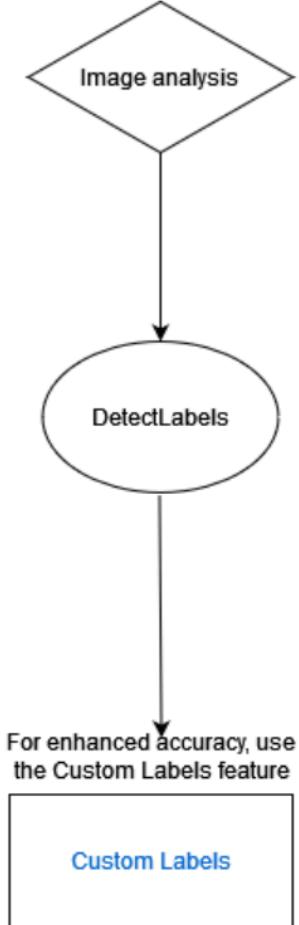


Custom Labels Model `badge_training_model`



This diagram displays the image processing flow. In the source code in file `labelDetection.js`, the function `detect_labels` grabs the uploaded image from an S3 bucket and then calls Amazon's Detect Labels service. The Detect Labels service returns JSON data with general information about the photo. Labels such as "Person", "Tree", "Blonde". The JSON information from `detect labels` is used to determine if there are people in a photo and how many, a person's position in the photo, and hats and sunglasses. Detect Labels detects a multitude of things but it is not specific enough for all of our needs. That is why we have our `badge_training_model` to support the more specific labels for our needs.

Over the semester we gathered data and manually labeled all of our test and training photos. We would add data and retrain the model to see what we needed to improve on. Here you can see over the semester we had 6 different versions of the model with increasing performance.

Projects (1) Info					
<input type="text"/> Search projects by project name					
	Name	Versions	Date created	Model performance	Model status
<input type="radio"/>	badge_training_model	6	2024-01-22		
<input type="radio"/>	badge_training_model.2024-03-06T14.19.59		2024-03-06	0.799	STOPPED
<input type="radio"/>	badge_training_model.2024-03-27T01.33.03		2024-03-27	0.787	RUNNING
<input type="radio"/>	badge_training_model.2024-03-22T01.48.22		2024-03-22	0.758	STOPPED
<input type="radio"/>	badge_training_model.2024-03-14T21.10.28		2024-03-15	0.727	TRAINING_COMPLETED
<input type="radio"/>	badge_training_model.2024-03-21T00.43.03		2024-03-21	0.702	TRAINING_COMPLETED
<input type="radio"/>	badge_training_model.2024-01-29T17.04.51		2024-01-29	0.469	STOPPED

Evaluation results			View test results
F1 score Info 0.787	Average precision Info 0.809	Overall recall Info 0.787	
Date completed March 27, 2024 Trained in 3.190 hours	Training dataset 11 labels, 626 images	Testing dataset 11 labels, 126 images	
Per label performance (11)			
<input type="text"/> Find labels			
Label name	F1 score	Test images	Precision
shadow	0.505	57	0.593
badBackground	0.683	66	0.667
lookingAtCamera	0.825	84	0.723
shirt	0.762	10	0.727
white background	0.761	87	0.743
head tilt	0.821	16	0.778
sunglasses	0.923	12	0.857
hat	0.887	31	0.875
notLookingAtCamera	0.693	27	0.938
hood	1.000	2	1.000
religious attire	0.795	3	1.000
			Assumed threshold

Above is the per label performance of our highest accuracy model version. For example, the shadows label has a lower precision as it is difficult to define how much of a shadow there needs to be to be labeled, this results in lots of false positives in our testing set.

Images (126)[Info](#)

Search images by file name

< 1 2 3 4 5 6 7 8 ... >

good_example4.png

Box #	Labels	Confidence
2	white background True positive	97.9%
3	white background True positive	97.1%
0	lookingAtCamera True positive	74.2%
4	white background True positive	45.7%
5	religious attire True positive	10.4%
1	lookingAtCamera False positive	50.5%

- Show less

good_example1.png

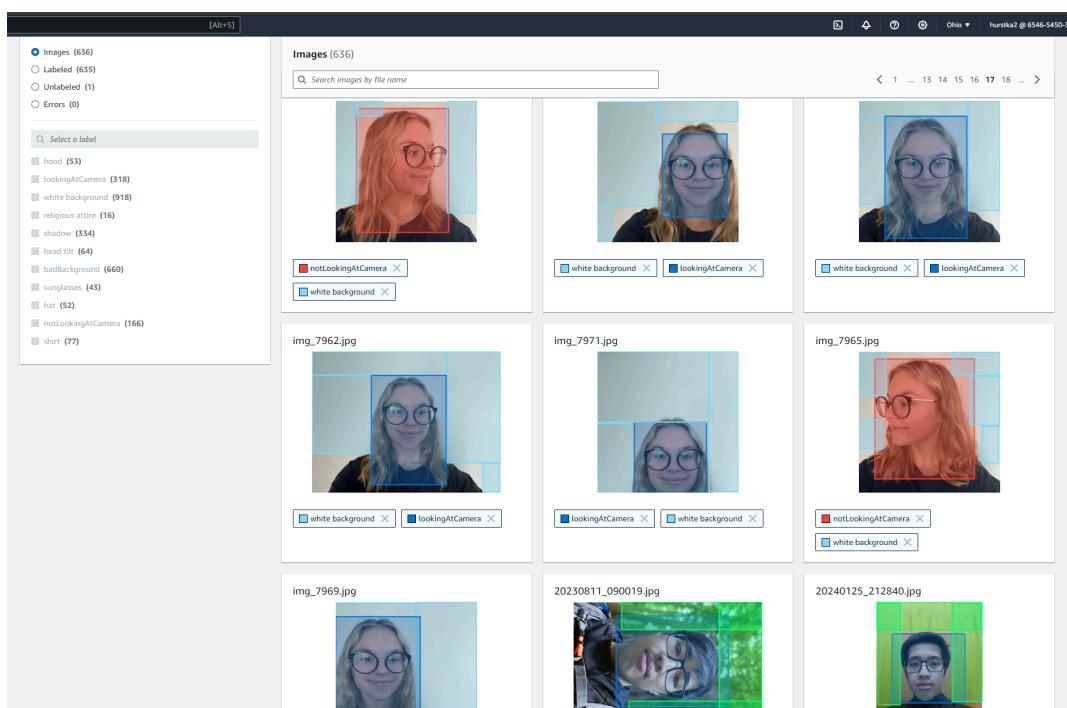
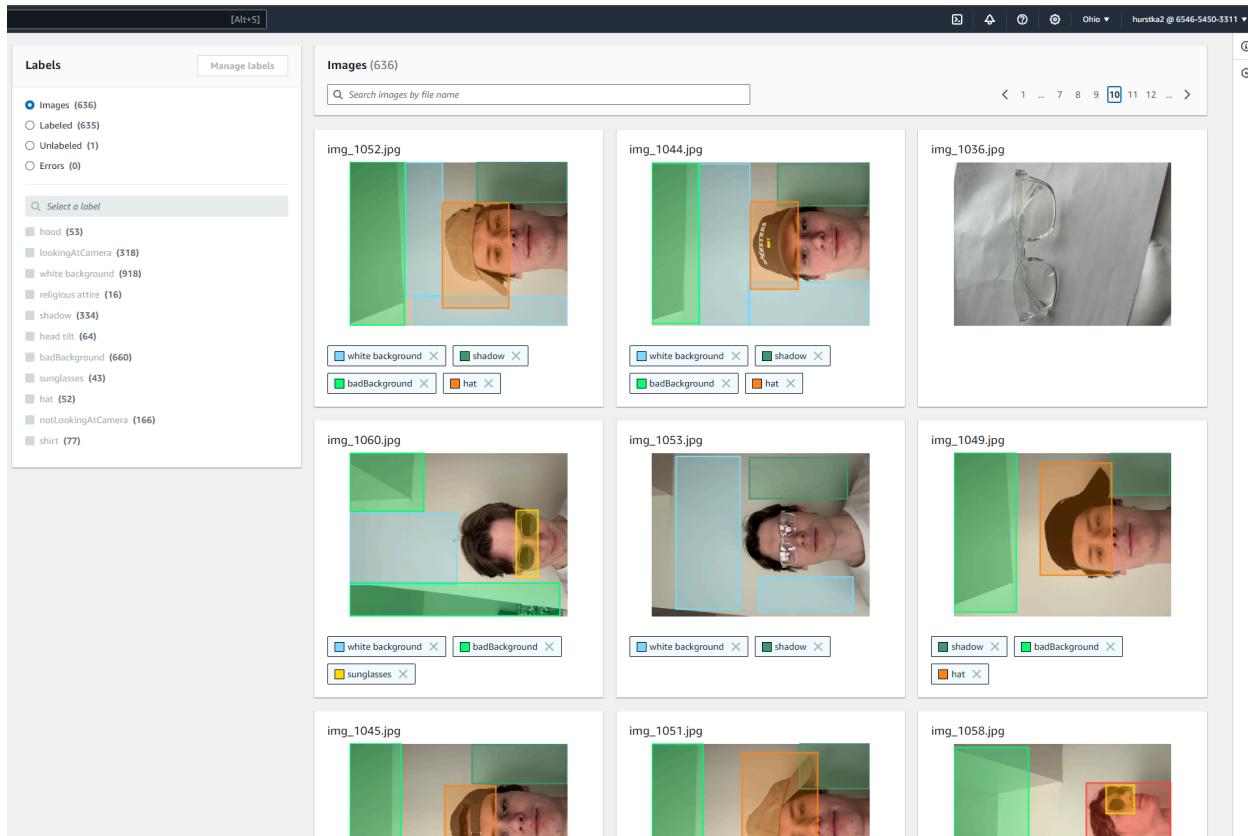
Box #	Labels	Confidence
1	white background True positive	97.8%
0	lookingAtCamera True positive	90.1%
2	white background True positive	77.5%
4	white background True positive	48.5%
3	white background False positive	71.5%
5	white background False positive	46.3%

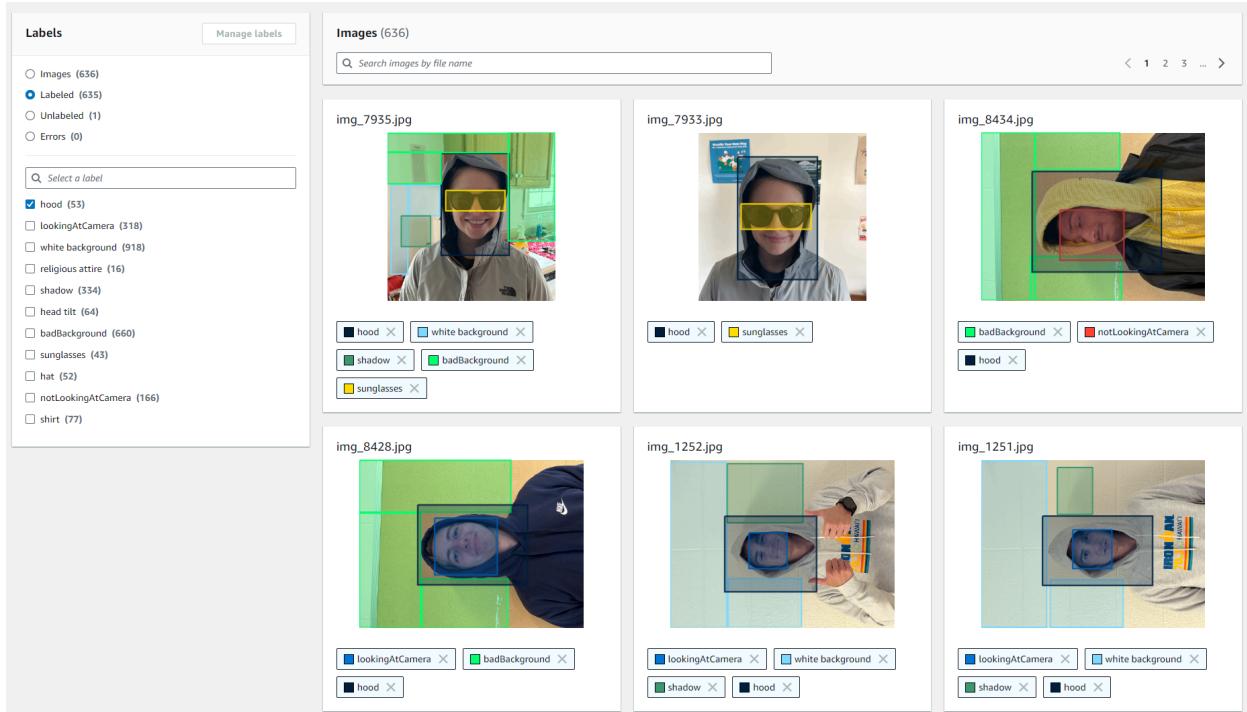
- Show less

We looked at results from our test images to see how accurately our model was detecting different aspects. For example in the right hand photo the model performed very well it just labeled some areas as white background that we did not that resulted in false positives.

Training Data Set

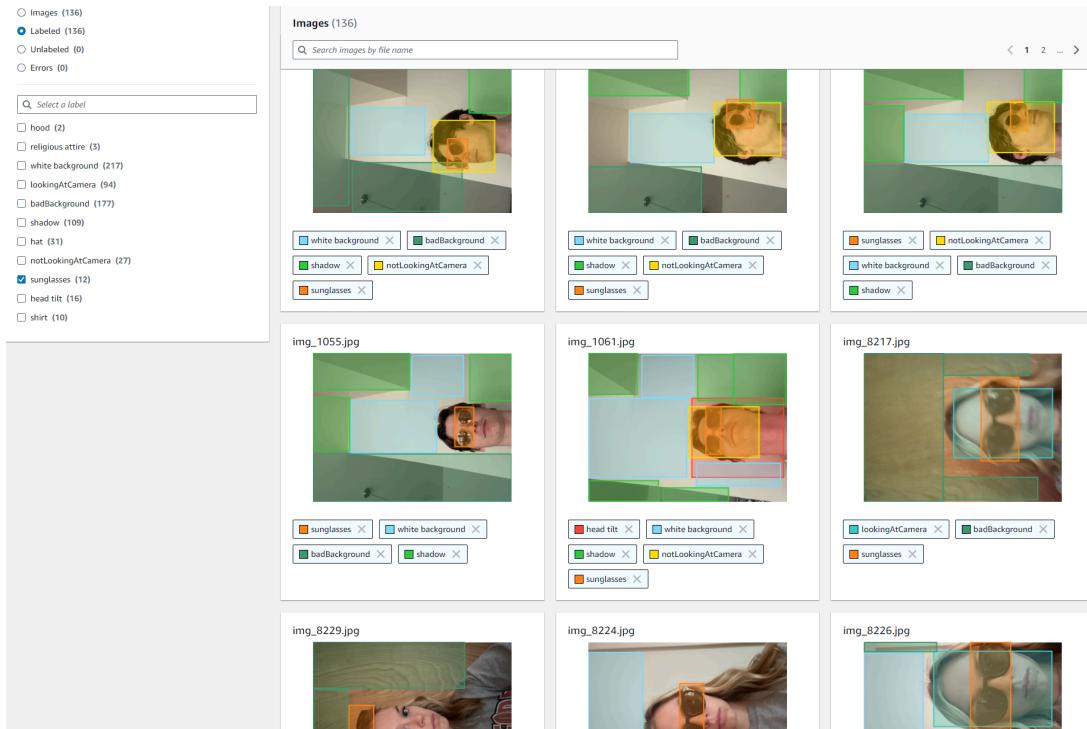
Below are a few examples of photos from our training data set that we labeled. The model will take in these bounding boxes we created and train on these examples to make predictions on a photo given to it.





Testing Dataset

Below are a few examples of photos in our testing data set. We again label all the images and when we train the model, the model will draw its own bounding boxes on the images and the performance of the model is based on how close the model's bounding boxes are in comparison to the bounding boxes we drew.



Labels

Images (136)

Labeled (136)

Unlabeled (0)

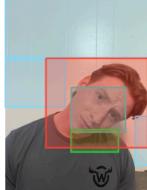
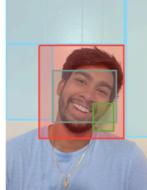
Errors (0)

Select a label

- hood (2)
- religious attire (3)
- white background (217)
- lookingAtCamera (94)
- badBackground (177)
- shadow (109)
- hat (31)
- notLookingAtCamera (27)
- sunglasses (12)
- head tilt (16)
- shirt (10)

Images (136)

Search images by file name

 <p>img_8552.jpg</p> <p><input type="checkbox"/> white background <input type="checkbox"/> notLookingAtCamera <input type="checkbox"/> head tilt</p>	 <p>img_8553.jpg</p> <p><input type="checkbox"/> white background <input type="checkbox"/> lookingAtCamera <input type="checkbox"/> head tilt</p>	 <p>microsoftteams_image_14_.png</p> <p><input type="checkbox"/> white background <input type="checkbox"/> lookingAtCamera <input type="checkbox"/> head tilt</p>
 <p>microsoftteams_image_15_.png</p> <p><input type="checkbox"/> white background <input type="checkbox"/> lookingAtCamera <input type="checkbox"/> shadow <input type="checkbox"/> head tilt</p>	 <p>microsoftteams_image_22_.png</p> <p><input type="checkbox"/> white background <input type="checkbox"/> lookingAtCamera <input type="checkbox"/> shadow <input type="checkbox"/> head tilt</p>	 <p>microsoftteams_image_18_.png</p> <p><input type="checkbox"/> lookingAtCamera <input type="checkbox"/> badBackground <input type="checkbox"/> head tilt</p>

Summary

In summary, our custom model contains 11 labels we are looking for: lookingAtCamera, head tilt, badBackground, etc. Our total data set contains 772 images Jack and Katelyn sourced and labeled over the semester. We have 636 training photos to supply our model with example data and 136 test images to evaluate the models performance with. Our custom model makes inferences about specific things needed for a photo to meet badge requirements and is supported by Amazons Detect Labels to provide accurate results.

Our custom model costs \$4/hour to run and takes about a half hour to start. For budgeting purposes we obviously cannot have the model on at all times so if when demoing our project you get the “Response Error: AWS is handling too many requests at this moment” popup that means the model is either not on or if the model is on it means the request for the JSON from the badge_training_model did not reach the react application which happens now and again especially when multiple people are submitting photos at the same time.