

Authors: Bhaskar Chandra Trivedi, Nolan Gutierrez, Pritam Pintu Karmokar, William J. Beksi

How to use the algorithm: When selecting the algorithm, it is recommended not to pipe the algorithm into the output of other algorithms. The algorithm will place bounding boxes around objects in each image specified in a batch. These objects must meet a threshold which is specified as a parameter to the algorithm.

Algorithm selection: This algorithm may be chosen when the user wants to quickly know if there are people in an image. The performance of the algorithm depends heavily on the specified threshold parameter. This algorithm is best used with high resolution images where a person may not be able to analyze the image as quickly as the model can.

Parameters modification: The default threshold for PersonThreshold is 0.5. This value can be changed such that objects which have a smaller probability of being a human are also identified as a human. For example, a threshold setting of 0.4 will identify more objects as people even when the model is unsure about its conclusions.

Examples: The following are examples of the algorithm being run with different thresholds.

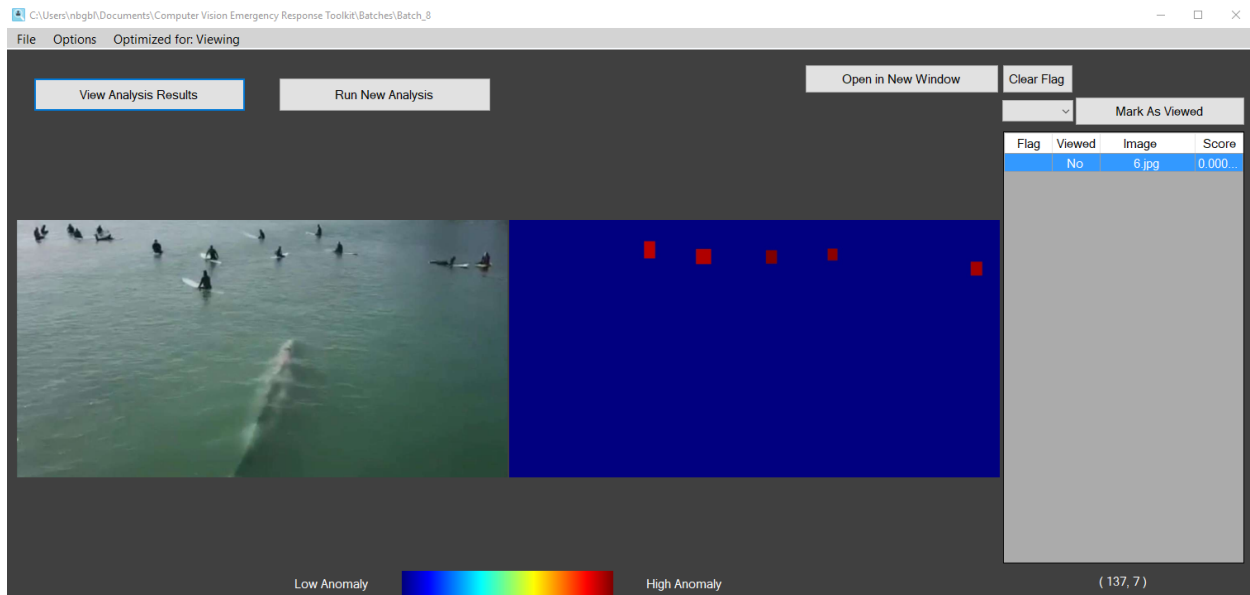
PersonThreshold of 0.99

The screenshot displays the 'Computer Vision Emergency Response Toolkit' interface. The main window shows an analysis of an image (6.jpg) with a score of 0.0. The interface includes a menu bar (File, Options, Optimized for: Viewing), buttons for 'View Analysis Results', 'Run New Analysis', 'Open in New Window', and 'Clear'. A table on the right lists the analysis results:

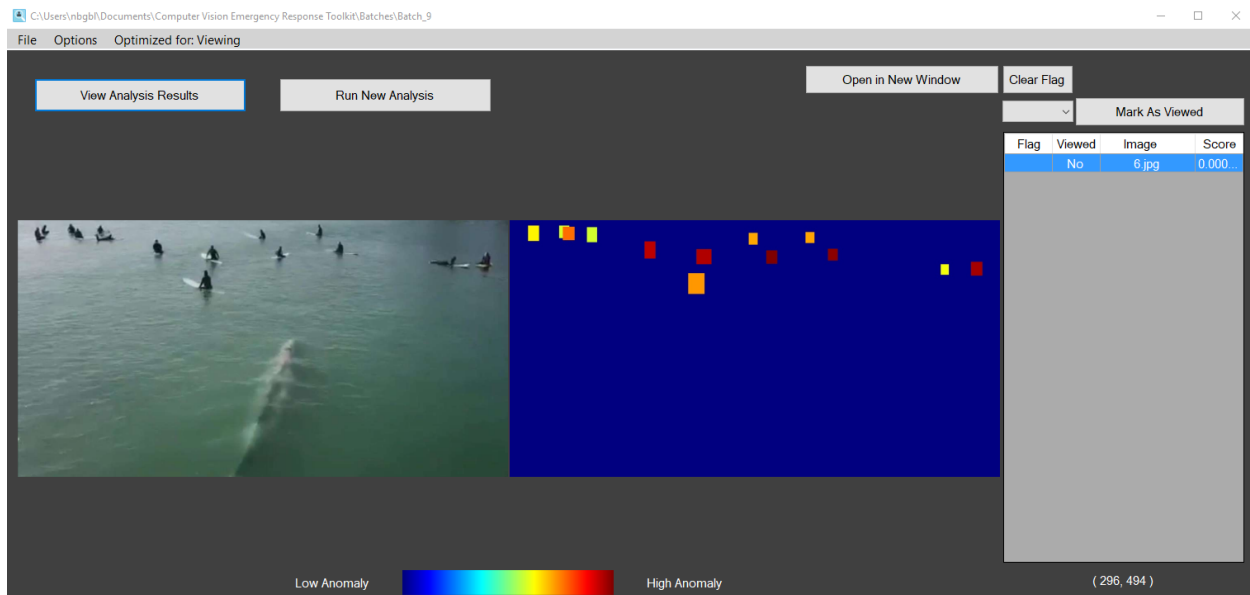
Flag	/iewed	Image	Score
No		6.jpg	0.0...

Below the image, a color bar indicates 'Low Anomaly' (blue) to 'High Anomaly' (red). The image shows a group of people in a body of water, with a large red area indicating high anomaly detection. The bottom right corner shows the coordinates (7, 262).

PersonThreshold of 0.7



PersonThreshold of 0.4



Notes: Before building the project in Visual Studio, please verify that the 'yolov3.weights' file is present in the 'Computer-Vision-Emergency-Response-Toolkit-CVERT-Competition-Master\Computer Vision Toolkit\Computer Vision Toolkit\lib\Algorithms' folder. If it is not present, it can be downloaded from <https://pjreddie.com/media/files/yolov3.weights>.

Acknowledgements: The authors acknowledge the open source community for providing source code used in the development of algorithms for this project: <https://github.com/experiencor/keras-yolo3>.