COL780 Computer Vision
Assignment-1

Somanshu Singla 2018EE10314
Lakshya Kumar Tangri 2018EE10222

September 27, 2021

Background Subtraction

Contents

1 Overview of Models Used

We have tried two models available in OpenCV for our tasks:

1. *Mixture of Gaussians*: It is a technique where every pixel of the image is modelled by a mixture of gaussians and we try to update the parameters of these gaussians as the model sees more and more of input data [Opeb]

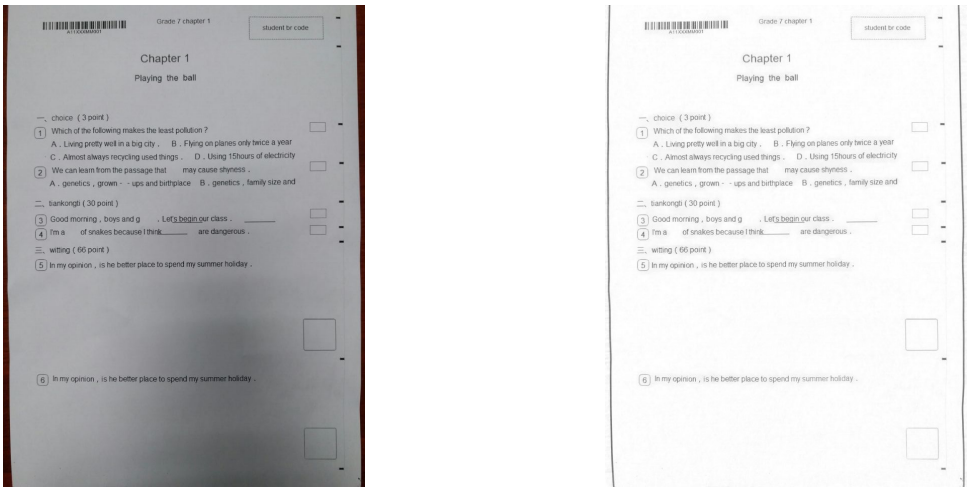
2. *k Nearest Neighbours*: It is a non-parametric method used for probability density approximation, it is based on the idea where we predict the outcome of a pixel based on it's k nearest neighbouring pixels. [Opea]

2 Overview of Image Processing Techniques Used

We have used multiple pre processing and post processing to improve the generated masks:

1. *Sharpening*: We have used sharpening as a post processing techniques as sharp edges are very important for a good foreground mask and also as a means to remove jitter noise.

2. *Shadow Removal*: We have used the combination of dilation, median filtering and image subtraction to handle the changing illumination in input images. [Maš]



(a) Input (b) Output

Figure 1: Example of shadow removal [Maš]

3. *Normalization*: We have used normalization on images so as to make use of wider range of intensity values in case of very dark or bright images. [Háj]

4. *Morphological Transformation*: We have used morphology techniques [Opening and Closing] to remove the noise in the obtained foreground masks.[Tut]

Figure 2: Morphological Transformations: Opening and Closing [Tut]

5. *Filtering*: We have used filtering (Gaussian and Bilateral) to reduce the noise in input images.

6. *Image Pyramid*: We have used the concept of Image Pyramid to take our image to a coarser scale so as to ignore noise such as sea waves which are in larger quantity in finer scales. [Gee]

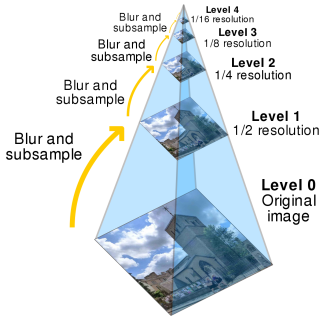


Figure 3: Image Pyramid [Wik]

3 Baseline

3.1 Mask Generation Method

The steps followed in order are:

- *Bilateral Filtering*: We have used bilateral filtering for noise removal, we choose this over guassian filtering because in gaussian filtering edges also gets smoothened out but for foreground detection edges play an important role.
- *kNN model* is used as the base model because it gave better results on testing, after this step the foreground masks are generated.
- *Morphological Transformations*: On the generated foreground masks we used two morphological transformations, the first one was opening which helps to remove noise in the mask which is then followed by closing which helps to fill black holes in the detected foreground. These two operations helped to improve the score by a lot.
- *Sharpening*: We used this as post processing after morphological transformation steps as it sharpens the edges and we know that sharp edges are important in a background subtraction problem.

3.2 Results

- mIOU: 0.7160
- Baseline Mask link

4 Illumination

4.1 Mask Generation Method

The steps followed in order are:

- *Bilateral Filtering*: We have used bilateral filtering for noise removal, we choose this over guassian filtering because in gaussian filtering edges also gets smoothened out but for foreground detection edges play an important role.
- *Normalization*: We use this to increase the contrast of the image that helps in better segmentation of image by making use of wider range of intensity values in case of very dark or bright images
- *Shadow Removal*: In order to remove effect of illumination effects on images, after normalization which makes contrast clearer, we employ dilation in order to get rid of the smaller foreground objects, median filtering which removes the residual noise and gets us a fairly good background image that contains all the shadows followed by image subtraction to finally remove shadows.
- *Mixture of Gaussians model*: is used as the base model because it gave better results on testing, after this step the foreground masks are generated.
- *Morphological Transformations*: On the generated foreground masks we used two morphological transformations, the first one was opening which helps to remove noise in the mask which is then followed by closing which helps to fill black holes in the detected foreground. These two operations helped to improve the score by a lot.

- *Sharpening*: We used this as post processing after morphological transformation steps as it sharpens the edges and we know that sharp edges are important in a background subtraction problem.

4.2 Results

- mIOU: 0.5227
- Illumination Mask link

5 Jitter

5.1 Mask Generation Method

The steps followed in order are:

- *Sharpening*: For the case of camera jitter we observed that the due the shaking of camera the output masks contain a lot of scurious edges, so bilateral filtering was not an option and due to this reason sharpening of actual foreground edges seemed to be a good option, this was also what we found while experimenting
- *kNN model* is used as the base model because it gave better results on testing, after this step the foreground masks are generated.
- *Morphological Transformations*: On the generated foreground masks we used two morphological transformations, the first one was opening which helps to remove noise in the mask which is then followed by closing which helps to fill black holes in the detected foreground. These two operations helped to improve the score by a lot.
- *Gaussian Blur*: We used gaussian blur as a post processing technique because as mentioned above we observed lot of scurious edges and for their removal gaussian filter was a very good choice
- *Sharpening*: We used this as post processing after morphological transformation steps as it sharpens the edges and we know that sharp edges are important in a background subtraction problem.

5.2 Results

- mIOU: 0.6872
- Jitter Mask link

6 Dynamic Scenes

6.1 Mask Generation Method

- *Image Pyramid* : Since we want to remove the effect of moving background like rustling of leaves or moving sea waves, we can look at all images at courser scale/lower resolution so that the motion in background becomes negligible with respect to motion of foregroound objects. So we decreased the size to half using gaussian pyramid.
- *kNN model* is used as the base model because it gave better results on testing, after this step the foreground masks are generated
- *Morphological Transformations*: On the generated foreground masks we used two morphological transformations, the first one was opening which helps to remove noise in the mask which is then followed by closing which helps to fill black holes in the detected foreground. These two operations helped to improve the score by a lot.
- *Sharpening*: We used this as post processing after morphological transformation steps as it sharpens the edges and we know that sharp edges are important in a background subtraction problem.

6.2 Results

- mIOU: 0.5049
- Dynamic Scenes Mask link

7 PTZ

7.1 Mask Generation Method

We weren't able to make much progress on the PTZ case so we just went forward with our pipeline of baseline case. The steps followed in order are:

- *Bilateral Filtering*: We have used bilateral filtering for noise removal, we choose this over guassian filtering because in gaussian filtering edges also gets smoothened out but for foreground detection edges play an important role.
- *Mixture of Gaussians model* is used as the base model because it gave better results on testing, after this step the foreground masks are generated.
- *Morphological Transformations*: On the generated foreground masks we used two morphological transformations, the first one was opening which helps to remove noise in the mask which is then followed by closing which helps to fill black holes in the detected foreground. These two operations helped to improve the score by a lot.
- *Sharpening*: We used this as post processing after morphological transformation steps as it sharpens the edges and we know that sharp edges are important in a background subtraction problem.

7.2 Results

- mIOU: 0.0719
- PTZ Mask link

References

[Gee] Geeks For Geeks. *Image Pyramid using OpenCV — Python*. Available at https://www.geeksforgeeks.org/image-pyramid-using-opencv-python/.

[Háj] Břetislav Hájek. *Normalizing images in OpenCV*. Available at https://stackoverflow.com/questions/38025838/normalizing-images-in-opencv.

[Maš] Dan Mašek. *Increase image brightness without overflow*. URL: https://stackoverflow.com/questions/44047819/increase-image-brightness-without-overflow/44054699#44054699.

[Opea] OpenCV. *cv::BackgroundSubtractorKNN Class Reference*. Available at https://docs.opencv.org/3.4.15/db/d88/classcv_1_1BackgroundSubtractorKNN.html.

[Opeb] OpenCV. *cv::BackgroundSubtractorMOG2 Class Reference*. Available at https://docs.opencv.org/3.4.15/d7/d7b/classcv_1_1BackgroundSubtractorMOG2.html.

[Tut] OpenCV- Python Tutorials. *Morphological Transformations*. Available at https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html.

[Wik] Wikipedia. *Pyramid (image processing)*. Available at https://en.wikipedia.org/wiki/Pyramid_(image_processing).