GolfDB: A Video Database for Golf Swing Sequencing

William McNally Kanav Vats Tyler Pinto Chris Dulhanty
John McPhee Alexander Wong
Systems Design Engineering, University of Waterloo
{wmcnally, k2vats, tyler.pinto, chris.dulhanty, mcphee, a28wong}@uwaterloo.ca

Abstract

The golf swing is a complex movement requiring considerable full-body coordination to execute proficiently. As such, it is the subject of frequent scrutiny and extensive biomechanical analyses. In this paper, we introduce the notion of golf swing sequencing for detecting key events in the golf swing and facilitating golf swing analysis. To enable consistent evaluation of golf swing sequencing performance, we also introduce the benchmark database ***GolfDB***,[1] *consisting of 1400 high-quality golf swing videos, each labeled with event frames, bounding box, player name and sex, club type, and view type. Furthermore, to act as a reference baseline for evaluating golf swing sequencing performance on GolfDB, we propose a lightweight deep neural network called SwingNet, which possesses a hybrid deep convolutional and recurrent neural network architecture. SwingNet correctly detects eight golf swing events at an average rate of 76.1%, and six out of eight events at a rate of 91.8%. In line with the proposed baseline SwingNet, we advocate the use of computationally efficient models in future research to promote in-the-field analysis via deployment on readily-available mobile devices.*

1. Introduction

It is estimated that golf is played by 80 million people worldwide [19]. The sport is most popular in North America, where 54% of the world's golf facilities reside [30]. In the United States, the total economic impact of the golf industry is estimated to be $191.9 billion [13]. In Canada, golf has had the highest participation rate of any sport since 1998 [36]. It would be reasonably contended that many golfers are drawn to the sport through the gratification of continuous improvement. The golf swing is a complex full-body movement requiring considerable coordination. As such, it can take years of practice and instruction to develop a repeatable and reliable golf swing. For this reason, golfers routinely scrutinize their golf swing and make frequent adjustments to their golf swing mechanics.

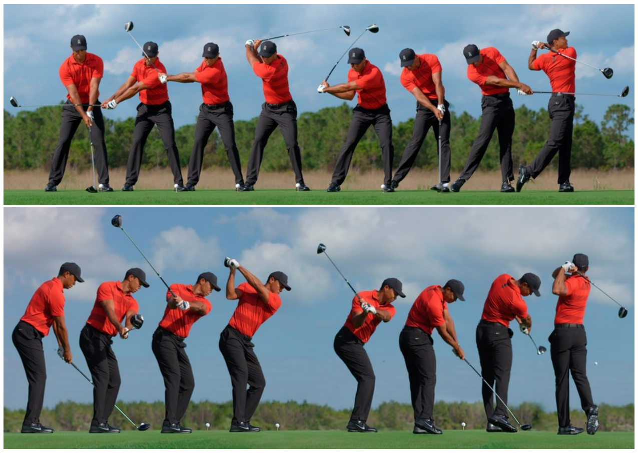

Figure 1: Eight events in a golf swing sequence. Top: face-on view. Bottom: down-the-line view. The names of the events from left to right are *Address*, *Toe-up*, *Mid-backswing*, *Top*, *Mid-downswing*, *Impact*, *Mid-follow-through*, and *Finish*. Images used with kind permission from Golf Digest [14].

Several methods exist for analyzing golf swings. In scientific studies, researchers distill golf swing insights using optical cameras to track reflective markers placed on the golfer [25, 24, 28, 4]. Often, these insights relate to kinematic variables at various *events* in the golf swing. For example, in [28] it was found that torso–pelvic separation at the top of the swing, referred to as the X-factor in the golf community, is strongly correlated with ball speed. Similarly, in [4] it was found that positive lateral bending of the trunk at impact (away from the target) was also correlated with ball speed, as it potentially promotes the upward angle of the clubhead path and more efficient impact dynamics [26]. Yet, examining a golf swing using motion capture requires special equipment and is very time consuming, making it impractical for the everyday golfer.

Traditionally, professional golf instructors provide instant feedback to amateurs using the naked eye. Still, the underlying problem is not always immediately apparent due to the speedy nature of the golf swing. Consequently, slow-motion video has become a popular medium for dissect-

[1] Available at https://github.com/wmcnally/GolfDB

ing the intricacies of the golf swing [15]. Moreover, slow-motion video is readily available to the common golfer using the advanced optical cameras in today's mobile devices, which are capable of recording high-definition (HD) video at upwards of 240 frames per second (fps). Given a slow-motion recording of a golf swing, a golfer or golf instructor may scrub through the video to analyze the subject's biomechanics at various key events. These events comprise a golf swing *sequence* [14]. For example, in the golf swing sequence of Tiger Woods depicted in Fig. 1, the X-Factor at the top of the swing and lateral bending of the trunk at impact, two strong indicators of a powerful golf swing, are easily identifiable in the face-on view. Still, scrubbing through a video to identify these events is time consuming and impractical because only one event can be viewed at a time.

In computer vision, deep convolutional neural networks (CNNs) have recently been shown to be highly proficient at video recognition tasks such as video representation [3], action recognition [27], temporal action detection [42], and spatio-temporal action localization [7]. Following this line of research, CNNs adapted for video may be leveraged to facilitate golf swing analysis through the autonomous extraction of event frames in golf swing videos. To this end, we introduce **GolfDB**, a benchmark video database for the novel task of *golf swing sequencing*. A total of 1400 HD golf swing videos of male and female professional golfers, comprising various native frame-rates and over 390k frames, were gathered from YouTube. Each video sample was manually annotated with eight event labels (event classes shown in Fig. 1). Furthermore, the dataset also contains bounding boxes and labels for club type (*e.g.*, driver, iron, wedge), view type (face-on, down-the-line, or other), and player name and sex. With this supplemental data, GolfDB creates opportunities for future research relating to general recognition tasks in the sport of golf. Finally, we advocate mobile deployment by proposing a lightweight baseline network called **SwingNet** that correctly detects golf swing events at a rate of 76.1% on GolfDB.

2. Related Work

2.1. Computer Vision in Golf

Arguably the most well-known use of computer vision in golf deals with the real-time tracing of ball flights in golf broadcasts [8]. The technology uses difference images between consecutive frames and a ball selection algorithm to artificially trace the path of a moving ball. In a different light, radar vision is used in the TrackMan launch monitor to precisely track the 3D position of a golf ball in flight and measure its spin magnitude [38].

Computer vision algorithms have also been implemented for analyzing golf swings. Gehrig *et al.* [9] developed an algorithm that robustly fit a global swing trajectory model to club location hypotheses obtained from single frames. Fleet *et al.* [39] proposed incorporating dynamic models with human body tracking, and Park *et al.* [29] developed a prototype system to investigate the feasibility of pose analysis in golf using depth cameras. In line with these research directions, GolfDB may be conveniently extended in the future to include various keypoint annotations to support human pose and golf club tracking. Moreover, automated golf swing sequencing using SwingNet is complementary to pose-based golf swing analysis.

2.2. Action Detection

In the domain of action detection, there exist several subproblems that correspond to increasingly complex tasks.

Action recognition is the highest-level task and corresponds to predicting a single action for a video. The first use of modern CNNs for action recognition was by Karpathy *et al.* [21], wherein they investigated multiple methods of fusing temporal information from image sequences. Simonyan and Zisserman [35] followed this work by incorporating a second stream of optical flow information to their CNN. A different approach to combine temporal information is to use 3D CNNs to perform convolution operations over an entire video volume. First implemented for action recognition by Baccouche *et al.* [2], 3D CNNs showed state-of-the-art performance in the form of the C3D architecture on several benchmarks when trained on the Sports-1M dataset [37]. Combining two-stream networks and 3D CNNs, Carreira and Zisserman [3] created the I3D architecture. Recurrent neural networks (RNNs) provide a different approach to combining temporal information. RNNs with long short-term memory (LSTM) cells are well suited to capture long-term temporal dependencies in data [16] and these networks were first used for action recognition by Donahue *et al.* [6] in the long-term recurrent convolutional network (LRCN), whereby features extracted from a 2D CNN were passed to an LSTM network. A similar method to Donahue *et al.* is adopted in this work.

Temporal action detection is a mid-level task wherein the start and end frames of actions are predicted in untrimmed videos. Shou *et al.* [32] proposed the Segment-CNN (S-CNN) in which they trained three networks based on the C3D architecture. In a different approach, Yeung *et al.* [41] built an RNN that took features from a CNN and utilized reinforcement learning to learn a policy for identifying the start and end of events, allowing for the observation of only a fraction of all video frames.

Spatio-temporal action localization is the lowest level and most complex task in action detection. Both the frame boundaries and the localized area within each frame corresponding to an action are predicted. Several works in this domain approach the problem by combining 3D CNNs with object detection models, such as in [11], where the I3D model is combined with Faster R-CNN, and in [17], where the authors generalize the R-CNN from 2D image regions to 3D video *tubes* to create Tube-CNN (T-CNN).

2.3. Event Spotting

After asking Amazon Mechanical Turk workers to re-annotate the temporal extents of human actions in the Charades [34] and MultiTHUMOS [40] datasets, Sigurdsson *et al.* [33] demonstrated that the temporal extents of actions in video are highly ambiguous; the average agreement in terms of temporal Intersection-over-Union (tIOU) was only 72.5% and 58.7%, respectively. This raised concerns over the inherent error associated with temporal action detection.

Considering the uncertainty surrounding the temporal extents of actions, Giancola *et al.* [10] proposed the task of *event spotting* within the context of soccer videos, arguing that in contrast to actions, events are anchored to a single time instance and are defined using a specific set of rules. As opposed to predicting the temporal bounds of an action, *spotting* consists of finding the time instance (or *spot*) when well-defined events occur. We consider event *spotting* and event *detection* as equivalent terminology, and use the latter moving forward.

3. Golf Swing Sequencing

Drawing inspiration from Giancola *et al.* [10], we describe the task of golf swing sequencing as the detection of events in *trimmed* videos containing a single golf swing. The reasoning behind using trimmed golf swing videos is three-fold:

1. We speculate that the most compelling use-case for detecting golf swing events is to obtain instant biomechanical feedback in the field, as opposed to localizing golf swings in broadcast video. Although GolfDB contains the necessary data to perform spatio-temporal localization of full golf swings in untrimmed video, we consider this a separate task and a potential avenue for future research.

2. Golfers or golf instructors who wish to view a golf swing sequence in the field can simply record a constrained video of a single golf swing on a mobile device, ensuring that the subject is centered in the frame. This eliminates the need for spatio-temporal localization.

3. A video sample containing a single golf swing instance will consist of a specific number of events occurring in a specific order. This information can be leveraged to improve detection performance.

Golf Swing Events. In [10], soccer events were resolved at a one-second resolution. In contrast, golf swing events can be localized to a single frame using strict event definitions. Although various golf swing events have been proposed in the literature [23], we define the eight contiguous events comprising a golf swing sequence as follows:

1. *Address (A).* The moment just before the takeaway begins, *i.e.*, the frame before movement in the backswing is noticeable.

2. *Toe-up (TU).* Shaft parallel with the ground during the backswing.

3. *Mid-backswing (MB).* Arm parallel with the ground during the backswing.

4. *Top (T).* The moment the golf club changes directions at the transition from backswing to downswing.

5. *Mid-downswing (MD).* Arm parallel with the ground during the downswing.

6. *Impact (I).* The moment the clubhead touches the golf ball.

7. *Mid-follow-through (MFT).* Shaft parallel with the ground during the follow-through.

8. *Finish (F).* The moment just before the golfer's final pose is relaxed.

The above definitions do not always isolate a single frame. For example, a golfer may not hold a finishing pose at all, making the selection of the *Finish* event frame subjective. These issues are discussed further in the next section.

4. GolfDB

In this section we introduce GolfDB, a high-quality video dataset created for general recognition applications in golf, and specifically for the novel task of golf swing sequencing. Comprising 1400 golf swing video samples and over 390k frames, GolfDB is relatively large for a specific domain. To our best knowledge, GolfDB is the first substantial dataset dedicated to computer vision applications in the sport of golf.

4.1. Video Collection

A collection of 580 YouTube videos containing real-time and slow-motion golf swings was manually compiled. For the task of golf swing sequencing, it is important that the shaft remains visible at all times. To alleviate obscurities caused by motion blur, only high quality videos were considered. The YouTube videos primarily consist of professional golfers from the PGA, LPGA and Champions Tours, totalling 248 individuals with diverse golf swings. The

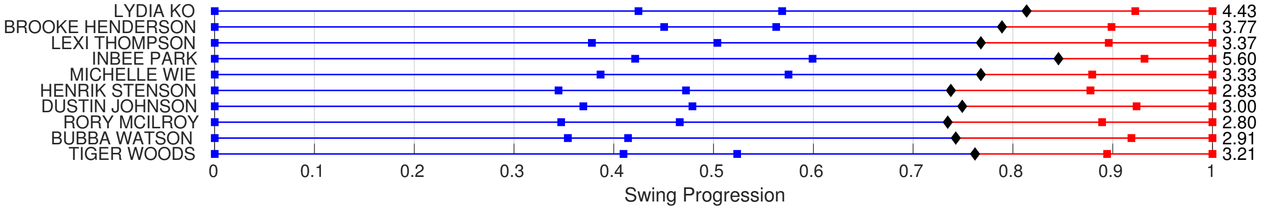

Figure 2: Average event timings from *Address* to *Impact* for 5 female (top 5) and male (bottom 5) professional golfers using a driver or fairway wood. Event timings normalized from *Address* to *Impact*. The diamond indicates the *Top* event. Blue represents the backswing, red represents the downswing. Average golf swing tempos (backswing duration/downswing duration) shown on the right.

videos were captured from a variety of camera angles, and a variety of locations on various golf courses, including the driving range, tee boxes, fairways, and sand traps. The significant variance in overall appearance, taking into consideration the different players, clubs, views, lighting conditions, surroundings, and native frame-rates, benefits the generalization ability of computer vision models trained on this dataset. The YouTube videos were sampled at 30 fps and 720p resolution.

4.2. Annotation

A total of 1400 trimmed golf swing video samples were extracted from the collection of YouTube videos using an in-house MATLAB code that was distributed to four annotators. For each YouTube video, the annotators were asked to identify full golf swings (*i.e.*, excluding pitch shots, chips, and putts) and label 10 frames for each: the start of the sample, eight golf swing events, and the end of the sample. The number of frames between the start of the sample and *Address*, and similarly, between *Finish* and the end of the sample, was naturally random, and the beginning of samples occasionally included practice swings. Depending on the native frame-rate of the sample, it was not always possible to label events precisely. For example, in real-time samples with a native frame-rate of 30 fps, it was rare that the precise moment of impact was captured. The annotators were advised to chose the frame closest to when the event occurred at their own discretion.

Besides labeling events, the annotators were asked to draw bounding boxes, enter the club and view type, and indicate whether the sample was in real-time or slow-motion (considering a playback speed of 30 fps). The bounding boxes were drawn to include the clubhead and golf ball through the full duration of the swing. Player names were extracted from the video titles, and sex was determined by cross-referencing the player name with information available online. The annotators were briefed on domain-specific knowledge before the annotation process, and the dataset was verified for quality by an experienced golfer. Fig. 2 provides the average timing of events from *Address* to *Impact* for five male and female golfers using a driver or fairway wood, and Fig. 3 illustrates the distribution of the dataset.

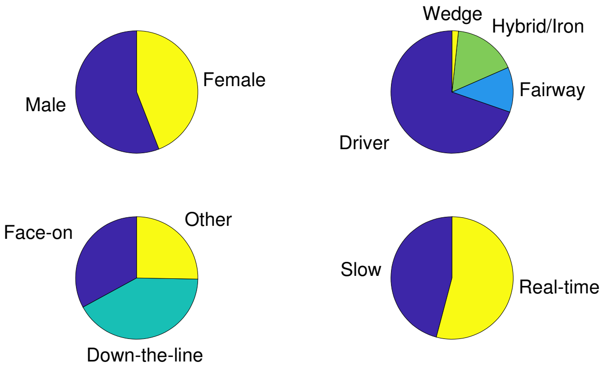

Figure 3: Distribution of GolfDB. The total number of frames in real-time and slow-motion samples was roughly equal (≈195k each). The event densities for real-time and slow-motion samples were 3.072×10^{-2} and 2.628×10^{-2} events/frame, respectively.

4.3. Evaluation Metric and Experimental Protocol

In a similar fashion to Giancola *et al.* [10], we introduce a tolerance δ on the number of frames within which an event is considered to be correctly detected. In the real-time samples, if an event was thought to occur between two frames, it was at the discretion of the annotator to select the event frame. Given the inherent variability of the annotator's selection, we consider a tolerance $\delta = 1$ for real-time videos sampled at 30 fps. For the slow-motion videos, the tolerance should be scaled based on the native frame-rate, but the native-frame rates are unknown. We therefore propose a sample-dependent tolerance based on the number of frames between *Address* and *Impact*. This value was approximately 30 frames on average for the real-time videos, matching the frame-rate of 30 fps (*i.e.*, the average duration of a golf swing from *Address* to *Impact* is roughly 1s). Thus, we define the sample-dependent tolerance as

$$\delta = \max(\left\lfloor \frac{n}{f} \right\rceil, 1) \qquad (1)$$

where n is the number of frames from *Address* to *Impact*, f is the sampling frequency, and $\lfloor x \rceil$ rounds x to the nearest integer.

Drawing inspiration from the field of human pose estimation, we introduce the PCE evaluation metric as the "Percentage of Correct Events" within tolerance. PCE is the temporal equivalent to the popular PCKh metric used in human pose estimation [1], which scales a spatial detection tolerance using head segment length. For the experimental protocol, four random splits were generated for cross-validation, ensuring that all samples from the same YouTube video were placed in the same split. PCE averaged over the 4 splits is used as the primary evaluation metric.

5. SwingNet: A Swing Sequencing Network

In this section, we describe SwingNet, a network architecture designed specifically for the task of golf swing sequencing, but generalizable to the swings present in various sports, such as baseball, tennis and cricket. Additionally, the implementation details are discussed.

5.1. Network Architecture

MobileNetV2 is a CNN based on an inverted residual structure and makes liberal use of lightweight depthwise convolutions [18]. As such, it is well suited for mobile vision applications. Furthermore, MobileNetV2 includes a "width multiplier" that scales the number of channels in each layer, providing a convenient trade-off for network complexity and speed. For the task of image classification, it runs at 75ms per frame on a single core of the Google Pixel using an input size of 224×224 and width multiplier of 1 [31]. Placing emphasis on mobile deployment, we employ MobileNetV2 [31] as the backbone CNN in an end-to-end network architecture that maps a sequence of $d \times d$ RGB images $\mathbf{I} = (\mathbf{I_1}, \mathbf{I_2}, ..., \mathbf{I_T} : \mathbf{I_t} \in \mathbb{R}^{d \times d \times 3})$ to a corresponding sequence of event probabilities $\mathbf{e} = (\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_T} : \mathbf{e_t} \in \mathbb{R}^{C})$, where T is the sequence length and C is the number of event classes. For the task of golf swing sequencing, there are 9 event classes: eight golf swing events and one *No-event* class.

Detecting golf swing events using a single frame would likely be a difficult task. Precisely identifying *Address* requires knowledge of when the *actual* golf swing commences. Without this contextual information, *Address* may be falsely detected during the pre-shot routine, which often includes full or partial practice swings, and frequent clubhead waggling. In a similar manner, precisely identifying *Top* is generally not possible using a single frame, based on its event definition. Moreover, *Mid-backswing* and *Mid-downswing* are relatively similar in appearance. For these reasons, temporal context is likely a critical component in the task of golf swing sequencing. To capture temporal information, the sequence of feature vectors $\mathbf{f} = (\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_T} : \mathbf{f_t} \in \mathbb{R}^{1280})$ obtained by applying global average pooling to the final feature map in MobileNetV2 is used as input to an N-layer bidirectional LSTM [16] with H hidden units in each layer. At each frame t, the H-dimensional output of the LSTM is fed through a final fully-connected layer, and a softmax is applied to obtain the class probabilities $\mathbf{e}$. The weights of the fully-connected layer are shared across frames. The overall architecture is illustrated in Fig. 4. In Section 6.1, an ablation study is performed to identify a suitable model configuration.

5.2. Implementation Details

The frames were cropped using the golf swing bounding boxes. They were then resized using bilinear interpolation such that the longest dimension was equal to d, and padded using the ImageNet [5] pixel means to reach an input size of $d \times d$. Finally, all frames were normalized by subtracting the ImageNet means and dividing by the ImageNet standard deviation.

The network was implemented using PyTorch version 1.0. Each convolutional layer in MobileNetV2[2] is followed by batch normalization [20]. Unique to MobileNetV2, ReLU is used after batch normalization everywhere except for the final convolution in the inverted residual modules [31], where no non-linearity is used. The running batch norm parameters were updated using a momentum of 0.1. The MobileNetV2 backbone was initialized with weights pre-trained on ImageNet, and the weights in the fully-connected layer were initialized following Xavier Initialization [12]. Given the significant class imbalance between the golf swing events and the *No-event* class (roughly 35:1), a weighted cross-entropy loss was used, where the golf swing events were each given a weight of 1, and the *No-event* class was given a weight of 0.1.

Training samples were drawn from the dataset by randomly selecting a start frame and, if the number of frames remaining in the sample was less than T, the sample was looped to the beginning. Randomly selecting the start frame serves as a form of data augmentation and is commonly used in action recognition applications [3, 27]. Other forms of data augmentation used included random horizontal flipping, and slight random affine transformations ($-5°$ to $+5°$ shear and rotation). The intent of the horizontal flipping was to even out the imbalance between left- and right-handed golfers, and the intent of the affine transformations was to capture variations in camera angle. To enable training on batches of image sequences, multiple sequences of length T were concatenated along the channel dimension before being input to the network. The output features $\mathbf{f}$ were reshaped to (batch size, T, 1280) before being passed to the LSTM. Various batch sizes and sequence lengths T were

[2] PyTorch implementation of MobileNetV2 available at https://github.com/tonylins/pytorch-mobilenet-v2

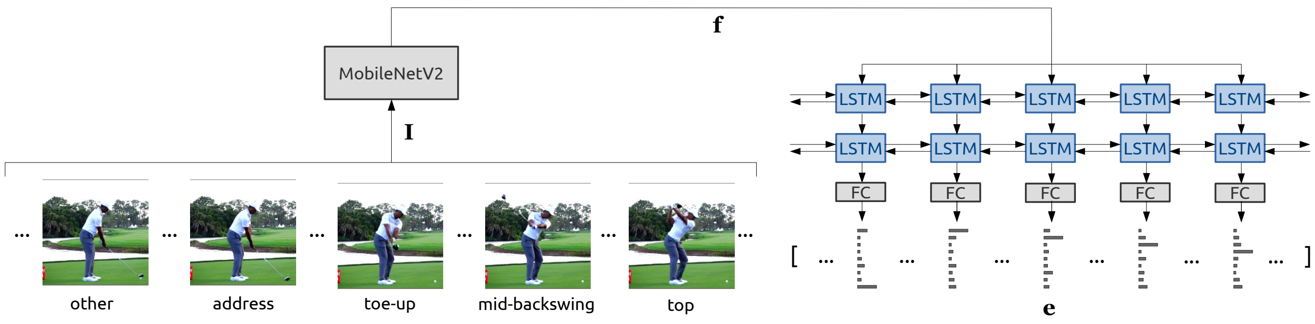

Figure 4: The network architecture of SwingNet, a deep hybrid convolutional and recurrent network for swing sequencing. In an end-to-end architectural framework, SwingNet maps a sequence of RGB images **I** to a corresponding sequence of event probabilities **e**. The sequence of feature vectors **f** generated by MobileNetV2 are input to a bidirectional LSTM. At each frame t, the LSTM output is fed into a fully-connected layer, and a softmax is applied to obtain the event probabilities.

explored in the experiments; they are explicitly declared in Section 6. In all training configurations, the Adam optimizer [22] was used with an initial learning rate of 0.001. The number of training iterations and learning rate schedules are discussed in Section 6. All training was performed on a single NVIDIA Titan Xp GPU.

At test time, a sliding window approach is used over the full-length golf swing video samples. The sliding window has a size T and, to minimize the computational load, no overlap was used. Knowing that each sample contains exactly eight events, event frames were selected using the maximum confidence for each event class. The exploration of alternative selection criteria that leverages event order is encouraged in future research.

6. Experiments

In this section, an extensive ablation study is performed to determine suitable hyper-parameters. Following the ablation study, a final baseline SwingNet is proposed and evaluated.

6.1. Ablation Study

The hyper-parameters of interest are the input size (d), sequence length (T), batch size, number of LSTM layers (N), and number of hidden units in each LSTM layer (H). It was also of interest to see whether initializing with pre-trained ImageNet weights was advantageous as opposed to training from scratch, and if LSTM bidirectionality had an impact. The goal was to identify a computationally efficient configuration to maximize performance with limited compute resources. Normally, MobileNetV2's width multiplier would be a key hyper-parameter to include in the ablation study; however, for reasons to be discussed, using a width multiplier other than 1 was not feasible. Table 1 provides the PCEs of 11 model configurations, along with the number of parameters and floating point operations (FLOPs) for each. Each configuration was trained for 10k iterations on the first split of GolfDB, providing a proxy for overall performance. For the ablation study, no affine transformations were used, and the learning rate was held constant at 0.001. Hyper-parameters used in comparison are shown in bold.

Remarkably, it was found that the model could not be trained effectively unless the pre-trained ImageNet weights were used. We acknowledge that if Configuration 1 were trained longer, it may have eventually converged to a similar level of performance. In any case, we speculate that using weights pre-trained on large and diverse image datasets is critical in domain-specific tasks where the variation in overall appearance is minimal. Because the pre-trained weights were only available for a width multiplier of 1, adjusting the width multiplier was not feasible.

Another interesting finding was that the correlation between input size and performance did not behave as expected. With increasing input size, one would expect a monotonically increasing PCE. However, it was found that that input sizes of 160 and 128 outperformed 192 by a large margin, and the PCE using an input size of 160 was only 4.4 points worse than the 224.

The sequence length T had a significant impact on performance, supporting the importance of temporal context. Additionally, increasing the batch size improved performance dramatically. Still, it is difficult to fairly assess these hyper-parameters as they may have had a significant impact on convergence speed. Regarding the LSTM, bidirectionality led to a 12.1 point improvement in PCE. The single-layer LSTM outperformed the two-layer LSTM, and 256 hidden units performed better than 64 and 128.

6.2. Frozen Layers

The results of the ablation study revealed that increasing the sequence length and batch size provided large performance gains. On a single GPU, the sequence length and batch size are severely limited. Knowing that the model relies heavily on pre-trained ImageNet weights, we hypothe-

Config.	Input Size (d)	Seq. Length (T)	Batch Size	ImageNet Weights	LSTM Layers (N)	LSTM Hidden (H)	Bidirect.	Params (10^6)	FLOPs (10^9)	PCE (10k iter.)
0	**224**	32	6	**Yes**	**2**	**128**	**Yes**	4.07	10.32	66.8
1	224	32	6	**No**	2	128	Yes	4.07	10.32	1.5
2	224	32	6	Yes	2	128	**No**	3.08	10.26	54.7
3	**192**	32	6	Yes	2	128	Yes	4.07	7.62	45.7
4	**160**	**32**	**6**	Yes	2	128	Yes	4.07	5.33	62.4
5	**128**	32	6	Yes	2	128	Yes	4.07	3.45	57.7
6	160	**64**	6	Yes	2	128	Yes	4.07	10.65	71.1
7	160	32	**12**	Yes	2	128	Yes	4.07	5.33	70.1
8	224	32	6	Yes	**1**	128	Yes	3.67	10.39	69.4
9	224	32	6	Yes	2	**64**	Yes	3.01	10.26	66.9
10	224	32	6	Yes	2	**256**	Yes	6.96	10.51	69.3

Table 1: Hyper-parameter search for the proposed SwingNet. Each configuration was trained for 10k iterations on the first split of GolfDB, providing a proxy for final performance. Parameters used in comparison are in **bold**.

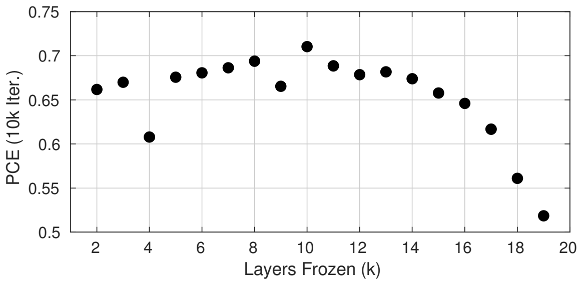

Figure 5: PCE after 10k iterations using configuration 4 from Table 1 and freezing the ImageNet weights in the first k layers. Freezing the first 10 layers provided optimal results.

sized that some of the ImageNet weights could be frozen without a significant loss in performance. Thus, we experimented with freezing layers in MobileNetV2 prior to training to create space in GPU memory for larger sequence lengths and batch sizes. Fig. 5 plots the results of freezing the first k layers using configuration 4 from Table 1. Despite a few outliers, the PCE increased until $k = 10$ and then began to decrease. We leverage this finding in the next section to train a baseline SwingNet using a larger sequence length and batch size.

6.3. Baseline SwingNet

The results in Table 1 suggest that an input size of 160 is more cost effective than an input size of 224; the latter requires double the computation for a relative increase in PCE of just 7%. Taking this into consideration, as well as the comparative results of the other hyper-parameters, we propose a baseline SwingNet with an input size of 160, sequence length of 64, and a single-layer bidirectional LSTM with 256 hidden units. By initializing with pre-trained ImageNet weights and freezing 10 layers, the model can be trained using a batch size of 24 on a single GPU with 12GB memory. With this relatively large batch size, the model converges faster and does not need to be trained for as many iterations. Thus, the baseline SwingNet was trained for 7k iterations on each split of GolfDB, and the learning rate was reduced by an order of magnitude after 5k iterations. Affine transformations were used to augment the training data (see Section 5.2).

Table 2 provides the event-wise and overall PCE averaged over the 4 splits of GolfDB. Overall, SwingNet correctly detects events at a rate of 76.1%. As expected, relatively poor detection rates were observed for the *Address* and *Finish* events. This was likely caused by the compounding factors of subjective labeling and the inherent difficulty associated with precisely localizing these events temporally. These factors may have also played a role in detecting the *Top* event, which was detected in real-time videos more consistently than in slow-motion videos; in slow-motion, the exact frame where the club changes directions is difficult to interpret because the transition is more fluid. Moreover, the detection rate was generally lower in slow-motion videos for the backswing events. This was likely due to the fact that the backswing is much slower than the downswing, so there are more frames in the backswing that are similar in appearance to the ground-truth event frames.

Impact was the event detected the most proficiently. This result is intuitive because the model simply needs to detect when the clubhead is nearest the golf ball. Interpreting when the arm or shaft is parallel with the ground, which is required for events like *Toe-up* and *Mid-backwing*, requires more abstract intuition. Disregarding the *Address* and *Finish* events, the overall PCE was 91.8%. Fig. 6 illustrates the inferred event probabilities for a slow-motion swing. Within a 5-frame tolerance, SwingNet correctly detected all but the *Address* and *Finish* events, which were off by 10 and

Model	A	TU	MB	T	MD	I	MFT	F	PCE
SwingNet-160 (slow-motion)	23.5	80.7	84.7	75.7	97.8	98.3	98.0	21.5	72.5
SwingNet-160 (real-time)	38.7	87.2	92.1	90.8	98.3	98.4	97.2	30.7	79.2
SwingNet-160	31.7	84.2	88.7	83.9	98.1	98.4	97.6	26.5	76.1

Table 2: Event-wise and overall PCE averaged over the 4 splits for the proposed baseline SwingNet. Configuration: bidirectional LSTM, $d = 160$, $T = 64$, $L = 1$, $N = 256$, $k = 10$. This configuration has 5.38×10^6 parameters and 10.92×10^9 FLOPs.

Seq. Length (T)	FLOPs (10^9)	CPU (ms)*	PCE
64	10.92	10.6	76.2
32	5.41	10.8	74.0
16	2.70	11.5	71.0
8	1.35	12.0	66.0
4	0.68	13.8	63.1

Table 3: SwingNet-160 performance and CPU runtime on GolfDB split 1 using various sequence lengths at test time. *Effective processing time for a single frame, excluding I/O, using an Intel i7-8700K processor.

7 frames from their respective ground-truth frames.

Table 3 demonstrates that, after being trained using a sequence length of 64, smaller sequence lengths can be used at test time with only a modest decrease in performance. This has implications to mobile deployment, where smaller sequence lengths may be leveraged to reduce the memory requirements of the network.

7. Conclusion

This paper introduced the task of golf swing sequencing as the detection of key events in trimmed golf swing videos. The purpose of golf swing sequencing is to facilitate golf swing analysis by providing instant feedback in the field through the automatic extraction of key frames on mobile devices. To this end, a golf swing video database (GolfDB) was established to support the task of golf swing sequencing. Advocating mobile deployment, we introduce SwingNet, a lightweight baseline network with a deep hybrid convolutional and recurrent network architecture. Experimental results showed that SwingNet detected eight golf swing events at an average rate of 76.1%, and six out of eight events at a rate of 91.8%. Besides event labels, GolfDB also contains annotations for golf swing bounding boxes, player name and sex, club type, and view type. We provide this data with the intent of promoting future computer vision research in golf, such as the spatio-temporal localization of golf swings in untrimmed broadcast video, and view type recognition.

Acknowledgments. We acknowledge the NVIDIA GPU Grant Program, and financial support from the Canada Research Chairs program and the Natural Sciences and Engineering Research Council of Canada (NSERC).

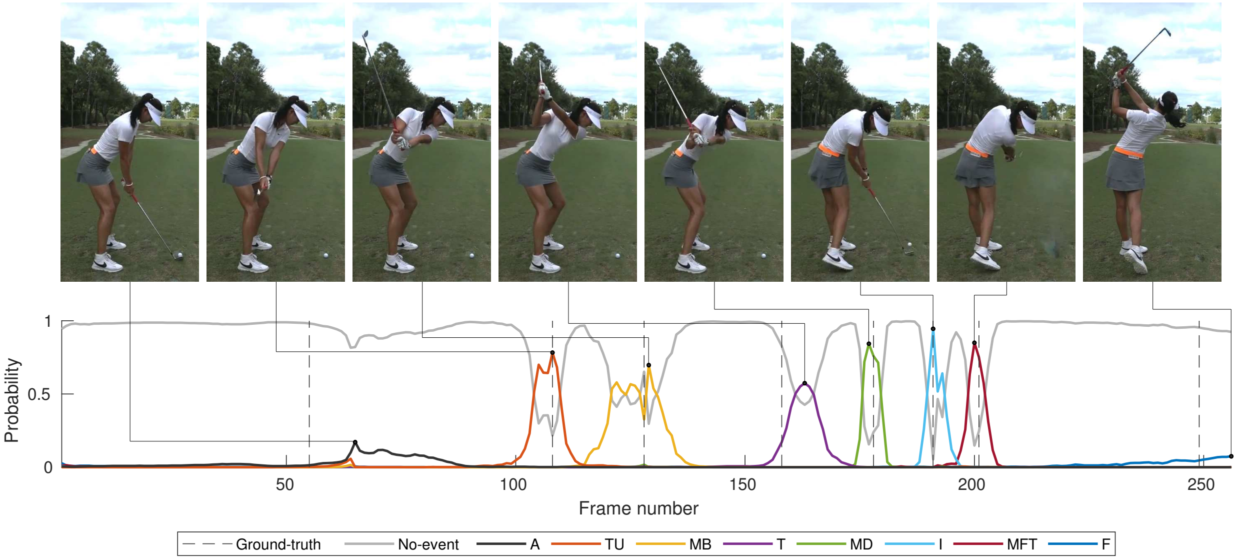

Figure 6: Using SwingNet to infer event probabilities in the slow-motion golf swing of LPGA Tour player Michelle Wie. Six out of eight events were correctly detected within a 5-frame tolerance. *Address* and *Finish* were missed by 10 and 7 frames, respectively. Best viewed in color. Video available at https://youtu.be/Q1KodM7RhH4?t=36.

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, June 2014. 5

[2] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *International Workshop on Human Behavior Understanding*, pages 29–39. Springer, 2011. 2

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2, 5

[4] Y. Chu, T. C. Sell, and S. M. Lephart. The relationship between biomechanical variables and driving performance during the golf swing. *Journal of sports sciences*, 28(11):1251–1259, 2010. 1

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 5

[6] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2

[7] A. El-Nouby and G. W. Taylor. Real-time end-to-end action detection with two-stream networks. *arXiv preprint arXiv:1802.08362*, 2018. 2

[8] D. Forsgren. Systems and methods for enhancing images in a video recording of a sports event. US Patent 8,077,917 B2. 2

[9] N. Gehrig, V. Lepetit, and P. Fua. Visual golf club tracking for enhanced swing analysis. In *BMVC*, 2003. 2

[10] S. Giancola, M. Amine, T. Dghaily, and B. Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *CVPR Workshops*, 2018. 3, 4

[11] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. A better baseline for ava. *arXiv preprint arXiv:1807.10066*, 2018. 3

[12] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 5

[13] Golf 2020. 2016 golf economy report. Retrieved February 11, 2019, from https://golf2020.com/research/. 1

[14] Golf Digest. Golf Swing Sequences: Tips, Instruction, Pro Swings, July 2018. Retrieved February 12, 2019, from https://www.golfdigest.com/golf-instruction/swing-sequences. 1, 2

[15] M. Guadagnoli, W. Holcomb, and M. Davis. The efficacy of video feedback for learning the golf swing. *Journal of sports sciences*, 20(8):615–622, 2002. 2

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2, 5

[17] R. Hou, C. Chen, and M. Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *CVPR*, 2017. 3

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 5

[19] HSBC. Golf's 2020 vision: The HSBC report, 2012. Retrieved February 11, 2019, from http://www.golf.org.au/site/_content/document/00017543-source.pdf. 1

[20] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5

[21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 2

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[23] Y.-H. Kwon, K. H. Han, C. Como, S. Lee, and K. Singhal. Validity of the x-factor computation methods and relationship between the x-factor parameters and clubhead velocity in skilled golfers. *Sports Biomechanics*, 12(3):231–246, 2013. 3

[24] S. M. Lephart, J. M. Smoliga, J. B. Myers, T. C. Sell, and Y.-s. Tsai. An eight-week golf-specific exercise program improves physical characteristics, swing mechanics, and golf performance in recreational golfers. *The Journal of Strength & Conditioning Research*, 21(3):860–869, 2007. 1

[25] A. M. Burden, P. N. Grimshaw, and E. S. Wallace. Hip and shoulder rotations during the golf swing of sub-10 handicap players. *Journal of sports sciences*, 16(2):165–176, 1998. 1

[26] W. McNally and J. McPhee. Dynamic optimization of the golf swing using a six degree-of-freedom biomechanical model. In *Proceedings*, volume 2, page 243, 2018. 1

[27] W. McNally, A. Wong, and J. McPhee. Star-net: Action recognition using spatio-temporal activation reprojection. *arXiv preprint arXiv:1902.*, 2019. 2, 5

[28] J. Myers, S. Lephart, Y.-s. Tsai, T. Sell, J. Smoliga, and J. Jolly. The role of upper torso and pelvis rotation in driving performance during the golf swing. *Journal of sports sciences*, 26(2):181–188, 2008. 1

[29] S. Park, J. Yong Chang, H. Jeong, J.-H. Lee, and J.-Y. Park. Accurate and efficient 3d human pose estimation algorithm using single depth images for pose analysis in golf. In *CVPR Workshops*, 2017. 2

[30] R&A. Golf around the world 2017. Retrieved February 11, 2019, from https://www.randa.org/. 1

[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 5

[32] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016. 2

[33] G. A. Sigurdsson, O. Russakovsky, and A. Gupta. What actions are needed for understanding human actions in videos? In *CVPR*, 2017. 3

[34] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 3

[35] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2

[36] Statistics Canada. Sport participation 2010, February 2014. Retrieved February 11, 2019, from

http://publications.gc.ca/collections/collection_2013/pc-ch/CH24-1-2012-eng.pdf. 1

[37] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *CVPR*, 2015. 2

[38] F. Tuxen. Determination of spin parameters of a sports ball. US Patent 8,845,442. 2

[39] R. Urtasun, D. J. Fleet, and P. Fua. Monocular 3d tracking of the golf swing. In *CVPR*, 2005. 2

[40] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 126(2-4):375–389, 2018. 3

[41] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016. 2

[42] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *CVPR*, 2017. 2