PadhAI: From Convolution Operation to Neural Network

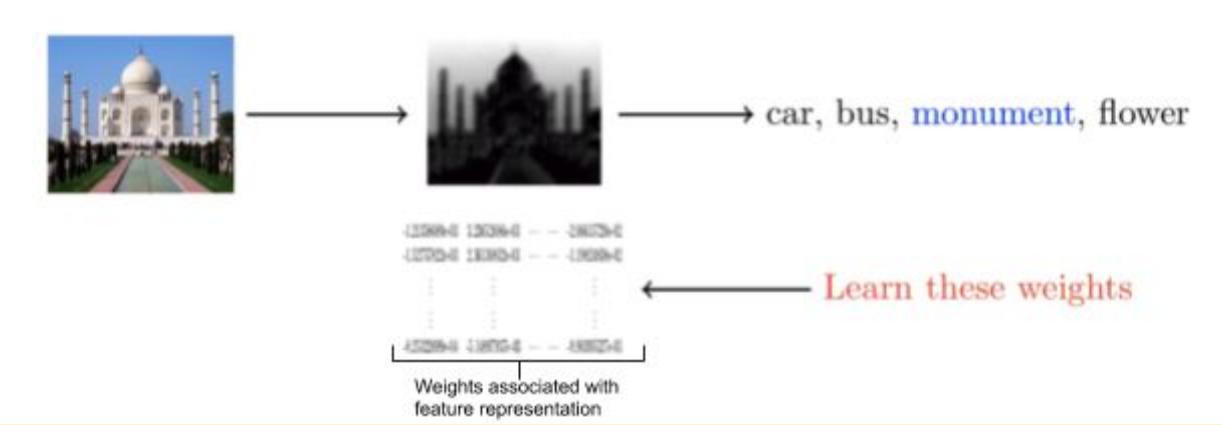
One Fourth Labs

The convolution operation and neural networks

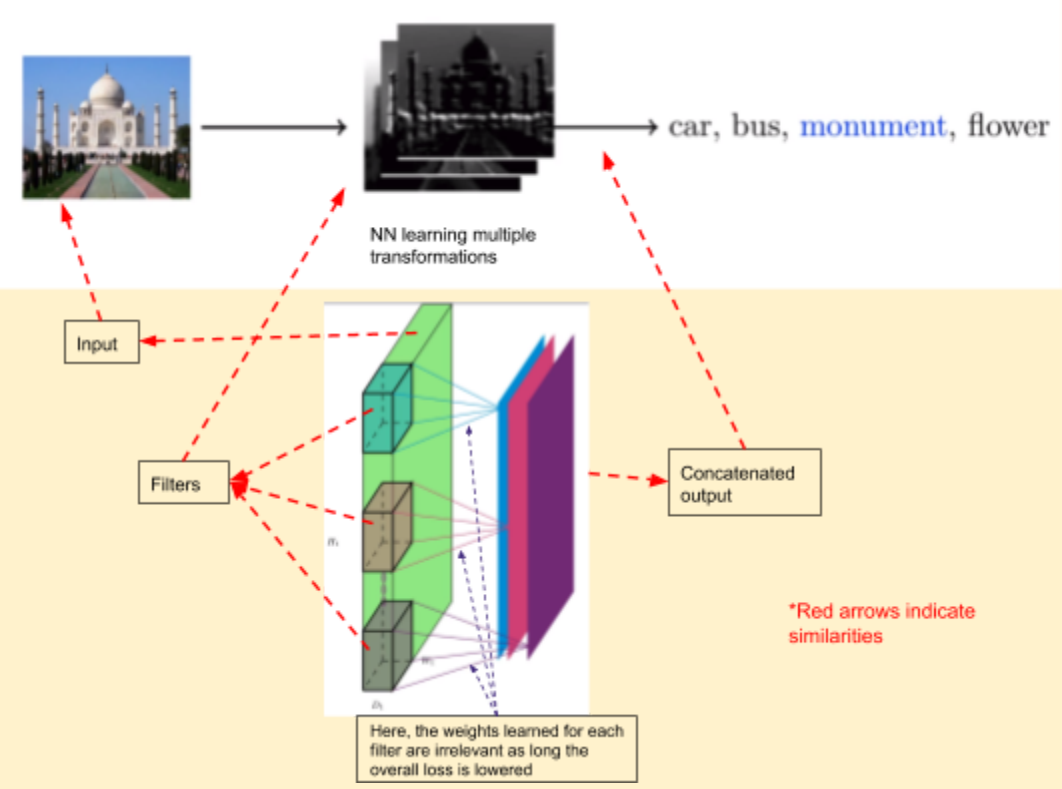
Part 3

Why not let the network learn the feature representation also?

1. So far, we have intuitively determined what kind of transformation to apply to the input image before passing it through the classifier. For eg: we saw that performing edge detection yielded higher accuracy.
2. Now, another thing to think about is, why must we leave the choice of transformation to our human intuition? Why can't we let the DNN learn for itself the best transformation to apply?



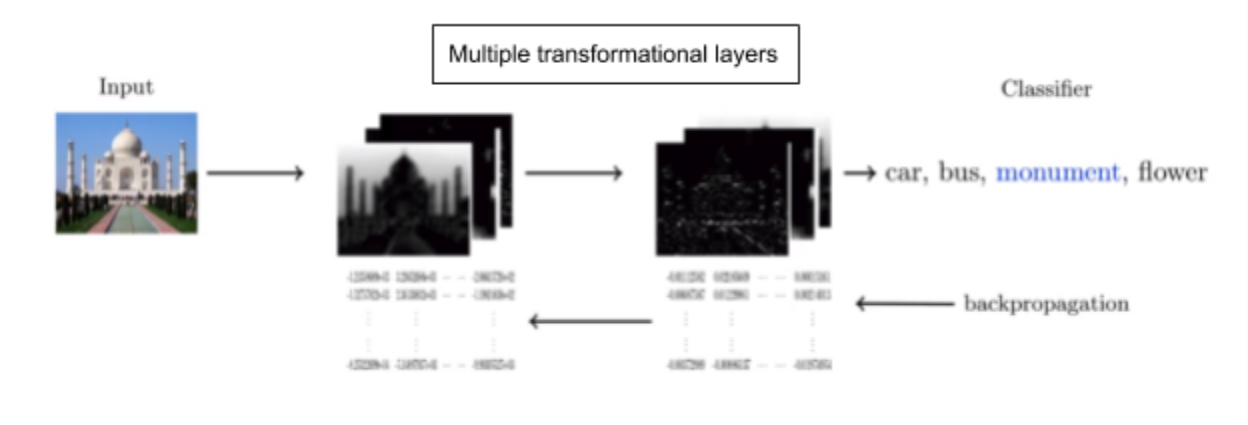
3. **Why not let the network learn the multiple feature representation**?
4. Another point to consider is that we might not necessarily benefit from only one transformation, so letting the DNN learn the number of transformations to perform is also very useful.



a. Here, we are not concerned with the type of transformation that occurs, so long as the overall loss/error is reduced. In the above image, we cannot say that the transformations are anything

definitive like edge-detection or blurring but still choose to accept them because they lower the loss.

5. **Why not let the network learn the multiple layers of feature representation**?
6. It stands to reason that we can consider several transformational layers like the one seen in the previous example.



   a. The above process is quite similar to what we have learned in the DNN. There are 3 sets of weights, one for each of the two representational layers and one for the classification layer.
   b. These weights are calculated using backpropagation
   c. The only difference between a DNN and what we're looking at now (CNN) is that for a CNN we only consider a small localised neighborhood of inputs when calculating the output, instead of the entire input layer as in the case of DNNs.
7. In a nutshell, for CNNs, instead of learning the final classifier weights directly, we should also learn to transform the input into a suitable representation through multiple layers of representations and learn the kernel weights for all of those representations instead of using hand-crafted kernels.