Golf Swing Sequencing Using Computer Vision

Article *in* Pattern Recognition and Image Analysis · April 2022

DOI: 10.1007/978-3-031-04881-4_28

2 authors:


Marc Marais
Rhodes University


Dane Lesley Brown
Rhodes University

Golf Swing Sequencing using Computer Vision*

Marc Marais[1[0000−0002−6952−4406]] and Dane Brown[2[0000−0001−7395−7370]]

Department of Computer Science, Rhodes University, Grahamstown, 6140, South Africa
marcmarais07@outlook.com, d.brown@ru.ac.za

Abstract. Analysis of golf swing events is a valuable tool to aid all golfers in improving their swing. Image processing and machine learning enable an automated system to perform golf swing sequencing using images. The majority of swing sequencing systems implemented involve using expensive camera equipment or a motion capture suit. An image-based swing classification system is proposed and evaluated on the GolfDB dataset. The system implements an automated golfer detector combined with traditional machine learning algorithms and a CNN to classify swing events.
The best performing classifier, the LinearSVM, achieved a recall score of 88.3% on the entire GolfDB dataset when combined with the golfer detector. However, without golfer detection, the pruned VGGNet achieved a recall score of 87.9%, significantly better ($> 10.7\%$) than the traditional machine learning models. The results are promising as the proposed system outperformed a Bi-LSTM deep learning approach to achieve swing sequencing, which achieved a recall score of 76.1% on the same GolfDB dataset. Overall, the results were promising and worked towards a system that can assist all golfers in swing sequencing without expensive equipment.

Keywords: Golf Swing Sequencing · Histogram of Oriented Gradients · Linear Discriminate Analysis · Support Vector Machines · Gradient Boosting Machines · Deep Learning.

1 Introduction

Golf is one of the more complex and challenging sports today and is highly popular as more than 80 million people play the sport worldwide [16]. The fine-tuning and complexity behind the golf swing itself comes under much scrutiny. With little research on the effect of different swings, much of the debate regarding the correct body posture and swing technique stems from professional golfers and coaches' personal opinions [20]. Correct body posture and golf ball alignment at each swing event is one of the key elements to a good swing [10, p. 42].

The golf swing action can be recognised using machine learning, deep learning, and neural network algorithms. This study will attempt to detect and classify golf swing events. Golf swing sequencing has recently been achieved using a deep neural network [16]. Their study, which uses a labelled video database to sequence golf swing events, enables head alignment to be checked for synchronisation with the swing sequence events.

Many studies around golf swing sequencing use sensors and optical cameras to detect the swing pattern and events [12, 14, 16]. Sensors and optical cameras with hardware processing are costly implementations to study swing events and sequencing. On the other hand, computer vision and machine learning can analyse golf swing events in a practical way that is accessible to all golfers.

Therefore, a cost-effective and easily accessible solution in the field of computer vision can help aid golfers in the analysis of their golf swing events.

This paper aims to provide an image-based system that implements golf swing event classification for the various phases during a golf swing, using the GolfDB dataset [16]. The events are as follows: address, toe-up, mid-backswing, top, mid-downswing, impact, mid-follow-through and finish. The proposed system incorporates different feature extraction methods, a variety of classification algorithms and a Histogram of Oriented Gradients (HOG) golfer detector. This led to contributions towards applying HOG-based person detection for the purpose of golfer detection.

The rest of the paper is structured as follows: Section 2 analyses related studies, Section 3 and 4 detail the methodology and implementation of the proposed system, including applying the

* This study was funded by the National Research Foundation of South Africa. This work was undertaken in the Distributed Multimedia CoE at Rhodes University.

HOG golfer detector. The analyses of the dimensionality reduction techniques and results of the proposed system are discussed in Section 5. Section 6 concludes the paper and discusses future work.

2 Related Studies

Golf swing analysis by McNally *et al.* [16] aimed to create a sequence of golf swing events through localising each event to a single frame. Events included the address, toe-up, mid-backswing, top, mid-downswing, impact, mid-follow-through and finish. These eight events were analysed and enabled consistent evaluation of the golf swing.

Using a benchmark database GolfDB, consisting of 1400 labelled videos and a combination of SwingNet, McNally *et al.* [16] implemented a lightweight deep learning neural network to analyse the performance of golf swings. SwingNet compromises a network architecture design, incorporating lightweight convolutional neural networks (CNNs) to enable effective mobile deployment.

The CNN, SwingNet, averaged a rate of 76.1% Percentage of Correct Events[1] (PCE) at detecting all eight events in the golf swing. A 91.8% PCE was achieved when detecting 6 out of 8 events, excluding the address and finish events. These events were often misclassified with the start and end frames, respectively. Four random splits were generated for cross-validation to ensure the system generalised well.

Ko and Pan [14] also looked into swing sequencing but included body sway analysis during the swing. Using a single frontal facing camera and a motion capture suit to perform 3D analysis of the swing and body motion. The regression model is based on a Bidirectional Long Short Term Memory[2] (Bi-LSTM) neural network.

Swing events are captured and extracted as a sequence of images using the Sequence Feature Extraction and Classification Network (SFEC-net). SFEC-net is made up of three convolution layers, three pooling layers, and two fully connected layers. After extracting the swing event images, head-up analysis is conducted, creating three-dimensional and three-axis rotation angles of head movement.

Gehrig *et al.* [9] used single frames to robustly fit a golf club's location to a swing trajectory model.

Implementing traditional machine learning algorithms or simple CNNs combined with a histogram of oriented gradients golfer detector appears to be top candidates for yielding the best accuracy performance while minimising computational processing time for the proposed swing event classification system. Therefore, if CNNs were to be implemented, the frame foreground requires focusing purely on the golfer with minimal background variation. If there tends to be a lot of variation in the background and foreground, as is the case with the changing terrain on golf courses and the changing angle of viewing the golf swing, then using state-of-the-art models such as YOLO [17] may be warranted.

3 Methodology

3.1 Methodology Overview

The system is broken down into two stages to detect the eight golf swing events: an image processing stage and a machine learning stage. It was developed to accurately detect each golf swing event from the frames extracted from the GolfDB video dataset. The system will only analyse full golf swings based on the GolfDB dataset.

The main focus of image processing was to detect the golfer in the extracted frames and crop out the region of interest, which consisted of the golfer and the golf club swing region. The different camera views, down-the-line, face-on and other, were taken into account. To detect the golfer in an image, a 90×120 sliding window object detector was combined with the HOG feature descriptor for golfer detection, which is discussed in Section 3.4. If the golfer was located within the sliding

[1] PCE closely relates to the recall, the ratio of the number of true positives to the combined number of true positives and false negatives, metric used to measure machine learning models performance.

[2] Bi-LSTM models fall into the category of Bidirectional Recurrent Neural Networks [18]

window, illustrated in Figure 1a, the image was cropped using the sliding window as a bounding box, illustrated in Figure 1b. Determining the optimal sliding window for the golfer location is based on Non-Maxima Suppression (NMS) window scoring.

(a) Golfer localisation and bounding box sliding window. **(b)** Cropped golfer image, 90×120.

Fig. 1: Automated Golfer Detector.

For the machine learning phase of the system, feature extraction was applied to the data to reduce dimensionality. Fitting and transforming the model using LDA reduces the dimensionality of the data and improves linear separability. The feature extraction stage was not applied to the pruned VGGNet deep learning model.

The proposed system aimed to compare the limits of traditional machine learning models when combined with image processing techniques to the pruned VGGNet as well as the related studies. State-of-the-art detectors for golfer detection were not explored due to the computational power and training time required by these systems. Machine learning algorithms considered were LinearSVM, CatBoost, Decision Tree Classifier, Random Forests and KNN.

3.2 Multi-class Classification

The eight golf swing events are made up as follows:

1. *Address (A)*. The frame before the initial backswing movement begins.
2. *Toe-up (TU)*. Backswing stage where the golf club shaft is parallel to the ground.
3. *Mid-backswing (MB)* Arm is parallel to the ground during the backswing.
4. *Top (T)* The stage when the golf club changes directions and transitions from backswing to downswing.
5. *Mid-downswing (MD)* Arm parallel to the ground during the downswing phase.
6. *Impact (I)* The stage when the golf club head makes contact with the golf ball.
7. *Mid-follow-through (MFT)* Golf club shaft is parallel to the ground during the follow-through phase of the swing.
8. *Finish (F)* The frame before the golfer relaxes their final pose.

3.3 Frame Extraction from GolfDB video dataset

The GolfDB dataset was developed by McNally *et al.* [16] as a benchmark dataset. The dataset consists of 1400 labelled full swing golf video samples. The viewing angles of the videos vary between down-the-line, face-on and other views, illustrated in Figure 2.

The video sample frames were looped over, and where the frame corresponded to an element of the events array[3], the frame was extracted and added to the corresponding event class folder

[3] An array containing ten items each corresponding to the frame of an event, [SF, A, TU, MB, T, MD, I, MFT, F, EF].

(a) Down-the-line View. **(b)** Face-on View. **(c)** Other View.

Fig. 2: The three different golf swing views.

creating the images of the different swing event classes; creating the image-based golf swing event dataset.

This study's main focus was the different viewing angles. Consequently, an attempt was made to build a model that generalises well across the different viewing angles. An 80:20 train test split was implemented for validation and testing data for all test models.

3.4 HOG Golfer Detection

Histogram of Oriented Gradients (HOG) object detection involves machine learning and image processing techniques to detect semantic object instances within an image or video. Popular domains of object detection include face detection or pedestrian detection. HOG is a feature descriptor for images, used for object detection and recognition tasks in computer vision and machine learning [6]. A feature descriptor defines an image by analysing pixel intensities and the gradients of pixels. Through this, the vital information from an image is extracted.

A HOG-based golfer detector can perform person or non-person classification [6]. The HOG-based golfer detector uses a 64×128 sliding window containing the combined HOG feature vectors, fed into a LinearSVM for final classification. There are three elements to the human detector: winStride, padding, and scale. WinStride is a tuple (x, y) that defines the horizontal and vertical shifts of the window. Padding pads the sliding window with pixels and scale refers to the resize factor for each image layer of the resolution pyramid.

A smaller winStride and scale generally generate better results but are more computationally expensive. NMS is used to select the highest scoring window. NMS aims to retain only one window by eliminating low-score windows by thresholding those overlapping. NMS's limitation is that it is a greedy algorithm that does not always find the optimal solution.

3.5 Feature Extraction

Feature extraction is a form of dimensionality reduction where an image with a large number of pixels is efficiently represented so that the informative aspects of the image are represented effectively.

Principal Component Analysis(PCA) involves computing the Eigenvectors using linear algebra and implementing them to transform the basis on the data [22]. Typically the first few principal components are used, as the most significant explained variance is stored in the first components. The total scatter matrix S_T is defined as [2]:

$$S_T = \sum_{k=1}^{N} (\mathbf{x}_k - \mu)(\mathbf{x}_k - \mu)^T \tag{1}$$

where N is the number of sample images x_k, and $\mu \in \mathbb{R}^n$ is the mean image obtained from the sample images.

Linear Discriminate Analysis (LDA) is a generalisation of Fisher's linear discriminant [21] and aims to learn a linear combination of features to consider the between-class and within-class scatter matrix differences.
The between-class scatter matrix S_B for C number of classes is defined as [2]:

$$S_B = \sum_{i=1}^{c} N_i \left(\mu_i - \mu\right)\left(\mu_i - \mu\right)^T \tag{2}$$

and the within-class scatter matrix S_W is defined as:

$$S_W = \sum_{i=1}^{c} \sum_{\mathbf{x}_k \in X_i} \left(\mathbf{x}_k - \mu_i\right)\left(\mathbf{x}_k - \mu_i\right)^T \tag{3}$$

where μ_i is the mean image and N_i refers to the number of samples of class X_i.

3.6 Classification

Based on validation data, classification is implemented to identify which category a new observation belongs to. Multi-class classification is applied to the system as there are more than two classes in the classification problem.

Support Vector Machine (SVM) is a supervised kernel-based machine learning model used for classification and regression, using discriminative classification. The algorithm separates validation data into their respective labelled classes by drawing a hyperplane.

Gradient Boosting Machines (GBMs) are based on a statistical framework to minimise the loss of a numerical optimisation problem through the addition of weak learners using a gradient descent procedure and boosting models [8]. Boosting machine models consists of iteratively learning weak classifiers to a specific distribution of data and adding them to a final robust classifier. Three commonly implemented GBMs are XGBoost, LightGBM and CatBoost [5, 13, 7].

Decision Trees (DTs) are implemented for either classification or regression and form part of non-parametric supervised learning algorithms. Through learning simple decision rules based on the data features, a model is generated to predict the value of a target variable. DTs implement recursive partitioning based on the attribute value learnt from the partition to construct the tree nodes.

Random Forests builds on decision trees through the application of a large number of individual decision trees implemented as an ensemble method [3]. An ensemble method uses multiple learning algorithms to improve predictive performance. Therefore, random forest implements a combination of tree predictors where the values of each tree are dependent on a random vector sampled independently with equal distribution across all trees in the forest.

K-Nearest Neighbors (KNN) classification model is a distance-based supervised learning algorithm where classification is determined through a simple majority vote of the nearest neighbours of each point [1].

Convolutional Neural Networks (CNN) form part of deep neural network algorithms utilising convolutional layers. CNNs are efficient at solving complex image pattern recognition tasks. The architecture of the VGG-16 [19] CNN consists of many convolutional layers containing 3×3 filters. Maximum pooling layers are applied in between the various convolutional layers. The architecture concludes with two fully connected layers followed by a softmax layer for output.

VGG-16 is predominately designed for problems relating to the much larger ImageNet [15] dataset. Therefore, this paper implements a pruned VGGNet [4]. The CNN architecture adopted is a pruned structure using the first 2/5 blocks of the VGG-16, one fully connected layer and a softmax layer. Each block consists of a convolution followed by batch normalisation and a dropout layer of 10%. Dropout regularisation randomly takes each hidden neuron and sets the output to zero based on a certain probability.

Images were augmented to aid the model's generalisation ability on unseen data. The data augmentation generates random transformations with a zoom of ±5% and horizontal and vertical shifts of ±10%. Data augmentation increases the amount of data through minor modifications of the existing data.

4 Evaluation Metrics

Accuracy is defined as the proportion of test samples for which the actual label matched the predicted label [11], the calculation is illustrated in Equation 4.

$$Accuracy = \frac{TruePositives + TrueNegatives}{TruePositives + FalsePositives + TrueNegatives + FalseNegatives} \tag{4}$$

Precision defines the reliability of a classifier's predictions and is a good measure to determine, when the cost of false positives is high [11].

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{5}$$

Recall is a good measure when determining how comprehensive the classifier is in finding cases that are actually positive [11].

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{6}$$

F-score is a combination of precision and recall, representing a harmonic mean between the two [11]. The F-score provides a much more balanced measure of a models performance.

$$F\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

Accuracy works well only if there is an even distribution of samples between the classes. The F-score metric is the preferred metric for measuring a model's performance when presented with an imbalanced dataset.

5 Results

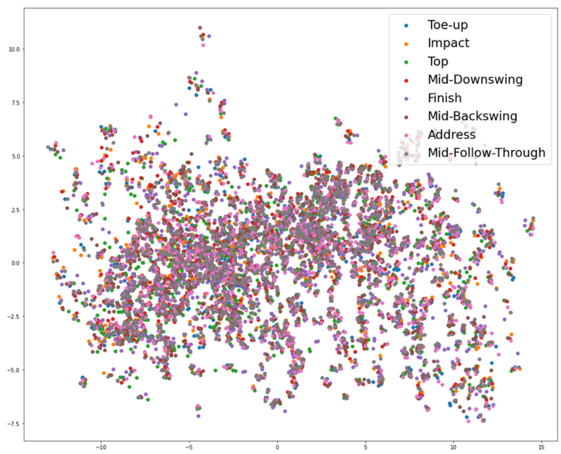
5.1 Validation of Parameters

All classifier and dimensionality reduction parameters were validated using the entire GolfDB dataset, and all the views were included without applying the HOG golfer detector. The implementation was done in this manner to ensure better robustness and generalisation ability.

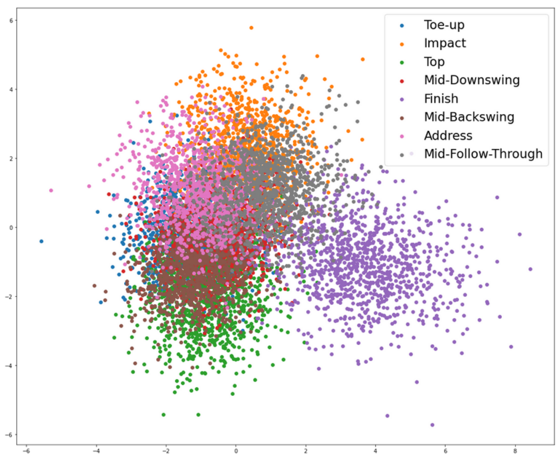
Dimensionality Reduction and Model Selection. Hyperparameter tuning is critical for technical comparison between PCA and LDA dimensionality reduction techniques. The critical parameter for both PCA and LDA is the number of components. Hence, a grid search is implemented to determine the optimal number of components.

PCA poorly separates the different swing event classes, as illustrated in Figure 3a. In contrast, LDA groups the different classes in a linearly separable way, Figure 3b. Therefore, LDA using seven components was chosen as the dimensionality reduction technique for the system. As the LDA transformation was applied to the entire dataset, data leakage[4] occurred in the traditional machine learning models.

[4] Data from outside the training dataset is used to create the model, sharing information between the validation and training data sets [23, p. 93].



(a) Scatter plot of first two components for PCA on the entire GolfDB dataset using 500 components.



(b) Scatter plot of first two components for LDA on the entire GolfDB dataset using 7 components.

Fig. 3: PCA vs LDA dimensionality reduction, on all viewing angles of GolfDB validation dataset.

Classification Model Hyperparameter Tuning. All machine learning classifiers were tuned using a grid search to optimise the hyperparameters. Each grid search applied four-fold cross-validation. To reduce the bias of the classifiers and for comparability to McNally *et al.'s* study [16]. Tables 1 – 5 describe the optimal parameters found for each machine learning classifier.

Table 1: LinearSVM optimal tuning parameters.

C = 10
kernel = linear
multi-class = crammer-singer

Table 2: CatBoost optimal tuning parameters.

learning-rate = 0.03
depth = 4
bagging-temperature = 1.5
l2-leaf-reg = 7

Table 3: Decision Tree Classifier optimal tuning parameters.

min-samples-leaf = 5
min-samples-split = 3

Table 4: Random Forests optimal tuning parameters.

min-samples-leaf = 3
min-samples-split = 3

Table 5: KNN optimal tuning parameters.

```
n-neighbors = 7
p = 2
```

5.2 Test Models

This Section evaluates the various experiments to gauge the overall performance of the golf swing classification system. The system evaluates the multi-class classification on the entire image without implementing the golfer detector. These results are compared to the classification results achieved with the golfer detector.

Experiment 1: Entire GolfDB dataset without golfer detection. This Section evaluates the systems classification strength on the entire GolfDB dataset without golfer detection and cropping of the golfer region of interest. Experiment 1 takes in uncropped images of size 160×160. The multi-class classification problem aims to address the system's generalisation ability by evaluating the system's identification accuracy of all the different viewing angles.

Table 6: Experiment 1 performance Metrics.

Model	Accuracy	Recall	Precision	F-score
LinearSVM	77.1	77.1	77.3	77.2
CatBoost	77.0	77.0	77.3	77.1
Decision Tree	68.3	68.3	68.4	68.3
Random Forests	75.4	75.4	75.7	75.4
KNN	73.1	73.1	73.6	73.3
Pruned VGGNet	87.9	87.9	88.0	87.9

Using F-score, the pruned VGGNet significantly outperformed the LinearSVM machine learning algorithm by 10.7%. Table 6 shows that amongst the traditional machine learning models, the LinearSVM marginally outperformed CatBoost achieving an accuracy of 77.1% compared to the 77.0% of CatBoost. The KNN classifier yielded an accuracy of 73.1%. Random Forests and the decision tree classifier yielded accuracy scores of 75.4% and 68.3%, respectively.

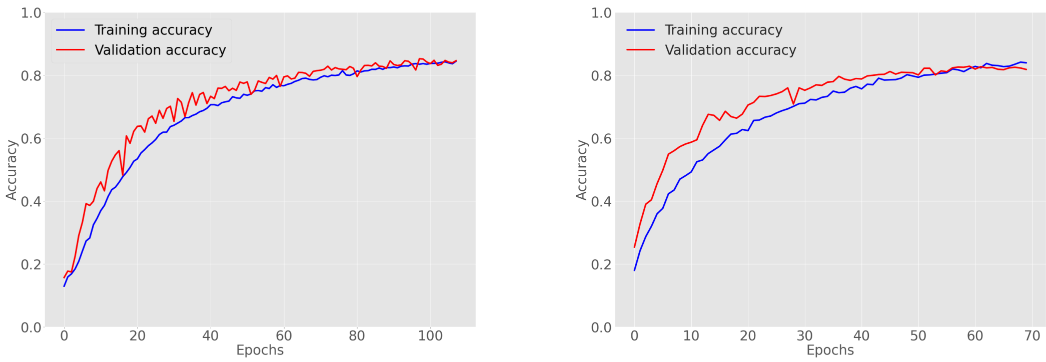
The pruned VGGNet with data augmentation achieved an accuracy score of 87.9, which was 10% better than the pruned VGGNet with no data augmentation.

Experiment 2: Entire GolfDB dataset with golfer detection. The golfer detection crops the region of interest containing the golfer and swing radius, using a sliding window of size 90×120. The golfer detector successfully predicted 7786 images containing a golfer, using a 30% confidence level. Some residual false detections were manually observed using visual inspection. These residual false detections were not manually removed and may negatively affect the subsequent classification results.

Table 7: Experiment 2 performance metrics.

Model	Accuracy	Recall	Precision	F-score
LinearSVM	88.3	88.3	88.3	88.3
CatBoost	87.9	87.9	88.0	87.9
Decision Tree	78.8	78.8	79.2	78.9
Random Forests	86.6	86.6	86.6	86.6
KNN	85.9	85.9	85.9	85.9
Pruned VGGNet	86.3	86.3	86.3	86.3

The best performing model was the LinearSVM, again narrowly outperforming the Catboost classifier with an F-score of 88.3%, compared to the 87.9% F-score achieved by CatBoost, as described in Table 7. Figure 4 illustrates the training and validation accuracy of the pruned VGGNet for experiments 1 and 2.



(a) Experiment 1. **(b)** Experiment 2.

Fig. 4: Training and validation accuracy of pruned VGGNet.

(a) Input Layer. **(b)** First convolutional layer in first block. **(c)** Final dropout layer in second block.

Fig. 5: Heat map activations of the pruned VGGNet showing improved localisation of salient features with the convolutional and dropout layers.

Figure 5 illustrates the heat maps of the activations found through the progression in layers of the pruned VGGNet. Activations were visualised using the Keract[5] python library. The activation on the input layer, Figure 5a, is an example of a poor activation, failing to focus the golfer. However, as the VGGNet progresses through the layers, the localisation of the golfer and swing region vastly improves, as seen in figure 5b and 5c, respectively.

5.3 Discussion

Overall, the multi-class golf swing classification system performed well. The initial hyperparameter tuning to optimise the proposed system's classification models was critical to improving the system's performance. LDA significantly outperformed PCA due to its ability to generate linearly

[5] https://github.com/philipperemy/keract

separable classes. The golfer detector increased the accuracy and F-score of the system by 10%, even though falsely detected ROIs were included.

The pruned VGGNet significantly outperformed the machine learning models in experiment 1. However, when implemented on the golfer detector images, the pruned VGGNet declined in performance, reinforcing the concept that CNNs are effective on busy backgrounds compared to other feature extraction and traditional machine learning approaches.

Table 8: Comparison of the different Swing Sequencing Systems using Recall. The best performing classification algorithm's recall scores were selected from each experiment.

System	**A**	**TU**	**MB**	**T**	**MD**	**I**	**MFT**	**F**	**Recall**
Experiment 1: Pruned VGGNet	93.3	86.5	84.9	86.3	81.8	86.2	89.5	93.7	87.9
Experiment 2: LinearSVM	90.2	83.8	80.2	89.6	88.2	90.1	89.8	92.6	88.3
McNally *et al.* [16]	31.7	84.2	88.7	83.9	98.1	98.4	97.6	26.5	76.1
Ko and Pan [14]	93.9	N/A	96.9	92.2	89.2	84.9	97.0	99.2	93.3

Table 8 compares the the proposed system to McNally *et al.* and Ko and Pan's studies [16, 14]. Experiment 1 and 2 significantly outperformed McNally *et al.'s* system with recall scores of 87.9% and 88.3%, respectively, compared to the 76.1% achieved by McNally *et al.* Both Ko and Pan and McNally *et al.* implemented a time series. However, McNally *et al.'s* system struggled with the address and finish classes due to difficulty precisely localising the two events temporally. Ko and Pan's system benefited from a motion-capture suit. However, the proposed system avoided it to allow for easier access to all golfers without incurring high costs.

The proposed system can generalise well across the various swing event classes. Exception for the consecutive toe-up and mid-backswing classes, classifying a video with time information may improve the classification between consecutive classes. The main area of concern within the system is the golfer detector, which does not handle images with multiple people in the background compared to neural networks. Neural networks are especially effective on busy backgrounds compared to other feature extraction and machine learning approaches. However, the proposed system showed that traditional machine learning algorithms combined with feature extraction can still achieve results on par with or better than deep learning techniques when carried out systematically.

6 Conclusion

Overall the proposed system performed well and, when implemented with no golfer detection, achieved an F-score of 87.9% using the pruned VGGNet. The application of the golfer detection to the GolfDB dataset improved the LinearSVMs F-score by 11.1%, achieving an F-score of 88.3%. The pruned VGGNet declined in performance when implemented with the golfer detector. Misclassifications were mainly present between the toe-up and mid-backswing for the LinearSVM classifier, which was expected due to similarity between the two swing events. However, the proposed system significantly outperformed McNally's *et al.* [16] deep learning study under the same conditions.

This study notably contributed to applying a HOG-based golfer detector for golfer detection to improve event classification, which has not been explored before. The proposed golf swing classification is thus promising as a system that can be made available to golfers and is a starting point towards golf swing form analysis. The system also shows that image processing combined with traditional machine learning models remain relevant and, when implemented effectively, can outperform lightweight deep learning models.

References

1. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician **46**(3), 175–185 (1992)
2. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on pattern analysis and machine intelligence **19**(7), 711–720 (1997)

3. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
4. Brown, D., Bradshaw, K.: Deep palmprint recognition with alignment and augmentation of limited training samples. SN Computer Science **3**(1), 1–17 (2022)
5. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005 **I**, 886–893 (2005). https://doi.org/10.1109/CVPR.2005.177
7. Dorogush, A.V., Ershov, V., Gulin, A.: Catboost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363 (2018)
8. Friedman, J.H.: Stochastic gradient boosting. Computational Statistics & Data Analysis **38**(4), 367–378 (feb 2002). https://doi.org/10.1016/S0167-9473(01)00065-2
9. Gehrig, N., Lepetit, V., Fua, P.: Visual golf club tracking for enhanced swing analysis. In: British Machine Vision Conference (BMVC). pp. 1–10 (2003)
10. Glazier, P., Lamp, P.: Golf Science: Optimum Performance From Tee to Green. In: Golf Science: Optimum Performance From Tee to Green, chap. The swing. University of Chicago Press (2013)
11. Hossin, M., Sulaiman, M.N.: A review on evaluation metrics for data classification evaluations. International journal of data mining & knowledge management process **5**(2), 1 (2015)
12. Hsu, Y.L., Chen, Y.T., Chou, P.H., Kou, Y.C., Chen, Y.C., Su, H.Y.: Golf swing motion detection using an inertial-sensor-based portable instrument. In: 2016 IEEE international conference on consumer electronics-Taiwan (ICCE-TW). pp. 1–2. IEEE (2016)
13. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems **30**, 3146–3154 (2017)
14. Ko, K.R., Pan, S.B.: CNN and bi-LSTM based 3D golf swing analysis by frontal swing sequence images. Multimedia Tools and Applications pp. 8957–8972 (2020). https://doi.org/10.1007/s11042-020-10096-0
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM **60**(6), 84–90 (2017)
16. McNally, W., Vats, K., Pinto, T., Dulhanty, C., McPhee, J., Wong, A.: GolfDB: A video database for golf swing sequencing. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops **2019-June**, 2553–2562 (2019). https://doi.org/10.1109/CVPRW.2019.00311
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
18. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. Signal Processing, IEEE Transactions on **45**, 2673 – 2681 (12 1997). https://doi.org/10.1109/78.650093
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
20. Smith, A., Roberts, J., Wallace, E., Forrester, S.: Professional golf coaches' perceptions of the key technical parameters in the golf swing. In: Procedia Engineering. vol. 34, pp. 224–229. Elsevier Ltd (Jan 2012). https://doi.org/10.1016/j.proeng.2012.04.039
21. Tharwat, A., Gaber, T., Ibrahim, A., Hassanien, A.E.: Linear discriminant analysis: A detailed tutorial. AI communications **30**(2), 169–190 (2017)
22. Wold, S., Esbensen, K., Geladi, P.: Principal Component Analysis. Chemometrics and intelligent laboratory systems **2**(1-3), 37–52 (1987)
23. Zheng, A., Casari, A.: Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. O'Reilly Media, Inc., 1st edn. (2018)