Table of Contents

1 Short Answer

1.1 Advantage of Convolutional Layers vs. Dense Layers for Feature Extraction in Image Processing

The architecture of a *convolutional layer* compared to a *dense layer* illustrates the advantage of the former in terms of feature extraction as shown in the 2D representation of a convolutional unit and its dense equivalent in Fig. 1.

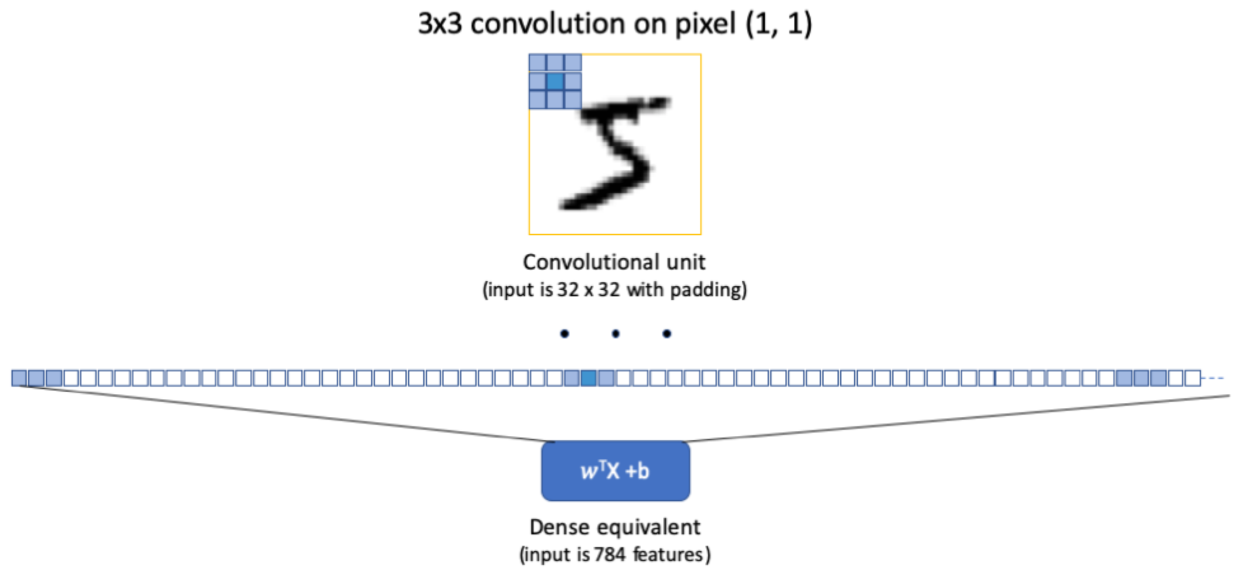

Fig. 1. 2D convolutional unit in the LeNet-5 image processing network and its dense layer equivalent [1].

Whereas dense layers are linear, convolutional layers are not and, in the instance above (or generally, in cases of image recognition/processing) nearby inputs are connected to nearby outputs via feature maps with each convolution. A dense layer is one in which "every input is connected to every output by a weight," whereas in a convolutional layer, "weights…for each location are share [and] due to weight sharing, and the use of a subset of weights of a dense layer, there are far less weights that in a dense layer [2]." As such, when evaluating the performance-size trade-off on two regularized (penalization, early stopping, dropout, data augmentation, and model size reduction) networks, the CNN was found to perform better than one with dense layers with only 1/3 of the number of coefficients [1]. Simply put, CNN layer can perform better, faster than a dense layer.

1.2 Benefits and Applications of a 1×1 Convolutional Layer

A 1×1 convolutional layer is a down-sample feature channel-wise pooling/sampling technique or a projection layer, which "acts like a coordinate-dependent transformation in the filter space [3]." It simply means that the filter is a size of 1×1 as opposed to a conventional CNN matrix (e.g. 3×3) which, "convolves over the *entire* input image pixel by pixel [4]. It can be used in the following applications [4]:

1. Dimensionality reduction or augmentation
2. Deeper networks using residual connections and a bottle-neck layer
3. Smaller, accurate models by stacking 'fire-module' layers, which comprise of a *squeeze layer* with only 1×1 convolutional filters and an *expansion layer* with a mix of 1×1 and 3×3 filters.
4. Reduce computational complexity by reducing the number of parameters
5. Introducing additional non-linearity to the network

1.3 Drawbacks of Accuracy/Performance of Deep*er* Networks

Intuitively, the deeper a neural network, the more information is lost and the increased complexity consequently makes, "deeper neural networks…more difficult to train [5]." According to the paper which proposes residual nets as an alternative to CNN's, "when deeper networks are able to start converging, *a degradation problem* has been exposed: with the network depth increasing, *accuracy gets saturated* (which might be unsurprising) and then *degrades rapidly*. Unexpectedly, such degradation is *not caused by overfitting*, and *adding more layers to a suitably deep model leads to higher training error* [5]." Furthermore, deeper networks also result in "parts of the problem that will remain untouched [6]."

1.4 Benefits of Odd Numbered Kernel Sizes in Convolutional Layers

The intention of a convolution is to encode an input data matrix (i.e. an image) as a filter or a kernel as shown in Fig. 2, where a source pixel can be defined as the "anchor point at which the kernel is centered and we are encoding all the neighboring pixels, *including* the anchor/source pixel." Odd-numbered kernel sizes account for the symmetry of kernels around the origin, whereas the origin is less obvious in asymmetric even-numbered filters, which may lead to aliasing errors [7].

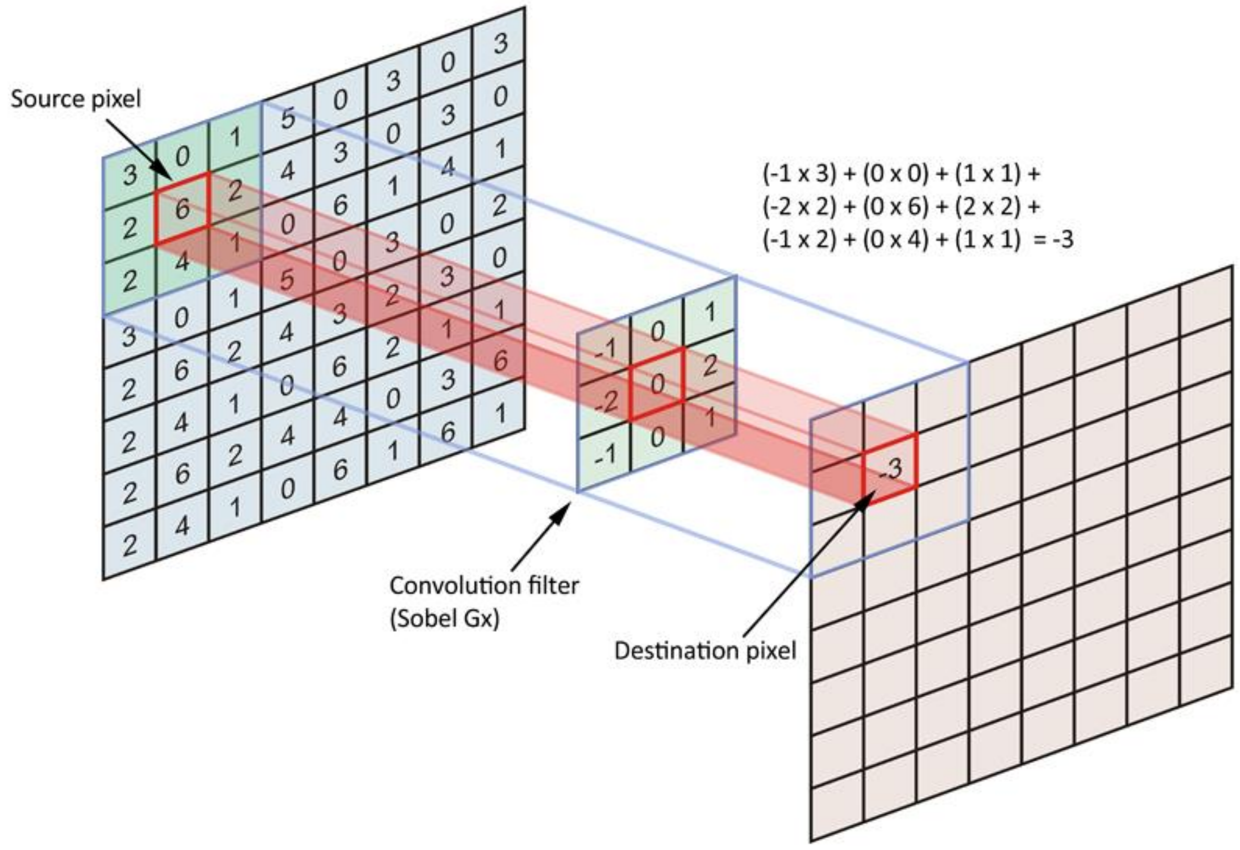
Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
Convolution filter
(Sobel Gx)
Destination pixel

Fig. 2 Visual illustration of pixel encoding, where the source pixel, convolutional filter, and destination pixel represent the input, kernel, and output, respectively.

1.5 Purpose of and Designing a Fully Connected Layer in Convolution Neural Networks

Fully connected layers are computationally expensive (and should be limited to combining the upper layer features). However, whereas a convolutional layer is limited by design to "local spatial coherence with a small receptive field," a fully connected layer "learns features from all the combinations of the features of the previous layer. [8] A fully connected layer best predicts the label to describe an image by takes the output of the convolution/pooling layers. A fully connected input layer flattens the output of the previous layers and turns it into a single vector and a fully connected output layer outputs the final probabilities of each label [9].

2 Applying a Kernel Filter to an Image

The code using the cv2 library below is used to apply a kernel filter to a random image, with the original and filtered images shown in Fig. 3 and 4, respectively.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

image = cv2.imread('image.jpg')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)


kernel = np.array([[-1,-1,-1], [-1, 9,-1], [-1,-1,-1]])
img = cv2.filter2D(image, -1, kernel)

fig1 = plt.figure(figsize=[12,8])
org = plt.imshow(image)

fig2 = plt.figure(figsize=[12,8])
filt = plt.imshow(img)

fig1.savefig('original.png', format = 'png')
fig2.savefig('filtered.png', format = 'png')

plt.show()
```

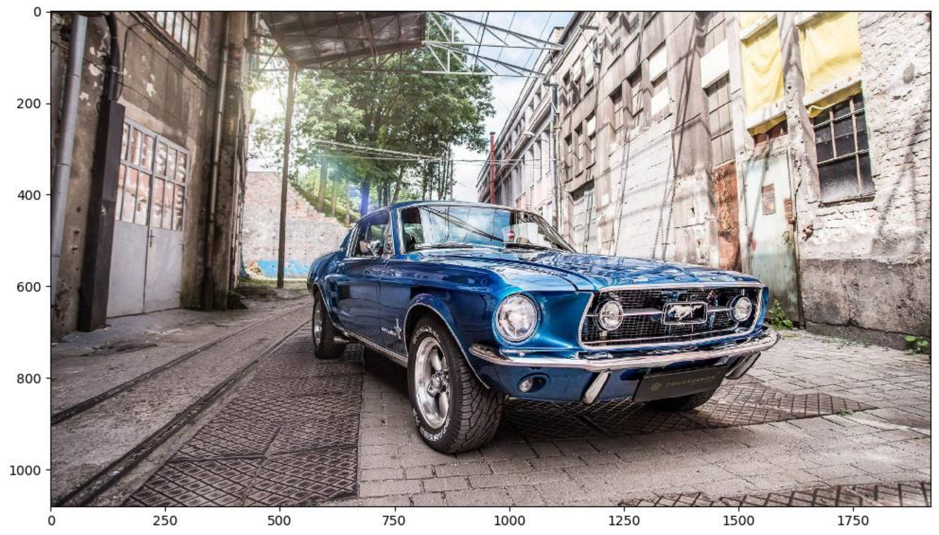

Fig. 3. Original image; imported original file saved as ‘original.png’.

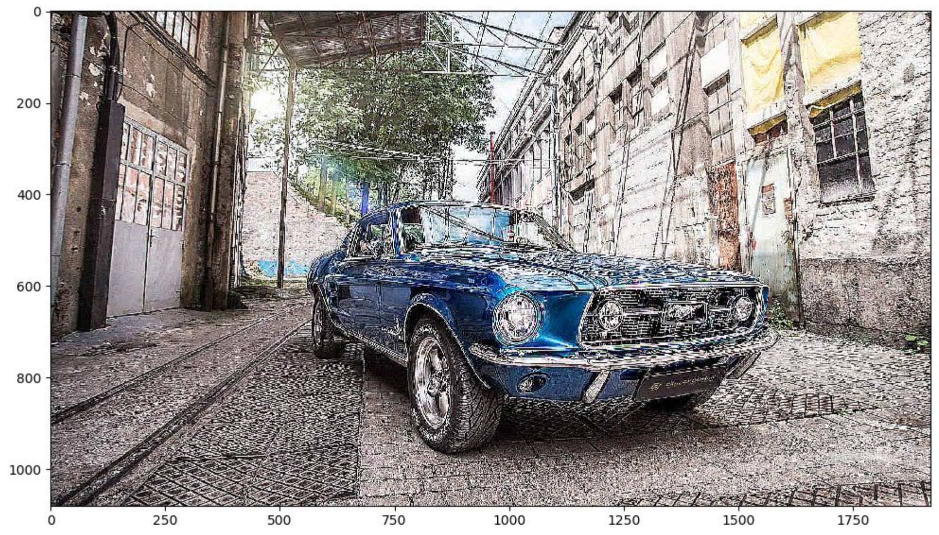

Fig. 4. Image with the kernel filter applied; filtered image file saved as ‘filtered.png’.

3 Classification Model for CIFAR-10 Photo Dataset in Keras

Please refer to 'ChE788_A4_Q4.ipynb'; the model was optimized for the following, where hyperparameter tuning was achieved with the *HParams Dashboard*:

:

1. Number of filters in each CNN layer (HP_NUM_FILTERS1, HP_NUM_FILTERS2, HP _NUM_FILTERS3)
2. Batch size (HP_BAT)
3. Batch Normalization (HP_BATNORM_MOMENT)
4. L2 regularization (HP_L2)
5. Epochs (HP_EPOC)
6. Dropout (HP_DROPOUT)
7. Optimization algorithm (HP_OPTIMIZER)

The Google Colab file may be accessed at the link below and the optimized values can be visualized on TensorBoard:
https://colab.research.google.com/drive/1ES4sniJhVwoecTHC7muHZFqcPgLFBdlD?usp=sharing

A References

[1] A. Hue, "Dense or Convolutional Neural Network", Medium, 2020. [Online]. Available: https://medium.com/analytics-vidhya/dense-or-convolutional-part-1-c75c59c5b4ad.

[2] [T. Elliot and J. Howard, "Dense vs convolutional vs fully connected layers", Deep Learning Course Forums, 2016. [Online]. Available: https://forums.fast.ai/t/dense-vs-convolutional-vs-fully-connected-layers/191/2.

[3] A. Prakash, "One by One [1 x 1] Convolution - counter-intuitively useful", Iamaaditya.github.io, 2020. [Online]. Available: https://iamaaditya.github.io/2016/03/one-by-one-convolution/.

[4] R. Sakthi, "Talented Mr. 1X1: Comprehensive look at 1X1 Convolution in Deep Learning", Medium, 2020. [Online]. Available: https://medium.com/analytics-vidhya/talented-mr-1x1-comprehensive-look-at-1x1-convolution-in-deep-learning-f6b355825578.

[5] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *arXiv:1512.03385v1 [cs.CV]*, 2015.

[6] P. Canuma, "Wide Residual Nets: “Why deeper isn’t always better…”", Medium, 2020. [Online]. Available: https://medium.com/@prince.canuma/wide-residual-nets-why-deeper-isnt-always-better-f2a02947dca3.

[7] "Why convolutions always use odd-numbers as filter_size", Data Science Stack Exchange, 2017. [Online]. Available: https://datascience.stackexchange.com/questions/23183/why-convolutions-always-use-odd-numbers-as-filter-size.

[8] J. Despois, "What are the advantages and disadvantages of fully connected layers in convolutional neural network? - Quora", Quora.com, 2018. [Online]. Available:

https://www.quora.com/What-are-the-advantages-and-disadvantages-of-fully-connected-layers-in-convolutional-neural-network#:~:text=Advantages%3A%20A%20fully%20connected%20layer,layers%20are%20incredibly%20computationally%20expensive.

[9] "Fully Connected Layers in Convolutional Neural Networks: The Complete Guide - MissingLink.ai", MissingLink.ai. [Online]. Available: https://missinglink.ai/guides/convolutional-neural-networks/fully-connected-layers-convolutional-neural-networks-complete-guide/#:~:text=Fully%20connected%20layers%20are%20an,features%2C%20and%20analyzing%20them%20independently.