CS230

Tennis Serve Classifier

Vincent Xia
Department of Management Science & Engineering
Stanford University
vincexia@stanford.edu

Sandy Lee
Department of Management Science & Engineering
Stanford University
sl269@stanford.edu

Code: https://github.com/vxia777/tennis_serve_recognition

Abstract

The motivation behind this research project is to leverage computer vision and sequence models to identify real-time tennis serves. In particular, we utilize various CNN architectures to generate features from serving video data then pipe these features through a many-to-many RNN architecture to evaluate which CNN/RNN mix produces the best results. From our empirical analyses, we saw that the best deep learning architecture for classifying real-time tennis serves came from the features generated using a InceptionV3 model and many-to-many LSTM RNN architecture.

1 Introduction

The serve is one of the most fundamental and critical components of tennis. Like the opening move in chess, the serve sets the tone and direction of a match, hence making it a critical component of the sport to analyze. There are three main types of serves: flat, slice, and kick serves. The motivation of this work is to utilize computer vision and sequence modeling techniques to identify these serves based on real-time serving video data. After feeding the input tennis serve videos into a CNN, we generate sequenced image features which are then fed into an RNN model to generate a prediction of what type of serve a video is.

2 Related Work

Computer vision modeling in sports has been leveraged a fair amount before, though less so in tennis and specifically with analyzing serves. For example, Mottaghi and Mohsen used action recognition models to analyze postures in freestyle wrestling [1], while others proposed a color segmentation based method for soccer shot video recognition [2]. In the realm of tennis, we draw inspiration from the works of Chow and Dibua [3] as well as from Cai, Tang, Xia, and Chi [4]. Their works focus on tennis shot classification generally via various deep learning techniques. Our project intends to build on the current work by focusing specifically on tennis serve predictions as well as utilizing new vision models like XCeption to test out performance.

CS230: Deep Learning, Winter 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 Dataset and Features

We utilize the THree dimEnsional TennIs Shots (THETIS) dataset [5] for all modeling and analyses for this project. The total dataset consists of 12 different shots made by 24 experienced players and 31 amateurs, with each player performing each shot around 3-5 times. There are 1980 RBG videos in total in .avi format, with about 80 frames per video. Each of the 12 types of shots has 165 .avi video files, and the 3 specific shots we are interested in are the flat, kick, and slice service shots, as mentioned before. This amounts to 495 total videos for the scope of our work.

In this project, we down-sample each video to 16 frames during preprocessing to condense the video data into only the relevant serving motion. We preprocess each video frame by standardizing RGB pixel values.

We ultimately split our data 80/10/10, employing 395 videos for training, 50 videos for development, and 50 videos for test, with each partition having an even number of each of the 3 types of serves so that our dataset is balanced.

Figure 1: Example frames from THETIS dataset

4 Methods

4.1 Frame feature extraction – InceptionV3 and XCeption CNN models

After data preprocessing, we tested 2 different CNN models for feature generation: InceptionV3 and XCeption. We learned InceptionV3 in class and did some research on our own to decide and try out XCeption, a CNN model inspired by Inception but having Inception modules replaced with depthwise separable convolutions. Both CNN models are pre-trained on the ImageNet database to extract relevant image features from our frames. For each frame in each video sequence, we stored the output of the second-to-last layer – a fully connected layer producing a 2048x1 vector of features. The architectures for Inceptionv3 and XCeption are depicted below:

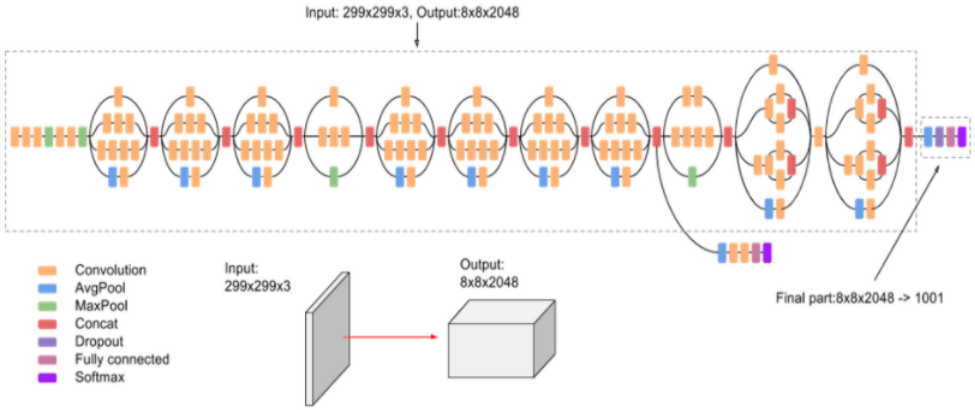

Figure 2: InceptionV3 architecture

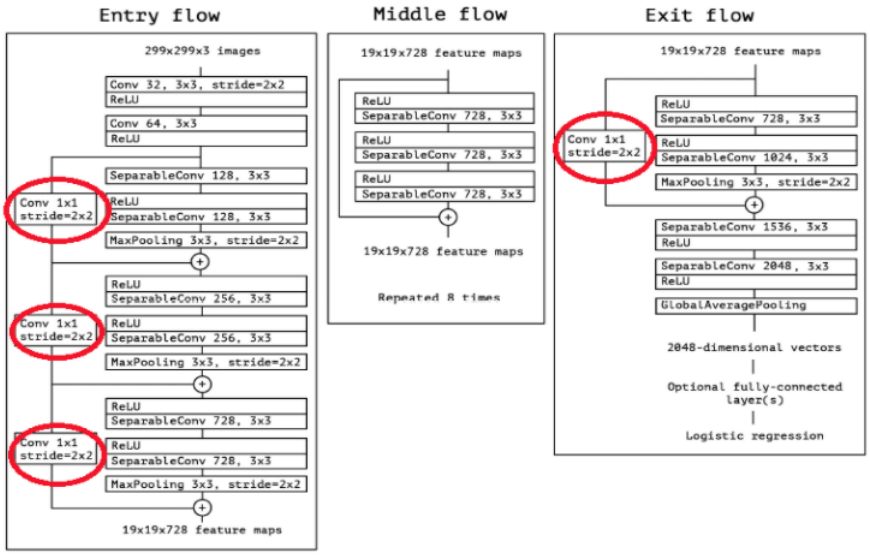

Figure 3: XCeption architecture

4.2 Multi-class classification task – many-to-many LSTM

The final component of the work involves feeding the aforementioned CNN-generated features into a many-to-many LSTM. The loss function we used was categorical cross entropy:

$$J = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n} y_j^{(i)} log(p_j^{(i)}) \tag{1}$$

where m is the number of training samples, n is the number of classes, y is the one-hot vector for ground truth label. The last layer of the LSTM is a softmax layer that outputs probabilities corresponding to each of the 3 serving classes. The classification is done by averaging the result of the softmax outputs across all frames for a given video.

5 Results

We first tuned our hyperparameters on the RNN model using the input data that was pre-trained with InceptionV3. To test out model performance, we performed a uniform grid search on a logarithmic scale. First, we tried controlling the batch size and learning rate to achieve good generalization and acceptable running time. Our model performed best with the smallest batch size of 32 and the highest learning rate of 0.01. Our model was able to converge fast to a good solution, and perhaps this can be intuitively explained by the fact that the amount of input data we had is small. We then tried tuning the number of LSTM hidden and dense units, and were able to achieve the best result with a moderately sized model. In summary, we achieved the following optimal hyperparameters:

Learning Rate	Batch Size	Num of Hidden Units	Num of Dense Units	Dropout
1e-3	32	128	128	0.3

Table 1: Hyperparameters

Using the model with the optimal hyperparameters, we also tested on the input data that was generated by our second CNN architecture Xception. We achieved better training and validation errors on Xception while test error was better on InceptionV3. The difference in errors are small and could be purely due to chance. Here is the comparison of the results between the two architectures:

Architecture	Training	Validation	Test
InceptionV3	98.2%	39.6%	54.9%
XCeption	99.7%	47.9%	51.0%

Table 2: Accuracy Performance

Here is the confusion matrix for InceptionV3->LSTM model for visualization:

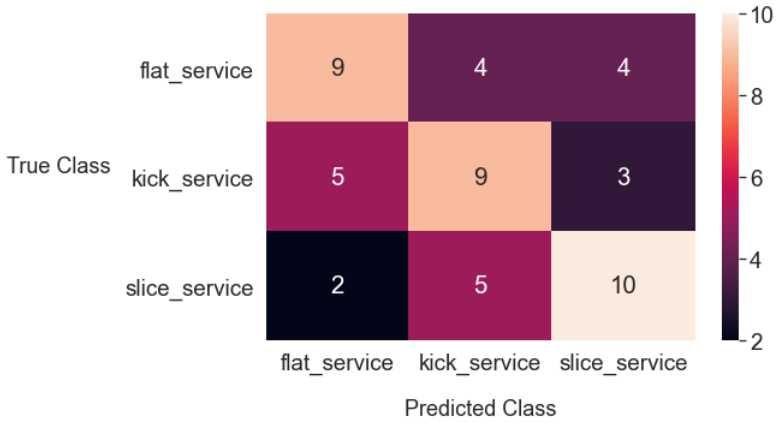

Figure 4: Confusion Matrix

We were satisfied with the results since our prediction performance on the test set is significantly better than making a random guess (1/3 since we have three classes). However, it looks like the model overfit to the training data. The performance on the validation set was worse than on the test set. This could be purely due to chance since our overall data is pretty small (400 videos) and our test and validation sets both use only 10% of the overall dataset.

6 Conclusion/Future Work

Hence, in our research, we built out a novel baseline deep learning model for predicting a tennis player's serve based on features derived from videos. Specifically, we utilized CNN architectures like InceptionV3 and Xception in tandem with LSTM sequence models to develop our classifier models. For further exploration, if we had time, we would have liked to implement some data augmentation techniques into our pipeline, such as manipulating frames of videos via rotations, shearing, and color shifting. This data augmentation may increase our training data size and resolve overfitting issues better. Similarly, if we had more time, we could have pursued more exploration of other RNN models as well as more rigorous hyperparameter tuning for the sequential classification portion. We focused our time more on the CNN feature generation side, so in a future iteration, we could work more on improving the sequential RNN prediction side.

7 Contributions

Both team members contributed equally to completing this project. Specifically, Vincent worked on data engineering/processing, feature generation, and CNN model implementations. Sandy tuned and trained the LSTM sequence model and focused on synthesizing empirical results of the various methodologies described previously.

References

[1] Ali Mottaghi, Mohsen Soryani, and Hamid Seifi. Action recognition in freestyle wrestling using silhouette-skeleton features. *Engineering Science and Technology, an International Journal*, 23(4):921–930, 2020.

[2] Y. Tabii, M. O. Djibril, Y. Hadi, and R. O. H. Thami. A new method for video soccer shot classification. In *VISAPP*, 2007.

[3] Dibua O Chow, V. Action recognition in tennis using deep neural networks. Course project for CS 230. 2018.

[4] Jiaxin Cai, Xin Tang, Haifeng Xia, and Meihong Chi. RGB video based tennis action recognition using a deep historical long short-term memory. *arXiv*, pages 1–6, 2018.

[5] Sofia Gourgari, Georgios Goudelis, Konstantinos Karpouzis, and Stefanos Kollias. THETIS: Three dimensional tennis shots a human action dataset. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 676–681, 2013.

[6] G Buddhiraju S., Sankararaman. Can AI Make You a Better Athlete? Using Machine Learning to Analyze Tennis Serves and Penalty Kicks. 2020.