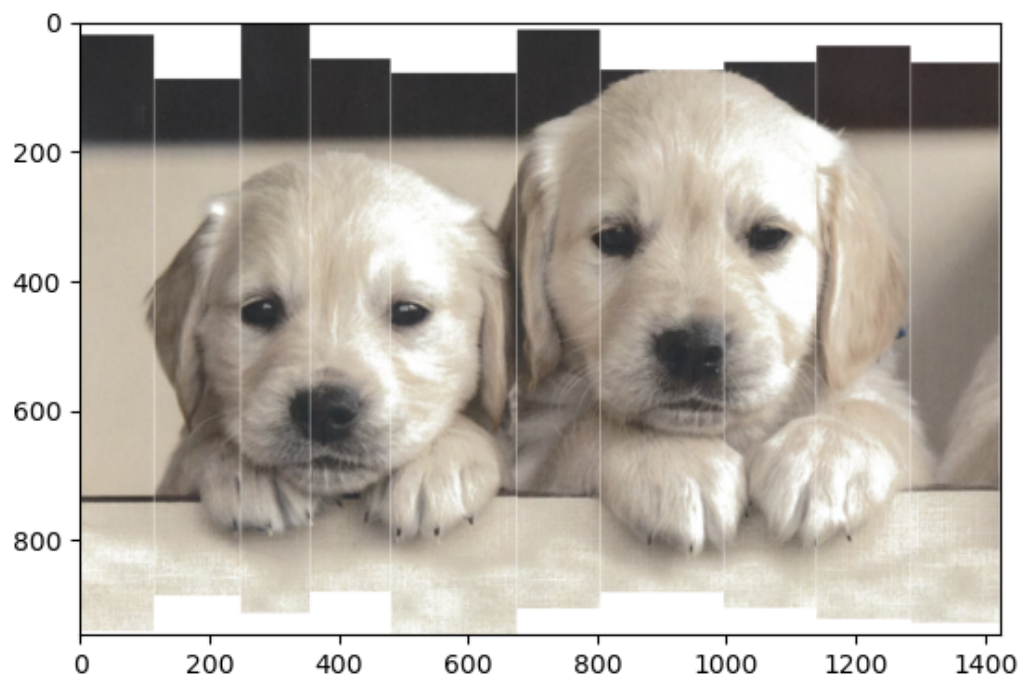
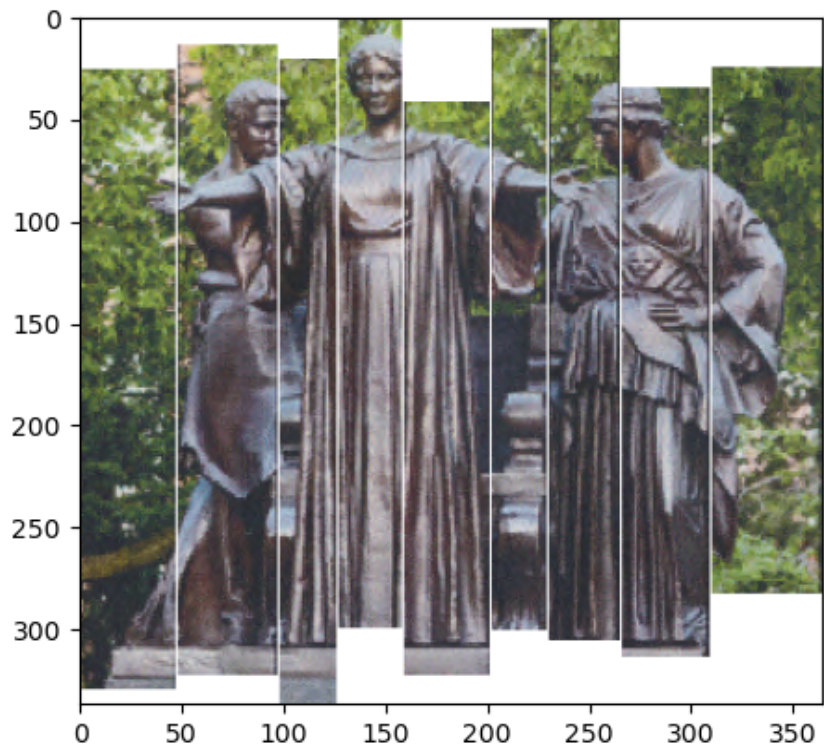


Q 3.3

Stitched results:



Description

I use exactly zero mean normalized cross correlation as my similarity functions, and I also use 3 columns to calculate similarity value.

```
for current_offst in range(-offst, offst + 1):
    start_idx = max(0, -current_offst)
    end_idx = min(img1.shape[0], img2.shape[0] - current_offst)
    if end_idx > start_idx:
        # pixel_number = end_idx - start_idx + 1
        # diff = img1[start_idx:end_idx, 0] - img2[start_idx +
current_offst:end_idx + current_offst, -1]
        # sum_double = np.sum(diff ** 2) / pixel_number
        distance = 0
        for i in range(3):
            A = img1[start_idx:end_idx, i]
            B = img2[start_idx + current_offst:end_idx + current_offst, -1-
i]

            mean_A = np.mean(A)
            mean_B = np.mean(B)
            norm_A = A - mean_A
            norm_B = B - mean_B
            numerator = np.sum(norm_A * norm_B)
            denominator = np.sqrt(np.sum(norm_A ** 2) * np.sum(norm_B ** 2))
            zncc = numerator / denominator if denominator != 0 else 0
            distance += 1 - zncc / (i+1)
        if distance < min_dist:
            min_dist = distance
            min_offst = current_offst
```

I firstly used 3 columns together to calculate ZNCC. However, I realized that similar pixels calculated ZNCC should have a higher weight, and then it turned out to be correct.