Feed into overleaf the below to understand formula

% First, we must understand our labels and our outputs. Labels are edge maps, which are essentially black and white (boolean) images, with white representing that there is an edge at that pixel location. The images that we produce as outputs are edge probability maps, where each pixel has an assigned floating point number representing the probability that the model believes it is an edge. \\\\

% The following describes the math behind the evaluation metrics and the formulas for calculating them, but in reverse order, sorted by their difficulty to describe. Important Note: We have commented out the following sections which include ODS, OIS, AP, and their respective math in the interest of saving space for the milestone.\\

% Average precision is rather simple, where we go through every pixel in every image and sum up the probability for all of the pixels that are labeled as edges and divide it by the total number of pixels that are edges. One of the following formulas is below: \[

% \text{AveP} = \frac{\sum\_{k=1}^n (P(k) \cross rel(k))}{\text{number of relevant documents}},

% \]

% where $\text{rel}(k)$ is an indicator function equaling 1 if the item at $k$ is an edge, and zero otherwise. Note that the average is over all edges and all non-edges get a precision score of zero. Another way to calculate Average Precision is integrating over the precision-recall curve. Notice that this metric strictly measures precision because an edge map with all probabilities of 1 will get an average precision of 1.\\\\

% The next metric to describe is per-image best threshold (OIS). The idea behind per-image best threshold is that there is an image has a threshold selected such that all pixels with probability above it is white, and all pixels with probability below it is black. Then, we can conduct an F1 score between the this new thresholded edge map and the label edge map. The value that is determined to be OIS is actually the max F1 score that can be attained by setting the threshold to any value. Let this threshold value be represented by the symbol $\theta$. Mathematically, we can express OIS as \[

% \text{OIS}(\text{img}, \text{label}) = \frac{1}{\text{len(images)}} \sum\_{\text{img} \in \text{images}} \text{max}\_{\theta} \ F\_1 \text{Score} (\text{thresholdImage}(\text{img}, \theta), \text{label})

% \]. When we average this for every image, we get the per-image best threshold, or OIS.

% \\\\The final metric, fixed contour threshold (ODS), is much easier to explain in the context of OIS. The thought is the exact same as OIS, except that the threshold, $\theta$, is kept fixed for the entire evaluation. For testing/evaluation, the threshold value is set for the optimum for the training set. More formally, we can express ODS as \[

% \text{ODS}(\text{img}, \text{label}) = \frac{1}{\text{len(images)}} \ \text{max}\_{\theta} \sum\_{\text{img} \in \text{images}} \ F\_1 \text{Score} (\text{thresholdImage}(\text{img}, \theta), \text{label})

% \]

% The difference is very subtle and lies in the ordering of the maximization and summation. Notice that OIS for a dataset will always be greater than or equal to the ODS of a dataset.

Just for extra simplification or verification, we decided to evaluate the baselines of our model with Mean Square Error (MSE). For preliminary results, the Canny Edge Detector has an average MSE of 3519.6 over the 200 BIPED images, and the Sobel Filter has an average MSE of 2067.4 over the 200 BIPED images. This discrepancy is most likely due to the fact that our Sobel filter is not yet thresholded, while the Canny Edge Detector already has detected its optimal threshold.

Links to understand:

<http://www.cs.ucf.edu/~bgong/CAP6412/lec10.pdf>

<https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/papers/amfm_pami2010.pdf>

<https://answers.opencv.org/question/196702/optimal-dataset-scale-optimal-image-scale-metric/>

<https://ttic.uchicago.edu/~mmaire/papers/pdf/amfm_cvpr2009_poster.pdf>

<https://mftp.mmcheng.net/Papers/19PamiEdge.pdf>

<http://cs.brown.edu/courses/csci1430/2011/results/proj2/lbsun/>