Sign Language to Text Conversion

Deep Learning for Robot Vision (DLRV)

February 27, 2022

Alex Jude, Elanton Fernandes

Sign Language to Text Conversion

- **Input**: Image containing a hand making the shape of an ASL letter.
- **Output**: ASL letter

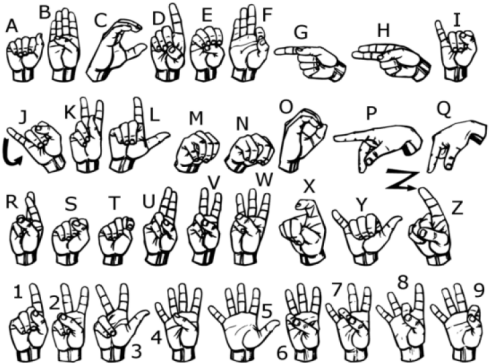

Figure 1: ASL Hand gestures [1]

[1] Masood, Sarfaraz Thuwal, Harish Srivastava, Adhyan. (2018). American Sign Language Character Recognition Using Convolution Neural Network.

Objective

- A comparative study of the effectiveness of various deep learning architectures on the use case of sign language conversion to text.
- **Use case**: Sign Language or hand gestures requires an interpreter at every instance.
  - Conversation between hearing impaired people (the deaf/mute) and the non-hearing impaired people.

Dataset

- The dataset consists of 1400 images which are 720x720 pixels. There are 26 classes for the letters A-Z.

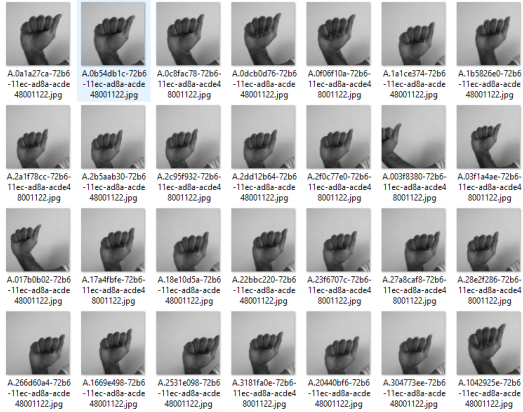

Figure 2: Dataset structure

Data pre-processing

- Image resizing
- Data normalization
- Conversion to grayscale

Models used for study

- CNN based architecture [2]
- Fine tuning pre-trained VGG16 model [3]

[2] Dabwan , Basel. (2020). Convolutional Neural Network based Sign Language Translation System.

[3] Masood, Sarfaraz Thuwal, Harish Srivastava, Adhyan. (2018). American Sign Language Character Recognition Using Convolution Neural Network.

CNN based architecture

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 128)       2176

 max_pooling2d (MaxPooling2D  (None, 14, 14, 128)      0
 )

 conv2d_1 (Conv2D)           (None, 14, 14, 64)        131136

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 64)         0
 2D)

 conv2d_2 (Conv2D)           (None, 7, 7, 32)          32800

 max_pooling2d_2 (MaxPooling  (None, 3, 3, 32)         0
 2D)

 conv2d_3 (Conv2D)           (None, 3, 3, 16)          8208

 max_pooling2d_3 (MaxPooling  (None, 1, 1, 16)         0
 2D)

 flatten (Flatten)           (None, 16)                0

 dense (Dense)               (None, 512)               8704

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 26)                13338

=================================================================
Total params: 196,362
Trainable params: 196,362
Non-trainable params: 0
```

Figure 3: CNN based architecture

VGG16 model

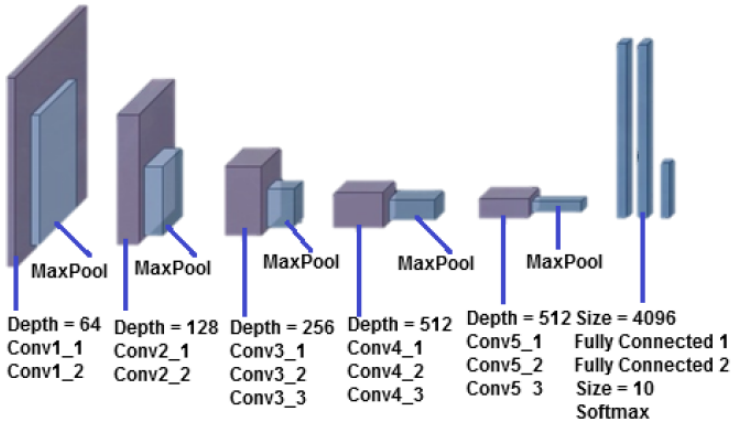

Figure 4: VGG16 Architecture.[3]

Evaluation: CNN based architecture

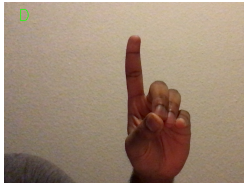

Figure 5: True value: D

Figure 6: True value: M

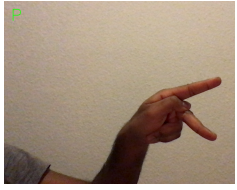

Figure 7: True value: P

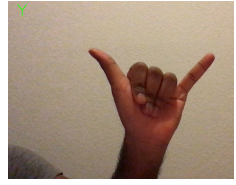

Figure 8: True value: Y

Evaluation: VGG16

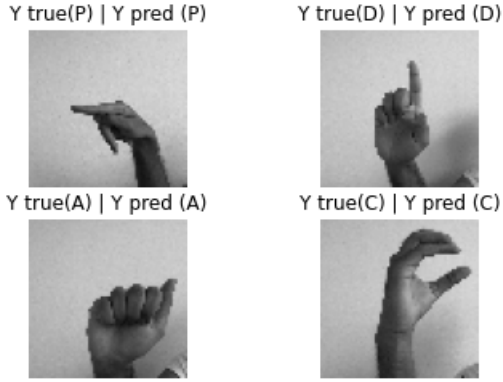

Figure 9: Example predictions from VGG16

Performance Comparison

- Test accuracy of CNN based architecture and VGG16 on our dataset:

CNN based model	VGG16
97.3%	98.6%

Table 1: Test accuracy

Lessons Learnt

- Live prediction depends on the background of the image.
- Creating a dataset is challenging.
- Performance of VGG16 model is slightly better than CNN based model.
- Since the hand gestures for some characters in ASL such as "M", "S", "E", "N" and "T" were similar, we observed that the models were biased in predicting "M" and "S".

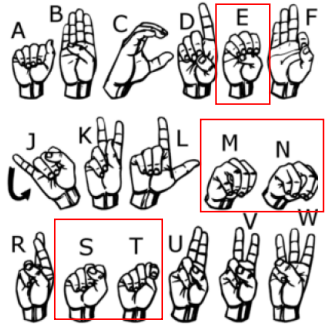

Figure 10: ASL hand gestures