Neural Networks Project - Gesture Recognition

Project By: Karan Kajrolkar and Viraj Kadam

Problem Statements:

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

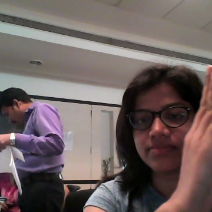
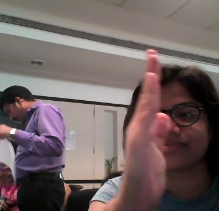
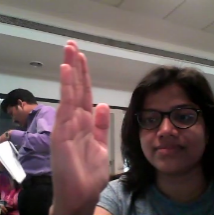
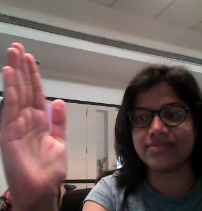
The gestures are continuously monitored by the webcam mounted on the TV. Each gesture corresponds to a specific command:

* 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

Each video is a sequence of 30 frames (or images).

Understanding the Dataset:

The training data consists of a few hundred videos categorized into one of the five classes. Each video (typically 2-3 seconds long) is divided into a sequence of 30 frames (images). These videos have been recorded by various people performing one of the five gestures in front of a webcam - similar to what the smart TV will use.



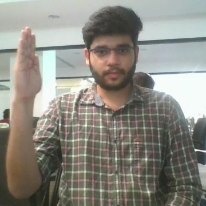
Right Swap:



Thumb down:



Thumb Up



Left Swap

**Generator:**

* This is one of the most important part of the code. Generators are functions that return traversable objects. They produce items one at a time and only when required.
* In Generator function we provided parameters are folder path, path\_doc, Batch Size, No of Samples of frame, Height, Width of image. We are using two dimensional images in (360 x 360 and 120 x 160) for preprocessing in generator.
* Generator should be able to take a batch of videos frames and applying augmentation are cropping, normalizing, rotation.

**Data Pre-processing:**

* Resizing and cropping of the images:
  + Here we are only focuses on gestures having high intensity pixels rather than noise. Here we cropping image only for top and bottom instead of right and left because we have image of swipes.
* Normalization of the images:
  + Normalizing the RGB channel image by dividing 255.
* Rotation: Here we apply slight rotation on images.
  + For more variant images to learn.

**Architecture Development and Training:**

* **On 1st model of Con3D**
  + we used batch size of 64 because it help us to train model faster and reduce computational power.
  + We used 18 frames per folder of gesture with dimensions of 120 x 120 and instead of focusing on whole image we crop in [:, 20:140 , :] size.
  + We model with less memory size (parameter). Also applied ReduceLRONPlateau which help in decreasing the learning rates if monitored metrics (Val \_loss) remains unchanged in between epoch.
  + Also applied SGD () optimizer for not to lose near negative values with lr=0.001.An accuracy we found that in model is getting under fitting because poor accuracy in train and validation.
* **On 2nd model of Conv3D**
  + we change optimizer to Adam with lr=0.002 help in increasing accuracy by rectifying high variance in the model’s parameters and add augmentation of cropping and rotation and reducing batch size to 32 and frame sample size to 20 .And we observed an accuracy is increasing with stable validation accuracy near 33%.
  + Here again we see overffting.so now **in model 3** we increases parameters size with dropout to 0.5 to handle over fitting.
  + you can see an result in following table

Now we used Batch Normalization, pooling and dropout layers when our model started to over fit, this help us to achieve good accuracy

Early stopping used to put a halt at training process when the val\_loss would start to model performance would stop improving.

* At last in we used transfer learnings model for CNN and GRU for RNN because GRU can be a better choice than an LSTM since it has lesser number of gates (and thus parameters).
* Transfer learning boosted the overall accuracy of the model.
* We choose mobileNET due to light weight design and high speed performance coupled with low maintenance as compared to other models.
* With MobileNET + GRU we uses Adam optimizer with learning rate of 0.002,batch size of 64 and only 15 frame from each video with dimensions 120 X 120.
* We try train by both trainable and non-trainable mobile NET weights we found interesting factor that with trainable=False give training accuracy up to 99% and validation accuracy to 93%,

**Model Overview:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model Name | Model Type | | | Number Of Parameter | | Augmented Data | Highest Training  Accuracy | | Highest validation  Accuracy | | Decision + Experiments |
| con3D\_2\_model | Conv3D | | | 698,533 | | No [Normalization] | 0.3970 | | 0.2833 | | Here we seen poor validation accuracy and training accuracy with custom cropping size and batch size of 30 with less parameter  - We observed model are over fitting |
| con3D\_3\_model | Conv3D | | | 698,533 | | Yes  [ Cropping as intensity, Rotation, Normalization] | 0.6561 | | 0.3625 | | - Here we get better accuracy than previous model after applying affine transformation [Crop, Rotation]  - We observed that accuracy increasing after each epoch continuously with stable validation accuracy.  - Model are completely over fitting.  - So now we are trying denser CNN with more parameters and for over fitting we apply dropout increases to 50%. |
| Model Name | Model Type | | | Number Of Parameter | | Augmented Data | Highest Training  Accuracy | | Highest validation  Accuracy | | Observations |
| Con3D\_4\_model | Conv3D | | | 1,966,309 | | Yes  [ Cropping as intensity, Rotation, Normalization] | 0.6644 | | 0.23 | | -Still overfitting found in model  -So we move toward Transfer Learning model to achieve good accuracy model. |
| Transfer Learning Models | | | | | | | | | | | |
| MobileNET+GRU  (trainable – False) | | CNN-RNN | Trainable prams: 265,477  Non-trainable prams: 2,260,544 | | Yes  [ Cropping as intensity, Rotation, Normalization] | | 0.8967 | 0.7850 | | -Here I Found Good Training and Validation accuracy from past Conv3D models.  -Still Over fitting is there | |
| MobileNET+GRU  (trainable – True) | | CNN-RNN | Trainable prams: 2,489,349  Non-trainable prams: 36,672 | | Yes  [ Cropping as intensity, Rotation, Normalization] | | 0.9955 | 0.9300 | | -These is best model till we found | |