Bokeh Effect for Videos

- Rupesh	160050042
- Deep	160100024
- Vinod	160050016

The Process

1

Extracting the Frames

Extracting individual frames from a video file

2

Foreground Extraction

Automated calculation using Grab-cut Algorithm

3

Temporal Smoothing

Ensuring the foreground mask changes gradually with time

4

Emulating Bokeh

Simulating a Lens Blur

Foreground Extraction

For Foreground Extraction, frame-by-frame, **Grab Cut algorithm** of openCV python library is used.

- Initially, we select everything but a bounding rectangle in the frame as background
- This is the initial estimate given to the algorithm.
- These input pixels are hard-classified as background and remaining pixels have to be classified.
- The iterative algorithm proceeds to find an instantiation of the mask.
  - The algorithm’s goal is to find a min-cut in a graph.
  - The spatial distance and the intensity difference between 2 pixels are taken as the cost of the edge between them

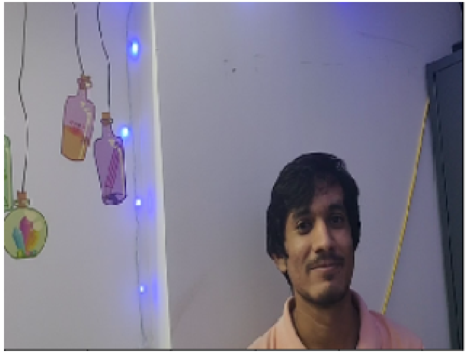
Temporal Smoothing

Initial mask detection by Grab-cut Algorithm is done separately for each frame. So, the following step is performed to ensure temporal consistency in the mask.

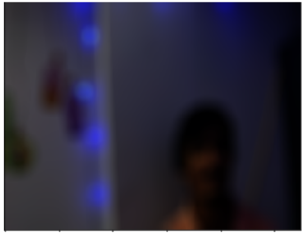
- For a 30fps video, we represent the mask as a 3d (num_frame * 640 * 480) binary matrix
- To ensure temporal as well as spatial smoothness in the mask:
  - Convolve the mask matrix with a 3d (10*3*3) all ones filter
  - This will count for every pixel, the number of neighbours detected as foreground.
  - For every pixel, if the output is >=60, the pixel is ‘1’ else ‘0’
- Thus, out of 90 possible pixels in volume, only if >= 60 neighbours are in the mask originally, then the pixel is considered to be in the foreground of the video.
- The mask thus obtained, is much more stable across time, compared to the older mask

Emulating the Bokeh Effect

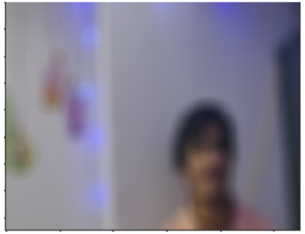
- Image 1
  - Perform Gamma-correction (image^3)
  - Disc-Blur the resultant image
  - Inverse the Gamma correction (image^(⅓))
- Image 2
  - Directly disc blur the Image
- Bokeh Image = Pixel-wise max of (Image 1, Image 2)
- Final Image = Foreground + (Bokeh Image * Background Mask)



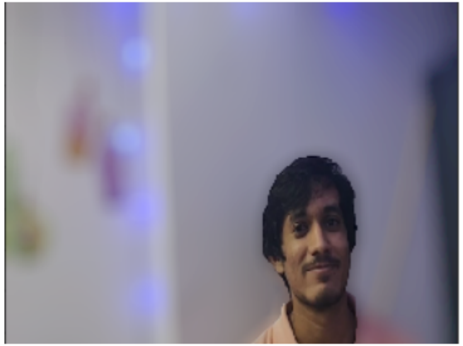
Original Image



Gamma Correction + Disc Blur

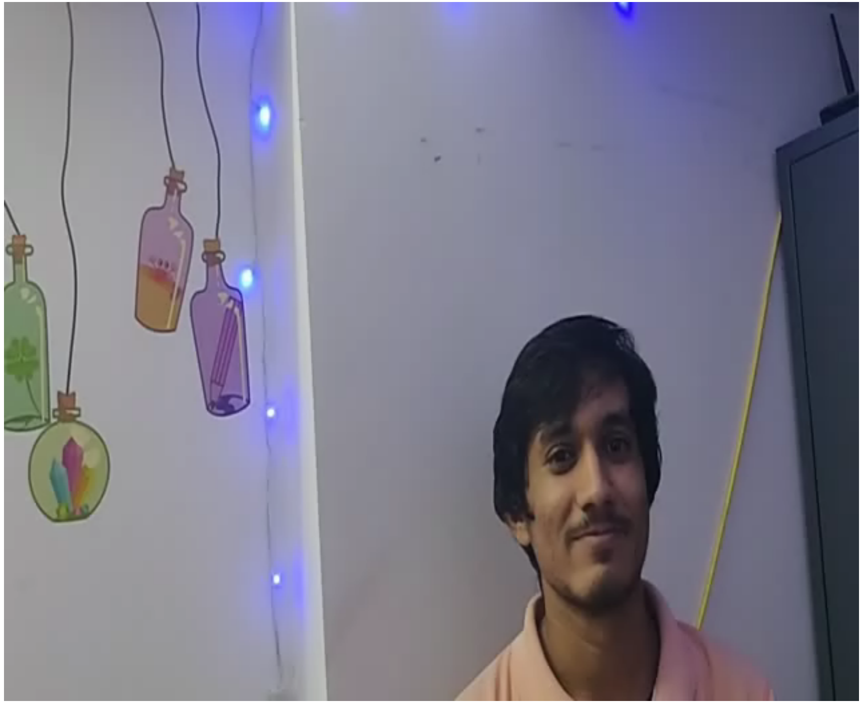


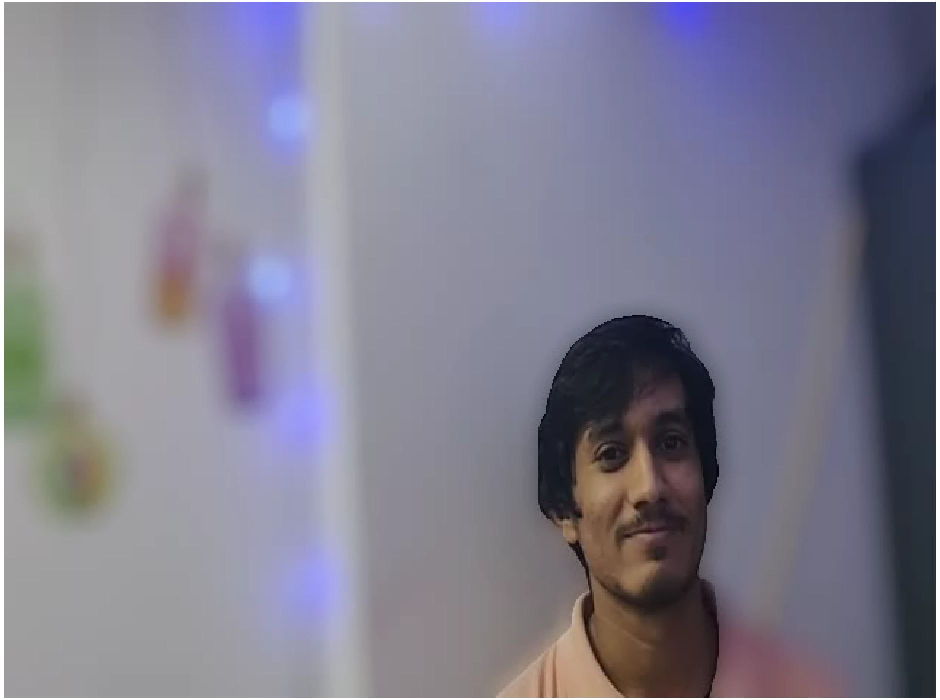
Disc Blur



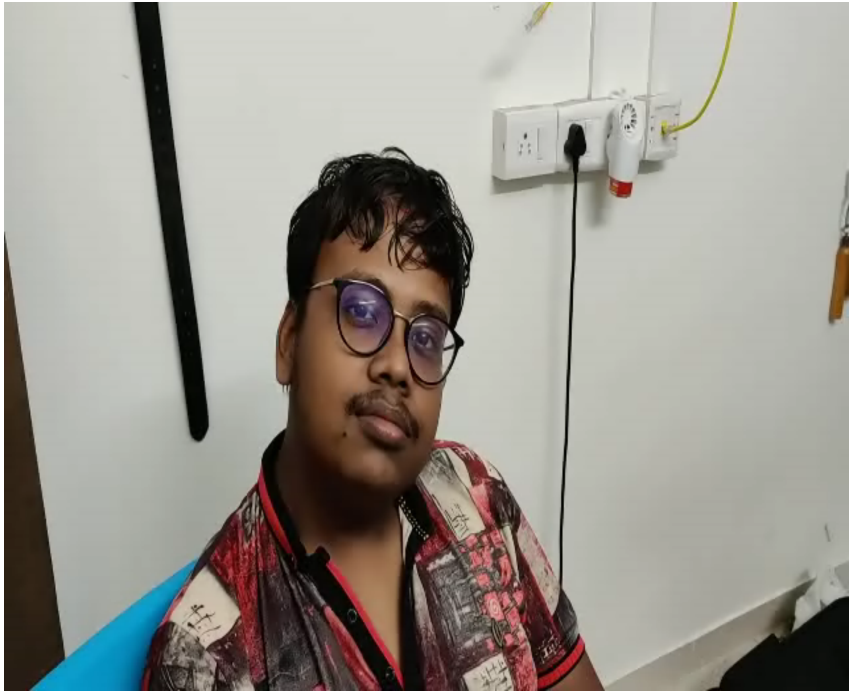
Final Image

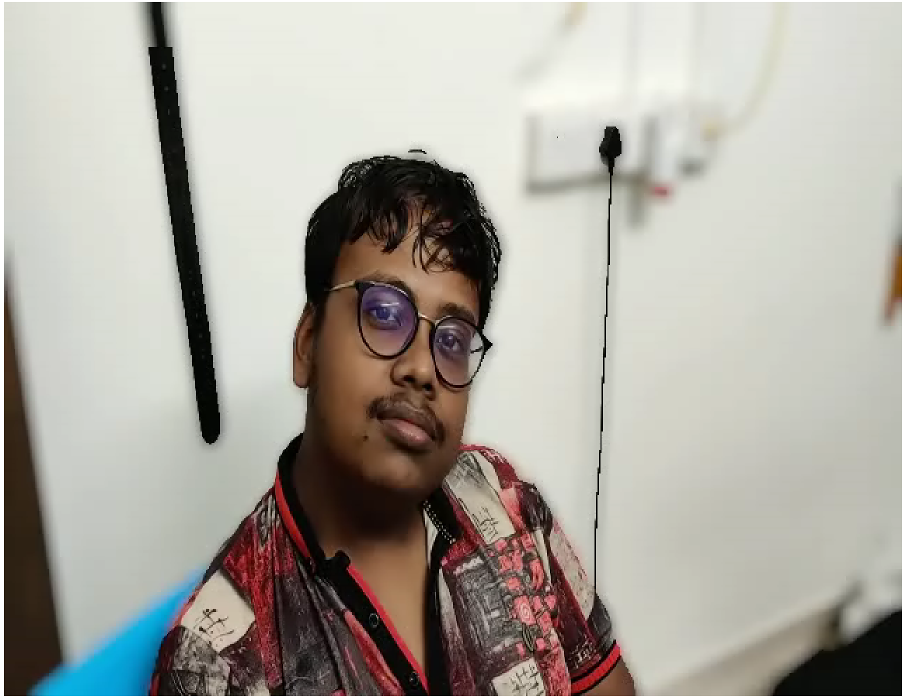
Results (Best)





Some hindrances due to background





Parameters

Rectangle input [GrabCut]	(320,180,200,300)
Iteration Count [GrabCut]	5
Temporal Smoothing Filter	numpy.ones([10,3,3])
Radius of blur kernel (K)	32
Gamma (for gamma correction)	3.0

Conclusion

Pros:

- The algorithm for bokeh emulation is **simple, fast and efficient**
  - <1min for 5s video of 480p@30fps
- **High temporal smoothness** is observed even though masks vary greatly across frames

Cons:

- The algorithm relies greatly on the quality of the foreground mask
- The mask is calculated separately for each frame

Thank You