```
self.Conv1 = nn.Conv2d(3, 16, 3, padding=1)
self.Conv2 = nn.Conv2d (16, 32, 3, padding=1)
self.Conv3 = nn.Conv2d (32,64, 3, padding=1)
self.maxpool = nn.MaxPool2d(2, 2)
```

Kernel/filter size

Stride

We achieve a deep but with small width and height output.

regarding the size and stride in maxpool:

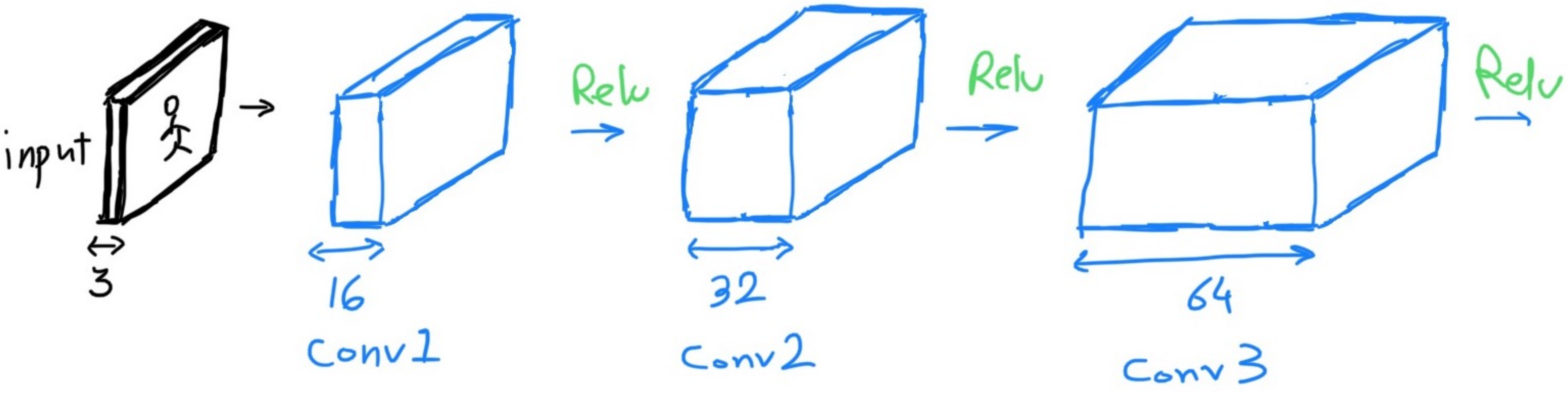
* if we want to see all pixels and down-sample an image by a factor of 4, then, MaxPool2d (4,4) is our best choice.

assume this is the final maxpool layer.

we can flatten it and give it to a fully-Connected layer to make a binary classification.

Note: as we go through more and more layers, the representation becomes more and more specific, i.e., feature-level.

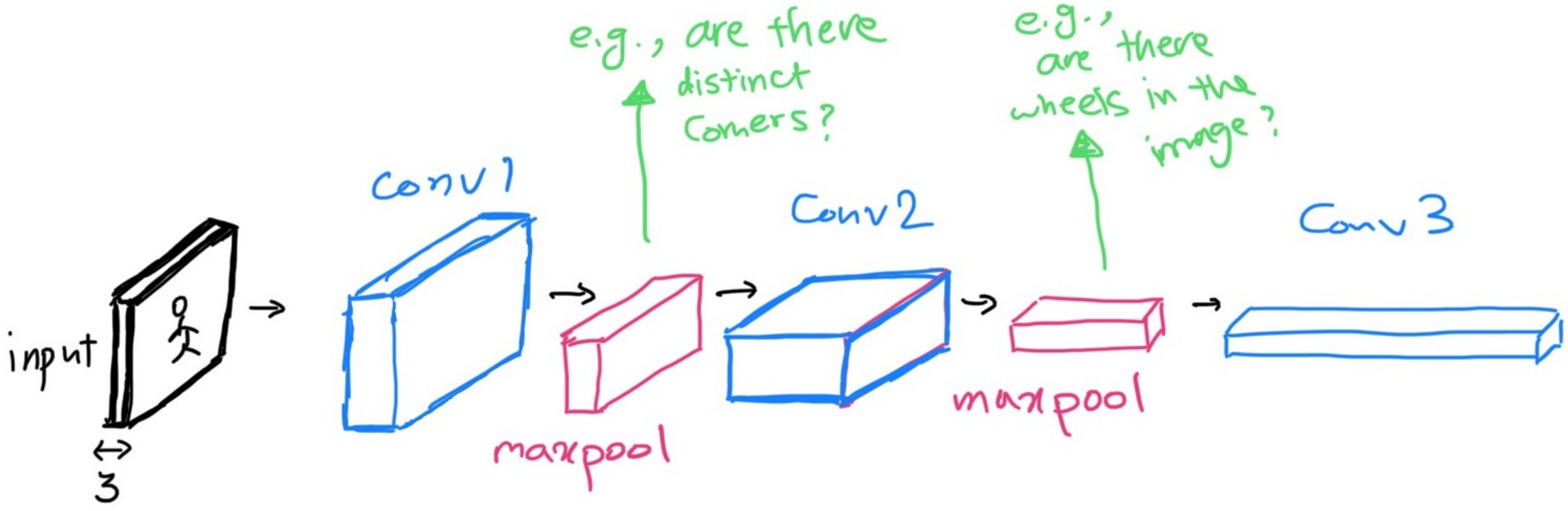
Overview of a full CNN model



```
self.Conv1 = nn.Conv2d(3, 16, 3, padding=1)
self. Conv2 = nn. Conv2d (16, 32, 3, padding=1)
self. Conv3 = nn. Conv2d (32,64, 3, padding=1)
```

When we pass the image from these Conv. layers, the width and height does not change but the depth increases. This results in too many params.

So, we use max pooling !


e.g., are there distinct corners?
e.g., are there wheels in the image?
input
3
Conv1
maxpool
Conv2
maxpool
Conv3