**NEURAL NETWORK PROJECT – GESTURE RECOGNITION CASE STUDY**

Submitted by: Akanksha Dwivedi, Mohit Dhawan, Snehal Kedar; DSC64

**1. Problem Statement**

As a data scientist at a home electronics company which manufactures state of the art smart televisions. We 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 five gestures are as follows:

1. Thumbs up: Increase the volume.
2. Thumbs down: Decrease the volume.
3. Left swipe: 'Jump' backwards 10 seconds.
4. Right swipe: 'Jump' forward 10 seconds.
5. Stop: Pause the movie.

**2. Objective**

Our task is to train different models on the 'train' folder to predict the action performed in each sequence or video. The final test folder for evaluation is withheld - final model's performance will be tested on the 'test' set. **Two** types of architectures suggested for analysing videos using deep learning:

1. **CNN + RNN architecture**: The conv2D network will extract a feature vector for each image, and a sequence of these feature vectors is then fed to an RNN-based network. The output of the RNN is a regular softmax (for a classification problem such as this one).

2. **3D Convolutional Neural Networks (Conv3D)**: 3D convolutions are a natural extension to the 2D convolutions, just like in 2D conv, you move the filter in 2 directions, in 3D conv, you move the filter in 3 directions.

**3. Data Generator**

In the generator, we are going to pre-process the images as we have images of 2 different dimensions (360 x 360 and 120 x 160) as well as create a batch of video frames. The generator should be able to take a batch of videos as input without any error. Steps like cropping, resizing and normalization should be performed successfully.

**4. Data Pre-processing**

1. **Resizing and cropping of the images**: This was mainly done to ensure that the NN only recognizes the gestures effectively rather than focusing on the other background noise present in the image.
2. **Normalization of the images**: Normalizing the RGB values of an image can at times be a simple and effective way to get rid of distortions caused by lights and shadows in an image.

**5. Methodology**

The approach for this project is as follows:

1. The appropriate generator function was written for Batch processing.
2. Certain utility functions were initialised to aid in model development and assessment.
3. Based on different hyperparameters model experimentation was carried out and total of **EIGHT** models were tested.
4. After each model experiment its training and validation accuracy and loss were checked.

SUMMARY OF EXPERIMENTS:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model No.** | **Model Type** | **Hyperparameters** | **Accuracy** | **Remarks**  **(Explanation & Decisions)** |
| 1 | Conv3D | image resolution 100,  batch size=8,  num\_epochs=25,  filter size=(3,3,3),  SGD(learning\_rate=0.001),  Dropout(0.5) | Training accuracy: 32%  Validation accuracy: 46% | * Hyperparameters selected as starting point to get an initial model * Low accuracy and underfitting observed, model isn’t learning enough features. * Learning rate of 0.001 with SGD may be too small, resulting in slow convergence, consider switching to Adam optimizer in next model. * Reducing image resolution, to speed up training. Need to select resolution which will not comprise accuracy in next model. |
| 2 | Conv3D | image resolution 50,  batch size=4,  num\_epochs=25,  filter size=(3,3,3),  Adam(learning\_rate=0.0002),  Dropout(0.5) | Training accuracy:  53%  Validation accuracy: 73% | * Underfitting still observed. * Training accuracy though improved is less. * Dropout of 0.5 might be too high (as model is underfitting), reducing dropout to 0.25 in next model to make learning more effective. |
| 3 | Conv3D | image resolution 50,  batch size=4,  num\_epochs=25,  filter size=(3,3,3),  Adam(learning\_rate=0.0002),  Dropout(0.25),  num\_frames 15 to 10 | Training accuracy:  83%  Validation accuracy: 79% | * Underfitting resolved and accuracy score also improved. * Attempt to further improve accuracy, will try by increasing the number of epochsin next model, allowing the model to train longer and capture more details. |
| 4 | Conv3D | image resolution 50,  batch size=4,  num\_epochs=35,  filter size=(3,3,3),  Adam(learning\_rate=0.0002),  Dropout(0.25),  num\_frames 15 to 10 | Training accuracy:  85%  Validation accuracy: 71% | * Training accuracy is improved with increasing epochs from 25 to 35 but caused Overfitting hence not a good option. * Will try modifying filter to capture local patterns and change batch size in next model. |
| 5 | Conv3D | image resolution 50 ,  batch\_size = 20,  num\_epochs=25,  filter size =(2,2,2) ,  Adam(learning\_rate=0.0002),  Dropout(0.25) | Training accuracy:  51%  Validation accuracy: 43% | * Modification of filter size not suitable for model as with smaller filter model may have missed larger features. * Filter size being reverted in next model. |
| 6 | Conv3D | image resolution 50 ,  batch\_size = 10,  num\_epochs=25,  filter size=(3,3,3),  Adam(learning\_rate=0.0002),  Dropout(0.25) | Training accuracy: 65%  Validation accuracy: 63% | * Accuracy improved as compare to model 5, still not optimal. * Will further experiment with learning rate in next model. |
| 7 | Conv3D | image resolution 50 ,  batch\_size = 7,  num\_epochs=25,  filter size=(3,3,3),  Adam(learning\_rate=0.0001),  Dropout(0.25) | Training accuracy: 60%  Validation accuracy: 60% | * Change in learning rate reduced accuracy. * Will add layer 64 to try increase next model’s capacity allowing it to learn more complex features. |
| 8 | Conv3D | image resolution 50 ,  batch\_size = 7,  num\_epochs=25,  filter size=(3,3,3),  add layer 64,  Adam(learning\_rate=0.001),  Dropout(0.25) | Training accuracy:  76%  Validation accuracy: 67% | * Adding layer 64 improved the model accuracy but also causing overfitting. * learning rate = 0.001 might still be too high probably causing overfitting. * We finalized **Model 3**. |

**6. Conclusion**

The project was completed after detailed experimentation and the process has been recorded along with interim results. The ipython file has the full code along with different metrics displayed for all models.

We finalized **Model 3.**

**Parameters used:**

* total\_frames = 30
* num\_frames = 10
* gestures = 5
* image\_height = 50
* image\_width = 50
* batch\_size = 4
* filter size = (3,3,3)
* Optimizer = Adam(learning\_rate=0.0002)
* Dropout(0.25)
* num\_epochs=25

**Reason:**

* **Training Accuracy:** 0.83%
* **Validation Accuracy:** 0.79%
* A **drop out rate of 0.25** gives up a better training and validation accuracy
* Underfitting resolved and better accuracy score as compared to other models