Lab4: Hu's invariant moments

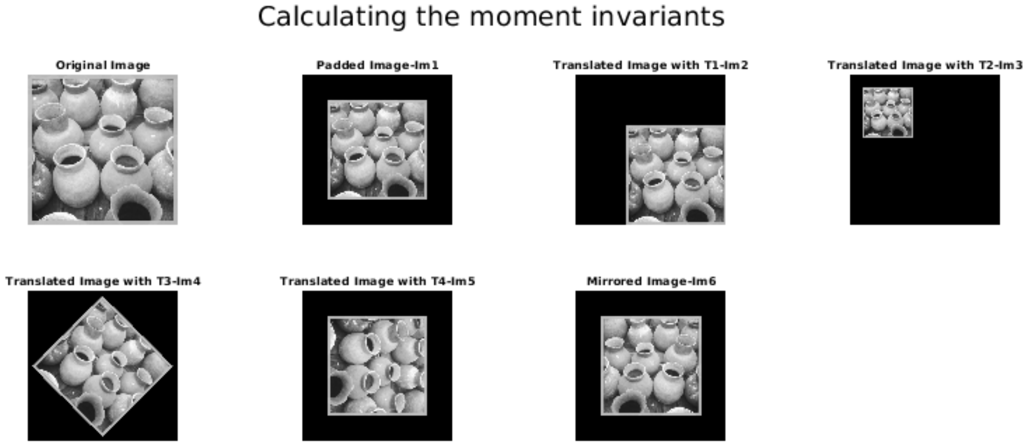
Sourav Sarker
ID: 201892985
souravs@mun.ca

```
I = imread('image.png');

img = imread('image.png');
[X,Y,channel] = size(img);

% padding one fourth of size of image
Im1 = padarray(img,1/4*[X Y],0,'both');

figure('position', [0, 0, 800, 300])
subplot(2,4,1);
imshow(I);
title('Original Image', "FontSize",6);

[rows, columns, numberOfColorChannels] = size(I)

subplot(2,4,2);
imshow(Im1);
title('Padded Image-Im1', "FontSize",6);

% Create spatial transformation matrix T1 for translation as follows
T1 = maketform('affine', [1 0 0; 0 1 0; 90 90 1]);

% Use the above transformation matrix T1 and perform the transformation as follows:
Im2 = imtransform(Im1, T1, ...
'XData',[1 size(Im1,1)], 'YData',[1 size(Im1,2)]);

subplot(2,4,3);
imshow(Im2);
title('Translated Image with T1-Im2', "FontSize",6);

% Create spatial transformation matrix T2 for scaling to 0.5 (of original size) as follows
T2 = maketform('affine', [0.5 0 0; 0 0.5 0; 0 0 1]);
% Use the above transformation matrix T2 and perform the transformation as follows
Im3 = imtransform(Im1, T2, ...
'XData',[1 size(Im1,1)], 'YData',[1 size(Im1,2)]);

subplot(2,4,4);
imshow(Im3);
title('Translated Image with T2-Im3', "FontSize",6);

% Create spatial transformation matrix T3 to rotate the image by 45 degrees as follows
T3 = maketform('affine', [cos(pi/4) sin(pi/4) 0; -sin(pi/4) cos(pi/4) 0; 0 0 1]);
% Use the above transformation matrix T3 and perform the transformation as follows
Im4 = imtransform(Im1, T3, ...
'XData',[-269 size(Im1,1)-270], 'YData',[+111 size(Im1,2)+110]);

subplot(2,4,5);
imshow(Im4);
title('Translated Image with T3-Im4', "FontSize",6);


% Create spatial transformation matrix T4 for rotating the image 90 degrees as follows
T4 = maketform('affine', [cos(pi/2) sin(pi/2) 0; -sin(pi/2) cos(pi/2) 0; 0 0 1]);
% Use the above transformation matrix T4 and perform the transformation as follows
Im5 = imtransform(Im1, T4, ...
'XData',[-539 size(Im1,1)-540], 'YData',[1 size(Im1,2)]);

subplot(2,4,6);
imshow(Im5);
title('Translated Image with T4-Im5', "FontSize",6);

% Flip the original image Im1 from left to right using following command and generate image Im6
Im6=flipdim(Im1,2);

subplot(2,4,7);
imshow(Im6);
title('Mirrored Image-Im6', "FontSize",6);

suptitle('Calculating the moment invariants')

Moment_invariants(Im1)
Moment_invariants(Im2)
Moment_invariants(Im3)
Moment_invariants(Im4)
Moment_invariants(Im5)
Moment_invariants(Im6)
```

Output:

```
rows =

   360


columns =

   360
```

```
numberOfColorChannels =

     1


Moment_Invariants =

   -0.5738   -4.2405   -4.4618   -4.5262   -9.1160    6.6464    9.2440


Moment_Invariants =

   -0.5738   -4.2405   -4.4618   -4.5262   -9.1160    6.6464    9.2440


Moment_Invariants =

   -0.5735   -4.2480   -4.4569   -4.5111   -9.1012    6.6353    9.2014


Moment_Invariants =

   -0.5738   -4.2397   -4.4612   -4.5266   -9.1165    6.6465    9.2440


Moment_Invariants =

   -0.5738   -4.2405   -4.4618   -4.5262   -9.1160    6.6464    9.2440


Moment_Invariants =

   -0.5738   -4.2405   -4.4618   -4.5262   -9.1160    6.6464   -9.2440
```



Published with MATLAB® R2020a

Discussion:

We have measured moment invariants for every image and analyzed the similarities and inconsistencies in terms of image features. Translation, scaling and rotation are invariant features of the moment invariants that aid in recognizing image patterns. As we know, a digital image is discrete, whereas moments are invariant in a continuous function.