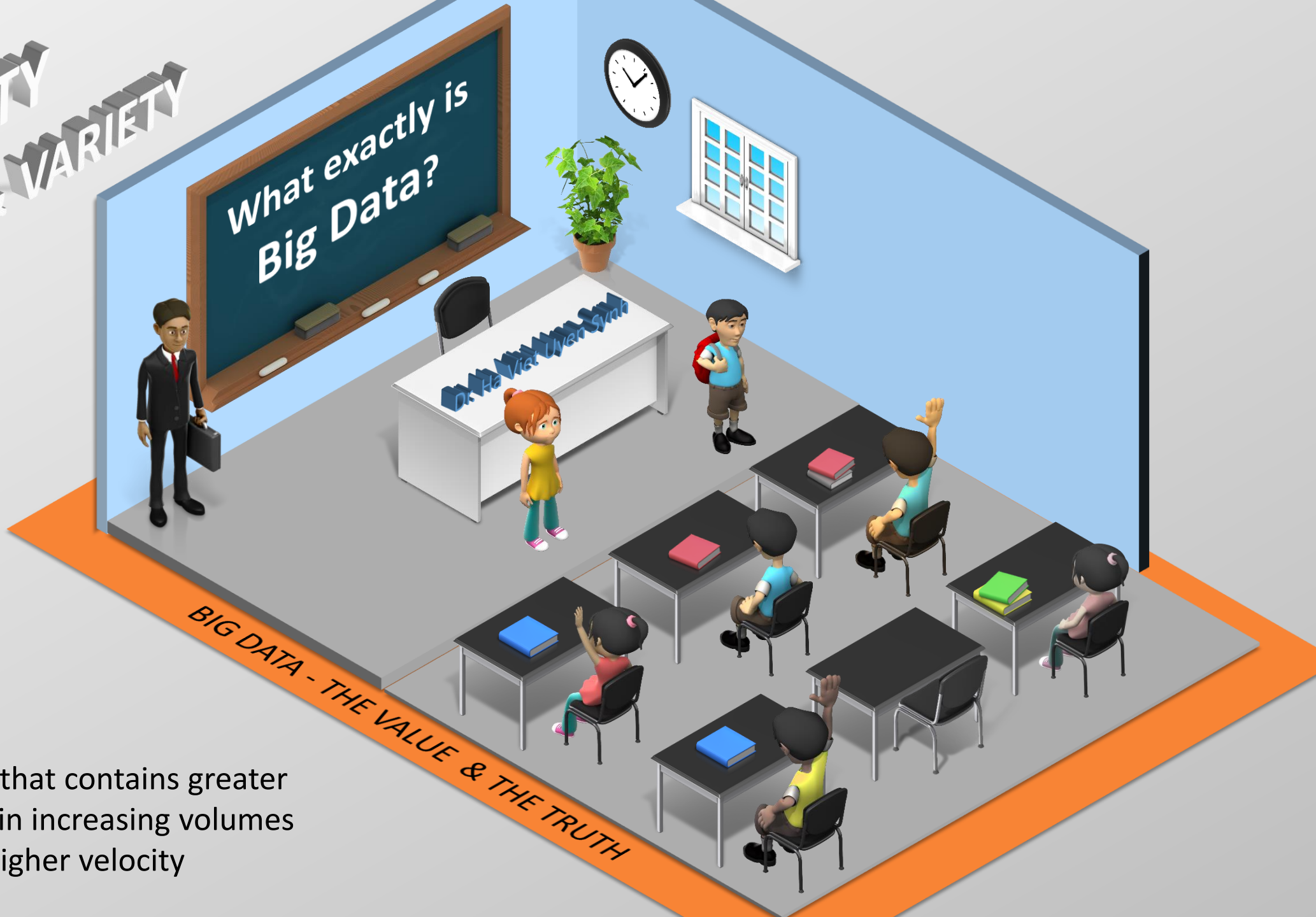

HOCHIMINH CITY INTERNATIONAL UNIVERSITY
HCM-IU
SCHOOL
Past, Present & Future
BIG DATA
POLICE

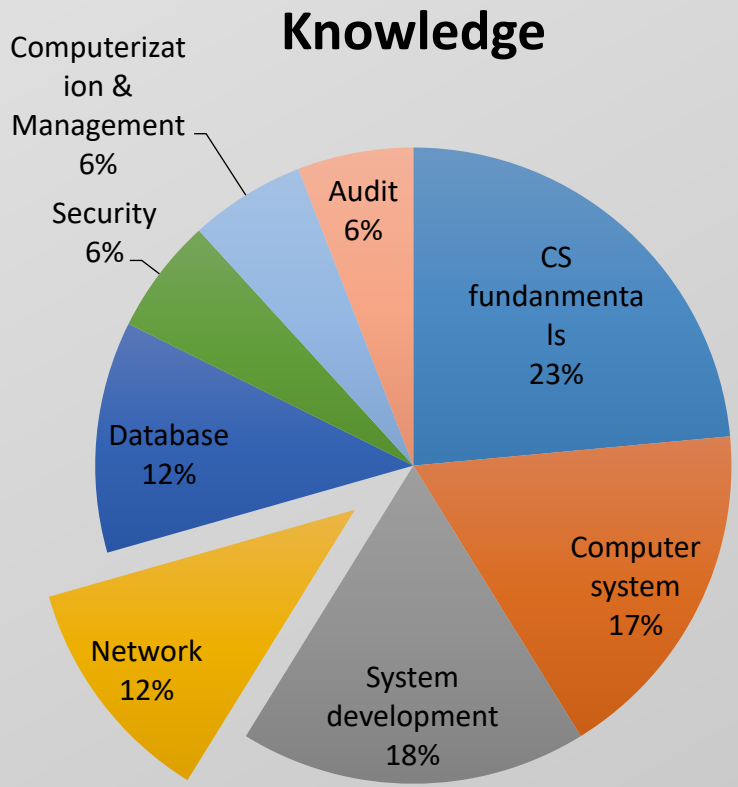


Big data is larger, more complex data sets, especially from new data sources

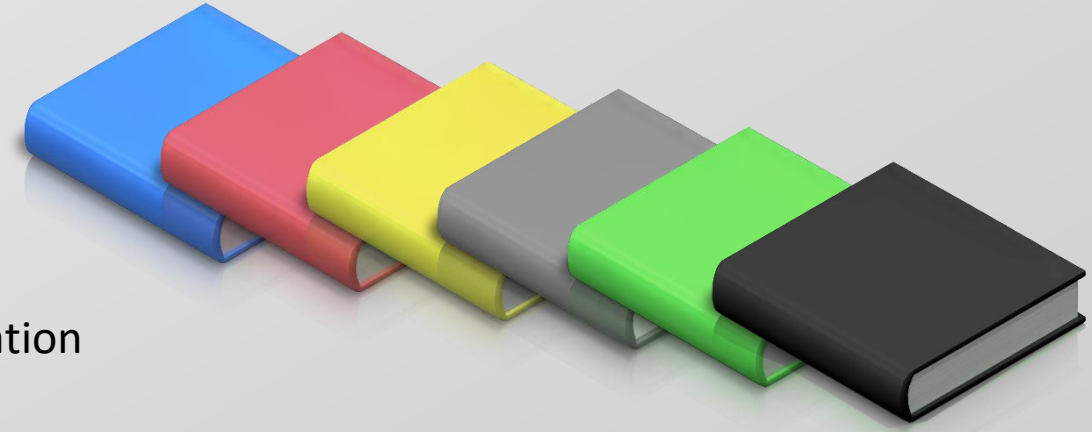
Big data is data that contains greater variety arriving in increasing volumes and with ever-higher velocity

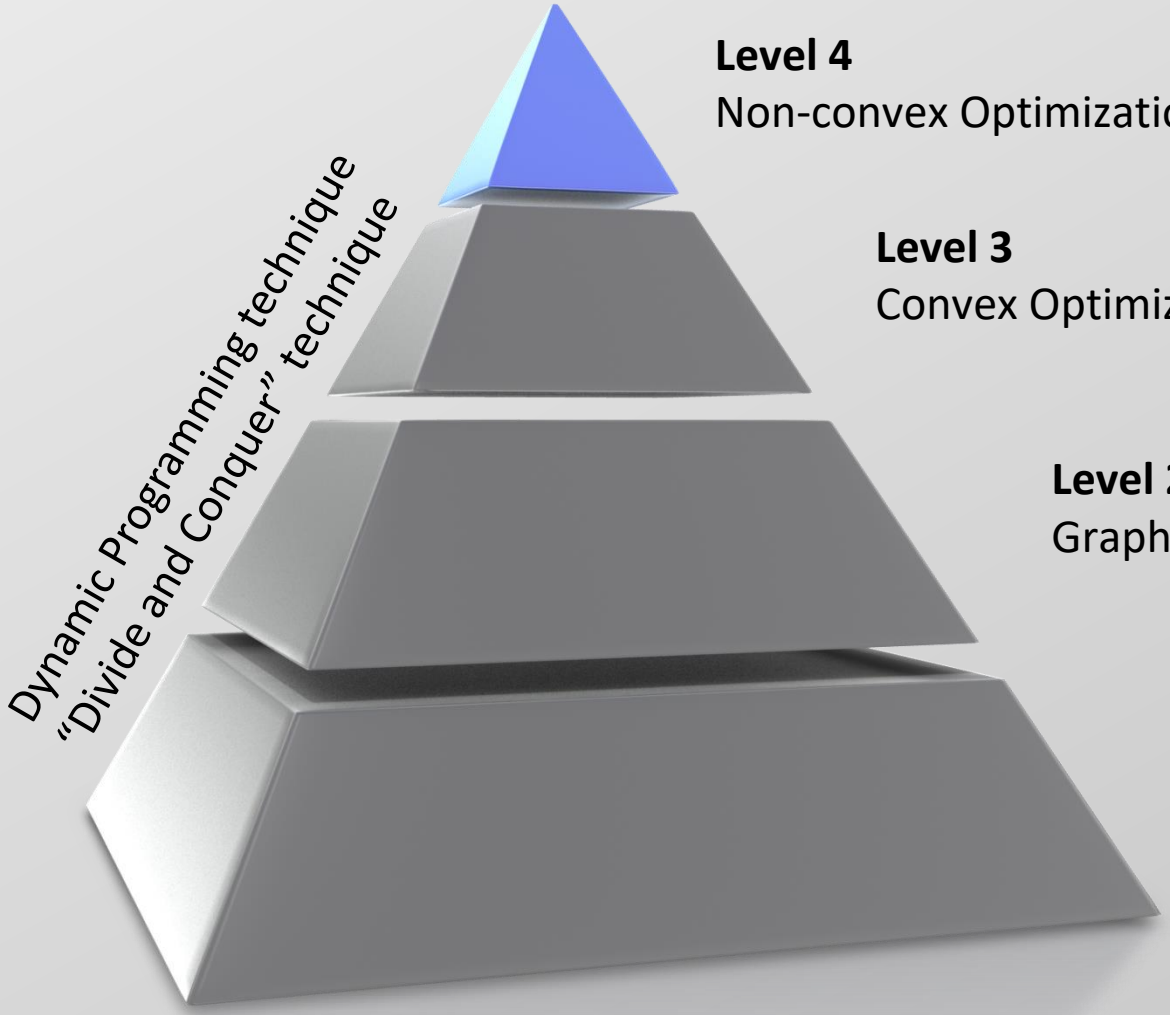
The Knowledge of IT Engineers

- The knowledge of Computer Engineer deals with the **theoretical foundations** of information and computation, and with **practical techniques** for their implementation and application.



How to solve a problem in computer science?





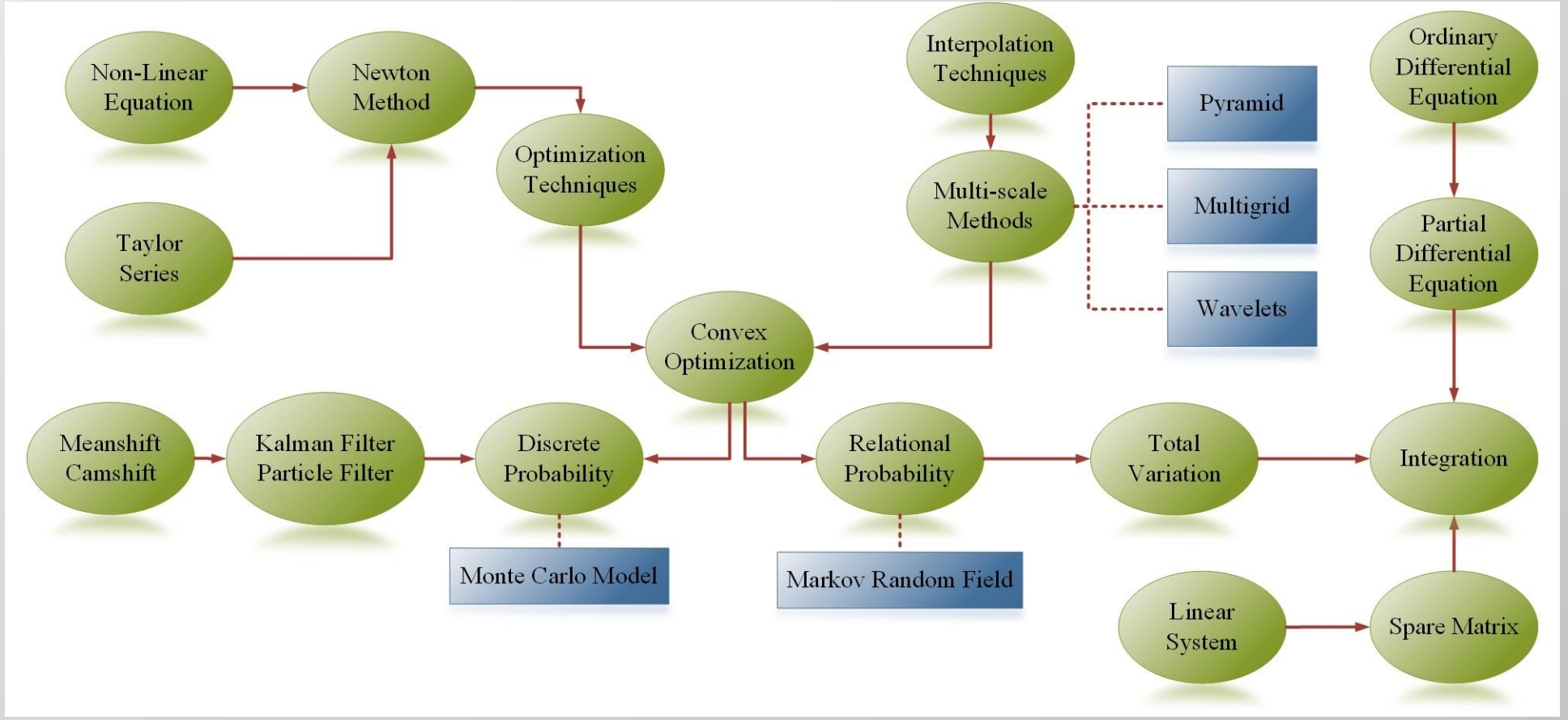
Level 4
Non-convex Optimization

Level 3
Convex Optimization

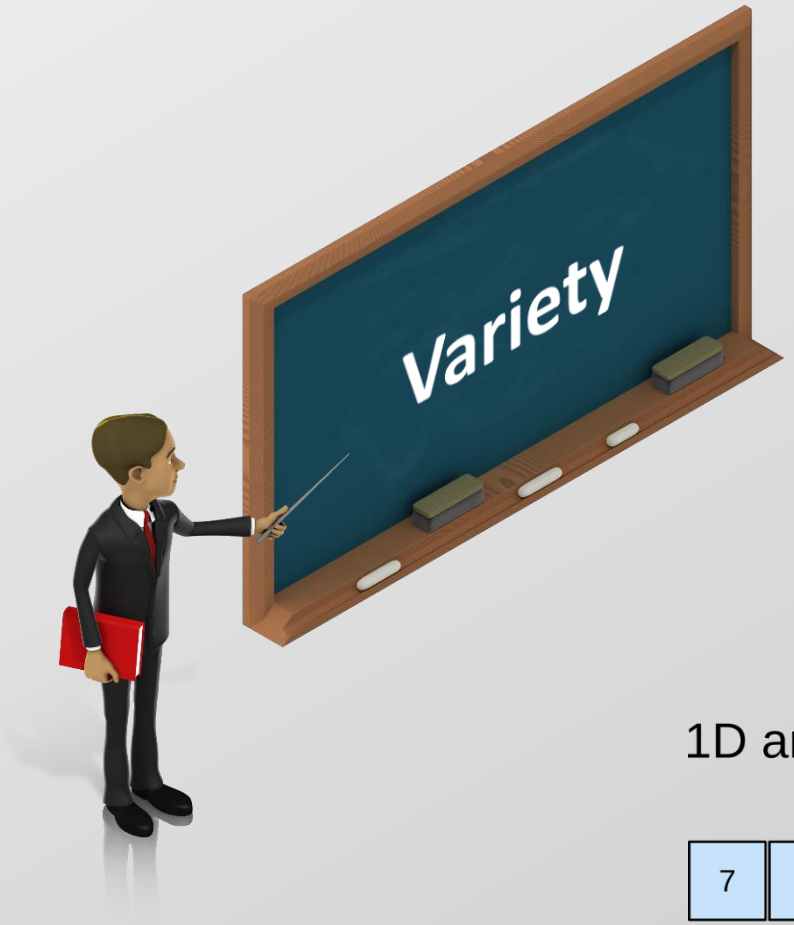
Level 2
Graphs

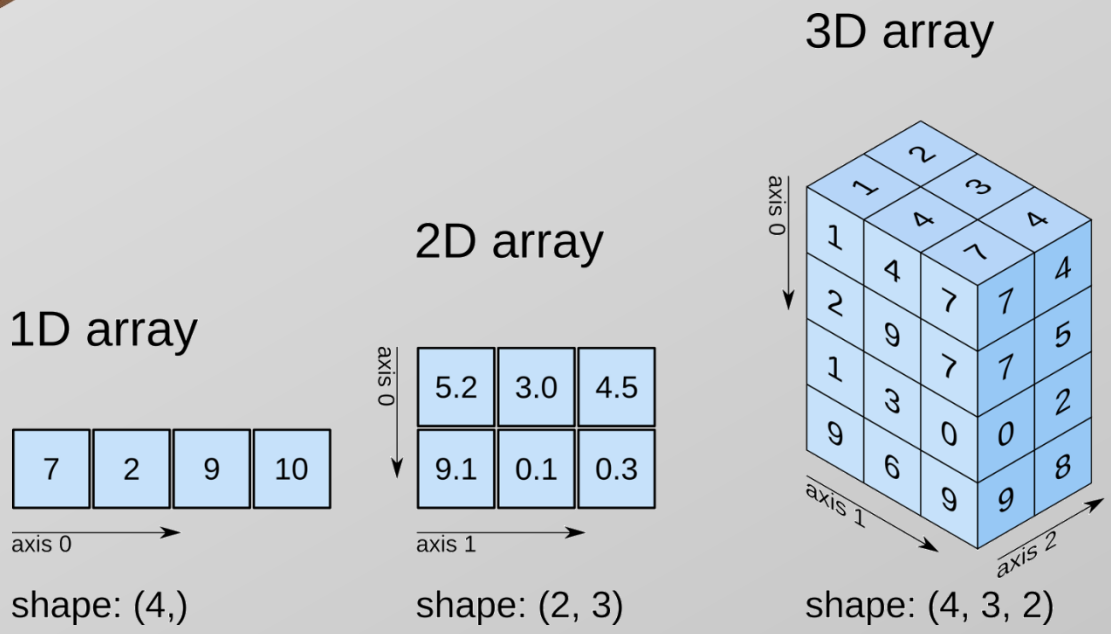
Level 1
"Exhausted check" method

The Mind map of Engineering Mathematics

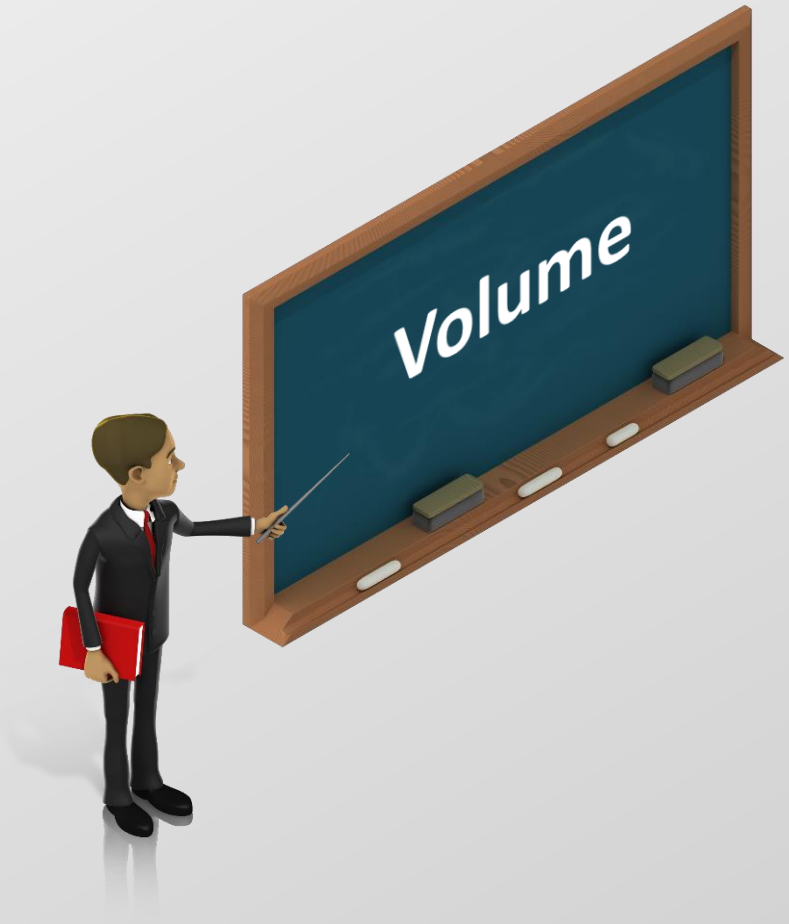


1. How to find a right solution in Big Data?





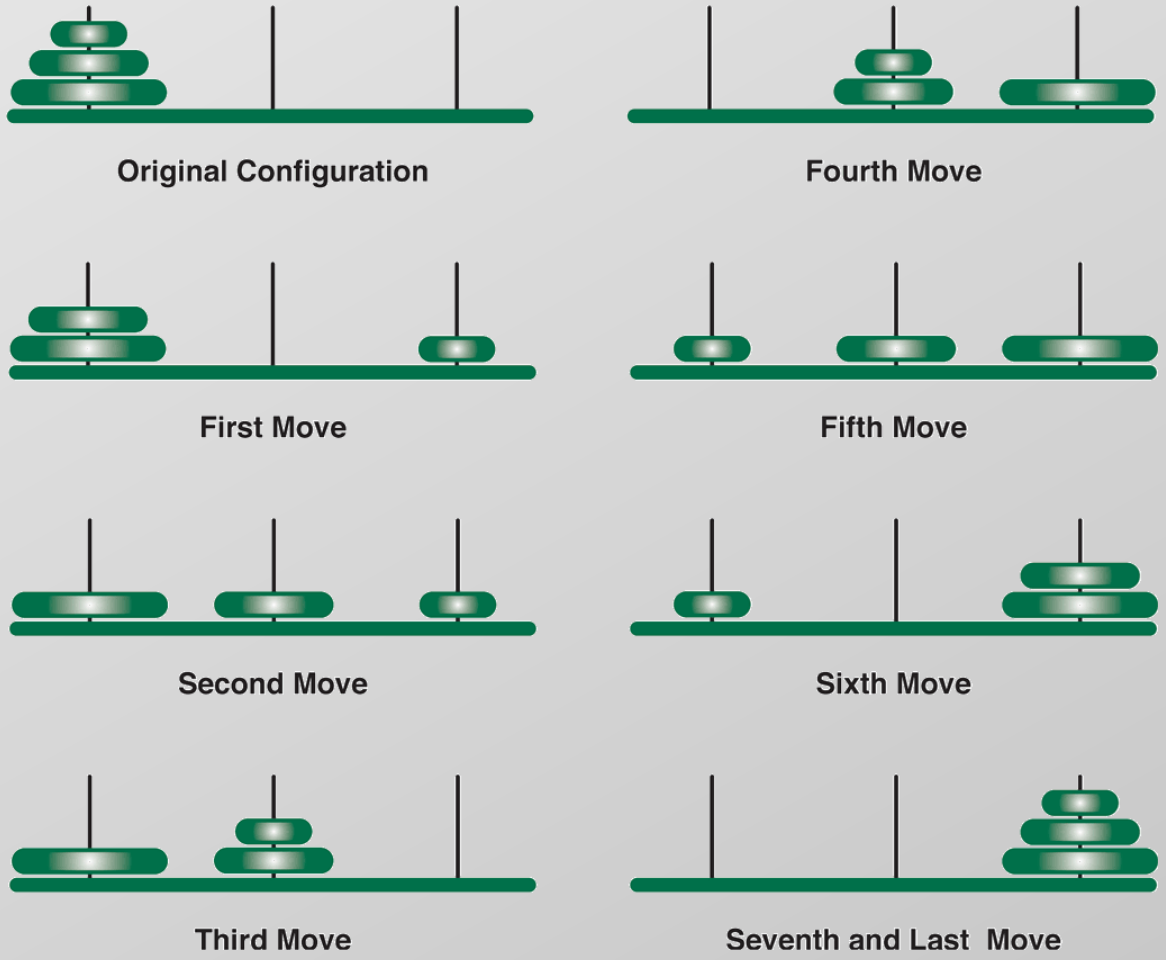
2. How to debug a program in Big Data?



Nobody writes perfect code!

How to debug a program in Big Data?

Example: A solution to the three-disk Towers of Hanoi puzzle



C/C++

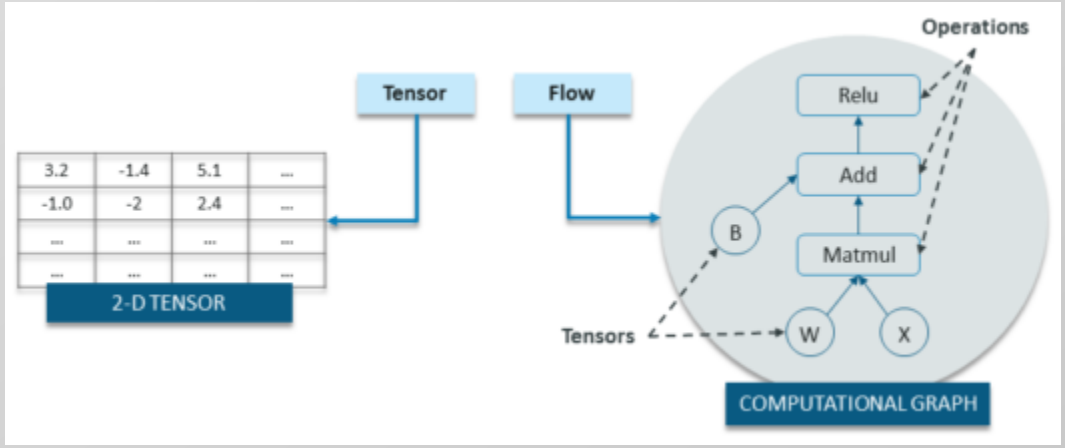
```
void main () {
      moveTower(4, 1,3,2);
}
void moveTower (int numDisks, int start, int end, int temp)    {
     if (numDisks == 1)
       moveOneDisk (start, end);
     else      {
       moveTower (numDisks-1, start, temp, end);
       moveOneDisk (start, end);
       moveTower (numDisks-1, temp, end, start);
     }
}
void moveOneDisk (int start, int end)    {
     printf ("Move one disk from " + start + " to " +   end);
  }
```

Prolog

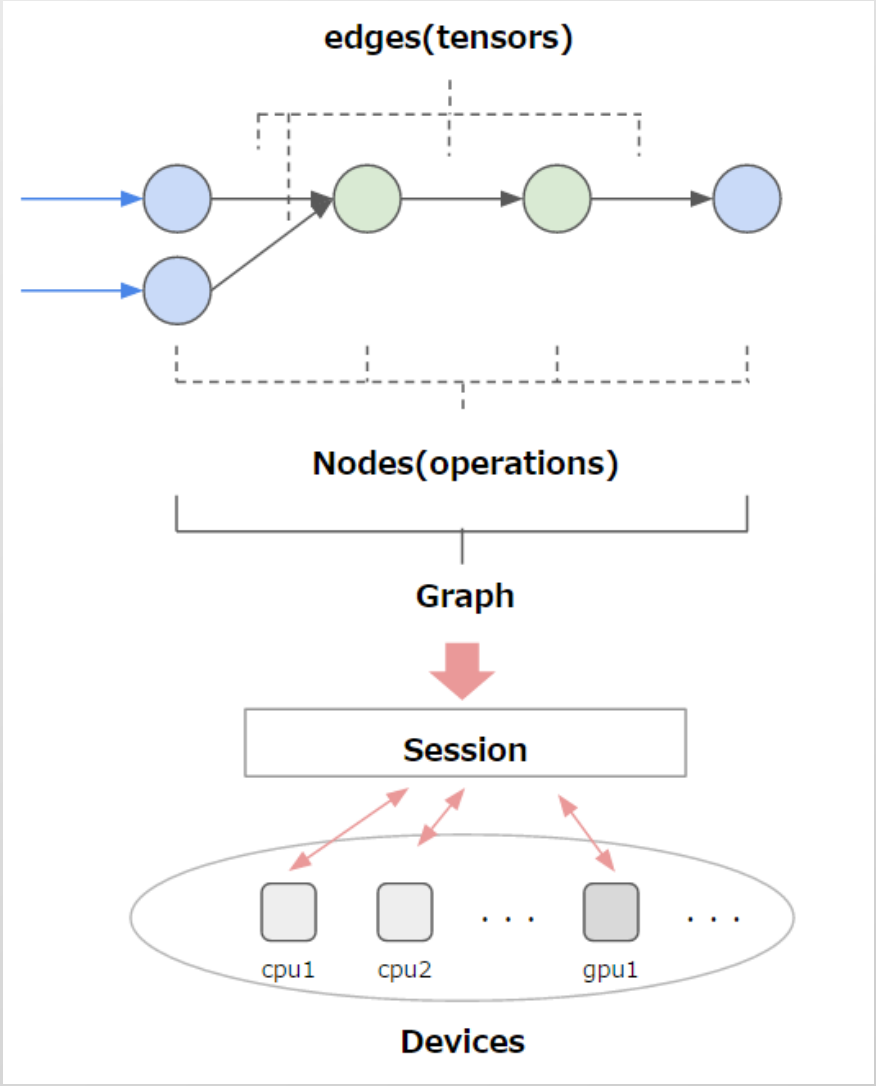
```
move(1,X,Y,_) :-
   write('Move top disk from '),
   write(X),
   write(' to '),
   write(Y),
   nl.
move(N,X,Y,Z) :-
   N>1,
   M is N-1,
   move(M,X,Z,Y),
   move(1,X,Y,_),
   move(M,Z,Y,X).


?-  move(3,left,right,center).
```

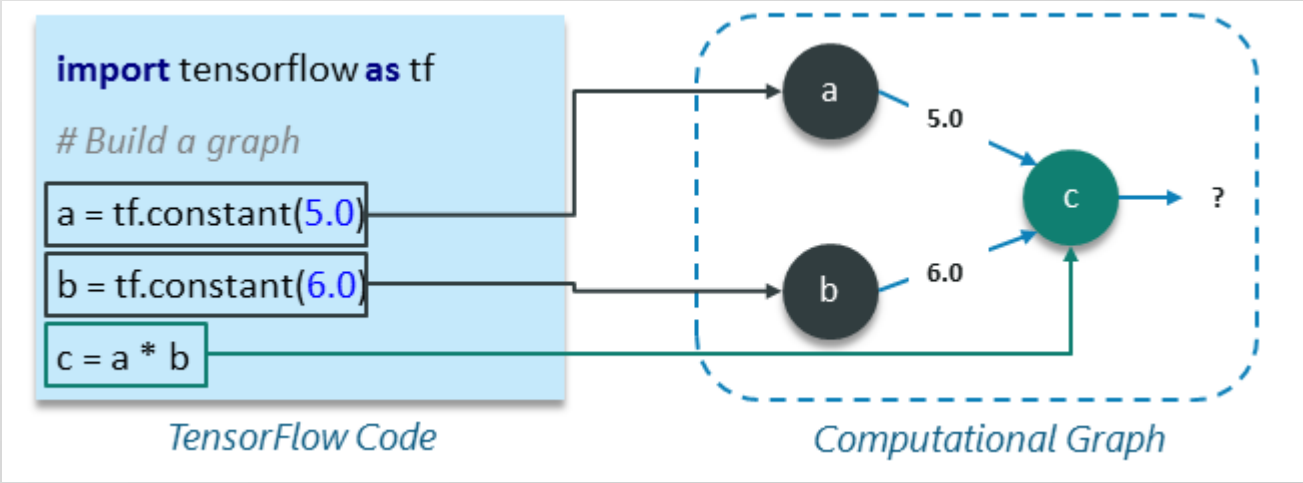
What's TensorFlow™?

- TensorFlow is a library for numerical computation where data flows through the graph.
- The term TensorFlow is made up of two terms – Tensor & Flow
- The term tensor refers to the representation of data as multi-dimensional array whereas the term flow refers to the series of operations that one performs on tensors.



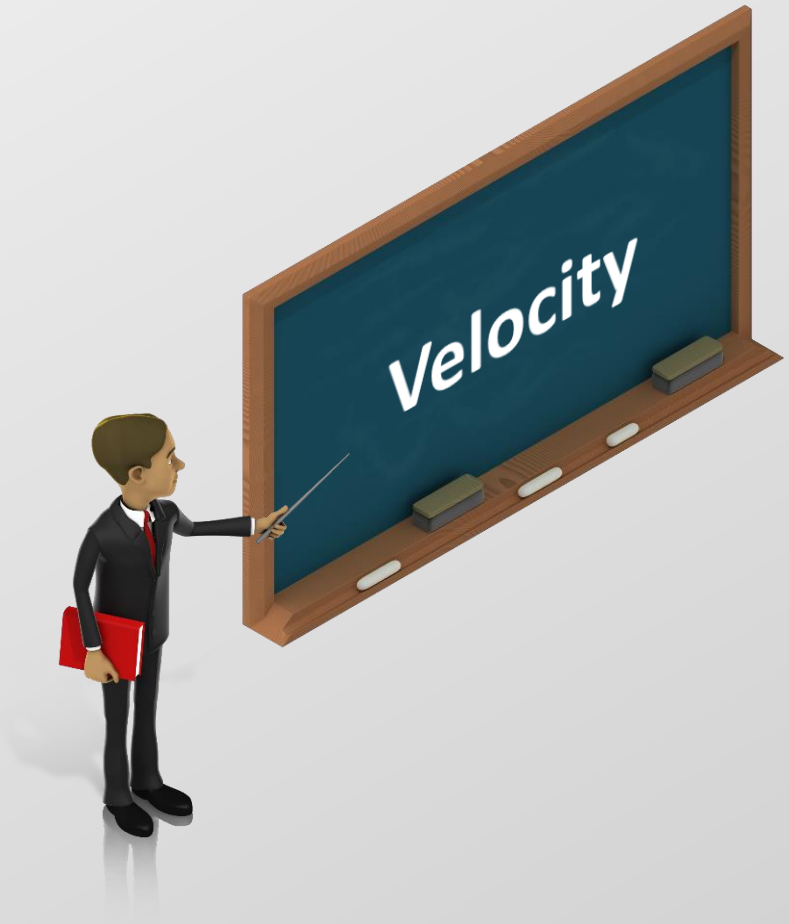
A TensorFlow program

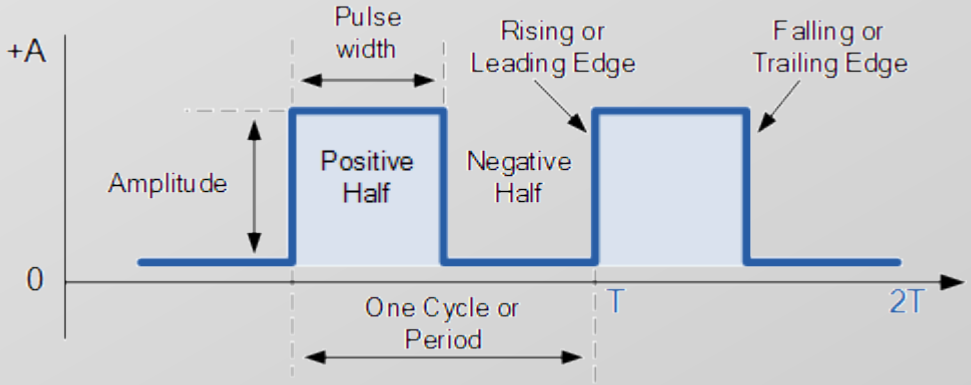


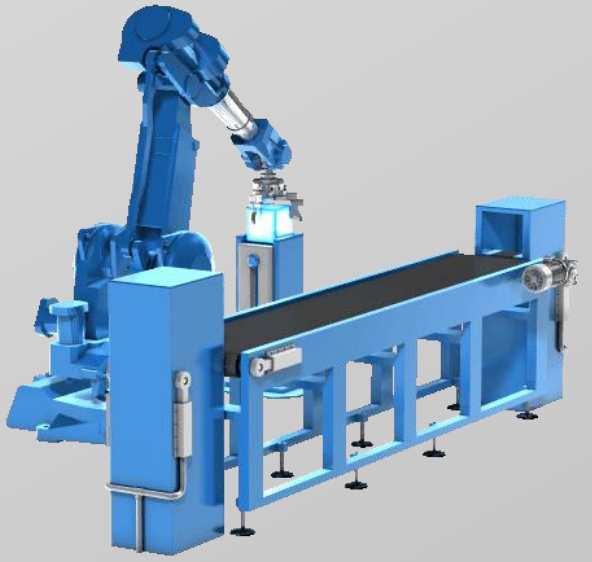



```
# Create the session object
sess = tf.Session()
 #Run the graph within a session and store the output to a variable
output_c = sess.run(c)
 #Print the output of node c
print(output_c)
 #Close the session to free up some resources
sess.close()
```

3. How to write a fast-running program in Big Data?



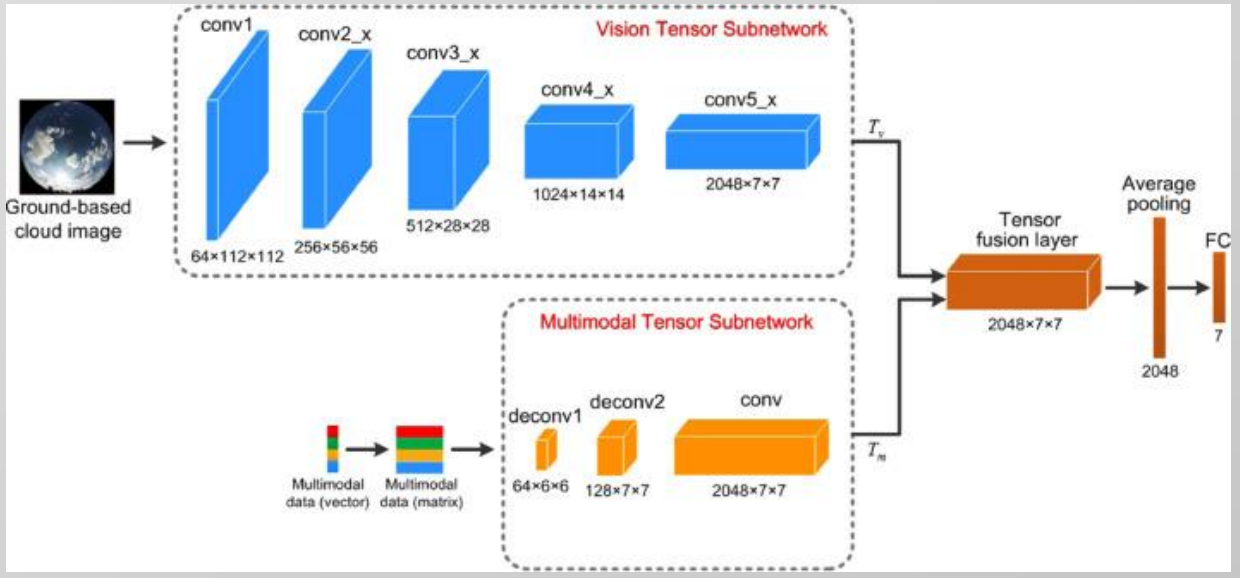




Pseudo Code

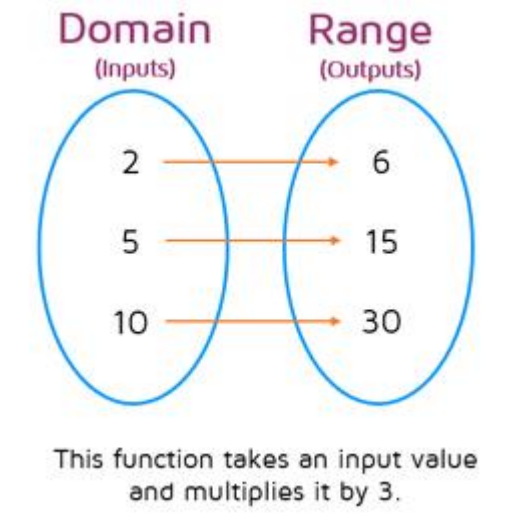
Algorithm 1

1. **For** each vortex filament *i* **do**
2. **For** each edge *j* of *i* **do**
3. Update the radius δj
4. Map the vorticity to grid by Peskin function Equation 5
5. **End for**
6. End for
7. Solve Poisson Equation 7
8. Add in the velocity change induced by external force
9. **For** each vortex filament *i* **do**
10. **For** each edge *j* of *i* **do**
11. **If** *j* gets overly stretched after advection **then**
12. Take middle point as new sampling point
13. **End if**
14. **End for**
15. **End for**
16. Advect the vortex filaments
17. Advect the smoke particles



Function

- A function assigns only output to each input. The value that is put into a function is the input. The result is the output.



Basic Operators _ Convolution

$$f(x,y) * w(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) w(x-m, y-n)$$

Or

$$g(x,y) = \sum_{m=-a}^{a} \sum_{n=-b}^{b} w(m,n) f(x+m, y+n)$$

Example:

$$g = f * \begin{bmatrix} 1/2 \\ 0 \\ -1/2 \end{bmatrix}$$

means that

$$g(x,y) = \frac{1}{2} f(x, y+1) - \frac{1}{2} f(x, y-1)$$

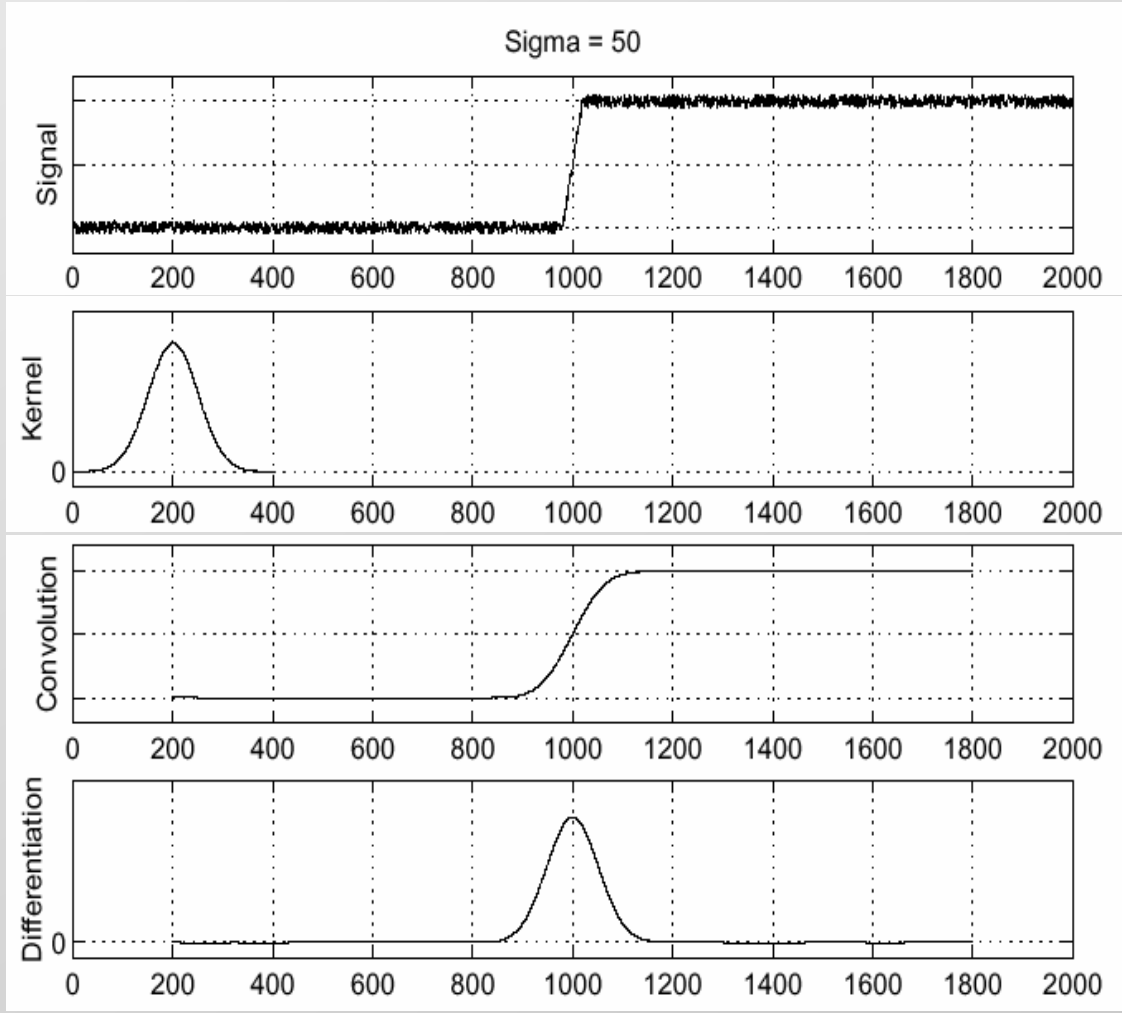
Associative property of convolution

f

h

$f * h$

$$\frac{d}{dx}(f * h)$$



Associative property of convolution

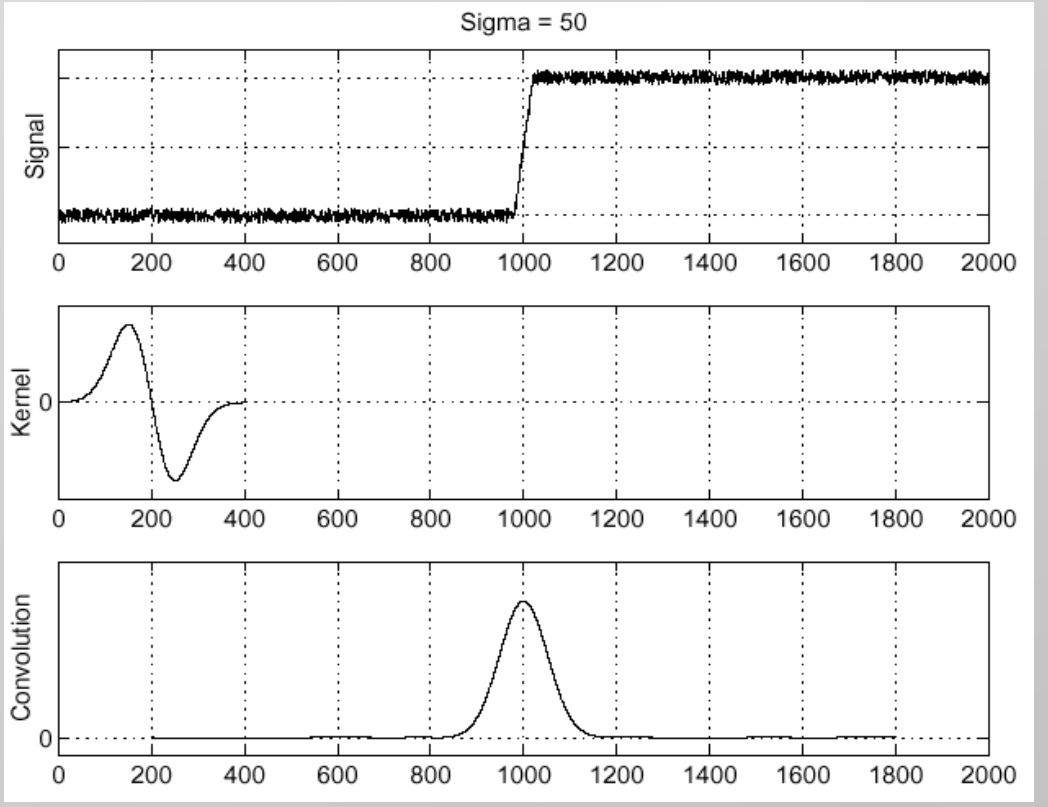
- Differentiation is convolution, and convolution is associative: $\frac{d}{dx}(f * h) = f * \frac{d}{dx}h$

- This saves us one operation:

f

$\frac{d}{dx}h$

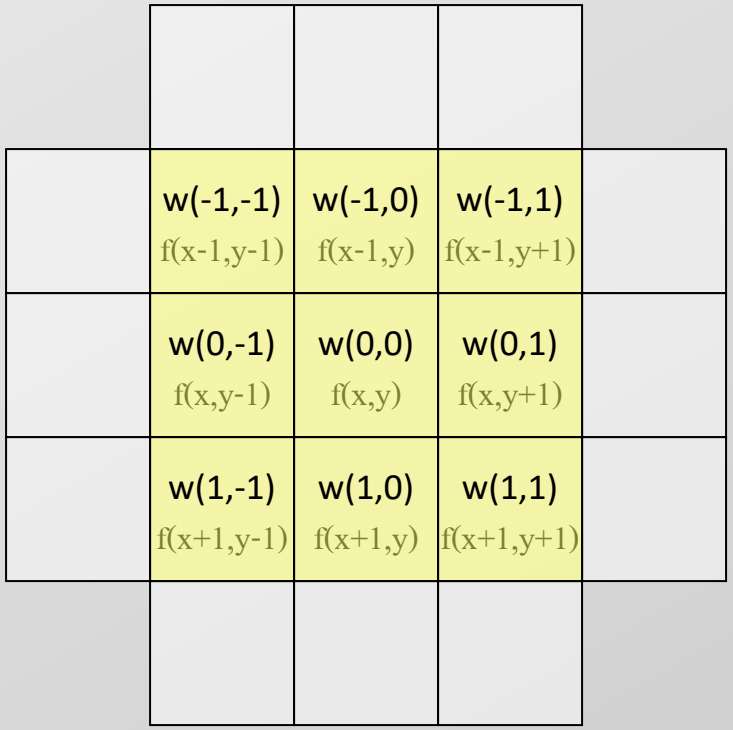
$f * \frac{d}{dx}h$



Linear spatial filtering

The result is the sum of products of the mask coefficients with the corresponding pixels directly under the mask

Pixels of image

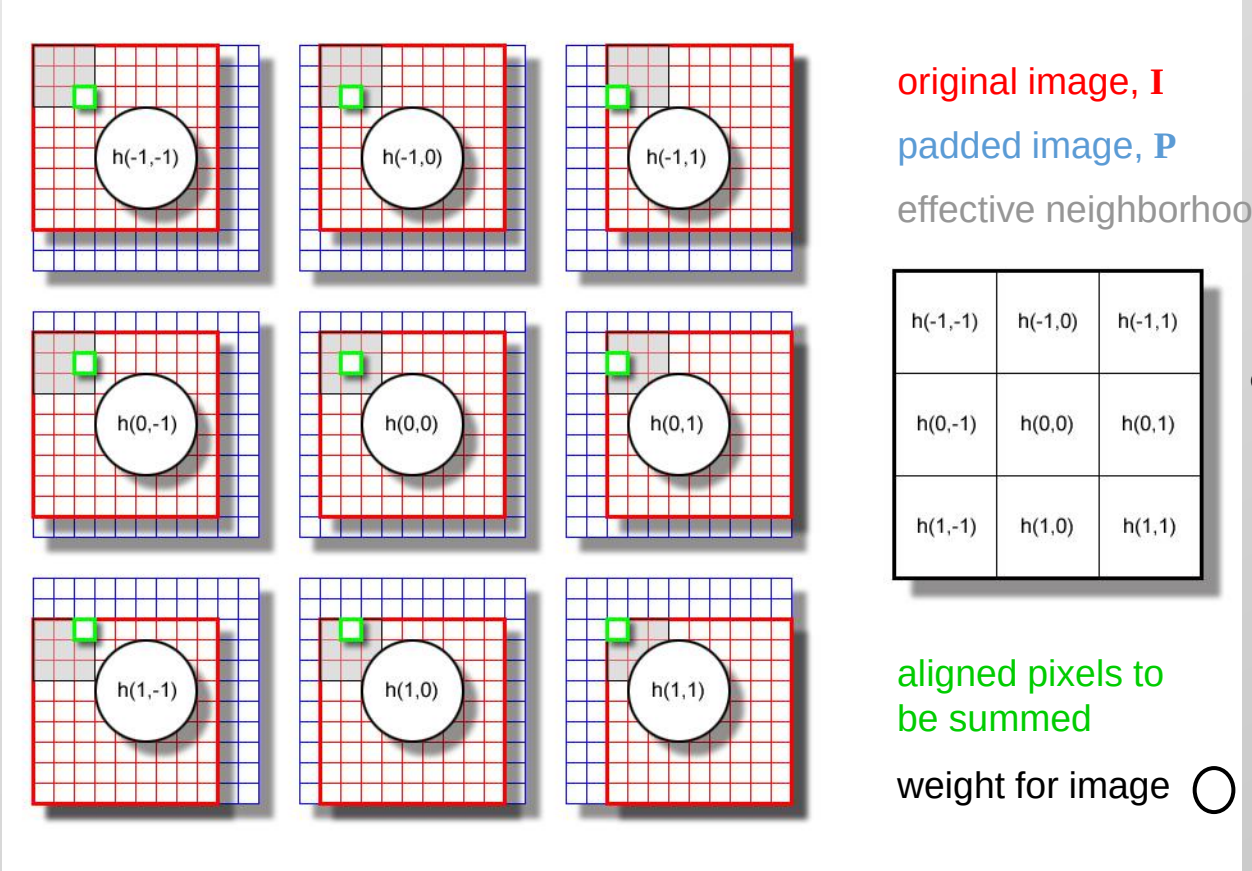


Mask coefficients

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

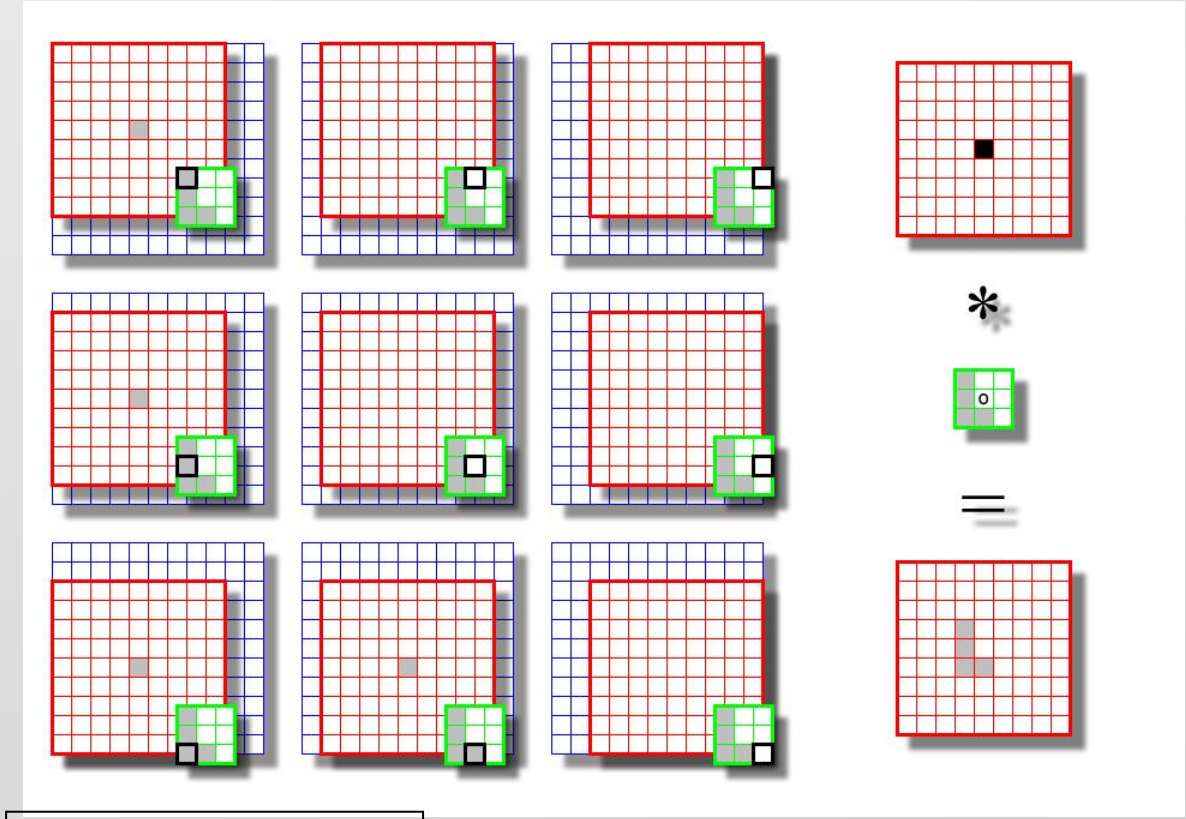
$$g(x,y) = w(-1,-1)\,f(x-1,y-1) + w(-1,0)\,f(x-1,y) + w(-1,1)\,f(x-1,y+1) +$$
$$w(0,-1)\,f(x,y-1) + w(0,0)\,f(x,y) + w(0,1)\,f(x,y+1) +$$
$$w(1,-1)\,f(x+1,y-1) + w(1,0)\,f(x+1,y) + w(1,1)\,f(x+1,y+1)$$

Convolution by Copying, Multiplying, and Shifting the Image



For each element $\mathbf{h}(r_h,c_h)$ in weight matrix, **h,** image **I** is copied into a zero-padded image, **P**, starting at (r_h,c_h).
Each **P** is multiplied by the corresponding weight, $\mathbf{h}(r_h,c_h)$.
All the **P** images are summed pixel-wise then divided by the sum of the elements of **h**.
The result is cropped out of the center of the accumulated **P**s.

Convolution by Copying, Multiplying, and Shifting the Image



The original image has a black impulse at the center and zeros (white) elsewhere.

The weight matrix has a gray 'L' at its left and zeros (white) elsewhere.

The resulting image has a copy of the weight matrix pegged to the impulse location.

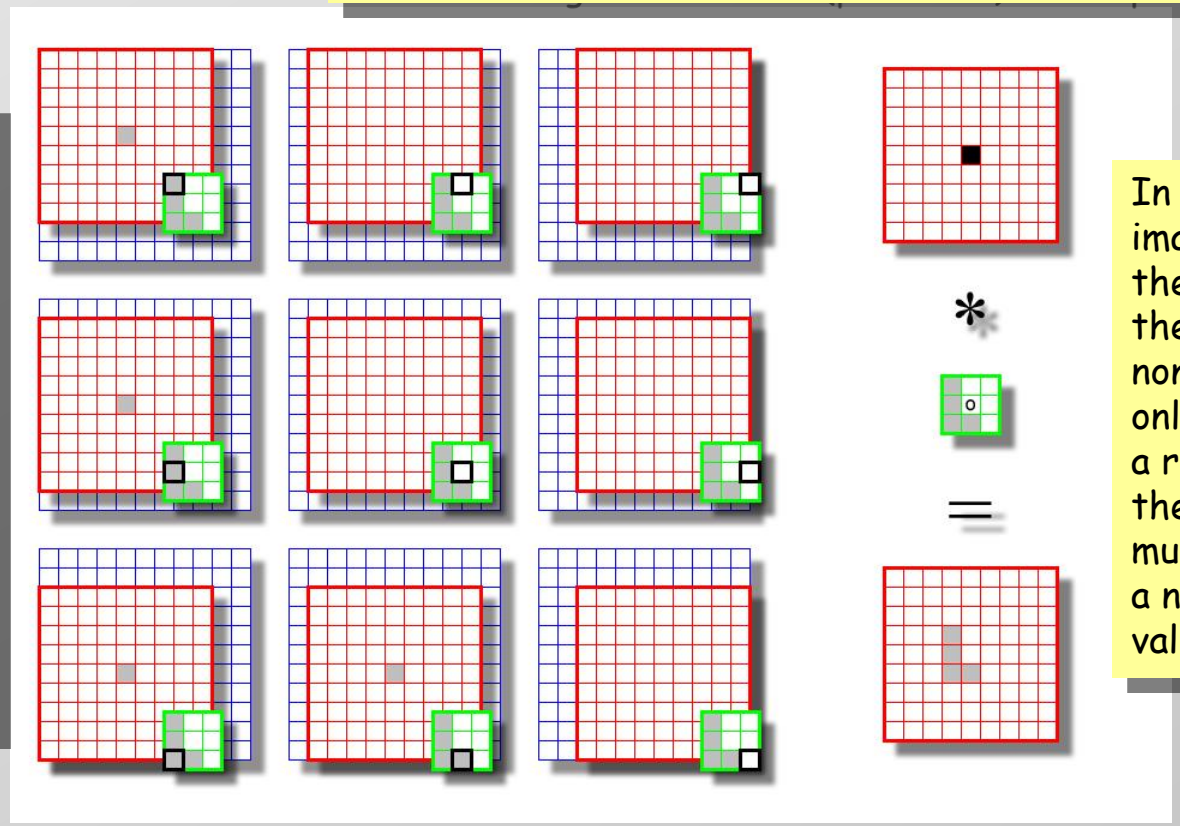
original image, **I**
effective neighborhood
padded image, **P**

In the result, the origin of the weight matrix coincides with the original location of the impulse.

Convolution by Copying, Multiplying, and Shifting the Image

Each copy of the (entire) image is multiplied by the value of the weight matrix in black square (here, white = 0) before being accumulated (pixelwise) in the padded image

The position of the black square relative the center of the weight matrix indicates the shift of the original image relative to the middle of the padded image.

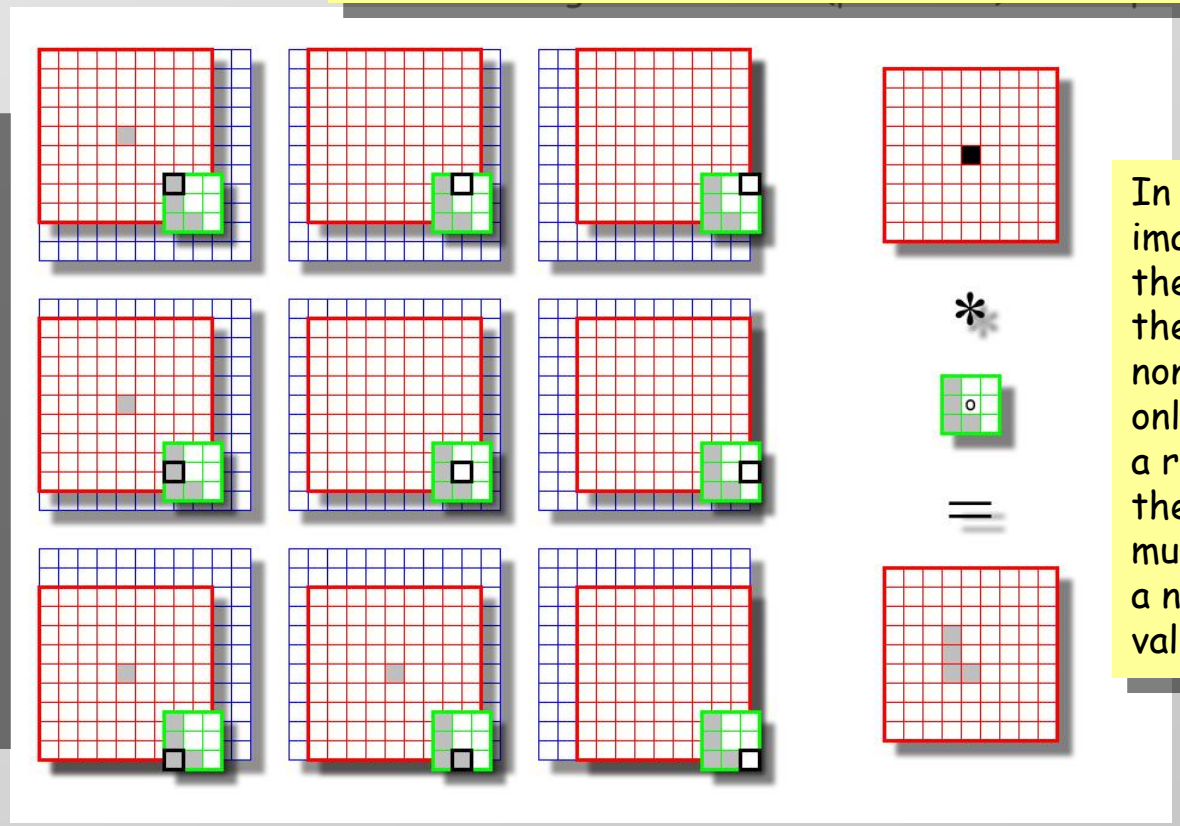


In this image, only the pixel in the center is nonzero so only it shows a result when the image is multiplied by a nonzero value

Convolution by Copying, Multiplying, and Shifting the Image

Each copy of the (entire) image is multiplied by the value of the weight matrix in black square (here, white = 0) before being accumulated (pixelwise) in the padded image

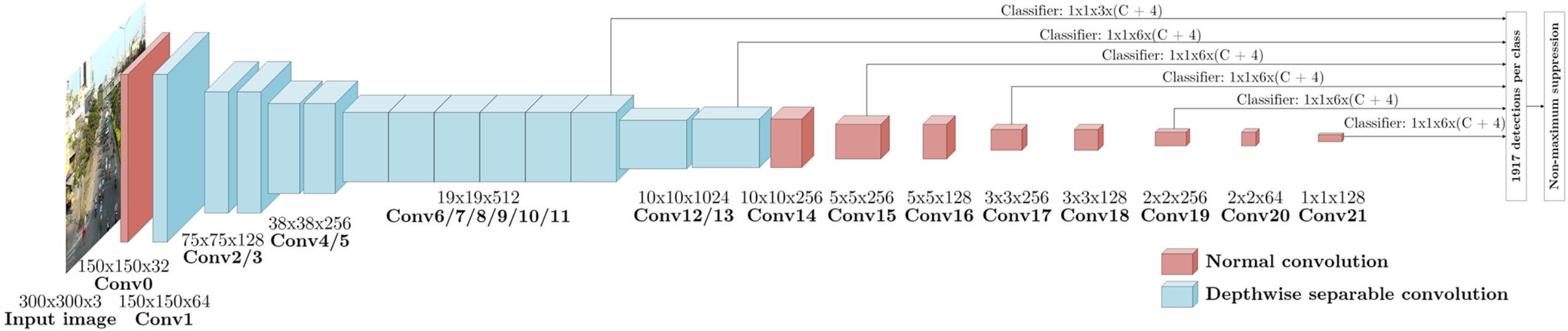
The position of the black square relative the center of the weight matrix indicates the shift of the original image relative to the middle of the padded image.

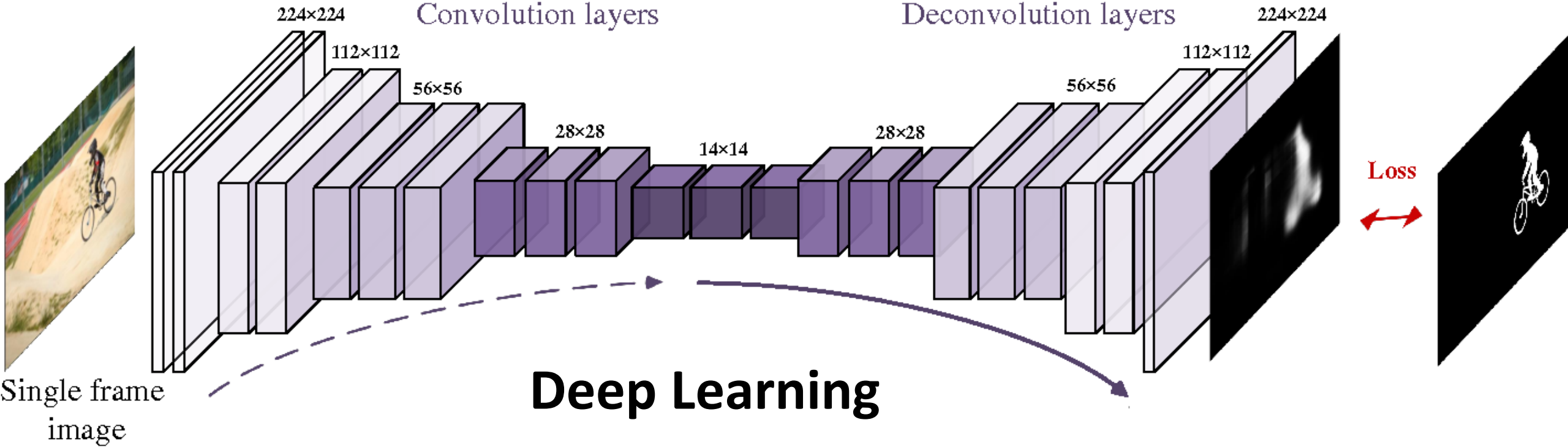


In this image, only the pixel in the center is nonzero so only it shows a result when the image is multiplied by a nonzero value

Machine Learning

Concept Learning and the General-to-Specific Ordering	Decision Tree Learning	Artificial Neural Networks	Evaluating Hypotheses
Bayesian Learning	Computational Learning Theory	Instance-Based Learning	Genetic Algorithms
Learning Sets of Rules	Analytical Learning	Combining Inductive and Analytical Learning	Reinforcement Learning





Deep Learning

Richardson's Extrapolation

$$I = I(h) + E(h)$$

$$h = (b-a)/n$$

$$I(h_1) + E(h_1) = I(h_2) + E(h_2)$$

$$n = (b-a)/h$$

$$E \cong \frac{b-a}{12} h^2 \bar{f}''$$

$$\frac{E(h_1)}{E(h_2)} \cong \frac{h_1^2}{h_2^2}$$

Assumed constant regardless of step size

$$E(h_1) \cong E(h_2)\left(\frac{h_1}{h_2}\right)^2$$

$$I(h_1) + E(h_2)\left(\frac{h_1}{h_2}\right)^2 \cong I(h_2) + E(h_2)$$

$$E(h_2) \cong \frac{I(h_1) - I(h_2)}{1 - \left(h_1 / h_2\right)^2}$$

$$I = I(h_2) + E(h_2)$$

$$I \cong I(h_2) + \frac{1}{\left(h_1 / h_2\right)^2 - 1}\left[I(h_2) - I(h_1)\right]$$

I =exact value of integral

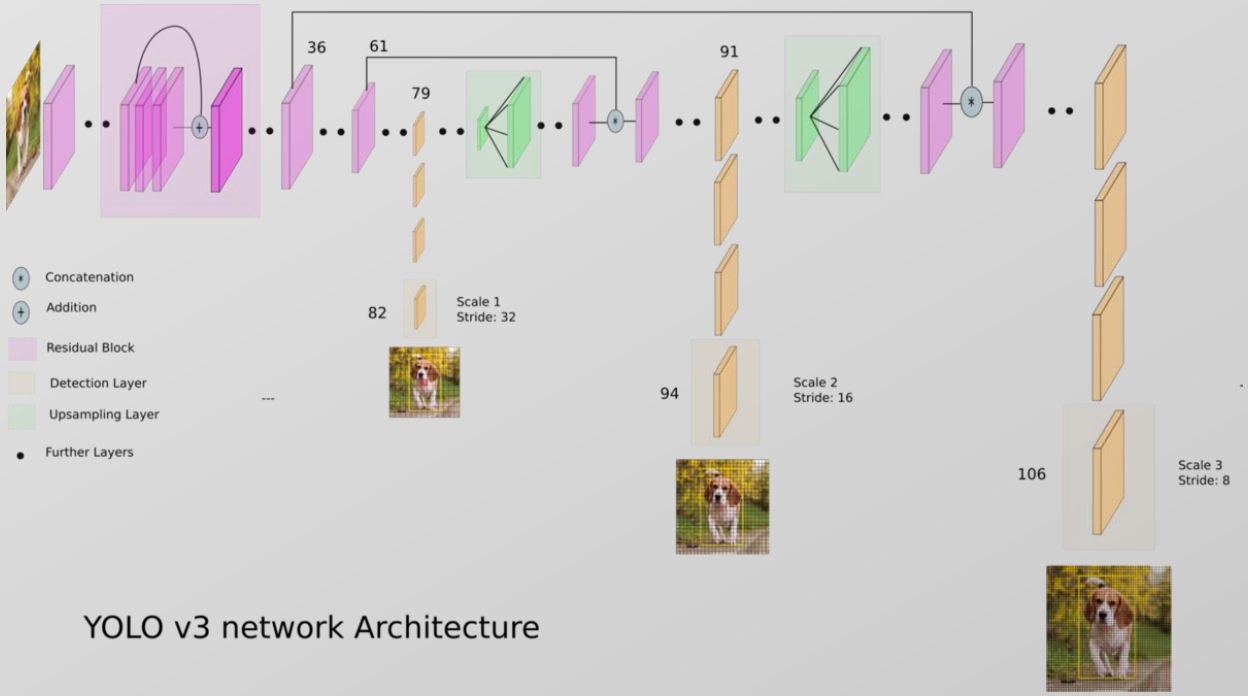
$I(h)$ =the approximation from an n segment application of trapezoidal rule with step size h

$E(h)$ =the truncation error

YOLO is magic?

In theory,

- CNN is a statistical filter.
- Training data must be complete.

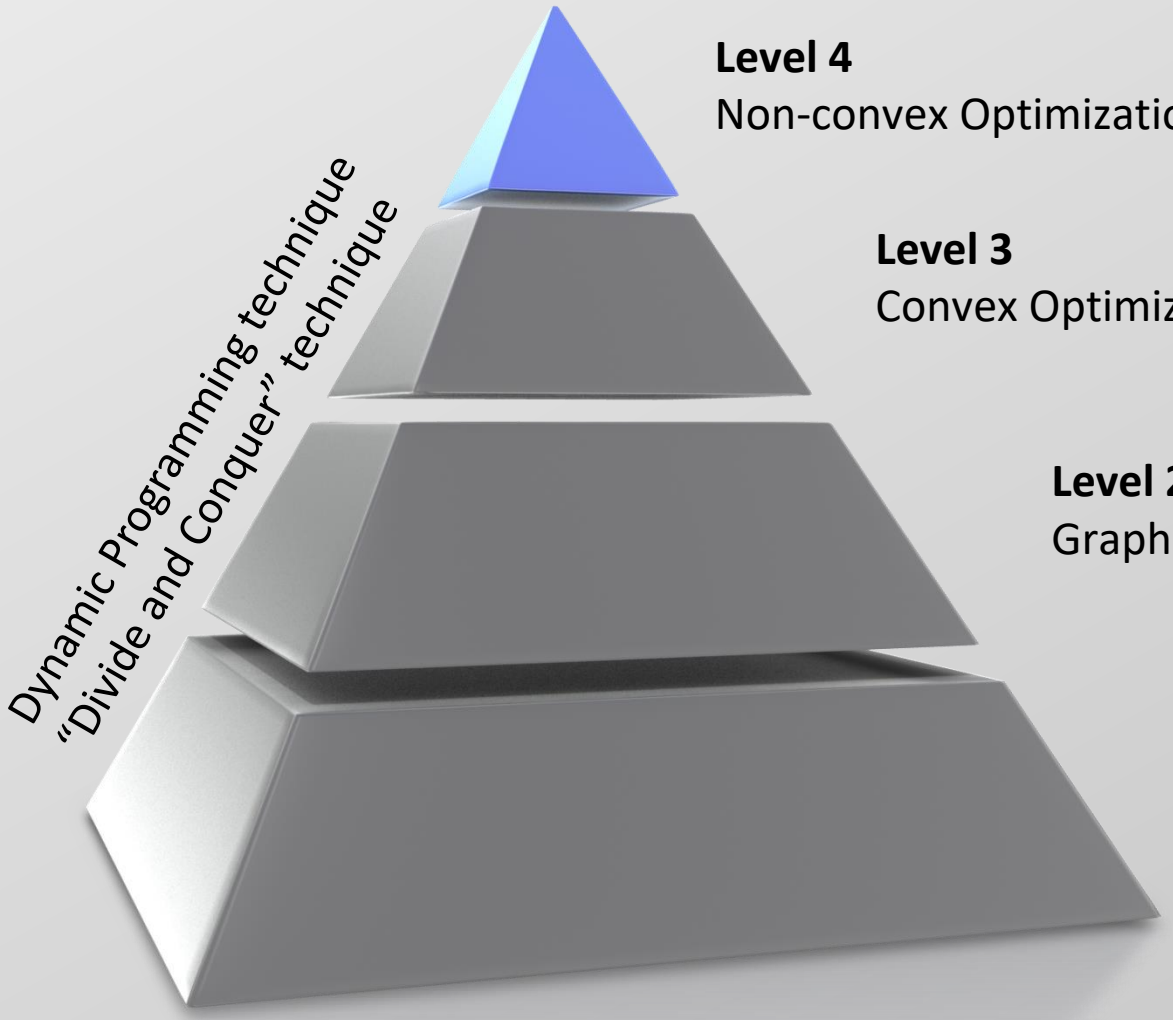


YOLO v3 network Architecture

But

- The environment is always changing.
- Actual training data is incomplete.

Summary

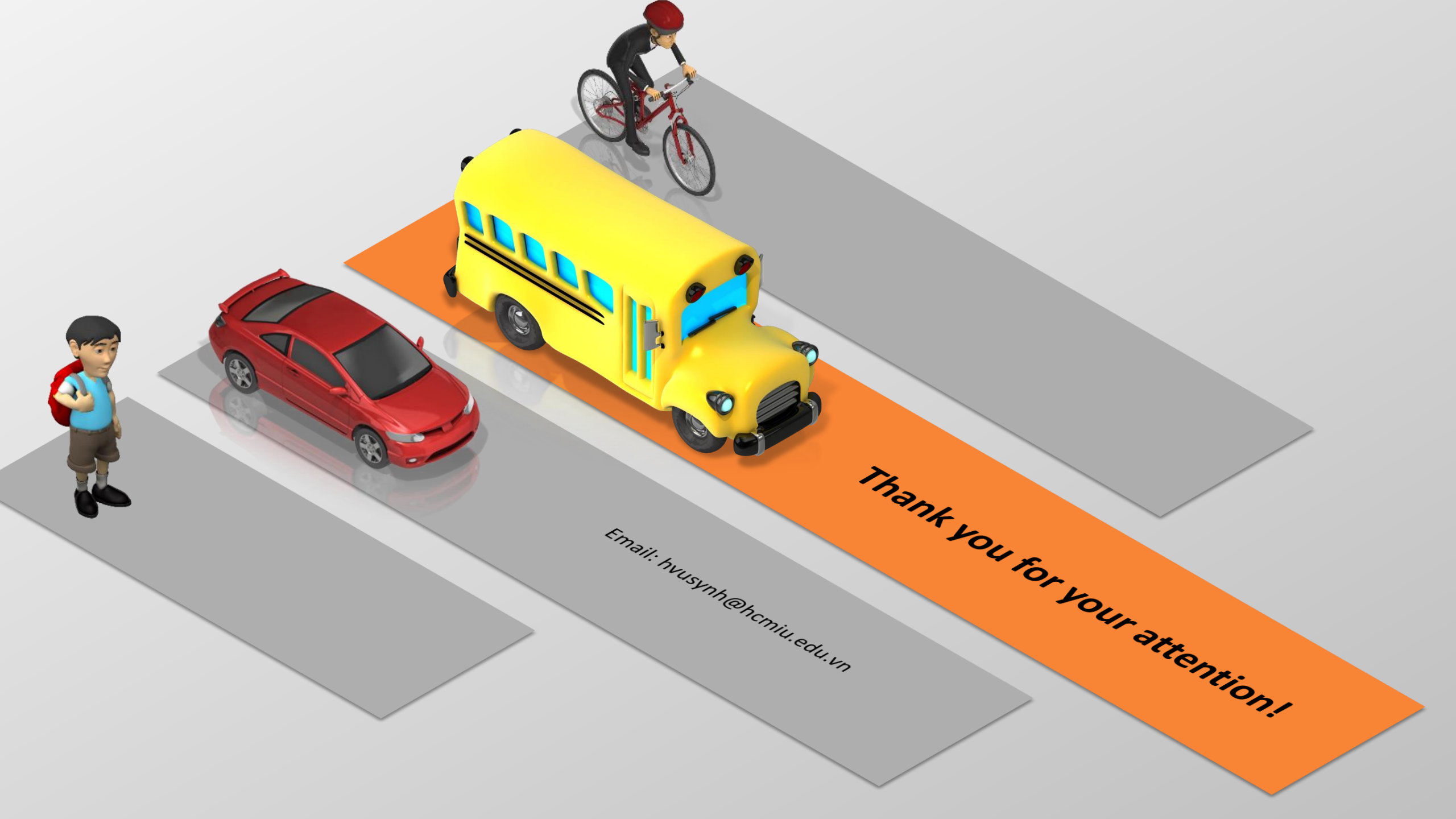


Level 4
Non-convex Optimization

Level 3
Convex Optimization

Level 2
Graphs

Level 1
"Exhausted check" method


Thank you for your attention!
Email: hvusynh@hcmiu.edu.vn