Updates for the month

The major developments in the month of July-August in the Thermal Imaging project have been listed in this documentation.

**New Neck Algorithm use in Function NeckCalc() :**

The new algorithm is similar to the realmeasureneck() algorithm, with few changes and a new function NeckCalc() has been developed. Thresholding method has been dropped in this function and insted of using different values of thresholding a correction variable has been calculated manually and is used to refine the points found.

About the correction variable: Five person’s thermal images were randomly picked up from the data-sets. Each of the images was manually tested and found the gap between the actual neck point and their thermal boundary on all four sides(left, right, front, back)as shown in figure below with blue dots.



The distance between the blue points gave us the correction variable for a single person. After calculating the gap for 5 persons, an average value was found for which is used as the correction variable in the algorithm being 15 units (+&-) for left and right points and 5 units (+&-) for front and back neck points.

**The algorithm goes like-**

1. For Side View Image, mark the minimum(Y) point and the maximum(Y) point on the cropped 1/3rd portion of the blob image.
2. Find the point which has minimum value in X axis and value in Y axis same as maximum calculated above. Name the point say ‘basep’.
3. From’ basep’ trace the region towards the minimum of Y point and find the peaks. Find the maximum peak out of the peaks as a temporary back neck point.
4. Verify the found temporary backneck point if it is being the same by traversing from centroid left location to the maximum of Y point and finding peaks in same manner. If the point is verified then this point will be used to calculate the front neck point.
5. Calculate the distances from the above calculated point to every point in the region Max(Y) to centroid right location point. Find the point at which minimum distance occurs from the backneck point. This point is the front neck point.
6. In the same manner as above find the left and right neck points on the front view cropped image.
7. Refine the 4 points using the correction variable before going on for measurement and label them as prefix new.
8. Calculate the distance between new back neck point and new front neck point and this shall be the major axis (a) length in the ellipse.
9. Calculate the distance between left and right neck points and this will be the minor axis (b) length.
10. Using the Ramanujam’s approximation formula for calculating the circumference of an ellipse find the circumference.
11. Multiply the circumference value with the pixel to physical ratio to find the neck measurement.

**Note:-** This method is working for all images ie, it does not give any compilation errors from person to person, as the old measureneck and realmeasureneck functions did for some people, however the accuracy is not yet perfect. So this function NeckCalc has been used as an along with the other two neck measurement functions in the main program for measurement and the mean value from all the three(if the other two work as well for that image) has been calculated as final neck point. Using error & exception handling techniques the results for other images who used to give error using either of the two old neck measurement functions the neck measurement has been taken.

Another function plays an important role in calculating a very accurate value of neck for some and has been used with error handling in the main program is Neckhandle(). This function uses the old two neck measurement functions and thresholds the images at all values in range from 5 to 70. It then calculates the neck at each threshold value and then stores only those values which are between 12 and 18 (12 and 18 have been used as a general minimum and maximum neck measurements a person can have by referring to original measurements in the data -sets). Then mean of those values in the range is being calculated as a measurement for neck. However this function also was not accurate for all the images that is why in order to globally use these functions for all images a lot of complexity has been added in them using all of them together. You can find them by the names of those functions along with the main code on **GitHub(username- RtjShreyD).**

**Result-** Struggling from having 4, 5 persons out of 14(that were working with the global code) in 6%(+or-) error range, now the scenario has changed to a maximum of 22 people whose images are working and returning measurements. Out of these 22 people 14 are in a range of 6%(+or-) error and still working on a little improvement based on the present dataset available.