**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**LÊ MINH PHÚC**

**BÙI NGUYỄN PHÁT**

**BÙI HỮU TRÍ**

**ĐỒ ÁN MÔN HỌC**

**CHUYÊN ĐỀ THIẾT KẾ VI MẠCH I**

**KỸ SƯ KĨ THUẬT MÁY TÍNH**

**TP. HỒ CHÍ MINH, 2021**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**LÊ MINH PHÚC – 17520903**

**BÙI NGUYỄN PHÁT – 17520875**

**BÙI HỮU TRÍ - 17521153**

**ĐỒ ÁN MÔN HỌC**

**CHUYÊN ĐỀ THIẾT KẾ VI MẠCH I**

**KỸ SƯ KĨ THUẬT MÁY TÍNH**

**GIẢNG VIÊN HƯỚNG DẪN**

**TRƯƠNG VĂN CƯƠNG**

**TP. HỒ CHÍ MINH, 2021**

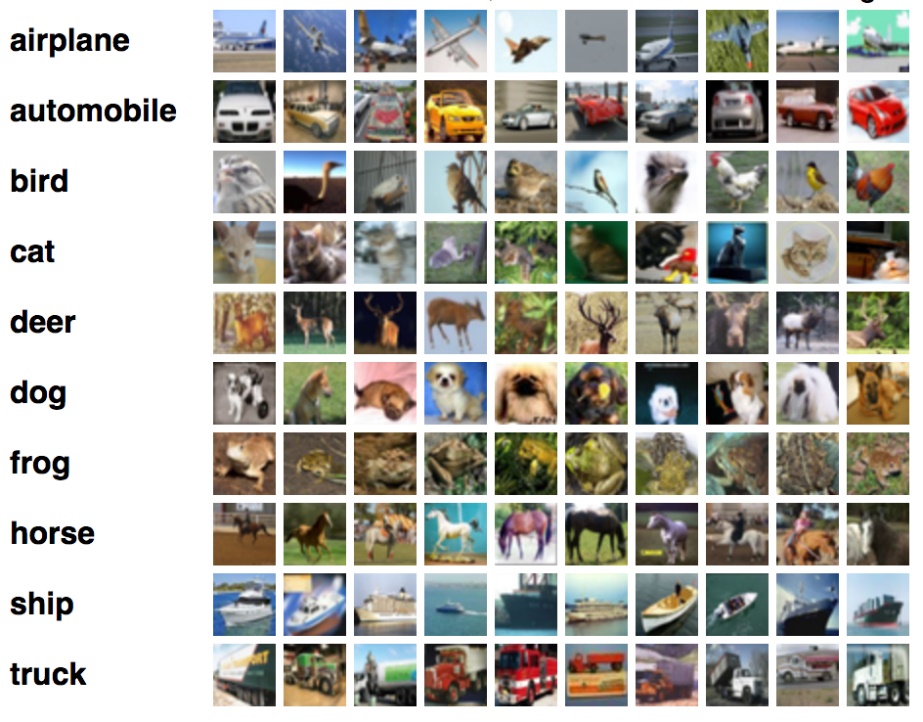
**I. TÌM HIỂU TỔNG QUAN:**

**1. BỘ DỮ LIỆU CIFAR-10:**

Bộ dữ liệu CIFAR-10 (Canadian Institute For Advanced Research - Viện Nghiên cứu Nâng cao của Canada) là một tập hợp các hình ảnh thường được sử dụng để đào tạo các thuật toán máy học và thị giác máy tính. Nó là một trong những bộ dữ liệu được sử dụng rộng rãi nhất cho nghiên cứu máy học. Bộ dữ liệu CIFAR-10 chứa 60.000 hình ảnh màu 32x32 ở 10 lớp khác nhau. 10 lớp khác nhau đại diện cho máy bay, ô tô, chim, mèo, hươu, nai, chó, ếch, ngựa, tàu và xe tải. Có 6.000 hình ảnh của mỗi lớp.

Các thuật toán máy tính để nhận dạng các đối tượng trong ảnh thường học bằng ví dụ. CIFAR-10 là một bộ hình ảnh có thể được sử dụng để dạy máy tính cách nhận dạng các đối tượng. Vì hình ảnh trong CIFAR-10 có độ phân giải thấp (32x32), bộ dữ liệu này có thể cho phép các nhà nghiên cứu nhanh chóng thử các thuật toán khác nhau để xem cái nào hoạt động. Nhiều loại mạng nơ-ron phức hợp khác nhau có xu hướng tốt nhất trong việc nhận dạng hình ảnh trong CIFAR-10.

CIFAR-10 là một tập hợp con được gắn nhãn của tập dữ liệu 80 triệu hình ảnh nhỏ. Khi tập dữ liệu được tạo, sinh viên được trả tiền để gắn nhãn tất cả các hình ảnh.



Hình 1. Bộ dữ liệu CIFAR-10

**2. MẠNG NEURON TÍCH CHẬP (CONVOLUTIONAL NEURAL NETWORK - CNN) :**

Trong Deep Learning, mạng nơ-ron tích chập (CNN, hoặc ConvNet) là một lớp của mạng nơ-ron sâu, được áp dụng phổ biến nhất để phân tích hình ảnh trực quan. Chúng còn được gọi là mạng nơ-ron nhân tạo bất biến dịch chuyển hoặc bất biến không gian (SIANN), dựa trên kiến ​​trúc trọng số chia sẻ của các nhân hoặc bộ lọc tích chập trượt dọc theo các đối tượng đầu vào và cung cấp các phản hồi tương đương dịch được gọi là bản đồ đối tượng. Về mặt phản trực giác, hầu hết các mạng nơ-ron tích chập chỉ tương đương, trái ngược với bất biến, đối với phép dịch. Họ có các ứng dụng trong nhận dạng hình ảnh và video, hệ thống khuyến nghị, phân loại hình ảnh, phân đoạn hình ảnh, phân tích hình ảnh y tế, xử lý ngôn ngữ tự nhiên, giao diện não-máy tính, và chuỗi thời gian tài chính.

CNN là phiên bản chính quy của perceptron nhiều lớp. Các perceptron nhiều lớp thường có nghĩa là các mạng được kết nối đầy đủ, tức là mỗi nơ-ron trong một lớp được kết nối với tất cả các nơ-ron trong lớp tiếp theo. "Kết nối đầy đủ" của các mạng này làm cho chúng dễ bị trang bị quá nhiều dữ liệu. Các cách điển hình để chính quy hóa hoặc ngăn chặn việc trang bị quá mức, bao gồm: phạt các thông số trong quá trình đào tạo (chẳng hạn như giảm trọng lượng) hoặc cắt bỏ kết nối (kết nối bị bỏ qua, bỏ qua, v.v.) CNN có cách tiếp cận khác theo hướng chính quy hóa: họ tận dụng mô hình phân cấp trong dữ liệu và lắp ráp các mẫu ngày càng phức tạp bằng cách sử dụng các mẫu nhỏ hơn và đơn giản hơn được in nổi trong bộ lọc của chúng. Do đó, trên quy mô kết nối và độ phức tạp, CNNs ở mức độ thấp hơn.

Mạng liên kết được lấy cảm hứng từ các quá trình sinh học trong đó mô hình kết nối giữa các tế bào thần kinh giống với tổ chức của vỏ não thị giác động vật. Các tế bào thần kinh vỏ não riêng lẻ chỉ phản ứng với các kích thích trong một vùng hạn chế của trường thị giác được gọi là trường tiếp nhận. Các trường tiếp nhận của các tế bào thần kinh khác nhau chồng lên nhau một phần sao cho chúng bao phủ toàn bộ trường thị giác.

CNNs sử dụng tương đối ít tiền xử lý so với các thuật toán phân loại hình ảnh khác. Điều này có nghĩa là mạng học cách tối ưu hóa các bộ lọc (hoặc hạt nhân) thông qua học tự động, trong khi trong các thuật toán truyền thống, các bộ lọc này được thiết kế thủ công. Sự độc lập này so với kiến ​​thức trước đây và sự can thiệp của con người trong việc khai thác tính năng là một lợi thế lớn.

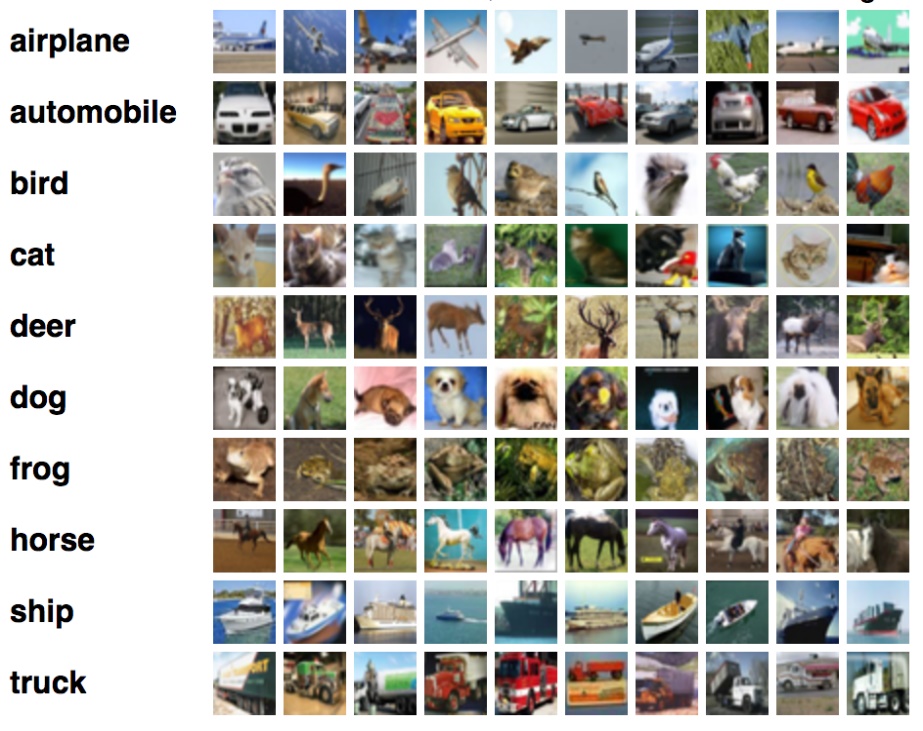
**II. MỤC TIÊU:**

**1. Input:**

Các tấm ảnh màu có kích cỡ 32x32x3 được trích trong tập dữ liệu CIFAR-10. Ảnh sẽ được lưu trong 3 file text ở định dạng số nguyên 8-bit, mỗi file tương ứng với 1 trong 3 mức màu R, G, B.

**2. Output:**

10 bit dữ liệu ở định dạng one-hot, mỗi bit sẽ thể hiện 1 lớp của bức ảnh. Với mỗi thời điểm chỉ có một bit dữ liệu ở mức cao, ví dụ nếu bức ảnh là ảnh mèo, output sẽ có dạng là “0001000000” thể hiện bức ảnh đó là ảnh mèo.



0

0

0

1

0

0

0

0

0

0

Hình 2. Output của mạng CNN với CIFAR-10 dataset

**III. NỘI DUNG:**

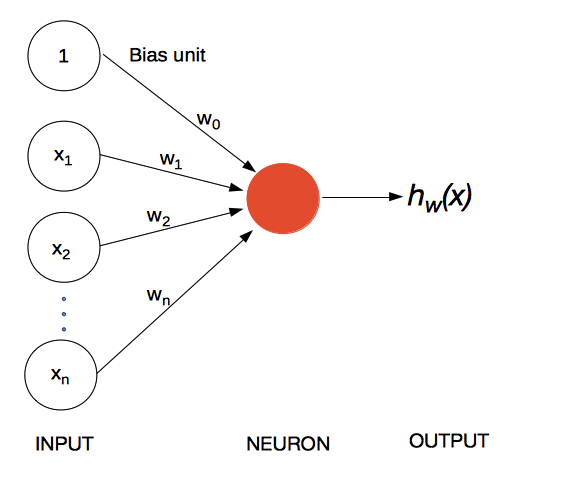
**1. Khái niệm về Deep learning:**

Là một lĩnh vực thuộc Machine Learning.

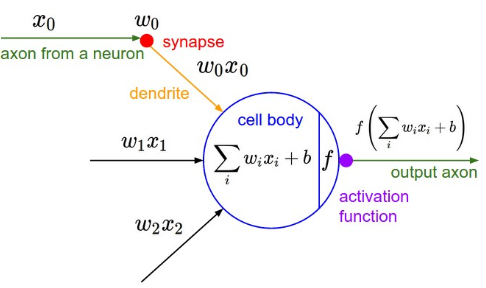
Sử dụng các thuật toán lấy ý tưởng từ khả năng hoạt động của bộ não người, gọi chung là Artificial Neural Networks (ANN).

Mô hình của DL gồm 3 lớp chính:

* Lớp Input là một vector dữ liệu đầu vào.
* Lớp Hidden Layers gồm một hoặc nhiều Perceptrons, thực hiện quá trình học.
* Lớp Output cho ra vector nhãn của dữ liệu sau khi quá trình học hoàn tất.



Hình 3. Mô hình Neuron (1)



Hình 4. Mô hình Neuron (2)

Kiến trúc cơ bản của một Neuron bao gồm 2 phần :

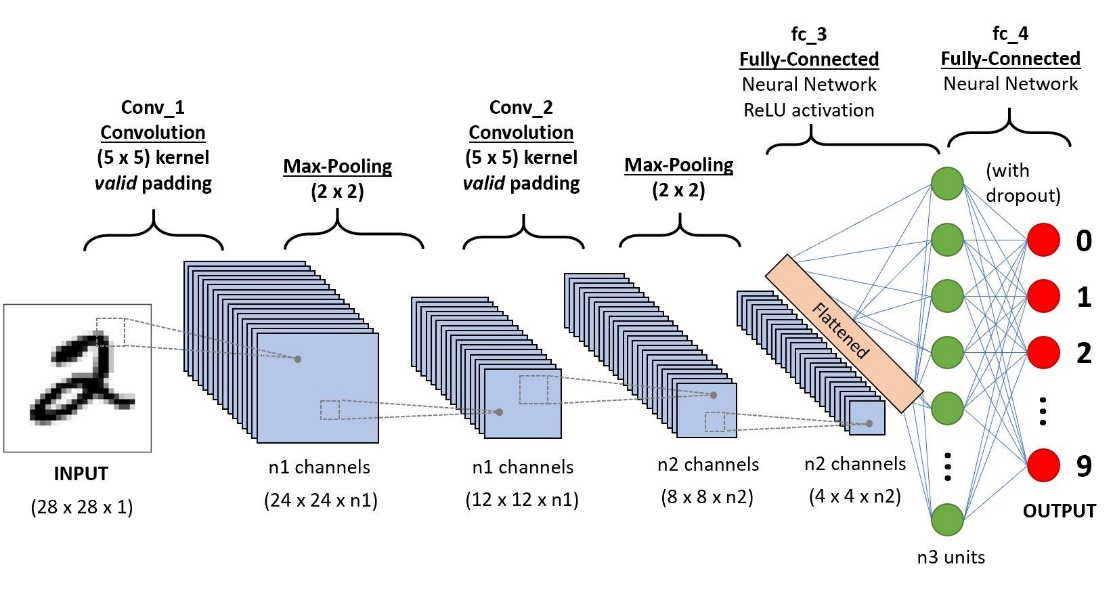
* Phần 1 : , trong đó:
  + w: Trọng số weight của mạng neural.
  + x: Input của mô hình neuron.
  + b: Bias, trọng số độ dời của mô hình neuron.
* Phần 2: , với f là một hàm kích hoạt, thông thường sẽ là hàm ReLU để lọc bỏ các giá trị âm, thay thế chúng bằng 0.

**2. Khái niệm về CNN:**

CNN được sử dụng để học nhiều đặc trưng của dữ liệu đầu vào đồng thời làm giảm kích thước của dữ liệu, giúp việc xử lý đơn giản hơn.

Output là các nhãn của dữ liệu, mô hình này có khả năng học khá tốt và thường được sử dụng nhiều trong xử lý hình ảnh, chữ viết, …

Về cơ bản CNN được chia làm 2 khối: Khối Conv-Pooling và khối Fully-Connected.

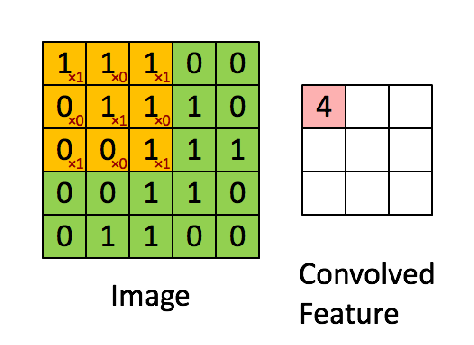


Hình 5. Mô hình mẫu CNN.

**a. Khối tích chập:**

Thực hiện phép tính tích chập cho ra những đặc trưng của dữ liệu đầu vào tùy thuộc vào loại ma trận Kernel được sử dụng.

Việc thực hiện phép tính Convolution phụ thuộc vào các yếu tố: kích cỡ ma trận Kernel, padding, stride.

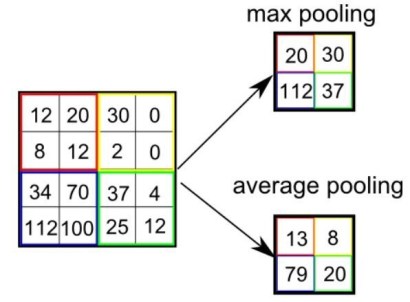


Hình 6. Phép tính tích chập.

**b. Khối pooling:**

Pooling layer được sử dụng giữa các Convolution layer, giúp giảm kích thước dữ liệu vào nhưng vẫn giữ nguyên đặc trưng của nó.

Pooling thường được sử dụng là max pooling để giữ những chi tiết đặc trưng nhất của bức ảnh, có kích cỡ 2x2, stride=2, padding=0.

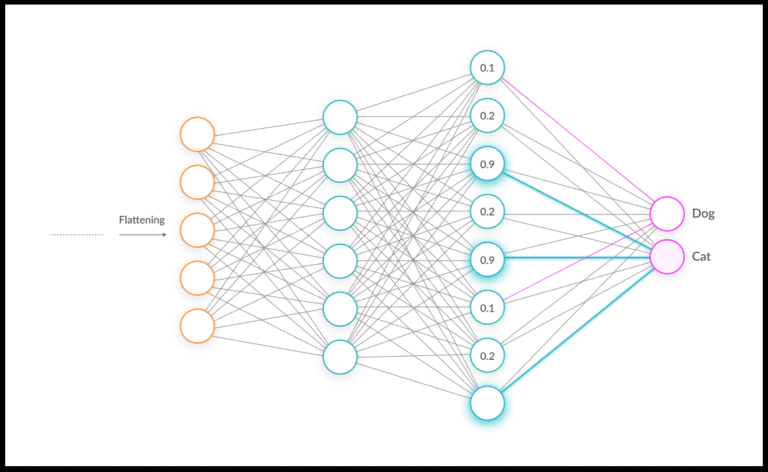


Hình 7. Phép toán pooling.

**c. Khối fully-connected:**

Sau các bước xử lý ở khối Conv-Pooling, các ma trận đầu ra của layer cuối sẽ được flatten trở thành dạng vector và đưa vào khối Fully-Connected (Cũng chính là một neural network).

Khối Fully-Connected này có nhiệm vụ tìm các trọng số w để thực hiện việc gán nhãn thích hợp cho dữ liệu đầu vào.



Hình 8. Khối Fully-connected

**d. hàm activation function**

hàm activation function ( hàm kích hoạt) được sử dụng ở mỗi hidden layer. Có rất nhiều hàm kích hoạt có thể dụng nhưng thông thường nhất là các hàm relu, sigmoid, softmax :

**sigmoid:**



**Relu:**



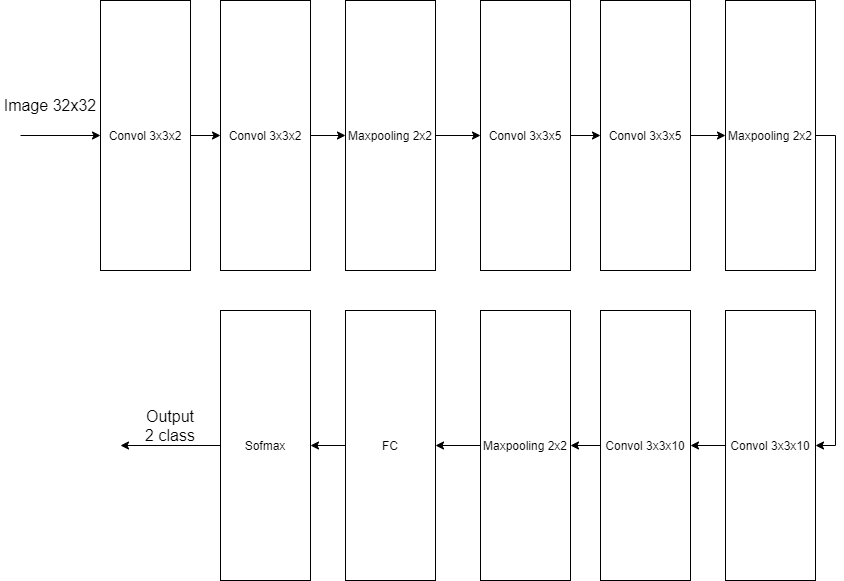
**Softmax:**

=

**3. Mô hình Cifar thực hiện**

Cifar 10 tiêu tốn quá nhiều tài nguyên phần cứng nên khó implement lên kit, vì thế nên nhóm hạ thấp số class xuống còn 2 class : máy bay và xe cộ

Mô hình Cirfar nhóm thực hiện:



Hình 9. Sơ đồ thiết kế

**IV. HIỆN THỰC :**

**1. Khối chuyển đổi Int sang Float:**

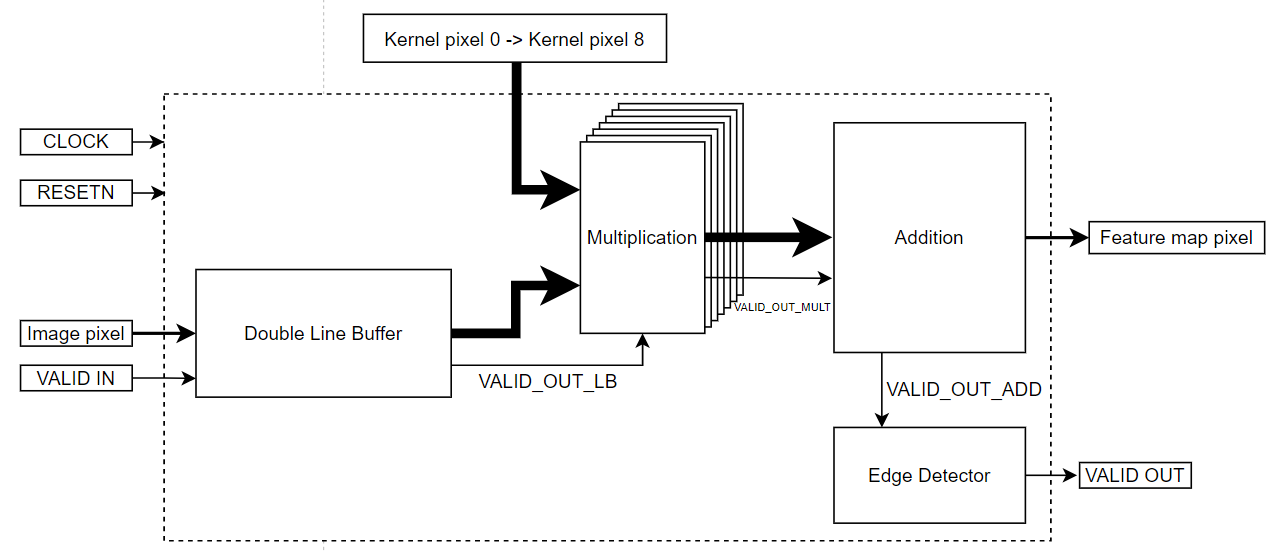
Các pixel ảnh có giá trị là một số nguyên 8-bit. Để có thể tính toán các giá trị khi bước vào mô hình CNN sử dụng kiểu dữ liệu Float 32-bit, các pixel ảnh cần được chuyển đổi sang định dạng tương ứng.

**2. Khối tích chập :**

Khối tích chập được thiết kế riêng cho việc nhân tích chập với kernel có kích cỡ là 3x3. Khối này sẽ trích xuất ma trận ở các vị trí khác nhau trong một bức ảnh đầu vào dưới dạng cửa sổ trượt, sử dụng double line buffer.

Các pixel ảnh sau khi được trích ở dạng ma trận sẽ qua một bộ nhân song song bao gồm 9 mạch nhân để thực hiện việc nhân với ma trận kernel. 9 kết quả nhân sau đó sẽ qua một bộ cộng 9 input để cho ra một pixel của feature map.

Tín hiệu VALID sẽ được truyền qua các bộ tính toán, sau đó qua bộ kiểm tra rìa (Edge Detector) để kiểm tra xem cửa sổ trượt có di chuyển vào những vị trí rìa hay không.



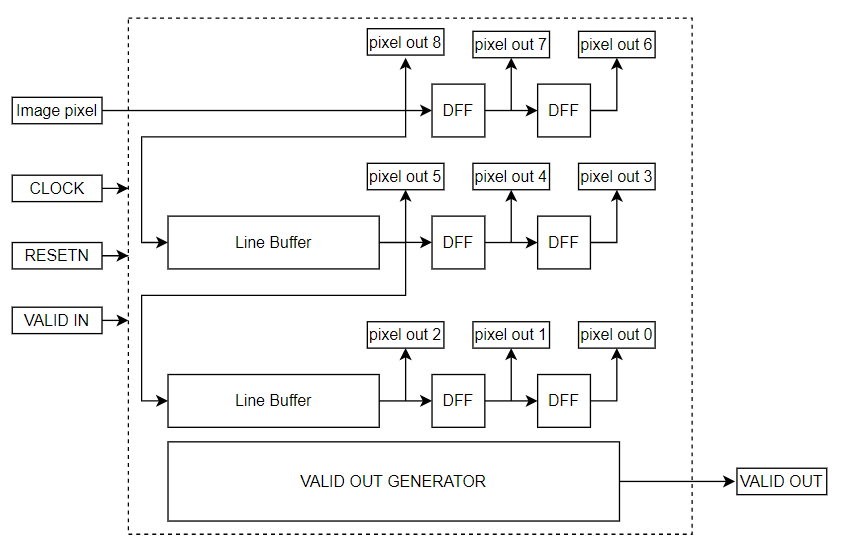
Hình 10. Khối Conolution

\*Sơ đồ chân hệ thống :

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin** | **Direction** | **No. bit** | **Description** |
| CLOCK | Input | 1 | Xung clock hệ thống |
| RESETN | Input | 1 | Tín hiệu reset hệ thống |
| Image pixel | Input | 32 | Pixel ảnh |
| VALID IN | Input | 1 | Valid in |
| Kernel pixel 0 ->  Kernel pixel 8 | Input | 32x9 | Pixel của kernel 3x3 được đẩy tuần tự vào hệ thống |
| Feature map pixel | Output | 32 | Pixel của feature map |
| VALID OUT | Output | 1 | Valid out |

**3. Khối Double Line Buffer :**

Khối Double Line Buffer được dùng để trích ma trận ảnh 3x3 có kiến trúc bao gồm 2 Line Buffer (FIFO) được dùng để lưu 2 line của ảnh. Cứ mỗi chu kì, khối Double Line Buffer sẽ trượt qua mỗi vị trí 3x3 trong bức ảnh đầu vào, lấy ra các pixel ảnh hợp lệ để thực hiện nhân tích chập.



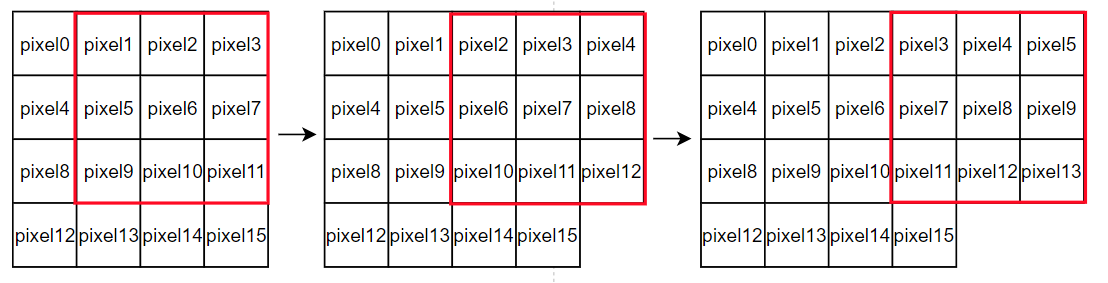
Hình 11. Khối Double Line Buffer

\*Sơ đồ chân hệ thống :

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin** | **Direction** | **No. bit** | **Description** |
| CLOCK | Input | 1 | Xung clock hệ thống |
| RESETN | Input | 1 | Tín hiệu reset hệ thống |
| Image pixel | Input | 32 | Pixel ảnh |
| VALID IN | Input | 1 | Valid in |
| pixel out | Output | 32x9 | Ma trận ảnh 3x3 |
| VALID OUT | Output | 1 | Valid out |

**4. Khối Edge Detector:**

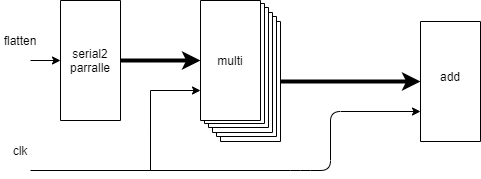
Khối Edge Detector được tạo ra để dò vị trí rìa trong bức ảnh khi cửa sổ trượt đến. Khối này là cần thiết vì nếu không, các giá trị rìa sẽ được đem vào làm pixel của feature map, làm cho feature map đầu ra bị nhiễu nặng.



Hình 12. Cửa sổ trượt đến rìa.

**5. khối Fully connect**

Khối fully connect được tạo ra để tổng hợp các node từ flatten ra output ( giảm lượng perceptron ). Khối này có thể có nhiều hidden layer tùy thuộc vào người tạo mô hình và layer cuối thường dùng activation function là softmax.



Hình 13. Khối fully connected