**Lớp CNN đầu tiên**

nn.Conv2d(in\_channels, 256, 9, stride=1, padding=1)

**Kernel size:** 9×9

**Padding:** 1 (thêm 1 pixel xung quanh mỗi biên)

**Stride:** 1 (di chuyển kernel 1 pixel mỗi bước)

**Số kênh đầu ra:** 256

Công thức tính chiều cao (Hout​) và chiều rộng (Wout ​): Hin

Win

Hout= = = 46

Wout= = = 892

Kích thước sau lớp **Conv2d** đầu tiên: (256, 46, 892)

**Lớp MaxPooling \**

**Kernel size:** 3×3

**Stride:** 3 (di chuyển kernel 3 pixel mỗi bước)

Công thức tính:

Hout= = = 15

Wout= = = 297

Kích thước sau lớp MaxPool2d: (256, 15, 297)

**Lớp CNN thứ hai**

nn.Conv2d(256, 256, (4, 3), stride=1, padding=1)

**Kernel size:** 4×3

**Padding:** 1

**Stride:** 1

Công thức:

Hout== = 12

Wout== = 295

Kích thước sau lớp Conv2d thứ hai: (256, 12, 295)

Trước khi chuyển đầu ra sang lớp nn.Linear, đầu ra cần được làm phẳng. Feature Dimensionđược tính như sau:

Feature Dimension = số kênh x Hout x Wout

Feature Dimension = 256 x 12 x 295 = 90240  
Vì vậy, kích thước đầu vào cho lớp nn.Linear phải là 90240, không phải 3584.

Tuy nhiên, chương trình vẫn hoặc động tốt với dòng lệnh :

self.linear = nn.Linear(3584, 256)

### Điều này có thể là do dữ liệu không bị sử dụng toàn bộ sau CNN

* Có khả năng, sau khi CNN xử lý, phần nào đó của đầu ra bị giảm kích thước (ví dụ, cắt bớt hoặc thay đổi cấu trúc tensor trước khi vào nn.Linear).
* Có thể, đang sử dụng một phần dữ liệu (như slicing) ở đâu đó trong quá trình xử lý trước khi đưa vào self.linear.