Assignment 4 - Camera-based Music Player Control

Krishanu Das Baksi **Deepak Raina**
2019ANZ8274 **2019MEZ8497**

COL-780 Computer Vision
November 13, 2019

Approach:

1. **Automating the dataset collection:**
   Like any learning-based task, the effectiveness of this assignment is heavily dependent on data preparation. So considering this as a crucial part, we applied few techniques in advance to make an effective dataset.
   i) First, the image collection and annotation process for various gestures has been automated by writing a code, that will make it less tedious for the one who is collecting the data.
   ii) With this automation, it has become quite easy to record various variations like hand orientation, left or right hand, background and light intensity changes, etc. This will help in generalizing the model.

2. **Preprocessing in images:**
   While capturing the frames from the webcam, background subtraction using the MOG background subtraction in OpenCV was used. These were concatenated to the RGB image array to give a 4 channel image array. These concatenated arrays were the inputs to the Convolutional Neural Networks (CNN).

3. **CNN network architecture:**
   The architecture of our network is summarized in Figure 1. It contains 4 learned layers — three convolutional and one fully-connected. The input to the network is 50X50 4 channel (RGB + Background Subtraction) image/array of a hand gesture and output is one of the four classes i.e. None, Stop, Back, Next. Briefly, the network has 5 layers, 3 convolutional layers and 2 fully connected layers. All the convolutional layers have kernels size of 3 and the number of kernels is 32, 64 and 128, respectively. The fully connected layer has 7X7X 128 and 64 nodes in the hidden layers and 4 outputs as described above. Below, we describe some of the techniques applied in the network to obtain a better prediction rate.

i) Batch Normalization: It helps in preventing overfitting because it has slight regularization effects. It allows us to use higher learning rates because it makes sure that there's no activation that goes very high or low.

ii) Dropout: It refers to ignoring units (i.e. neurons) during the training phase of certain set of neurons which is chosen at random. It is also a regularisation technique to prevent overfitting. We applied a dropout rate of 0.2 for 3 convolution layers and of 0.5 for fully connected layers.

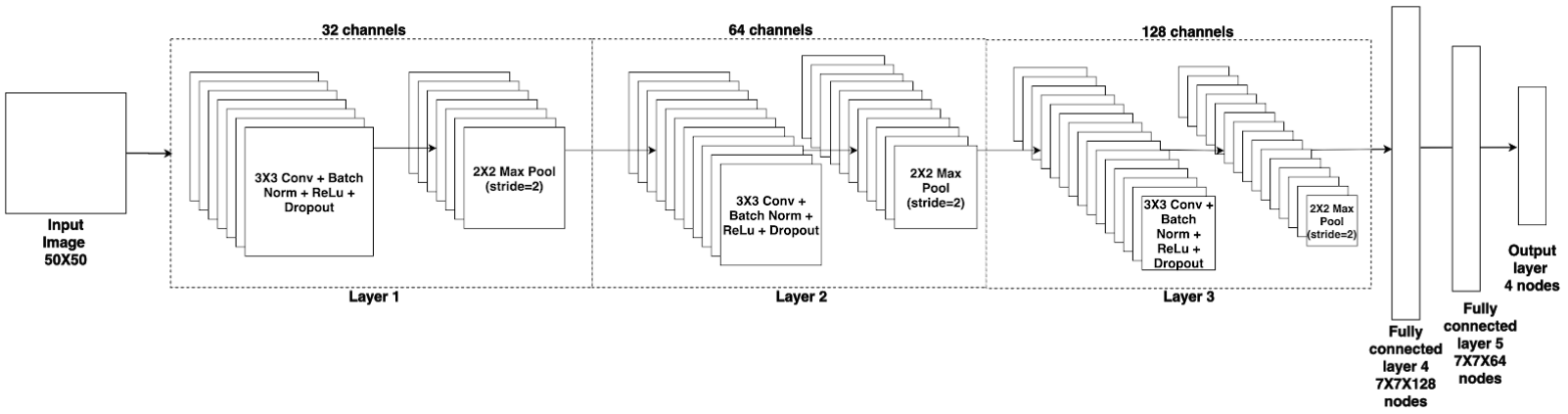


Fig. 1: A network architecture. This is a representation of various techniques used in each layer of the network. The size and number of images shown in the figure are not drawn to scale.

4. Training

a. Data Splitting: The model was trained on 16000 image-arrays out of around 22000 images that were collected. In our training set, the distribution of the different classes were roughly equal. So the resulting training set will also contain roughly similar number of images (4000 per class). A validation set of 400 images was constructed by random sampling.

b. Data Augmentation:

Since we had only a small number of images to begin with (4000 per class), we augmented the data using several processes to make the model generalize better. While augmenting, we also had to make sure that the spatial orientation of the hand doesn't change too much so as to make the augmented image looks like it belongs to the original class. We applied the following augmentations:

1. Random Brightness, Hue, Saturation, Contrast Jitter
2. Random rotation of the images by angles between -30 to +30 degrees.
3. Random flipping of the images.

c. Training:

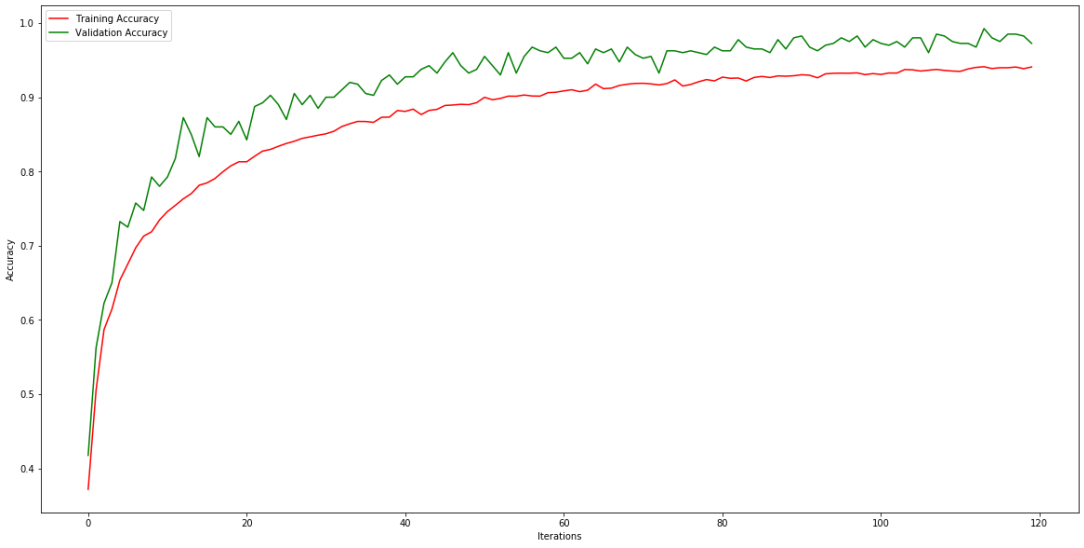
The model was trained for around 120 epochs, using a learning rate of 0.001 and the Adam optimizer. Loss function used was Cross Entropy.

5. Results:

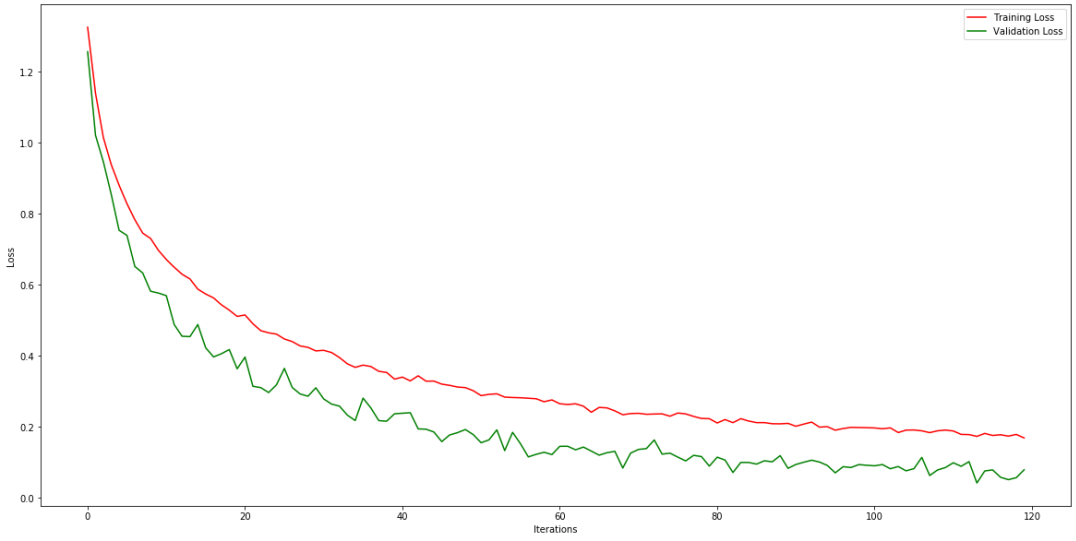
a. Loss and Accuracy Curves:

The training and validation loss and accuracies are plotted.

Plot of accuracy vs Iteration Number (Green: Validation, Red: Training)



Plot of Loss vs Iterations Number (Green: Validation, Red: Training)



b. Inference:

Two methods have been used for inference. In the simple method, the class having the highest probability for a frame as predicted by the model is the final class prediction for the frame. In the other technique, smoothing/averaging the

predictions in order to reduce fluctuations in predictions was also tried. The final probability predictions of the current frame were found by a weighted sum of the model probability outputs of the current frame and the final predictions of the last frame. The class prediction was the class having a maximum probability among the weighted sum of probabilities.

c. Out of Sample Test Results

To improve our results, we trained different models and tested their accuracy in different background and lighting conditions, which were not there in the training set. We tried to improve the accuracy by changing the hyperparameters of the network and inference techniques. The table comparing the accuracy of various trained models on the test dataset is given below:

Table 1: Comparison of test accuracy in different trained models

Class	**None**	**Stop**	**Next**	**Previous**	**Average**
Model 1	0.64	0.74	0.86	0.94	0.80
Model 2	0.98	0.74	0.62	0.85	0.80
Model 3	0.98	0.78	0.62	0.93	0.83
Model 4	0.99	0.88	0.89	0.90	0.92
Model 5	1.00	0.41	0.57	0.38	0.59

6. How to run the code

1. Without Media player integration

   python test.py

2. With media player integration (requires few media *.mp3* files in the current directory)

   python vlctest.py

Files:

The link of the training scripts, trained models, testing codes is given below

<files>