Improved Mining Shovel Tooth Failure Detection Using Computer Vision-Based Methodologies

MARCH | 2018

Presented by Michael Catalano
Business and Analytics Advisors Practice

World Wide Technology
www.wwt.com

Table of Contents

Abstract

Current computer vision-based methods for identifying broken teeth on mining shovels suffer from a prohibitively high false-positive rate (FPR) of 25 percent. We describe a two-stage methodology for the detection of broken teeth that reduces the FPR to 5 percent. First, we used a Haar wavelet feature cascade based on the Viola-Jones object detection framework to detect the row of shovel teeth from the input image. The second stage is a classification step that takes the detections from stage 1 as input and produces a binary score indicating whether the equipment is intact or damaged. We evaluated two methods for stage 2: 1) Dynamic Time Warping with *k*-Nearest Neighbors (DTW—*k*-NN) and 2) Convolutional Neural Network (CNN). The accuracies of the two methods on an out-of-sample image set were 96.3 and 95.5 percent, respectively.

Business Justification

The steel teeth on mining excavation equipment, like rope shovels and front-end loaders, are wear items that must be replaced as part of regular maintenance. During normal operation, the connection that affixes a tooth to the shovel or loader bucket occasionally fails, causing tooth detachment. A detached tooth presents a serious hazard if it enters the haulage cycle and makes its way into a crushing unit, where it may become stuck and require the dangerous task of manual removal. Furthermore, wayward teeth cause substantial lost time and production due to jammed crushers and damage to downstream processing equipment. Therefore, it is critical to detect when a shovel tooth goes missing as soon as possible so that preventive action may be taken.

Current methods use equipment-mounted cameras and computer vision techniques to generate real-time automated alerts in the event of a missing tooth. While these methods can identify missing teeth with good sensitivity, they produce an unacceptable number of false alarms, which causes equipment operators to ignore the alerts entirely. In some cases, a false positive rate (FPR) of 25 percent has been observed. Due to the relative infrequency of broken shovel teeth, the false discovery rate (FDR) may be greater than 99 percent.

There are several challenges associated with real time detection of broken shovel teeth. For example, the quality of captured images is compromised by a variety of factors. Dusty operating conditions and variations in lighting, location and orientation of the shovel bucket, and background composition can make the shovel teeth difficult to distinguish from the material behind it. Furthermore, the biting edge of the shovel bucket is often partially or completely obscured by mined material during operation, which can cause a failure detection algorithm to produce undesirable results. In addition to image quality challenges, the problem itself does not fall neatly into the paradigm of traditional object detection because the target object is an anomalous nuance of the image subject. We propose a two-stage approach to address these challenges: 1) row-of-teeth detection and 2) equipment status classification.

Methods

The location and orientation of the shovel teeth within the images captured by shovel-mounted cameras are highly variable. The purpose of stage 1 in our approach is to isolate relevant information from the image and disregard the rest. This step both normalizes and reduces the size of the images for downstream processing. We used a Viola-Jones Haar wavelet cascade classifier to detect the row of shovel teeth in the raw image.[1] The Viola-Jones detection algorithm applies a series of Haar wavelet-based features to an image in sliding window fashion to identify regions of the image that likely contain an object of interest. This technique has been used extensively in real-time facial detection, as the nature of Haar-like features makes them effective in detecting rigid, symmetrical objects like frontal faces. Although the Viola-Jones detection framework was designed for facial detection, it can be used to detect other objects as well, provided that the object does not express significant constitutional or conformational variability. The symmetry and rigidity of shovel teeth lend themselves to this type of detection framework.

Stage 2 of our approach performs a binary classification on the detected region from stage 1. We employed two methods independently for the classification step: 1) Dynamic Time Warping with *k*-Nearest Neighbors (DTW—*k*-NN) and 2) Convolutional Neural Network (CNN). Dynamic time warping is a technique used to determine the similarity of time series signals. Unlike a simple Euclidean distance measure, which is sensitive to frequency and phase, DTW uses an optimization procedure that aligns sequences by warping them in temporal space such that the distance between signals is minimized. This alignment enables matching of time series based on underlying patterns, irrespective of non-linear temporal variations. DTW used in concert with k-nearest neighbors (*k*-NN) has been applied in voice identification, human activity recognition

from wearable accelerometer data, and epilepsy diagnosis from magnetoencephalography (MEG) signals.[2-4] DTW—*k*-NN may also be used for image classification, provided that the image subject matter can be reliably converted to time series format.[5] We demonstrate that this technique is applicable in the identification of missing teeth with appropriate preprocessing of the input images.

We also evaluated the performance of a convolutional neural network (CNN). CNNs are shift invariant, multi-layer feed-forward artificial neural networks based on the model of human perception proposed by Hubel and Wiesel.[6] This type of network was first successfully applied by LeCun et al. in identification of hand-written digits and later by Krizhevsky et al. to win the ImageNet competition.[7,8] Since then, CNNs have become a staple in the field of computer vision. With recent advancements in GPU computing, CNNs have grown increasingly complex to solve increasingly difficult problems. We devised an ad hoc CNN to identify missing teeth from images of mining shovels. The implementation of the CNN model along with that of the DTW—*k*-NN classification method are described herein.

Experimental Setup

DATASET, HARDWARE AND SOFTWARE

A total of 285 306x214 greyscale JPEG images were used for model development and testing. The class distribution of the image set was 277 negatives (all teeth intact) and eight positives (tooth missing). One hundred fourteen of the negative images were annotated by manually defining the bounding box around the shovel teeth using the *opencv_annotation* utility in the OpenCV library. Augmentation was used to increase the number of positive samples in the training set for stage 2 classification. The region of the image containing only the shovel teeth was manually cropped out of each of the positive samples, then the cropped samples were subjected to the following random modifications using the *imgaug* Python library to generate 20 new samples for each image (160 total augmented positives): cropping (maximum of 5-pixels from image boundary), flipping around vertical axis, Gaussian blur, inversion, and pixelwise addition (min. = -90; max = 90). This augmentation procedure was also performed on negative annotated images at a rate of two per image to generate 228 augmented negative samples. Additionally, 163 unaugmented cropped samples were generated by applying the trained Haar cascade detector on 500 out-of-sample negative images and selecting only detections with very high detection confidence (minimum neighbors = 50). This obviated the need for manual annotation. All cropped images were resized to 160x40 for stage two training and testing. Training set compositions for

DTW—*k*-NN and CNN were 100:200:140 and 0:200:140 (unaugmented negatives : augmented negatives : augmented positives), respectively. Both methods were evaluated on the same test image set, which was composed of 63 unaugmented negatives, 28 augmented negatives, and 20 augmented positives. The 20 augmented images in the test set were derived from a single positive sample, which was excluded from the training set.

Development was conducted on a Linux machine running Red Hat Enterprise Linux Server 7.4 equipped with an Intel Xeon E5-2665 CPU @ 2.40GHz and Python 2.7.5 installation. The Haar cascade classifier was trained using OpenCV 3.3.0 for Linux. The convolutional neural network (CNN) was constructed, trained and validated with Keras 2.1.0 using TensorFlow 1.2.1 on the backend.

HAAR CASCADE CLASSIFIER

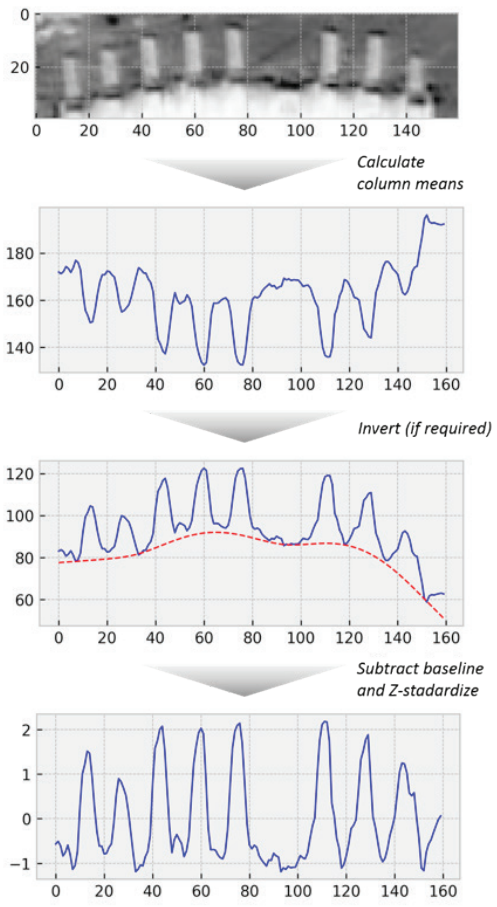
The *opencv_createsamples* utility was used to generate a vector file from the 114 annotated images with image dimensions 75w x 15h. A 10-stage cascade was trained using the *opencv_traincascade* function with sampling rates of 80 positives and 500 negatives, where positives represent the annotated images in vector file format and negatives represent random samples drawn from background images passed to the training function. Background images were generated by cropping the region above or below the bounding box in each annotated image, depending on whether the bounding box occupied the lower or upper half of the original image, respectively. The cascade was trained with a full set of Haar feature types.[9] Testing was performed with the *detectMultiScale* feature of OpenCV. A scale factor of 10% and a minimum neighbors criterion of two were found to give the best overall performance.

DTW—*k*-NN

Cropped images were flattened to one-dimensional arrays by computing the column means for each 2-D image. Curvature in the resulting time series was corrected by subtracting a baseline fitted to each series by asymmetric least squares smoothing (ALS). Images were inverted where necessary to maintain a consistent foreground-background color bias; if the ALS-fitted baseline was a convex function, the root image was inverted, and the column means and baseline were re-computed. Finally, baseline corrected series were normalized by Z-score standardization (Figure 1). The processed time series were arranged into 2-D training and testing arrays for input to the DTW—*k*-NN routine. The optimum time warp window size was determined by leave-one-out cross-validation on the training array using window sizes of 1, 3, 5, 10, 15 and 20. A window size of three gave the highest overall classification accuracy. Class membership (*positive* = missing tooth; *negative* = teeth intact) for each test series was assigned based on the single nearest neighbor (1-NN) in the training set via DTW distance.

FIGURE 1:

Processing steps to convert image data to normalized time series signals for DTW—*k*-NN classification.

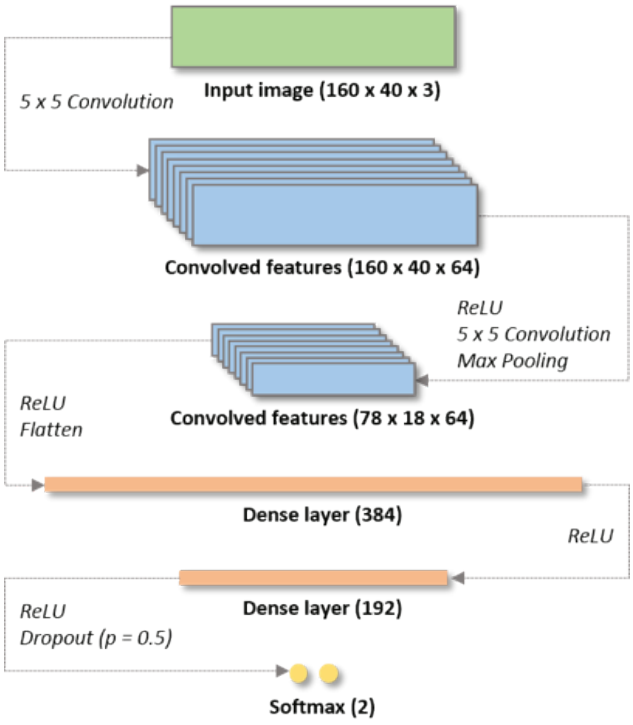


CNN

The architecture of the CNN is illustrated in Figure 2. The network consists of 2 convolutional layers of 64 filters each, a 2x2 max pooling layer (stride = 2), two fully connected layers of 384 and 192 nodes, and a final two-node softmax layer. Each convolutional layer uses a kernel size of 5x5 with a stride of one. The Rectified Linear Unit (ReLU) activation function was used for all network layers, and neuronal dropout was applied at the last fully connected layer (192) with a dropout rate of 0.5.[10] Optimization was performed by stochastic gradient decent with a constant learning rate of 1×10^{-4}. Training was conducted using a batch size of 16 images for a total of 200 epochs. The Keras *ImageDataGenerator* function was used to normalize the input image batches to values between 0 and 1, as well as apply random manipulations to further increase the diversity of the training set: shear factor = 0.2, zoom factor = 0.2, and horizontal flipping.

FIGURE 2:

Convolutional neural network (CNN) architecture.



Results

The Haar cascade model was tested with several hundred out of sample images. Figure 3 shows detector performance on a series of images captured from a single shovel over a seven-minute period. The detector correctly identified the location of shovel teeth in all 18 images with a precision of 100 percent. Our early attempts toward a Haar feature-based detector suffered from a high number of false positives. This was rectified by modifying the background image set from which negative samples were drawn during model training. While Haar cascades are often trained using arbitrary background images, this approach yielded poor performance for our specific use case. Previously, we used images captured from side or rear-mounted cameras where the shovel bucket was absent as background negatives. We found that sampling background information from the front-facing shovel bucket images instead dramatically reduced the number of false detections. Our methodology for background image creation is described in the experimental section of this paper.

FIGURE 3:

Shovel teeth detections produced by the trained Haar cascade. Images are consecutive captures from a single shovel over a 7-minute period (left to right – top to bottom).

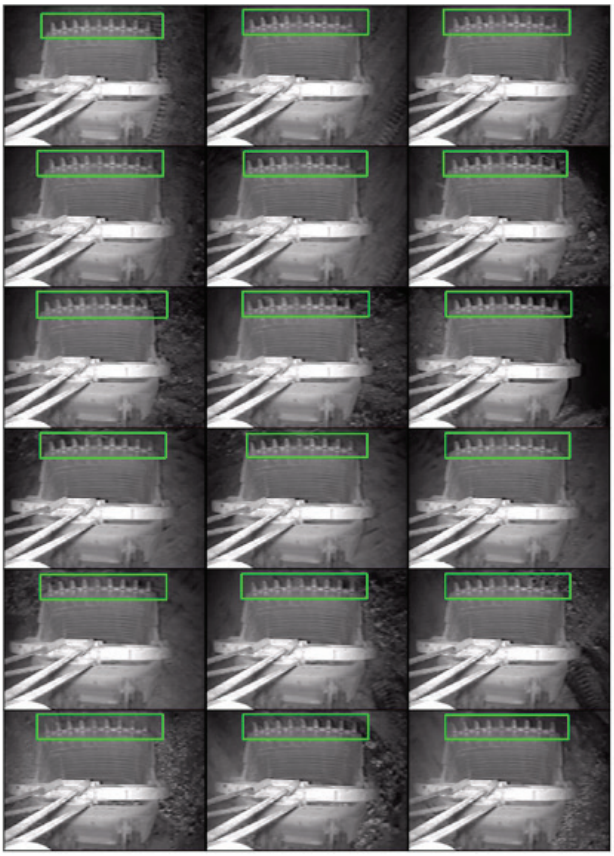


Figure 4 depicts the learned Haar features used for detection at each stage in the cascade. Most of the stages contain at least one horizontally elongated vertical gradient feature, ostensibly to identify a sharp change in vertical contrast between the background and the horizontally oriented shovel bucket. Because captured images often include the lower bucket edge and the bucket crossbar (Figure 3), both of which are horizontal edges, our early attempts frequently mis-identified such image features as shovel teeth. This was likely due to a similar long gradient Haar feature learned by the model. We can account for these adversarial image features using the background sampling method described previously, whereby the model is forced to learn other features that differentiate a row of teeth from simple horizontal edges. False positives may also be managed by adjusting the detection sensitivity via the *minNeighbors* parameter in the detection function call. Our present detection model performs well with a minimum neighbors threshold as low as two, which indicates it is rather disinclined to return false detections.

FIGURE 4:

Haar features learned for each cascade stage projected onto a sample detection region.

The second stage classification methods were evaluated on a hold-out set of images composed of 63 unaugmented negatives, 28 augmented negatives, and 20 augmented positives (positive prevalence = 18 percent). A statistical performance summary for both models is given in Table 1. The accuracies of the DTW—*k*-NN and CNN classifiers were 96.3 and 95.5 percent, respectively. Both methods correctly labeled all 20 images of shovels with missing teeth in the test set with false positive rates of 4.4 percent for DTW—*k*-NN and 5.5 percent for CNN.

Unlike DTW—*k*-NN, which is a non-parametric method, CNNs are prone to overfitting on the training set. We employed several techniques to mitigate overfitting. First, we used a relatively shallow network that contained only two convolutional layers, which constrains the total number of learned parameters. Second, we trained the model using only aggressively augmented images to prevent a learned bias toward texture and lighting. Finally, we applied dropout regularization ($p = 0.5$) on the last layer of the network. The resulting CNN model performed quite well on the unaugmented images in the test set, misclassifying only one of the 63 samples.

In our initial evaluation of the DTW—*k*-NN method, the training set was composed of the same augmented images used for the CNN model. This yielded a classification accuracy of only 83.8 percent. One possible cause of poor performance with DTW—*k*-NN is the endpoint sensitivity of the DTW distance measure. The DTW warping path between two time series must always begin an end at the terminal points. Consequently, variation in the endpoint signals between two otherwise very similar time series increases their separation in DTW space, which can cause *k*-NN classification error. Because our processed signals are inherently sensitive to things like shadows and background artifacts in the raw images, we reasoned that we could improve performance by increasing the number of reference signals to better represent such irregularities. We found that including 100 additional unaugmented images in the training set dramatically increased performance to give an overall misclassification rate of 3.6 percent on the test set.

TABLE 1.

	DTW—*k*-NN (%)	CNN (%)
Accuracy	96.4	95.5
Sensitivity	100.0	100.0
Specificity	95.6	94.5
False Positive Rate	4.4	5.5
False Discovery Rate	16.7	20.0

Conclusions and Future Work

We have demonstrated that the Viola-Jones Haar cascasde object detection framework can be used in concert with a DTW—*k*-NN or CNN classifier to identify missing teeth on mining shovels with a false positive rate of 5 percent. This methodology could be used to improve current industry methods, which produce false alarms for 25 percent of image captures. There are, however, some important points of consideration in the direction of developing a robust, deployable implementation of this methodology. For example, a very limited number of true positive samples required us to use a rather aggressive data augmentation strategy to train the stage 2 classifiers. Both classification algorithms would benefit from increased representation of actual tooth failures in the training set, and testing on more real failure cases would provide a more comprehensive view of failure detection sensitivity. Additionally, DTW distance computation is slow relative to prediction with a trained CNN model. This limits the frequency of predictions in a real time application. CNNs, while requiring a long time to train, do not suffer from this time penalty at the prediction end. We are currently investigating the viability of GPU-accelerated implementations of DTW for real time prediction.

References

1. Viola, P., Jones, M. Robust real-time object detection. 2nd Intl. Workshop on Statistical and Computational Theories of Vision 2001.

2. Muda, L., Begam, M., Elamvazuthi, I. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. Journal of Computing 2010 2(3): 138-143.

3. Seto, S., Zhang, W., Zhou, Y. Multivariate time series classification using dynamic time warping template selection for human activity recognition. CoRR 2015 1512.06747.

4. Khalid, M., Alotaiby, T., Aldosari, S., Alshebeli, S., Alhameed, M., Poghosyan, V. Epileptic MEG spikes detection using amplitude thresholding and dynamic time warping. IEEE Access 2017 5: 11658-11667.

5. Santosh, K. Use of dynamic time warping for object shape classification through signature. Journal of Science, Engineering and Technology 2010 6(1): 33-49.

6. Hubel, D., Wiesel, T. Receptive fields and functional architecture of monkey striate cortex. The Journal of Physiology 1968 195(1): 215-243.

7. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE 1998 86(11): 2278-2324.

8. Krizhevsky, A., Sutskever, I., Hinton, G. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 2012 1: 1097-1105.

9. Lienhart, R., Maydt, J. An extended set of Haar-like features for rapid object detection. Proceedings of the International Conference on Image Processing 2002 1: 900-903.

10. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 2014 15(1): 1929–1958.