zj722, 02215217

1. Classification

I explored various CNN design choices, including activation functions, pooling, convolutional layers, and filter configurations.
For activation function, ReLU and Leaky ReLU outperform Sigmoid and Tanh, with minor differences between them.
For pooling layer, pooling parameters (kernel size and stride) have little effect, so I standardize ReLU and 2×2 pooling for all models.
The number of filters and convolution layers strongly impacts accuracy and loss: too few cause underfitting, while too many cause overfitting. 128 filters generally perform well, while 256 filters may lead to diminishing returns.
By fixing a moderate filter size (e.g., 128) and gradually increasing the number of convolutional layers (1, 2, then 4), I find that four convolutional layers strike a balance between performance and overfitting risk, as loss begins to increase when beyond this point(4-layers).
To optimize filter arrangement in a 4-layer CNN, I tested:
(1) *Uniform* (e.g., $128 \rightarrow 128 \rightarrow 128 \rightarrow 128$),
(2) *Progressively doubling* from 32 to 256,
(3) *Progressively halving* from 256 down to 32.
The doubling strategy (32 to 256) provides the best balance of accuracy, loss, and generalization, minimizing the train-validation gap. This strategy enhances feature extraction efficiency while avoiding redundancy. Conversely, halving filters (256 to 32) was tested to assess whether critical features emerge earlier or later, but it produced similar results with a little worse generality. The sketch of model structure are shown in Figure 1.

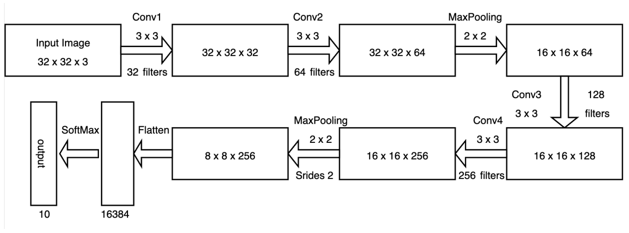

Figure 1. best CNN architecture for classificationa and regression

The comparison between each model are shown in figure 2. The gap between training and validation result are primarily due to overfitting, the model becomes increasingly tailored to the training data, reducing its generalization ability.

2. Regression

After the previous evaluation, the CNN model with progressively doubling filters in each convolutional layer achieved good accuracy. When applied previous model to this regression problem, it directly resulted in a validation error below 75% for all four inputs. Specific result are shown in table 1

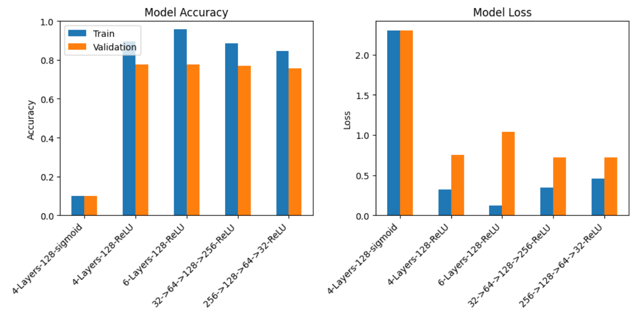

Figure 2. model comparison among different CNN structure

Picture	Estimation error (%)
frontal	66.69
kitchen	58.13
bedroom	67.88
bathroom	67.53

Table 1. error rate of regression result based on different input

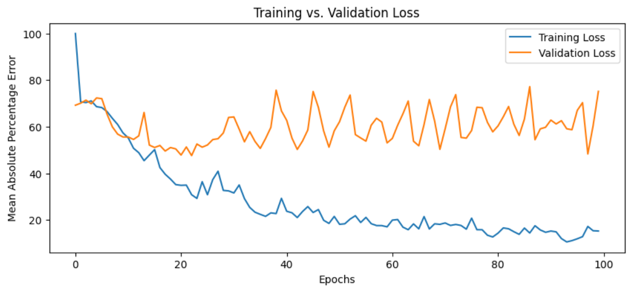

Figure 3. validation and train loss vs epochs 4-layers

However, the performance of this model is not fixed. As training epochs increase, the validation-training error gap widens, indicating overfitting (Figure 3). Less layers models leads to lower overfitting due to lower complexity shwon in(Figure 4).

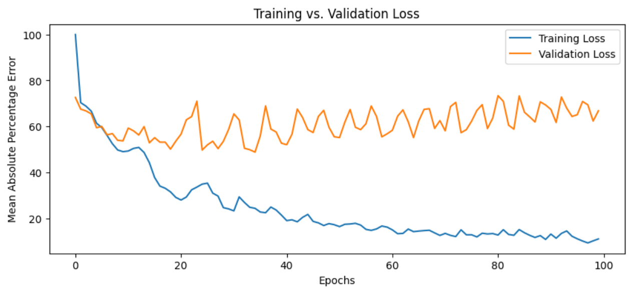

Figure 4. validation and train loss vs epochs 2-layers

However, the gap for these two model are similar, and more layers can capture finer details, enhancing adaptability to diverse inputs like front views of houses and kitchens, etc. makes it to a more effective solution for this problem.