

Evaluating Temporal Sequence Modeling for Swimming Stroke Classification in Video-Based Human Action Recognition

Robert Thomas Nitto
Information Sciences, Syracuse University
Spring 2026

Abstract—Swimming stroke classification is a specialized human action recognition task that requires modeling both spatial appearance and temporal motion patterns from video sequences. This study evaluates the impact of temporal sequence modeling on swimming stroke classification using a controlled comparison of three CNN-RNN hybrid architectures: CNN-LSTM, CNN-GRU and CNN-BiGRU. A pretrained ResNet18 network is used to extract spatial features from fixed-length video clips while recurrent neural networks modeled temporal dependencies across frame sequences. To evaluate generalization across unseen recordings, a Leave-One-Video-Out Cross-Validation framework was used with swimming stroke clips extracted from publicly available instructional videos. All experiments shared identical preprocessing, feature extraction and training procedures allowing differences in performance to be primarily attributed to temporal modeling architecture. Results showed that all models learned meaningful temporal representations despite the limited and highly variable dataset. The CNN-BiGRU model achieved the most balanced classification performance across all stroke classes. The CNN-LSTM produced similar results however the CNN-GRU demonstrated greater class confusion. Overall classification accuracy was moderate however the findings suggest that bidirectional temporal modeling can improve swimming stroke representation learning and provide a foundation for future work using larger and more diverse swimming video datasets.

Keywords—human action recognition, computer vision, video classification, swimming stroke classification, deep learning, convolutional neural networks, recurrent neural networks, long-short term memory, gated recurrent unit, bidirectional GRU.

I. INTRODUCTION

Human action recognition is an important area of computer vision with applications in sports analytics, athlete evaluation and rehabilitation. Unlike image recognition that relies on static visual information, action recognition requires models to learn both spatial features and temporal patterns from sequences in video frames. Recent advances in deep learning have significantly improved video classification performance by combining convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

Swimming stroke classification represents a specialized human action recognition problem where subtle differences in motion, stroke cadence and swimmer positioning distinguish the target classes. Accurate classification of swimming stroke has potential applications in coaching, athlete monitoring and race performance analysis. Swimming videos also present unique

challenges due to variations in camera angle, swimmer technique, water conditions and recording quality.

CNN-RNN hybrid architectures provide a framework for addressing these challenges. CNNs can be used to extract spatial features from individual video frames and leverage recurrent networks to model temporal dependencies across frame sequences. Both Long-Short Term Memory networks and Gated Recurrent Units are commonly used for temporal sequence modeling although their effectiveness for swimming stroke classification remains largely unexplored.

This research investigates the impact of temporal modeling architecture on swimming stroke classification using controlled Leave-One-Video-Out Cross-Validation framework. Three CNN-RNN architectures are evaluated: CNN-LSTM, CNN-GRU and CNN-BiGRU. A consistent preprocessing pipeline, feature backbone and training procedure across experiments ensures the effect of temporal sequence modeling can be isolated and compared. These results provide insight into the role of recurrent architectures for swimming stroke recognition and establish a foundation for future work using larger and more diverse swimming datasets.

II. LITERATURE REVIEW

A. Human Action Recognition

Human action recognition in video sequences is a widely studied problem in computer vision that requires modeling both spatial appearance and temporal dynamics. It has been studied using benchmark datasets like UCF101 that contain a range of real-world action and sports video sequences[1]. Early approaches relied on manually designed features like optical flow and histogram-based descriptors that were limited in their ability to capture complex patterns and variations in viewpoint.

Deep learning methods like convolutional neural networks have become the dominant approach for extracting spatial representations from individual video frames. Recently, CNN-based models have been extended by incorporating temporal modeling to capture motion across sequences. These hybrid architectures form the foundation of modern video classification systems and have applications in sports motion analysis like swimming stroke classification[2].

B. Deep Learning for Swimming and Sports Motion Analysis

Swimming stroke classification is a specialized form of human action recognition where small differences in limb motion and cadence define each class. Large-scale video

datasets like Sports-1M demonstrate the effectiveness of convolutional neural networks for learning spatial representations from raw video frames in sports classification tasks [3].

It has also been shown that frame-level CNN features alone are insufficient for capturing temporal dynamics in motion-based problems. To address this limitation, hybrid architectures combining spatial and temporal modeling have been widely adopted. Two-stream convolutional neural networks explicitly model appearance and motion information and have shown strong performance in video classification tasks [4].

Prior work in swimming-specific applications has demonstrated the effectiveness of combining CNN-based feature extraction with recurrent neural networks for stroke classification. For example, CNN-LSTM architectures have been used to estimate swimming stroke types from video sequences by modeling temporal motion patterns across frames [5]. More advanced approaches extend this idea to real-time analysis. SWIM-STAR introduces online stroke detection and stroke rate estimation for multiple swimmers. This capability highlights the importance of temporal modeling in dynamic aquatic environments [6].

These studies motivate the use of CNN-based spatial feature extraction combined with recurrent temporal modeling for swimming stroke classification.

C. Temporal Sequence Modeling with RNNs

Recurrent Neural Networks (RNNs) are commonly used for modeling sequential data for their ability to capture temporal dependencies. These networks operate on frame-level feature embeddings extracted from convolutional neural networks to model motion over time.

Long-Short Term Memory (LSTM) networks introduce gated memory cells to mitigate vanishing gradient issues and enable learning longer-term temporal dependencies in video sequences [7]. Gated Recurrent Units (GRUs) simplify this approach with fewer gates while maintaining comparable performance and improved computational efficiency [8].

Both architectures are widely used in video classification tasks where both temporal order and motion cadence are essential which make them suitable for modeling the cyclical pattern of swimming strokes.

D. CNN-RNN Hybrid Architectures for Video Classification

CNN-RNN hybrid architectures combine convolutional neural networks for spatial feature extraction with recurrent neural networks for temporal modeling. Video classification tasks leverage CNNs to process individual frames and generate feature embeddings which are passed to an RNN to model temporal dependencies across frames.

A pretrained convolutional backbone like ResNet18 is commonly used to extract features from each frame for its strong ability to learn image representations [2]. Feature sequences are then modeled using recurrent architectures like LSTMs or GRUs to capture motion dynamics over time. LSTM networks are used in sequence modeling due to their ability to learn long-term dependencies through gated memory mechanisms [7]. GRUs

provide a simplified alternative with fewer parameters while maintaining comparable performance in many sequence learning tasks [8].

The experiments presented here use a frozen ResNet18 backbone to extract frame-level features. These features are passed into different recurrent models to evaluate the effect of temporal modeling strategies on classification performance.

E. Summary and Gap

CNN-RNN architectures have demonstrated strong performance in general video recognition but relatively less research has been done focusing on the controlled comparison of temporal modeling strategies under strict cross-video evaluation.

The impact of different recurrent architectures such as LSTMs, GRUs and BiGRUs on swimming stroke classification has yet to be explored. This research works to address this gap by conducting a controlled study using identical spatial feature extraction and training configurations to isolate the effect of temporal sequence modeling on classification performance.

III. PROPOSED METHODOLOGY

A. Data Acquisition

Swimming instructional videos were collected from a publicly available source. Videos were selected based on the clarity of the stroke demonstration, swimmer visibility and variation in camera angles above and below the water. Each video contains continuous swimming sequences performed by a single athlete.

To ensure reproducibility, videos were manually organized and assigned a unique identifier used throughout the experiments.

B. Video Preprocessing

Raw swimming videos were standardized prior being segmented. Frames were extracted at a consistent frame rate and corrupt or invalid frames were removed. Non-swimming sequences like starts, turns or sequence transitions were also filtered out. All frames were resized to a fixed resolution to ensure consistent model input.

C. Stroke Segmentation and Annotation

Each video was segmented into smaller temporal units representing consistent views of a swimmers stroke cycles. Segmentation was performed to isolate stable swimming motion while preserving stroke boundaries and transitions. Each segment was assigned a corresponding stroke label based on the swimming technique being demonstrated.

D. Clip Extraction and Dataset Construction

Segmented stroke videos were converted into fixed-length clip representations stored as .npy arrays. Each sample included a sequence of frames defined as a temporal clip and a corresponding stroke label. The dataset is organized hierarchically by stroke, video and clip. Each clip is treated as a single training instance during modeling.

E. Cross-Validation Strategy

A Leave-One-Video-Out cross-validation strategy is used to evaluate generalization across unseen video. For each fold, one full video is held out as a test set and the remaining videos are split into training (80%) and test (20%) sets. This ensures no frames from the test video are used in training or evaluation and reflects a true cross-validation generalization.

F. Dataset Statistics

The final dataset consists of four swimming strokes: backstroke, breaststroke, butterfly and freestyle each with three videos per class (12 videos total). Each video was segmented into fixed-length clips using a sliding temporal window resulting in 411 fixed-length temporal clips across all classes. Each clip contains a sequence of 64 frames that were extracted at a constant rate and resized to a fixed spatial resolution.

In total the dataset contains 26,304 frames used for model training and evaluation. Despite its small size the dataset provides substantial variability in swimmer appearance, point of view and recording conditions making it a challenging real-world task for temporal sequence modeling.

G. Data Loading and Augmentation

Data is loaded using a custom PyTorch dataset class and batched using DataLoader. Deterministic loading is used during validation and testing to ensure reproducibility.

H. Model Input Representation

Each sample is represented as a 5D tensor:

$$(B, T, C, H, W)$$

Where:

B : Batch size

T : Temporal Sequence

C : Color channels

H, W : Spatial Dimensions

I. Feature Extraction

Each frame is processed using a pretrained ResNet18 network for spatial feature extraction. The CNN outputs a 512-dimensional embedding per frame. Temporal sequences of embeddings are passed into a recurrent neural network (LSTM or GRU) which produce sequence-level representations used for classification.

J. Model Architecture

The model consists of frozen ResNet18 backbone for spatial encoding followed by a recurrent neural network for temporal modeling. The final sequence representation is aggregated using max pooling and passed through a dropout layer ($p=0.5$) and a fully connected layer for swimming stroke classification.

K. Training Procedure

Models are trained using the Adam optimizer with a learning rate of 1×10^{-4} and cross-entropy loss. A batch size of 4 is used to optimize for memory constraints. A ReduceLROnPlateau scheduler drops the learning rate by a factor of 0.5 after three

epochs without validation improvement. Training is performed on up to 40 epochs and early stopping is triggered based on validation loss.

IV. EXPERIMENTS AND RESULTS

To evaluate the impact of temporal sequence modeling on swimming stroke classification, a controlled comparison of three experimental configurations is conducted. All experiments share the same preprocessing pipeline, data splits and CNN backbone. The varying components are temporal modeling architecture and hyperparameters (Experiment 3).

All models are evaluated using Leave-One-Video-Out Cross-Validation. One full video is held out per training fold as the unseen test set. The remaining videos are used for training and validation. Clip-level accuracy, video-level accuracy and Cross-fold mean and variance are captured and evaluated for each experiment.

The objective of these experiments was to determine whether alternative recurrent architectures improve temporal modeling for swimming stroke classification and generalization across unseen videos.

A. Experimental Setup

The following components remain fixed across all experiments

TABLE I. SHARED MODEL COMPONENTS

Component	Description
<i>Spatial Feature Extractor</i>	Pretrained ResNet18 (Frozen CNN Backbone)
<i>Input Representation</i>	Fixed-length frame sequences ($T=64$)
<i>Dataset Format</i>	.npy clip sequences
<i>Optimizer</i>	Adam ($lr = 1e-4$)
<i>Loss Function</i>	CrossEntropyLoss
<i>Learning Rate Scheduling</i>	ReduceLROnPlateau
<i>Training Duration</i>	~40 epochs
<i>Evaluation Protocol</i>	Leave-One-Video-Out Cross-Validation
<i>Aggregation Method</i>	Clip-level prediction + video-level probability averaging

This design ensures that differences in performance can be primarily attributed to temporal modeling.

B. Experiment 1 – CNN + LSTM (Baseline)

In Experiment 1, a hybrid CNN-LSTM architecture was used for swimming stroke classification. Sequences of 64 frame feature-vectors were processed by an LSTM with a hidden dimension of 256. Temporal max pooling was applied to the output of the LSTM and a dropout layer is applied before a fully connected classification layer produces stroke class predictions. Training components and hyperparameters were held constant for all experiments and defined in Section A. Experimental Setup.

Figure 1 shows the validation loss curves for the CNN-LSTM model. Validation loss decreases steadily throughout

training across all LOVO-CV folds which indicates stable convergence and effective learning of stroke-temporal patterns in the training data. Validation accuracy improved drastically during early epochs and remained consistently high as training continued.

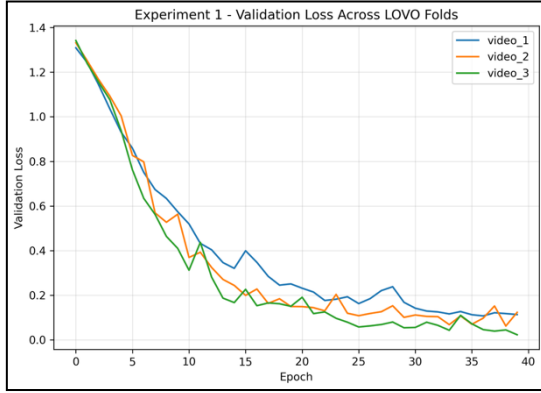


Fig. 1. Validation Loss – CNN + LSTM (LOVO-CVM)

Minor fluctuations in validation loss were observed after convergence and the learning rate reduction triggered late in the training cycle led to small performance. The CNN-LSTM model demonstrated stable training behavior with no substantial evidence of overfitting the training data.

Table II summarizes the classification performance of the CNN-LSTM model on the aggregated LOVO-CV test predictions. The model achieved an overall accuracy of 51% and a macro-averaged F1-score of 0.49.

TABLE II. CLASSIFICATION REPORT - CNN-LSTM (LOVO-CV)

Class	Precision	Recall	F1-score	Support
<i>Backstroke</i>	0.50	0.10	0.17	20
<i>Breaststroke</i>	0.73	0.55	0.63	20
<i>Butterfly</i>	0.72	0.65	0.68	20
<i>Freestyle</i>	0.35	0.75	0.48	20

Classification performance varied across stroke classes. Butterfly showed the strongest results with an F1-score of 0.68 followed by Breaststroke with an F1-score of 0.63. Freestyle had a high-recall of 0.75 but low precision indicating the model often incorrectly predicted freestyle for samples containing frames of other strokes. Backstroke proved to be the most challenging stroke to classify with a recall of only 0.10 and F1-score of 0.17.

These results suggest the CNN + LSTM model was able to learn discriminative temporal features for a few stroke classes although there was substantial confusion with others that limited the overall classification performance.

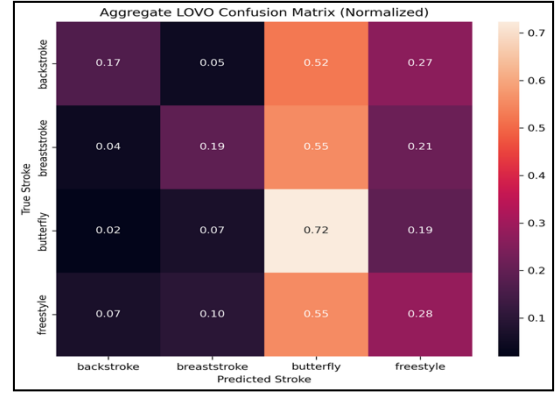


Fig. 2. Confusion Matrix – CNN + LSTM (LOVO-CVM)

Figure 2 presents the normalized confusion matrix for the CNN-LSTM model. The results indicate a strong bias towards the butterfly class which had the highest recall of 72%. More than half of all other strokes were incorrectly classified as butterfly suggesting the model struggled to learn distinct temporal patterns for those strokes.

Backstroke was the most challenging stroke to classify having only 17% of samples predicted correctly. Breaststroke and Freestyle also led to substantial confusion with Butterfly. These misclassifications help explain the moderately low overall accuracy and macro F1-score observed for the CNN-LSTM model.

The CNN-LSTM model provided a stable baseline for swimming stroke classification. With an overall accuracy of 51% and macro F1-score of 0.49, the model performed moderately well. Discriminating temporal features were learned for Butterfly and Breaststroke although Freestyle and Backstroke proved to be more difficult. The strong prediction bias toward Butterfly suggests limitations in the LSTM’s temporal representation and motivated the exploration of alternative recurrent neural network architectures in the following experiments.

C. Experiment 2 – CNN + GRU

Experiment 2 replaced the LSTM-based temporal module from the baseline module with a GRU. All other components are held constant including the ResNet18 CNN backbone, sequence length of 64 frames and training configuration defined in Section A. Experimental Setup. The GRU was chosen because it has fewer parameters than an LSTM, improved gradient flow over long sequences and better stability with the limited video data used in the experiment[8]. This experiment evaluated whether the GRU provides improve temporal modeling compared to the LSTM using the same feature extraction pipeline.

Figure 3 shows the validation loss curves for the CNN-GRU model across all LOVO-CV folds. Validation loss decreased steadily throughout training and stabilizes in later epochs which indicates a consistent convergence. Validation accuracy improved rapidly in early training epochs and remained relatively stable after convergence.

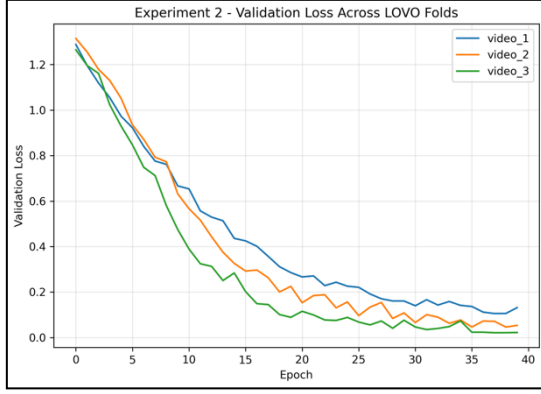


Fig. 3. Validation Loss – CNN + GRU (LOVO-CVM)

The model demonstrated stable training behavior with no signs of overfitting and the learning rate reduction in late epochs provided minor improvements to overall performance.

Table III summarizes the classification performance of the CNN-GRU model on the aggregated LOVO-CV test predictions. The model achieved an overall accuracy of 38% and a macro-averaged F1-score of 0.35.

TABLE III. CLASSIFICATION REPORT - CNN-GRU (LOVO-CV)

Class	Precision	Recall	F1-score	Support
<i>Backstroke</i>	0.00	0.00	0.00	20
<i>Breaststroke</i>	0.35	0.45	0.39	20
<i>Butterfly</i>	1.00	0.45	0.62	20
<i>Freestyle</i>	0.27	0.60	0.37	20

Performance varied significantly across stroke classes. Butterfly achieved the strongest precision while Breaststroke and Freestyle showed moderate but inconsistent performance. Backstroke was not correctly identified at all which resulted in zero precision and recall for this class. This result indicates a failure to learn discriminative temporal features for this stroke under this configuration.

These results suggest that replacing the LSTM with a GRU does not substantially improve overall classification performance under the current pooling strategy. The model remains biased toward Butterfly predictions revealing a persistent class imbalance or insufficient temporal aggregation.

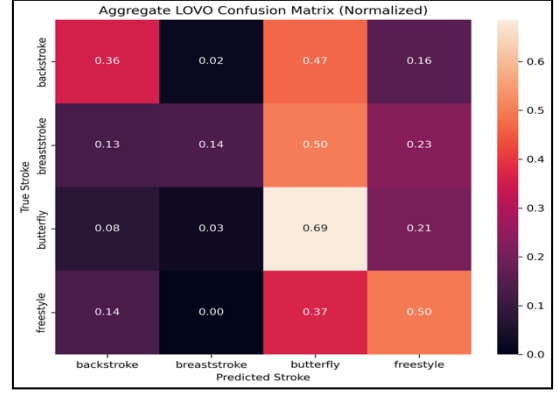


Fig. 4. Confusion Matrix – CNN + GRU (LOVO-CVM)

The normalized confusion matrix in Figure 4 shows a strong tendency for misclassification into Butterfly. Backstroke is rarely correctly predicted, while Breaststroke is frequently confused with Butterfly and Freestyle. Freestyle shows improved recognition compared to Backstroke but remains unstable across folds.

Overall, the CNN-GRU model performs similarly to the CNN-LSTM baseline in terms of macro-level performance but exhibits more severe class collapse in Backstroke. This suggests that the choice of recurrent unit alone is not the primary limiting factor in model performance. Instead, the temporal aggregation strategy and class separability in feature space likely play a larger role. These observations motivate further evaluation of bidirectional temporal modeling in Experiment 3.

D. Experiment 3 – CNN + BiGRU

In Experiment 3, the temporal modeling component was replaced with a bidirectional GRU (BiGRU). Unlike the previous experiments, the recurrent layer processes feature sequences in both forward and backward directions. This allows temporal information from the entire clip to contribute to the learned representation[4]. All other model components and training procedures remained unchanged.

Figure 5 shows the validation loss curves for the CNN-BiGRU model across all LOVO-CV folds. Validation loss decreased consistently throughout training and converged to low values across all folds. Validation accuracy improved rapidly during early epochs and remained stable for the remainder of training. Similar to the previous experiments, only minor fluctuations were observed after convergence and no substantial evidence of overfitting was detected.

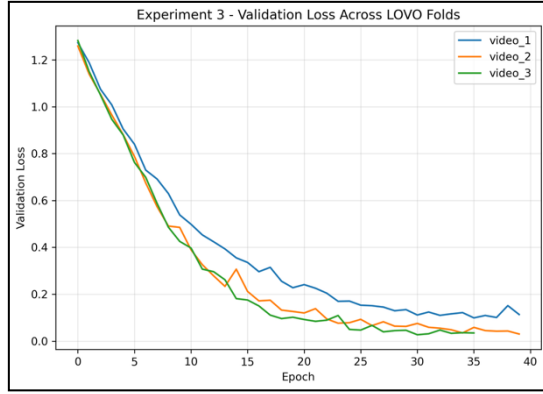


Fig. 5. Validation Loss – CNN + BiGRU (LOVO-CVM)

Table IV summarizes the classification performance of the CNN-BiGRU model on the aggregated LOVO-CV test predictions. The model achieved an overall accuracy of 50% and a macro-averaged F1-score of 0.50.

Classification performance varied across stroke classes. Butterfly achieved the highest F1-score (0.69), followed by Breaststroke (0.62). Compared with the CNN-GRU model, Backstroke performance improved substantially, achieving an F1-score of 0.38. Freestyle remained the most challenging stroke to classify with an F1-score of 0.30.

TABLE IV. CLASSIFICATION REPORT - CNN-BiGRU (LOVO-CV)

Class	Precision	Recall	F1-score	Support
Backstroke	0.83	0.25	0.38	20
Breaststroke	0.49	0.85	0.62	20
Butterfly	0.92	0.55	0.69	20
Freestyle	0.26	0.35	0.30	20

These results suggest that bidirectional temporal modeling improved the model's ability to learn discriminative features for several stroke classes. While overall accuracy remained similar to the previous experiments, classification performance was distributed more evenly across classes.

Figure 6 presents the normalized confusion matrix for the CNN-BiGRU model. Butterfly remained the strongest class with 64% correct classification, while Breaststroke achieved the highest recall among all classes at 85%. Backstroke performance improved relative to the previous experiments, although substantial confusion with Butterfly remained. Freestyle continued to be confused with both Breaststroke and Butterfly, limiting overall performance.

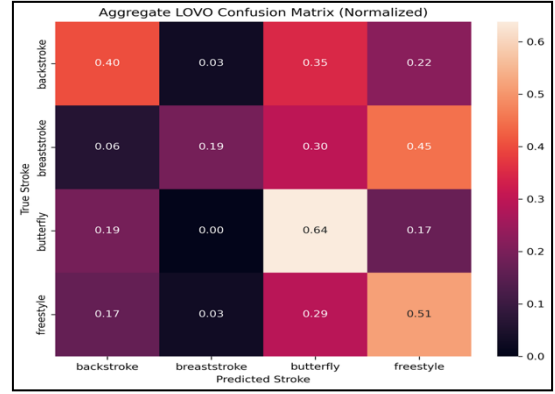


Fig. 6. Confusion Matrix – CNN + BiGRU (LOVO-CVM)

The CNN-BiGRU model achieved performance comparable to the CNN-LSTM and CNN-GRU models in terms of overall accuracy. However, the bidirectional architecture produced a more balanced classification profile across stroke classes. It also significantly improved Backstroke recognition compared to the GRU model. These findings suggest that access to both past and future temporal context can improve stroke representation learning, although additional improvements in feature extraction and temporal aggregation are likely necessary to achieve stronger overall classification performance.

E. Cross-Experiment Comparison

Table 5 summarizes the overall performance of all three temporal modeling architectures.

TABLE V. CROSS-EXPERIMENT COMPARISON

Model	Accuracy	Macro F1	Notes
CNN-LSTM	0.51	0.49	Strong Butterfly and Breaststroke performance; weak Backstroke recall
CNN-GRU	0.38	0.35	Severe Backstroke collapse; high class imbalance towards Butterfly and Freestyle
CNN-BiGRU	0.50	0.50	Most balanced performance; improved Backstroke recognition and more stable class distribution

The results show a clear performance gap between the CNN-GRU model and the two other architectures. The CNN-LSTM and CNN-BiGRU models achieved similar overall accuracy around 0.50 and the CNN-GRU underperformed with an overall accuracy of 0.38. This performance drop is primarily driven by the model's inability to learn a separable representation for the Backstroke class.

Training behavior was stable and showed consistent decreases in validation loss with no clear evidence of overfitting. The CNN-GRU exhibited slightly noisier convergence compared to the other two architectures although all models did eventually reach a stable minima.

Similar patterns were observed across all experiments. Butterfly was the most accurately predicted class although Breaststroke and Freestyle were often confused with it.

Backstroke was the most challenging class for all three models even though the BiGRU improved its recognition over the other two architectures. This suggests that bidirectional temporal context improves class-level separability under the current dataset constraints.

F. Discussion

This study evaluated the impact of temporal sequence modeling on swimming stroke classification using a controlled Leave-One-Video-Out Cross-Validation framework. The results indicate that temporal sequencing can learn meaningful swimming stroke patterns despite the challenging nature of the dataset. The CNN-LSTM and CNN-BiGRU models achieved similar results while the CNN-GRU demonstrated slightly lower classification performance and stronger prediction bias towards Butterfly.

Classification accuracy varied for all stroke techniques. Butterfly consistently achieved the strongest performance across all experiments and Backstroke and Freestyle remained most challenging to distinguish. These results suggest that some swimming strokes exhibit more distinct temporal characteristics than others in this dataset.

There were several limiting factors affecting overall performance. The dataset consisted of a small number of videos collected from publicly available YouTube footage. This resulted in significantly different viewpoints, swimmer techniques, video quality and recording conditions. The ResNet18 backbone also remained frozen during training which limited adaptation to swimming specific visual features.

Despite these limitations, the experiments demonstrate that recurrent neural networks can learn useful temporal representations from swimming video sequences. The controlled experimental design provides meaningful comparison of temporal modeling architectures and establishes a foundation for future work using larger, more diverse swimming datasets.

V. CONCLUSION

This research evaluated the effectiveness of CNN-RNN hybrid architectures for swimming stroke classification using Leave-One-Video-Out Cross-Validation. CNN-LSTM, CNN-GRU and CNN-BiGRU temporal modeling approaches were compared. Each experiment used the same preprocessing

pipeline, feature extraction backbone and training procedure so that the impact of temporal sequence modeling could be isolated and evaluated.

The results demonstrated that meaningful temporal features can be learned from swimming video sequences despite the challenges and limitations with limited and highly variable training data. The CNN-BiGRU model performed best overall and the CNN-LSTM provided a comparable classification baseline. The CNN-GRU showed lower classification performance and more class confusion.

Classification accuracy was overall moderate but this study establishes a controlled experimental framework for evaluating temporal architectures in swimming stroke recognition. A key contribution of this work is the strict Leave-One-Video-Out evaluation which ensured no temporal leakage across training and testing. Future work could focus on larger and more diverse datasets and additional temporal modeling approaches. Fine-tuning of the networks used to extract features may also help improve generalization across unseen swimmers and video recording conditions.

REFERENCES

- [1] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes from Videos in The Wild," arXiv:1212.0402, 2012.
- [2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.
- [3] A. Karpathy et al., "Large-scale Video Classification with Convolutional Neural Networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2014.
- [4] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in Advances in Neural Information Processing Systems (NeurIPS), 2014.
- [5] K. Hakozi, N. Kato, M. Tanabiki, J. Furuyama, Y. Sato, and Y. Aoki, "Swimmer's Stroke Estimation Using CNN and Multi-LSTM," Journal of Signal Processing, vol. 22, no. 4, pp. 219–222, 2018.
- [6] G. M. Mashrur, E. Elahi, Y.-H. Yang, Y.-B. Kim, and W. K. Mummery, "SWIM-STAR: Online detection of swimming strokes and stroke rates for multiple swimmers in videos," in Proc. IEEE Int. Conf. Image Processing (ICIP), 2017.
- [7] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
- [8] K. Cho et al., "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," arXiv preprint arXiv:1406.1078, 2014.