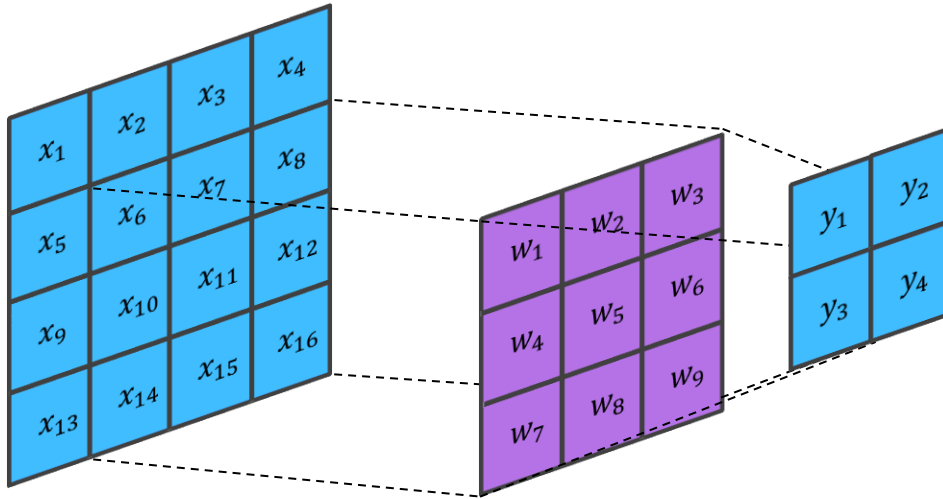


Image Blur

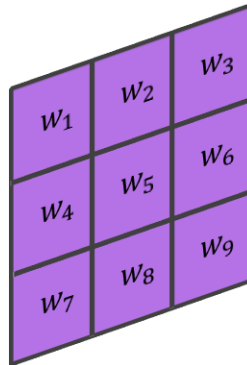
- **Kernel size can be different**

- In our implementation, we will use 3×3 , 5×5 , and 7×7
- Integer mode requires all kernel values to add up to 255
- Floating-point mode requires all kernel values to add up to 1.0
 - Since the kernel values are floating-points, the output must be casted to integer

Integer Mode

$$\begin{aligned} &w_1 + w_2 + w_3 \\ &+ w_4 + w_5 + w_6 \\ &+ w_7 + w_8 + w_9 = 255 \end{aligned}$$

$$w_5 = w_5 + 3$$



Floating-Point Mode

$$\begin{aligned} &w_1 + w_2 + w_3 \\ &+ w_4 + w_5 + w_6 \\ &+ w_7 + w_8 + w_9 = 1 \end{aligned}$$

Image Blur

- **Output image size depends on the kernel size**

- If kernel is $k \times k$, then the height and the width of the output is decreased by $k - 1$

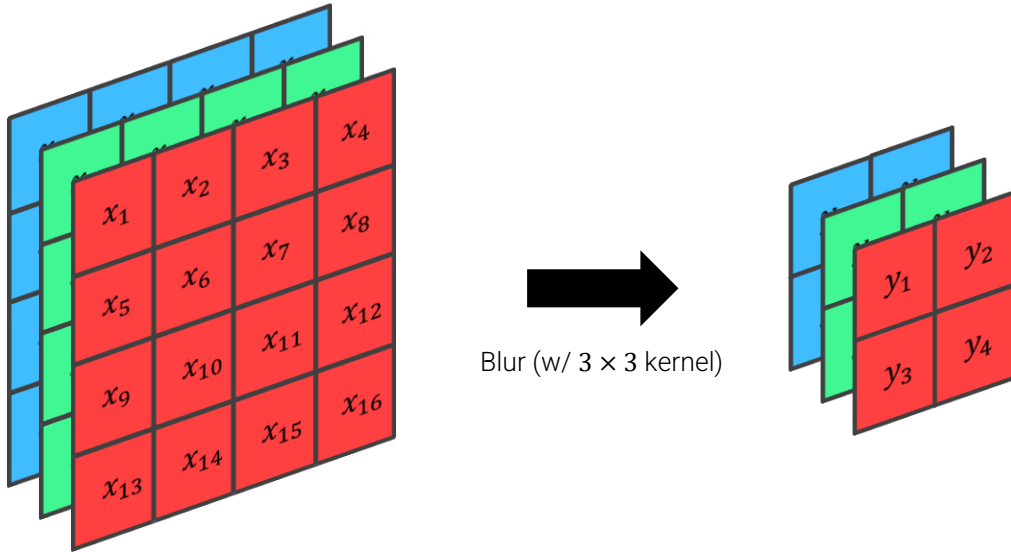
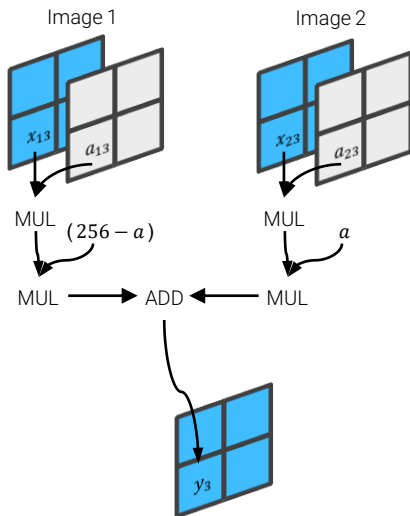


Image Blend

- Merge

- Merge two input pixels with its alpha-channel values and the alpha parameter
- Integer mode will require fixed-point arithmetic

BGR Integer Mode



BGR Floating-Point Mode

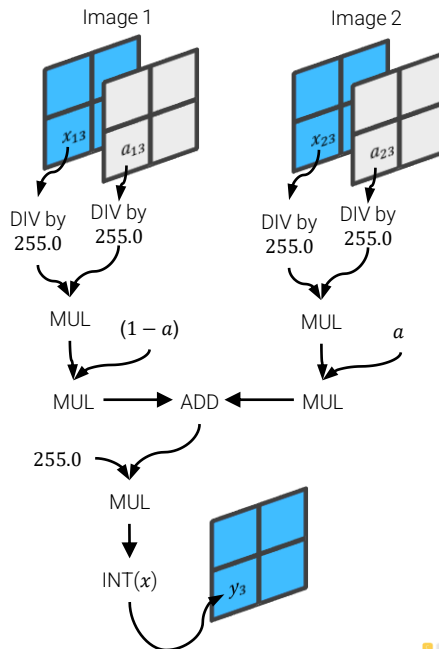


Image Blend

- Merge

- Merge two alpha-channel values with alpha parameter
- Integer mode will require fixed-point arithmetic

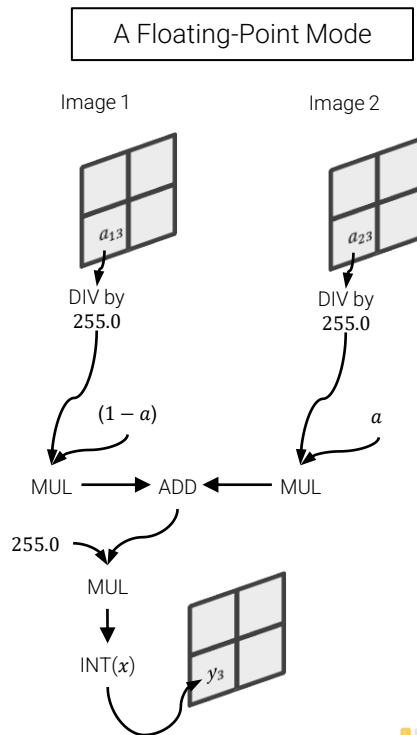
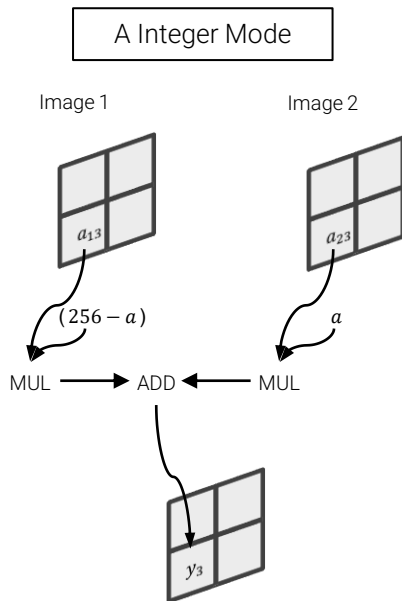
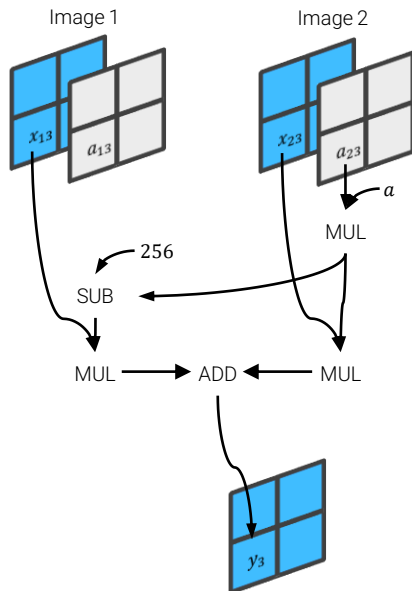


Image Blend

• Overlay

- Overlay a foreground(image 1) to the background(image 2)
- Integer mode will require fixed-point arithmetic
- To avoid any confusion, subtraction in integer mode means $(256 - a_{13} \cdot \alpha)$, and subtraction in floating-mode means $(1 - a_{13} \cdot \alpha)$

BGR Integer Mode



BGR Floating-Point Mode

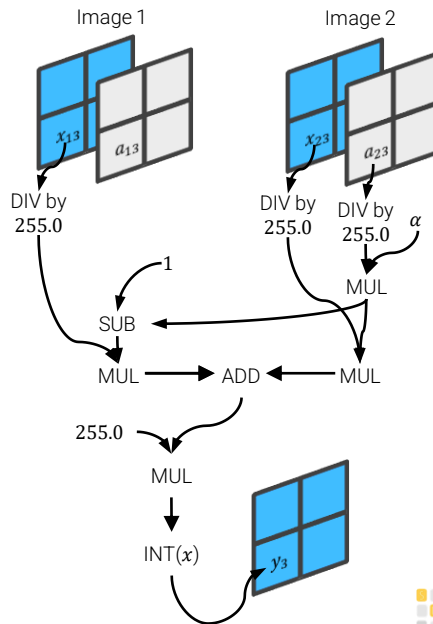
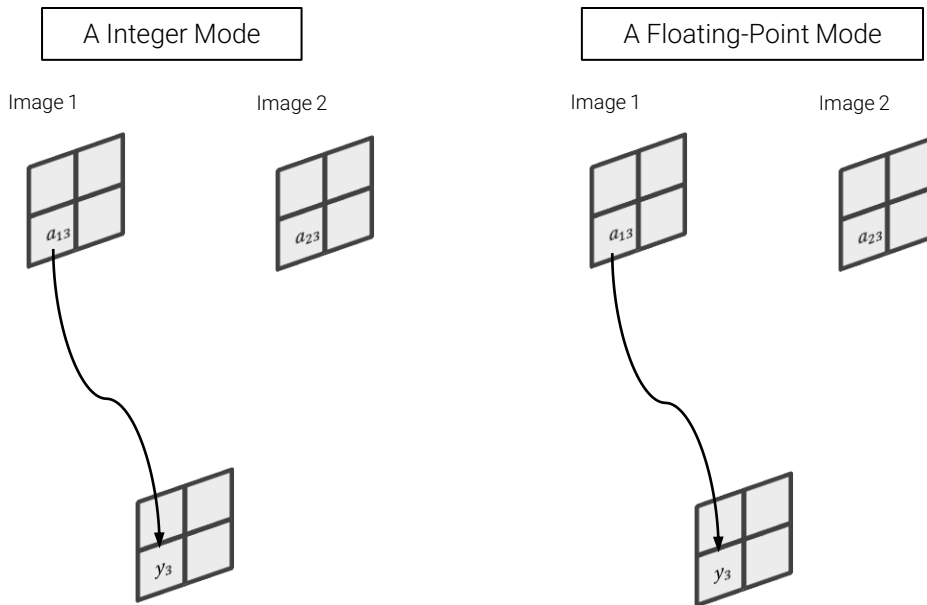


Image Blend

- **Overlay**

- Overlay a foreground(image 2) to the background(image 1)
- Alpha channel output value is set to that of the background's



Raw Image Format

- Input and output to `blend()` and `blur()` should be raw image format
- Raw image format uses multi-level list

- To access a pixel at channel `c`, height `h`, and width `w`:

`image[h-1][w-1][c-1]`

```
[[[255, 255, 255, 255],  
  [255, 255, 255, 255],  
  [255, 255, 255, 255],  
  ...,  
  [255, 255, 255, 255],  
  [255, 255, 255, 255],  
  [255, 255, 255, 255]],  
 ...,  
 [[255, 255, 255, 255],  
  [255, 255, 255, 255],  
  [255, 255, 255, 255],  
  ...,  
  [255, 255, 255, 255],  
  [255, 255, 255, 255],  
  [255, 255, 255, 255]],  
 [255, 255, 255, 255],  
 [255, 255, 255, 255],  
 ...,  
 [255, 255, 255, 255],  
 [255, 255, 255, 255],  
 [255, 255, 255, 255]]]
```

Overview

- Project structure

```
image-lab
|-> blend_driver.py
|-> blend_float.py
|-> blend_int.py
|-> blur_driver.py
|-> blur_float.py
|-> blur_int.py
|-> img2raw.py
|-> imlib.py
-> raw2img.py
```

- Commands

```
// img2raw
python img2raw.py /path/to/img
// raw2img
python raw2img.py /path/to/raw
// blur
python blur_driver.py /path/to/raw --kernel 3x3
// blend
python blend_driver.py /path/to/raw1 /path/to/raw2
// img2raw -> blend -> raw2img
python img2raw.py /path/to/img
python img2raw.py /path/to/img
python blend_driver.py /path/to/raw1 /path/to/raw2
python raw2img.py /path/to/raw
// convert all images to raw
cd images
make all
```

- Notes

- No libraries should be used in the skeleton code
- Use for loops for implementations
- Fork the lab and submit by pushing your code with "Submission" tag
- Try to ask questions after pushing your code to Gitlab
- Make sure your Gitlab username is your student ID