# GESTURE RECOGNITION ASSIGNMENT

**Problem Statement**

Imagine you are working as a data scientist at a home electronics company that manufactures state of the art smart televisions. You want to develop a cool feature in the smart-TV that can recognise five different gestures performed by the user which will help users control the TV without using a remote.

The gestures are continuously monitored by the webcam mounted on the TV. Each gesture corresponds to a specific command:

Thumbs up:  Increase the volume

Thumbs down: Decrease the volume

Left swipe: 'Jump' backwards 10 seconds

Right swipe: 'Jump' forward 10 seconds

Stop: Pause the movie

Three Experiments were conducted to choose the optimal model, which are as follows-

1. **MODEL 1 – CNN (CONV3D) WITH DENSE LAYER**

* **Activation function – ELU**

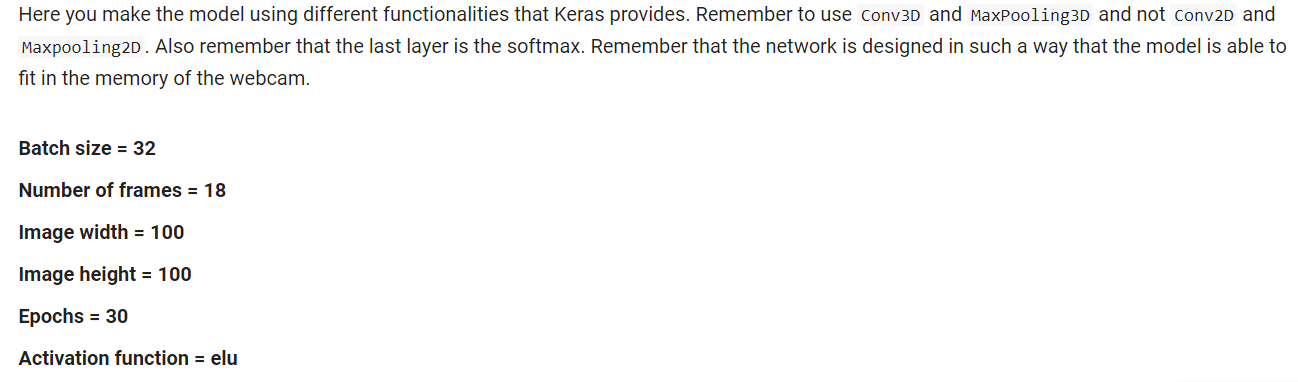


Image was resized to (100,100) and normalization was applied. Instead of RELU, the activation function used was ELU. Exponential Linear Unit or its widely known name ELU is a function that tend to converge cost to zero faster and produce more accurate results. As compared to RELU, ELU has an extra alpha constant which should be positive number. Optimiser used is SGD. Dropout of 50% is used between the final ReLu and Softmax layer.

**Inference: Training accuracy = 77.80% and validation accuracy = 76%. Model doesn’t overfit much. Further, we need to improve the accuracy.**

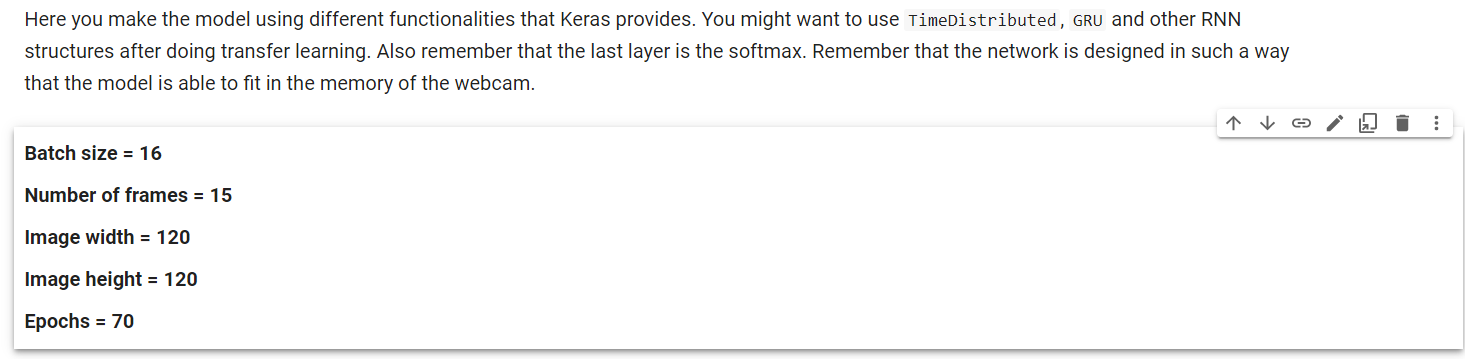
* **Activation function – RELU, different image dimensions and epochs.**



Image dimensions were increased to (120,120) and normalized. To try out different activation functions, this time RELU was used. It avoids and rectifies vanishing gradient problem. RELU is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations. Epoch was also reduced from 30 to 20. Dropouts of 25% added in between CNN layers. Optimiser used is SGD.

**Inference: Training accuracy = 53.70% and validation accuracy = 69%. Model is now underfitting. Our base model with eLu activation function was better than this model.**

1. **MODEL 2 – CNN (CONV3D) WITH RNN AND TIME DISTRIBUTED LAYER**



The batch size is reduced from 32 to 16 and the epochs increased to 70. The base model is created using VGG16 architecture. Two GRU layers are used and dropout of 50% is incorporated.

**Inference: Training accuracy = 82.17 % and validation accuracy = 65%. Model is overfitting.**

1. **MODEL-3: CNN(CONV3D) WITH TRANSFER LEARNING USING EFFICIENTNETB0**



The base model created uses the EfficientNetB0 architecture takes the input (120,120,3). Time distributed layer is still used here with the base model.

**Inference: Training accuracy = 99.70% and validation accuracy = 97%. This will be the final model as it has the highest accuracy and doesn’t overfit either.**