Wave Statistic Estimation from Surfline Video Data Using ConvNets

Galen Egan
Stanford University, Department of Civil and Environmental Engineering
473 Via Ortega, Stanford, CA
gegan@stanford.edu

Abstract

Coastal communities worldwide have a vested economic interest in monitoring wave properties in real time. Here, two neural network architectures are applied to learn wave height and wave period from cropped and downsampled video data sourced from the surf condition website, Surfline.com. While a traditional ConvNet architecture performed poorly in terms of predicting the wave parameters, a ConvLSTM-based network demonstrated significant improvements, and predicted the wave characteristics with accuracy comparable to recent work in this field. Results indicate that recurrent neural networks show particular promise in predicting wave conditions from video data, and future work should focus on fine-tuning these methods and applying them to different coastal regions.

1. Introduction

Despite only comprising 10% of our nation's land area, approximately 40% of the population lives in a coastal county. These regions are both vital economic hubs, and popular destinations for recreation. Proximity to the ocean, however, comes with risks, including damage to property and infrastructure from waves and storm surge. Therefore, it is in the economic best interest of coastal communities to monitor wave conditions in real time, allowing them to plan for recreation and potentially dangerous storm events. However, this level of monitoring generally requires *in situ* pressure transducers, which are expensive and difficult to properly deploy. There is, however, a widely available dataset that provides qualitative wave conditions: Surfline.com maintains a network of cameras across the country with live streams of beaches and waves for the purpose of informing surf conditions.

The goal of this project was to leverage Surfline's database to predict wave height and wave period, two common parameters that are widely used to inform shoreline recreation and coastal infrastructure design. Specifically, the input to our model was Surfline video footage from the Scripps Pier in La Jolla, CA [7]. These data were fed to two algorithms, a traditional Convolutional Neural Network (CNN) and a hybrid Long Short-Term Memory Convolutional Nueral Network (ConvLSTM). Each of these models output a predicted wave period and wave height, which were compared to wave period and height measurements collected adjacent to the video camera using traditional pressure sensors. If the wave parameters can be predicted with reasonable accuracy using only the video data, then the algorithm presented here could potentially be trained on other Surfline video feeds, providing cheap and reliable estimates of wave conditions across the world.

2. Related Work

For decades, and indeed up to the present day, the most common way to measure wave statistics was using pressure sensors deployed in the water [6]. By measuring the time-varying pressure changes that accompany wave-induced deflections of the water surface, the dominant wave period and wave height can be determined via spectral analysis [2]. While these methods are reliable, the necessary instrumentation is expensive and requires significant expertise to properly deploy. It is also common for pressure sensors to be lost or buried during storm events, or for the electronics housing to flood due to improper maintenance. And while high-frequency radar measurements have been used as a land-based alternative [3], they can be prohibitively expensive and difficult to operate.

The task of estimating wave statistics along our coasts could clearly benefit from a cheaper and easier measurement technique. Recently, machine learning algorithms have been applied to predict wave height from physics-based model outputs [4], and *in situ* accelerometer data [5]. Results have been promising, but still require significant effort in either model setup or data collection. However, a new study by [1] demonstrated the utility of CNNs in predicting wave height and period from static coastal image data. In that work, a number of pre-trained CNNs were tested on shorline image data, and it was found that MobileNetV1 and Inception-ResNetV2 generally offered the best predic-

tive performance, with RMS errors of 0.14 m and 0.41 s for wave height and period, respectively.

Here, that work will be expanded upon by training a custom CNN architecture on Surfline video footage. In this way, we will be able to determine whether video data offers significantly increased predictive capability over static image data. This will also enable us to compare different architectures, including recurrent neural networks (RNNs), which are tailored for processing time-varying data.

3. Methods

Two methods were attempted to solve this regression problem. In the first, video data were fed into a standard CNN architecture implemented in Pytorch, with a pipeline as shown in 1 below.

$$([\text{conv} \rightarrow \text{ReLU} \rightarrow \text{Dropout} \rightarrow \text{Batchnorm}] \times 2 \rightarrow \text{MaxPool}) \times 2 \rightarrow \text{Linear} \quad (1)$$

The convolutions operated on each frame of the video, treating each frame as a separate channel (see Section 4 for preprocessing details). Model parameters were chosen by minimizing the validation error on 40 videos, chosen randomly from a 400 video subset of the full dataset. Conv layer sizes were iterated over sizes from the set $[4, 8, 16, 32, 64]$, dropout was iterated over probabilities $[0.1, 0.3, 0.5, 0.7]$, and the learning rate was iterated over the range [1e-6, 1e-2]. Batch sizes were tested at $[4, 8, 20, 40, 60, 90, 120]$, and number of iterations for each batch was tested at $[100, 200, 400, 800]$. Optimization was performed using the Adam algorithm, minimizing the mean squared error.

After this procedure, it was found that the lowest validation error was achieved with conv layer sizes of 32, 32, 16, and 8 filters, respectively. Each layer had a kernel size of 3×3 and zero-padding of 1. Dropout was set to $p = 0.5$ for regularization, and each MaxPool layer kernel was size 2×2. The optimal learning rate was $lr = 5 \times 10^{-4}$, with an optimal batch size $b = 90$, and 400 epochs for each batch.

Figure 1 shows the model prediction on the validation set with the parameters chosen from this procedure. As indicated by the coefficient of determination, r^2, the model predicts wave height rather poorly, but predicts wave period with reasonable accuracy. A deeper network architecture was tested as well (4 conv-ReLU-Dropout-Batchnorm layers rather than 2), but the accuracy did not improve so the simpler model was retained.

Because of the questionable performance of model 1, a model more suited to video data was also tested. For this, we chose the ConvLSTM introduced by [9], and implemented in Keras. The ConvLSTM architecture leverages the spatial awareness of CNNs and embeds them in a time-resolving RNN framework. While a traditional LSTM would receive a 1D vector as an input at each time step, the ConvLSTM accepts the 3D (or in our case, 2D) image, performing convolution operations for each frame. The model architecture that we chose is shown in 2 below.

$$\text{ConvLSTM2D}_{32} \rightarrow \text{ConvLSTM2D}_{16} \rightarrow \text{Linear} \quad (2)$$

Here, the subscript on ConvLSTM2D refers to the number of filters. Similar to the parameter sweep for model 1, we optimized the number of filters at each layer, along with the kernel size. The minimum mean squared error optimized with the Adam algorithm was achieved with 32 and 16 filters, and kernel sizes of 5×5 and 3×3 in the first and second ConvLSTM layers, respectively. The learning rate was set at $lr = 1 \times 10^{-4}$. Due to time constraints, this optimization was performed on a much smaller subset of the data, so these parameters may not generalize well.

4. Dataset and Features

The raw data consisted of 10-minute .mp4 video files downloaded from Surfline.com. A total of 20 days of video were downloaded, spanning April 17, 2020 to May 7, 2020. Videos were only processed if they were filmed between 0600 and 1930 to ensure adequate sunlight. This resulted in a total of 1600 total videos, 80% of which were ultimately used for training. The remaining 20% were split into two equally-sized sets, one for validation and the other for hold-out testing.

Each video was filmed at 25 frames per second (fps), and each 3-channel color frame measured 720×1280 pixels. It would have been prohibitively expensive to process the full videos; therefore, each video was downsampled to a frame rate of 1 fps, which still allowed for resolution of wave frequencies up to the Nyquist frequency of 0.5 Hz. Given the average wave conditions at the study site [8], this level of downsampling should not negatively affect the wave period estimation. The videos were also shortened to 1 minute, resulting in a temporal dimension of 60 frames.

Next, the downsampled video data were converted to grayscale, and cropped at the top and bottom to exclude the sky and beach, respectively. They were then further cropped to reduce the image width by a factor of two, resulting in a frame dimension of 340×640 pixels. Finally, each frame was resized to a dimension of 128×128 pixels, resulting in a final video input of $60 \times 128 \times 128$. During the course of training the model, other input dimensions were experimented with, including 256×256 pixels, and 256×128 pixels, but these higher-resolution images did not improve model performance. Example frames from the processed input videos are shown in Figure 2.

The ground truth data (wave period and height) are collected by pressure sensors mounted on the Scripps pier by

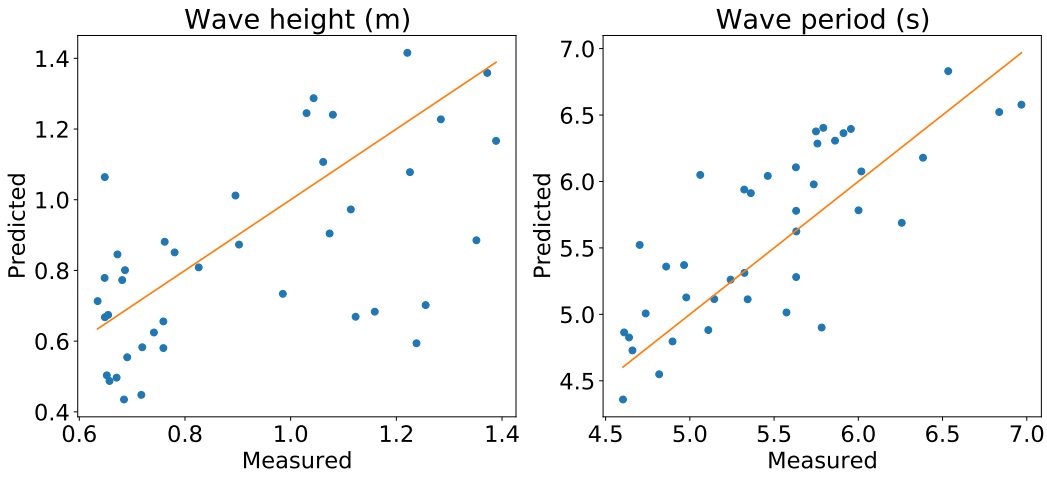

Figure 1. Predicted vs measured wave height ($r^2 = 0.05$) and period ($r^2 = 0.49$) on the validation set.

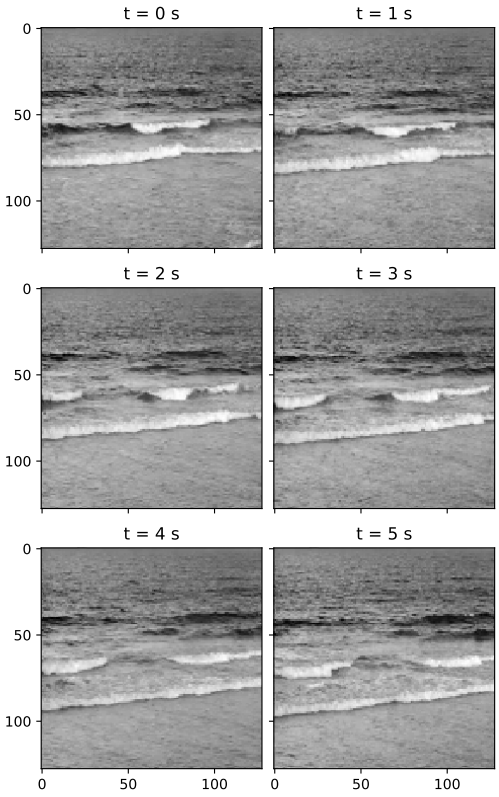

Figure 2. Six consecutive frames from a processed Surfline video.

the Scripps Institute of Oceanography (SIO). The data are transmitted to a SIO server, where they can be downloaded directly [8]. The data are reported at hourly intervals, so they were first upsampled to a frequency of one measurement every 10 minutes, corresponding to the frequency of the video data. These 10-minute data were then interpolated onto the same time vector as the video data. Because wave conditions vary primarily over tidal timescales (i.e., on the order of 6 hours) and longer (e.g., with daily wind patterns) [2], this level of interpolation should introduce negligible error in the regression.

During model training, standard normalization techniques were applied to the ground truth data, including detrending and normalizing by the standard deviation. However, these procedures appeared to decrease the model accuracy, so ultimately the model was trained on unnormalized input images and ground truth data.

5. Results and Discussion

Applying model 1 to the full training set resulted in poor performance for both wave height and wave period on the hold-out test set, as shown in Figure 3. Neither metric was predicted with a positive coefficient of determination, and the RMS error for wave height and period were 0.25 m, and 1.18 s respectively, which are both significantly higher than the errors reported in [1]. The wave height prediction fails to capture the ground truth trends, while the wave period prediction displays significant bias, nearly always overpredicting the true wave period. One potential reason for this failure is that the parameters chosen based on the optimization procedure described in Section 3 did not generalize well to the full training set. Indeed, results improved slightly when increasing the training batch size from $b = 90$ to $b = 160$, but there was not enough time to attempt a full parameter sweep.

In order to further probe these disappointing results, Figure 4 shows the training and validation errors output after 400 iterations on each batch of 160 videos. Though the training error decreases substantially, the validation error

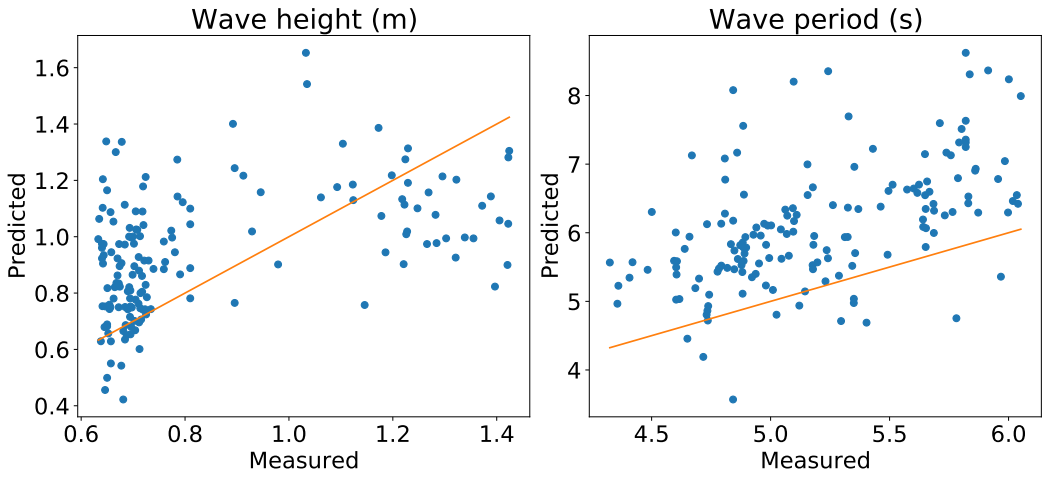

Figure 3. Predicted vs measured wave height ($r^2 = -0.10$) and period ($r^2 = -5.88$) on the test set for the standard ConvNet architecture (model 1), with the one-to-one line shown in orange.

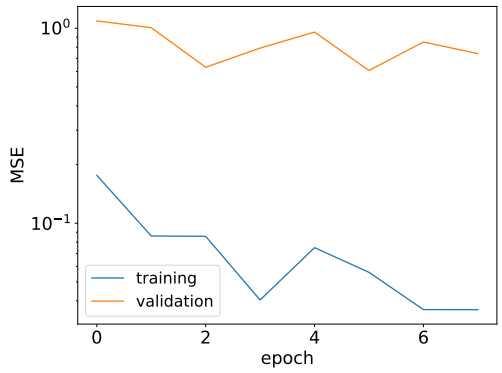

Figure 4. Training and validation errors for the standard ConvNet architecture (model 1).

remains high, indicating that the model may be overfitting the training data. This is surprising given the aggressive dropout parameter $p = 0.5$ applied after each ReLU layer, but clearly more work remains to be done in tuning model architecture 1 to this dataset.

Rather than fine-tune model 1 *ad nauseam*, we next applied the ConvLSTM architecture (model 2) to the full training set. The results on the test set were much more promising, as shown in Figure 5. The RMS errors in this case were 0.17 m and 0.4 s for the wave height and period, respectively, which are quite similar to the results described in [1]. Both the bias and variance around the ground truth values are improved compared to Figure 3, though the model often overpredicted the wave height at smaller measured wave heights, as indicated by the cluster of predictions between measured values of 0.6–0.8 m, and underpredicted at larger measured wave heights, especially beyond 1.2 m. On the other hand, the trend in the wave period is captured quite faithfully, though there is more variance around the one-to-one line.

The training behavior for the ConvLSTM architecture, shown in Figure 6, reflects the improved performance. Unlike the ConvNet architecture, where we trained each batch for a fixed number of iterations, the ConvLSTM model was trained with an early stopping condition on each batch, such that the training stopped if the validation error did not improve after 5 iterations. As such, the "epochs" in Figure 6 could more precisely be termed "iterations on each batch".

The validation error, though noisier, is not significantly larger than the training error, and both generally decrease with training iterations. Given these results, it may be that the model is underfitting, and could benefit from increased model complexity. While there was not sufficient time to test a larger (i.e., more filters) or deeper model (i.e., more layers), the results presented in Figure 5 are encouraging, and indicate that the ConvLSTM architecture is well-suited for extracting information from video data.

6. Conclusions and Future Work

Among the two architectures that were tested on the Surfline video dataset, the ConvLSTM network clearly outperformed the standard ConvNet architecture in predicting both wave parameters. This is not necessarily surprising, as RNN-based methods are designed explicitly for time series input. The ConvLSTM training behavior, which did not indicate overfitting, suggests that a more complex model could improve predictions even further. Given more time to fine-tune the model architecture, and a larger training set, a ConvLSTM-based model could prove quite robust in predicting wave parameters from video data.

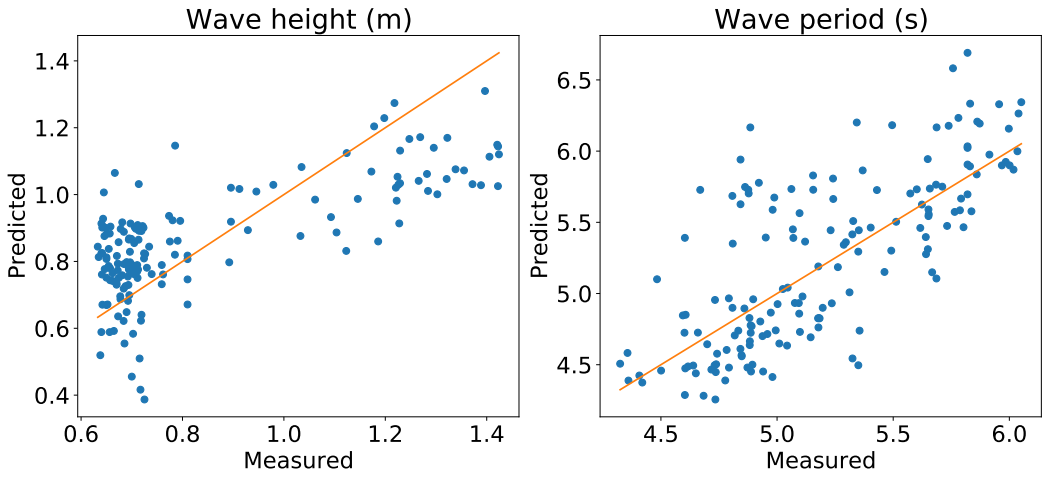

Figure 5. Predicted vs measured wave height ($r^2 = 0.51$) and period ($r^2 = 0.21$) on the test set for the ConvLSTM model (model 2), with the one-to-one line shown in orange.

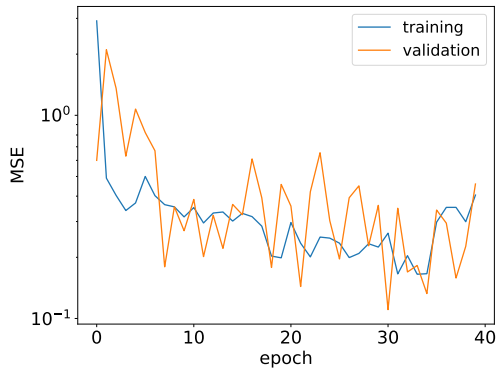

Figure 6. Training and validation errors for the ConvLSTM architecture (model 2).

Aside from model architecture, future work could focus optimizing input data characteristics. In particular, the input pixel resolution (especially in the height) could exert a strong control on the wave height prediction. The width of the input image may also affect the prediction, as a narrower frame may not adequately resolve the lateral variability in wave characteristics. For example, alongshore variations in the seabed could bias wave heights at a particular location relative to the ground truth measurements; this could be addressed by feeding the model a wider frame. Furthermore, our training set only consisted of 1280 videos. Given the widespread availability of training data, the model could easily benefit from a larger training set. Overall, the results presented here represent an encouraging first step in predicting wave parameters based on publicly available video data, and will hopefully inspire future work on this important and challenging problem.

7. Contributions

This project, including all data processing, model setup, analysis, and writing was completely independently by the sole author.

References

[1] D. Buscombe, R. J. Carini, S. R. Harrison, C. C. Chickadel, and J. A. Warrick. Optical wave gauging using deep neural networks. *Coastal Engineering*, 155:103593, 2020.

[2] R. G. Dean and R. A. Dalrymple. *Water wave mechanics for engineers and scientists*, volume 2. World Scientific Publishing Company, 1991.

[3] H. C. Graber and M. L. Heron. Wave height measurements from hf radar. *Oceanography*, 10(2):90–92, 1997.

[4] S. C. James, Y. Zhang, and F. O'Donncha. A machine learning framework to forecast wave conditions. *Coastal Engineering*, 137:1–10, 2018.

[5] T. Liu, Y. Zhang, L. Qi, J. Dong, M. Lv, and Q. Wen. Wavenet: learning to predict wave height and period from accelerometer data using convolutional neural network. In *IOP Conference Series: Earth and Environmental Science*, volume 369, page 012001. IOP Publishing, 2019.

[6] M. S. Longuet-Higgins. On the statistical distribution of the height of sea waves. *Journal of Marine Research*, 11:245–266, 1952.

[7] D. Sellin. *Scripps Pier, Northside Surf Report and Forecast*, 2020 (accessed April 22 - May 20, 2020).

[8] SIO. *Scripps Pier Station Data Home Page*, 2020 (accessed May 15, 2020).

[9] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.