**Capstone Weekly Project Summary**

ObjectIdentifier is an image recognition application that is designed to recognize household objects. ObjectIdentifier is trained on a set of objects that have been manually classified. When a new image is encountered, the application will compare the object in the image to those it has already learned. Comparison will be by shape using edge detection and shape context. ObjectIdentifier will return the closest matching objects.

|  |  |
| --- | --- |
| Week 1 | Project Status: N/A (initial meeting) |
| Tasks Completed/New Functionality | * Research different machine learning implementations and choose one that works well for object recognition * Set up initial repository |
| Comments |  |

|  |  |
| --- | --- |
| Week 2 | Project Status: red |
| Tasks Completed/New Functionality | * Researched several machine learning algorithms that could be used for classification including Naïve Bayes, K-nearest Neighbor, neural networks * I will be using K-nearest Neighbor |
| Comments | I did not start working on my project until Monday this week so I did not get much coding done |

|  |  |
| --- | --- |
| Week 3 | Project Status: green |
| Tasks Completed/New Functionality | * Implemented a K-nearest Neighbor classifier that classifies solid color images based on training data * Researched edge detection algorithms * I will be implementing Canny edge detection algorithm |
| Comments | Color not the best indicator of object similarity. Shape comparison works better. |

|  |  |
| --- | --- |
| Week 4 | Project Status: red |
| Tasks Completed/New Functionality | * Began implementing Canny edge detection algorithm * Began applying Gauss filtering to smooth image and remove detail |
| Comments | I did not put enough time into my project this week. Will do better next week. |

|  |  |
| --- | --- |
| Week 5 | Project Status: yellow |
| Tasks Completed/New Functionality | * Finished implementing Gauss filter * Computed gradient magnitudes and angles to mark initial edges |
| Comments |  |

|  |  |
| --- | --- |
| Week 6 | Project Status: green |
| Tasks Completed/New Functionality | * Finished implementing the rest of Canny edge detection * Found appropriate high and low thresholds for hysteresis * Researched shape context to match objects by shape |
| Comments | I chose thresholds that detect real edges for most images while minimizing extra noise edges.  Some images still have too much noise (car) or missing edges(orange) |

|  |  |
| --- | --- |
| Week 7 | Project Status: yellow |
| Tasks Completed/New Functionality | * Began implementing shape context * Sampled points along edge of image and calculate distance between point pairs. Normalize distance * Started calculating average vector between point pairs to approximate tangent line at each point * Angle between point pairs to be found using tangent line rather than x-axis to be invariant to rotation |
| Comments | Shape context algorithm not complete, cannot classify object yet. |

|  |  |
| --- | --- |
| Week 8 | Project Status: red |
| Tasks Completed/New Functionality | * Finished finding angle between point pairs with respect to tangent line * Constructed log-polar histogram that sort s point pairs by r, theta distance * Calculated cost matrix by matching point pairs from known object to unknown object |
| Comments | Shape context algorithm not complete, cannot classify object yet. |

|  |  |
| --- | --- |
| Week 9 | Project Status: yellow |
| Tasks Completed/New Functionality | * Used Hungarian Algorithm to find min cost mapping between point pairs * Can compare object shapes using min cost mapping (slow needs optimization) |
| Comments | Can compare objects based on shape similarity. Algorithm is slow needs optimization. Not always accurate. |

|  |  |
| --- | --- |
| Week 10 | Project Status: N/A (presentation week) |
| Tasks Completed/New Functionality | * Used another implementation of Hungarian algorithm to solve matching problem(much faster) * Implemented KNN classifier for shapes * KNN trained on a set of 11 images(4 apple, 3 car, 2 computer, 2 face) * Can classify unknown objects(2.5 min to classify) |
| Comments | I am sampling 350 points along the edges of each object. |