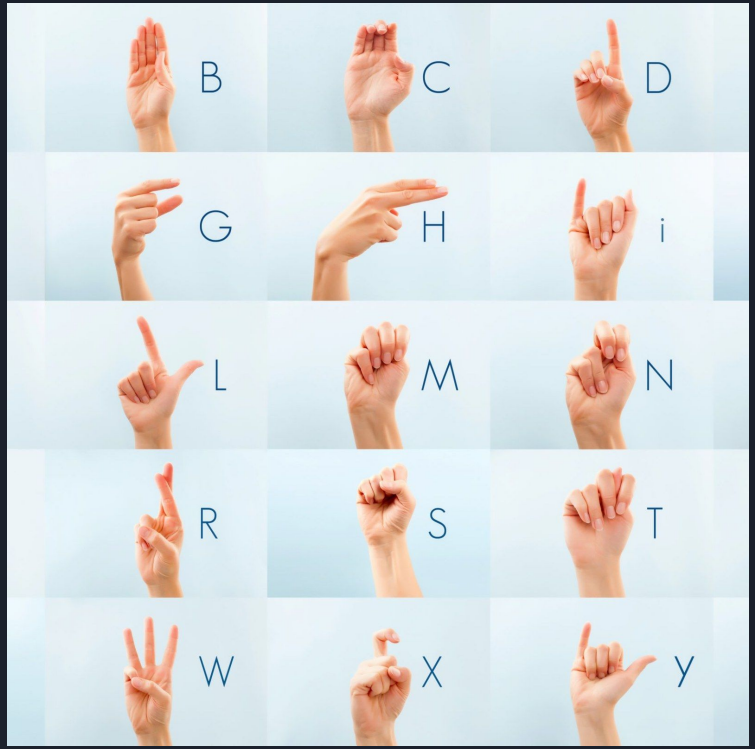
Convolutional Neural Network for ASL Detection

Aaryan Patel and Peter Morand

American Sign Language (ASL)

ASL

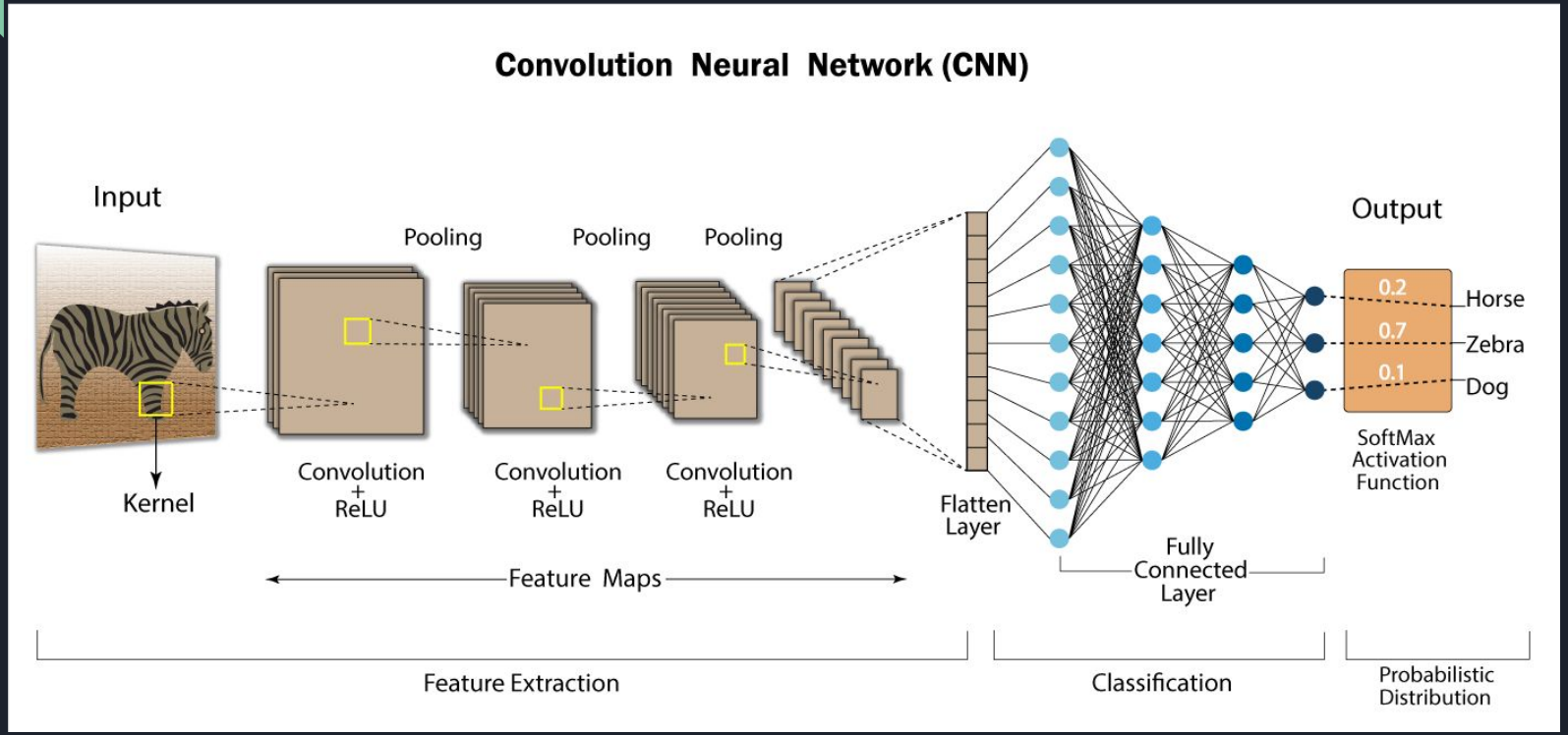
- ASL is a form of communication for deaf people using hand symbols.
- Each letter and number can be expressed by a unique symbol which has key distinctions from other characters.



https://www.newsweek.com/asl-day-2019-american-sign-language-1394695

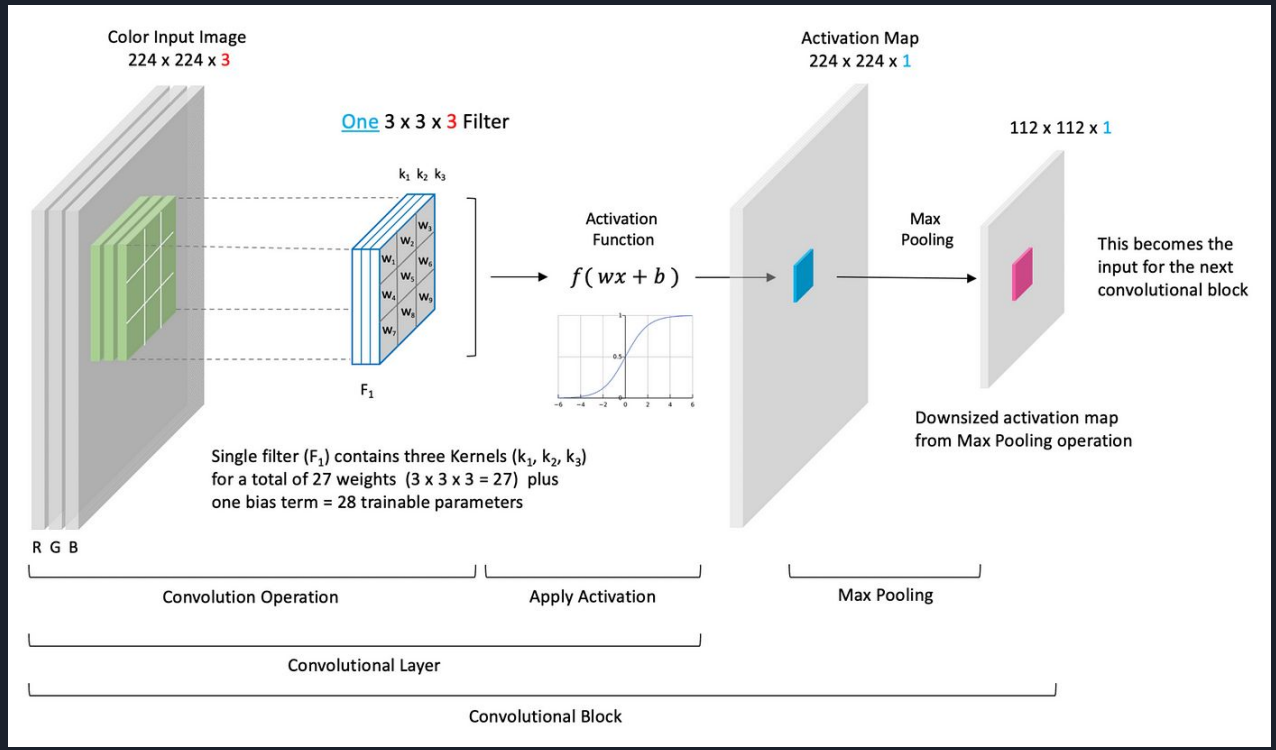
Convolutional Neural Network (CNN)

CNN Structure



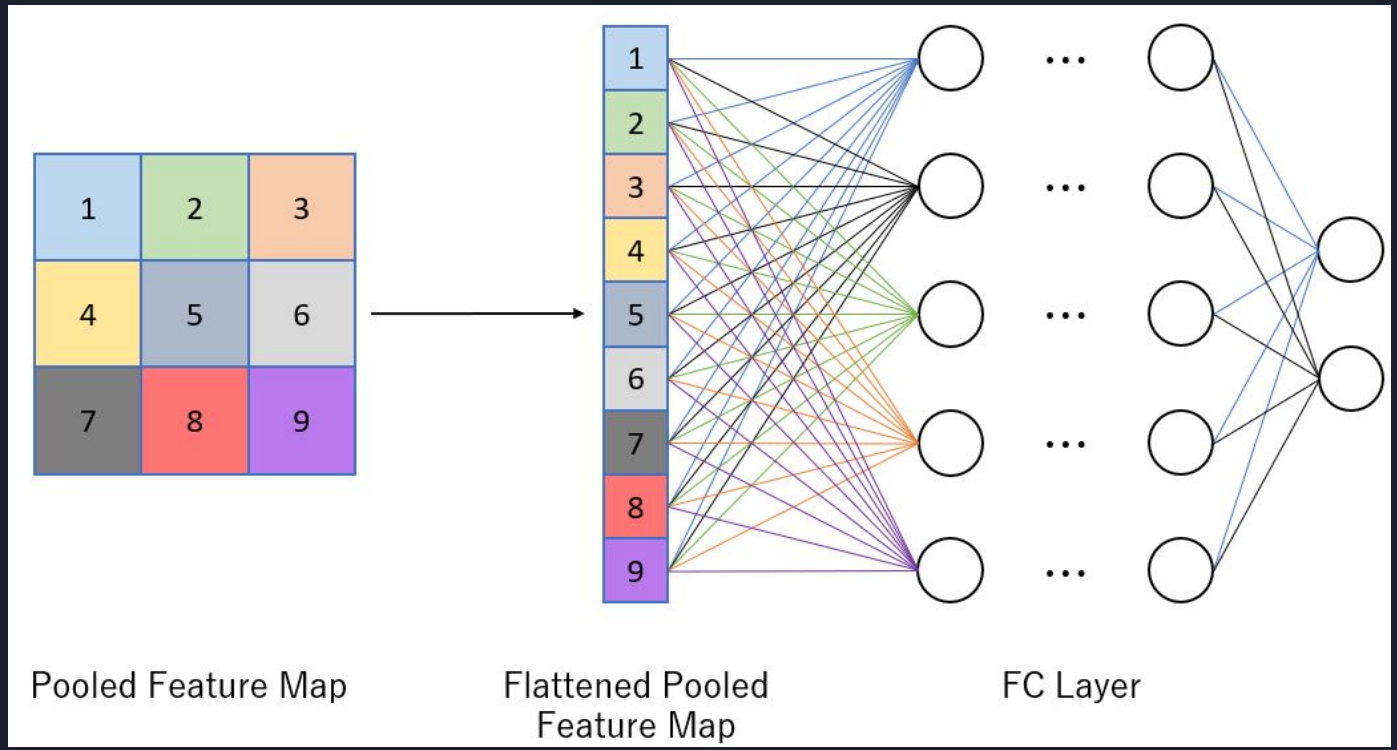
https://www.linkedin.com/pulse/what-convolutional-neural-network-cnn-deep-learning-nafiz-shahriar

Convolution and Max Pooling



https://learnopencv.com/understanding-convolutional-neural-networks-cnn/

Fully Connected layer



https://medium.com/@alejandro.itoaramendia/convolutional-neural-networks-cnns-a-complete-guide-a803534a1930

Our Model's Structure

```
# Define the loss function and network layers

loss_function = CrossEntropyLossFunction()

network_layers = [
    # Input (Image Shape): 1 (Grayscale Channel) * 64 * 64. Convolution using a kernel of size 3 to create an output with depth 16. Output: 16 * 62 * 62.
    Convolution(input_shape=(1, 64, 64), output_depth=16, kernel_size=3),
    ReLU(),
    # Max Pooling will divide shape by the stride. Output of Max Pooling: 16 * 31 * 31.
    MaxPooling(pool_size=2, stride=2),

    # Input: 16 * 31 * 31. Often, each Convolution Layer doubles the output_depth. Output: 32 * 29 * 29
    Convolution(input_shape=(16, 31, 31), output_depth=32, kernel_size=3),
    ReLU(),
    # Output: 32 * 14 * 14.
    MaxPooling(pool_size=2, stride=2),

    # Input: 32 * 14 * 14. Output: 64 * 12 * 12.
    Convolution(input_shape=(32, 14, 14), output_depth=64, kernel_size=3),
    ReLU(),
    # Output: 64 * 6 * 6.
    MaxPooling(pool_size=2, stride=2),

    # Flattening multidimensional array to 1D. Input: 64 * 6 * 6.
    Flatten(),

    FullyConnected(input_size=64 * 6 * 6, output_size=256),
    ReLU(),

    # Input: 256. Output: 36 (Number of total labels).
    FullyConnected(input_size=256, output_size=y_data.shape[1]),
    Softmax()
]
```

Results

Comparison of Models

Image Shape	Epochs	~Accuracy
28x28	20	88%
64x64	20	93%

Thanks! Questions?