An AI Approach to Pose-based Sports Activity Classification

Rajdeep Chatterjee*[1], Soham Roy[1], SK Hafizul Islam[2] and Debabrata Samanta[3]
[1]School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar 751024, Odisha, India
[2]Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, Kalyani 741235, West Bengal, India
[3]Department of Computer Science, CHRIST Deemed to be University, Hosur Road, Bengaluru 560029, Karnataka, India
Email: {cse.rajdeep, soham.roy2538, hafi786, debabrata.samanta369}@gmail.com

2021 8th International Conference on Signal Processing and Integrated Networks (SPIN) | 978-1-6654-3564-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/SPIN52536.2021.9565996

***Abstract*—Artificial intelligence systems have permeated into all spheres of our life-impacting everything from our food habits to our sleep patterns. One untouched area where such intelligent systems are still in their infancy is sports. There has not been enough indulgence of AI techniques in sports, and most of the works are carried on manually by coaching staff and human appointees. We believe that intelligent systems can make coaching staff's work easier and produce findings that the human eye can often overlook. Here, we have proposed an intelligent system to analyze the beautiful game of tennis. With the use of computer vision architecture Detectron2 and activity-based pose estimation and subsequent classification, it can identify an action from a tennis shot (activity). It can produce a performance score for the player based on pose and movement like forehand and backhand. It can also be used to understand and evaluate the strengths and weaknesses of the player. The proposed approach provides a piece of valuable information for a player's performance and activity detection to be used for better coaching. The study achieves a classification accuracy of 98.60% and outperforms other SOTA CNN models.**

***Index Terms*—Activity, Classification, Keypoints, Pose estimation, Sports, Tennis**

I. Introduction

One thing which unites us all is sports, which is loved by all. Billions across the globe watch the final of the FIFA world cup or Wimbledon slam. Though, most of the decision-making in sports is primarily human-centric and mechanical work. Intelligent systems have not established a stronghold in sports because it is challenging to establish sporting jargon. The sports-related activities in terms of technical jargon and the lack of human and non-human resources equate those two into one equation, bind them together, and find common grounds for both. Though recently this has been an area of improvement, the imagination is still in its infancy on how technology can impact sports as it has in various other domains like healthcare, E-commerce, etc. We propose a method to assess the performance of an individual in tennis in various avenues like posture, forehand, and backhand swing movements, feet movements, and return to base position [1] after a shot-making by leveraging computer vision. We have chosen this avenue because the points mentioned above are the basics of tennis, and many individuals trying to learn the game tend to make mistakes in these aspects. Such mistakes are tough to decipher by the naked eye because swing and foot movements are so fast and subtle that the mistakes almost go unnoticed. Such obstacle prevents players from improving their game in later stages and becomes the roadblock to achieving mastery. Issues such as this must be nipped in the bud. Elite tennis academies like Rafa Nadal Academy, Moratoglou Academy generally address this issue by capturing a player's video and then manually analyzing them frame by frame to assess the player's performance [2]. This practice is not only very time-consuming but requires a trained professional to make a detailed assessment. Such a level of privilege is not attainable by tennis academies on the lower end of the spectrum. Therefore, there is a scope for exploring the AI techniques which can do the activity classification of various tennis shots and monitor the sport with professional players like Nadal and Federer. It assesses the initial performance of an amateur player generally through shot-by-shot performance indices, which range between 0 to 1, 0 denoting very poor and plenty of mistakes, and 1 denoting near-perfect movements and performances. Based on the model's performance score, it is easier to understand the strengths and weaknesses of the player.

*Corresponding Author

A. Contribution

In this paper, an AI approach has been proposed to capture the human skeleton-based sports activity classification. This approach is called an AI scheme as it encompasses both deep learning and machine learning algorithms. The deep learning technique estimates the keypoints from computer vision generated human skeletons. Furthermore, these points are used to create new game-specific features for better classification using the traditional machine learning techniques.

B. Organization

The paper has been divided into six sections. The related and relevant research works have been discussed in Section II. It is followed by Section III containing the theoretical concepts used in this paper. The proposed method has been described in Section IV. The results and analysis are given in

Section V. Finally, the paper has been concluded in Section VI.

II. Related Work

There have been only a few peer-reviewed works done in this domain. Recently Sarlis et al. [3] discuss sports analytics on basketball to evaluate the player and team performance. They have used typical data science and data mining activities to reach team composition and team analytics strategy. However, basketball being a team game, is very different from tennis, primarily an individual's game. Thus, it becomes reasonably uncharted territory.

In the paper, Sun et al. [4] address the trade-off between accuracy and transparency for deep learning applied to sports analytics. They have used ice hockey and soccer to perform the analysis. They have shown that a mimic model is a linear model tree, which combines a collection of linear models with a regression-tree structure, a tree version of a neural network that explains itself and produces insights for expert stakeholders such as athletes and coaches.

A similar paper by Pantzalis et al. [5] aims at player's performance prediction in football. They present a prediction of the final league table for specific leagues, using past data and advanced statistics. They give predictions, such as whether a team will have a better or worse season than the last season.

The authors have not used the power of computer vision to extract meaningful information directly from the game. We have used the latest computer vision architecture, Detectron2, and pose estimation techniques to develop a lightweight classifier. The proposed approach takes images or videos as input and estimates the keypoints from each frame. Subsequently, these bodypoints and few newly generated features combined help to improve the overall sports activity classification.

III. Background Concept

We have used Detectron2 [6] Facebook Artificial Intelligence Research (FAIR) next-generation platform for object detection, segmentation, and other visual recognition tasks for pose estimation and keypoints detection. Recently Detectron2 had been used for a pose detection purpose by Zabifihar et al. [7] and showed very promising results. The platform is implemented in PyTorch. Specifically, we have used a pretrained model instead of training our model from scratch cause Detectron2 gives us a wide array of options to select from state-of-the-art (SOTA) pretrained models on human pose and keypoints detection. We use random forest [8] to classify activity classification (pose estimation) in tennis since Random forest works very well in these types of numerical datasets[9] and past works showed the efficacy of this algorithm in sports analytics[10]. As its name implies, the Random Forest consists of many individual decision trees that operate as an ensemble. Ensemble models are preferred as it provides better performance due to its inherent diversity. Each tree in the random forest gives out a class prediction, and the class with the most votes becomes the model's prediction. The decision tree does split into the data to build the tree. Each split is a single line that divides data points into nodes. At each node, the decision tree searches through the features for the value to split on, resulting in the greatest reduction in Gini Impurity. The Gini Impurity of a node is defined by the probability that a randomly chosen sample in a node would be incorrectly labeled if it has been labeled by the distribution of samples in the node. The following Eq calculates it. 1.

$$I_G(n) = 1 - \sum_{i=1}^{J} (p_i)^2 \tag{1}$$

Where I_G is the information gain at node n, p_i is the fraction of examples at class i. We will be using a standard scaler to standardize the Euclidean coordinates of the X, Y-axis, and the confidence score of the keypoints returned by the Detectron2 model for that player pose. The reason is that those coordinates are measured at different scales depending on the player's position, and using the raw values does not contribute equally to the model fitting to the learnt function and might end up creating a bias. Thus, to deal with this potential problem, feature-wise standardization is usually used before model fitting. The idea is to bring the data with a mean is 0, and a standard deviation is 1.

We use feature engineering to compute various angles and distances between body parts for a pose. The working principle is based on *Law of Cosine* [11]. The following mathematical expressions compute it (see Eq. 2).

$$c^2 = a^2 + b^2 - 2ab \ \ cos(\theta) \tag{2}$$

We are using cosine similarity to calculate the similarity between two performance vectors. The cosine similarity [12] is calculated as in Eq. 3.

$$Sim(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}.\mathbf{y}}{\|\mathbf{x}\|.\|\mathbf{y}\|} \tag{3}$$

Where $\|\mathbf{x}\|$ is the *Euclidean norm* of vector $\mathbf{x}=(x_1, x_2, x_3, \ldots, x_n)$.

IV. Proposed Approach

A. Dataset Preparation

To predict the action and movement of the athlete, we have categorized the player's movements into three main classes. These are the forehand motion, backhand motion, and the reset position or the base position of the player [13], [14]. We refer to the base position as a player's position after making a forehand or backhand shot. There are many names associated with the base position used in the tennis community, like split step, position zero, etc. These are the three main actions that are played repeatedly while a player is practicing. We have considered only the *court level view* of a player to maintain uniformity in the camera angle across the data. Our dataset consists of 1000 images for each class (that is of total 3K images) and has been collected from YouTube clips of players practicing, and they have been manually annotated

with class labels for model training purposes. The dataset has been divided into 8 : 2 for training and testing purposes. The dataset and analysis are done only considering right-handed players into account. Examples are shown in Fig. 1 of the three positions from a court level angle view.

(a) Forehand (b) Backhand (c) Base position

Fig. 1: Visualization of the three position from a court level camera views (forehand, backhand and base position) in Tennis

Our analysis has two parts. The first part involves the action or activity classification of the player. The approach being followed here is that the image of the player practicing is being read, and after some image processing like resizing, enhancement is being fed to the Detectron2, and the desired output is extracted from the overall Detectron2 keypoints detection model. Our interest here is the coordinates of the key body locations in a 2D Euclidean plane and the confidence score of the coordinate at that point while omitting other entities from the Detectron2 results. There are 17 keypoints (COCO format) that the Detectron2 keypoints detection model returns. These are the location of "nose", "left eye", "right eye", "left ear", "right ear", "left shoulder", "right shoulder", "left elbow", "right elbow", "left wrist", "right wrist", "left hip", "right hip", "left knee", "right knee", "left ankle", "right ankle". Out of these, some keypoints are of no use in our problem statement like "nose", "left eye", "right eye", "left ear", "right ear". Therefore, we ignore these 5 keypoints and work with only 12 keypoints. We standardize the raw coordinates to 0 mean 1 standard deviation as the absolute coordinates can differ based on player position in the frame. The standardization takes out the biasness from the player position in the frame. From these raw standardized coordinates, we calculate a total of 14 new game-specific features that are important in tennis. The newly engineered features are the angle at both the knees, elbows, hips, shoulder (see in Fig. 2) and the distances between the left-right elbows, left-right wrists, left-right knees, left-right shoulders, and hip. Tennis, being a geometry game, can be easily analyzed, and different metrics can be drawn from the newly engineered features. The international professional players have mastered the movements in these departments, and these can be used as a benchmark to analyze amateur players as a comparative study. We have built an ML model to do the classification task. With a combination of the standardized coordinates and newly engineered features, a data frame of 50 features (that is, 12 keypoints $\times 3 = 36$ most important features+14 newly created features) are created for each frame, and using the manually annotated label as the target variable; a random forest machine learning algorithm is trained for the action classification task. A bird's eye view of the overall proposed approach is shown in Fig. 3.

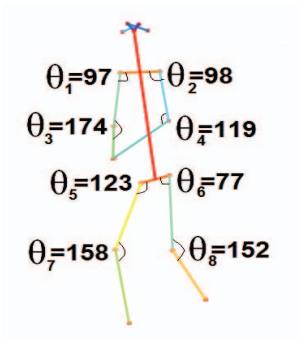


Fig. 2: An example of newly generated angle (in degrees) features from given keypoints based on Eq. 2

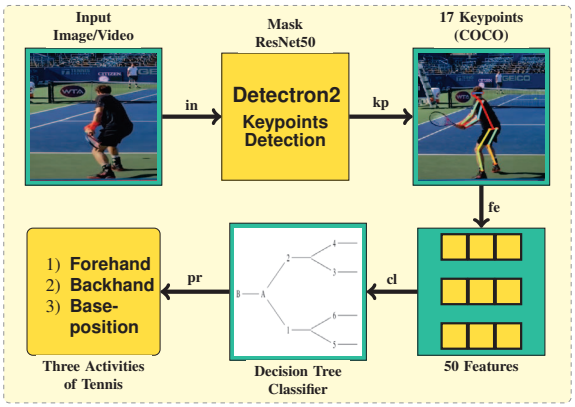


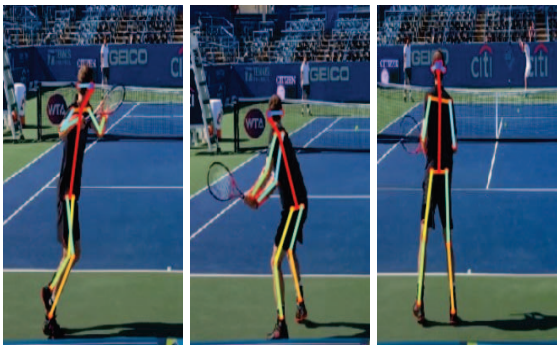
Fig. 3: The block diagram representation of the proposed approach to classify pose-based player's activities

The terms **in**, **kp**, **fe**, **cl** and **pr** suggest inputs, keypoints, feature extraction, classification and prediction, respectively.

We also derive the feature importance for each class which will be shown later. This model is used to test on static images as well on video clips (refer Fig. 4).

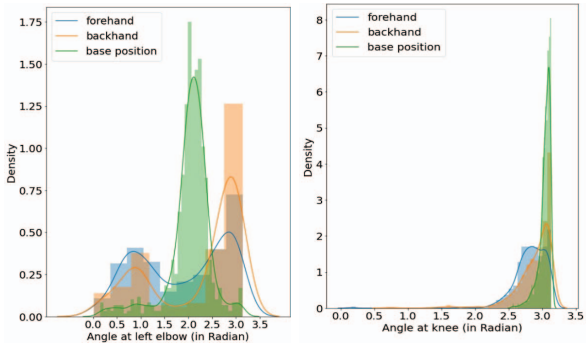
Few density plots for the newly calculated features (e.g., the angles at knee, hips, elbow, shoulder, and the distances between wrists and shoulder) are shown in Fig. 5.

In the second part of our analysis, we derive the performance score of the forehand and the backhand motion of a player. The performance score depicts how the player's



(a) Forehand (b) Backhand (c) Base position

Fig. 4: Visualization of the detected skeleton points for three positions (forehand, backhand and base position) in Tennis



(a) Angle at elbow (Radian) (b) Angle at knee (Radian)

(c) Distance between left-right wrists (d) Distance between left-right shoulders

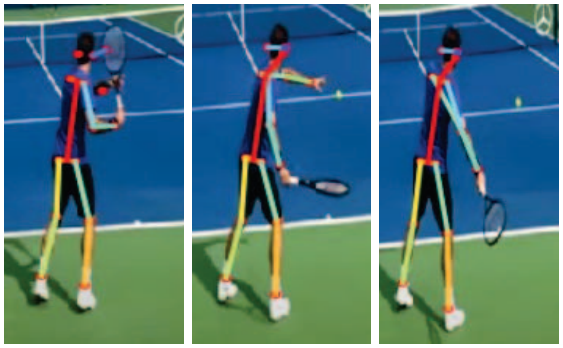
Fig. 5: Visualization of few density plots for the newly computed features

movement in the forehand/backhand motion resembles the actual motion of a professional player, which is used as a benchmark for comparison. We apply the *cosine similarity* (COS) on the two vectors of the datapoints to calculate the performance score, which is a constant value between 0 and 1. Here, 1 indicates perfect shot-making, and 0 indicates no similarity between the two shots/positions being compared.

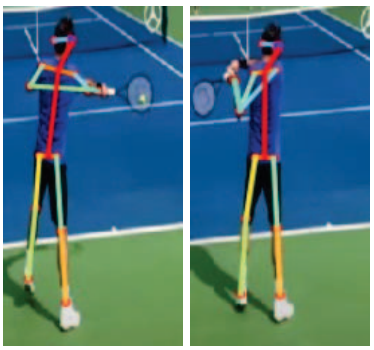
Here, we do a further breakdown of the classes into subcategories that define the critical milestones. The shot motion is broken down into the following subcategories[†].

1) Initialization: The base position transitions into the shot motion where the player prepares themselves for the shot.
2) Racket back: The player's position where the backward motion of the stops and the forward swing for the shot starts. A good racket back position is significant to initiate the shot swing.
3) Swing stage: This is the first swing stage that defines the zone from the racket back position to the contact point. The momentum to hit the ball is gathered in this swing.
4) Contact point: this is the contact point of the racket and the ball.
5) Follow through: This motion follows from the contact point and the end of the shot.

We analyze and give the performance score against these five subcategories respectively (for images, see Figs. 8).



(a) Initialization (b) Racket Back (c) Swing Stage



(d) Contact Point (e) Follow Through

Fig. 6: Visualization of the five stages of a forehand shot in Tennis

Based on the score, an athlete can figure out their strength and weakness and areas of improvement. A pseudo-code

of the working procedure of this paper has been given in Algorithm 1.

Algorithm 1: Working procedure of this paper

```
// Function Definitions
Def: getframe() {return frame one by one}
Def: preprocess_image(){resize to 448 × 448 (RGB)}
Def: detectron2(){return keypoints of
     shape(N × 17 × 3), where N is the no of
     persons In the frame}
Def: sort_by_bbox(){return the keypoint of the
     largest bounding box, return shape(1 × 17 × 3)}
Def: calc_features(){calculate the 14 new features}
Def: predict(){random forest model prediction of the
     action}
Def: check_5_position(){return true if the keypoints
     corresponds to one of the 5 subcategories†
     positions}
// Input and Output
Input: Images or Video of Court view Tennis
Output: Activity Classification Accuracy (%)
// Training
build the Model based on training images;
// Testing and Evaluation
while getframe() != NULL do
    img = getframe();
    img = preprocess_image(img);
    keypoints = detectron2(img);
    keypoints = sort_by_bbox(keypoints);
    newfeatures = calc_features(keypoints);
    combfeatures = combine(keypoints,newfeatures);
    daraframe = build_dataframe(combfeatures);
    stddataframe = StandardScale(dataframe);
    activity_pred = predict(stddataframe);
    if activity_pred == forehand OR backhand then
        // Performance Measures
        if check_5_position(keypoints) == TRUE
         then
            perf_score = COS(benchmark, keypoints);
Return: activity_pred, perf_score
```

B. Experimental Setup

We have used Google Colab for the online coding editor. The Colab provides us with a GPU and CPU with 12GB of RAM. Pytorch 1.7.0+CUDA 10.1 framework and Python 3.7 have been used for coding. The Detectron2 pretrained model for keypoints detection is Mask_rcnn_R_50_FPN_3x. Besides, the proposed pose-based classification accuracy has been compared with different widely used Convolutional Neural Network (CNN) models such as AlexNet, VGG16, ResNet50, MobileNetV2, and EfficientNetb7[1].

V. Result and Analysis

We want to know the dependency of the movements on the features. Hence, we perform feature importance for each of the class labels against the features. Among the newly generated features, the most critical features for the base position are the distance between hips, shoulder, elbow and the angle at the left knee and right shoulder. The essential features for the forehand are the angle on both the shoulders, right elbow, and the distance between wrists. Essential features for the backhand are the distance between the right to left elbow, the angle at the left shoulder and elbow.

When we test our model against the static test images, we get a 98.60% classification accuracy. Furthermore, we feed 6 minutes of videos to the model. Each video clip contains approximately 12K frames. We get an accuracy of 97.50% on the activity classification task. We also tested the performance of the approach against benchmark standard image classification models. We applied transfer learning [15] and fine-tuned standard models on the image data using pre-trained models: AlexNnet[16], ResNet50[17], VGG16[18], MobileNetV2[19], and EfficientNetb7[20][1]. Our approach stands out both in terms of accuracy and training time (refer to Table I). One area where our approach is lagging in the inference speed of a video clip. The probable reason is its compartmentalized approach: Detectron2 is for keypoints estimation, and the machine learning model is for activity classification. An end-to-end framework designing could improve the inference time. Our approach registers the speed of 5 FPS, where the other approaches perform between 8 to 10 FPS. Few classified frames from a given input video[2] has shown in Fig. 7.

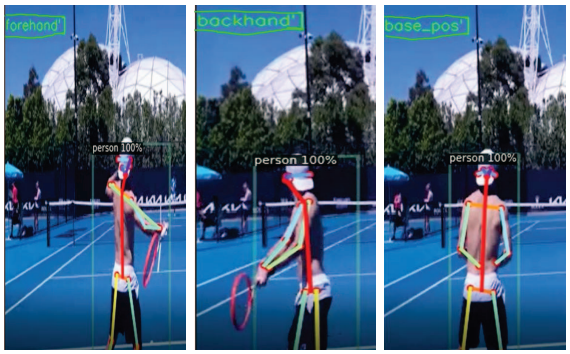
TABLE I: Test classification results obtained from different popular deep learning models and ours proposed approach

Model	Training (Mins.)	Accuracy (%)	FPS
AlexNet	45	93.20	09
VGG16	45	93.50	10
ResNet50	50	95.20	10
MobileNetV2	31	96.80	10
EfficientNetb7	62	97.20	10
Ours (proposed)	12	98.60	05

To analyze the player performance for a movement in a particular subcategory, we get the results as expected. For the analysis of professional players having near-perfect movement, we get a performance score close to above 0.95 for all subcategories while an amateur player's performance score ranges between 0.70 to 0.95, which ascertains our intuition for the working model. An example for the racket back position of forehand motion, Fig. 8(a) is the Benchmark position that is being compared with other photos. A similar positioning of a professional player in Fig. 8(b) yields a performance score of 0.98, while a casual and relaxed

[1]PyTorch EfficientNet: https://github.com/lukemelas/EfficientNet-PyTorch

[2]Tennis action classification: https://www.youtube.com/watch?v=8HjtI9uFPJE



(a) Forward (b) Backhand (c) Base position

Fig. 7: Detected poses using our proposed model obtained from a video clip

(a) Benchmark (b) Professional (c) Casual

Fig. 8: Racket back position for three instances for performance comparison

position in Fig. 8(c) which is not correct, gives a score of 0.92.

VI. Conclusion

Sports analytics is an area that is still in its early stage of evolution, and there is a huge potential for intelligent systems to exploit this domain. We try to formulate a method that could be a starting point to apply intelligent systems in the beautiful game of tennis. All in one, we showed action (activity) classification of different movements in tennis. We engineered features calculated from different body keypoints that define the changing geometry of the body parts in tennis motions, which can draw performance analysis and measuring capability. Moreover, this can be used by a coach or an athlete to pinpoint strengths and weaknesses. It can help the player in the areas of the game; one needs to improve. We believe this will be of immense help to budding players who privilege the high-end training facility provided by elite training academies. Coach Vijay from VMK Tennis Academy has said that solutions like these can be of immense significance to the tennis community, and we are planning to do a demo in his academy soon. It is also satisfactory that the proposed approach achieves 98.6% test accuracy and outperforms popular SOTA CNN image classifiers.

The study can be extended by introducing other activity classification and different motions such as one-handed backhand, serve, drop shots, and keeping left-handed players into account. Another area of improvement is the classification accuracy through sequence learning methods such as different RNN or LSTM variants. The final goal is to develop recommendations based on prescriptive analytics for the performance score.

References

[1] U. S. T. Association, *Coaching tennis successfully*. Human Kinetics, 2004.

[2] H. Brody, *Tennis science for tennis players*. University of Pennsylvania Press, 2010.

[3] V. Sarlis and C. Tjortjis, "Sports analytics—evaluation of basketball players and team performance," *Information Systems*, vol. 93, p. 101562, 2020.

[4] X. Sun, J. Davis, O. Schulte, and G. Liu, "Cracking the black box: Distilling deep sports analytics," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3154–3162.

[5] V. C. Pantzalis and C. Tjortjis, "Sports analytics for football league table and player performance prediction," in *2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA*. IEEE, 2020, pp. 1–8.

[6] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[7] S. Zabihifar, A. Semochkin, E. Seliverstova, and A. Efimov, "Unreal mask: one-shot multi-object class-based pose estimation for robotic manipulation using keypoints with a synthetic dataset," *Neural Computing and Applications*, pp. 1–18, 2021.

[8] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[10] G. Shobana and M. Suguna, "Sports prediction based on random forest algorithm," in *Advances in Materials Research*. Springer, 2021, pp. 993–1000.

[11] R. B. Nelsen, *Proofs without words: Exercises in visual thinking*. MAA, 1993, no. 1.

[12] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian conference on computer vision*. Springer, 2010, pp. 709–720.

[13] M. Kovacs, "Applied physiology of tennis performance," *British journal of sports medicine*, vol. 40, no. 5, pp. 381–386, 2006.

[14] M. S. Kovacs, "Tennis physiology," *Sports medicine*, vol. 37, no. 3, pp. 189–198, 2007.

[15] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[16] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," *arXiv preprint arXiv:1404.5997*, 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[20] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.