Hand Gesture Recognition

Aman Kumar (201911018) & Manan Gajjar (201911027)

Dhirubhai Ambani Institute of Information and Communication Technology

Prof. Pankaj Kumar

December 24, 2020

Overview

Introduction

- Hand Gestures and computer vision
- 3D model based approaches
- using skeletal models
- appearance based model

- touch-less, fast communication
- capable of carrying out enriched information
- robot control, media playback, household appliances and other aspects

Problem Statement

We are trying to recognize different gestures through live video feed. This requires..

- Extract the hand region - Remove unnecessary portions from each frame.
- Classify extracted foreground

Thus entire problem can be divided into two sub problems

- Foreground extraction
- Gesture prediction

Dataset Used

- We have defined 12 gestures for the experiment [Fig. 1]
- Training images for each of these categories have been gathered using webcam
- To make things easy, we have used images with uncluttered background
- Dataset contains 1000 images of training for each class and 100 images of testing for each class.
- Images are resized to 80×80.

Gestures

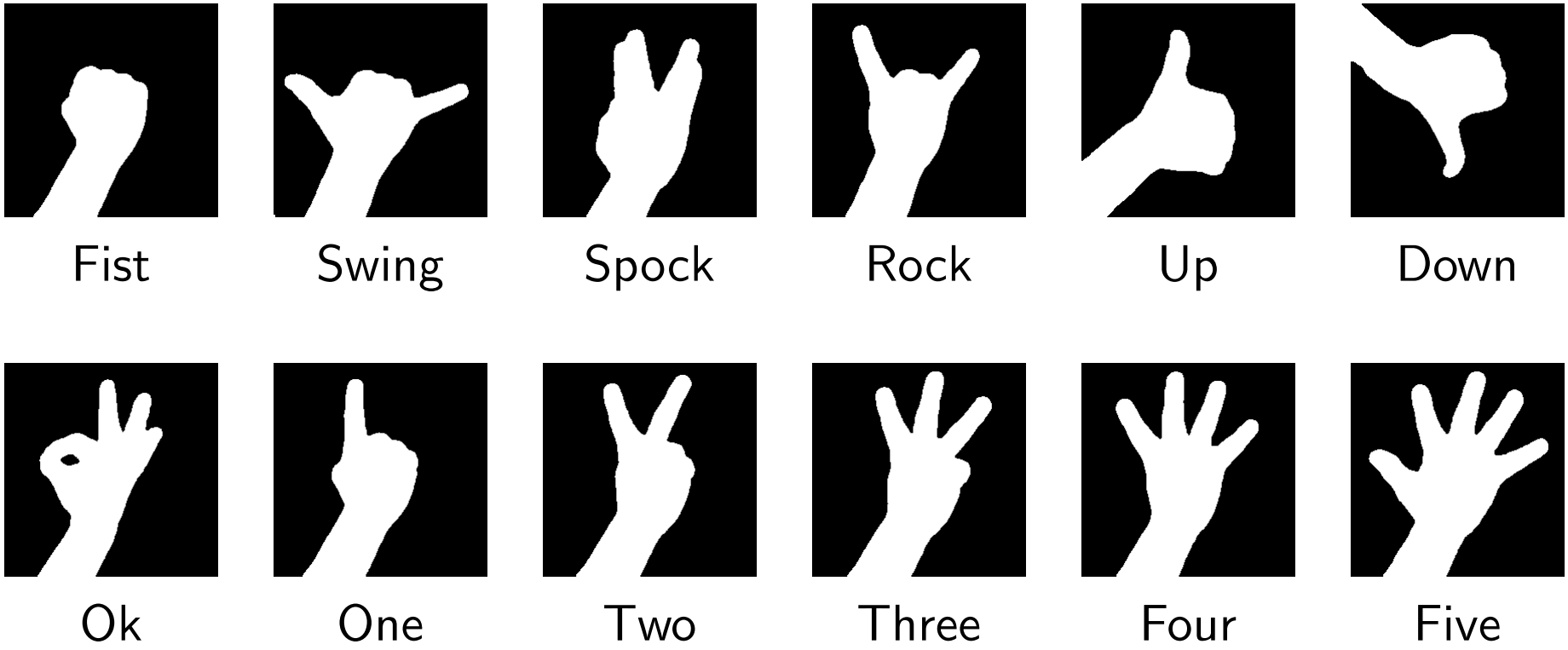

Figure: 1. Different Gestures

I. Foreground Extraction

First step is to get the Hand region from the frame.

- We marked top right corner in the live feed as region of interest.

Now to extract the hand gesture information and remove unnecessary background information

- We are using concept of running average.
- First few seconds each frame will be used to update the running average. Final value will be set as background.

$$dst(x, y) = (1 - a) \cdot \mathrm{dst}(x, y) + a \cdot \mathrm{src}(x, y) \tag{1}$$

- For all next frames from video feed, will be take absolute difference with this background. Which will give information about new foreground object.

II. Thresholding and Contour detection

Since We are interested in the shape of the gesture, we will threshold the foreground object.

- We found threshold of 25 was giving better results
- If the pixel intensity is less than the 25, we set it to 0 otherwise 255.

$$x(n) = \begin{cases} 255, & \text{if } n >= \textit{threshold} \\ 0, & \text{if } n < \textit{threshold} \end{cases} \tag{2}$$

Next we find contours in the difference image. Largest of which is most likely to contain full gesture information. We are using findContours from OpenCV for this functionality.

Gesture Recognition using Convolutional Neural Network

Gesture type will be predicted from trained CNN model. Since the image is already binary, we do not need a segmentation subnet here.
Configuration of the CNN model is as following.

- We are using a sequential model [Fig 2]
- We added Dropout layes since, custom dataset is small in size, and model was suffering from overfitting
- We added two of pooling layers to get only significant details of the gesture and remove less important ones.
- We kept 12 nodes in the output layer with softmax as activation function as this is a multi-class classification problem.

Network Architecture

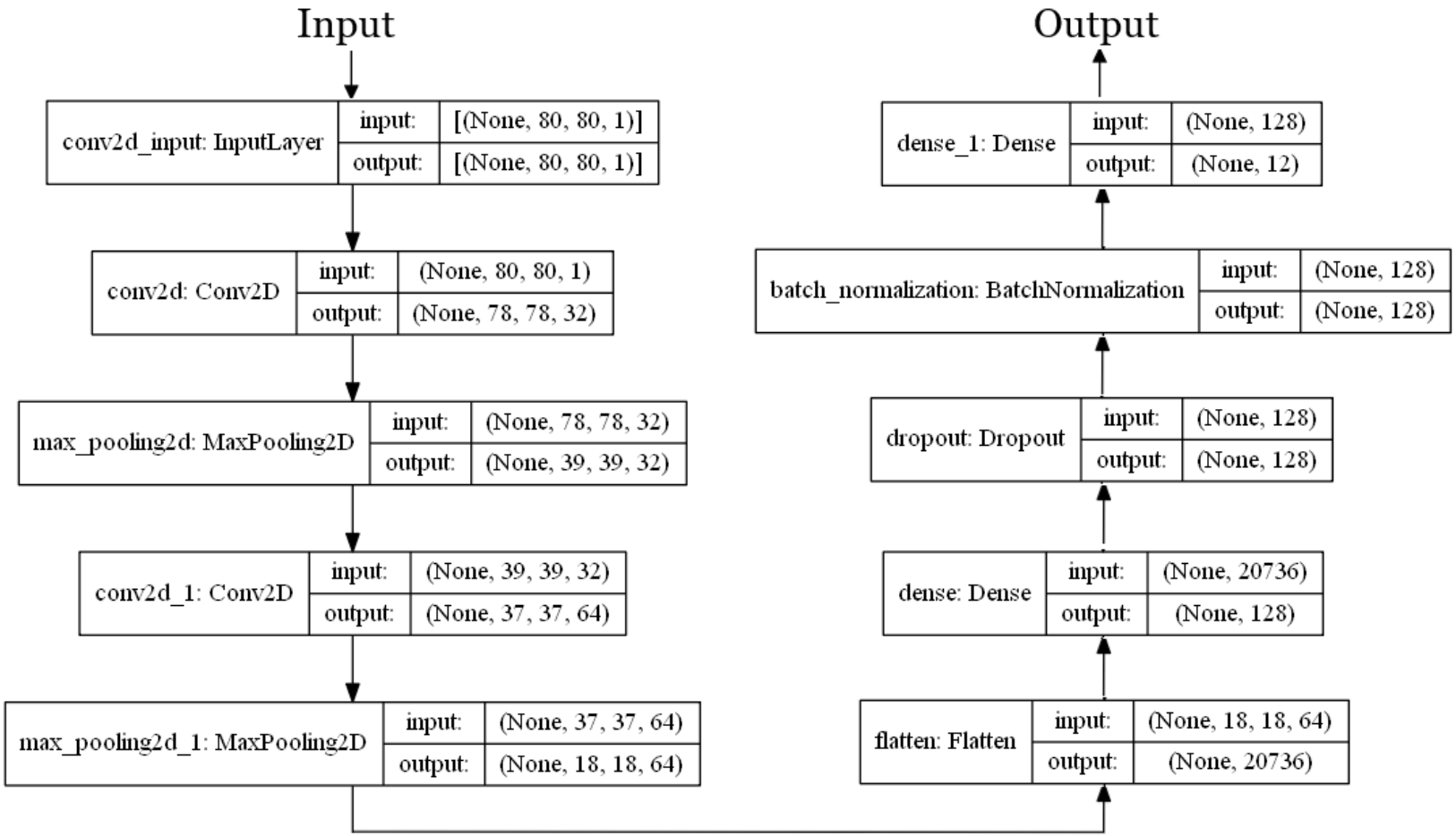

Figure: 2. Model Architecture

Results and Observations

- The validation accuracy and validation loss for testing images from dataset are shown in [Fig.3]. The model achieves **0.82** accuracy on validation dataset. Some samples from prediction with confidence value are shown in [Fig.4].

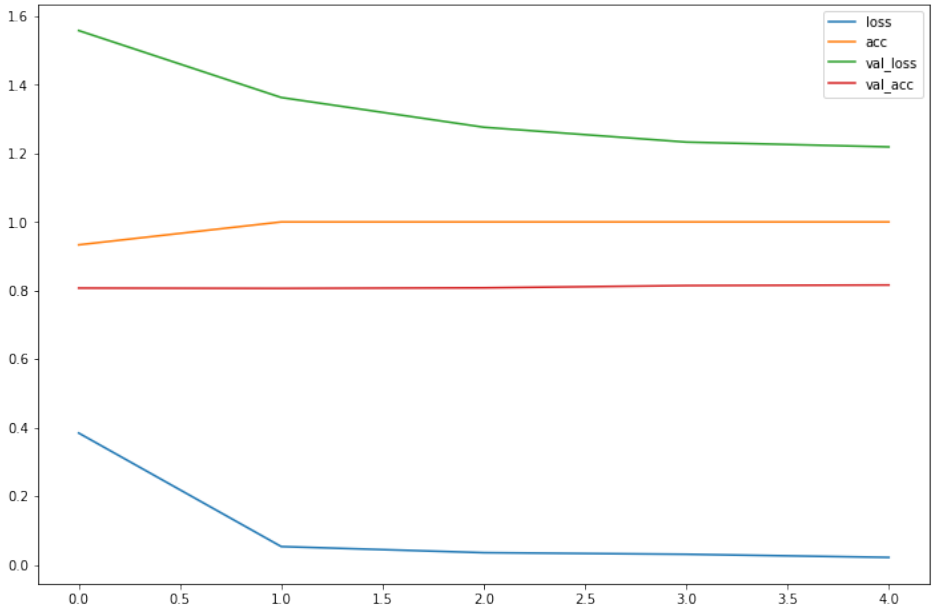

Figure: 3. accuracy-loss vs epochs

Results and Observations

Figure: 4. predictions with confidence value

References

OpenCV

3.0.0-dev documentation - accumulated weight.

https://docs.opencv.org/3.0-beta/modules/imgproc/doc/motion_analysis_and_object_tracking.html

OpenCV

OpenCV 3.4.13-dev documentation - finding contours.

https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html#ga17ed9f5d79ae97bd4c7cf18403e1689a

OpenCV

OpenCV 3.4.13-dev documentation - background subtraction methods.

https://docs.opencv.org/3.4/d1/dc5/tutorial_background_subtraction.html

TensorFlow

Core v2.4.0.

https://www.tensorflow.org/api_docs/python/tf

Thank You