- What progress have you made so far? (share any preliminary results)

  I have completed the following tasks:

  1. Data collection
  2. Data Preparation: The original dataset contained nearly 450 images and corresponding annotation files. The annotation files were in the form of `.xml` files which contained information about the images and multiple bounding boxes for each image. I used the `xml` library's `etree` module to parse the `.xml` files and extract the bounding box coordinates and the class labels. I then analyzed the various features and removed several features that were constant accross all images since these were unlikely to have much predictive power. I also detected several bounding boxes that were outside the image bounds and removed these. This resulted in the removal of about 20 bounding boxes from the expected 1450. I stored the information about the overall image in a file called `images.csv` and that about the bounding boxes in a file called `objects.csv`. `objects.csv` contains a foreign key mapping all bounding boxes to the images they belong to in `images.csv`.
  3. Image Snippet Extraction: Since we need to label each bounding box, I extracted the part of the image that was contained within the bounding box and saved it as a separate image. The image `BikesHelmets0.png` is shown below. Followed by the snippets extracted from it.

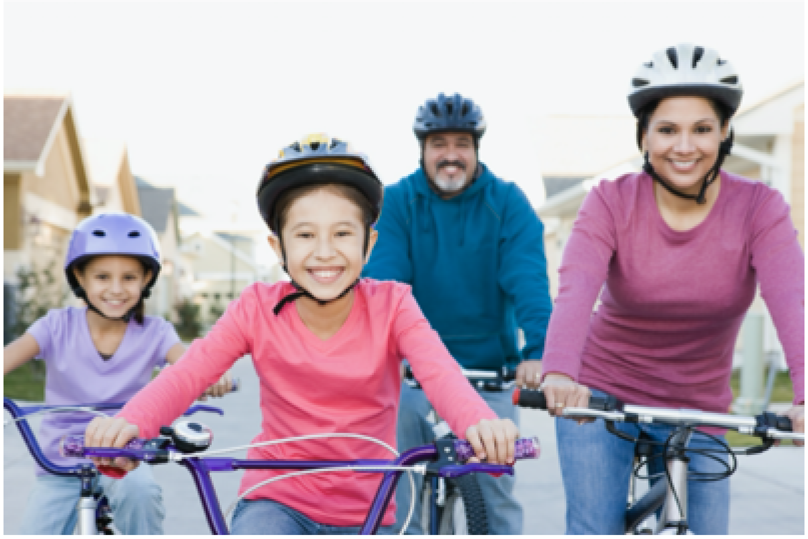
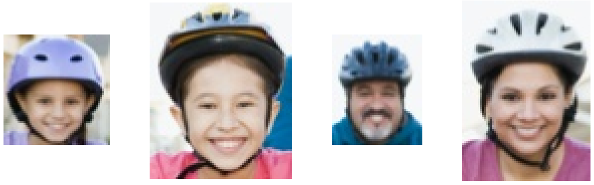


Image BikesHelmets0.png



---

  4. Snippet Resizing: Since I plan on using a CNN for this task, I had to resize all the images of the areas of interest to the same size. I used some visual data analysis

using scatterplots of image resolutions to get a fair estimate of the ideal size. The original scatterplot was dense in certain regions, so it was hard to get a good estimate and had to be zoomed into. I determined that a size of about 45x45 pixels would be a decent place to start.

- Some of the challenges I have so far have primarily been due to the differences between the `torchvision` library and `sklearn` which is what I have predominantly used in the past. However, the `torchvision` library is very well documented and I am beginning to grasp how to use transformers and define custom ones for various stages of the pipeline.

- Here is a tentative timeline for the rest of the project:

    1. **December 3rd**: Complete implementing the transformer steps for image resizing, initial model training and testing.
    2. **December 4th-5th**: Experiment with different model architectures and hyperparameters to improve performance.
    3. **December 6th-7th**: Finalize the model and generate the final outputs.
    4. **December 8th**: Prepare the final report and submit the project.

The code for the project has been sent to the email address provided in the instructions.