

MỘT SỐ HƯỚNG DẪN CHO PROJECT

NỘI DUNG

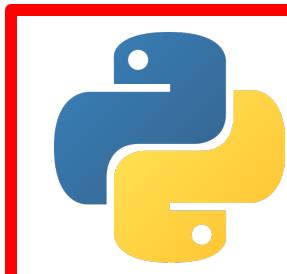
- Cần chuẩn bị những gì
 - Ngôn ngữ lập trình/ môi trường
 - Frameworks
 - Server huấn luyện
 - Phần cứng chạy
- Quy trình thực hiện một đề tài
- Một số kỹ năng quan trọng
 - Kỹ năng tìm kiếm trên google
 - Debug
 - Một số lỗi thường gặp

NỘI DUNG

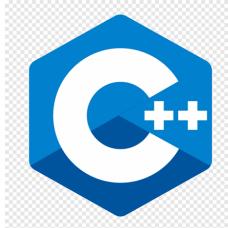
- Cần chuẩn bị những gì
 - **Ngôn ngữ lập trình/ môi trường**
 - Framework
 - Server huấn luyện
 - Phần cứng chạy suy luận
- Quy trình thực hiện một đề tài
- Một số kỹ năng quan trọng
 - Kỹ năng tìm kiếm trên google
 - Debug
 - Một số lỗi thường gặp

Chuẩn bị

Ngôn ngữ lập trình – môi trường



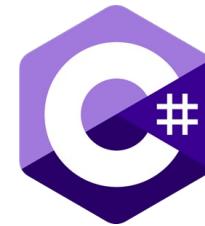
Python



C++



R



C#



Java



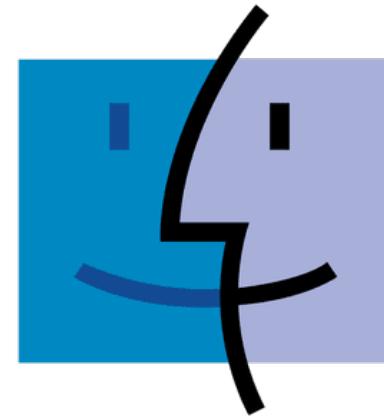
Javascript



Julia

Chuẩn bị

Ngôn ngữ lập trình – môi trường



MacTM OS

Chuẩn bị

Ngôn ngữ lập trình – môi trường - Window

- Một số trình IDE:

Pycharm



Jupyter notebook



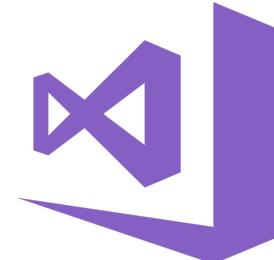
Sublime Text



Spyder



Visual studio



Jupyter notebook:

- Live code (tương tác trực tiếp từng dòng code)
- Mô phỏng
- Chạy trên nền web (IPython)

The screenshot shows a Jupyter Notebook interface with the title "jupyter tutorial" and a status bar indicating "Last Checkpoint: 3 minutes ago (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar below the menu has icons for file operations like Open, Save, and Run, along with a Python logo icon and a Logout button.

The main content area displays a section titled "PyCon 2018: Using pandas for Better (and Worse) Data Science". Below it is a GitHub link: <https://github.com/justmarkham/pycon-2018-tutorial>.

A red box highlights the first code cell (In [1]):

```
In [1]: import matplotlib.pyplot as plt  
import pandas as pd  
pd.__version__
```

The output (Out[1]) is: '0.24.1'

Below this, another red box highlights the second code cell (In [2]):

```
In [2]: # ri stands for Rhode Island  
ri = pd.read_csv('police.csv')
```

The third code cell (In [3]) is shown below:

```
In [3]: # what does each row represent?  
ri.head()
```

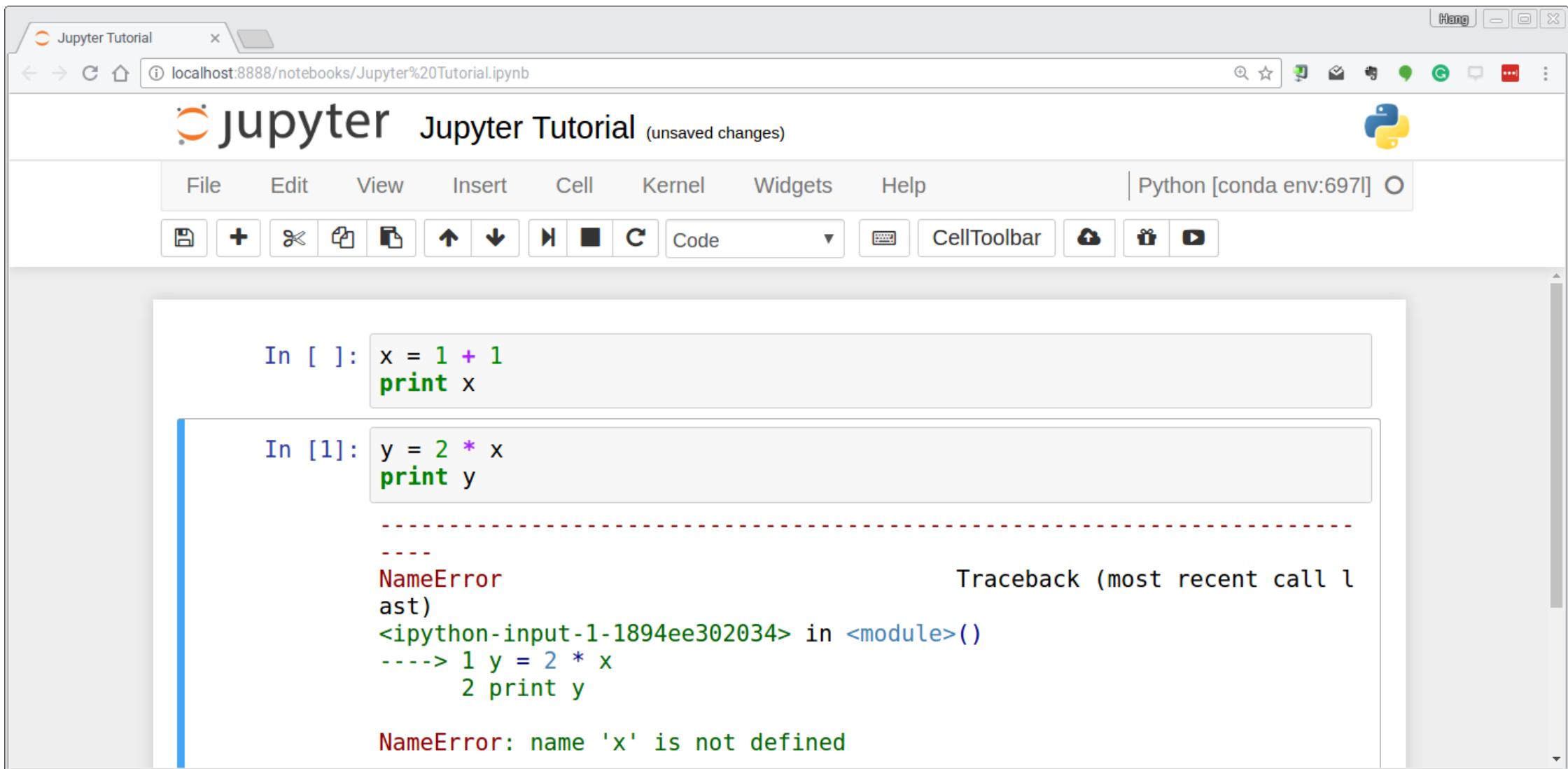
The output (Out[3]) is a table preview:

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_
0	2005-01-02	01:55	NaN	M	1985.0	20.0	White	Speeding	Speeding	
1	2005-01-18	08:15	NaN	M	1965.0	40.0	White	Speeding	Speeding	
2	2005-01-23	23:15	NaN	M	1972.0	33.0	White	Speeding	Speeding	
3	2005-02-20	17:15	NaN	M	1986.0	19.0	White	Call for Service	Other	
	2005-03									

<https://codelearn.io/sharing/jupyter-notebook-tutorial>

Chuẩn bị

Ngôn ngữ lập trình – môi trường - Window



The screenshot shows a Jupyter Notebook interface running on a Windows system. The title bar reads "Jupyter Tutorial". The URL in the address bar is "localhost:8888/notebooks/Jupyter%20Tutorial.ipynb". The main area displays two code cells:

```
In [ ]: x = 1 + 1
         print x
```

```
In [1]: y = 2 * x
         print y
```

Below the second cell, an error message is shown:

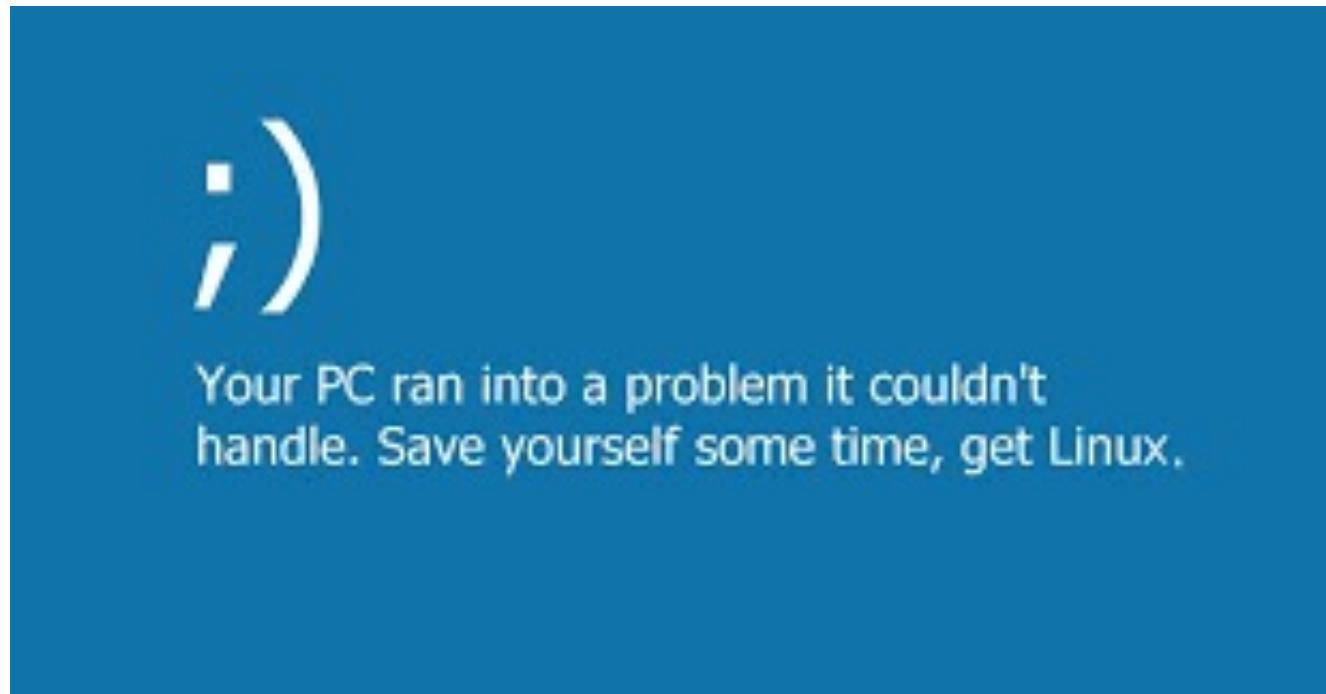
```
-----
NameError: name 'x' is not defined
```

The notebook interface includes a toolbar with various icons for file operations, cell selection, and help. The status bar at the bottom indicates "Python [conda env:697]".

Chuẩn bị

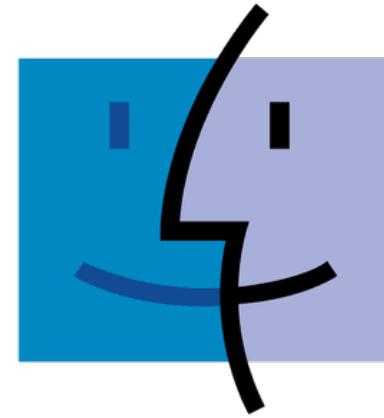
Ngôn ngữ lập trình – môi trường - Window

- Màn hình xanh
- Nhiều virus/ phần mềm độc hại tồn tại
- Một số phần mềm không hỗ trợ, rất khó để cài đặt



Chuẩn bị

Ngôn ngữ lập trình – môi trường



MacTM OS

Chuẩn bị

Ngôn ngữ lập trình – môi trường - Ubuntu

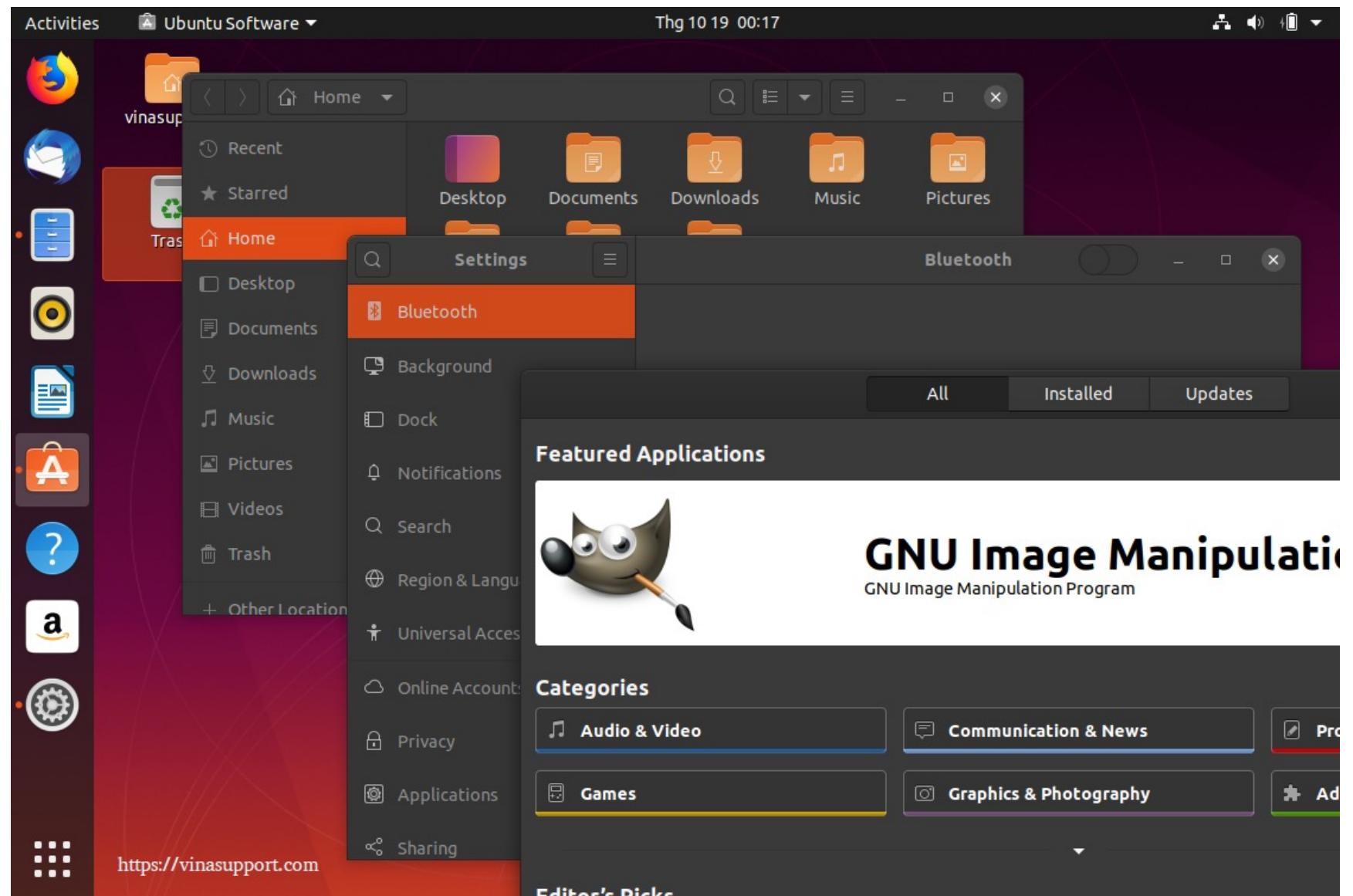
Tại sao phải sử dụng Linux- Ubuntu?

1. Hỗ trợ cộng đồng rộng lớn
2. Bỏ qua những quy trình phức tạp
3. Terminal dễ sử dụng hơn cmd
4. Không bị làm chậm RAM như Windows
5. Không bị lỗi đường dẫn PATH như Windows
6. Hỗ trợ sẵn hai phiên bản python 2 và python 3 độc lập
7. Phần cứng nhúng như : Jetson nano, Jetson tx2, Raspberry PI,... đều sử dụng hđh Linux

Chuẩn bị

Ngôn ngữ lập trình – môi trường - Ubuntu

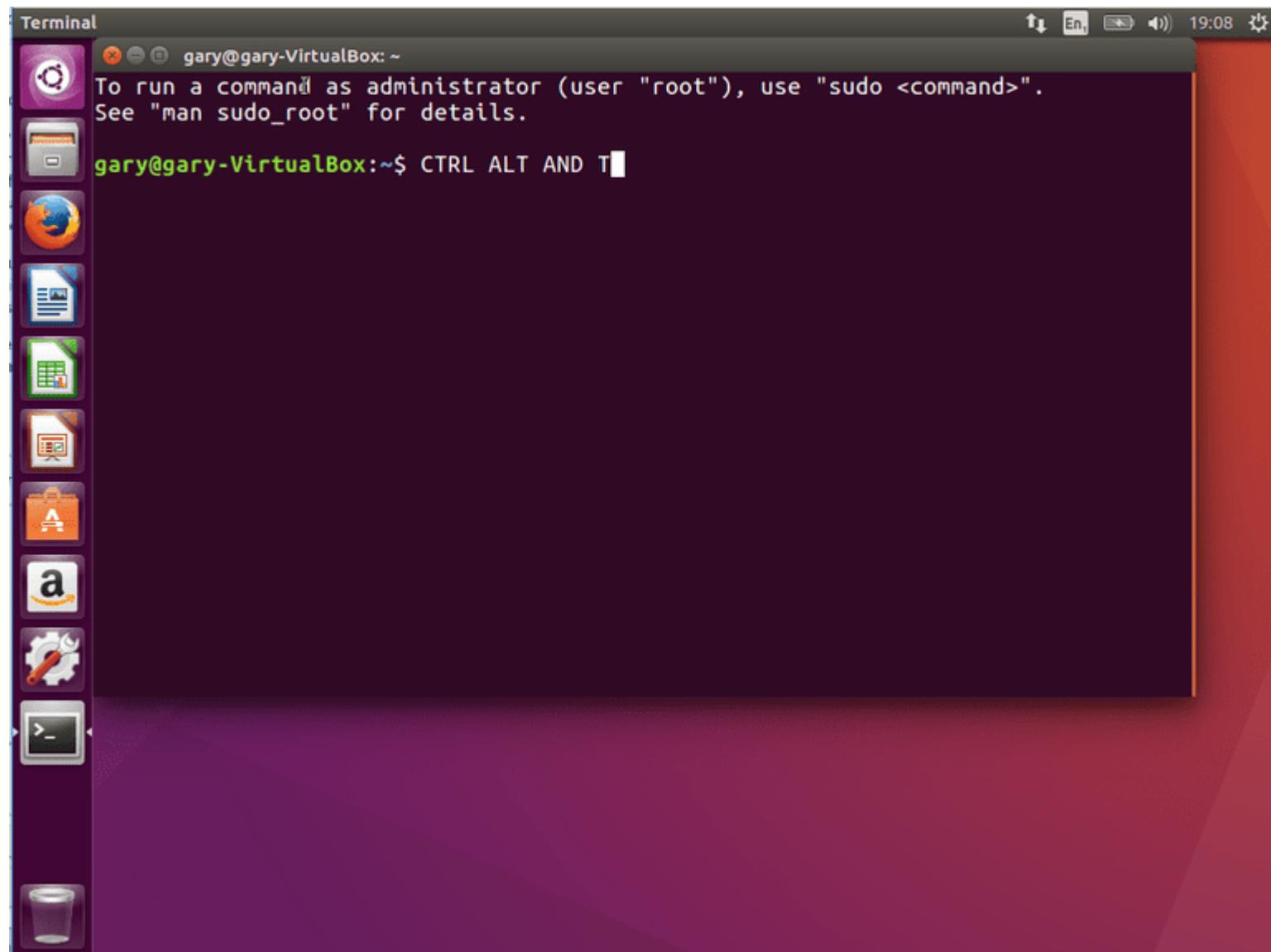
Giao diện



Chuẩn bị

Ngôn ngữ lập trình – môi trường - Ubuntu

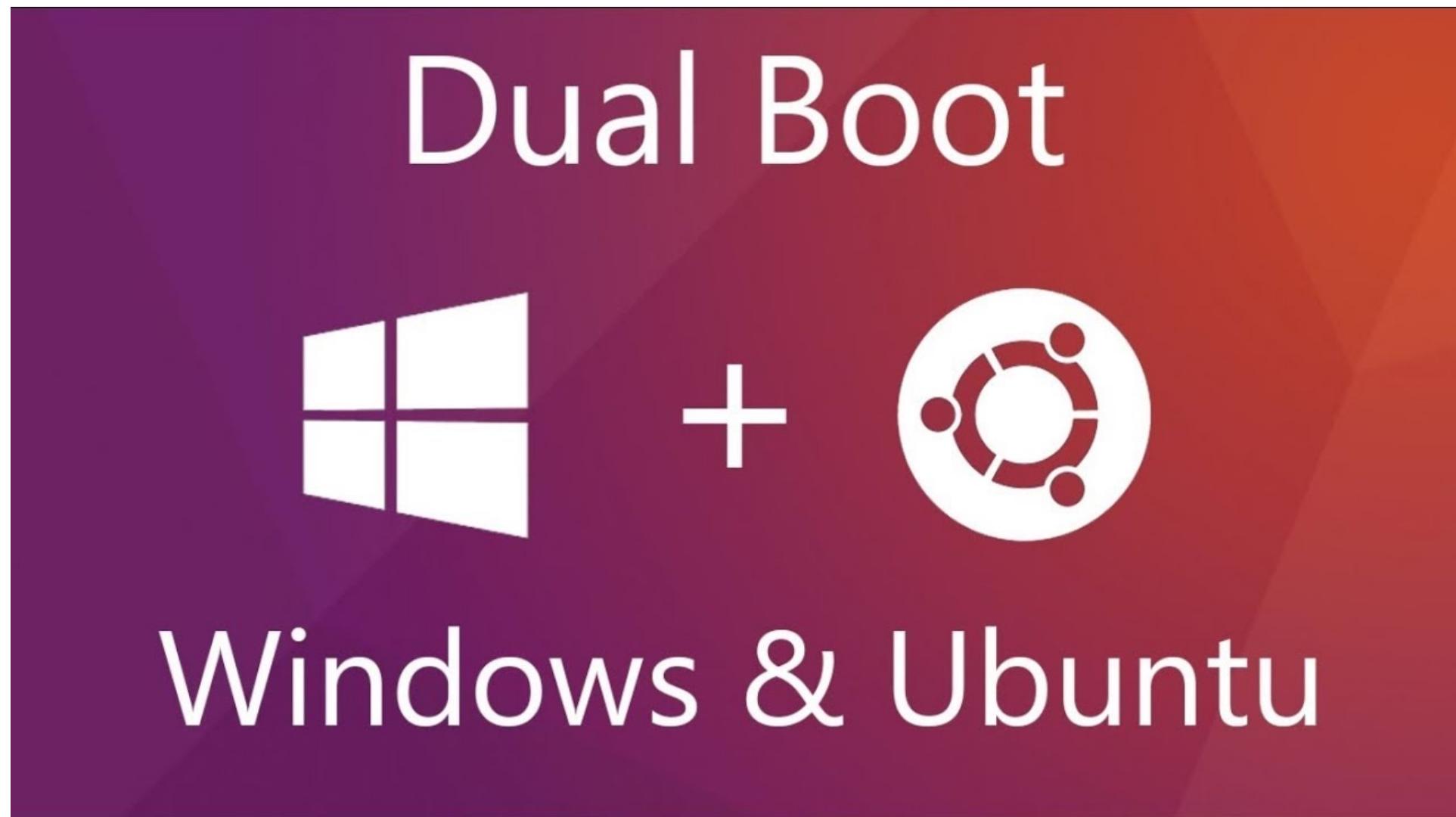
Terminal



Chuẩn bị

Ngôn ngữ lập trình – môi trường - Ubuntu

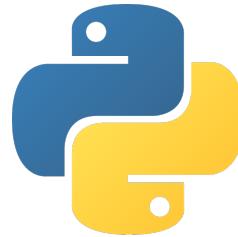
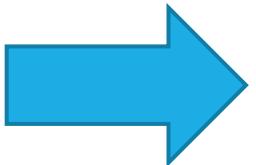
Cài đặt:



Chuẩn bị

Ngôn ngữ lập trình – môi trường

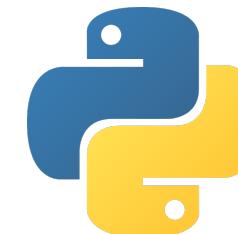
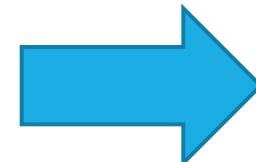
Project 1
Author 1



TensorFlow

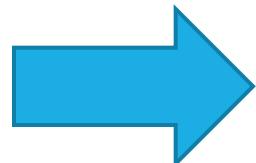
TF version 1.0.0

Project 2
Author 2



PyTorch

Pytorch version 1.6.0



CÓ NÊN XÓA VÀ CÀI ĐẶT LAI THƯ VIỆN MỚI CHO PHÙ HỢP ?

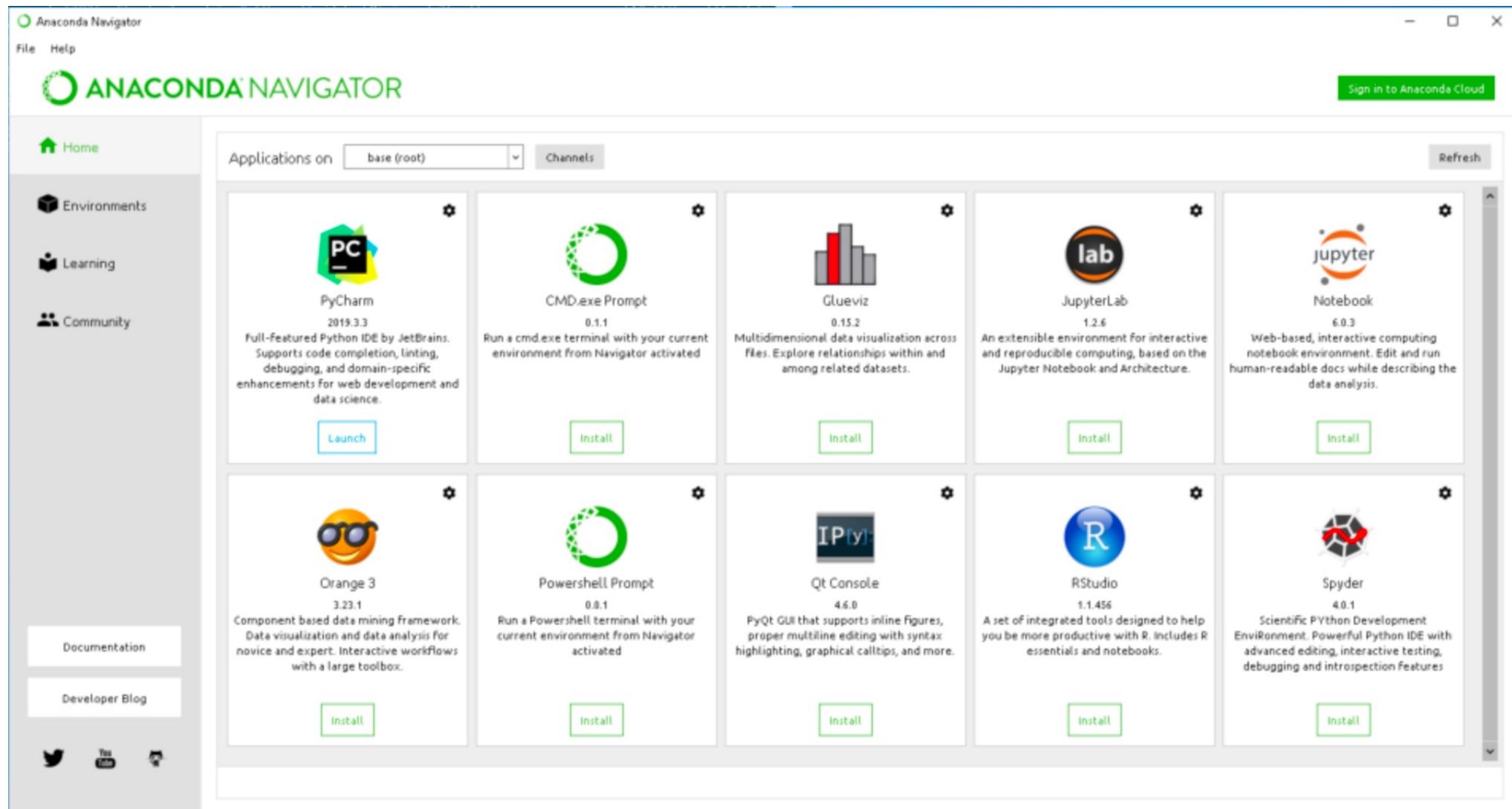
ANACONDA?



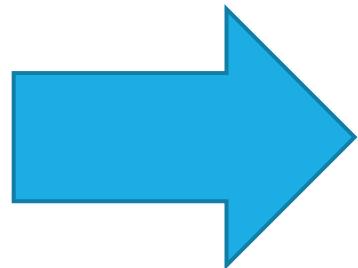
- Hỗ trợ nhiều IDE
- Có hơn 100 gói python (thư viện)
- Ngôn ngữ python là cốt lõi
- Tạo được nhiều môi trường chạy song song, dễ quản lý
- Dễ dàng cài đặt thư viện, tránh những rủi ro hỏng các gói khi cài.

Chuẩn bị

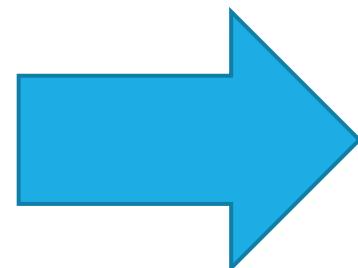
Ngôn ngữ lập trình – môi trường



Kết luận



Nên sử dụng Python vì hỗ trợ cộng đồng rộng lớn



Tùy vào dự án ngắn hạn hay dài hạn mà quyết định chọn môi trường

NỘI DUNG

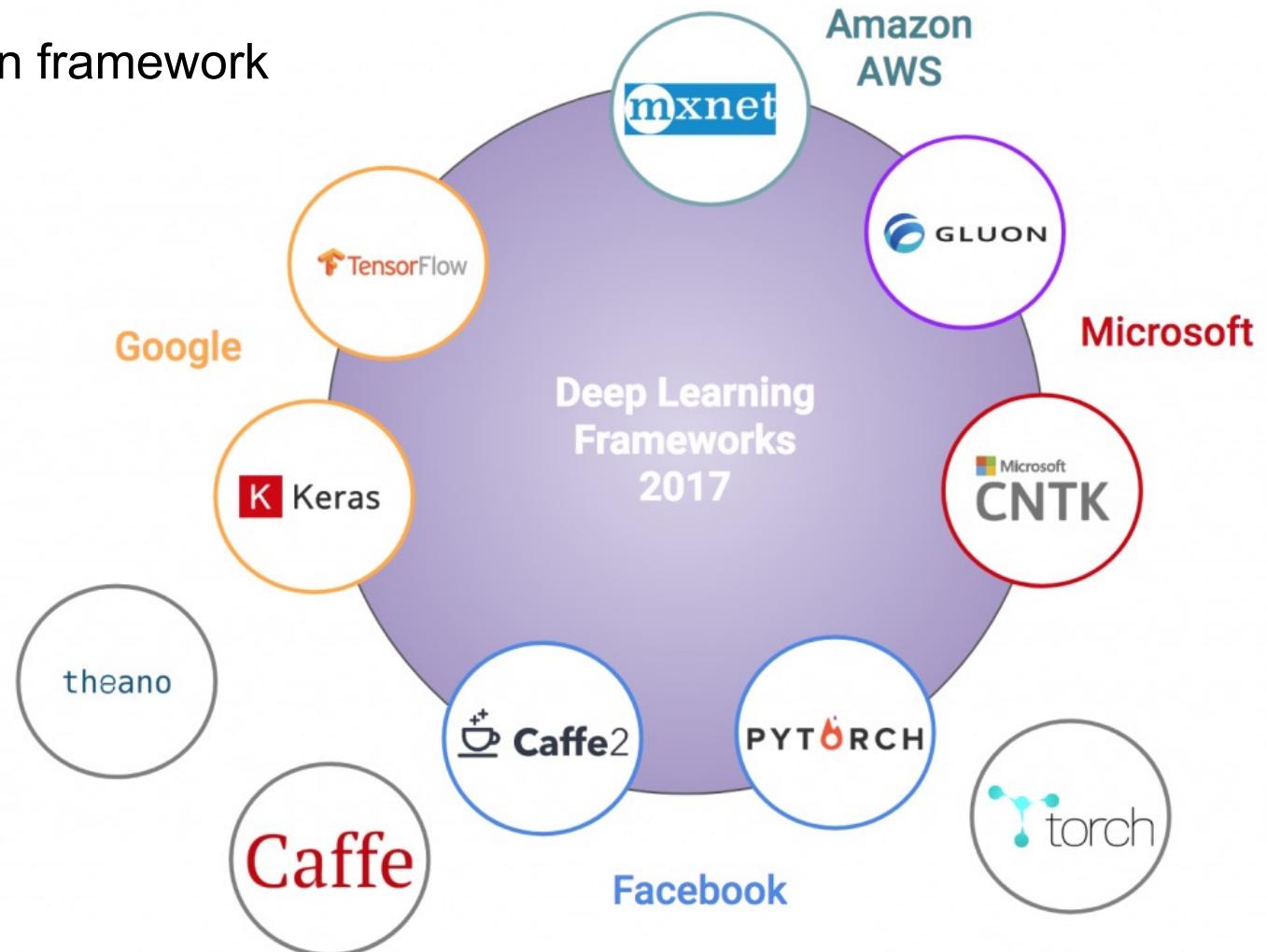
- Cần chuẩn bị những gì
 - Ngôn ngữ lập trình/ môi trường
 - **Framework**
 - Server huấn luyện
 - Phần cứng chạy suy luận
- Quy trình thực hiện một đề tài
- Một số kỹ năng quan trọng
 - Kỹ năng tìm kiếm trên google
 - Debug
 - Một số lỗi thường gặp

Chuẩn bị Framework

Có rất nhiều framework, làm thế nào để chọn framework phù hợp?

Theo một số tiêu chí :

- Ngôn ngữ của thuật toán (C++, Python,...)
- Phần cứng chạy suy luận là phần cứng nào (CPU, GPU, mobile device,...)
- Mục đích (nghiên cứu; học tập; làm cho dự án nâng cao)



Một số hàm cần ghi nhớ khi tiếp cận framework mới:

- Hàm huấn luyện
- Lưu trọng số
- Tạo model (gọi hàm ra, chi tiết bên trong hàm)
- Hàm optimizer
- Load trọng số (hoàn toàn, một phần) để huấn luyện hoặc test
- Định nghĩa loss function
- Data loader

NỘI DUNG

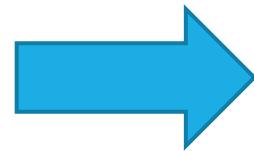
- Cần chuẩn bị những gì
 - Ngôn ngữ lập trình/ môi trường
 - Framework
 - **Server huấn luyện**
 - Phần cứng chạy suy luận
- Quy trình thực hiện một đề tài
- Một số kỹ năng quan trọng
 - Kỹ năng tìm kiếm trên google
 - Debug
 - Một số lỗi thường gặp

Chuẩn bị

Server huấn luyện

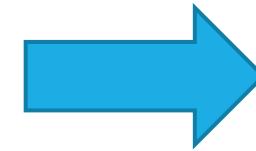
Simple task (SVM, K-NN, K-mean, small CNN, random forest,..)

CPU



Complex task (Object detection, Object segmentation, Deep neural network,...)

GPU - laptop



GPU - PC



Chuẩn bị

Server huấn luyện

Google Colab

- Miễn phí – GPU K80
- Dung lượng tối đa tải lên 12GB
- Sử dụng Ipython – live code

Free! Cloud Server
TPU & GPU



Google
colab



	Price	GPU	Runtime	Memory
Colab	Free	K80	Up to 12 hours	12GB
Colab Pro	\$9.99/m (before tax)	T4 & P100	Up to 24 hours	25GB with high memory VMs

Copy of TFJS-collab.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Remember not to use `const` or `let`! Use `var` instead

This is how you can execute shell commands:

```
▶ var { spawn } = require('child_process');
  var sh = (cmd) => {
    $$._async();
    var sp = spawn(cmd, { cwd: process.cwd(), stdio: 'pipe', shell: true, encoding: 'utf8' });
    sp.stdout.on('data', data => console.log(data.toString()));
    sp.stderr.on('data', data => console.error(data.toString()));
    sp.on('close', () => $$._done());
  };
  var run_async = async (pf) => {
    $$._async();
    await pf();
    $$._done();
  };
  sh('npm init -y');
```

NỘI DUNG

- Cần chuẩn bị những gì
 - Ngôn ngữ lập trình/ môi trường
 - Framework
 - Server huấn luyện
 - **Phần cứng chạy suy luân**
- Quy trình thực hiện một đề tài
- Một số kỹ năng quan trọng
 - Kỹ năng tìm kiếm trên google
 - Debug
 - Một số lỗi thường gặp

Chuẩn bị

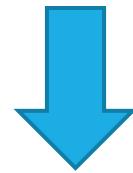
Phần cứng chạy suy luận

Laptop



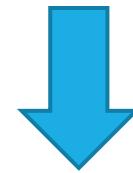
CPU or GPU

Smartphone



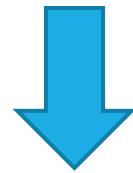
CPU (or GPU)

Pocket pc



CPU

PC



GPU

Chuẩn bị

Phần cứng chạy suy luận

Development kit

Raspberry Pi



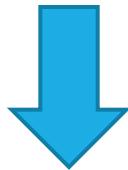
Latte panda



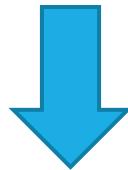
Jetson Nano



Jetson TX2



CPU



CPU



GPU

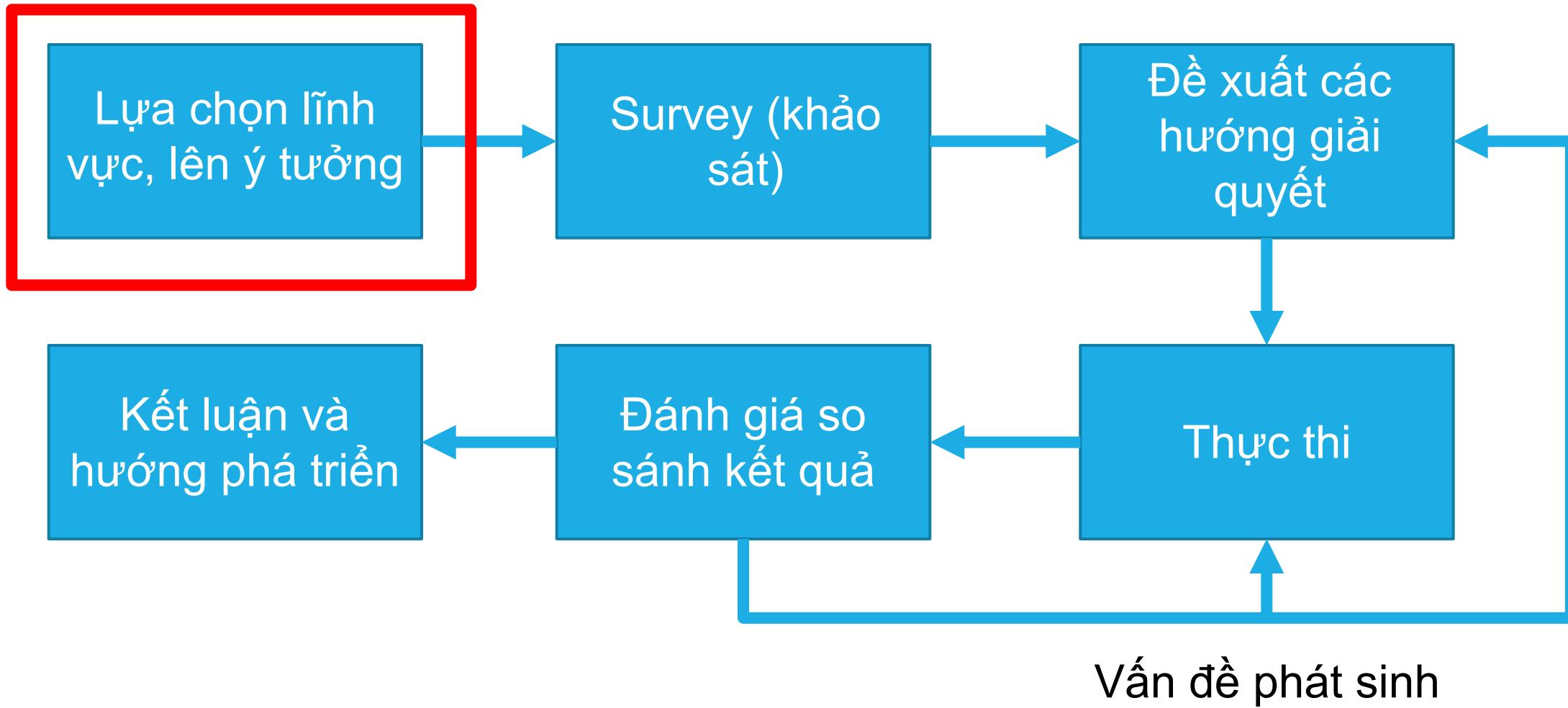


GPU

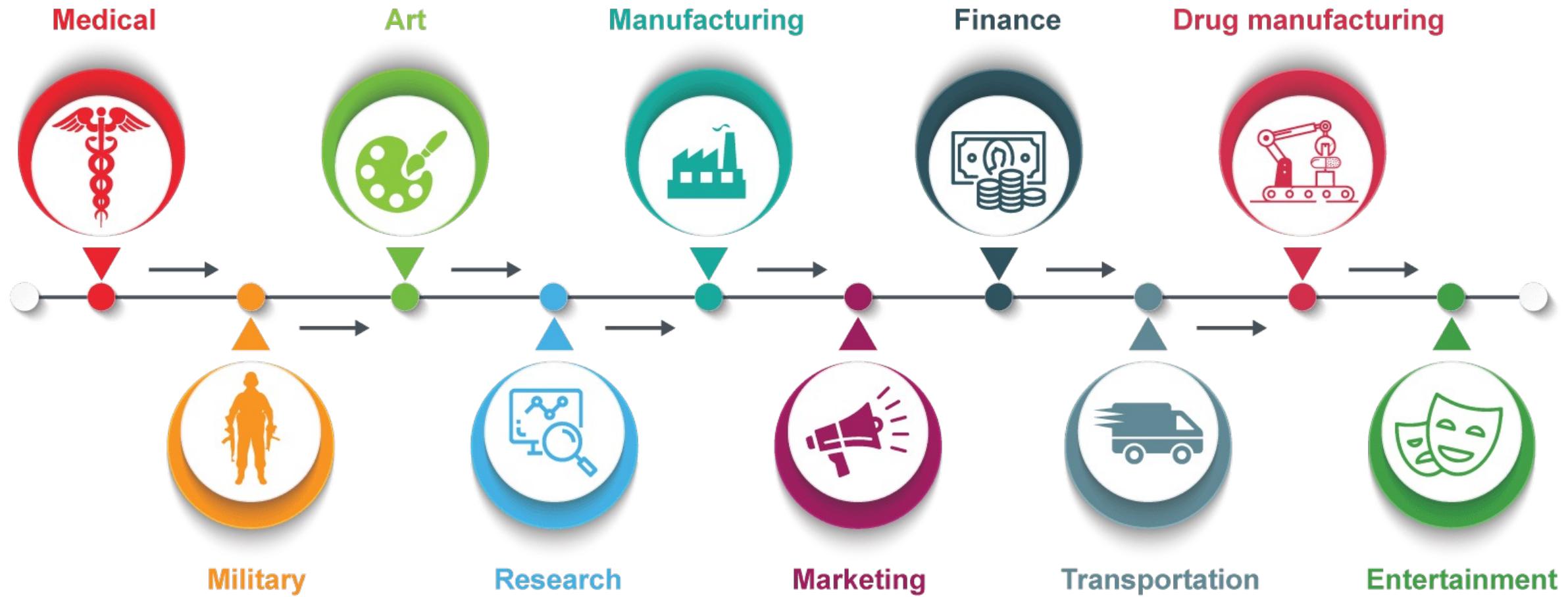
NỘI DUNG

- Cần chuẩn bị những gì
 - Ngôn ngữ lập trình/ môi trường
 - Framework
 - Server huấn luyện
 - Phần cứng chạy suy luận
- **Quy trình thực hiện một đề tài**
- Một số kỹ năng quan trọng
 - Kỹ năng tìm kiếm trên google
 - Debug
 - Một số lỗi thường gặp

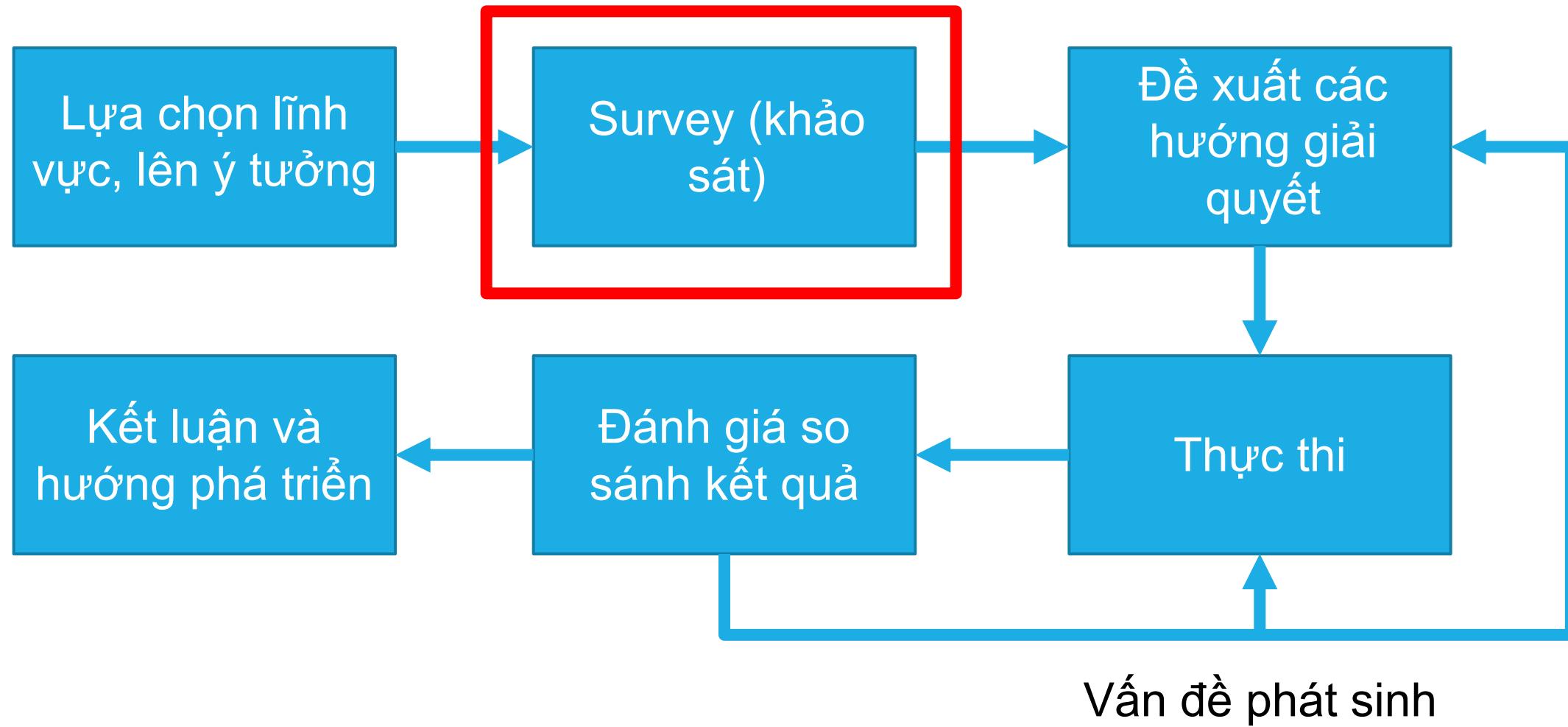
Quy trình



Quy trình Lựa chọn lĩnh vực, lên ý tưởng



Quy trình



- Phần cứng chạy chương trình
- Thách thức / khó khăn phải đổi mới
- Khảo sát các bài báo – các thuật toán khả thi
- Datasets
- Các phương pháp đánh giá

Quy trình

Khảo sát

Read paper

<https://ieeexplore.ieee.org/Xplore/home.jsp>

The screenshot shows the IEEE Xplore homepage. At the top, there's a navigation bar with links to IEEE.org, IEEE Xplore, IEEE-SA, IEEE Spectrum, and More Sites. On the right, there are buttons for SUBSCRIBE, Cart, Create Account, and Personal Sign In. The main search bar has 'All' selected and a magnifying glass icon. Below the search bar, there's a banner for the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. A document summary for 'MobileNetV2: Inverted Residuals and Linear Bottlenecks' is displayed, including the publisher (IEEE), citation information ('Cite This'), and download options ('PDF'). The document has 724 Paper Citations and 2199 Full Text Views. To the right, there's a sidebar with a blue banner asking 'Need Full-Text access to IEEE Xplore for your organization?' and a 'CONTACT IEEE TO SUBSCRIBE' button. The bottom of the sidebar shows a chart titled 'Top Organizations with Patents on Technologies Mentioned in This Article' with bars for Organization 4, Organization 3, Organization 2, and Organization 1.

Conferences > 2018 IEEE/CVF Conference on C...

MobileNetV2: Inverted Residuals and Linear Bottlenecks

Publisher: IEEE

Cite This

PDF

Mark Sandler ; Andrew Howard ; Menglong Zhu ; Andrey Zhmoginov ; Liang-Chieh Chen All Authors

724
Paper
Citations

2199
Full
Text Views



Abstract

Document Sections

1. Introduction
2. Related Work
3. Preliminaries,
Discussion and Intuition

Abstract:

In this paper we describe a new mobile architecture, MobileNetV2, that improves the state of the art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes. We also describe efficient ways of applying these mobile models to object detection in a novel framework we call SSDLite. Additionally, we demonstrate how to build mobile semantic segmentation models through a reduced form of DeepLabv3 which we call Mobile DeepLabv3. is based on an inverted residual structure where the shortcut connections are between the thin bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. Additionally, we find that it is important to remove non-linearities in the narrow layers in order to maintain

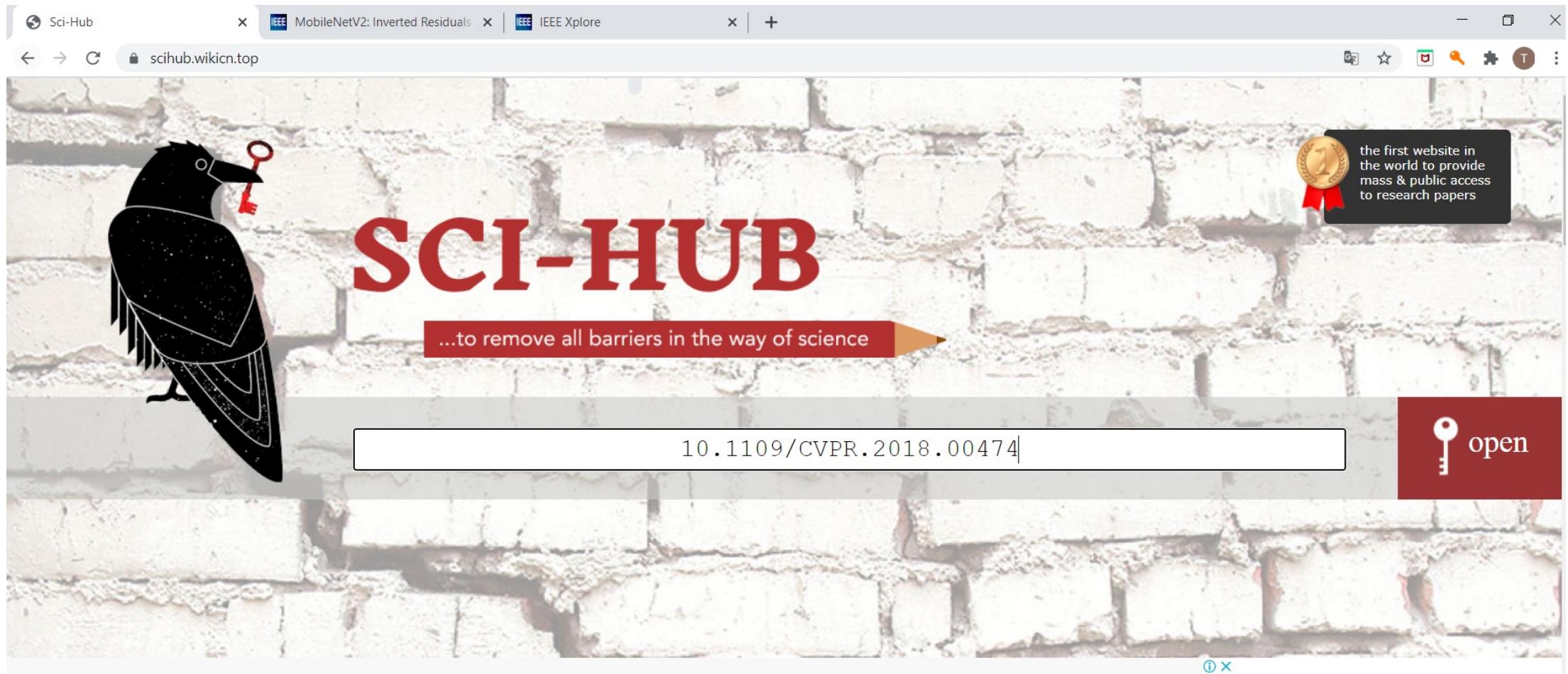
Top Organizations with Patents on Technologies Mentioned in This Article



Copy DOI

Abstract	Abstract: In this paper we describe a new mobile architecture, MobileNetV2, that improves the state of the art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes. We also describe efficient ways of applying these mobile models to object detection in a novel framework we call SSDLite. Additionally, we demonstrate how to build mobile semantic segmentation models through a reduced form of DeepLabv3 which we call Mobile DeepLabv3. is based on an inverted residual structure where the shortcut connections are between the thin bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. Additionally, we find that it is important to remove non-linearities in the narrow layers in order to maintain representational power. We demonstrate that this improves performance and provide an intuition that led to this design. Finally, our approach allows decoupling of the input/output domains from the expressiveness of the transformation, which provides a convenient framework for further analysis. We measure our performance on ImageNet [1] classification, COCO object detection [2], VOC image segmentation [3]. We evaluate the trade-offs between accuracy, and number of operations measured by multiply-adds (MAdd), as well as actual latency, and the number of parameters.
Document Sections	
1. Introduction	
2. Related Work	
3. Preliminaries, Discussion and Intuition	
4. Model Architecture	
5. Implementation Notes	
Show Full Outline ▾	
Authors	
Figures	Published in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition
References	Date of Conference: 18-23 June 2018 INSPEC Accession Number: 18311793
Citations	Date Added to IEEE Xplore: 17 December 2018 DOI: 10.1109/CVPR.2018.00474
Keywords	▶ ISBN Information: Publisher: IEEE
Metrics	▶ ISSN Information: Conference Location: Salt Lake City, UT, USA

Paste DOI



Result

sci-hub.st/10.1109/CVPR.2018.00474



sci-hub
to open science

↓ save

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2018.00474

url to share this paper:
sci-hub.st/10.1109/CVPR.2018.00474



MobileNetV2: Inverted Residuals and Linear Bottlenecks 1 / 11

2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition

MobileNetV2: Inverted Residuals and Linear Bottlenecks

Mark Sandler Andrew Howard Menglong Zhu Andrey Zhmoginov Liang-Chieh Chen
Google Inc.
{sandler, howarda, menglong, azhmogin, lcchen}@google.com

Abstract

In this paper we describe a new mobile architecture, MobileNetV2, that improves the state of the art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes. We also describe efficient ways of applying these mobile models to object detection in a novel framework we call SSDLite. Additionally, we demonstrate how to build mobile semantic segmentation models through

applications.

This paper introduces a new neural network architecture that is specifically tailored for mobile and resource constrained environments. Our network pushes the state of the art for mobile tailored computer vision models, by significantly decreasing the number of operations and memory needed while retaining the same accuracy.

Our main contribution is a novel layer module: the inverted residual with linear bottleneck. This module takes as an input a low dimensional convolutional

Quy trình Khảo sát – ví dụ

Xe chạy nhanh,
camera rung giật.



Biển báo và làn đường mờ
và xuất hiện bóng mờ.



Quy trình Khảo sát – ví dụ

Chạy ngoài
đời thực



Đa dạng về góc nhìn, màu
sắc và kích thước.



Quy trình Khảo sát – ví dụ

Làn đường thực
tế



Vách trăng đứt đoạn,
mất và uốn cong



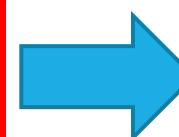
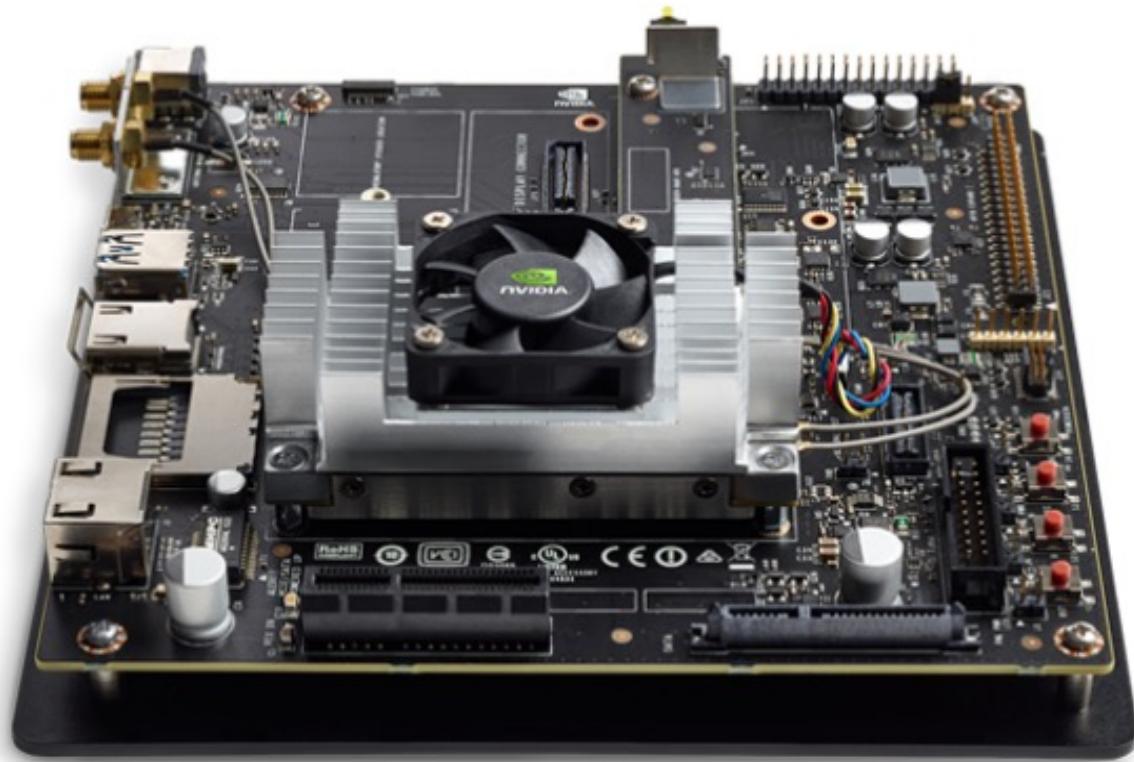
Quy trình

Khảo sát – ví dụ

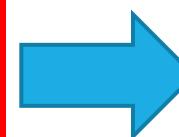
Phần cứng
Jetson TX2



Tốc độ đáp ứng thuật toán cao (FPS)



NHẸ



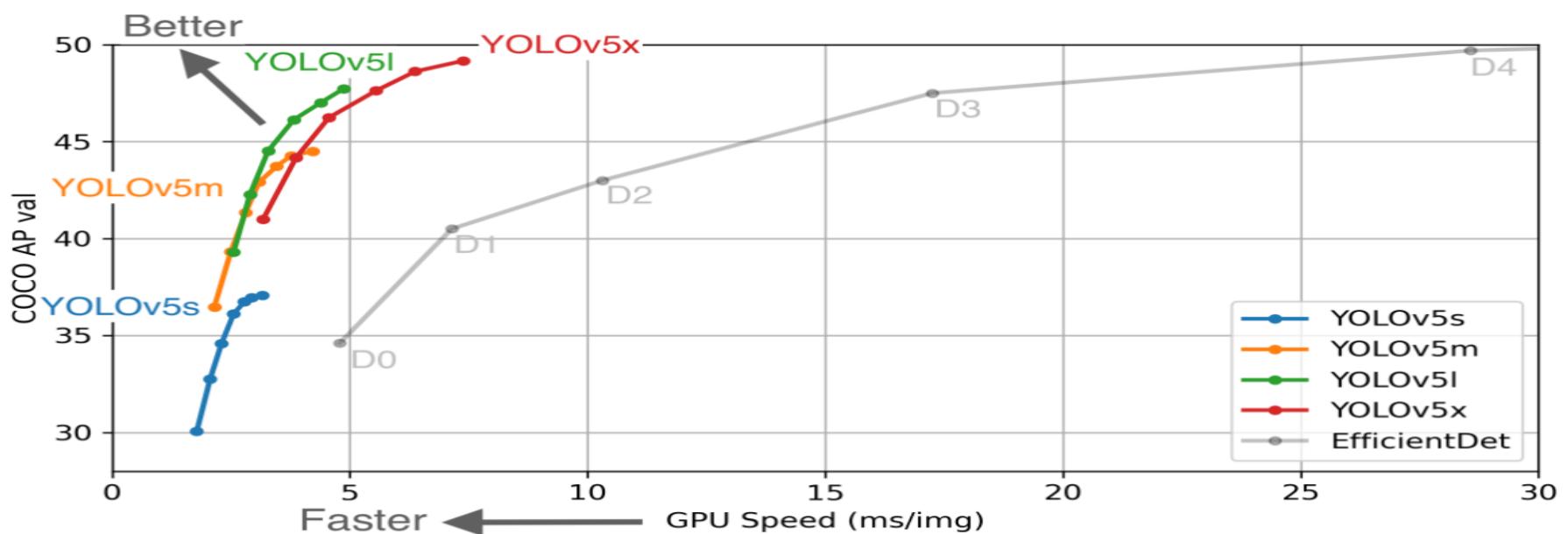
ÔN ĐỊNH

Các phương pháp đề xuất

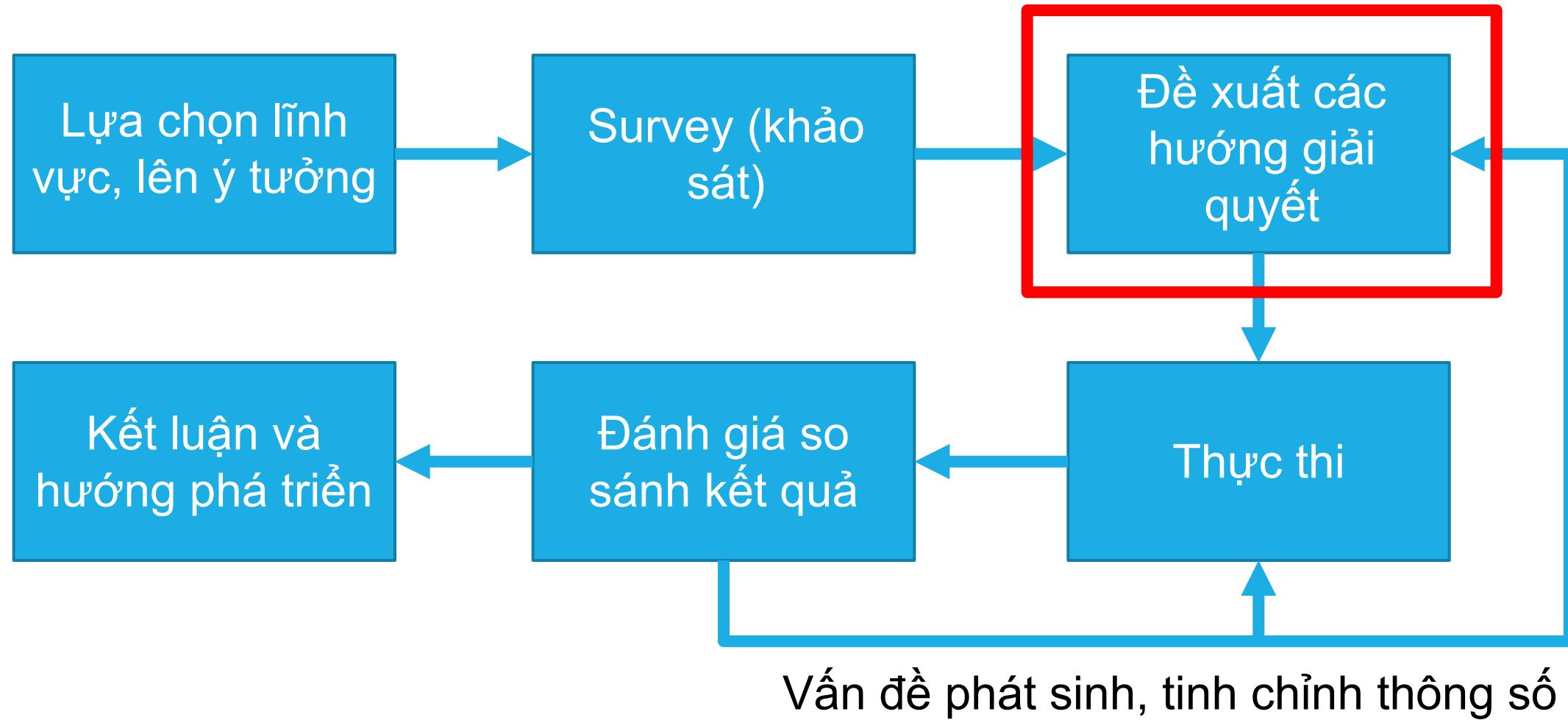
Methods	Lighting effects	Blur and fade	Motion artifact	Chaotic backgrounds	Viewing angle problems	Fast algorithm for real-time on Jetson TX2
Conventional Image Processing		✓	✓			✓
Support vector machine (SVM) [15]	✓	✓			✓	✓
Single Shot Multi-Box Detector (SSD) [16]	✓	✓	✓	✓	✓	
Faster Region-Based Convolutional Neural Network(Faster RCNN) [17]	✓	✓	✓	✓	✓	

Chỉ ra độ tốt của phương pháp hiện tại so với pháp cũ

	YOLO v5	Faster RCNN
Inference Speed	✓	
Detection of small or far away objects	✓	
Little to no overlapping boxes	✓	
Missed Objects	✗	✗
Detection of crowded objects	✓	✓



Quy trình



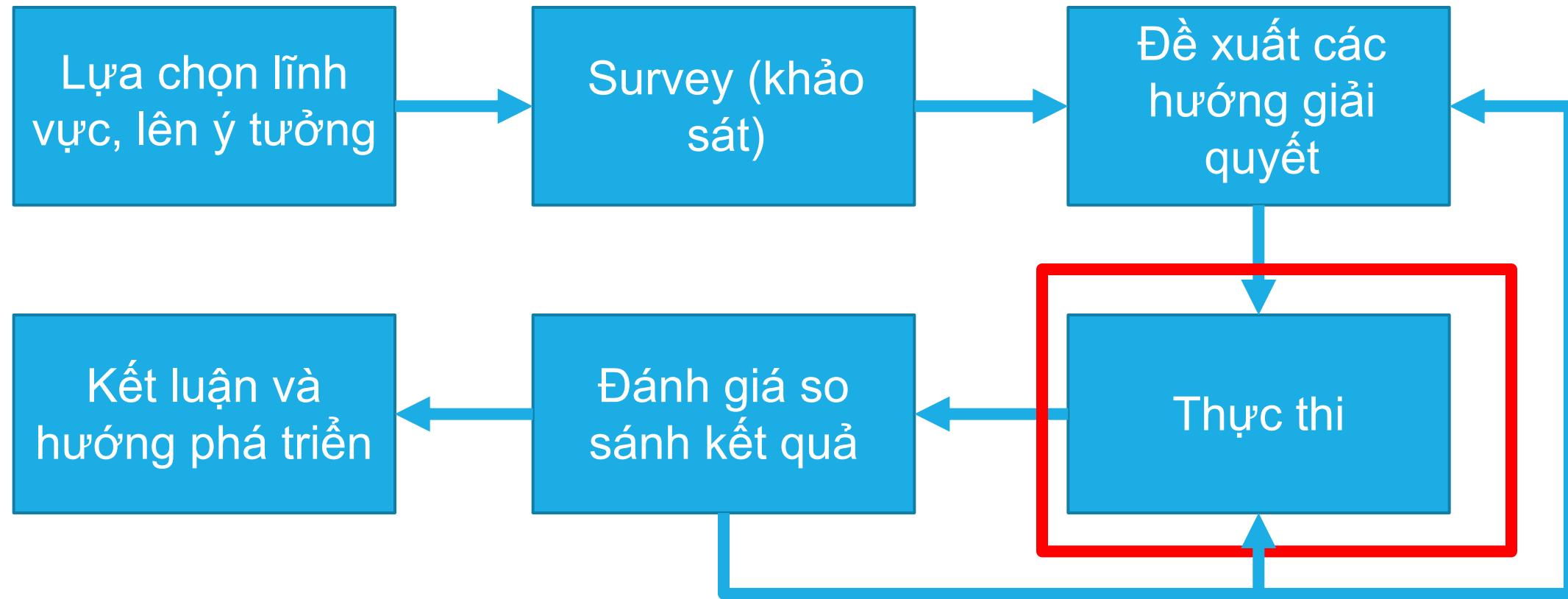
Các phương pháp đề xuất

Methods	Lighting effects	Blur and fade	Motion artifact	Chaotic backgrounds	Viewing angle problems	Fast algorithm for real-time on Jetson TX2
Conventional Image Processing		✓	✓			✓
Support vector machine (SVM) [15]	✓	✓			✓	✓
Single Shot Multi-Box Detector (SSD) [16]	✓	✓	✓	✓	✓	
Faster Region-Based Convolutional Neural Network(Faster RCNN) [17]	✓	✓	✓	✓	✓	

Lưu ý:

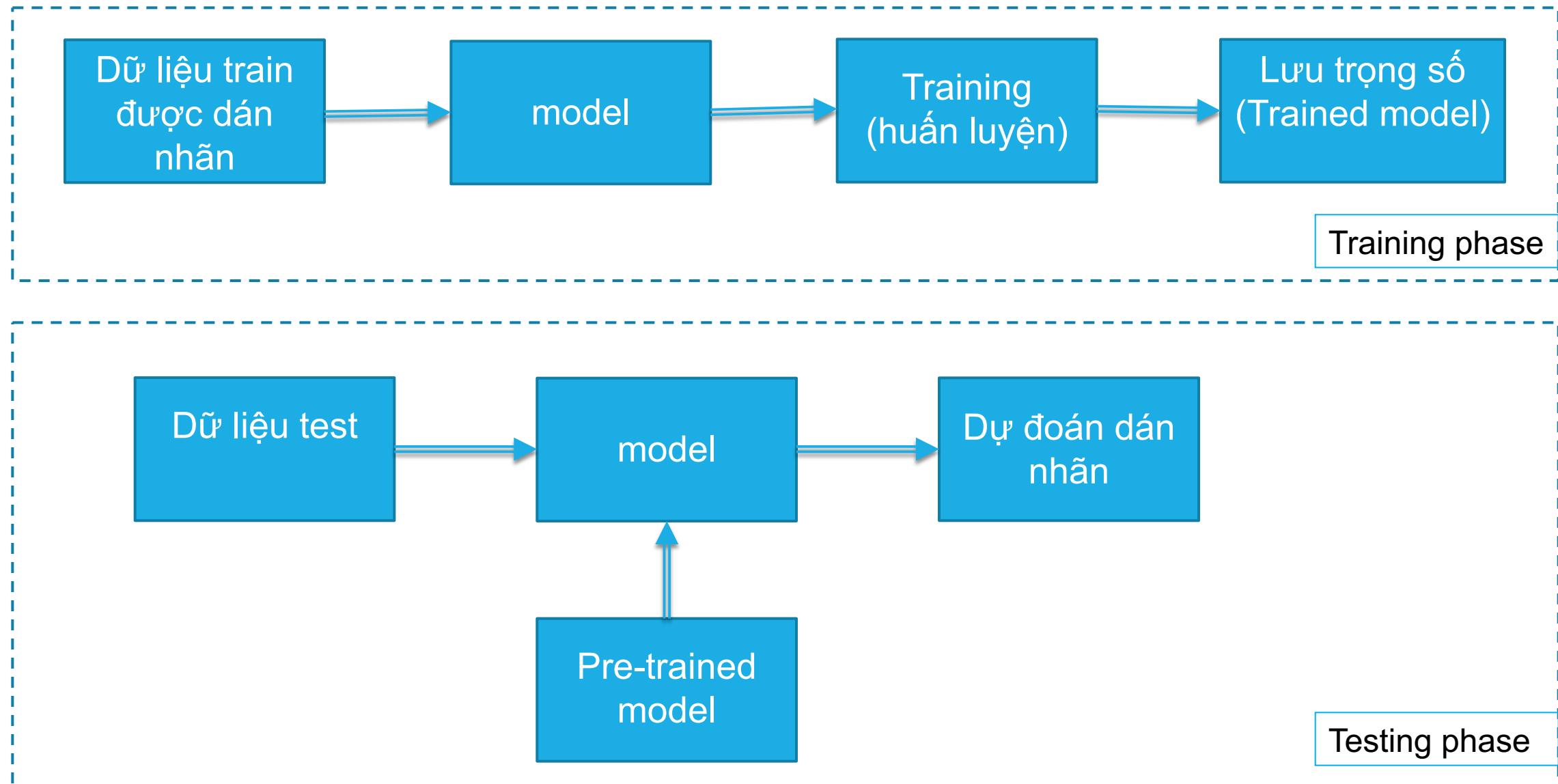
- Phải hiểu được thuật toán (loss function, kiến trúc mạng, kiến trúc mở rộng, tại sao nó giải quyết được vấn đề này, ..)
- Nên tham khảo source code nhưng không nên quá lạm dụng (không hiểu thuật toán, chỉ cho train và chạy theo hướng dẫn)
- Khi lựa chọn tham khảo code trên github, phải kiểm tra được xác nhận bởi tác giả của bài báo hoặc được nhiều sao

Quy trình



Vấn đề phát sinh, tinh chỉnh thông số

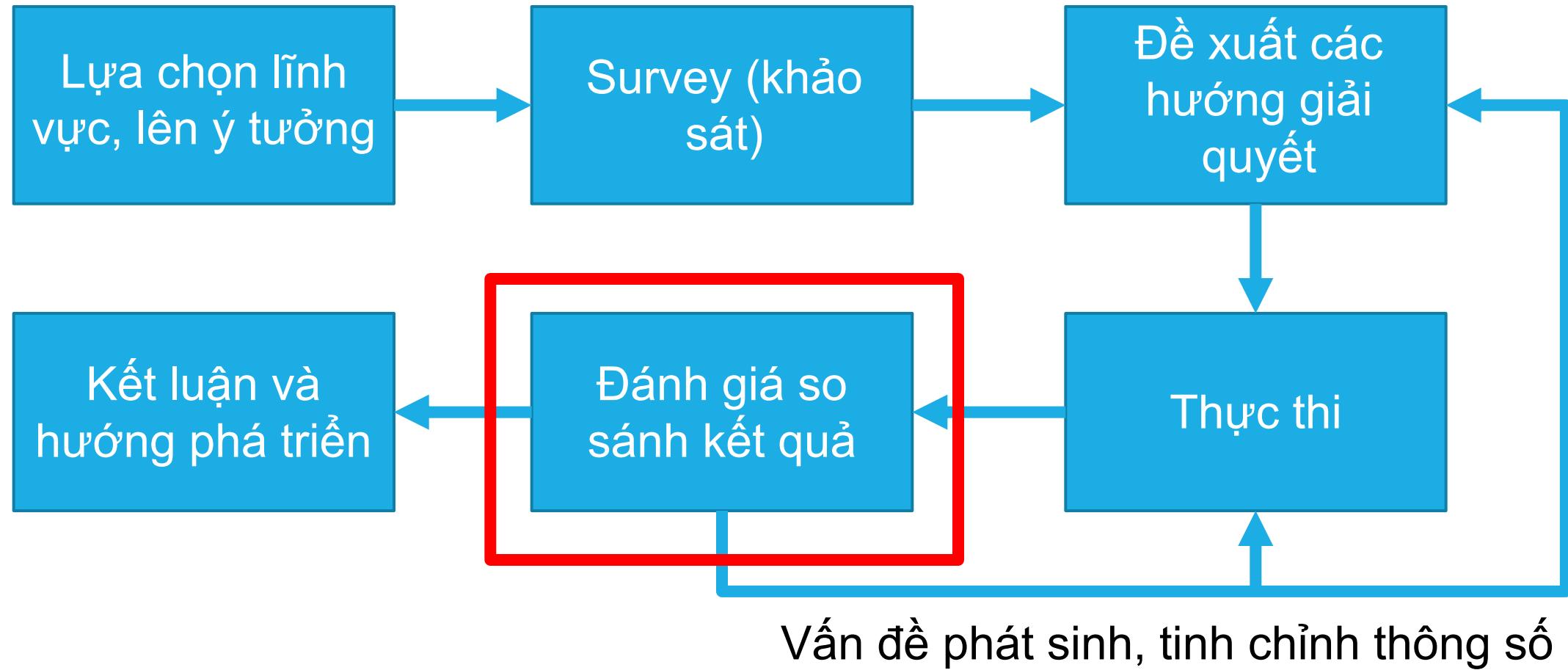
Quy trình Thực thi



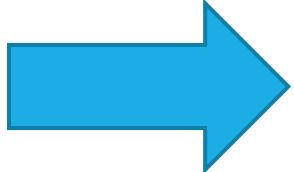
Nêu rõ:

- Dataset
- Thông số huấn luyện mạng
- Kỹ thuật huấn luyện mạng (data augmentation, multi-scale input, fine-tuned, feature extractor,...)

Quy trình



Nêu rõ:

- Phương pháp đánh giá
 - Nhận xét model (overfitting?, underfitting?)
 - Show kết quả
 - Có vấn đề nào phát sinh mới không?
-
- 
- Model đủ tốt chưa, nếu chưa quay lại bước thực thi tinh chỉnh thông số
 - Nếu có vấn đề phát sinh thêm thì quay lại bước đề xuất