Assignment 2: Unconstrained Phase Recognizer

CSE 666: Biometrics and Image Processing

Ved Harish Valsangkar
Person Number: 50290388
Department of CS&E
University at Buffalo
vedharis@buffalo.edu

Architecture

To create an unconstrained phase recognizer using the IJB-A data set. ResNet34 architecture was selected as a classifier. Model was trained from scratch on all 10 splits resulting in 10 different models. Architectural modifications are discussed below. The model was built using the following site as reference: `https://github.com/kuangliu/pytorch-cifar/blob/master/models/resnet.py`. The nested nn.Module class structure and _build_layer() method was adapted from aforementioned site.

The input is an image with dimensions $203 \times 202 \times 3$. The model has a penultimate feature layer of 2048 nodes. The last layer is a 500 node classifier layer.

Basic Block

This is the basic box of the ResNet. For ResNet34, this block consisted of two 2D convolutional layers followed by the short cut. Both kernels are of 3×3 dimensions with padding 1. Activation function for all layers was kept as ReLU.

ResNet34

This is the main model structure build using the basic blocks. An initial convolutional layer was needed to adapt the 3 channel input image to 16 channel tensor. This was followed by 4 layers made up of sequential basic blocks. Layers numbered 1 to 4 had 3,4,6,3 blocks respectively.

When the model is in training mode, the feature layer is connected to a 500 node class layer which classifies the output according to the classes in the dataset. Thus the model is trained as a classifier. When the model is set to evaluation mode, the forward operation omits the last classifier layer and outputs the feature layer instead. Thus the model need not be chopped to get the feature extractor.

Model Files

The model and the associated graphs can be downloaded from the following box folder:
`https://buffalo.box.com/s/scjr7fz8qycv1c077k91dfzzsibpxjor`

Each file contains the model, optimizer and the criterion used in training for the specific script. The folder also contains ROC graphs for the individual scripts.

Block Diagram

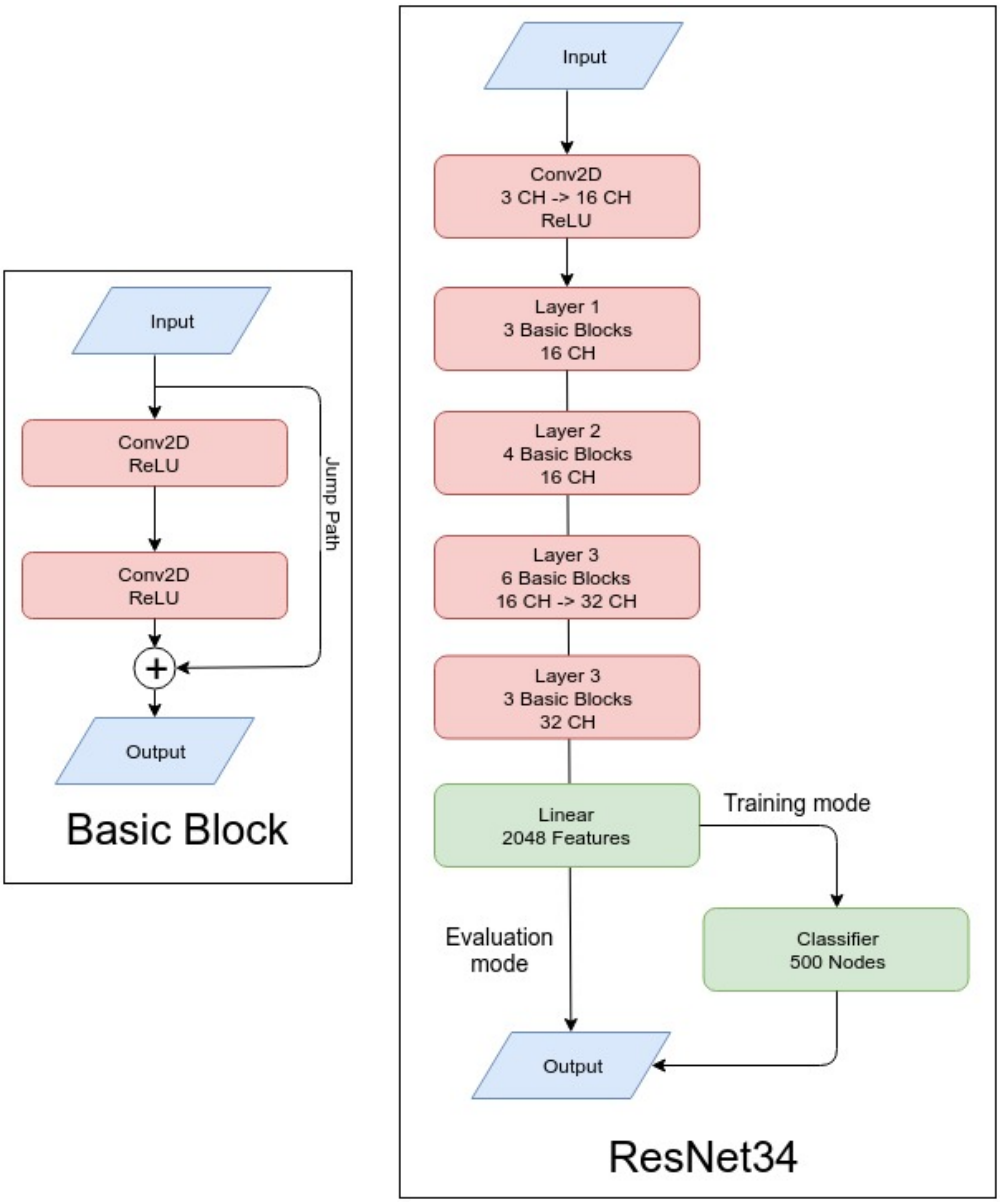

Figure 1: Model Block Diagram

ROC Graphs

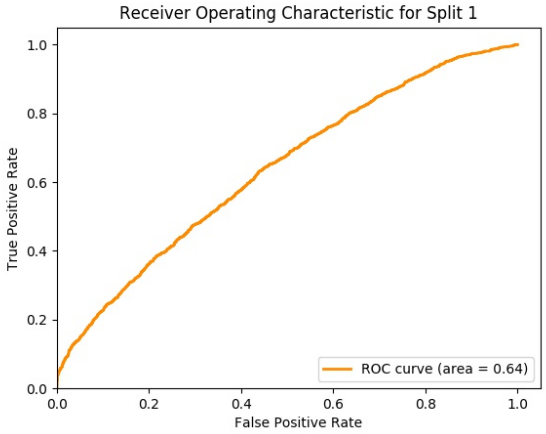

Figure 2: ROC graph for Split 1

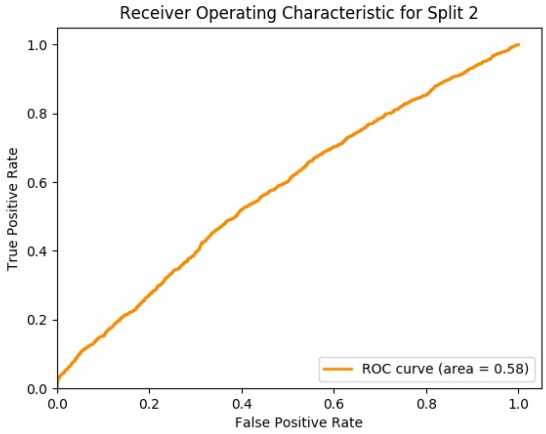

Figure 3: ROC graph for Split 2

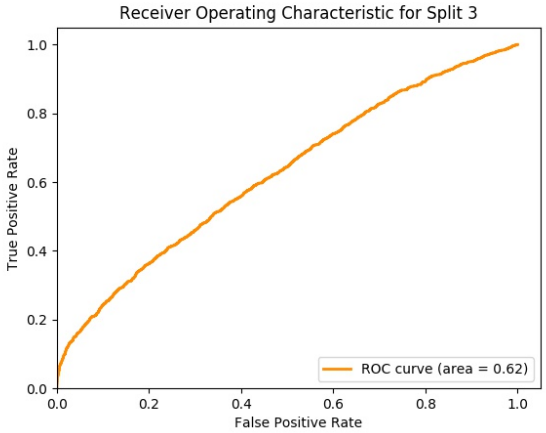

Figure 4: ROC graph for Split 3

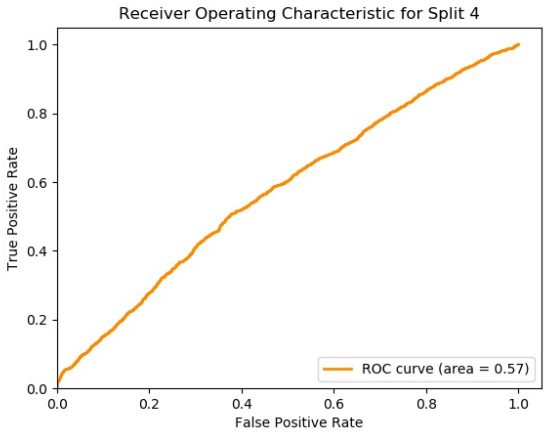

Figure 5: ROC graph for Split 4

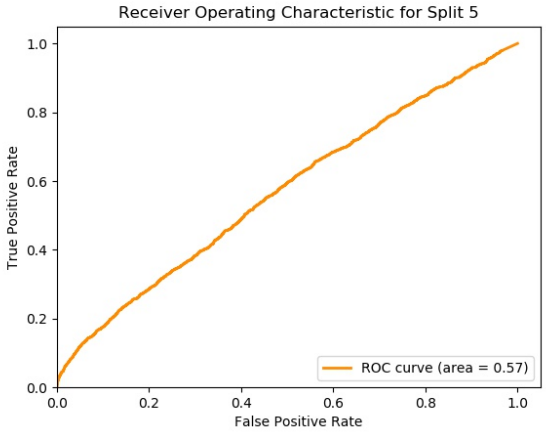

Figure 6: ROC graph for Split 5

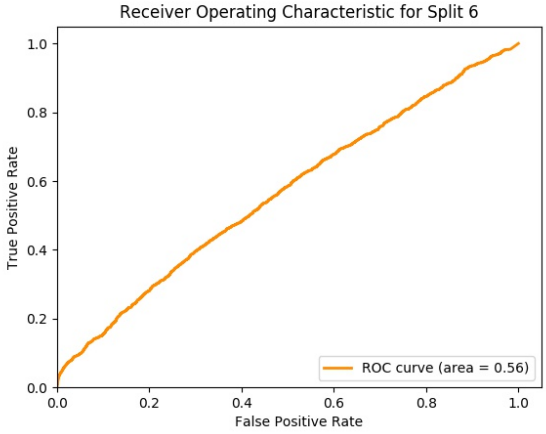

Figure 7: ROC graph for Split 6

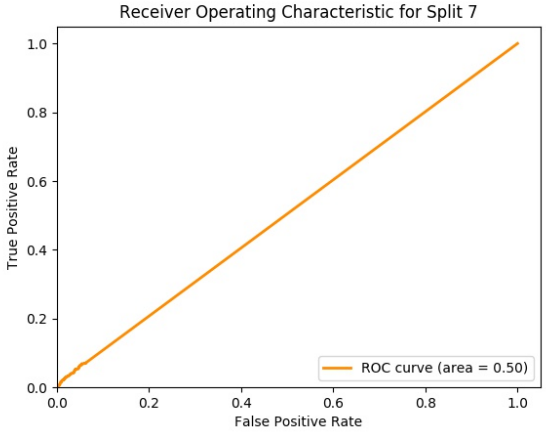

Figure 8: ROC graph for Split 7

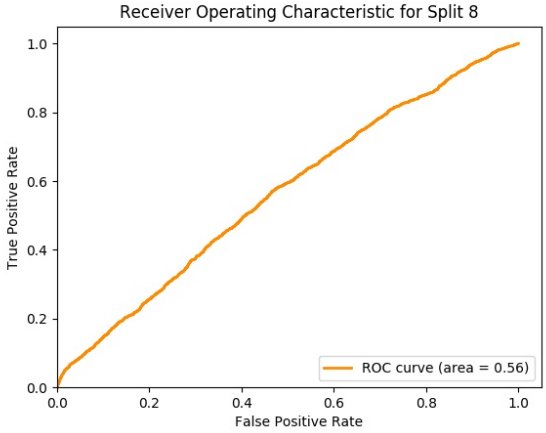

Figure 9: ROC graph for Split 8

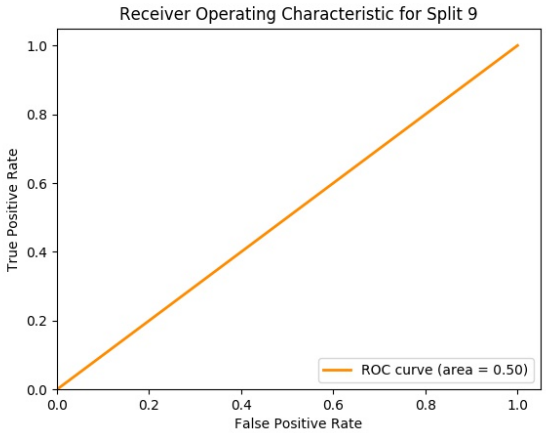

Figure 10: ROC graph for Split 9

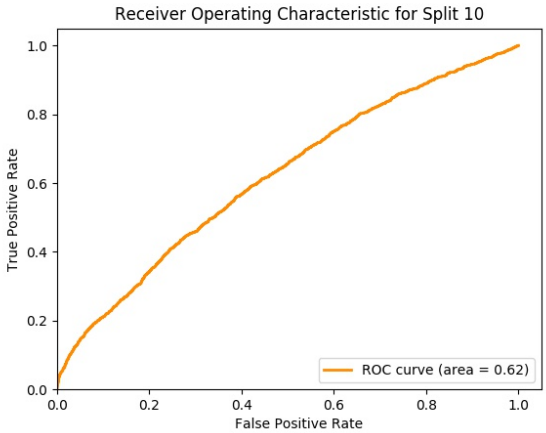

Figure 11: ROC graph for Split 10