**Edge Pruning**

* Only create edges if colour difference is below a threshold to simplify it or improve its performance
* It also allows to keep desired properties like connectivity or preserving important structures.
* You can prune edges after you construct the graph or while you are constructing it.
* How ever when an edge is “pruned off” we add up the number of pruned edges and the weighting to calculate the average
* This average will then be used so the next time when there is an edge that is not pruned, we use the average to influence the weighting of this edge

**Super pixels**

* A necessary evil
* We found a heap overflow error was thrown when we would use images of a high quality (more pixels mean more vertices)
* So, we introduced super pixels to reduce the number of vertices when the graph is being constructed, but in a way that does not distort the overall accuracy of the graph in relation to how it depicts the image
* With super pixels, we implement it in a way where we determine a region size, where each region of that multiple pixels in the image will be mapped to 1 vertex (point) when we convert the image to a graph
* How ever this degrades quality significantly, we combat this by only increasing the region size when it comes to images that have high quality (many pixels) and for other images the region size will be lower, even 1:1
* This kind of combats the accuracy issue because images of high quality (many pixels) will be less affected by quality degradation compared to images with lower quality (fewer pixels)