Lab 4 Bonus Report

王恩泽　郑云天

December 23, 2025

0 Overall Structure

Using the Netron visualizer, Figure (a) illustrates the architecture and parameter distribution of the SI100FaceNet model. The network primarily consists of three convolutional blocks (`conv1`, `conv2`, `conv3`) followed by a fully connected layer (`fc`). This structure is designed to progressively extract hierarchical features from input images for sentiment classification.

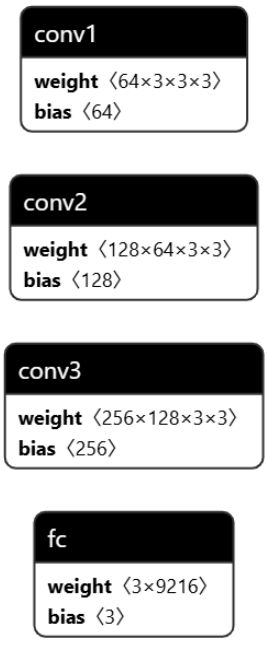

Figure 1: Netron visualization of SI100FaceNet model architecture

1 Conv1

The weight tensor $[64 \times 3 \times 3 \times 3]$ represents the parameters of the first convolutional layer:

- 64: Specifies the number of filters (output channels). Each filter learns to detect primary features, such as edges and basic textures.
- $[3 \times 3 \times 3]$: Defines the dimensions of each filter. The first dimension (3) corresponds to the RGB input channels. The subsequent two values denote the 3×3 spatial dimensions of the kernel. Thus, each filter is a three-dimensional tensor.

Bias [64]: These are the learnable bias terms for the 64 filters. Each filter has a unique bias value that is added to the output of the convolution operation.

2 Conv2

The weight tensor for the second layer is $[128 \times 64 \times 3 \times 3]$:

- 128: The filter count increases to 128, allowing the model to learn a more diverse set of complex feature combinations.
- $[64 \times 3 \times 3]$: The input depth (64) matches the number of output channels from `conv1`. Each of the 128 filters is a $64 \times 3 \times 3$ tensor.

Bias [128]: These are the bias values corresponding to the 128 output channels.

3 Conv3

The weight tensor for the third layer is $[256 \times 128 \times 3 \times 3]$:

- 256: The output channels increase to 256. At this depth, the network captures high-level, abstract semantic features, such as specific patterns of facial components.

- $[128 \times 3 \times 3]$: The input depth matches the previous layer's output of 128, while the kernel size remains 3×3.

Bias [256]: These are the bias values corresponding to the 256 output channels.

4 fc

Following the convolutional blocks, the final stage of the network is a fully connected (fc) layer, which performs the classification.

Weight $[3 \times 9216]$:

- **3**: This represents the final output dimensionality, which corresponds to the number of sentiment categories. In this project, the categories are Happy, Neutral, and Sad.
- **9216**: This is the number of input features, representing the flattened one-dimensional vector derived from the final convolutional feature maps.

  Calculation: This value is the product of the `conv3` output channels and the spatial dimensions ($H \times W$) after downsampling. Each convolutional layer is followed by a LeakyReLU activation function and a 2×2 max-pooling layer with a stride of 2, which progressively halves the size of the feature maps. Given an input image size of 48×48 and three pooling operations, the spatial dimensions are reduced to 6×6. Consequently, the flattened length is $256 \times 6 \times 6 = 9216$.

Bias [3]: These are the bias values corresponding to the 3 output classes.

5 Summary

The Netron visualization in Figure (a) provides a clear overview of the SI100FaceNet parameter structure:

- The number of channels increases sequentially ($64 \rightarrow 128 \rightarrow 256$), reflecting the increasing complexity and abstraction of features as the data flows deeper into the network.
- All kernels maintain a 3×3 size—a standard design choice that balances the receptive field with computational efficiency.
- Through successive convolution and pooling operations, the network compresses raw pixel data into high-level representations, which are ultimately mapped to a 3-dimensional classification space via the `fc` layer.

By utilizing Netron for architectural visualization, we have successfully mapped the theoretical design of SI100FaceNet to its practical parameter configurations, providing a clear and comprehensive understanding of how each block contributes to the final classification task.