(Very) High Level Description of ***Backend Algorithm:***

1 – Initialization (Physical Setup)

I. Camera is mounted on a fixed position in venue. The camera is/can be part of a cluster of cameras.
II. Each camera faces a designated area of the venue or location. The designated area is *initialized* physically during setup a single time to provide a clear focal view of the area for density approximation, minimizing other extraneous areas as much as possible by use of the hardware (camera) features itself.
III. The camera will be in night vision mode, regardless of lights at the venue. This allows for better low-light performance overall, as infrared beams (LED's) will illuminate the areas and result in a more hue-even image.
IV. A reference image ("reference") will be taken with the camera of the **unpopulated** focus area and stored in the individual database for a particular venue.
V. Camera will be securely connected to the internet and configured with a standard supported protocol (RTMP, RTSP, HLS, etc.) to stream consistently to a server ("server") in native resolution.
VI. Steps I-V are repeated for each camera in a cluster if necessary.

2 – Image Processing → Usable percentage data to feed to front-end

I. From Step V in part 1 above, a camera will be streaming live video data to a server. The server will receive the video as a constant stream with a supported streaming platform.
II. Server will then analyze the stream of video by capturing a fixed-resolution photo from the stream at a fixed interval, for example every 5 seconds. Each image will only be stored on the server for processing and be promptly deleted after the next image is successfully processed, for security maximization and storage minimization.
III. The image captured from the camera will be analyzed by the software. The software on the server will process each image by converting it to binary form in order for the necessary algorithms to be applied to it logically. A ***color map*** (RGB, HSV, YCbCr, L*a*b) will be applied to the binary image and the image will be reduced via the reduction algorithm to provide an approximation of the human density by removing any non-human objects. The image will then result in a binary photo with humans being illuminated (binary 1) and all other objects off (binary 0).
IV. Resultant processed image will be fed through a secondary base algorithm to provide a contour map of the processed image. The contour map ("contour") will now serve as the basis for density approximation.
V. With the reference from Part I – Step IV, the contour will be comparatively analyzed to predict the density of humans in the particular image.
VI. The resultant data (as an integer or float percentage, 0-100) will feed then into a front-end *(not described in this document*), providing an estimation in percentage [of the reference] to how humanly dense the area is.
VII. All steps can be repeated for a cluster of cameras; in the case of a cluster, Part 2 – Steps 1-V are completed normally and the data from Part 2 – Step VI is either aggregated as a simple sum of all camera-cluster image data or sent separately to the front-end.