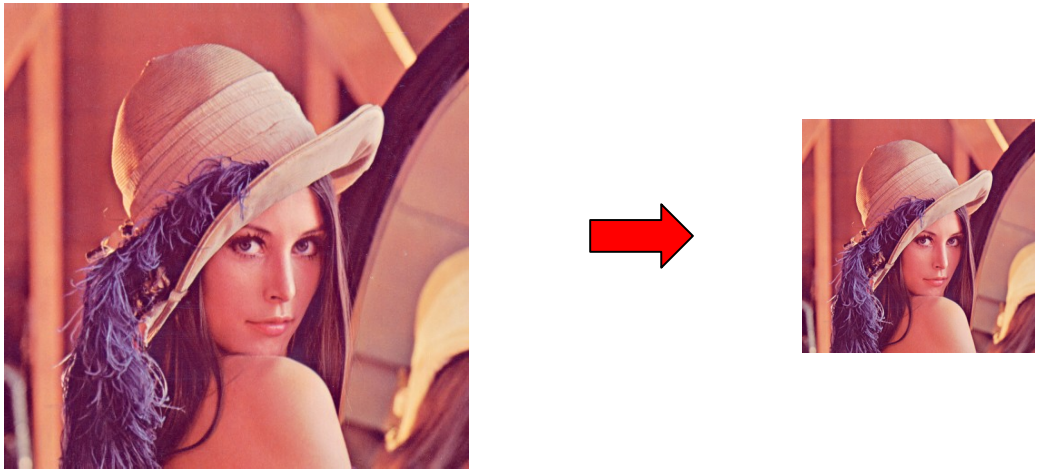
UNIVERSITÀ DI PARMA

LABORATORY #1

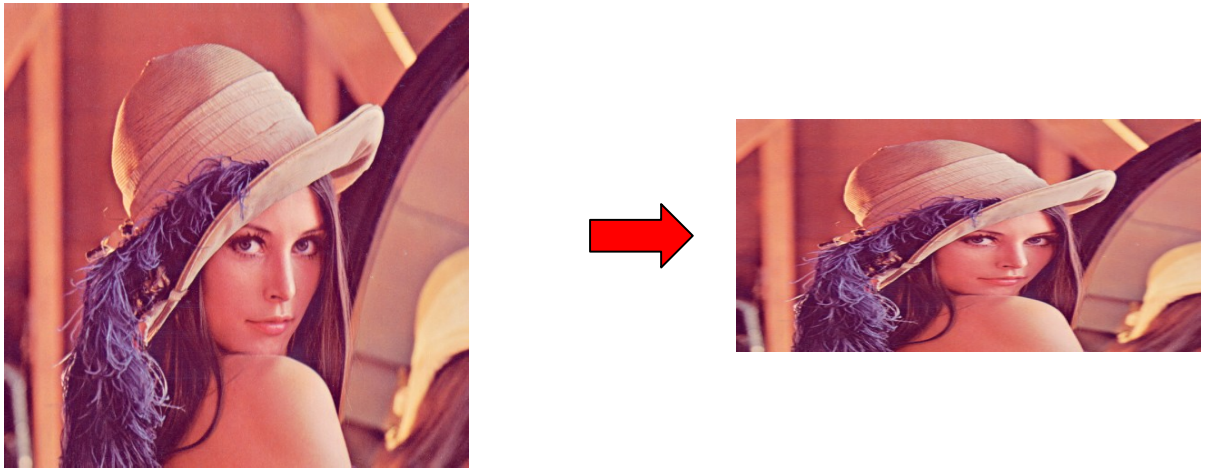
- Load lenna.png
- Print on console R G B channels using std::cout
- Initially in order
- Then with a \n at the end of each line

OpenCV uses BGR as default!

- Subsample lenna.png (2x)
- Basically create (and show) a new image with w/2 and h/2 size
- Take 1 column and 1 row every 2 from original image



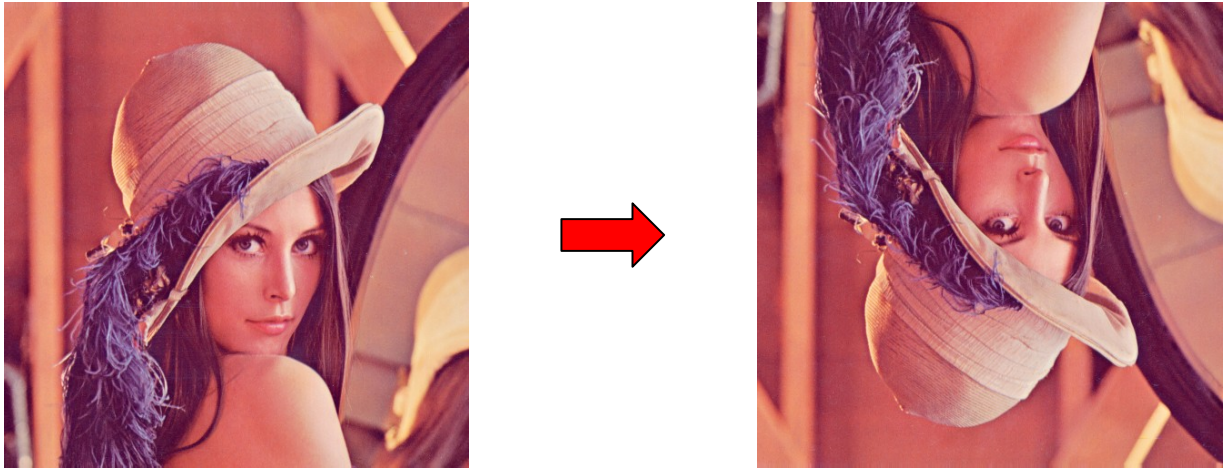
- Subsample lenna.png (2x) for rows only…
- Basically create (and show) a new image with w and h/2 size
- Use 1 row every 2 from original image



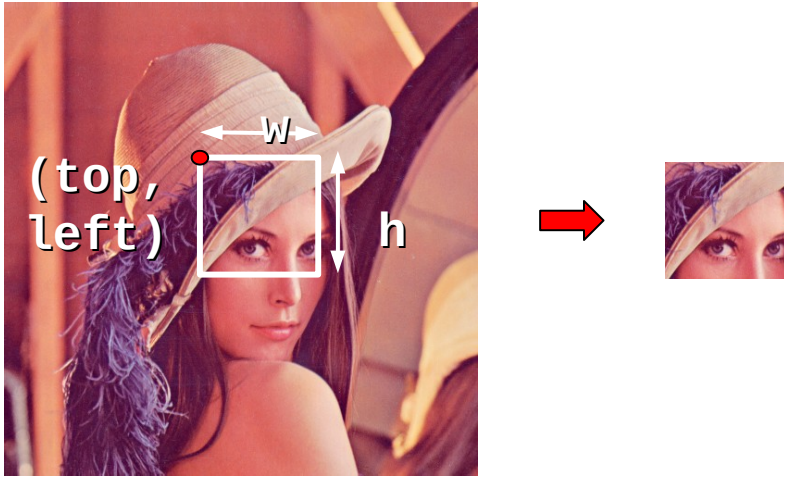
- Subsample lenna.png (2x) for columns only…
- Basically create (and show) a new image with w/2 and h size
- Use 1 column out of 2 from original image

- Horizontally flip lenna.png
- Create a new image (w*h) inverting columns order

- Vertically flip lenna.png
- Create a new image (w*h) inverting rows order



- Crop lenna.png
- Take from input (getopt()) the coordinates and size of a cropping area
- Create and show a new image using the defined cropping area
  - Check cropping area
  - Use cv::Mat::data only
  - Then use cv::Mat specific constructor



#7bis

UNIVERSITÀ DI PARMA

- Same as previous but
- Random position and size
- Please check image vs vrop position/size
  - Mandatory checks!

- Create a lenna image with padding
- 0 padding
- Use getopt() to specify padding size

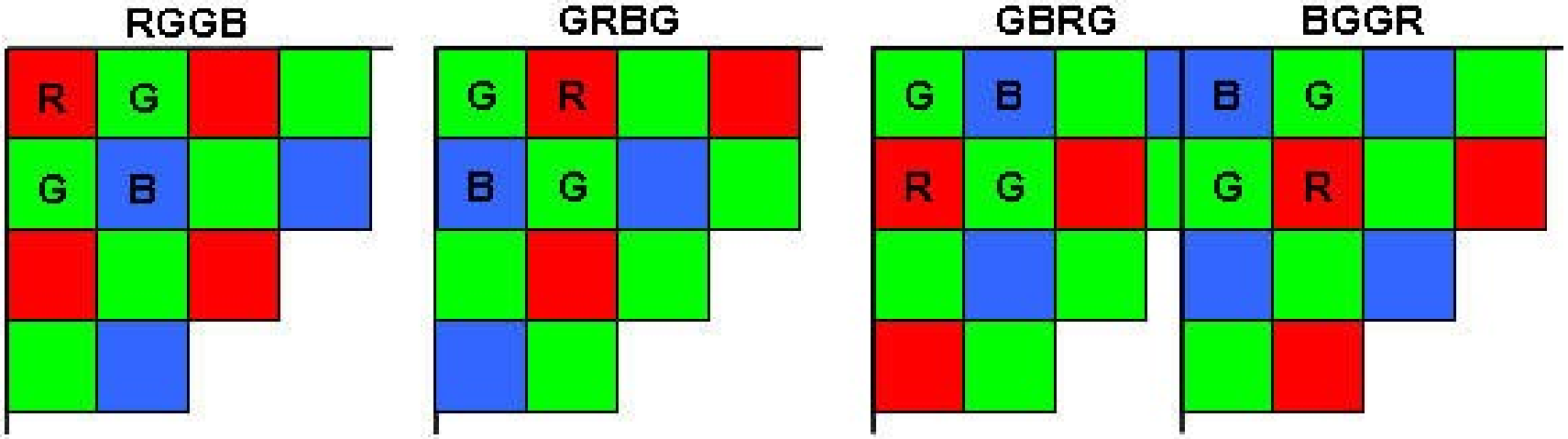
- Split lenna image in four parts (same size)
- Create a new image using those parts shuffling them

- Shuffle color channels
  - i.e. R⟶G, B⟶R, G⟶B

Bayer patterns

UNIVERSITÀ DI PARMA

- Load images in the bayer folder
- They are coded using some bayer pattern
  - Look at file name!



- Develope the following demosaicing procedure (DOWNSAMPLE)
- Output image should be CV_8UC1 with w/2 and h/2 size
- Use a 2x2 sliding window
- Use the average of G channels

- Develope the following demosaicing procedure (LUMINANCE)
- Output image should be CV_8UC1 with w and h size
- Use a 2x2 sliding window with the following formula
  - R*.3 + (G1 + G2)*.59/2.0 + B*.11
- Warning: you need to consider that for each sliding step the pattern is different…

- Develope the following demosaicing procedure (SIMPLE)
- Output image should be CV_8UC3 with w and h size
- Use a 2x2 sliding window and put channel values as
  - R and B ⟶ original R and B
  - G ⟶ average of 2 Gs
- Warning: you need to consider that for each sliding step the pattern is different…