



# THÔNG TIN CHUNG CỦA NHÓM

- Link YouTube video của báo cáo (tối đa 5 phút):  
<https://youtu.be/LvsbiyFaVxE>
- Link slides (dạng .pdf đặt trên Github của nhóm):  
[https://github.com/KhaVo12/CS519.O11/blob/master/Slide\\_TranQuocLuong-VoPhanHoangKha\\_CS519.DeCuong.FinalReport.Slide.pdf](https://github.com/KhaVo12/CS519.O11/blob/master/Slide_TranQuocLuong-VoPhanHoangKha_CS519.DeCuong.FinalReport.Slide.pdf)

<ul style="list-style-type: none"><li>● Họ và tên: Trần Quốc Lượng</li><li>● MSSV: 20521590</li></ul> 	<ul style="list-style-type: none"><li>● Lớp: CS519.O11</li><li>● Tự đánh giá (điểm tổng kết môn): 7.5/10</li><li>● Số buổi vắng: 3</li><li>● Số câu hỏi QT cá nhân: 3</li><li>● Số câu hỏi QT của cả nhóm: 15</li><li>● Link Github: <a href="https://github.com/KhaVo12/CS519.O11">https://github.com/KhaVo12/CS519.O11</a></li><li>● Mô tả công việc và đóng góp của cá nhân cho kết quả của nhóm:<ul style="list-style-type: none"><li>○ Chuẩn bị và trình bày nội dung của đề cương.</li><li>○ Chuẩn bị nội dung và làm slide.</li><li>○ Làm video YouTube.</li></ul></li></ul>
<ul style="list-style-type: none"><li>● Họ và tên: Võ Phan Hoàng Kha</li><li>● MSSV: 20521428</li></ul> 	<ul style="list-style-type: none"><li>● Lớp: CS519.O11</li><li>● Tự đánh giá (điểm tổng kết môn): 7.5/10</li><li>● Số buổi vắng: 3</li><li>● Số câu hỏi QT cá nhân: 3</li><li>● Số câu hỏi QT của cả nhóm: 15</li><li>● Link Github: <a href="https://github.com/KhaVo12/CS519.O11">https://github.com/KhaVo12/CS519.O11</a></li><li>● Mô tả công việc và đóng góp của cá nhân cho kết quả của nhóm:</li></ul>

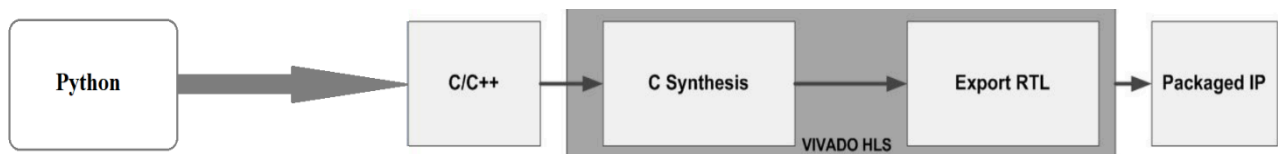
	<ul style="list-style-type: none"> <li>○ Chuẩn bị nội dung và làm poster</li> <li>○ Chuẩn bị nội dung và làm slide.</li> <li>○ Kiểm tra và sửa lỗi nội dung của đề cương.</li> </ul>
--	--

## ĐỀ CƯƠNG NGHIÊN CỨU

<b>TÊN ĐỀ TÀI:</b> NGHIÊN CỨU VÀ THIẾT KẾ PHẦN CỨNG CHO CONVOLUTIONAL NEURAL NETWORK
<b>TÊN ĐỀ TÀI TIẾNG ANH:</b> RESEARCHING AND DESIGNING HARDWARE FOR CONVOLUTIONAL NEURAL NETWORK
<p><b>TÓM TẮT:</b></p> <p>Đề cương nghiên cứu này sẽ xoay quanh hai phần chính, phần một sẽ tập trung tìm hiểu về Convolutional Neural Network, một loại mô hình deep learning phổ biến nhất trong lĩnh vực machine learning. Phần thứ hai là xây dựng cấu trúc Convolutional Neural Network trên ngôn ngữ lập trình C và Python theo hướng tiếp cận phần cứng. Chuyển đổi ngôn ngữ C sang Verilog bằng phương pháp tổng hợp cấp cao và kiểm chứng hoạt động. Bên cạnh đó, sau khi đã hoàn thành việc hiện thực thành công phần cứng, chúng tôi sẽ phát triển phần cứng về mặt tốc độ xử lý của hệ thống và tối ưu các tài nguyên mà phần cứng sử dụng.</p>
<p><b>GIỚI THIỆU:</b></p> <p>Machine Learning và Deep Learning đang có những bước phát triển vượt bậc và xuất hiện ngày càng phổ biến hơn, trở thành xu thế trong ngành công nghệ trên thế giới. Các mô hình học sâu, đặc biệt là Convolutional Neural Network, một loại mô hình học sâu rất mạnh trong các bài toán về thị giác máy tính, cần một lượng rất lớn tài nguyên tính toán cho việc vận hành, do đó các bộ xử lý đồ họa (Graphics Processing Unit) được phát triển và áp dụng cho các nền tảng học sâu nhằm tăng tốc độ xử lý nhờ</p>

kỹ thuật xử lý song song và đa luồng. Bên cạnh đó, FPGA (Field-Programmable Gate Array) cũng đang được hướng đến để sử dụng như một nền tảng phần cứng cho việc tính toán của các mô hình học sâu. FPGA đang cho thấy một **sự cải thiện lớn về năng lượng tiêu thụ và hiệu năng** trong các ứng dụng của các mạng học sâu yêu cầu độ chính xác cao như các bài toán phân lớp. Hơn thế nữa, FPGA cho phép triển khai các mô hình học máy và học sâu lên những hệ thống nhúng chiếm diện tích nhỏ mà vẫn đảm bảo về tốc độ xử lý và độ chính xác nhờ vào việc xử lý song song trên phần cứng, cùng với đó là việc dễ dàng thay đổi và nâng cấp nhờ vào đặc tính khả trình của FPGA. Do đó FPGA đang được áp dụng rộng rãi hơn trong lĩnh vực học máy nói chung và trong các Convolutional Neural Network nói riêng. Từ những lý do trên, đề tài “Nghiên cứu và thiết kế phần cứng cho Convolutional Neural Network” đã được chúng tôi chọn với mục đích nghiên cứu tìm hiểu và bước đầu đưa mô hình Convolutional Neural Network đơn giản lên FPGA và từ đó phát triển bằng cách tối ưu về thời gian xử lý, năng lượng tiêu thụ và tài nguyên tính toán so với các nền tảng tính toán cho các mô hình học sâu khác. Chi tiết cụ thể sẽ được biểu diễn với hình sau:

S



## MỤC TIÊU

- 1) Xây dựng thành công một Convolutional Neural Network với mô hình Lenet5 trên ngôn ngữ Python, sau đó chuyển đổi mô hình sang ngôn ngữ C/C++.
- 2) Thiết kế được IP cho mô hình Lenet5 từ ngôn ngữ lập trình C/C++ bằng phương pháp tổng hợp cấp cao Vivado HLS.
- 3) Từ một thiết kế CNN đơn giản để phân loại chữ viết tay, chúng tôi phát triển lên thiết kế có thể phát triển các vật thể khác như động vật, xe cộ trong cùng một khung hình.

## **NỘI DUNG VÀ PHƯƠNG PHÁP**

### **1) NỘI DUNG:**

- Nghiên cứu cách hoạt động của Convolutional Neural Network như phép tính tích chập, các lớp gộp và tìm hiểu về Neural Network thông thường như các lớp kết nối đầy đủ, các hàm kích hoạt phi tuyến.
- Tìm hiểu về mô hình Lenet5 được sử dụng phổ biến cho Convolutional Neural Network và tiến hành triển khai mô hình bằng ngôn ngữ Python.
- Nghiên cứu về cách mô phỏng lại hoạt động của mô hình bằng ngôn ngữ lập trình C/C++ với các trọng số và hiệu số được lấy từ mô hình trên Python.
- Tiến hành chuyển đổi mô hình sang ngôn ngữ mô tả phần cứng HDL bằng phương pháp tổng hợp cấp cao.

### **2) PHƯƠNG PHÁP:**

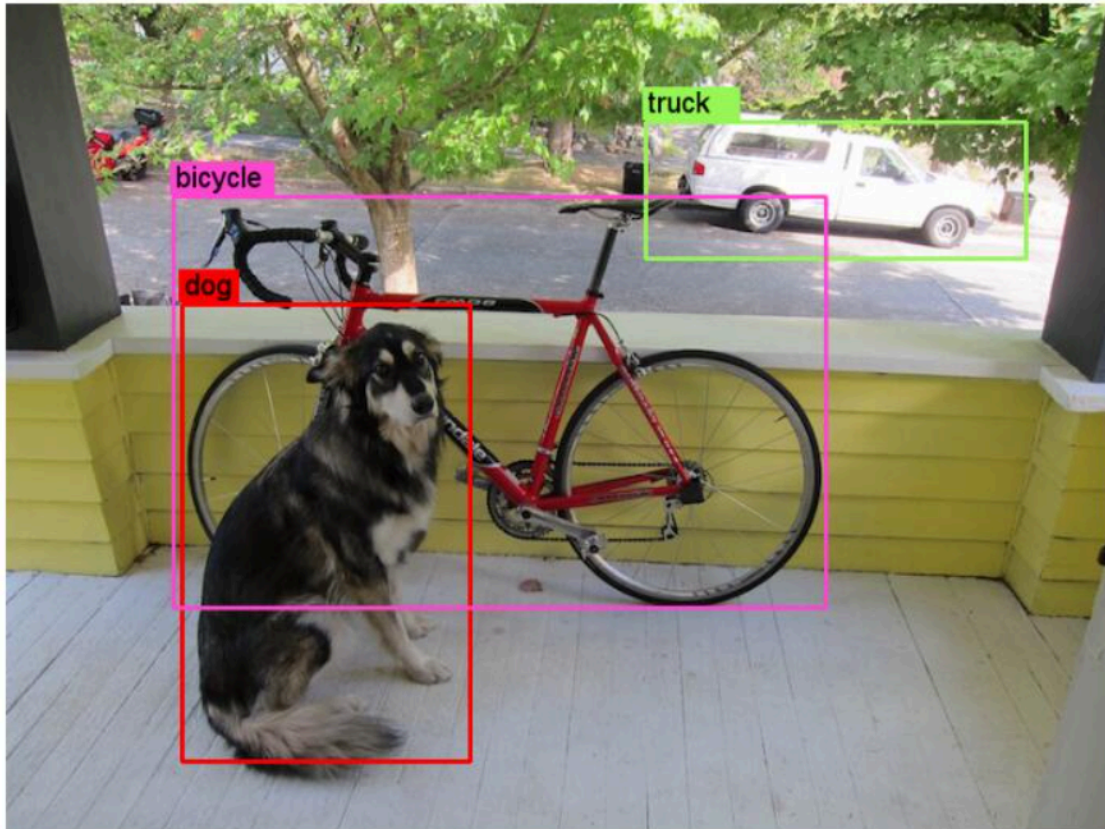
- Sử dụng các tập dữ liệu có sẵn phục vụ cho việc học máy bao gồm tập dữ liệu MNIST, CIFAR10, Imagenette cùng với thử nghiệm các mô hình khác ngoài Lenet5 như VGG16, MobileNetV2.
- Đánh giá độ chính xác của từng mô hình với từng loại tập dữ liệu, ở đây chúng tôi sử dụng mô hình Lenet5 với tập dữ liệu MNIST vì các thử nghiệm cho thấy độ chính xác cao khi mô hình được sử dụng cho tập dữ liệu này.
- Sử dụng công cụ tổng hợp cấp cao Vivado HLS để chuyển đổi mô hình từ ngôn ngữ C/C++ sang ngôn ngữ HDL, công cụ này giúp tạo ra các phần cứng từ đơn giản tới phức tạp mà không phải bắt đầu từ ngôn ngữ HDL, điều này giúp tiết kiệm thời gian thiết kế, đồng thời cũng giúp tối ưu phần cứng thay cho người thiết kế.

## **KẾT QUẢ MONG ĐỢI**

- 1) Mô hình CNN được thiết kế bằng ngôn ngữ Python và C/C++ có khả năng phân loại tập dữ liệu MNIST.
- 2) Tổng hợp thành công thiết kế sang ngôn ngữ HDL bằng phương pháp tổng hợp cấp cao với phần mềm Vivado HLS, cho ra hệ thống có thể phân loại vật thể, với

kết quả minh họa như hình bên dưới.

Out[18]:



## TÀI LIỆU THAM KHẢO

- [1]. Yongming Shen, Michael Ferdman, Peter Milder: Maximizing CNN Accelerator Efficiency Through Resource Partitioning. ISCA '17, 2017.
- [2]. Jiantao Qiu, Jie Wang, Song Yao, Kaiyuan Guo, Boxun Li, Erjin Zhuo, Jincheng Yu, Tianqi Tang, Ningyi Xu, Sen Song, Yu Wang, Huazhong Yang: Going Deeper with Embedded FPGA Platform for Convolutional Neural Network. FPGA '16, 2016.
- [3]. Lei Shan, Minxuan Zhang, Lin Deng, Guohui Gong: A Dynamic Multi-precision Fixed-Point Data Quantization Strategy for Convolutional Neural Network. NCCET 2016, 2016.
- [4]. Yufei Ma, Yu Cao, Sarma Vrudhula, Jae-sun Seo: An automatic RTL compiler for high-throughput FPGA implementation of diverse deep convolutional neural networks. FPL 2017, 2017.
- [5]. Chen Zhang, Peng Li, Guangyu Sun, Yijin Guan: Optimizing FPGA-based Accelerator Design for Deep Convolutional Neurtworks. FPGA '15, 2015.