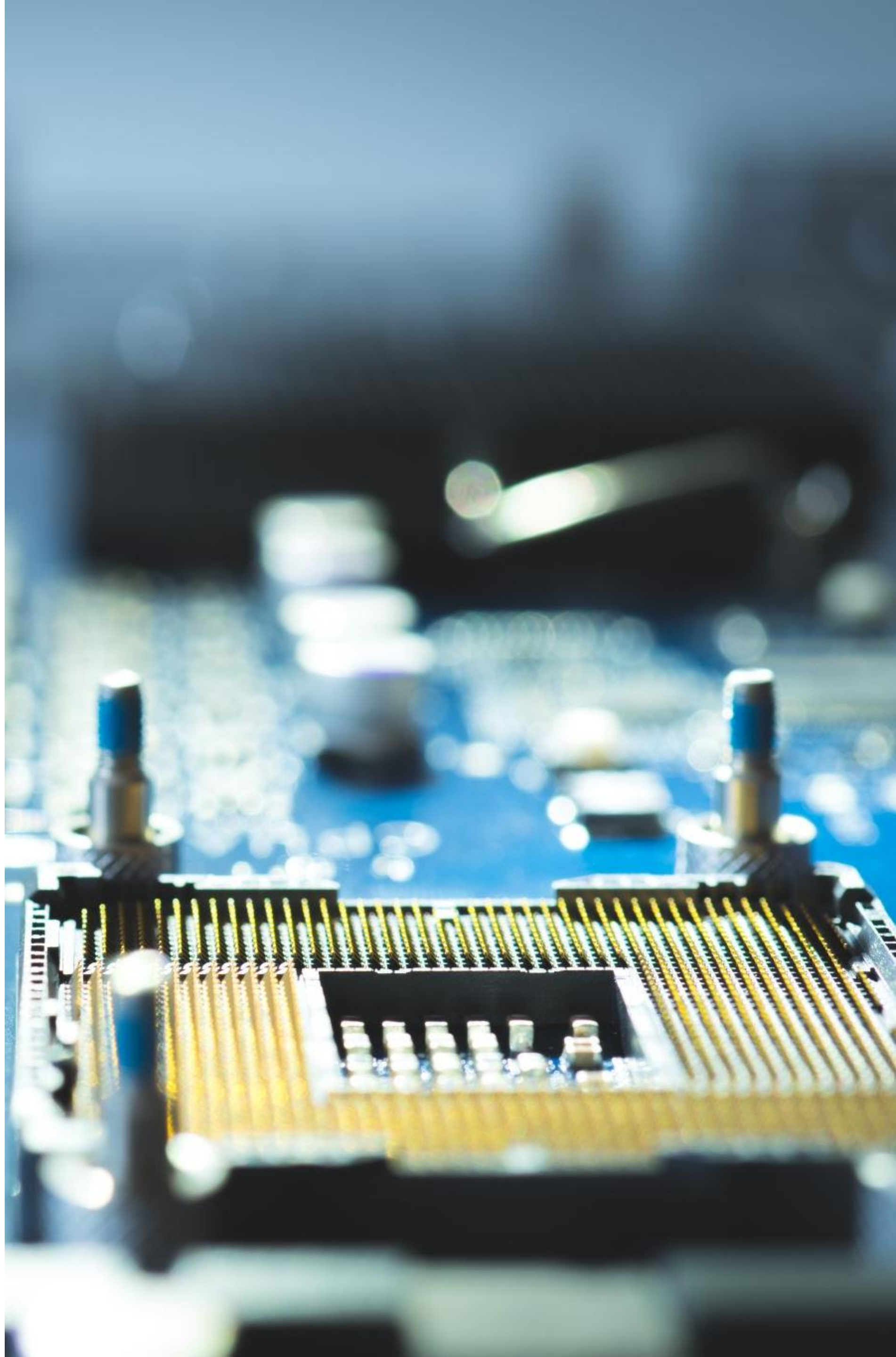
Thermal Occupancy Estimation

KARTHIK VISWANATHAN

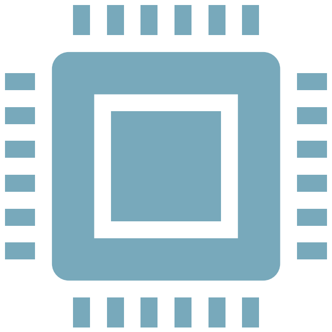
Why use Thermal Sensors ?

- In order to discuss this, let us first figure out the alternatives:

    1. Surveillance cameras : Privacy
    2. MAC Address trackers : Everyone may not carry a device
    3. Carbon Dioxide Sensors : Delayed diffusion rates



How do we use a thermal sensor?



Mounting sensors across the room can cause problems like field of view overlaps in small rooms and insufficient field of view in larger rooms.

In order to fix this problem, we mount the thermal sensor over the entry/exit points in a room and the count is only updated after an entry/exit.

Processing the frames

- The frames are proposed to be processed in the following steps:

    1. Background Subtraction
    2. Event detection
    3. Event Classification

Background Subraction

- In this process, we try to separate the foreground and backrground events by means of a binary hypothesis test.
- A single global constant cannot be able to model whether a pixel in a frame belongs to the background or the foreground.
- For this purpose, each pixel is treated as a Gaussian random variable with its probability density changing with every frame.

Why a random variable ?

- We model a pixel to be a random normal variable with its mean being a Running Gaussian Average. Some of the reasons are given below:

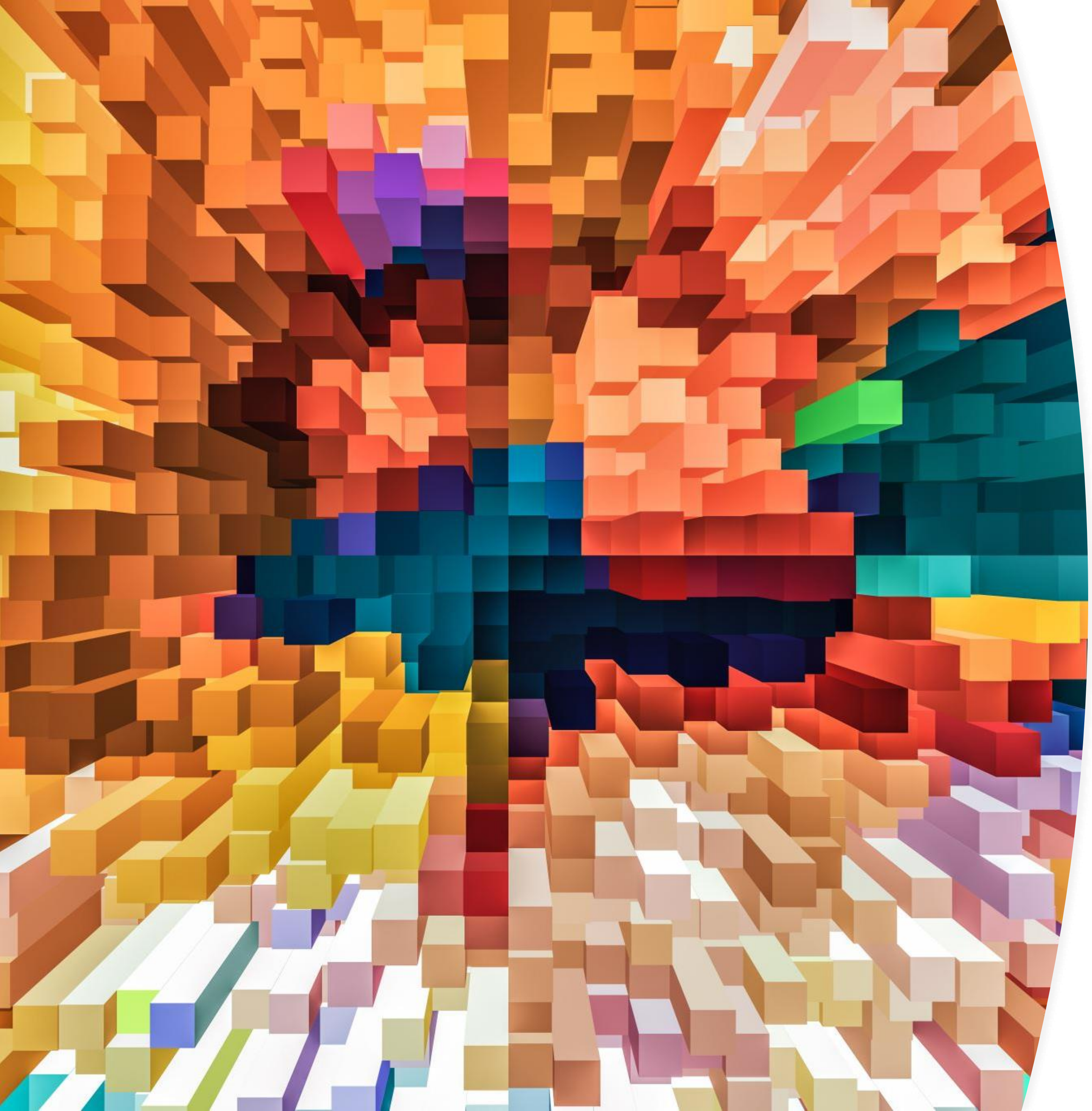
- Given a foreground event has already occurred at that pixel in its previous location, we must strengthen our probability of the current pixel being a foreground event over a background event and vice versa.

Modelling the pixel probabilities

$$\mu_t = M\mu_{t-1} + (1 - M)(\alpha I_t + (1 - \alpha)\mu_{t-1})$$

$$P_B(T_n[\boldsymbol{x}]) = \mathcal{N}(T_n[\boldsymbol{x}] - \mu_n[\boldsymbol{x}], \sigma)$$

$$P_B(T_n[\boldsymbol{x}]) \underset{F}{\overset{B}{\gtrless}} \eta$$



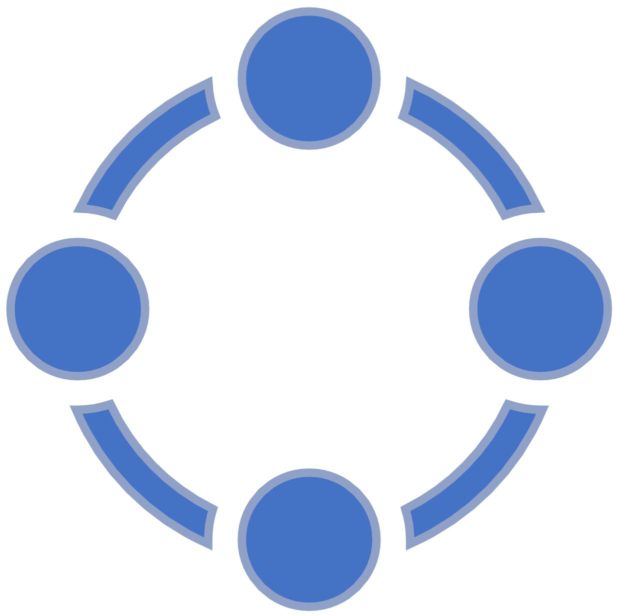
Non - Independency of Pixel

- We have assumed that the pixels we work with are independent; but it is very evident that foreground processes in a neighbouring pixel can affect the current pixel we are working with.
- Hence, we propose a markov random field in order to ensure spatial coherence.

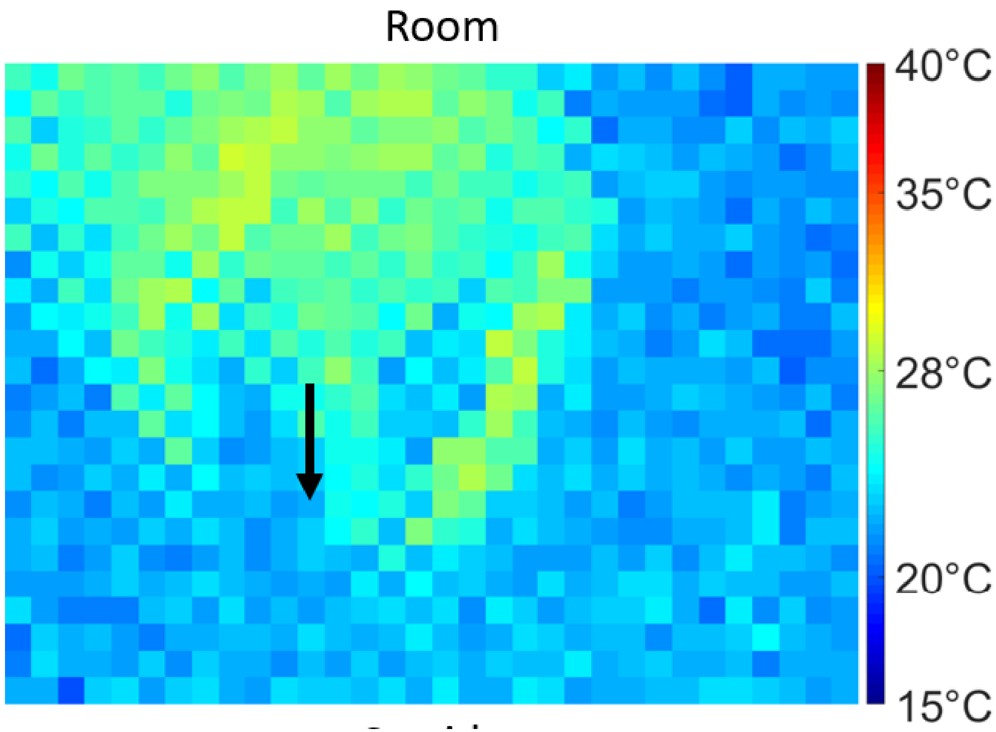
Markov Random Fields (MRF)

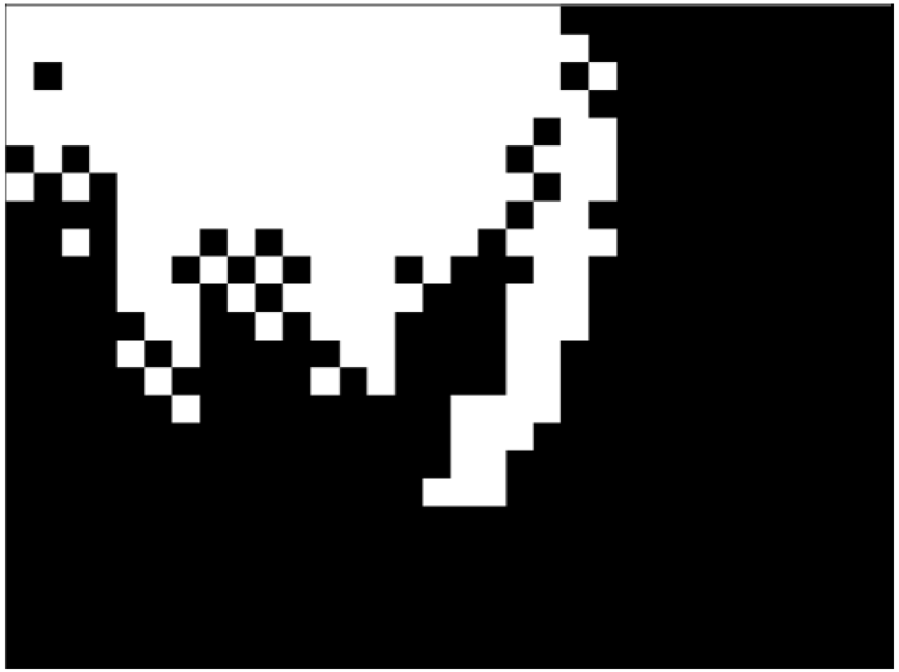
$$\frac{P_B(T_n[\boldsymbol{x}])}{P_F(T_n[\boldsymbol{x}])} \underset{F}{\overset{B}{\gtrless}} \theta exp\left(\frac{Q_F[\boldsymbol{x}] - Q_B[\boldsymbol{x}]}{\gamma}\right)$$

Foreground Probability has uniform distribution as foreground properties can vary with body type, clothing, hairstyle, etcetera.



Results





Event Detection : Single Person Algorithm

- For a single person (baseline algorithm), we determine the frame which has atleast K foreground pixels.

- We select neighbouring frames and analyse them further during event classification.

Event Classification : Single Person Algorithm

- For every foreground pixel at time n, the centroid is calculated in the following way:

$$C_n = \frac{1}{|F_n|} \sum_{\boldsymbol{x} \in F_n} \boldsymbol{x}.$$

Event Classification : Single Person Algorithm

- The centroid can be divided into vertical and horizontal vectors. Assuming that the vertical component makes the difference (with respect to the door-corridor orientation), we classify an event as "entering" if the centroid in the subsequent frames (n+1 , n+2 , ...) moves towards the door and vice versa.

Event Detection : Multi Person Algorithm

- In order to determine multiple number of people passing through a door, instead of determining a single cluster, we determine blobs.

- Each of these blobs consist of atleast L foreground points. For each of such blobs, we calculate centroids in the following manner:

$$\boldsymbol{C}_n = \frac{1}{|F_n|} \sum_{\boldsymbol{x} \in F_n} \boldsymbol{x}.$$

Event Classification : Multi Person Algorithm

- If the number of blobs between n and n+1 are the same, a one-to-one mapping is done between these blobs by mapping the nearest (Eucledian Distance) centroid in frame n to frame n+1.
- If the number of blobs increases, a new blob is born and the mapping depends on the process mentioned above.
- If the number of blobs reduces (when compared to previous frame), blob track is terminated.

Event Classification : Multi Person Algorithm

- The classification is done for each of the individual blobs. For a blob x, the centroid for that blob is divided into horizontal and vertical components.

- As in the previous case, we classify an event as "entering" if the centroid in the subsequent frames (n+1 , n+2 , ...) moves towards the door and vice versa.

Counting

- The number of people entering and exiting the room can be calculated using the algorithm presented in the previous slides.

Metrics

Challenge 1

- Lingering in doorway; wearing a hoodie or carrying a backpack; two people standing in a door and handshaking; multiple people passing through at the same time

Challenge 2

- Wearing a hoodie or thick coat; carrying a backpack; pushing a chair through doorway; leaning against a closed door; one person standing in a door while another one is passing through; multiple people passing through at the same time.

Mean Absolute Error

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

MAE = mean absolute error

y_i = prediction

x_i = true value

n = total number of data points

Closer it is to zero,
better our algorithm

Mean Absolute Error

- The mean absolute error is not a good way of examining the correctness of the algorithm as mean absolute error is scaled by the number of people entering and exiting the room and ignoring the initial count.

Challenge	MAE
1 (Baseline)	0.973
1 (Multi person Algorithm)	0.052
2 (Baseline)	1.431
2 (Multi person Algorithm0	0.945

Mean Squared Error (Per-person)

$$MAE_{PP} = \frac{\sum_{n=1}^{N} |\hat{y}_n - y_n|}{\sum_{n=1}^{N} y_n},$$

Closer it is to zero, better our algorithm

Mean Squared Error (Per-person)

Challenge	MAE
1 (Baseline)	0.137
1 (Multi person Algorithm)	0.007
2 (Baseline)	0.239
2 (Multi person Algorithm)	0.158

Windowed Count-Change

$$e_w(n) = \min_{-w \le \delta \le w} \left|(y_{n+1} - y_n) - (\hat{y}_{n+1+\delta} - \hat{y}_{n+\delta})\right|$$

Correct Classification Rate

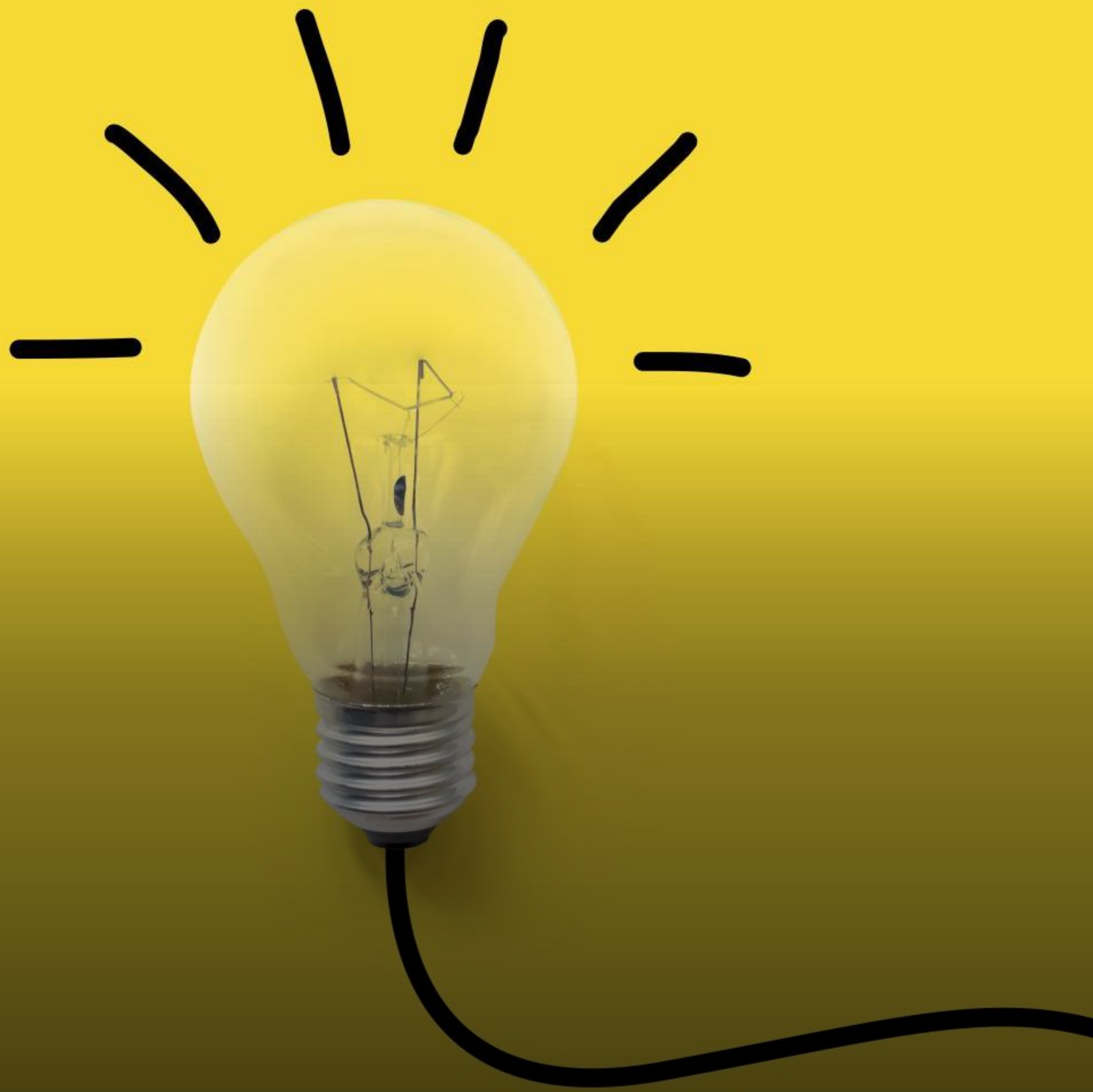
$$CCR_{WCC} = \frac{\sum_{n=1}^{N-1} \mathbf{1}(y_{n+1} \neq y_n) \cdot \mathbf{1}(e_w(n) = 0)}{\sum_{n=1}^{N-1} \mathbf{1}(y_{n+1} \neq y_n)}$$

Closer it is to 1, better is our algorithm

Correct Classification Rate

Challenge	MAE
1 (Baseline)	0.826
1 (Multi person Algorithm)	0.905
2 (Baseline)	0.651
2 (Multi person Algorithm)	0.753

Conclusions



Conclusion

- The results suggests that 80-90% of events are correctly classified and that the multi person algorithm along with a Markov Random field works very efficiently in handling edge cases and high activity areas.