Visual Analytics

Session 3: Even more image processing

Ross Deans Kristensen-McLachlan

rdkm@cas.au.dk

Course outline

- 1. Introducing Visual Analytics
- 2. Basic image processing
- **3. More image processing**
- 4. Convolutional kernels
- 5. Image classification 1
- 6. Image classification 2
- 7. From shallow to deep learning
- 8. Convolutional neural networks
- 9. Pretrained CNNs and transfer learning
- 10. Image embeddings
- 11. Project presentations
- 12. Text-to-image models
- 13. Project development

Plan for today

- No assignment from last week

- 1. More image processing
    - Translating, rotation
    - Bitwise operations and masks

- 2. Code-along session
    - Colour histograms, comparing images, masking
    - Introducing this week's assignment

New concepts

- We’re going to be looking at a number of techniques today that are super useful when doing image processing
  - Translation
  - Rotation
  - Bitwise operations
  - Masks

Translation

- One of the simplest techniques we can use in image processing is what is called *translation*

- Translation basically means move the centre of the image, so the whole image shifts

Original image

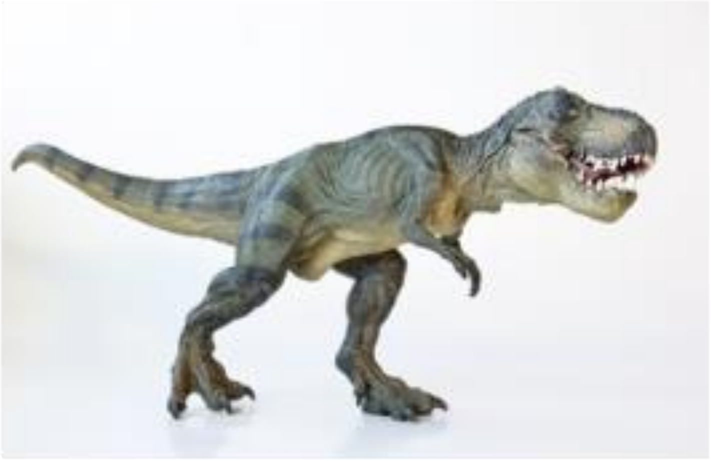
Translation

- One of the simplest techniques we can use in image processing is what is called *translation*

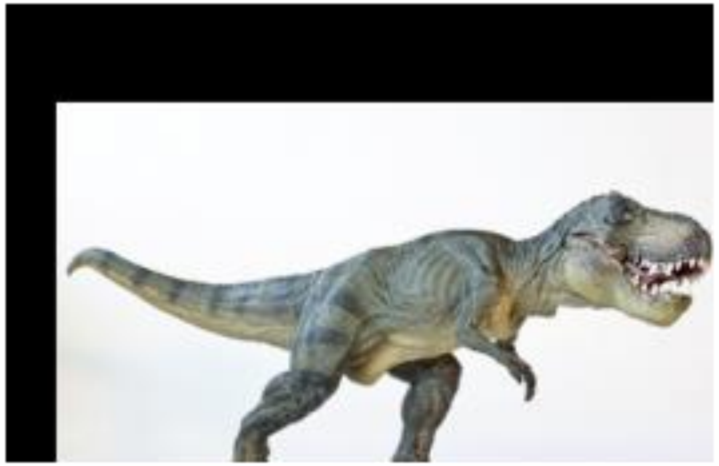
- Translation basically means move the centre of the image, so the whole image shifts

- In this case, the image has been moved 25 pixels right and 50 pixels down

Original image



Shifted down and right



Translation

- To do this, we have to *multiply* the original array by what's called a *translation matrix:*

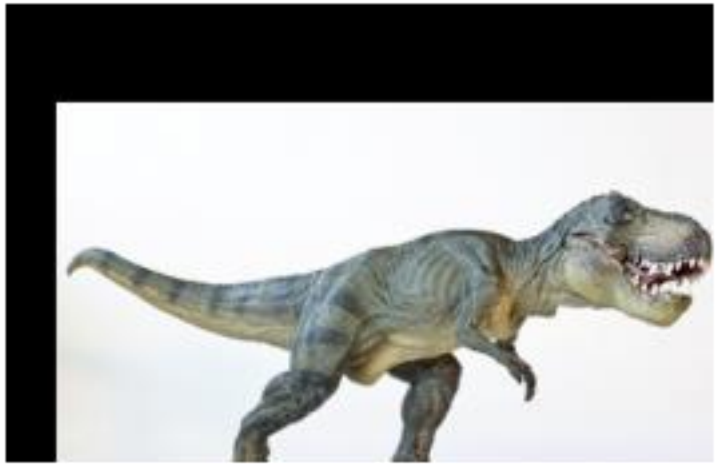
$$M = \begin{bmatrix} size & rotation & location \\ rotation & size & location \end{bmatrix}$$

- So in this case:

$$M = \begin{bmatrix} 1 & 0 & 25 \\ 0 & 1 & 50 \end{bmatrix}$$

Original image

Shifted down and right



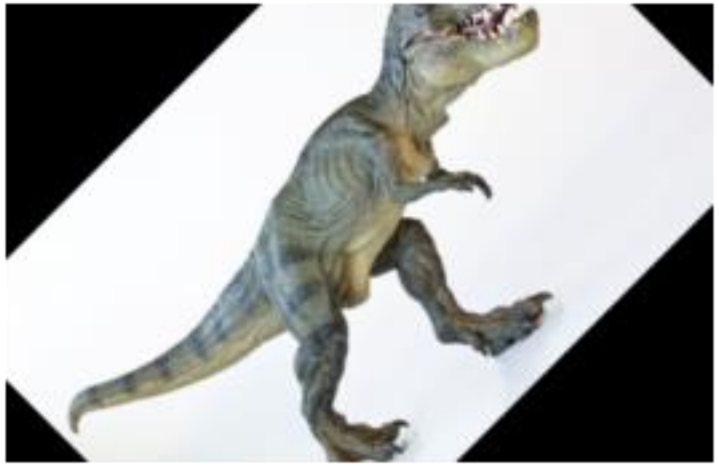
Rotation

- As we’ll see in the code along later on, rotation works in essentially the same way

- This time, we’re defining how much we want to rotate around the centre

- Why might translation and rotation be useful for image processing?

Original image



Bitwise operations

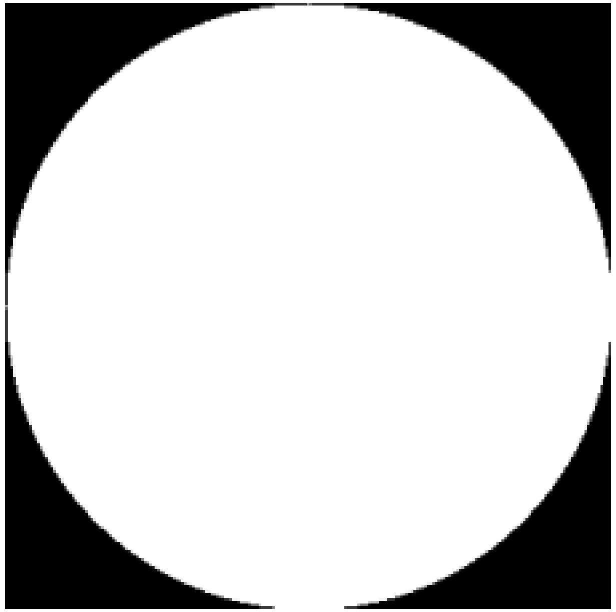
- Bitwise operations are basic low level computational operations
- Essentially, you’re manipulating the actual *bits* saved in memory
- See link in additional readings for a nice blogpost-style introduction
- Similar to logical operations but *not entirely identical*

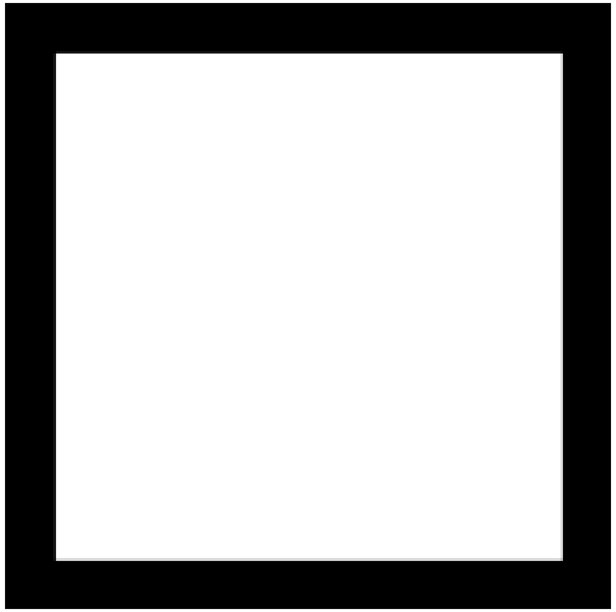
Bitwise operations

- For our purposes, there are four bitwise operators:
  - **AND:** A bitwise AND is true if and only if both pixels are greater than zero.
  - **OR:** A bitwise OR is true if either of the two pixels are greater than zero.
  - **XOR:** A bitwise XOR is true if and only if *either* of the two pixels are greater than zero, but not both.
  - **NOT:** A bitwise NOT inverts the “on” and “off” pixels in an image.

Bitwise operations

- Here we have two very simple images
- White regions are *1* and black regions are *0*
    - *Technically white regions are 255!*
- If we combine these using the bitwise operations on the previous slide, what would the resulting images look like?





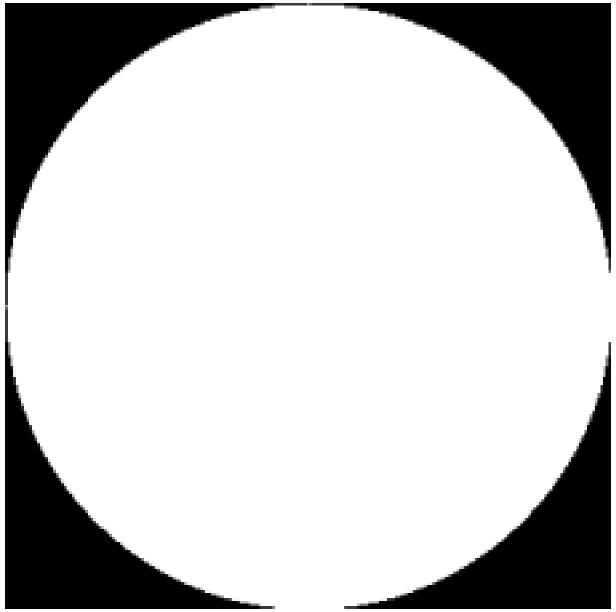
Bitwise operations

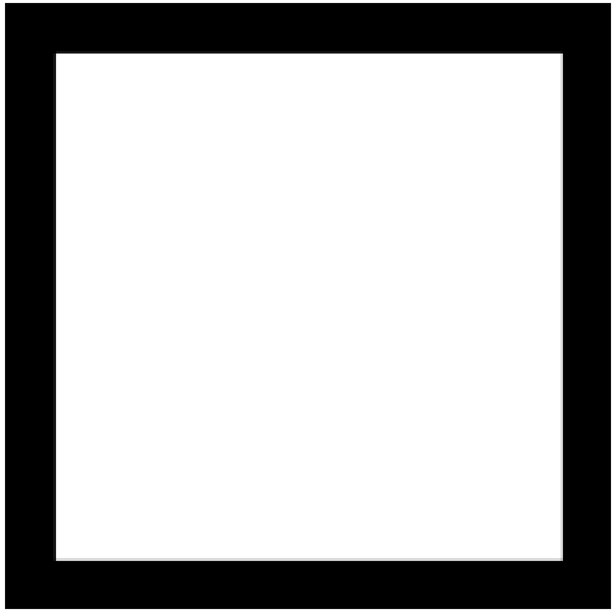
AND: A bitwise AND is true if and only if both pixels are greater than zero.

OR: A bitwise OR is true if either of the two pixels are greater than zero.

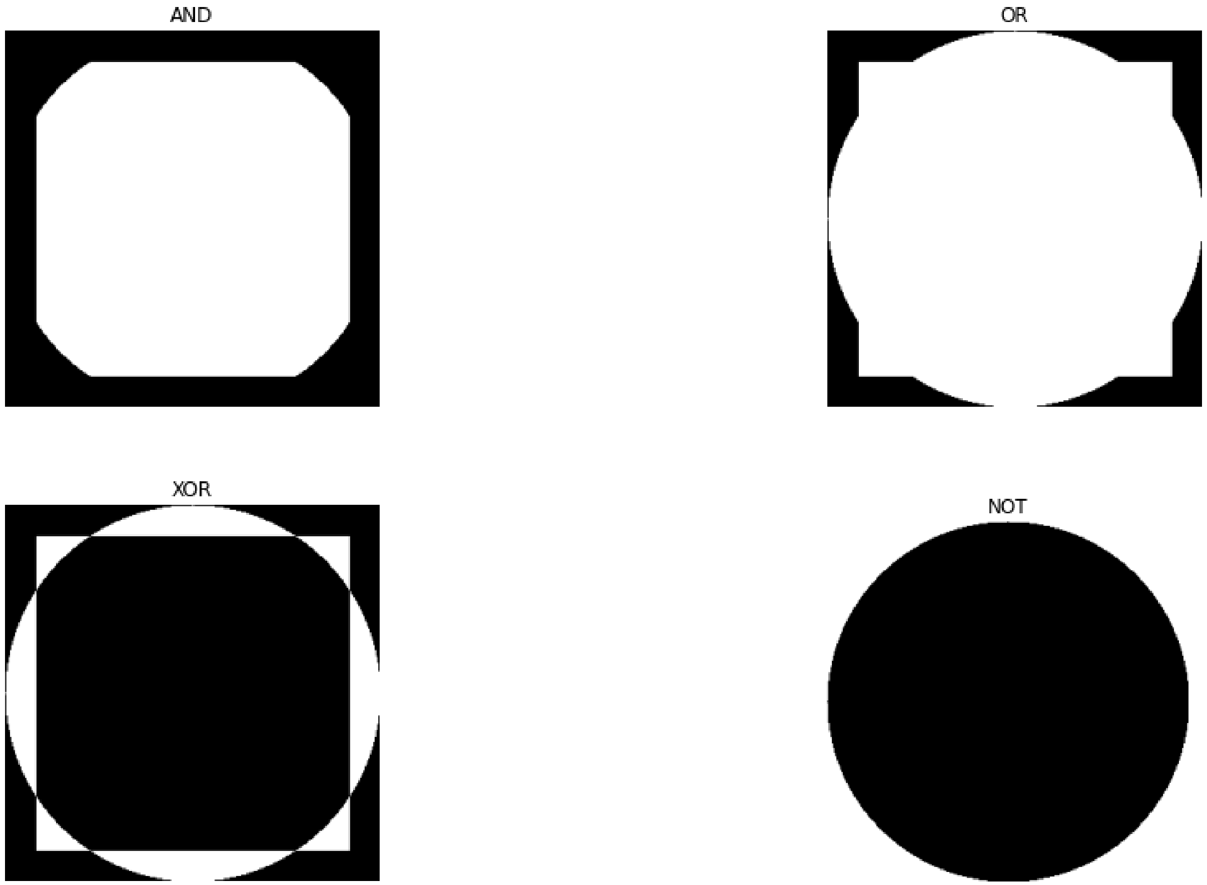
XOR: A bitwise XOR is true if and only if *either* of the two pixels are greater than zero, but not both.

NOT: A bitwise NOT inverts the “on” and “off” pixels in an image.



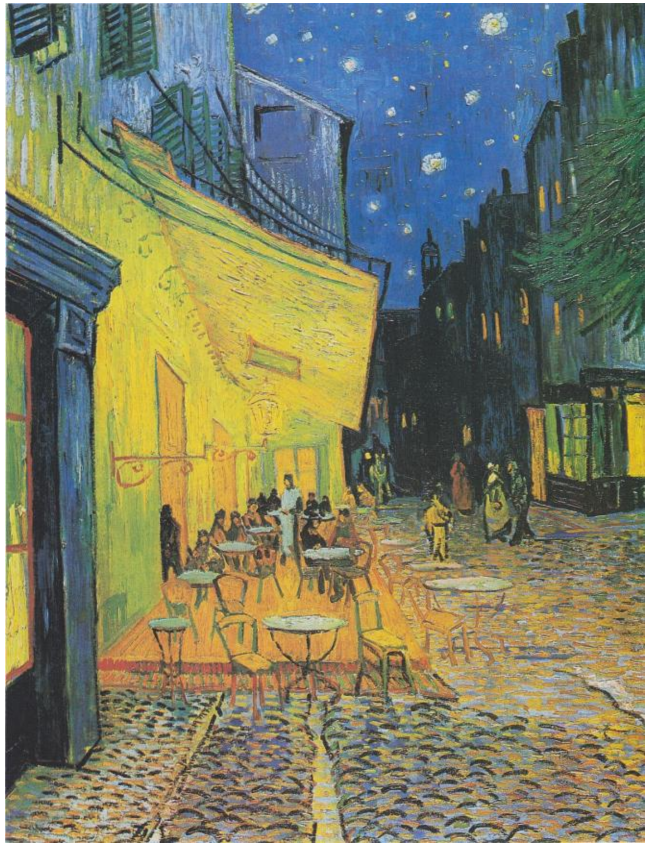


Bitwise operations

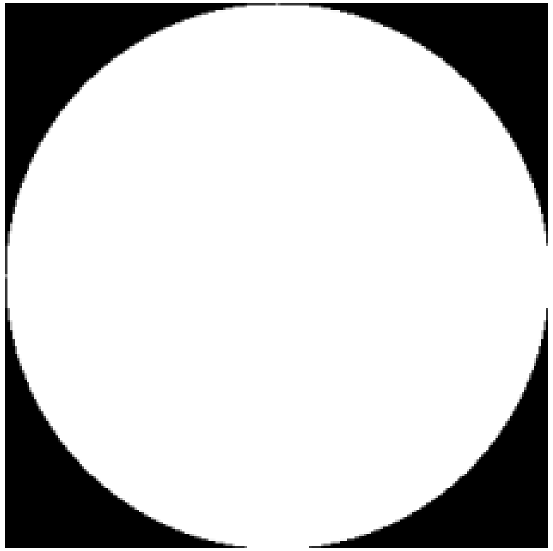


Masking

- The kind of bitwise operations are useful when we want to pick out only a specific part of an image



1576 x 1200

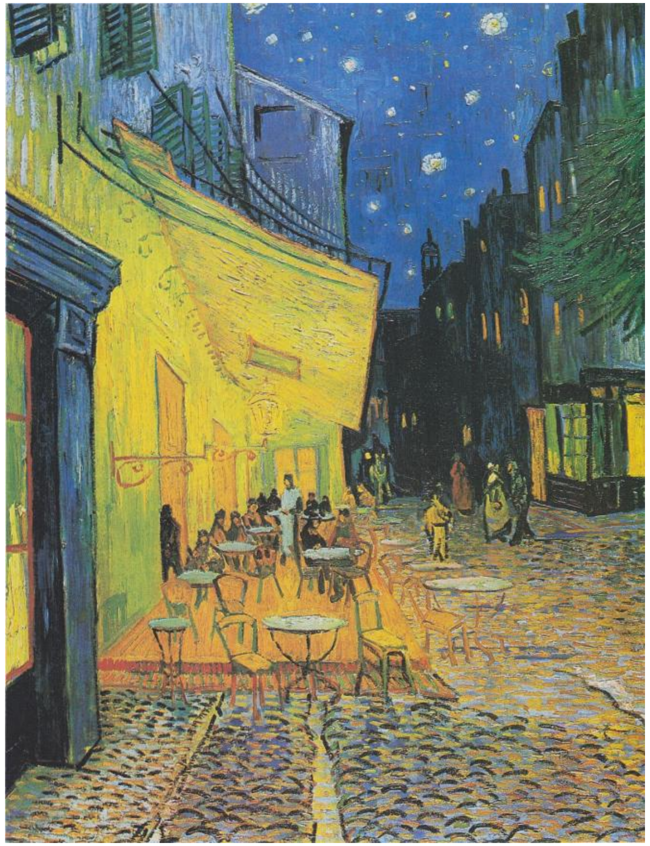


Radius = 500px

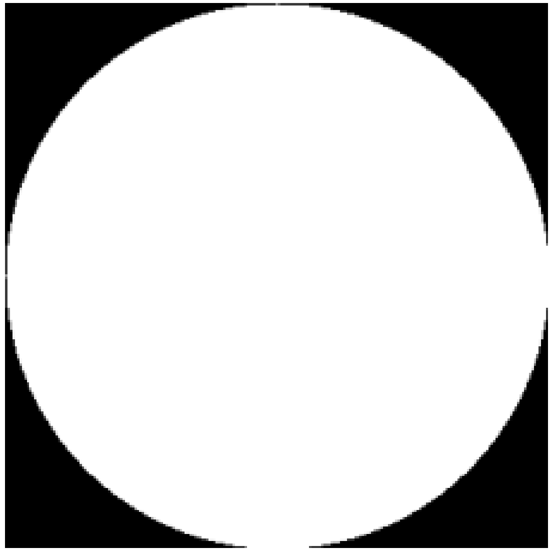
Masking

- The kind of bitwise operations are useful when we want to pick out only a specific part of an image

- Here we have a circle of radius 500 pixels – pixels in the circle are ‘on’ and the black pixels are ‘off’



1576 x 1200



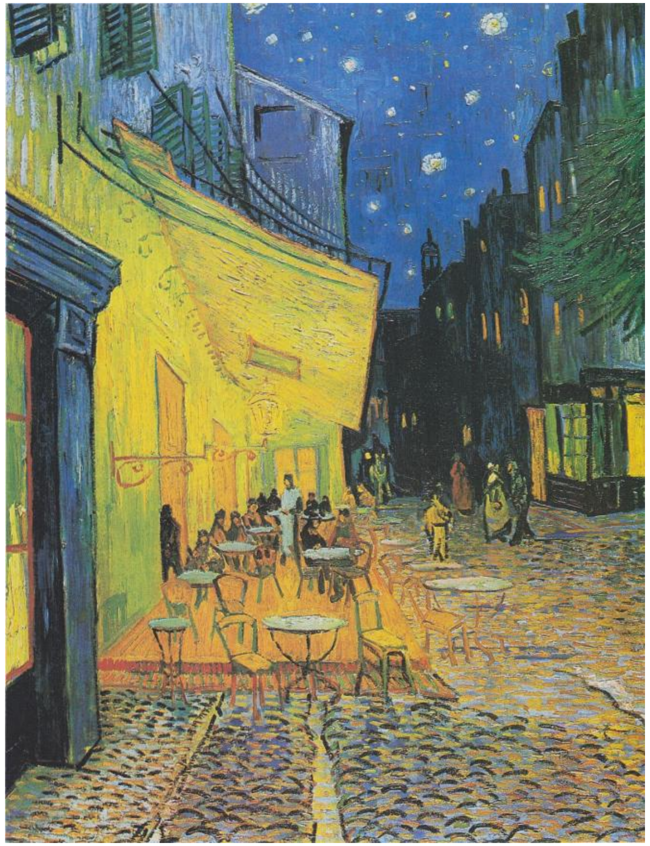
Radius = 500px

Masking

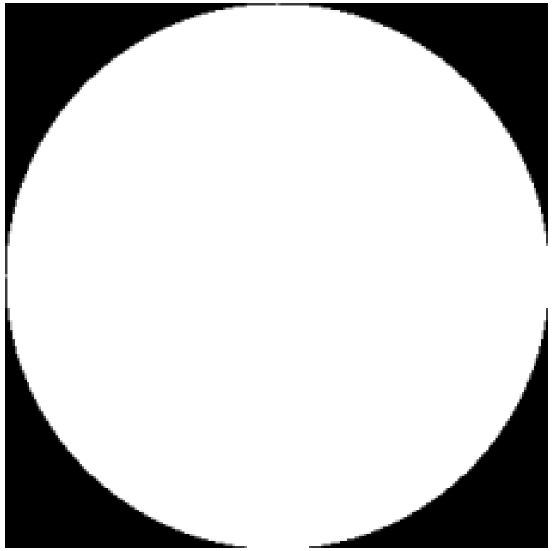
- The kind of bitwise operations are useful when we want to pick out only a specific part of an image

- Here we have a circle of radius 500 pixels – pixels in the circle are ‘on’ and the black pixels are ‘off’

- What happens if we combine these using the bitwise AND operator?

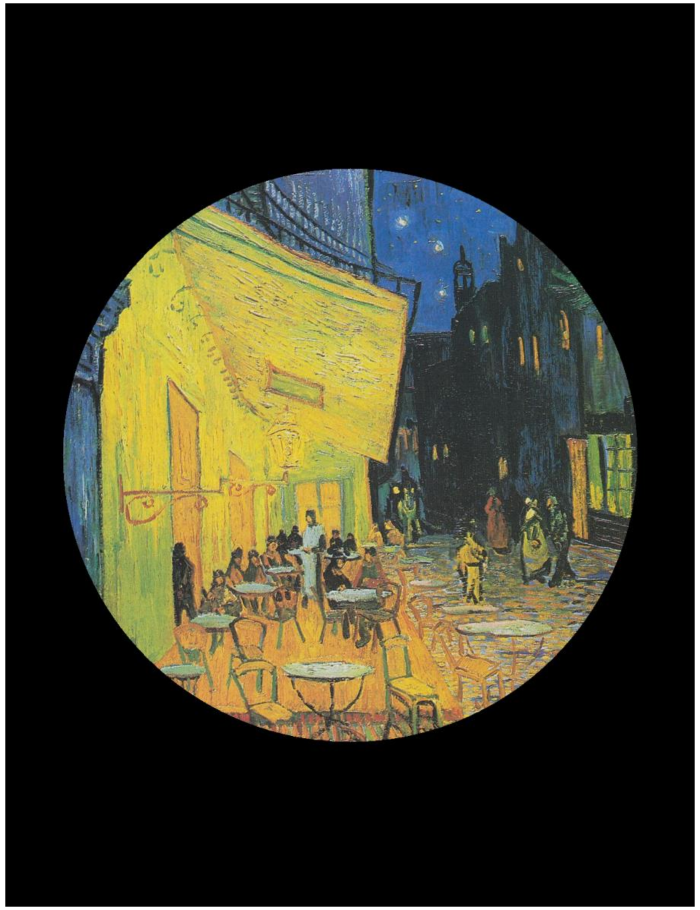


1576 x 1200

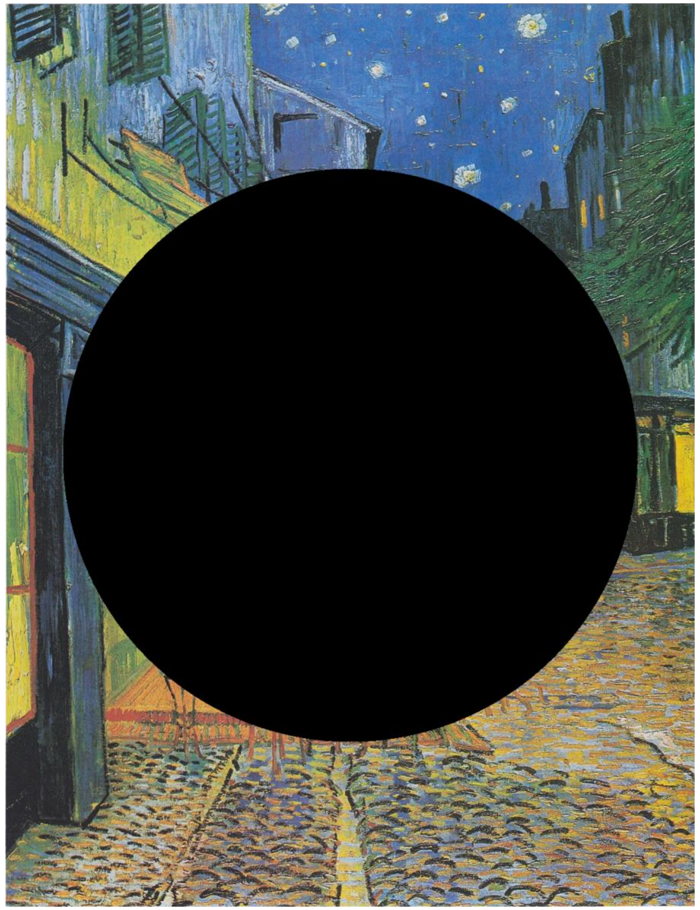


Radius = 500px

Masking – bitwise AND



Masking – subtracting mask



Take home points

- Translation and rotation allow us to manipulate the appearance of an image with some simple multiplication of matrices

- Bitwise operations are useful for manipulating images

- We can create *masks* which allow us to focus on particular parts of an image
  - Subtracting a mask means we are left with just the background

- Translation matrix, rotation matrix, masks – they're all just matrices of values whose size and values determine how the original image is to be transformed

- In computer vision, we call matrices of this sort *kernels* – we're going to be hearing a lot more about them!

Break

And then over to Ucloud!

Additional reading

- Carvalho, T. ‘Transformations with OpenCV’, https://towardsdatascience.com/transformations-with-opencv-ff9a7bea7f8b [Accessed February 2022]

- Zaczyński, B. ‘Bitwise operators in Python’, https://realpython.com/python-bitwise-operators/ [Accessed February 2022]