

Solving Car Break-ins with Data Science

Ian Andriot



Preview

- Data acquisition
- Preprocessing
- Models: Support Vector Machines, and Convolutional Neural Networks
- Results and opportunities
- Demo of final model
- Questions

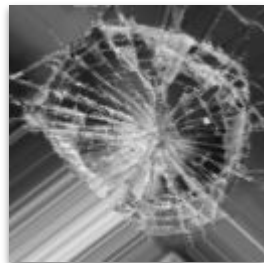
Data Acquisition

- Prebuilt datasets
- Flickr API
- Ended up with:
 - 618 broken glass images
 - 1,150 car images



Preprocessing

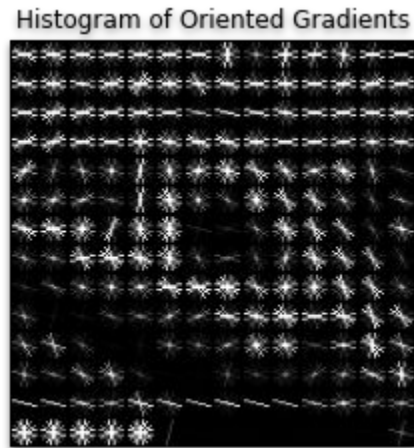
- Converting images to grayscale
- Resizing images to 128x128
- Oversample minority classes for train split
- Image Data Generator to generate more images



Support Vector Machines

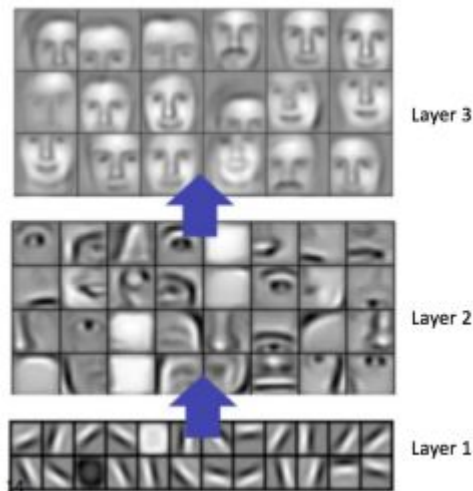
Definitions:

- Histogram of Oriented Gradients
 - Feature descriptor
- Principal Component Analysis
 - Feature reduction
- Bayesian Optimization
 - Hyperparameter tuning



Convolutional Neural Networks

- What sets them apart from regular Neural Networks?
- More layers or more neurons?
- Explored options
- Hyperband optimization



Resulting Models and Opportunities

Results

- Accuracy of 92% for the Support Vector Machine model
- Accuracy of 86% for the manually-tuned Convolutional Neural Network
- Accuracy of 96% for the Convolutional Neural Network after hyperparameter tuning

Opportunities

- A larger dataset to account for more situations.
- Processing a live cam feed to trigger an alert.
- An API that processes requests using this model.

✕

Augmentation Parameters

rotation

0 45 180

width shift

0.00 0.20 1.00

height shift

0.00 0.20 1.00

zoom

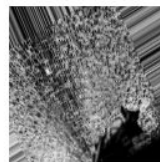
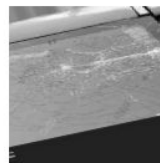
0.00 0.20 1.00

☒ Horizontal Flip

Generate and Predict

True Positives

Images correctly predicted to have broken glass



True Negatives

Images correctly predicted to have no broken glass



False Positives

Images incorrectly predicted to have broken glass

False Negatives

Images incorrectly predicted to have no broken glass

Questions?