Tennis stroke classifier

1) Background:
    a. In order to improve skills in the game of tennis, we need feedback. Taking a video of the game played and analyzing the strokes, such as serve, forehand or backhand, provides a great feedback. However, when we take a video, we have to edit the video and classify manually every stroke. This can be a very painful, time-consuming process. I'd like to address this problem using spatio-temporal convolutional3D neural networks and other transfer learning techniques to achieve best accuracy possible for the classification. In this project, I'll address classification of 3 shots/strokes (serve, forehand and backhand). Personally, this would be of great help to me in my tennis skill improvement. The complete solution of taking in an unedited video and outputting a fully edited video with classification labels is beyond the scope of this project. Instead, I'll take in short videos of varied lengths (1-10 seconds) and classify them accordingly (explained more in detail in the problem statement section.)
    b. I have not looked any academic journals in search of a similar topic except for the journals/papers involving video analysis using convolutional neural networks, udacity course projects, deeplearning.ai course projects and other opensource github repositories which will be cited accordingly in the main project.
2) Problem Statement:
    a. Given a video of varied lengths (1-10 seconds), I would like to accurately classify, if the given video is a serve, forehand stroke or a backhand stroke. Input is a video-file, output is either 1) serve or 2) forehand or 3) backhand. I will be using the accuracy metric (i.e number of correct classifications/ total classifications) to analyse the final solutions. In addition, this project assumes only right-handed players. One possible solution is to just create a convolutional3D architecture model using keras/tensorflow libraries. But since I'm creating the training datasets from scratch, I do not have many training samples. Hence, techniques such as transfer learning can significantly help in the improvement of accuracy.
3) Dataset and Inputs:
    a. I'll be using the full unedited videos of the tennis games I obtained from local tennis games. I'll edit the videos into short clips of varied lengths(1-10seconds) and classify the videos as either serve, forehand or backhand. I only have 5 sets of unedited videos. Hence, my training dataset is going to be small. I'll have to find creative ways to solve classification problem. Like using transfer learning or video datasets. I'll be splitting my dataset into 60-20-20 split for training-validation-testing.
4) Solution statement:
    a. I'll be evaluating my model based on accuracy metric. Accuracy %= (total correct classifications/total classifications)*100

    b. Model will be based on (conv3d neural networks) or (transfer learning+conv3d neural network) or (yolov2 person identification + transfer learning + conv3d neural network). More on these techniques and tradeoffs will be discussed in the project.
5) Benchmark Model:
    a. A random guess on 3-category classification problem has a probability of having 33.33% accuracy. I'm trying to achieve atleast 90% accuracy on the test set.
6) Evaluation metric:
    a. Accuracy % = (total correct classifications/total classifications)*100
    b. This metric for the benchmark random guess model yields 33.33% accuracy.
    c. I'm trying to achieve at-least 90% accuracy
7) Project Design:
    a. Acquiring unedited video files
    b. Personally editing video files to get short clips of input datasets of varied classification.
    c. Apply Conv3d neural network model and verify the accuracy metric.
    d. Apply transfer learning + Conv3d model and verify the accuracy metric.
    e. Apply transfer learning + concatenate + Conv2d model and verify the accuracy metric.
    f. Apply yolov2(you-only-look-once) person detection algorithm on short clips and edit the short clips with just the person in the video-clip. Now apply c,d,e algorithms to the edited yolo-video-clips. Verify the accuracy metric. I expect huge improvement in accuracy in this algorithm.