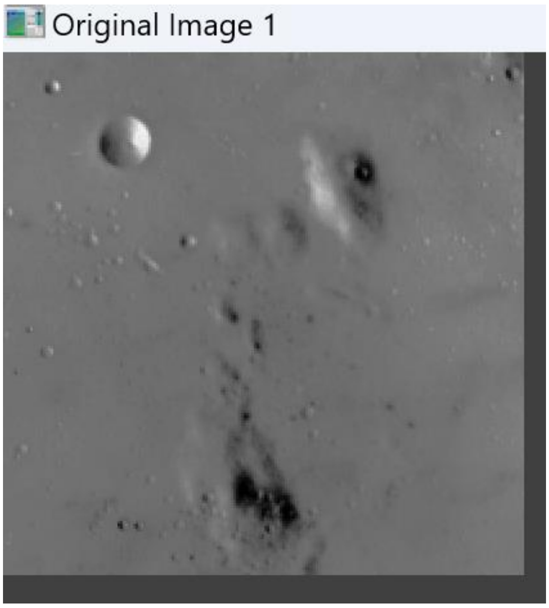
Problem 1 – Point operations (30 points):

a)

Image 1



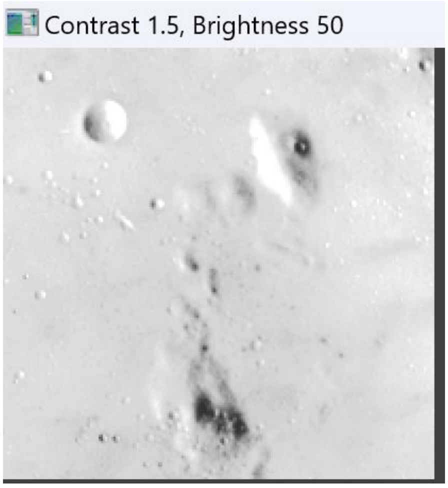
Brightness and Contrast Stretching:

Applying Brightness and Contrast Stretching to Image 1 aimed to improve pixel intensity distribution by adjusting the dynamic range. However, this method did not significantly enhance the overall image quality. The results showed minimal improvement in detail visibility, particularly in the darker regions, and failed to reveal additional information compared to the original image. Despite experimenting with various contrast and brightness values, the enhancements were limited and did not contribute to a noticeable difference.

Experimentation with Parameters:

Below are results obtained with various parameter adjustments:

- **Contrast: 1.5, Brightness: 50**:



- **Contrast: 2, Brightness: 10**:

- **Contrast: 3, Brightness: 0**:

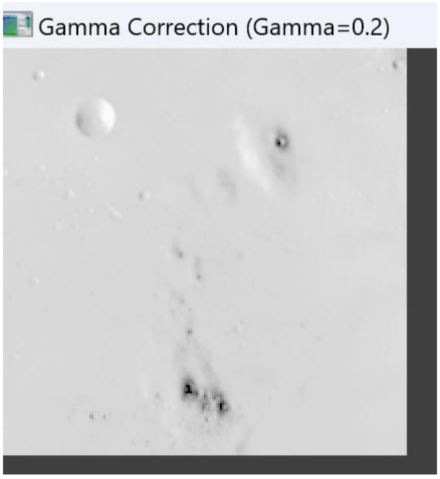
Gamma Correction:

Exploring various gamma correction parameters revealed that the adjustments did not result in noticeable improvements compared to the original image. When gamma values were less than 1, the image appeared brightened, reducing the contrast and obscuring finer details. Conversely, gamma values greater than 1 darkened the image excessively, further diminishing visibility in shadowed regions.

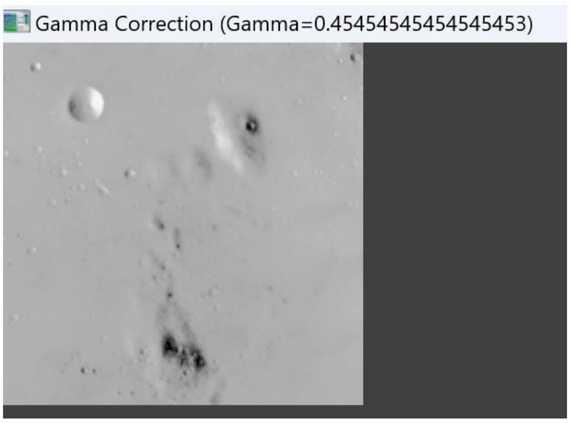
Experimentation with Parameters:

Below are results obtained with different gamma values:

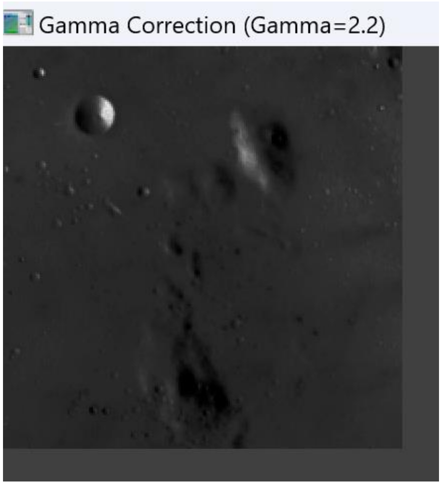
- **Gamma: 0.2**:



- **Gamma: 1/2.2**:



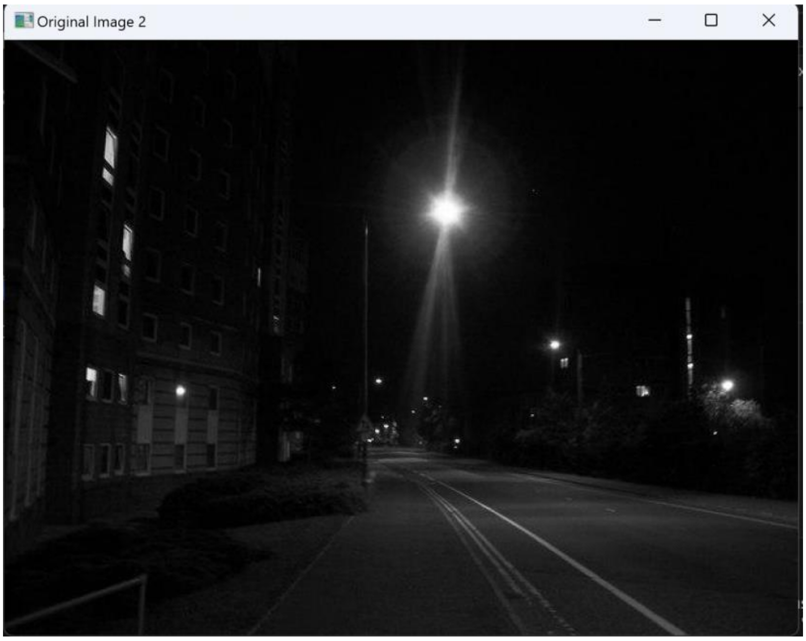
- **Gamma: 2.2**:



Histogram Equalization:

Histogram Equalization yielded the most significant improvement for Image 1. This method effectively enhanced the visibility of hidden details by redistributing pixel intensities. However, it introduced minor artifacts in smoother regions of the image.

Image 2



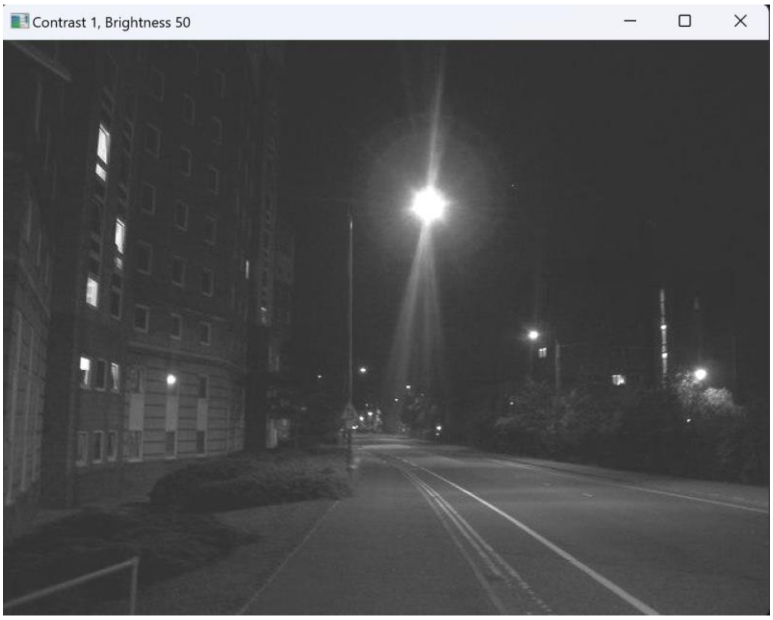
Brightness and Contrast Stretching:

Upon applying Brightness and Contrast Stretching to Image 2, we observed varying results depending on the parameter values chosen. The adjustments helped enhance the brightness of certain areas, such as the streetlights and their reflections. However, the overall contrast improvement was minimal, as the image already displayed significant differences between the lit and unlit regions. Higher brightness values tended to overexpose the image, while lower brightness failed to enhance the shadowed areas effectively.

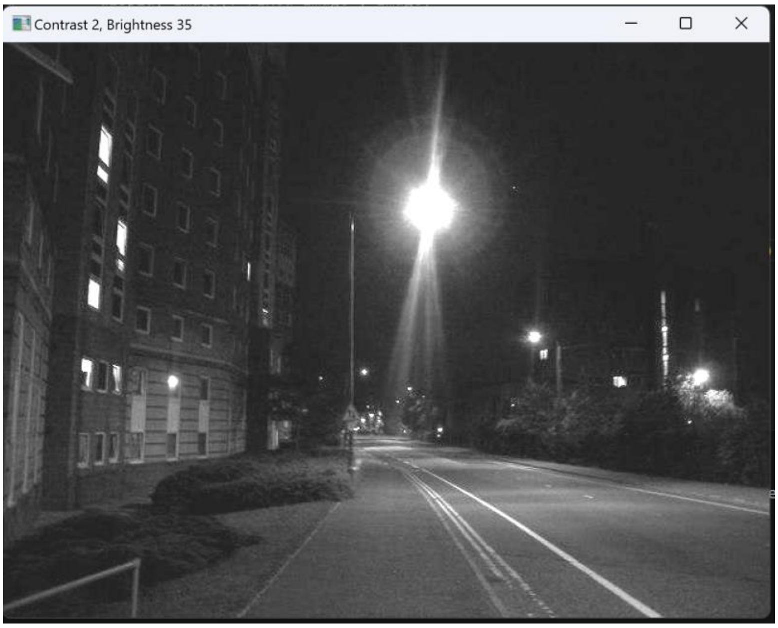
Experimentation with Parameters:

Below are results obtained with various parameter adjustments:

- **Contrast: 1, Brightness: 50**:



- **Contrast: 2, Brightness: 35**:



- **Contrast: 2, Brightness: 100**:

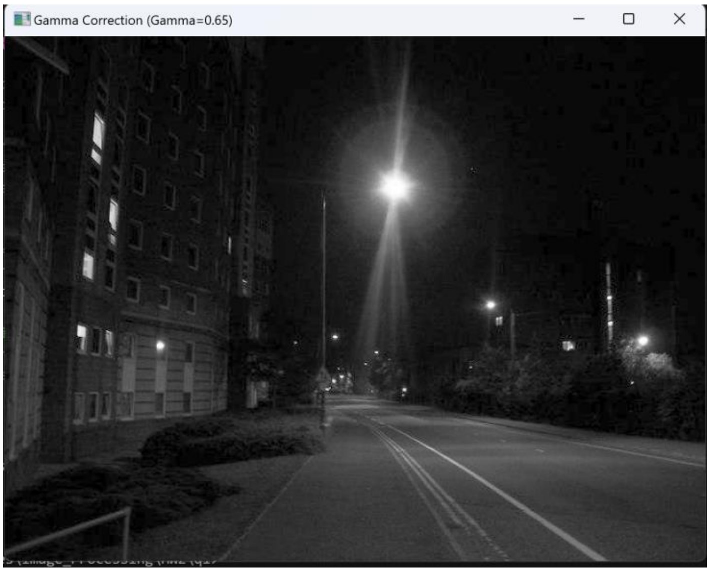
Gamma Correction:

For Gamma Correction, various values were tested to achieve an optimal adjustment. A gamma value of 1/2.2 brightened the darker regions effectively but introduced some noise into the brighter spots. With a gamma value of 0.65, the image maintained a better balance between dark and light areas, making it the most favorable correction in this case.

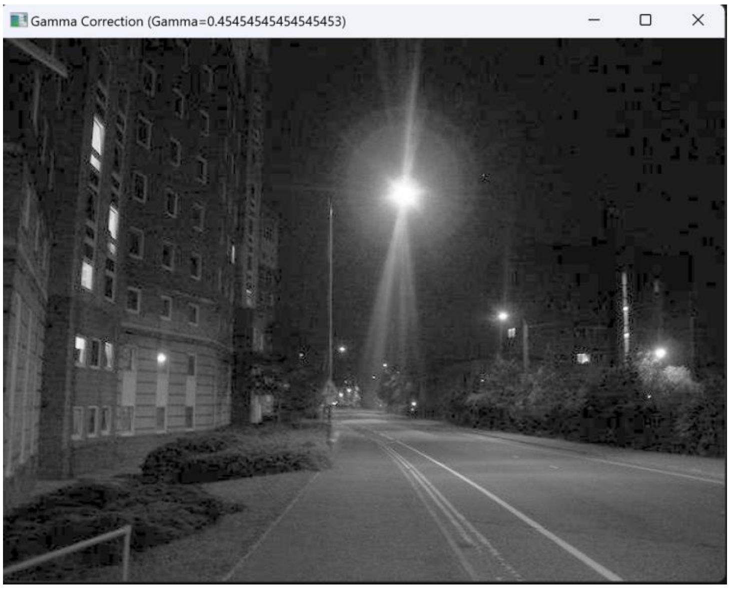
Experimentation with Parameters:

Below are results obtained with various parameter adjustments:

- **Gamma: 0.65**:



- **Gamma: 0.2**:



Histogram Equalization:

When applying Histogram Equalization to Image 2, the method significantly enhanced the contrast by redistributing pixel intensity values across the available range. While this process successfully illuminated darker regions and revealed previously hidden details, it also exaggerated the brightness of already bright areas, such as streetlights, causing overexposure in these regions. The method improved the overall visibility but resulted in a loss of detail in some highly illuminated areas.

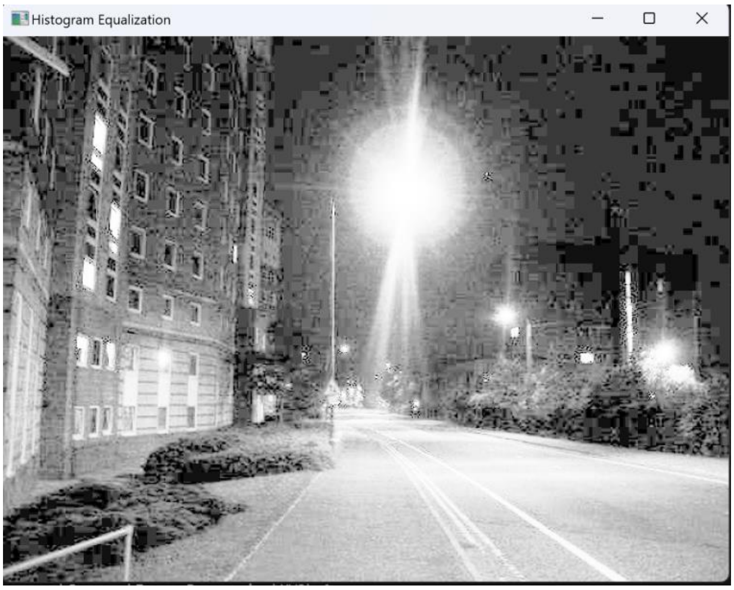


Image 3

Brightness and Contrast Stretching:

Three different adjustments were tested for brightness and contrast: **Contrast: 2, Brightness: 35**, **Contrast: 1, Brightness: 35**, and the calculated **maximum brightness and contrast**.

- For **Contrast: 2, Brightness: 35**, there was no noticeable enhancement observed in the subject (the girl), and significant detail was lost. The sky became severely overexposed, appearing uniformly white, while the subject also suffered from a loss of texture and detail. This setting provided no meaningful improvement, making it the least effective adjustment.
- For **Contrast: 1, Brightness: 35**, the result was slightly more balanced, but there was still a loss of details in both the subject and the sky. While the sky retained a bit more tonal range compared to the previous setting, mild overexposure persisted, and the subject remained unimproved in terms of visible details.
- For the **maximum brightness and contrast** adjustment, the result provided the best overall enhancement. The subject and foreground showed significant improvements in detail and contrast without sacrificing the tonal balance of

the sky. Unlike the previous settings, the sky retained a reasonable amount of texture and detail, preventing it from appearing uniformly bright. This adjustment demonstrated an optimal balance between enhancing the subject and preserving the overall visual harmony of the image.

- **Contrast: 2, Brightness: 35**:

- **Contrast: 1, Brightness: 35**:

- **Maximum Brightness and contrast**

Gamma Correction:

When applying gamma correction to Image 3, a range of gamma values were tested to evaluate their effect on the image's brightness and contrast. For gamma values less than 1, such as 0.3 and 0.65, the image exhibited an increase in brightness, particularly in regions with darker tones. This resulted in improved visibility of shadows and midtones, enhancing details that were previously less discernible.

For gamma values greater than 1, the image became darker, compressing the brightness levels and losing some fine details in the highlights.

Based on the observations, **Gamma = 0.65** provides the most visually balanced and effective correction for this image.

<u>Experimentation with Parameters:</u>

Below are results obtained with various parameter adjustments:

- **Gamma: 0.65**

- **Gamma: 0.3**

- **Gamma: 2.2**

Histogram Equalization:

Applying Histogram Equalization to Image 3 improved contrast in darker areas but caused overexposure in bright regions like the sky, leading to a loss of detail. This method is less suitable for images with high brightness variations.

b)

Image 1

Image 2

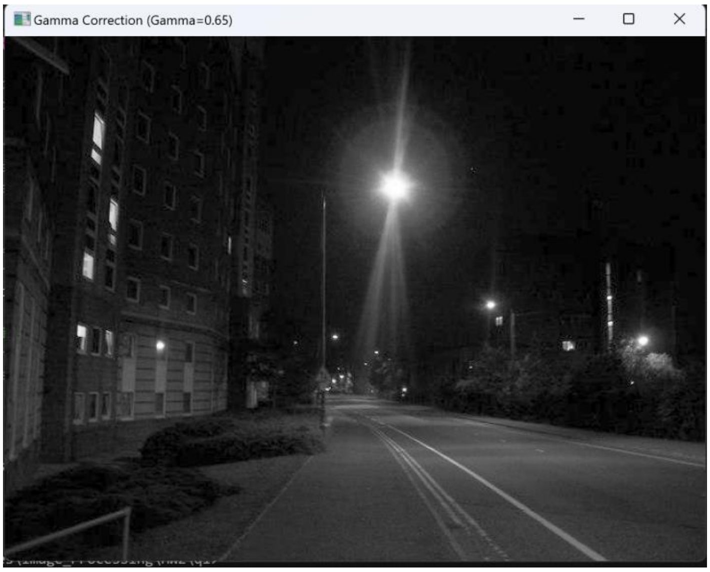


Image 3

Problem 2 – Puzzle solving with geometric operations (70 points):

a) In this step, we use Paint to manually identify corresponding landmarks between the first image, and each of the other puzzle pieces. For the first two puzzles, we identify 3 matching points per piece, while for the last puzzle, we identify 4 matching points per piece.

b) To ensure that the prepare_puzzle function processes the puzzle directory correctly and returns the expected data, we called the function with the first puzzle (puzzle_affine_1) and printed its returned values. The output was verified to match the input data provided in the matches.txt file and the expected format for affine transformations.

Below is the output captured while debugging the function for the first puzzle:

```
Starting puzzle_affine_1
Matches loaded successfully!
Is affine: True
Number of images: 2
Matches shape: (1, 3, 2, 2)
Matches data:
[[[[255 286]
   [104 103]]

  [[443 347]
   [ 45 243]]

  [[252 146]
   [190 112]]]]
```

c) we implemented the get_transform function, which computes the transformation matrix for aligning a puzzle piece to the reference image (first image). The function uses the flag is_affine to determine whether the transformation is **affine** or **projective**. we we use OpenCV functions to compute the transformation matrix:

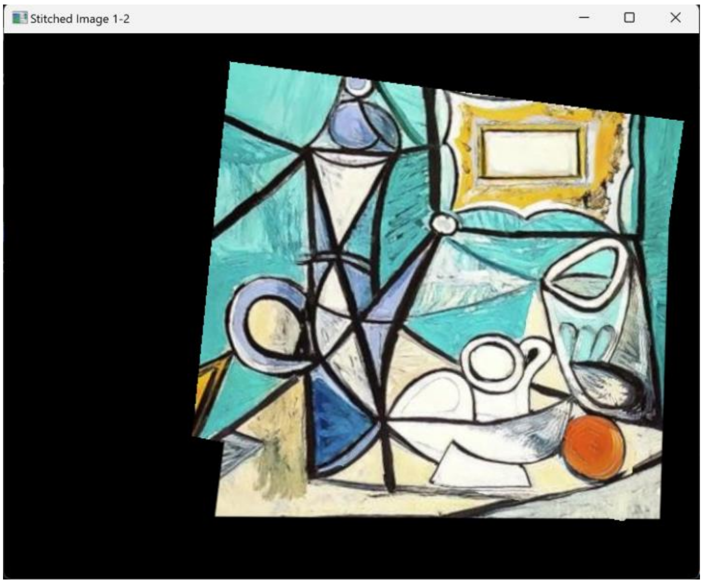
- Affine Transformations: **cv2.getAffineTransform** for 3 points.
- Projective Transformations: **cv2. getPerspectiveTransform** for 4 points.

d) We implement the function inverse_transform_target_image, which applies the inverse transformation to align a target image onto the reference canvas. Firstly, we determine the type of transformation based on the shape of the transformation matrix: a 3x3 matrix indicates a projective transformation, while a 2x3 matrix indicates an affine transformation. For projective transformations, we calculate the inverse using **np.linalg.inv** and apply it using **cv2.warpPerspective**, ensuring smooth interpolation and handling borders with a constant black color. For affine transformations, we use **cv2.invertAffineTransform** to calculate the inverse and apply it using **cv2.warpAffine**, with similar border handling and interpolation.

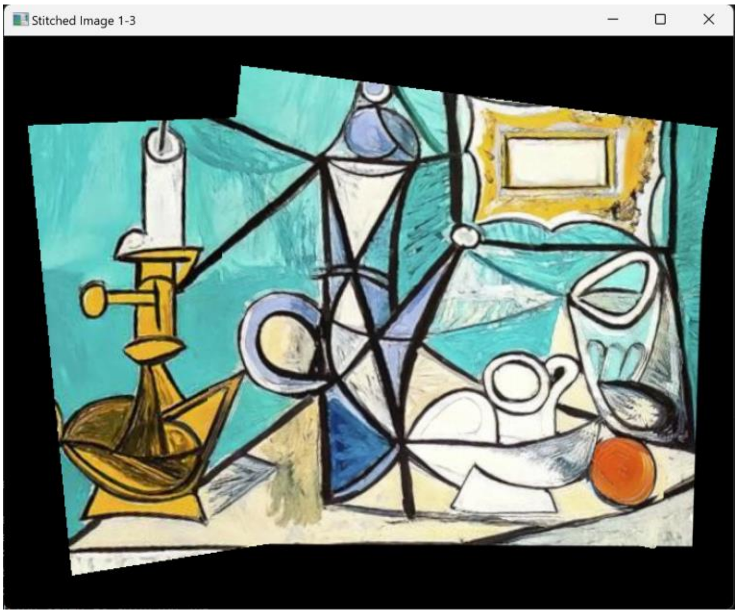
e) The stitch function combines two images, **img1** and **img2**, through a two-step process to achieve smooth transitions in overlapping regions. Initially, a weighted blending is applied to merge both images with equal contribution, creating a seamless combination. Next, the function calculates the absolute differences between the images to highlight variations and enhance the smoothness of transitions. These differences are further integrated into the blended image, resulting in a final stitched output.

For each puzzle, the stitched images will be presented step by step .
we will represent the third puzzle:

1. **Stitched Image 1-2:** Represents the combination of the first two pieces of the puzzle.



2. **Stitched Image 1-3:** Shows the integration of the third piece into the existing assembly.



3. **Final Stitched Image:** Displays the completed puzzle after all pieces have been stitched together, in this puzzle is the **Stitched Image 1-3.**

f) After we finish everything we needed with the puzzle pieces, we initialize the solution canvas using the first piece of the puzzle, which is directly saved as its absolute position without any transformation. Then, for each subsequent piece, we calculate its transformation matrix, apply an inverse transformation to align it to the canvas, and stitch it onto the existing solution. Finally, once all pieces are processed and stitched together, we use the **cv2.imwrite** function to save the fully reconstructed puzzle as **solution.jpg** in the corresponding puzzle directory.

Console Output: Processing and Saving Puzzle Solutions:

```
Starting puzzle_affine_1
Processing piece 2...
Final solution saved as solution.jpg
Starting puzzle_affine_2
Processing piece 2...
Processing piece 3...
Processing piece 4...
Processing piece 5...
Final solution saved as solution.jpg
Starting puzzle_homography_1
Processing piece 2...
Processing piece 3...
Final solution saved as solution.jpg
```