netCNN = googlenet;

dataFolder = "D:\hmdb51\_org"

[files,labels] = hmdb51Files(dataFolder);

idx = 1;

filename = files(idx);

video = readVideo(filename);

size(video)

labels(idx)

numFrames = size(video,4);

figure

for i = 1:numFrames

frame = video(:,:,:,i);

imshow(frame/255);

drawnow

end

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

layerName = "pool5-7x7\_s1";

tempFile = fullfile(tempdir,"hmdb51\_org.mat");

if exist(tempFile,'file')

load(tempFile,"sequences")

else

numFiles = numel(files);

sequences = cell(numFiles,1);

for i = 1:numFiles

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

video = readVideo(files(i));

video = centerCrop(video,inputSize);

sequences{i,1} = activations(netCNN,video,layerName,'OutputAs','columns');

end

save(tempFile,"sequences","-v7.3");

end

numObservations = numel(sequences);

idx = randperm(numObservations);

N = floor(0.9 \* numObservations);

idxTrain = idx(1:N);

sequencesTrain = sequences(idxTrain);

labelsTrain = labels(idxTrain);

idxValidation = idx(N+1:end);

sequencesValidation = sequences(idxValidation);

labelsValidation = labels(idxValidation);

numObservationsTrain = numel(sequencesTrain);

sequenceLengths = zeros(1,numObservationsTrain);

for i = 1:numObservationsTrain

sequence = sequencesTrain{i};

sequenceLengths(i) = size(sequence,2);

end

figure

histogram(sequenceLengths)

title("Sequence Lengths")

xlabel("Sequence Length")

ylabel("Frequency")

maxLength = 400;

idx = sequenceLengths > maxLength;

sequencesTrain(idx) = [];

labelsTrain(idx) = [];

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

numClasses = numel(categories(labelsTrain));

layers = [

sequenceInputLayer(numFeatures,'Name','sequence')

bilstmLayer(2000,'OutputMode','last','Name','bilstm')

dropoutLayer(0.5,'Name','drop')

fullyConnectedLayer(numClasses,'Name','fc')

softmaxLayer('Name','softmax')

classificationLayer('Name','classification')];

miniBatchSize = 16;

numObservations = numel(sequencesTrain);

numIterationsPerEpoch = floor(numObservations / miniBatchSize);

options = trainingOptions('adam', ...

'MiniBatchSize',miniBatchSize, ...

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

'MaxEpochs',10,...

'GradientThreshold',2, ...

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

'ValidationData',{sequencesValidation,labelsValidation}, ...

'ValidationFrequency',numIterationsPerEpoch, ...

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

'Verbose',false);

[netLSTM,info] = trainNetwork(sequencesTrain,labelsTrain,layers,options);