Team Members: Daniel Kehr, Ezechiel Dos Reis, Ngozi Omatu
CS585: Computer Vision
April 27, 2021

Semantic Segmentation of Recycling Data Discussion

This document has been a collaborative effort. Work has been completed in unison and with active discussion from all team members.

How the dataset distinguish itself from other semantic segmentation datasets:

The recycling dataset is different from other semantic segmentation datasets as it requires both semantic segmentation (all the objects of the same type are marked using one class label) and instance segmentation (only similar objects are classified in their own separate labels). We needed to detect and annotate four different labels (plastic, plastic bags, metal and cardboard) on to the objects in the clip. Each label is distinguished in appearance and shape, thus similar objects get their own separate labels. However, each label is also their class of objects (plastic refers to plastic containers while plastic bags refer to itself). Also the purpose of the dataset is to help sorting recyclable products that can one day be automatically. This is different to say the cityscape dataset, which is used for evaluating the performance of vision algorithms in urban scenarios or the bat dataset from class as that is used for tracking and predicting the bat's new location.

Challenges:

The dataset posed several challenges for efficient and effective annotating. Often it became difficult to discern not only material edges, but material types from one another. Blurring artifacts present in the image frame compounded with overlapping items caused a vagueness in edge certainty. Cardboard and paper were naturally the hardest to discern due to their similarities in material, but surprisingly metal and plastic posed a similar issue since their reflections would cause a loss in object clarity. This was not only a problem for human users, but for CVAT's OpenCV edge detection algorithms. If an object was unoccluded and not touching objects of a similar color, the automatic detection algorithms performed well, although the outlines would often cut the object instead of allowing a margin of space between the polygon and the material. However, when objects of similar color overlapped, OpenCV's algorithm would fail, forcing manual outlining to be the more viable approach.

Perhaps the most cumbersome challenge faced by the team was due to faulty interpolation of user-defined keyframes. Simple shapes containing a relatively small number of vertices, as well as shapes that did not deform or change location drastically throughout their lifespan, could successfully be interpolated given a limited number of user-defined keyframes. Unfortunately the vast majority of objects in this dataset had complex outlines due to material overlapping and occlusion. It seemed as though the model was calculating in-betweens based on vertex positions, as opposed to overall polygon shape, which resulted in stark shape deformation and polygon outlines that were completely uncharacteristic of the material shape. The load of overlapping edges was lightened by CVAT's automatic bordering setting, but frame interpolation does not preserve shared edges across keyframes, resulting in messy edges that will need to be redrawn. Aside from occlusion, specific types of material posed a challenge to annotate across several frames with interpolation; plastic bags were prone to deformation due to air escaping and cardboard pieces would often move in and out of frame on account of its lightweight property. These material characteristics, as well as the abundance of object occlusion and edge blurring in this dataset, resulted in a largely manual and time consuming annotation process.

Improvements:

There are several improvements to the annotation process that can be made to increase the efficiency of annotating and classifying objects in the dataset. First, using a different type of camera may help to disambiguate different materials by using different attributes of the objects (ex. heat) to classify them. This may take the confusion out of interpreting whether an object is paper or cardboard, plastic or plastic bag, etc. Another helpful improvement may be to introduce some form of custom polygon tracking, such as the object tracking used for the rectangle tool in CVAT, to reduce the amount of manual annotations that need to be made. Based on the examples provided in the manual on GitHub, the rectangle object tracking seems very effective and efficient in terms of annotation and tracking, and developing a polygon version to specifically draw the shape of a given object would be beneficial to the process. A small improvement to the review function would be to make editing issues as reviewers more accessible, as we had difficulties figuring out how to do so. Finally, a feature to classify separate polygons as part of the same object would help clarify annotations for objects that are partially occluded and split into multiple polygons.

Annotation Process and Toolkit Preferences:

Daniel: My typical process began with locating an object that I thought to be of a specific label. Before drawing any annotations I would step through each frame it was on camera to gain a better understanding of its edges and material. Luckily my years of recycling paid off in being able to recognize specific and fun types of garbage. After observing the object's history, I would find the first frame an object was visible in and manually draw an outline using the tracking polygon tool; manually drawing would continue until the object was completely visible. For the case of simple objects, I would step forward three or four frames and either redraw or move and fine-tune the polygon to declare a keyframe for interpolation. The process was more or less repeated for objects exiting the frame. However, for complex objects with shared edges, I would unfortunately end up drawing each frame manually more often than not. Due to the abundance of overlapping objects of similar colors/materials in this specific dataset, most of the objects that OpenCV's edge detection algorithm outlined correctly were easier and quicker to annotate manually. Originally I used my tablet for annotating while using the vertex line drawing function mapped to the Shift hotkey, but it became messy pretty fast and found that individual vertex placement with my mouse resulted in easier to interpolate polygons. Keyframe interpolation was by far the most useful item in the CVAT suite for me. Second would probably be the shared edges functionality; I would have been more impressed with this function though if shared edges were preserved between keyframes. The ease at which polygons could be placed and the intuitiveness of hotkeys really helped speed up the accuracy and efficiency of my annotating. The same goes for how easy it was to add review notes. All in all I was impressed with CVAT's features given the peculiarity of the dataset.

Ezechiel: I annotated the objects by determining which material they would be classified as by appearance, then using the polygon tool to annotate the object itself. If the object was sufficiently large and/or simple enough, I would use fewer keyframes and let the interpolation track the object, manually adjusting each keyframe to improve the interpolation. If they were more complex or were entering/exiting the frame, I would manually annotate each frame until the entire object was in frame. In these cases, I would typically use more keyframes to successfully track the object. Once the object was no longer visible, I used the "switch outside property" to hide it. If two or more objects overlapped or shared a border and all of them had to be annotated, I would use the automatic bordering feature to merge the edge(s) together and produce a cleaner set of annotations, which proved very helpful. This decreased the amount of manual and meticulous annotation at these borders. CVAT locks the position of new annotations by default, so I would have to manually unpin them each time I created a new annotation. I also found the review functionality to be helpful, especially with this dataset where it can be easy to gloss over or misinterpret objects in the frames, so the "issues" placed on the frame by teammates is a great way to make the process more effective.

Ngozi: I annotated the objects by finding the first frame in which the object appears. I used a bluetooth mouse to draw the annotations and did so based on how the object looked in the image at normal size. This made annotation easier to do then using the mouse pad. I originally thought that I had to add a fixed number of points when annotating. This led me to count the edges of the object before annotating before I found out that I could automatically annotate the objects. I also found out much later that i can use the zoom in option to best fit the object's edges into the annotation. This is a tool is the best way to annotate the objects while keeping everything within the space.The automatic annotation was especially useful for me when I realized that I was missing edges of the object. That way I can manually move the points to properly cover it instead of re-drawing the object again. This was also useful when i made a copy of the annotation of an object for the next frame to help save on time. The keyframe interpolation was especially helpful when completing the tasks. I like that CVAT allows us to request for the annotations to be reviewed by other members in the group. It was helpful to get feedback and learn different ways to perform the tasks. Also, there is an option to either write out the issue or use one of the most common issues as a way to communicate with teammates.

It really helped in saving on time and quickly getting the reviews sent back to the assignee on time.