David Ologan

**Team Bangalore Final Presentation Script**

Our method is composed of 5 main steps. The first is to preprocess our worksite by selecting four fixed points in the cameras field of view. This allows us account for the unique perspective of our camera and generate the respective transformation matrix. Next, we read our video stream, and divide it into individual frames that Yolo can process. Applying Yolo allows us to identify workers in each image, and using the outputted bounding box, we can map worker positions using our previously defined transformation. Finally, we generate a map of worker positions for each frame and stitch that back together to form an output video.

Given a video stream of our worksite, we need a way to map worker positions in the image, to worker positions in space. To do this we rely on the fact that the camera’s position on the construction site is fixed and given 4 known positions we can generate a transformation that maps pixel values to world coordinate values. This works because we assume all of the workers are on the same plane, implying that their height is the same all times. So as you can see above, x,y are world coordinates and u and v are pixel coordinates.

And now I’m going to pass it onto Arpit who’s going to talking a bit about YoloV7 and how we’ve implemented it in our method.

After we’ve run yolo on each of our frames, were given the pixels value for the center of the bounding box, and its dimensions. By averaging the bottom of our bounding box, we can find a pixel representation of each worker’s position in each frame. We transform those pixel values into world coordinates, using the matrix we found earlier.

To generate our output video, we scale our world coordinate positions by the size our output frame and boundary. We then stitch together each frame to create an animation. With our predefined areas, we can accurately classify workers in danger areas.