Determining Pavement PCIs Using a Stacking Ensemble Learning Approach

Pablo Raigoza, Devin X. Cheng, and Nubia J. Camacho-Reynaga
California State University, Chico
FHWA Turner-Fairbank Highway Research Center in McLean, Virginia
March 11-14, 2024

Background

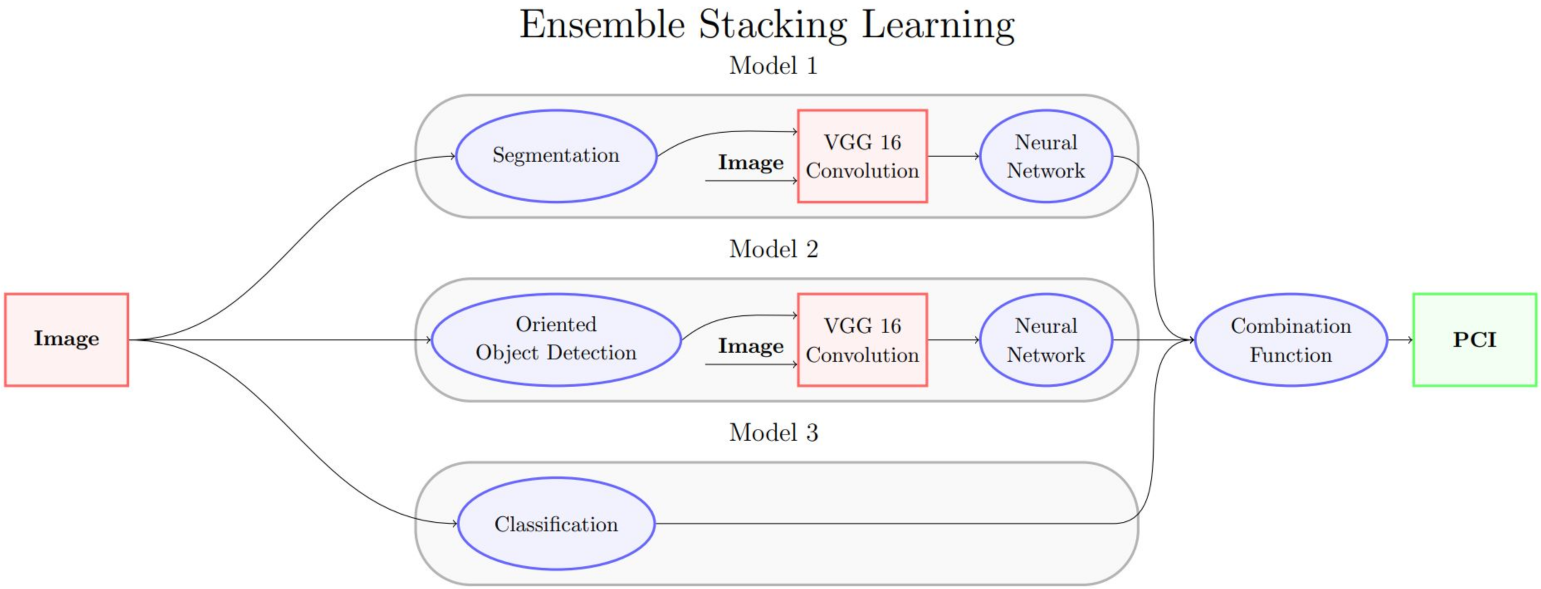
- Goal
  - Application of AI for pavement condition monitoring
- Method
  - Use novel machine learning algorithms to predict PCI for road sections
  - Pictures captured from infrastructure mounted sensors
  - Annotate Training Datasets as needed
  - Any model architecture allowed
- Knowledge Needed
  - Pavement Condition Index (PCI)
  - Pavement Distresses
  - Machine Learning Algorithms
  - Python

```
# Build Model
model = Sequential()
model.add(Flatten(input_shape=(2,7,7,512)))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(101, activation='sigmoid'))

# rewrite the model the compile and fit
model.compile(optimizer='adam', loss='CategoricalCrossentropy', metrics=['mean_squared_error'])
history = model.fit(x_train, y_train, epochs=7, batch_size=4)
```

Model Overview

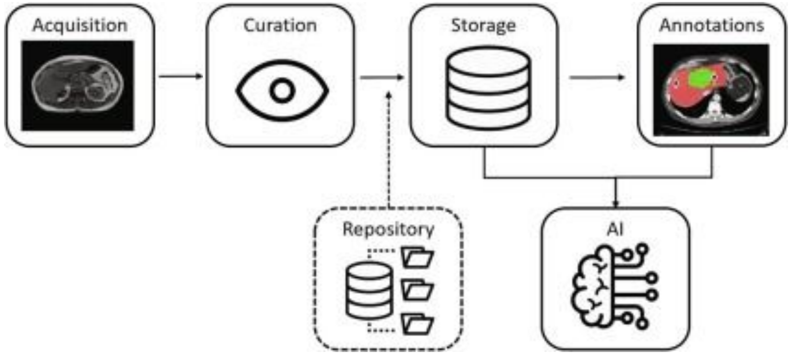
- Stacking Ensemble Learning Approach
- Segmentation and Oriented Object Detection models produce PCIs
- Classification classes (0-100) directly provide PCI
- Use Combination Function to combine three PCIs



Red squares are unlearned inputs, blue ovals are models that we train/improve, and green is output PCI

Data Preparation and Curation

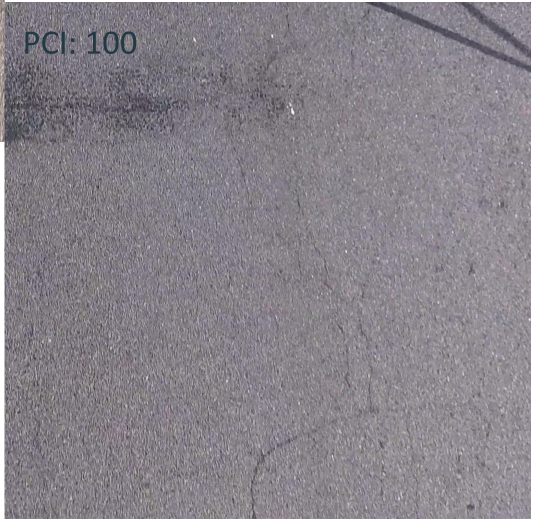
- December 2023
  - First Set of Training Data Received
- January 2024
  - First Set of Testing Data Received
  - Labeling of First Set of Training Data Completed
- February 2024
  - Second Set of Training Data Received
  - Second Set of Testing Data Received
    - PCI info not given



Data Filtering

- Data Issues
  - Images with High PCI, yet identifiable distresses
  - Images with Low PCI, yet lacking identifiable distresses

Unclear Data



Data Augmentation

- Augmentation on Training Dataset
    - Rotations (every 10 degrees)
    - Flips (horizontal, vertical)
    - Attempted brightness change, but did not use


Original Image


Vertical Flip


Rotated 90

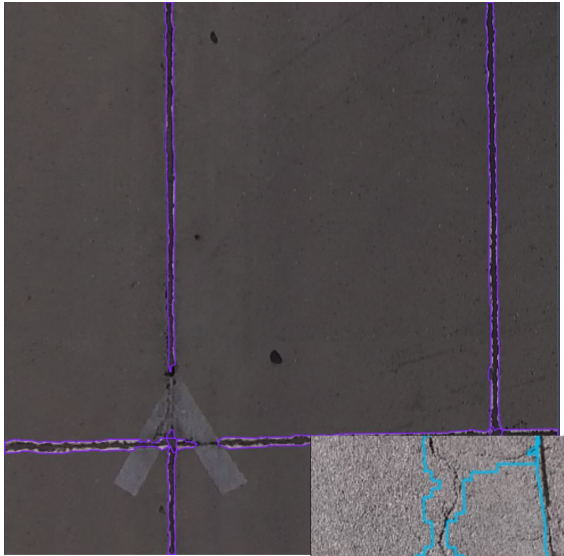

Rotated 270

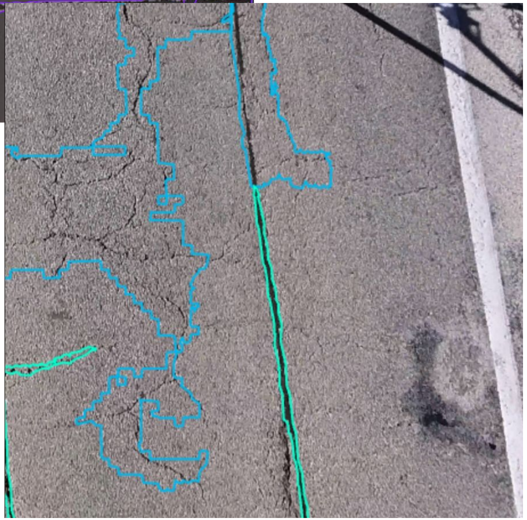
Example Augmentations

Data Labeling

Mainly for Segmentation and Oriented Object Detection

- Labeling with Roboflow
- Used Distress Types
  - Alligator, Medium and High
  - Longitudinal and Transverse, Medium and High
  - Block Cracking, Medium
  - Weathering and Ravelling
- Exported as YOLO v8 labeling format

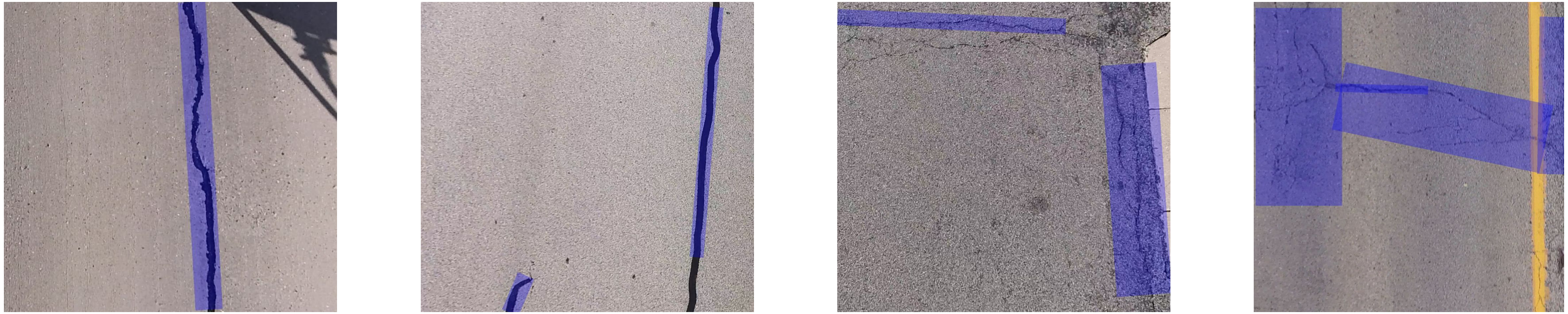




Data Usage Summary

A. Classification
B. Segmentation
C. Oriented Bounding Box (OBB) Object Detection

To predict PCI from segmentation and OBB, we provided the neural network output mask of each model as well as the convolved original image

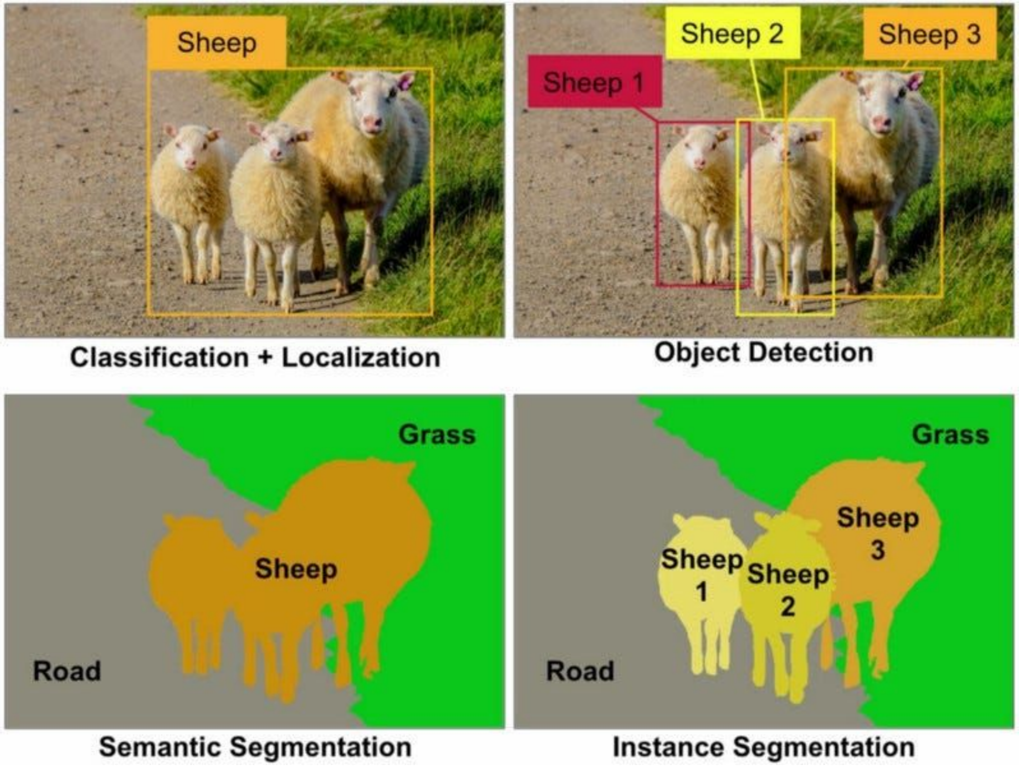


OBB Output Examples

Segmentation

Segmentation

- Instance Segmentation
- YOLO v8 architecture



Oriented Bounding Box Object Detection

OBB Object Detection

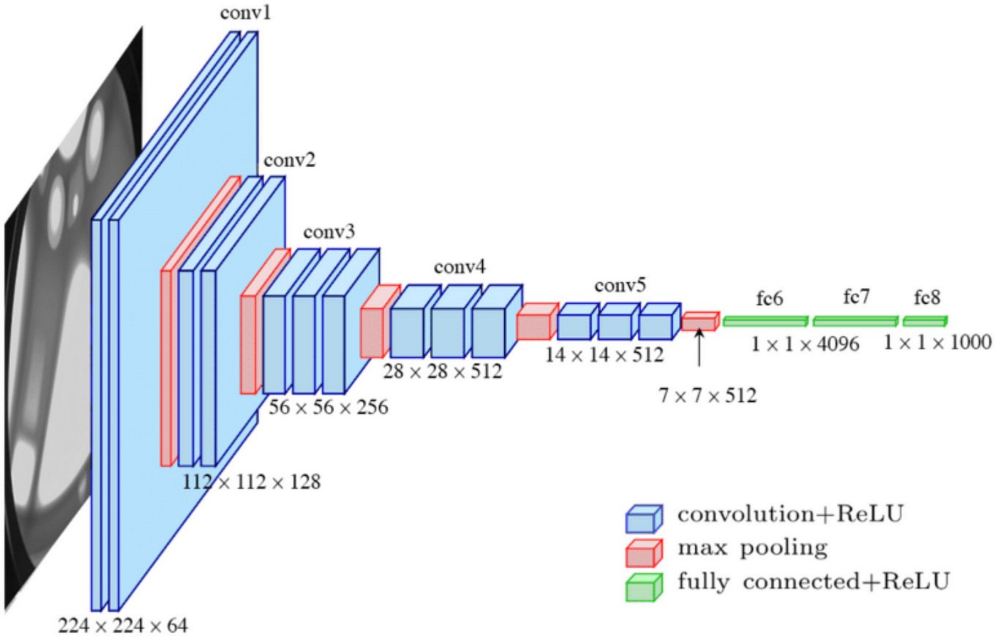
- Oriented Bounding Box Detection
- YOLO v8 architecture

VGG 16 Convolution

Visual Geometry Group (VGG) 16 (16 layers)

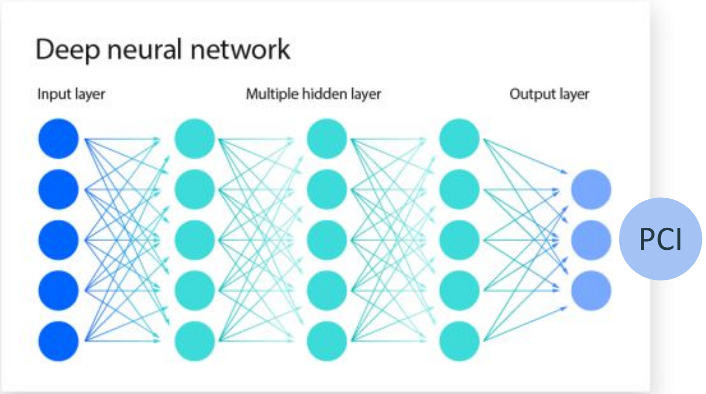
- VGG 16 is a convolutional neural network
- Extracts useful image features

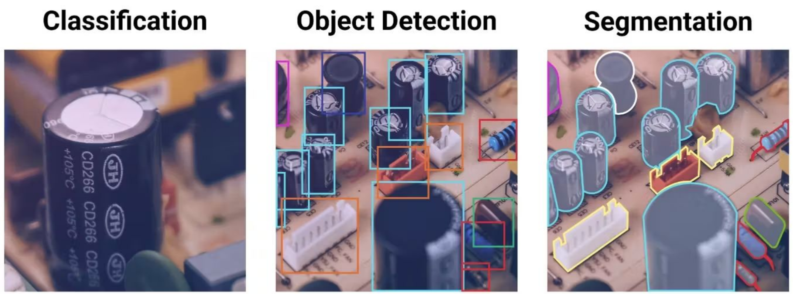
Applying VGG-16 convolution with output of segmentation and original image into standard neural network (NN)



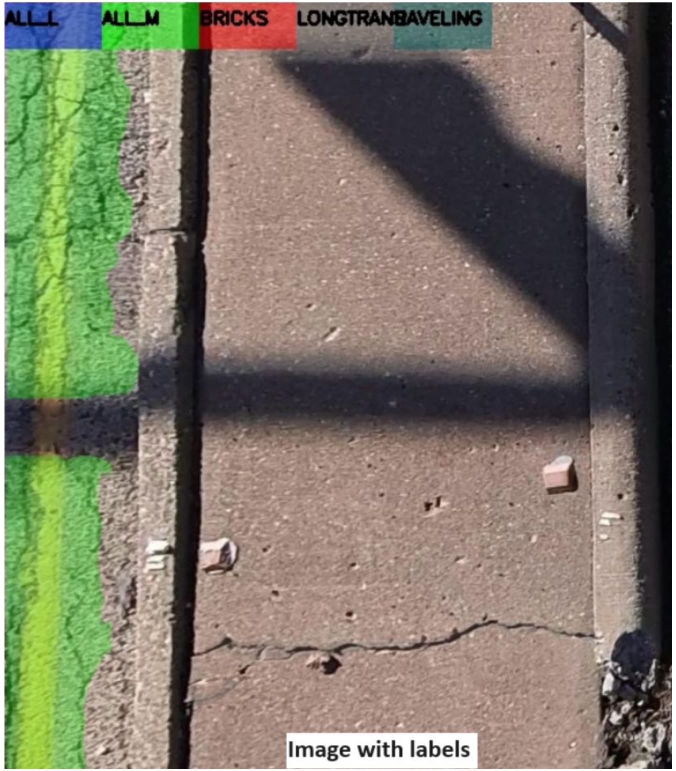
YOLOv8 Classification Model

- Procedure
  - Create 101 classes representing PCI of 0-100
  - Separate images according to respective PCI
  - Augment data
  - Train with YOLOv8 Classification Model
    - Tested nano, small, medium, large, and extra large neural network sizes
      - More nodes, layers, and weights
      - More complexity
  - Optimizers (AdamW, SGD, etc.), Learning Rate, Momentum
- Outputs PCI directly
  - By maximum probability (e.i. argmax)





Results: Segmentation



Training

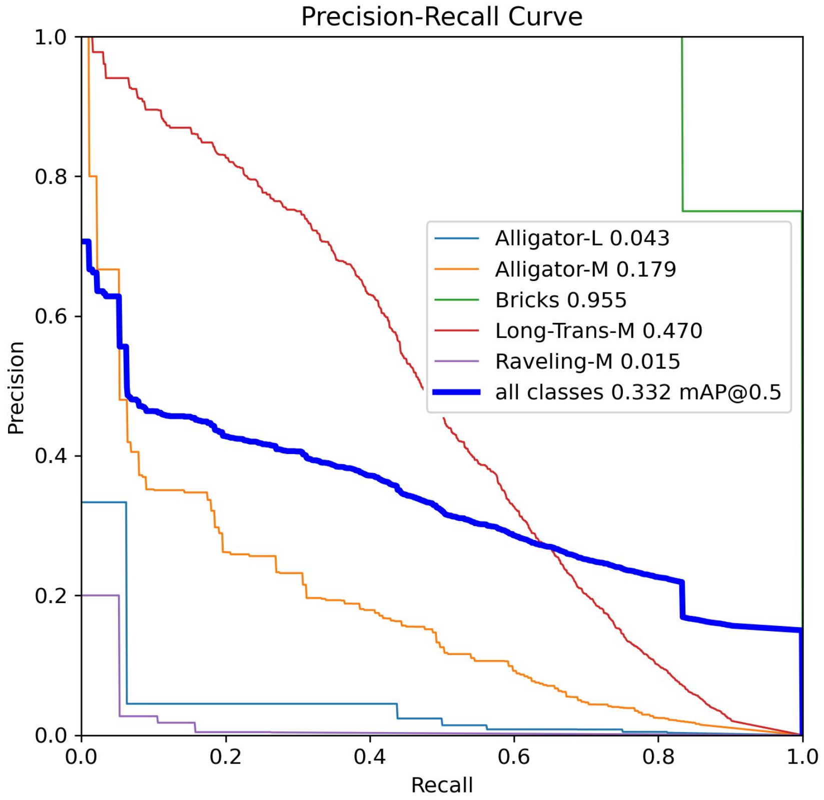
Result

Results: Segmentation (cont)

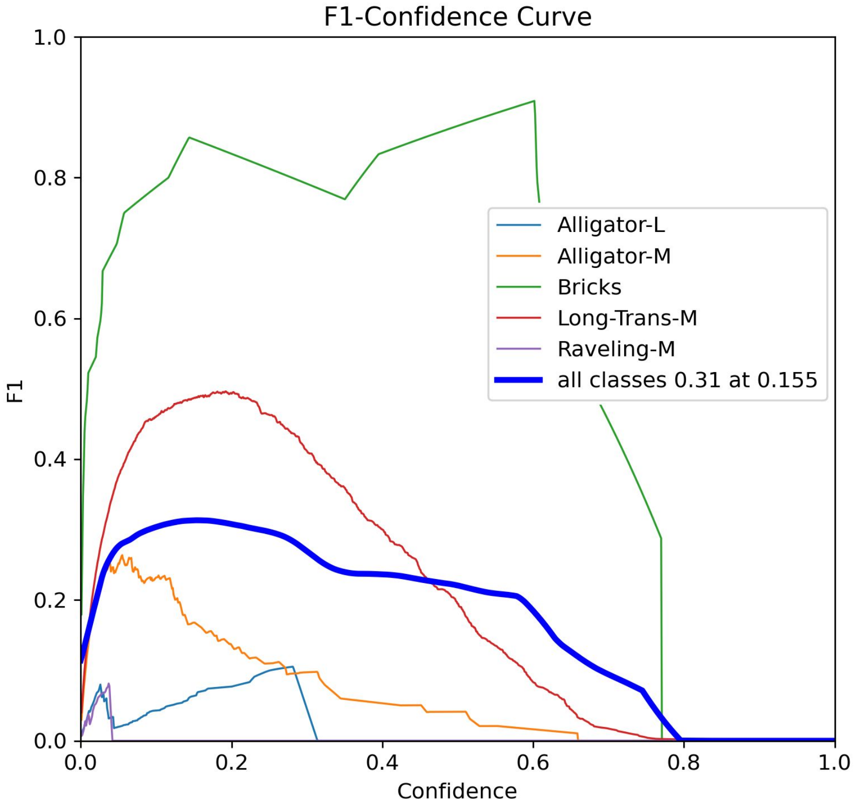
- Shows trade off between precision and recall for different thresholds
- High scores for both mean classifier returns accurate results (high precision), while returning majority of all positive results (high recall)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$



Results: Segmentation (cont)

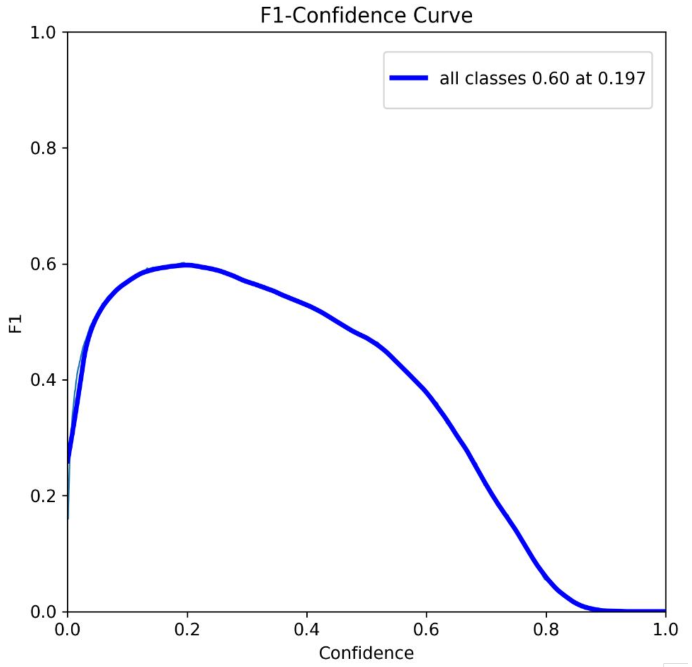


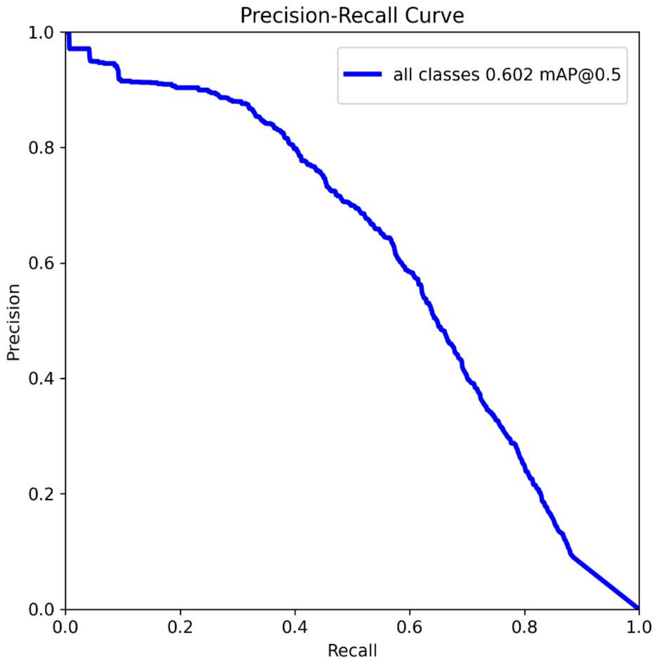
$$F_1 \text{ Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- F_1 is the harmonic mean of precision and recall
- F_1 against different confidence thresholds
- Higher F_1 score indicates better performance
- Confidence threshold

Results: OBB Object Detection

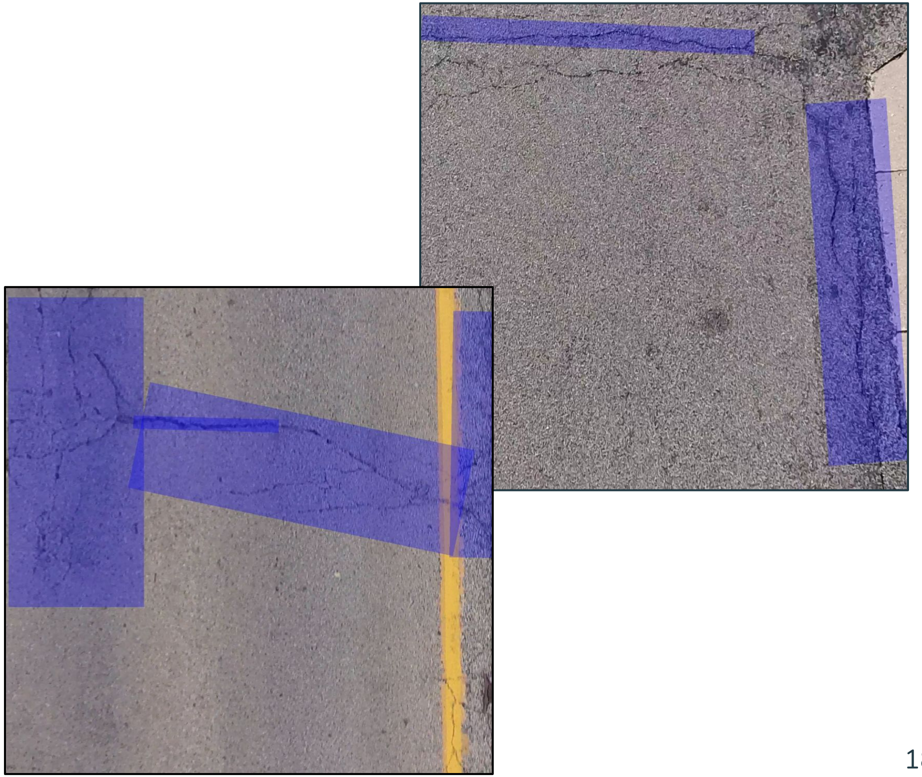
- Combined all distresses into one category
- Increased Recall compared to segmentation model





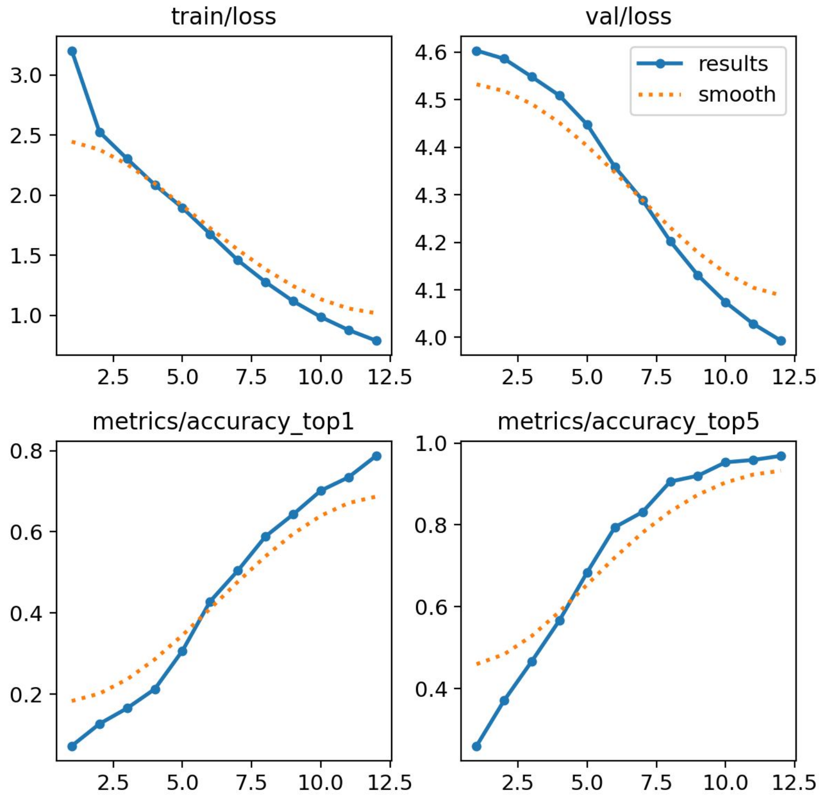
Results: OBB Object Detection (cont)

- More consistent PCI readouts
- Less MAE than segmentation model
- Achieves similar area estimates for distresses as segmentation
  - Much simpler problem

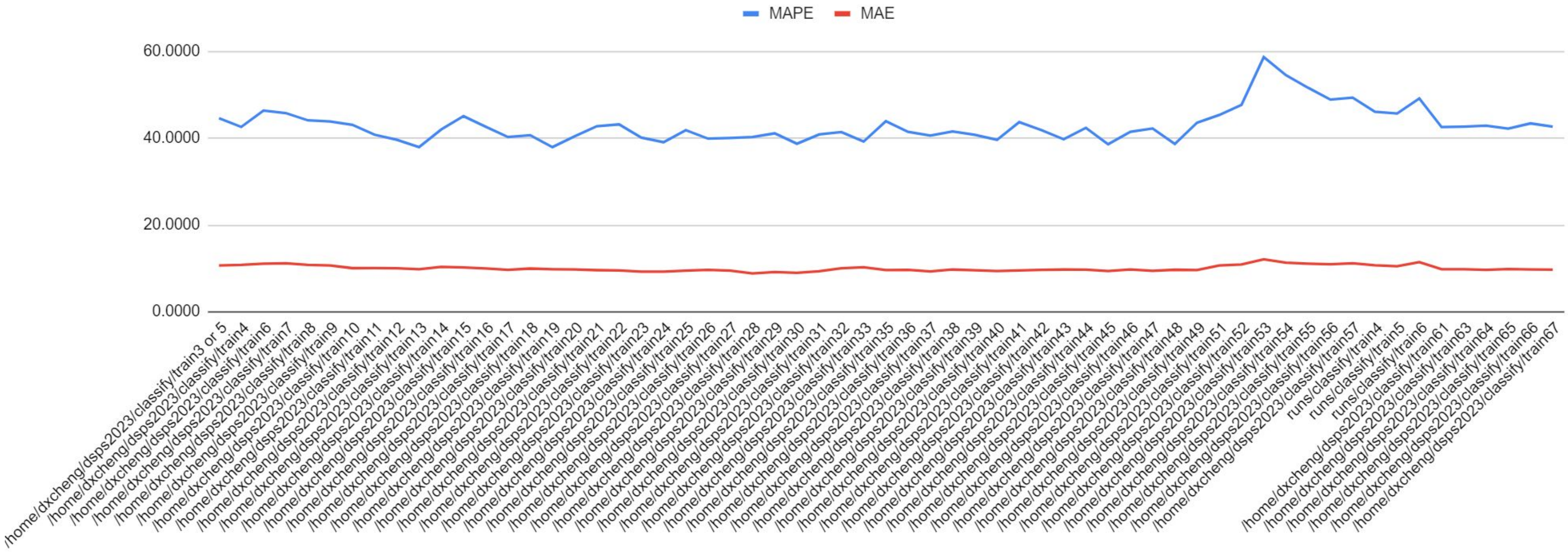


Results: YOLOv8 Classification

- Loss
  - error margin between a model's prediction and the actual target value
- Top 1 Accuracy
  - times the correct label is with the highest probability
- Top 5 Accuracy
  - times the correct label was in the top 5 predicted classes

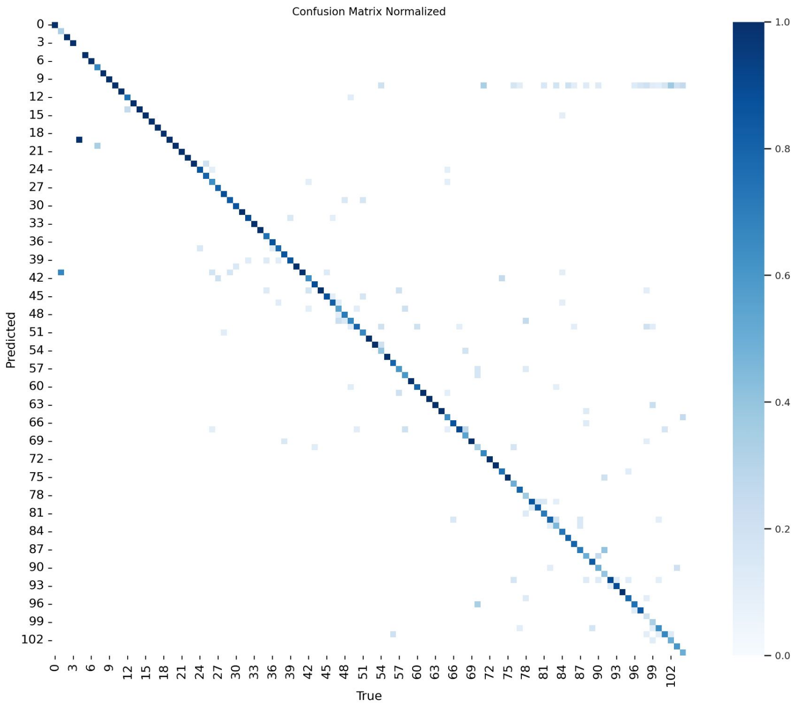


Results: YOLOv8 Classification (cont)



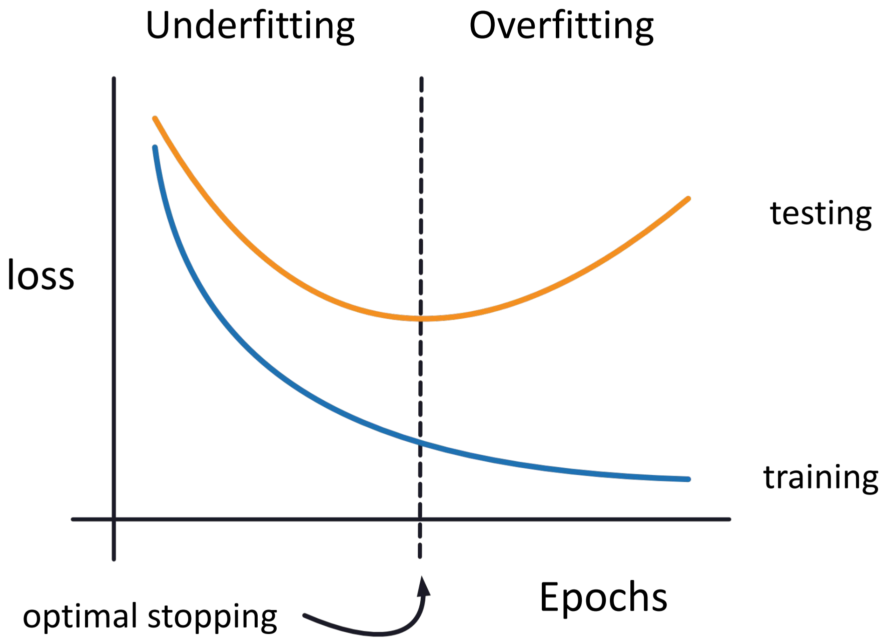
Results: Classification (cont)

- Each row in the matrix represents the instances in an actual class
- Each column in the matrix represents the instance in a predicted class
- The classes, 0-100, are normalized



Results: Classification (cont)

- The model can be overtrained in training and validation set
- Although loss continues to decrease, actual prediction accuracy worsens



Results: Combined

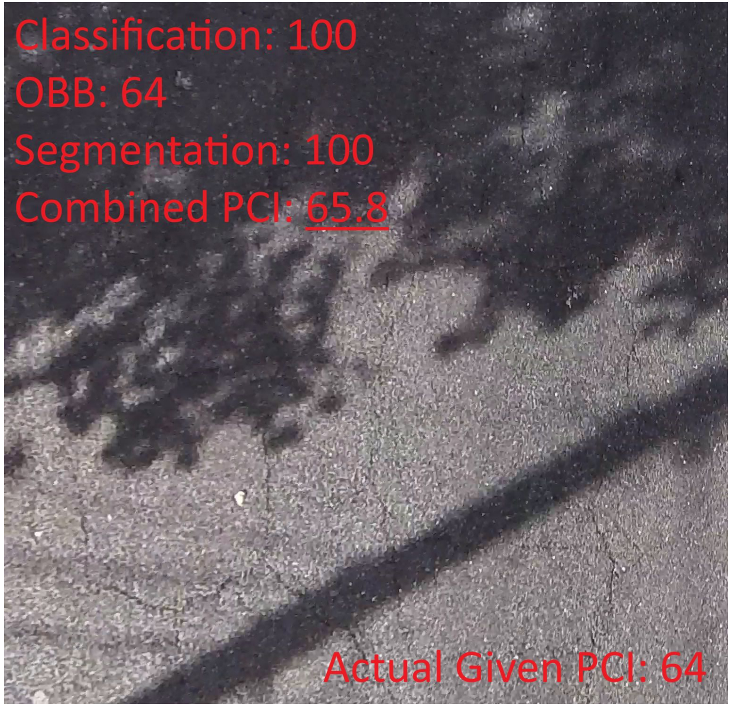
Based on a series of experiments, we derived a good heuristic for determining PCI

$$\boxed{\text{Combined PCI} = \min(C, O) \cdot 0.95 + \min(C, S) \cdot 0.05}$$

- C: Classification prediction
- O: OBB Object Detection
- S: Segmentation Model

Results: Combined (cont)

- In practice we found YOLOv8 had acceptable classifications
- However, often there were instances that segmentation or OBB found distresses, which is why we developed the combined equation



Conclusion

- Machine learning models predict PCI
  - Safer, faster, and more consistent than manual survey
- Advantage of stacking ensemble learning:
  - Ability for it to scale
  - Can add more models to identify key features

Recommendation

- To generate advanced models for PCI predictions:
  - High-resolution pavement images
  - Accurate labels
  - Advanced machine learning architectures
  - Well-designed algorithms
- Problem with 2D Top-Down Images
  - Lack depth information
- Thus…
  - 3D reconstruction to help detect height difference of a road section
  - It would be beneficial to develop a PCI prediction model that considers rutting depression in the future
- Potentially have more models with stacking ensemble learning

Acknowledgements

- Chico State Faculty members
- Chico State civil engineering students who contributed to distress labeling
- Symposium organizers for providing this great opportunity

Thank you, Questions?