%load resnet

resnet = resnet50;

%load train data

dataFolder = "temp";

fprintf("Storing Train Data\n")

[files, labels] = hmdb51Files(dataFolder);

inputSize = resnet.Layers(1).InputSize(1:2);

numFiles = numel(files);

data = cell(numFiles,1);

for i =1:numel(files)

fprintf("Reading file %d of %d...\n", i, numFiles)

video = readVideo(files(i));

video = centerCrop(video, inputSize);

data{i,1} = video;

end

%%

resnet = resnet50;

%Train Test Split

fprintf("Train Test Split\n")

numObservations = numel(data);

idx = randperm(numObservations);

N = floor(0.7 \* numObservations);

idxTrain = idx(1:N);

dataTrain = data(idxTrain);

labelsTrain = labels(idxTrain);

idxValidation = idx(N+1:end);

dataValidation = data(idxValidation);

labelsValidation = labels(idxValidation);

%Transfer Learning

layersTransfer = resnet.Layers(1:end-3);

numFeatures = size(dataTrain{1},1);

numClasses = numel(categories(labelsTrain));

layers = [

layersTransfer

fullyConnectedLayer(numClasses,'WeightLearnRateFactor',20,'BiasLearnRateFactor',20)

softmaxLayer

classificationLayer];

miniBatchSize = 8;

numObservations = numel(dataTrain);

numIterationsPerEpoch = floor(numObservations / miniBatchSize);

options = trainingOptions('adam', ...

'MiniBatchSize',miniBatchSize, ...

'InitialLearnRate',1e-4, ...

'GradientThreshold',2, ...

'Shuffle','every-epoch', ...

'ValidationData',{dataValidation,labelsValidation}, ...

'ValidationFrequency',numIterationsPerEpoch, ...

'Plots','training-progress', ...

'Verbose',false);

fprintf("Start Training\n")

[resnet,info] = trainNetwork(dataTrain,labelsTrain,layers,options);