Exercise on Geometrical transformations in computer vision

ANNAPOORNIMA S

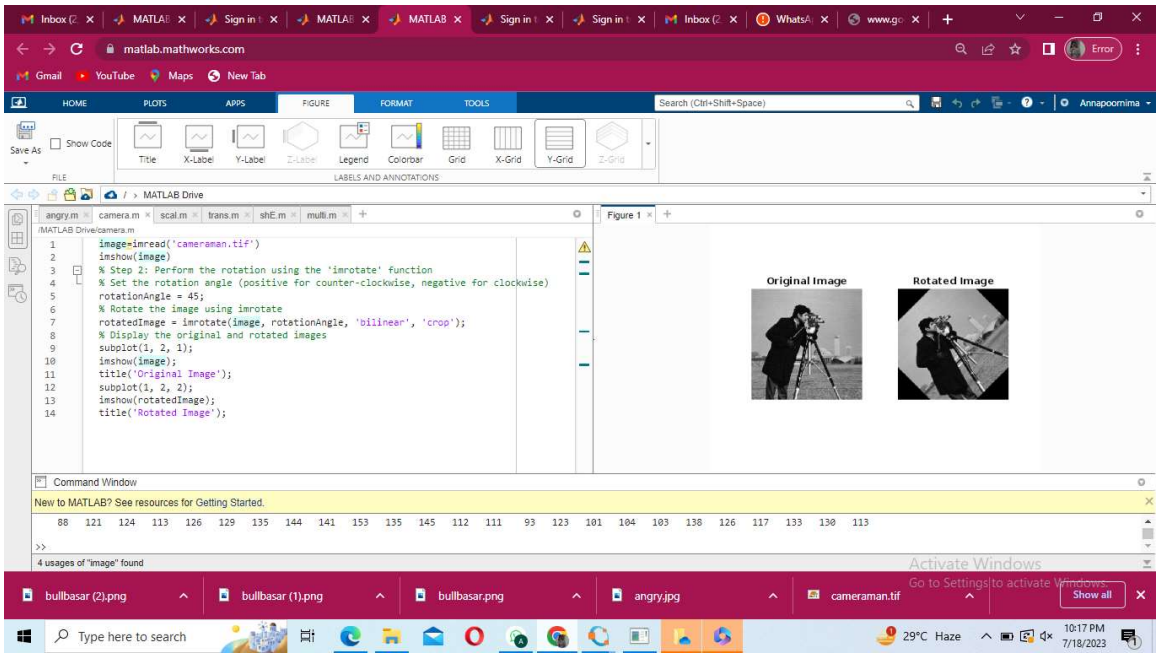
225229101

1.ROTATION

Step 1: Read the original image using the ‘imread’ function Replace 'input_image.jpg' with the actual path and filename of your image.

Step 2: Perform the rotation using the ‘imrotate’ function The imrotate function takes the original image and the rotation angle θ as input and returns the rotated image.

Step 3: Display the original and rotated images side by side Create a figure with two subplots using the subplot function. The first subplot displays the original image using ‘imshow’, and the second subplot displays the rotated image. The title function is used to provide titles for each subplot

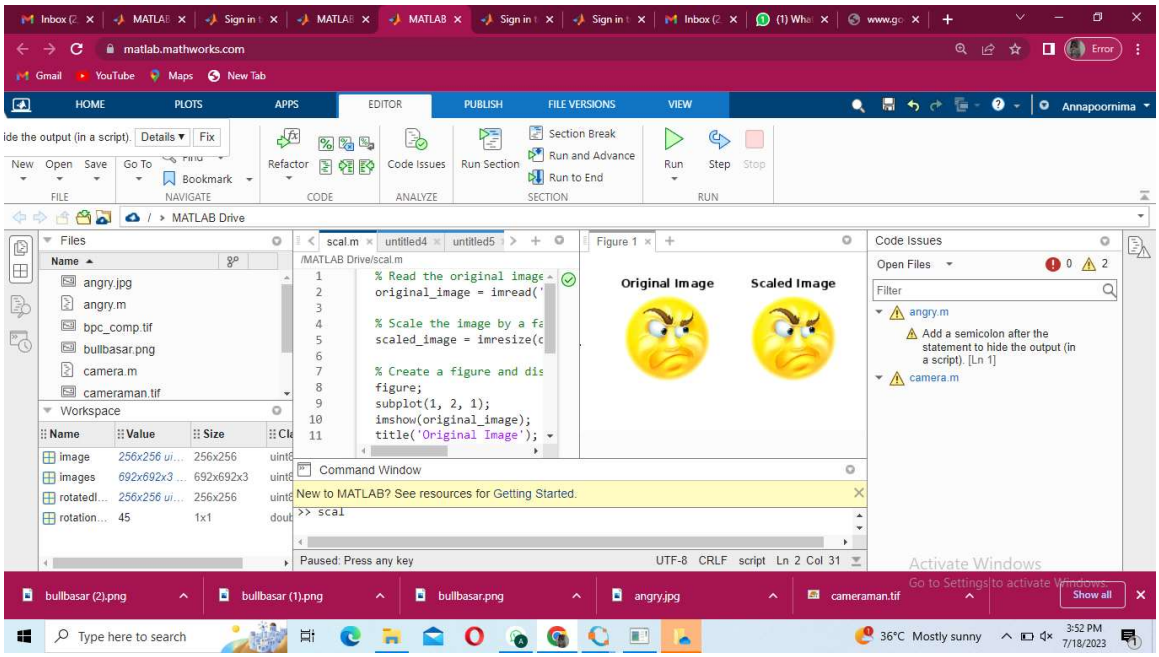


2.SCALING

Step 1: Read the original image using the ‘imread’ function Replace'input_image.jpg' with the actual path and filename of your image.

Step 2: Perform the scaling transformation using the ‘imresize’ function The imresize function takes the original image and the scaling factor as input and returns the scaled image.

Step 3: Display the original and scaled images side by side Create a figure with two subplots using the subplot function. The first subplot displays the original image using ‘imshow’, and the second subplot displays the scaled image. The title function is used to provide titles for each subplot.



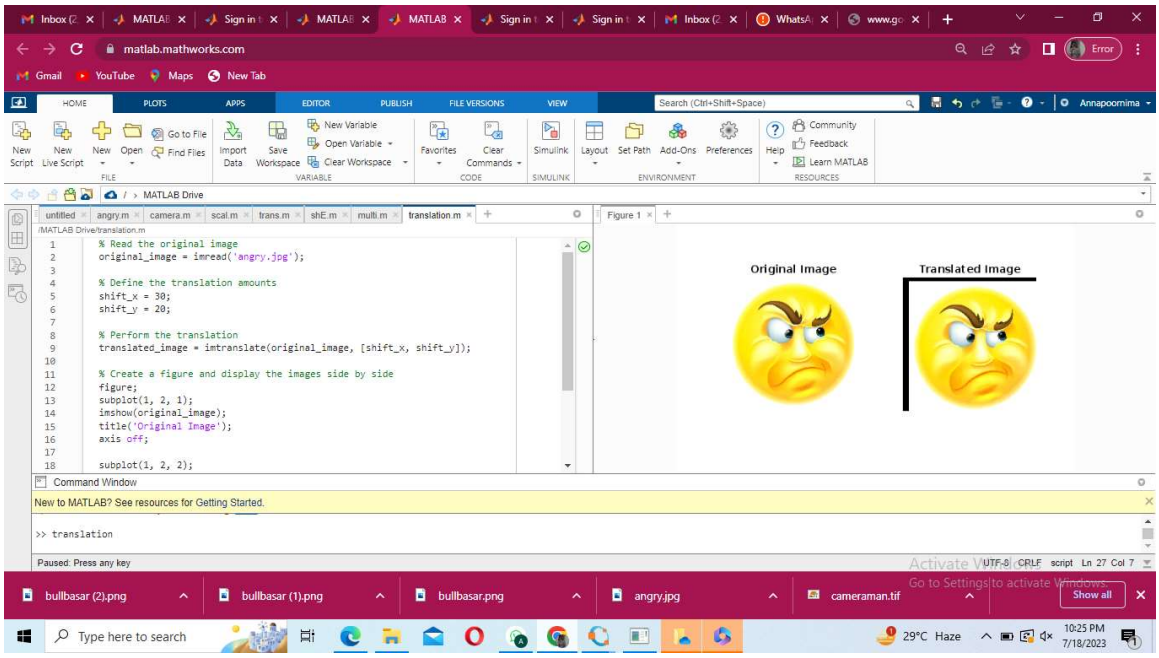
3.TRANSLATION

Step 1: Read the original image using the ‘imread’ function Replace 'input_image.jpg' with the actual path and filename of your image.

Step 2: Define the translation amounts in pixels. translationMatrix = [1 0 tx; 0ty];

Step 3: Create a 2x3 translation matrix using the translation amounts. The translation matrix represents the translation transformation and specifies the shift in the x-axis (tx) and y-axis (ty).

Step 4: Perform the translation using the ‘imwarp’ function The ‘imwarp’ function applies the translation to the original image using the translation matrix. The ‘affine2d’ function creates an affine transformation object based on the translation matrix



4.SHEARING

Step 1: Read the original image using the imread function Replace 'input_image.jpg' with the actual path and filename of your image.

Step 2: Define the shear factor: shearFactor = 0.5; % Shear factor along the xaxis

Step 3: Create a shear transformation matrix: The shear matrix represents the shear transformation and specifies the shear factor along the x-axis (shearFactor).

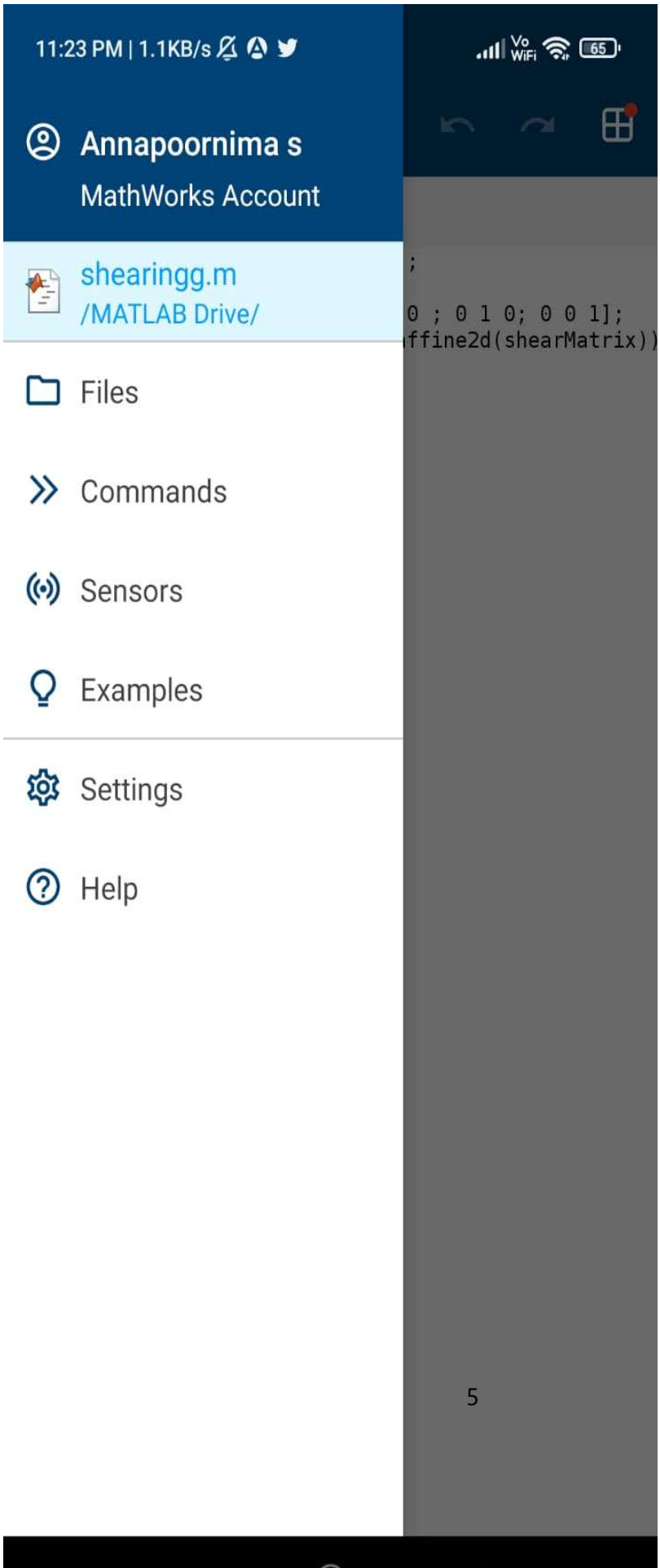
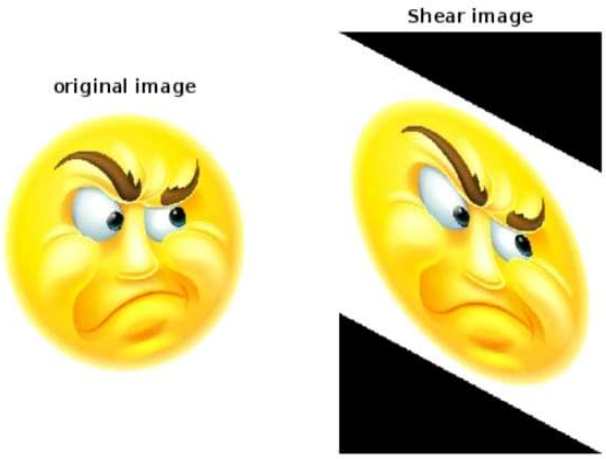

11:23 PM | 1.1KB/s
Annapoornima s
MathWorks Account
shearingg.m
/MATLAB Drive/
Files
Commands
Sensors
Examples
Settings
Help
;
0 ; 0 1 0; 0 0 1];
ffine2d(shearMatrix))

Figure 1: shearingg




11:24 PM | 0.4KB/s
MATLAB Drive
shearingg.m
original = imread('angry.jpg');
shear_factor = 0.5;
shearMatrix = [1 shear_factor 0 ; 0 1 0; 0 0 1];
shearImage = imwarp(original,affine2d(shearMatrix))
figure;
subplot(1,2,1);
imshow(original);
title("original image");
subplot(1,2,2);
imshow(shearImage);
title("Shear image");

5.TRANSFORMED IMAGE

Step1: Read the original image using the imread function Replace 'input_image.jpg' with the actual path and filename of your image.

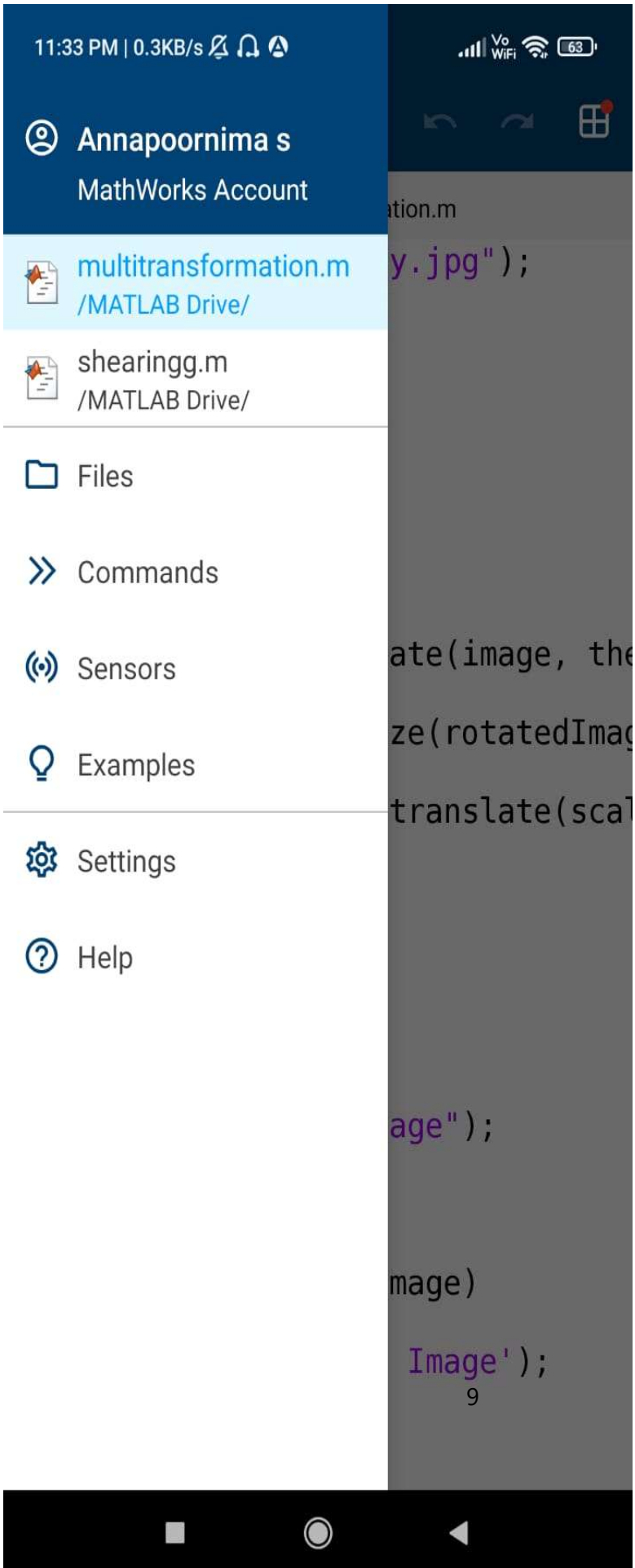
Step 2: Define the transformation parameters

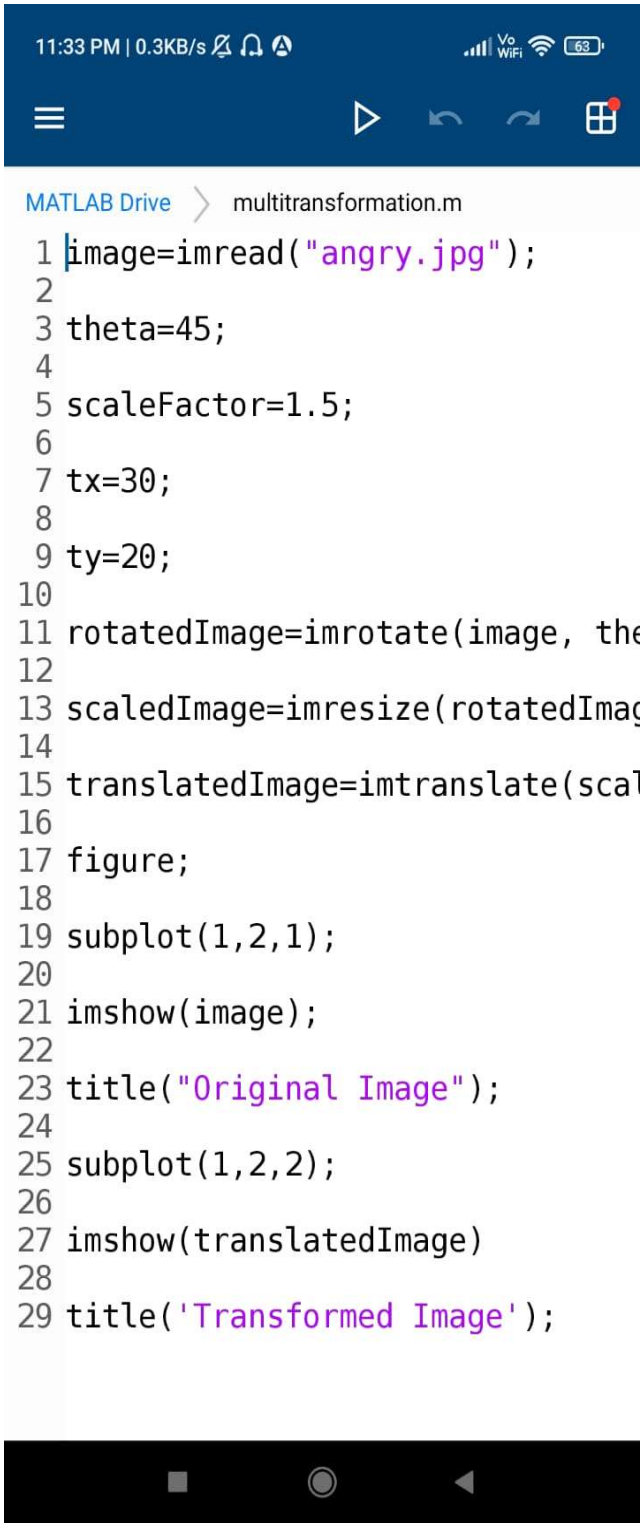
Step 3: Create the transformation matrices The rotation matrix represents the rotation transformation, the scaling matrix represents the scaling transformation, and the translation matrix represents the translation transformation.

Step 4: Combine the transformations combinedMatrix = translationMatrix * scalingMatrix * rotationMatrix;

Step 5: Perform the combined transformation using the imwarp function The 'imwarp' function applies the combined transformation to the original image using the combined matrix. The 'affine2d' function creates an affine transformation object based on the combined matrix.

Step 6: Display the original and transformed images side by side Create a figure with two subplots using the subplot function. The first subplot displays the original image using 'imshow', and the second subplot displays the transformed image. The title function is used to provide titles for each subplot


11:33 PM | 0.3KB/s
Annapoornima s
MathWorks Account
multitransformation.m
/MATLAB Drive/
shearingg.m
/MATLAB Drive/
Files
Commands
Sensors
Examples
Settings
Help
ation.m
y.jpg");
ate(image, the
ze(rotatedImag
translate(scal
age");
mage)
Image');
9


11:33 PM | 0.3KB/s
MATLAB Drive
multitransformation.m
image=imread("angry.jpg");
theta=45;
scaleFactor=1.5;
tx=30;
ty=20;
rotatedImage=imrotate(image, the
scaledImage=imresize(rotatedImag
translatedImage=imtranslate(scal
figure;
subplot(1,2,1);
imshow(image);
title("Original Image");
subplot(1,2,2);
imshow(translatedImage)
title('Transformed Image');

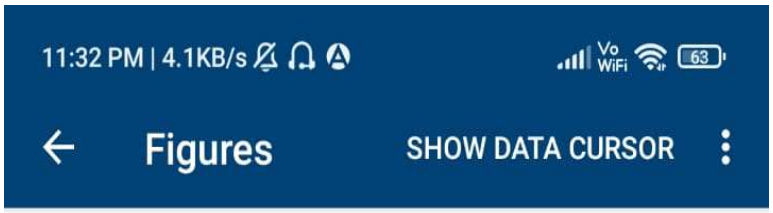


Figure 6: multitransformation