# AI Golf Swing Analyzer

Zihan Yi
Stanford University
yizihan@stanford.edu

## Abstract

*Golf is a sport famous for its complexity and need for continuous, diligent practice to perfect the various swing techniques. The deficiency of immediate feedback for each swing in conventional golf classes created a demand for an immediate swing training tool. Ideally, our proposal involves a mechanism that uses novice golf swing videos to identify flaws and offer real-time feedback. However, given the project's time limitations and the small team size, our main attention was on distinguishing between amateur and professional golf swings using a single video. The procedure we adopted included breaking down the input video into several essential swing postures, and then applying an existing AI framework to accentuate human skeletal points within these specific swings. We then input these pre-processed swing frames into a ResNet for binary output. This method delivered an impressive accuracy of 99% on our test set, indicating a promising solution for a real-time, efficient golf swing analysis tool, if we choose to expand to a comprehensive experience in the future. Code, sample datasets, and pre-trained models are available at* https://github.com/harryyizihan/ai_golf_swing.

## 1. Introduction

In this project, our goal is to determine whether a recorded golf swing, as depicted in a single video, is either at an amateur or professional standard. We've trialed numerous strategies, including those detailed in the AI Golf paper [5] as well as our own original methods. These techniques entail: 1. Directly inputting the raw swing video into a vision transformer, then replacing the final layer with a linear layer and a softmax classifier to derive the output result. 2. Initially dissecting the input video and extracting only the critical golf swing frames using GolfDB's SwingNet [8], then subsequently inputting these frames into a neural network like ResNet [3] to get a binary result. 3. Starting with the same video dissection process, but with an added emphasis on highlighting the skeletal points of the human body using **OpenMM's MMPose library** [1] [1] prior to inputting it into the neural network. For this third technique, we deliberated whether to remove all background, showing only the skeleton points, or to maintain the original scenery. In the end, we found that both the second and third strategies delivered encouraging results, securing an accuracy of over 98% in the test set.

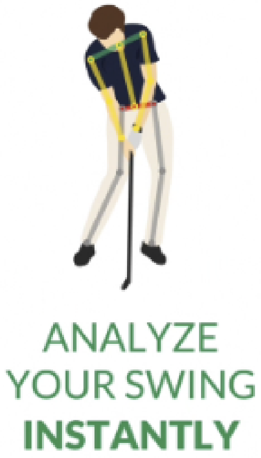

Figure 1. From the App Store page of existing AI golf training apps.

## 2. Related Work

### 2.1. AI Golf Paper

The paper [5] has significantly influenced our project workflow. Its aim is to create a tool for analyzing golf swings, aiding users in recognizing the distinctions between their techniques and those of professional players. The procedure it uses is segmented into three stages: synchronization, discrepancy identification, and manipulation. We find that the synchronization and discrepancy identification stages integrate effortlessly into our workflow.

Firstly, consider that people may execute a golf swing

[1] https://mmpose.readthedocs.io/en/latest/

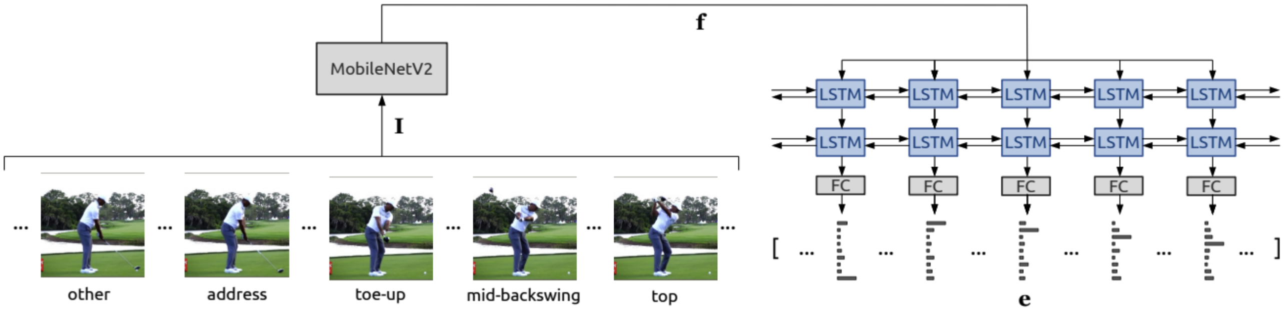

Figure 2. The network architecture of SwingNet, a deep hybrid convolutional and recurrent network for swing sequencing.

at various moments in the video. The motion synchronizer has the capacity to align movements that occur at different phases and times, allowing the subsequent neural network to concentrate more effectively on the crucial parts of the video. Secondly, they also ventured the hypothesis that using skeleton inputs could outperform current video methodologies. This hypothesis encouraged us to examine the strategy of utilizing skeletal points.

## 2.2. Vision Transformer (ViT)

The Vision Transformer (ViT) [2] model is a state-of-the-art deep learning architecture for computer vision tasks. It applies the Transformer architecture, originally developed for natural language processing, to process visual data. ViT divides an image into patches, which are then transformed into sequences of tokens. By leveraging self-attention mechanisms, ViT captures global and local relationships within the image. It has achieved impressive performance on various image classification tasks, rivaling convolutional neural networks (CNNs) and demonstrating the potential of using transformer-based models for computer vision.

## 2.3. SwingNet: Swing Sequences Synchronizer

We utilized an existing pretrained model, SwingNet, created by GolfDB [8], for the aspect of synchronizing swing sequences. As depicted in Figure 2, from an all-encompassing architectural framework standpoint, SwingNet translates a sequence of RGB images $I$ into a corresponding series of event probabilities $e$. The series of feature vectors $f$ produced by MobileNetV2 [6] are fed into a bidirectional LSTM [4]. For each frame $t$, the LSTM output is directed into a fully-connected layer, with a softmax then applied to derive the event probabilities. SwingNet yielded a total of eight sequences, which are arranged in chronological order in Table 1. Additional examples from these results will be given in the dataset section.

| Swing Position | Defining Feature |
|---|---|
| P1 | Address |
| P2 | Toe-up |
| P3 | Mid-backswing (Arm-parallel) |
| P4 | Top |
| P5 | Mid-downswing (Arm-parallel) |
| P6 | Impact |
| P7 | Mid-followthrough |
| P8 | Finish |

Table 1. 8 sequences of a golf swing generated by the SwingNet

## 2.4. MMPose: 3D Human Pose Inference

MMPose is an open-source pose estimation toolkit that uses Pytorch and is part of the OpenMMLab Project. The library includes a wide range of algorithms designed to identify the positions of 133 keypoints on the human body. We discovered that it was exceptionally efficient in highlighting the skeletal points of the human body throughout all the important swing frames we obtained in the previous step. During this stage, we put forth two different approaches for the experiment after producing the skeletal points: one strategy involved directly superimposing the skeletal points on the original image, while the other method displayed them on a new, blank white background.

## 2.5. ResNet: Residual Network

With all the frames pre-processed, we aim to employ a neural network to train a binary classifier that will predict whether a swing originates from an amateur or professional level. We can treat this as an image classification task and intend to tackle it using two widely recognized and established neural networks. ResNet [3], an abbreviation for Residual Networks, was launched by Microsoft Research in 2015. The key innovation of ResNet is its inception of ”skip connections” or ”shortcut connections”, facilitating the gradient’s direct backpropagation to earlier layers. This inno-

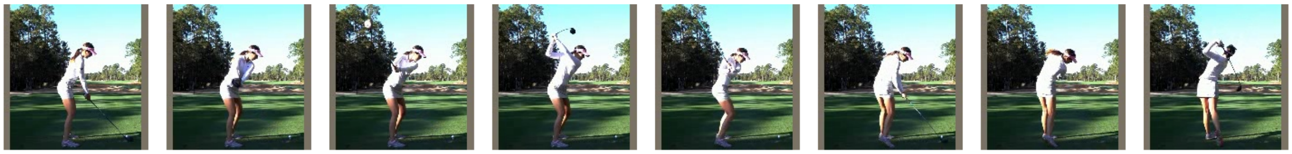

Figure 3. from left to right, SwingNet predicts the eight key frames with a confidence score

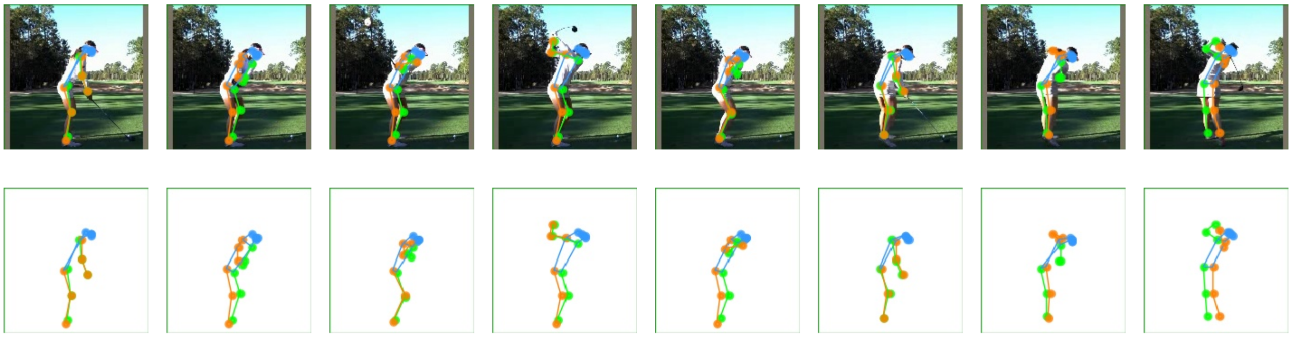

Figure 4. Top 8 frames are with original background, and bottom 8 frames are constructed by a 160x160 white background image

vation enables the training of very deep networks (such as those with over 100 layers) without falling into the vanishing gradient issue. ResNet models have achieved significant success and have clinched numerous competitions and accolades.

# 3. Dataset

## 3.1. Video Dataset

We have compiled 1,400 videos featuring professional golf swings from GolfDB [2]. This dataset has undergone preprocessing and been trimmed and resized to a frame size of 160x160. A smaller frame size for any given image equates to reduced resolution. We use this method to standardize our image dataset and maintain its compact size, which promotes faster training operations.

We plan to use this as our positive dataset for the binary task. Regarding the negative dataset, composed of amateur swings, as an enthusiastic golfer, I have collected 500 videos of my own golf swings and those of my friends. I attempted to accumulate data from YouTube, but it was challenging to trim and capture the exact swing duration, which usually only lasts a few seconds in videos that span several minutes. In the end, we chose to apply `RandomHorizontalFlip(p=1)` to create mirror images as part of the data augmentation process, aiming to ensure a balance between the positive and negative data. This rationale is backed by the consideration that a left-handed golfer could be mirrored as a right-handed golfer, as demonstrated in Figure 5. We continue to seek equilibrium between the positive and negative datasets by implementing `Pad()` with varying border lengths. Ultimately, we end up with 1,400 professional golfer swings and 1,400 amateur golfer swings (comprised of 500 original, 500 mirrored, and 400 padded swings).

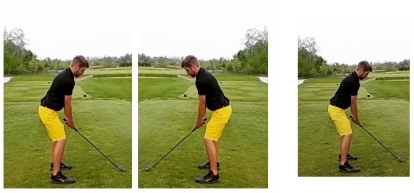

Figure 5. The data augmentation process for amateur swings is as follows, from left to right: the original image, a mirrored image created by a random horizontal flip, and an image with white padding added to the border.

## 3.2. Eight Frames of a Golf Swing

As mentioned previously, we used SwingNet to divide the input video into eight essential golf swing positions in Figure 3. The resulting images might seem a bit blurred due to the earlier preprocessing, which downscaled them to a size of 160x160.

## 3.3. 3D Pose Skeleton Points Inference

Subsequently, we use the MMPose library to emphasize the skeletal points within those eight sequences. We present

[2] `https://github.com/wmcnally/golfdb`

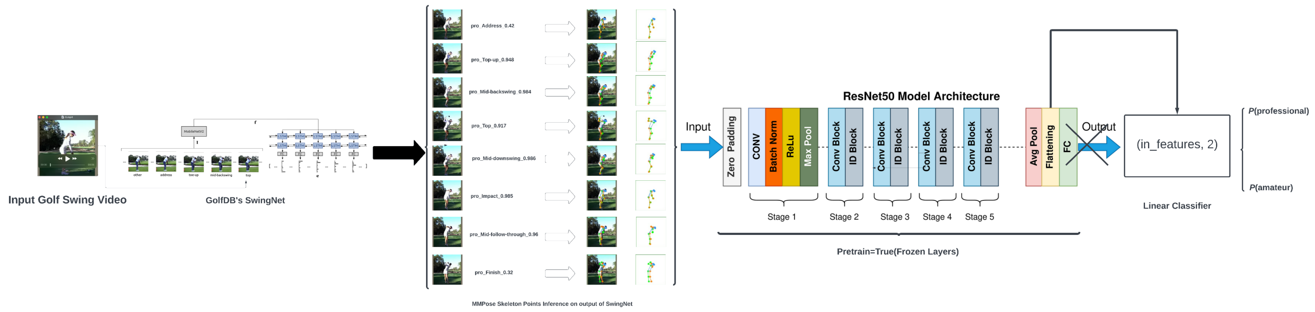

Figure 6. SwingNet + MMPose + ResNet diagram, mentioned in section 4.3 for more details

two versions: one maintaining the original background, and another on a white backdrop. We plan to assess their performance under identical algorithmic conditions to ascertain whether the inclusion of a background image impacts the model's prediction abilities in Figure 4.

# 4. Methods

## 4.1. Baseline: Vision Transformer

The fundamental principle of this technique involves interpreting the video as a series of pictures. The aim is to extract 10 frames from each video using two suggested methods: the first method involves obtaining every other frame at uniform intervals, like every 30th frame. If the video's length results in collecting fewer than 10 frames, padding is done with zeros to compensate all the uncollected frames. The second method involves grouping video frames into sets of 10 and taking the mean of each group. These preprocessed frames are then fed into the Vision Transformer. The `vit_b_16(pretrained=True)` model is specifically employed, and the final layer is modified by integrating a new linear layer. This fresh linear layer has `in_features = 768` (matching the original output size of the Vision Transformer) and `out_features = 2`. The goal is to fine-tune just the last linear layer for the binary task.

$$\text{BCE}(y, \hat{y}) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (1)$$

We use binary cross entropy as the lost function 1. Here, In this equation: $y$ represents the true label (either 0 or 1) indicating golf swing is at professional or amateur level. $\hat{y}$ represents the predicted probability (between 0 and 1) for the positive class. Compared to normal cross entropy function, the `BCEWithLogitsLoss()` function is advantageous as it avoids the numerical instability that can occur when applying the sigmoid activation and calculating the binary cross entropy loss separately.

## 4.2. SwingNet Eight Frames with ResNet

In this strategy, we employ the output from SwingNet, which comprises eight frames of swing positions. These frames are combined horizontally into a single elongated image using `torch.hstack()`. This resultant image is then supplied to a neural network, which is ResNet-50 with `pretrained=True`. We adapt the network's final layer by incorporating a linear layer that ties into the binary cross-entropy loss, replicating the process mentioned previously.

## 4.3. SwingNet + 3D Pose Skeleton Points Frames with ResNet

This method, much like the one before, leverages additional processed images that include the resulting swing position and highlight the skeletal points of the human body. Two variations are presented: one retaining the original background, and the other with a white backdrop. These images are then fed into a neural network in a manner akin to the procedure outlined in the first method. ResNet-50 will be served as the neural network, which is followed by a binary classifier and binary cross-entropy loss to generate predictions. The common rationale behind both the first and second methods is to steer the model's focus towards significant frames representing pivotal golf swing positions and critical image crops within each frame. By concentrating specifically on the human body, vital attributes like the angle between the legs and hips or the stability of the head during the swing can be captured.

# 5. Experiments and Discussion

The experiments begin with the use of the Adam optimizer with a default learning rate of 0.001. The size of the mini-batches is established as 16. The `num_epochs` is designated as 10. A score greater than 0.5 is set as the threshold for positive prediction, and the outcomes of these settings are detailed in Table 2.

|  | Method Description | Test Set Accuracy |
|---|---|---|
| 1 | ViT with fetching frames in every 30 other frames, zero padding if needed | 0.635 |
| 2 | ViT with averaging collection of frames based on video length | 0.959 |
| 3 | ResNet w/ SwingNet frames | 0.984 |
| **4** | **ResNet w/ skeleton points + original background** | **0.996** |
| 5 | ResNet w/ skeleton points + white background | 0.984 |

Table 2. All methods and experiment results

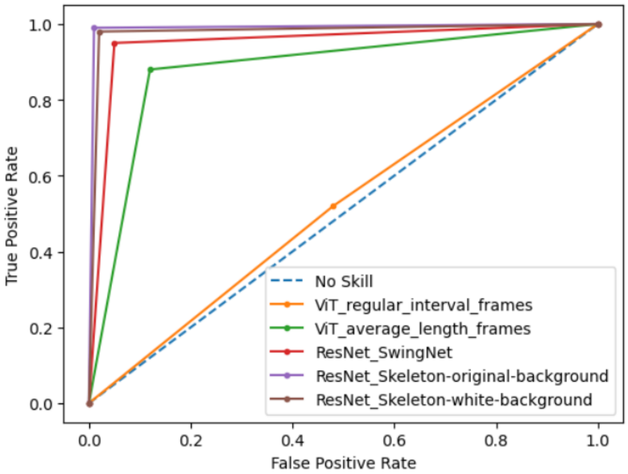

Figure 7. ROC curve comparison beteen ViT and ResNet experiments

## 5.1. ROC curve

From the results gathered, all methods, with the exception of the first one, show commendably high accuracy, going beyond 95%. The use of SwingNet's key position frames plays a substantial role in reaching an accuracy around 98%. The decision to not use skeletal points or to use a white background does not cause significant changes in accuracy. As per the ROC curve plot 7 shows, it's evident that the ViT model, which chooses frames at regular intervals, closely aligns with the central diagonal line. This suggests limited learning during the training stage, leading to difficulties in effectively differentiating between the positive and negative classes, thereby causing a high misclassification rate. On the other hand, the ResNet model showcases curves in the top-left corner, indicative of high accuracy along with a high true positive rate.

## 5.2. Saliency Map

In order to better comprehend how the model interprets data, saliency maps [7] were utilized to emphasize the critical regions in the images for making predictions. These maps consistently highlight the significance of the human body's outline, aligning with key principles of a golf swing such as maintaining head stability and preserving specific angles in the arms and legs.

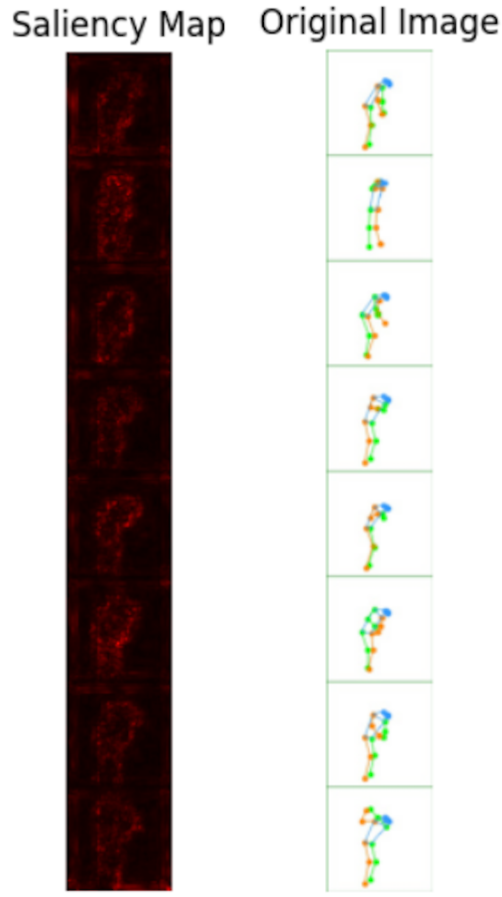

Figure 8. Using saliency map to highlight the most important regions. It provides insights into which parts of the image the model focuses on when making predictions.

## 5.3. Best Threshold Score Search

We performed some hyperparameter tunings to determine the best threshold score for positive predictions using our binary classifier, as illustrated in Figure 9. In this particular scenario, professional golfers are renowned for the consistency of their swings, thus it's improbable to encounter a professional golfer misrepresenting their swing to an amateur level. Conversely, avid amateur golfers, who regularly

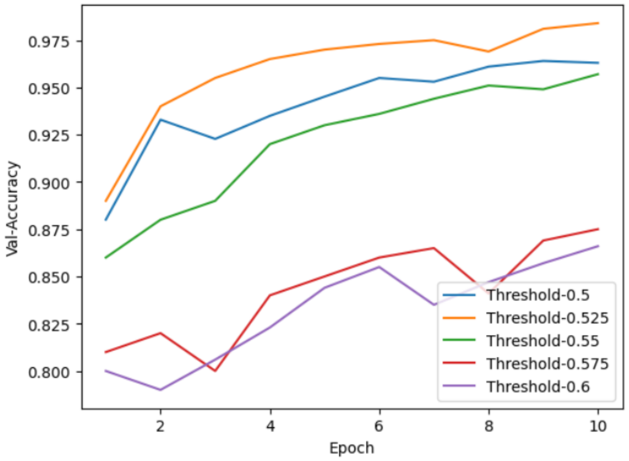

Figure 9. Hyperparameter tuning: finding the threshold score that achieves the best validation accuracy, looking at the graph, the optimal threshold score is around 0.5 to 0.525.

participate in golf classes aiming to perfect their swings, may successfully mimic the swing of a professional golfer once in a while. Overall, false positives are more expected to occur than false negatives. As such, it seems logical to slightly raise the threshold score just above 0.5.

### 5.4. SwingNet Low Confidence Scores Drop

| | Accuracy |
|---|---|
| SwingNet with original 8 frames | 0.984 |
| SwingNet dropping lowest 1 frame | 0.913 |
| SwingNet dropping lowest 2 frames | 0.855 |

Table 3. Experiments of SwingNet's confidence scores

The GolfDB paper indicates that while the accuracy for identifying all eight swing positions stands at 76.1%, it can increase to 91.8% for six out of the eight events. This led us to consider whether omitting the one or two lowest-confidence swing positions could bolster accuracy. After holding all other parameters constant and training the model using three distinct strategies, as outlined in Table 3, the results turned out to be surprisingly contrary to our expectations. We examined the final training image prior to its input into the neural network and discovered that the lowest-ranked frames can differ from one sample to another. For instance, one image might eliminate the 'Top' swing position, causing a direct transition from 'Mid-backswing' to 'Mid-downswing', which is nearly identical in terms of normal golf swing sequence and could lead to model confusion. Furthermore, since we concatenate all frames into a single long image in sequence, they are trained to examine the edges of your human body relative to a specific position in the long image to determine if your swing is correct. However, if we merely discard the frames and concatenate the remaining ones, the altered order could cause mismatches and lead the prediction astray.

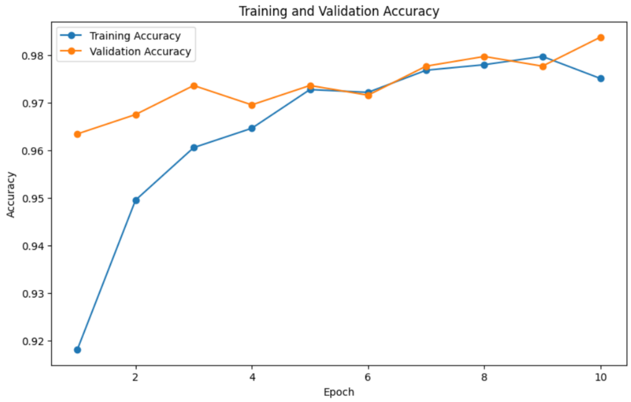

Figure 10. ResNet 3 experiments average training accuracy and validation accuracy

### 5.5. High Initial Validation Accuracy

Finally, it's important to note the relatively high initial training and validation accuracy shown in Figure 10. While randomness and dataset balance were maintained, the extraordinary initial validation accuracy can be credited to the impactful selection of SwingNet and the advanced pretraining of the ResNet architecture. This could also be due to the limited size of the dataset, stemming from the unique problem setting, and the effective preprocessing employed.

## 6. Conclusion

In summary, each algorithm tested, including ViT, SwingNet, and skeleton points inference, surpassed the 95% accuracy mark when appropriate feature construction settings were applied. Among them, the ResNet model, in combination with SwingNet's eight frames and skeleton points inference under the original background, attained the peak accuracy of 99.6%. We also explored various hyperparameters, such as the threshold score and the possibility of discarding the lowest confidence scores coming from SwingNet, but we didn't venture into adjusting fundamental hyperparameters like learning rate or mini-batch size. The reason behind this is that we've already achieved impressive accuracy with the default settings, which might partly be due to the scarcity yet well-preprocessed datasets in general. Given more time, we'd aim to gather more data to initially establish our training and validation accuracy at a reasonably low level. We would then further expand our project to offer specific swing improvement suggestions to amateurs by identifying their particular swing flaws.

# References

[1] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 1

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 1, 2

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. 2

[5] Chen-Chieh Liao, Dong-Hyun Hwang, and Hideki Koike. How can i swing like pro?: Golf swing analysis tool for self training, 2021. 1

[6] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019. 2

[7] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014. 5

[8] T. Pinto C. Dulhanty J. McPhee W. McNally, K. Vats and A. Wong. Golfdb: A video database for golf swing sequencing. *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 13(1):2553–2562, 2019. 1, 2