Assignment #5

* Adwaith Venkataraman (Andrew ID: adwaithv)

**Question 1**

**Analysis:**

1. The video used for this question was ‘store1(1).mp4’, as given with the assignment.
2. Using read function, the first 30 frames of the video was retrieved and stored in a matrix.
3. To make an image with 30 thumbnails, in an arrangement of 5\*6, the horzat and vertcat functions were utilized.
4. With horzcat, rows of 6 thumbnails were made. And with vertcat, the 5 rows were made into a column.

**Results:**

|  |
| --- |
|  |
| Thumbnails of 30 frames concatenated into one picture |

**Matlab Code:**

obj=VideoReader('store1(1).mp4');% read a video file

n=obj.NumberOfFrames;

objHeight=obj.Height;

objWidth=obj.Width;

for i=1:30% retrieve 30 frames from the video

img(i).cdata=read(obj,i);

end

imshow(img(1).cdata);

for i=1:6% concatenate a 6\*5 matrix of images using horzcat and vertcat

image1=horzcat(img(1).cdata,img(2).cdata,img(3).cdata,img(4).cdata,img(5).cdata,img(6).cdata);

end

for i=1:6

image2=horzcat(img(7).cdata,img(8).cdata,img(9).cdata,img(10).cdata,img(11).cdata,img(12).cdata);

end

for i=1:6

image3=horzcat(img(13).cdata,img(14).cdata,img(15).cdata,img(16).cdata,img(17).cdata,img(18).cdata);

end

for i=1:6

image4=horzcat(img(19).cdata,img(20).cdata,img(21).cdata,img(22).cdata,img(23).cdata,img(24).cdata);

end

for i=1:6

image5=horzcat(img(25).cdata,img(26).cdata,img(27).cdata,img(28).cdata,img(29).cdata,img(30).cdata);

end

image\_final=vertcat(image1,image2,image3,image4,image5);

imshow(image\_final);

**Question 2**

**Analysis:**

1. To obtain a sequence of images, the given video file was used to retrieve the consecutive frames.
2. Once the frames were retrieved, a set of 4 consecutive images were obtained.
3. To each of those images, a red rectangular box of the dimensions ‘framewidth/4’ \* ‘frameheight/4’ was added.
4. Now these images were now written into a video format as shown in the matlab code.

**Results:**

The sequence of images used is shown below.

|  |  |
| --- | --- |
|  |  |
|  |  |
| Sequence of Images used to form the video | |

**Matlab Code:**

obj=VideoReader('store1(1).mp4');% read a video file

n=obj.NumberOfFrames;

objHeight=obj.Height;

objWidth=obj.Width;

for i=1:30% read a sequence of images

img(i).cdata=read(obj,i);

end

imshow(img(1).cdata)

rectangle('position',[objWidth/4,objHeight/4,objWidth/2,objHeight/2],'LineWidth',2,'EdgeColor','r');% a red rectangle box is added to each of the 30 images

workingDir = pwd;

imageNames = dir(fullfile(workingDir, 'images', '\*.jpg'));

imageNames = {imageNames.name}';

% Set up output video

outputVideo = VideoWriter(fullfile(workingDir,'newvideo'));

outputVideo.FrameRate = 4;

open(outputVideo);

for ii = 1:length(imageNames)

img = imread(fullfile(workingDir,'images',imageNames{ii})); % read the image

writeVideo(outputVideo,img); % Write to video

end

close(outputVideo);

NOTE: The code snippet to write a set of frames into a video was taken from ‘10-Video Basics/Video Reading and Writing Sample Code.pdf’, page 4.

**Question 3**

**Analysis:**

1. The sample video given was used and 30 images were extracted from it and stored in a folder.
2. The images were now arranged and made as sequential frames to form a video file.
3. The frame rate was altered and the differences were observed.

**Result:**

The images used are shown below.

|  |
| --- |
|  |
| Sequence of 30 images |

**Matlab Code:**

workingDir = pwd;

imageNames = dir(fullfile(workingDir, 'images', '\*.jpg'));

imageNames = {imageNames.name}';

% Set up output video

outputVideo = VideoWriter(fullfile(workingDir,'newvideo2'));

outputVideo.FrameRate = 15;

open(outputVideo);

for ii = 1:length(imageNames)

img = imread(fullfile(workingDir,'images',imageNames{ii})); % read the image

writeVideo(outputVideo,img); % Write to video

end

close(outputVideo);

**Question 4**

**Analysis:**

1. The video given ‘store1.mov’ was used to extract the number of frames.
2. The first frame was retrieved and used as background.
3. The background was converted to gray scale and the frame size was computed.
4. Each frame was read and converted to gray scale.
5. On converting to gray scale, the corresponding pixel values were compared with the threshold value.
6. Based on the comparison and difference, the foreground pixel values were altered.
7. These frames were written into a video.
8. Between the Approximate Filter and the Frame Difference filter methods, the alterations in the foreground pixel values vary. This results in the change in precision of the formed objects.
9. This can be seen from the formed video.

**Results:**

The difference between the Approximate Filter and Frame Difference filter methods are evident from the videos formed.

**Matlab Code:**

% Approximate Median filter method

obj = VideoReader('/Users/Adwaith/18-798/asgn5/Q4/store1(1).mov');

no\_frames = get(obj, 'numberOfframes');

obj2 = VideoWriter('/Users/Adwaith/18-798/asgn5/Q4/q4\_video');

obj2.FrameRate = 5;

open(obj2);

threshold = 25;

background = read(obj,1);% The first frame of the movie is read as the background

background\_bw = double(rgb2gray(background));% the background is converted to grayscale

frame\_size = size(background);

objwidth = frame\_size(2);

objheight = frame\_size(1);

foreground = zeros(objheight, objwidth);

for i = 2:no\_frames

frame = read(obj,i);% every frame of the movie is read

frame\_bw = rgb2gray(frame);% each frame is converted to grayscale

frame\_difference = abs(double(frame\_bw) - double(background\_bw));% negative values are prevented by taking the absolute

for j=1:objwidth% checking if frame\_difference > threshold pixel in foreground

for k=1:objheight

if ((frame\_difference(k,j) > threshold))

foreground(k,j) = frame\_bw(k,j);

else

foreground(k,j) = 0;

end

if (frame\_bw(k,j) > background\_bw(k,j))

background\_bw(k,j) = background\_bw(k,j) + 1;

elseif (frame\_bw(k,j) < background\_bw(k,j))

background\_bw(k,j) = background\_bw(k,j) - 1;

end

end

end

writeVideo(obj2,uint8(foreground));% the resultant set of frames are written into a video format

end

title('Contour of extracted objects using Approximate median filter');

imcontour(foreground);

hold on;

close(obj2);

% Frame difference method

obj = VideoReader('/Users/Adwaith/18-798/asgn5/Q4/store1(1).mov');

no\_frames = get(obj, 'numberOfFrames');

obj2 = VideoWriter('/Users/Adwaith/18-798/asgn5/Q4/q4\_video1');

obj2.FrameRate = 5;

open(obj2);

threshold = 10;

background = read(obj,1);% The first frame of the movie is read as the background

background\_bw = rgb2gray(background);% the background is converted to grayscale

frame\_size = size(background);

objwidth = frame\_size(2);

objheight = frame\_size(1);

foreground = zeros(objheight, objwidth);

for i = 2:no\_frames

fr = read(obj,i);% every frame of the movie is read

frame\_bw = rgb2gray(fr);% each frame is converted to grayscale

frame\_diff = abs(double(frame\_bw) - double(background\_bw));% negative values are prevented by taking the absolute

for j=1:objwidth% checking if frame\_difference > threshold pixel in foreground

for k=1:objheight

if ((frame\_diff(k,j) > threshold))

foreground(k,j) = frame\_bw(k,j);

else

foreground(k,j) = 0;

end

end

end

background\_bw = frame\_bw;

writeVideo(obj2,uint8(foreground));% the resultant set of frames are written into a video format

end

close(obj2);

**Question 5**

**Analysis:**

1. From the foreground images, using the imcontour function plots the contour.

**Results:**

The contour for the Approximate Median filter and the Frame Difference filter methods are shown below.

|  |  |
| --- | --- |
|  |  |
| Approximate Median filter Contour | Frame Difference filter Contour |

**Matlab Code:**

% Approximate Median filter method

obj = VideoReader('/Users/Adwaith/18-798/asgn5/Q4/store1(1).mov');

no\_frames = get(obj, 'numberOfframes');

obj2 = VideoWriter('/Users/Adwaith/18-798/asgn5/Q4/q4\_video');

obj2.FrameRate = 5;

open(obj2);

threshold = 25;

background = read(obj,1);% The first frame of the movie is read as the background

background\_bw = double(rgb2gray(background));% the background is converted to grayscale

frame\_size = size(background);

objwidth = frame\_size(2);

objheight = frame\_size(1);

foreground = zeros(objheight, objwidth);

for i = 2:no\_frames

frame = read(obj,i);% every frame of the movie is read

frame\_bw = rgb2gray(frame);% each frame is converted to grayscale

frame\_difference = abs(double(frame\_bw) - double(background\_bw));% negative values are prevented by taking the absolute

for j=1:objwidth% checking if frame\_difference > threshold pixel in foreground

for k=1:objheight

if ((frame\_difference(k,j) > threshold))

foreground(k,j) = frame\_bw(k,j);

else

foreground(k,j) = 0;

end

if (frame\_bw(k,j) > background\_bw(k,j))

background\_bw(k,j) = background\_bw(k,j) + 1;

elseif (frame\_bw(k,j) < background\_bw(k,j))

background\_bw(k,j) = background\_bw(k,j) - 1;

end

end

end

writeVideo(obj2,uint8(foreground));% the resultant set of frames are written into a video format

end

title('Contour of extracted objects using Approximate median filter');

imcontour(foreground);

hold on;

close(obj2);

% Frame difference method

obj = VideoReader('/Users/Adwaith/18-798/asgn5/Q4/store1(1).mov');

no\_frames = get(obj, 'numberOfFrames');

obj2 = VideoWriter('/Users/Adwaith/18-798/asgn5/Q4/q4\_video1');

obj2.FrameRate = 5;

open(obj2);

threshold = 10;

background = read(obj,1);% The first frame of the movie is read as the background

background\_bw = rgb2gray(background);% the background is converted to grayscale

frame\_size = size(background);

objwidth = frame\_size(2);

objheight = frame\_size(1);

foreground = zeros(objheight, objwidth);

for i = 2:no\_frames

fr = read(obj,i);% every frame of the movie is read

frame\_bw = rgb2gray(fr);% each frame is converted to grayscale

frame\_diff = abs(double(frame\_bw) - double(background\_bw));% negative values are prevented by taking the absolute

for j=1:objwidth% checking if frame\_difference > threshold pixel in foreground

for k=1:objheight

if ((frame\_diff(k,j) > threshold))

foreground(k,j) = frame\_bw(k,j);

else

foreground(k,j) = 0;

end

end

end

background\_bw = frame\_bw;

writeVideo(obj2,uint8(foreground));% the resultant set of frames are written into a video format

end

figure, title('Contour of extracted objects using Frame difference filter');

imcontour(foreground);

hold on;

close(obj2);