MAIS 202 Final Project Deliverable 2: Sign Language Translator

Gohar Saqib Fazal

1. **Problem Statement**
   Use a convoluted neural network model to implement an Indo-pakistani alphabet sign language translator. The neural network takes in the input image of the user i.e. a frame from the live video feed and then displays the alphabet that his/her hand sign corresponds to on the screen.

2. **Data Preprocessing**
   In the last deliverable I proposed a plan to take pictures for the training set of the model manually but on receival of feedback from the executive I realized that the dataset was too small. Therefore, upon a second look I found another dataset online. The dataset is small as well but is comparatively better than the previous method suggested for building the dataset manually. This new dataset includes over 2000 images where 9 non-expert subjects executed 37 Urdu alphabet signs along with over 700 images where 9 non-expert subjects executed 12 Urdu words. The dataset is available online at the following link: https://www.kaggle.com/saadbutt321/pakistan-sign-language-dataset
   The dataset consists of JSON files. Each JSON file contains human body key points that are estimated by Open Pose and each data point in the dataset is annotated with an Urdu label. If the dataset is insufficient for any reason the option of scraping online videos is always available.

3. **Machine Learning model**
   The model that I used is VGG-16 Convoluted Neural Network (CNN). The main advantage of using this model lies in its ability to learn features as well as weights corresponding to each feature. The CNN aims to optimize the objective function/loss function [1]. Each image needs to be scaled to 224 x 224 resolution as the model takes an input image of fixed size [2].

$$Loss = \frac{1}{N}\sum_{i=1}^{N} -\log\left(\frac{e^{f_{i,y_i}}}{\sum_{j=1}^{C} e^{f_{i,j}}}\right) \quad (1)$$

$$f_j(z) = \frac{e^{z_j}}{\sum_{k=1}^{C} e^{z_k}} \quad (2)$$

N = total number of training examples
C = total number of classes

Fig 1. The SoftMax-based loss function that is maximized [1]

4. **Preliminary results**

   There are no preliminary results to display right now but are expected to be available towards the end of the next deliverable. This delay in results is to allow for me to get more accustomed with OpenCV, TensorFlow and Keras.

5. **Next steps**

   There is a lot to be done for the completion of the project from this point forward. The model still needs to be trained and then the test dataset used to see the accuracy of the dataset. Once that is done the model needs to be integrated in a flask app that uses live video feed and then displays the results using real time data. Any other feedback related to the choice of the new dataset is going to be appreciated so that I may be able to change my approach to the video scraping while the project is in its preliminary stages.

Reference:

[1] *Cs231n.stanford.edu*, 2019. [Online]. Available: http://cs231n.stanford.edu/reports/2016/pdfs/214_Report.pdf. [Accessed: 21- Oct- 2019].

[2] "VGG16 - Convolutional Network for Classification and Detection", Neurohive.io, 2019. [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/. [Accessed: 21- Oct- 2019].