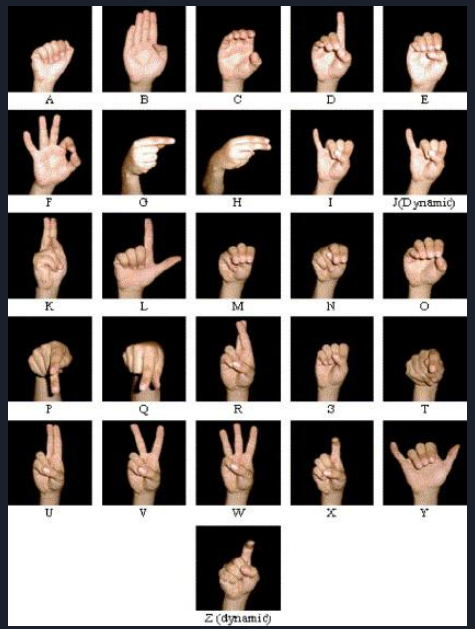
# AMERICAN SIGN LANGUAGE IMAGE CLASSIFICATION

ISHITA JAIN
JYOTHI GANJI
MIHIR GADGIL

# American Sign Language

- ASL is the natural language that serves as predominant sign language for hearing-impaired communities in US and most of Canada.
- ASL is most closely related to French Sign Language
- Originated in early 19th century in American School of the Deaf in West Hartford, Connecticut.
- Lots of study has been done on sign recognition in last few years, namely HMM (Hidden Markov Model), ANN (Artificial Neural Network) and SVM (Support Vector Machines)



# Dataset

- https://www.kaggle.com/datamunge/sign-language-mnist
- 27,455 samples in the training set, with features as pixel values
- Each image sample is 28x28 pixels
- Target is composed of 24 letters, excluding J and Z as they involve motion
- Each letter is encoded as a number from 0 to 25 (excluding 9 and 25)
- Fairly balanced classes, with minority class E with 957 samples and majority class R with 1294 samples
- Added two more images per class, using our own hands
- Test set size : 7,172 samples

# Initial Steps

- Initially worked with Keras
- Custom build architectures were tried which performs well
- But fails with test data and then, tried with pre-trained model “ResNet”

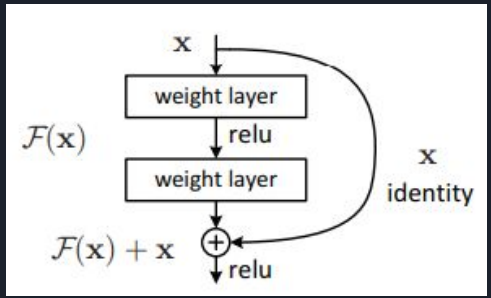
# Model

- ResNet 18
- Pre-trained on ImageNet (2012) dataset, with 1000 classes
- Last layer changed to 26 neurons linear layer
- Further trained on our dataset

# ResNet 18 Architecture

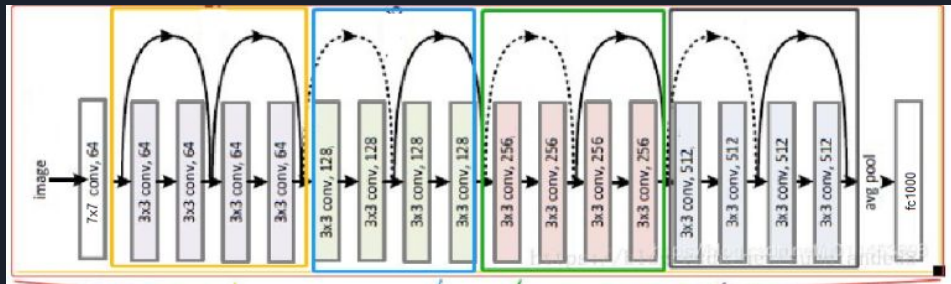
Residual Learning

- Hypothesis: it is easier to approximate the residual function (F(x) := H(x) - x) than to learn the original function (H(x))
- Learning by feedforward neural networks with “shortcut connections” by skipping one or more layers
- Shortcut connections help speed up the learning process




Residual Learning: a building block
(Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015)

# ResNet 18 Architecture (contd.)




| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|---|---|---|---|---|---|---|
| conv1 | 112×112 | 7×7, 64, stride 2 | | | | |
| | | 3×3 max pool, stride 2 | | | | |
| conv2_x | 56×56 | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$ |
| conv3_x | 28×28 | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$ |
| conv4_x | 14×14 | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$ |
| conv5_x | 7×7 | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$ |
| | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | $1.8\times10^9$ | $3.6\times10^9$ | $3.8\times10^9$ | $7.6\times10^9$ | $11.3\times10^9$ |

(Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015)

# Why ResNet18?

- Tried ResNet18, GoogleNet and DenseNet
- DenseNet runs out of VRAM
- GoogleNet similar results to ResNet
- But longer time to train

# Preprocessing

1. Grayscale to RGB: Expected by model
2. Resize: 28 x 28 to 224 x 224
3. Horizontal Flip: Simulate right and left hand
4. Rotate: Variations in hand orientation
5. Colour Jitter: Lighting conditions, different skin colours
6. Normalization: Expected by model

# Results

- Custom-built architectures
  - Decent performance on non augmented training data
  - Poor performance on test and augmented data
- ResNet 18 model performance on both training and testing data:
  - Validation loss (Cross Entropy): ~$10^{-4}$
  - Accuracy: 100%
  - F1-Score: 1.0
  - Cohen’s Kappa: 1.0

# References


1. https://www.kaggle.com/datamunge/sign-language-mnist.
2. Sahoo, Ashok K., Gouri Sankar Mishra, and Kiran Kumar Ravulakollu. "Sign language recognition: state of the art." ARPN Journal of Engineering and Applied Sciences 9.2 (2014): 116-134.
3. Vogler, Christian, and Dimitris Metaxas. "Parallel hidden markov models for american sign language recognition." Proceedings of the Seventh IEEE International Conference on Computer Vision. Vol. 1. IEEE, 1999.
4. Nikam, Ashish S., and Aarti G. Ambekar. "Sign language recognition using image based hand gesture recognition techniques." 2016 Online International Conference on Green Engineering and Technologies (IC-GET). IEEE, 2016.
5. He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.