Automated Surf Reports from Image Data

Stuart Cornuelle
Department of Computer Science | Stanford University
stuartcc@stanford.edu

Abstract

We seek to generate automated surf reports from image data. We explore and compare single-task and multitask architectures in an attempt to produce accurate characterizations of (1) wave size and (2) wave conditions (described below) using convolutional neural networks. Our final model achieves 62.3% test set accuracy across these two tasks in a simplified multiclass classification setting.

1 Introduction

Surfers want to be where the best waves are. Wave conditions, however, vary constantly along a multitude of dimensions (from the wavelength and period of the swell to tidal conditions to wind speed and direction to the number of other surfers in the water). Thus over the past 50 years an industry of increasingly refined surf forecasting, reporting, and live-streaming of current conditions (via cameras permanently positioned onshore) has evolved to help users make informed decisions about where and when to surf.

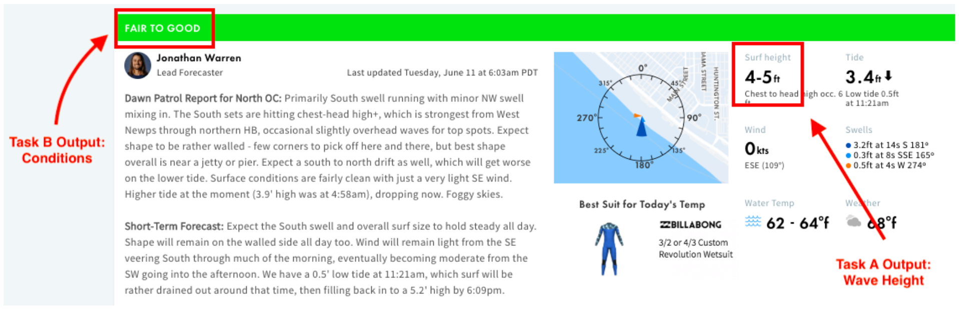
Much of the final qualitative, human-language surf reporting consumed by the end user, however, is still generated manually by human experts. These specialists survey the breadth of data on hand each day and translate it into a concise, once- or twice-daily briefing for publication on the web or via mobile applications. (See Figure 1.) In this paper we explore whether it is possible to extract a useful report of surfing conditions from image data alone, without the need for manual human interpretation.

The input to our algorithm is still frame images from the live-streaming cameras of a popular surf reporting website. We use a convolutional neural network with multitask learning to output:

1. Wave height as a class label in $[0, 5]$ (bucketing heights in the range of 0-25 feet)
2. Surf conditions[1] as a label in $[0, 7]$ (a scale developed by the leading surf forecasting service)

This project shares a dataset and preprocessing code with a CS230 project this quarter, in which the task attempted is to predict traffic volume to the pages of the leading surf forecasting service using image, numerical and categorical data as inputs to a multimodal learning architecture. While that project explores how to combine disparate (image vs. non-image) inputs toward a single output, this project instead seeks to learn from one input a useful representation from which two distinct task outputs can be generated. (The linear and logistic regression baselines are also unique to CS 229.)

[1]Measure that encompasses effects of wind, surface conditions, wave shape and overall surf quality.



FAIR TO GOOD

Jonathan Warren
Lead Forecaster

Last updated Tuesday, June 11 at 6:03am PDT

Dawn Patrol Report for North OC: Primarily South swell running with minor NW swell mixing in. The South sets are hitting chest-head high+, which is strongest from West Newps through northern HB, occasional slightly overhead waves for top spots. Expect shape to be rather walled - few corners to pick off here and there, but best shape overall is near a jetty or pier. Expect a south to north drift as well, which will get worse on the lower tide. Surface conditions are fairly clean with just a very light SE wind. Higher tide at the moment (3.9' high was at 4:58am), dropping now. Foggy skies.

Short-Term Forecast: Expect the South swell and overall surf size to hold steady all day. Shape will remain on the walled side all day too. Wind will remain light from the SE veering South through much of the morning, eventually becoming moderate from the SW going into the afternoon. We have a 0.5' low tide at 11:21am, which surf will be rather drained out around that time, then filling back in to a 5.2' high by 6:09pm.

Best Suit for Today's Temp
BILLABONG
3/2 or 4/3 Custom Revolution Wetsuit

Surf height
4-5ft
Chest to head high occ. 6 ft

Tide
3.4ft
Low tide 0.5ft at 11:21am

Wind
0kts
ESE (109°)

Swells
● 3.2ft at 14s S 181°
● 0.3ft at 8s SSE 165°
● 0.5ft at 4s W 274°

Water Temp
62 - 64°f

Weather
68°f

Figure 1: Sample surf report interface with task outputs indicated.

2 Related work

Many multitask learning approaches rely on hard parameter sharing—the joint learning of shared weights from multiple task outputs. This method, introduced in [2], is beneficial for solving related tasks, wherein the shared parameters mutually benefit the tasks to which they contribute.

Knowing what constitutes a related task, however, and therefore which tasks stand to gain from such an architecture, is a matter of art and domain insight. To improve the quality and rigor of these determinations, some researchers have sought to instead learn from data when and how to implement sharing across tasks; [4] is an example.

Finally, by contrast with such approaches, soft parameter sharing in multitask learning involves training separate weights for each task and then penalizing the distance between them in a form of regularization; [3] and [9] are representative instances. [8] Our work in this paper uses hard parameter sharing, since we can assume with high probability that the tasks are indeed related.

3 Dataset and Features

Our data[5], spanning one year (April 2018 – April 2019) across three popular surf spots, includes:

1. Twice-daily surf reports logging wave height and conditions
2. Bi-hourly still frame surf cam images (12 per day; see Figure 2 for a sample image)

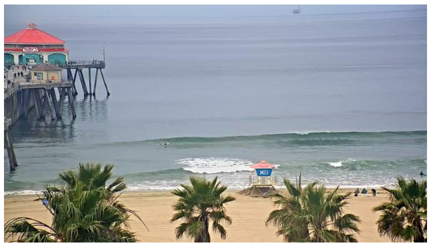


Figure 2: Sample surf cam still frame image.

Our image preprocessing and data augmentation efforts include the following[7][6]:

1. Filtering nighttime and otherwise obscure still frames from the dataset by thresholding images at a manually determined average pixel value.
2. Resizing to 64×64 pixels, a dimension balancing computational efficiency with maintenance of the visual features of wave conditions (assessed via domain knowledge).
3. Augmenting the dataset with horizontally flipped copies of each still frame—since aspects of wave size and quality are invariant to horizontal inversion—and adding random perturbations to image brightness and saturation.

The resulting image dataset, shuffled and split, consists of 10,838 training images (80%); 1,355 dev images (10%); and 1,355 test images (10%).

Labels for Task A (wave height) are constructed as a weighted average of the minimum, maximum, and "occasional" (extreme upper bound) reported surf heights in the dataset for a given day. Raw mean daily wave heights range from 0.0 to 23.25 feet; we then bin these values into six roughly balanced classes (e.g. $y = 4$ corresponds to 6- to 8-ft. waves).

Labels for Task B (conditions) are the human expert's holistic assessment of surf quality as governed by wave shape, quality, consistency and surface texture. This data takes the form of a rating from an eight-element set:

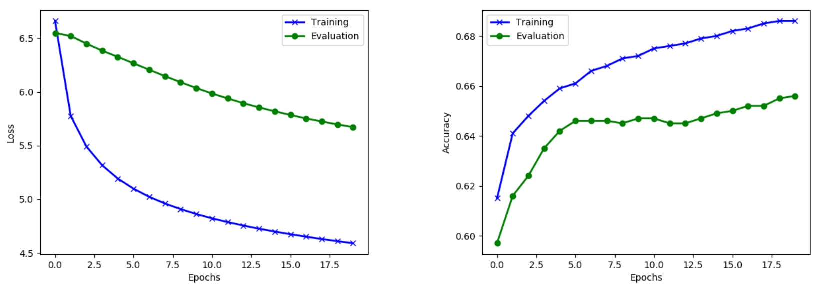
1. Flat
2. Very poor
3. Poor
4. Poor to fair
5. Fair
6. Fair to good
7. Good
8. Very good

4 Methods

We first implement naive baseline algorithms for Tasks A and B in a simplified setting. We then move on to building and tuning more sophisticated single-task models for wave height and conditions separately; these are the benchmarks against which we ultimately hope to improve with our multitask architecture. Finally, we implement and tune the multitask model itself.

Our Task A baseline performs linear regression on the flattened pixel values of each input image to predict raw mean wave heights directly. Unlike a convolutional network, this model cannot avail itself of the spatial relationships among image features, and so its potential is limited. Figure 3(a) displays training and dev loss of the resulting model. (This being a regression task, accuracy is unavailable, and so we cannot compare the baseline directly to later implementations.)

The naive model does learn—we might expect that concentrations of whitewater in the center of the input image reflect elevated wave activity, for example—but the final dev loss represents more than a full standard deviation of error in predicted wave height.



(a) Task A baseline performance (loss). (b) Task B baseline performance (accuracy).

Figure 3: Naive baseline results for Tasks A and B.

Our Task B baseline collapses the conditions label to $\mathbb{R}^2$ ("bad" conditions vs. "good") and performs logistic regression, again on the flattened image pixel values, to assess whether condition quality is fundamentally learnable from image data. Results are in Figure 3(b): $> 65\%$ dev set accuracy after 20 epochs of training, which we expect from the variance observed might be improved by regularization efforts.

Next we build and tune separately two convolutional neural networks to perform multiclass classification on Tasks A and B respectively. These models convolve filters of parameters over the spatial extent of their layer inputs to extract increasing abstract, semantically recognizable features from the raw pixel values of the input as they propagate into deeper layers. Thus while an early layer might identify edges and colors, for example, the activation of a later layer's neurons could reflect the observed presence of a surfer bobbing in the ocean in the input frame, or of the contrast of whitewater against the dark green of steep wave faces that is the signature of high surf.

The activations then pass through dense layers—linear transformations combined with ReLU nonlinearities, which give the model its expressive power—to produce logits in $\mathbb{R}^6$ (for Task A) or $\mathbb{R}^8$ (for Task B). The loss assessed on these outputs is the softmax cross entropy loss, given by

$$L = -\frac{1}{n}\sum_{i=n}^{n} \log\left(\frac{\exp(o_{y_i})}{\sum_{j=1}^{K}\exp(o_j)}\right) \tag{1}$$

Here o is the logit vector, y_i is the ground truth label of the i^{th} training example, and n is the minibatch size.

Finally, we construct a multitask architecture that seeks to generate predictions on Tasks A and B jointly from a single model. We can defend this architectural choice by observing that the tasks are closely related from a computer vision standpoint: the image features that determine wave height are very likely to contribute or be closely linked to those that predict wave quality, and include elements such as contrast lines in the ocean surface or whitewater regions.

As a result, we expect that the model should be able to learn a representation of the input that is an improvement over those learned by models for each task in isolation, via a number of mechanisms[8][1]:

- Joint learning introduces an inductive bias toward a representation favored by a different but related task, which allows the learning to better generalize (not overfit to the noise of a single task).
- The shared layers in the multitask model allow for one task to benefit from the learning that accrues to the other task (what [1] calls "eavesdropping"), improving its capacity to make good predictions.

Our multitask architecture is in Figure 4. We use hard parameter sharing in the convolutional layers, followed by independent branching networks of fully connected layers for Tasks A and B. The multitask loss is a non-weighted average softmax cross entropy loss [see (1)] on the two tasks.

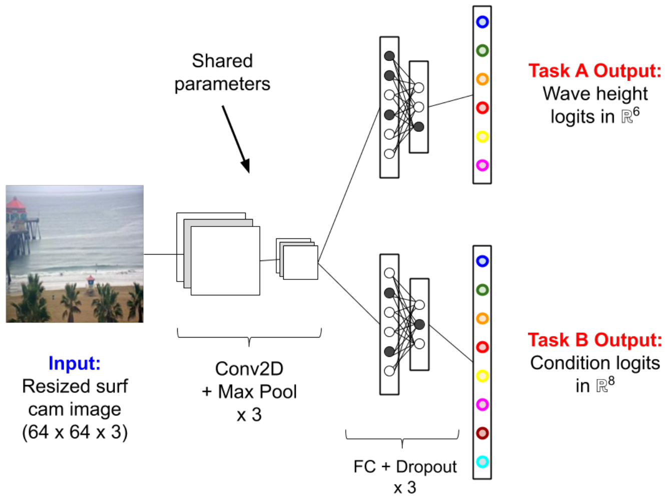

Shared parameters
Task A Output:
Wave height logits in $\mathbb{R}^6$
Input:
Resized surf cam image (64 x 64 x 3)
Conv2D + Max Pool x 3
Task B Output:
Condition logits in $\mathbb{R}^8$
FC + Dropout x 3

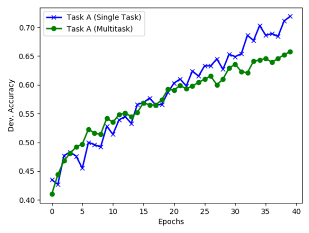
Figure 4: Final model architecture.

5 Experiments/Results/Discussion

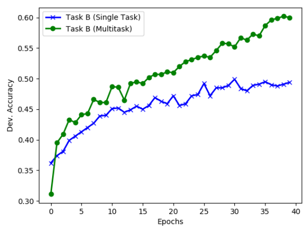
Our metric of interest across all implementations is test set accuracy. For each of our three main models (Task A, Task B and multitask) we separately tune the following hyperparameters in order, leaning on an adaptation of the CS230 example code for the MNIST Sign Language classification task:

- Number and size of convolutional layers and fully connected layers; we tune greedily with respect to accuracy in order to ensure sufficient bias reduction in the resulting model.
- Learning rate and number of training epochs, to optimize convergence.
- Regularization (dropout in the dense layers) to reduce variance and overfitting.

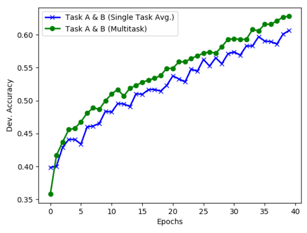
Dev set learning curves for Tasks A and B are presented in Figure 5: We compare the accuracy on Task A as achieved by the single-task model as well as by its output in the multitask model; as well for Task B. Then we compare the combined accuracies by averaging performance of the single-task models and graphing the average against the multitask performance.



(a) Task A



(b) Task B



(c) Tasks A & B (avg. performance)

Figure 5: Experimental results: Dev set accuracy comparing single-task vs. multitask models.

It appears that while the single-task implementation outperforms on Task A, the multitask model achieves higher accuracy on Task B as well as the combination of both tasks. Test set evaluation confirms these results, as shown in Table 1:

Model	Test Set Acc. (%)
A (single-task)	71.7
A (multitask)	66.1
B (single-task)	49.8
B (multitask)	58.7
A & B (single-task avg.)	60.1
A & B (multitask)	62.3

Table 1: Test set performance comparison.

6 Conclusion/Future Work

While it appears that Task A is better performed in isolation (at least with this particular model and dataset), Task B does benefit from the multitask architecture and its promise of an improved joint representation of the input space. Indeed, if we weight equally the importance of reporting wave size and wave quality, the multitask model yields a strictly better automated surf report. In further work, with more time and data, we would split the "conditions" label from Task B into two further variables: wave shape (as good waves for surfing have a universal and recognizable visual fingerprint) and surface texture, mainly due to the effect of wind. With a large dataset expertly (and manually) labeled with these features, we expect that a multitask model might produce a truly useful wave quality recognition engine.

References

[1] Rich Caruana. "Multitask learning". In: *Machine learning* 28.1 (1997), pp. 41–75.

[2] Richard Caruana. "Multitask Learning: A Knowledge-Based Source of Inductive Bias". In: *Proceedings of the Tenth International Conference on Machine Learning*. Morgan Kaufmann, 1993, pp. 41–48.

[3] Long Duong et al. "Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Vol. 2. 2015, pp. 845–850.

[4] Zhuoliang Kang, Kristen Grauman, and Fei Sha. "Learning with Whom to Share in Multi-task Feature Learning." In: *ICML*. Vol. 2. 3. 2011, p. 4.

[5] Ketron Mitchell-Wynne, Ben Freeston, and Ross Garrett. *Surfline.com*. URL: `https://www.surfline.com/`.

[6] Olivier Moindrot and Guillaume Genthial. *Hand Signs Recognition with Tensorflow.* Stanford University. 2018. URL: `https://github.com/cs230-stanford/cs230-code-examples/tree/master/tensorflow/vision`.

[7] Christopher Re. *Lecture 2: Supervised Learning Setup. Linear Regression.* Stanford University. 2019. URL: `http://cs229.stanford.edu/notes-spring2019/Datasets.ipynb/`.

[8] Sebastian Ruder. "An overview of multi-task learning in deep neural networks". In: *arXiv preprint arXiv:1706.05098* (2017).

[9] Yongxin Yang and Timothy M. Hospedales. "Trace Norm Regularised Deep Multi-Task Learning". In: *CoRR* abs/1606.04038 (2016). arXiv: `1606.04038`. URL: `http://arxiv.org/abs/1606.04038`.

Project code link: https://www.overleaf.com/project/5cffb6ee5bbdf458f95423ed