Q1:

1. $\text{softmax}(\vec{z} - c1) = \dfrac{\exp(z_i - c1)}{\sum_{j=1}^{K} \exp(z_j - c1)}$

$$= \frac{1}{\sum_{j=1}^{K} \exp(z_j - z_i)}$$

$$= \frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)} = \text{softmax}(\vec{z}).$$

2. ① if $i = j$.

$$\frac{\partial y_i}{\partial z_i} = \exp(z_i) \cdot \frac{1}{\sum \exp(z_j)} + \exp(z_i) \cdot \frac{-\exp(z_i)}{(\sum \exp(z_j))^2}$$

$$= \text{sofmax}(z_i) - \text{sofmax}^2(z_i).$$

② if $i \neq j$.

$$\frac{\partial y_i}{\partial z_j} = - \text{sofmax}(z_i) \cdot \text{softmax}(z_j).$$

3. Chain Rule:

$$\text{softmax}(\vec{z}) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}.$$

①

$$\therefore \frac{\partial y_i}{\partial x} = \sum \frac{\partial y_i}{\partial z_j} \cdot \frac{\partial z_j}{\partial x}$$

$$\therefore \frac{\partial z_i}{\partial x} = w_i \quad , \quad \frac{\partial z_j}{\partial x} = w_j .$$

$$\therefore \frac{\partial y_i}{\partial x} = [s(z_i) - s^2(z_i)] \cdot w_i - \sum_{i \neq j} s(z_i) \cdot s(z_j) \cdot w_j .$$

②

$$\frac{\partial y_i}{\partial w_j} = \sum \frac{\partial y_i}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_j} .$$

$$= [s(z_i) - s^2(z_i)] \cdot x_j - \sum_{i \neq j} s(z_i)(s(z_j)) \cdot w_i$$

③

$$\frac{\partial y_i}{\partial \vec{u}} = \sum \frac{\partial y_i}{\partial z_j} \cdot \frac{\partial z_j}{\partial u} = -\sum \frac{\partial y_i}{\partial z_j} .$$

$$= \left[\sum_{i \neq j} s(z_i) s(z_j)\right] - s(z_i) + s^2(z_i)$$

Q2:

1. $V\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix}$ $\qquad Vx$: spin.

$V\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}-\frac{\sqrt{2}}{2}\\ \frac{\sqrt{2}}{2}\end{bmatrix}$

2. $V^T x$: spin $\vec{x}$.

3. $\Sigma = \begin{pmatrix}3 & 0\\ 0 & 5\end{pmatrix}$ $\quad V^T = \begin{pmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{pmatrix}$

$\therefore x = \begin{pmatrix}\frac{1}{\sqrt{2}}\\ 0\end{pmatrix} = \Sigma V^T x = \begin{pmatrix}\frac{3}{2}\\ -\frac{5}{2}\end{pmatrix}$

$x = \begin{pmatrix}0\\ \frac{1}{\sqrt{2}}\end{pmatrix} = \Sigma V^T x = \begin{pmatrix}\frac{3}{2}\\ \frac{5}{2}\end{pmatrix}$

$x = \begin{pmatrix}-\frac{1}{\sqrt{2}}\\ 0\end{pmatrix} = \Sigma V^T x = \begin{pmatrix}-\frac{3}{2}\\ \frac{5}{2}\end{pmatrix}$

$x = \begin{pmatrix}0\\ -\frac{1}{\sqrt{2}}\end{pmatrix} = \Sigma V^T x = \begin{pmatrix}-\frac{3}{2}\\ -\frac{5}{2}\end{pmatrix}$.

4. calculate SVD of B:

$B = U\Sigma V^T$

$\therefore V^T$ rotate vector

Σ rescale

U rotate again.

5. Σ is singular value.

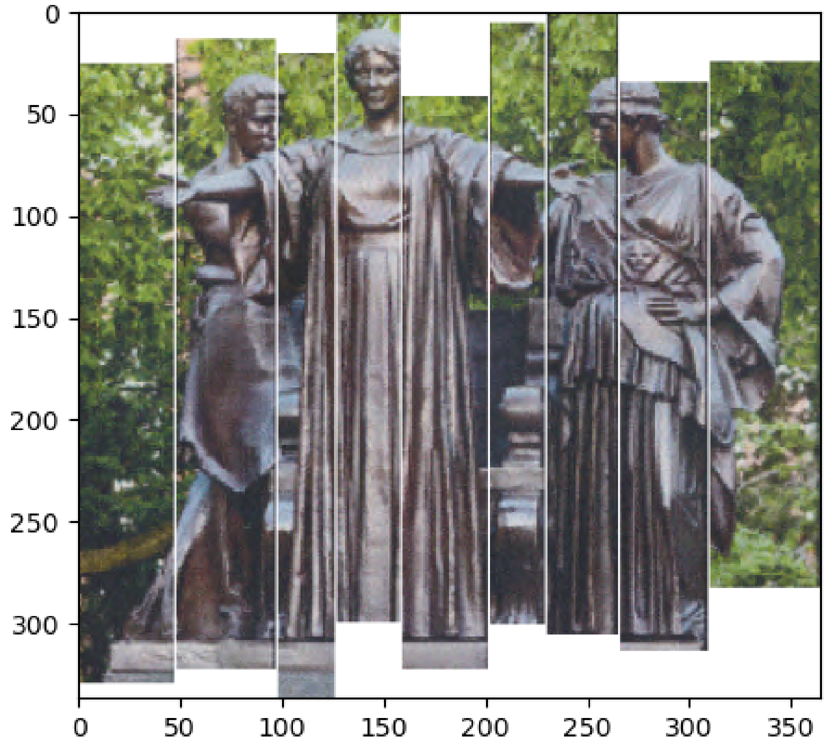
$$U \Sigma V^T x = 0$$

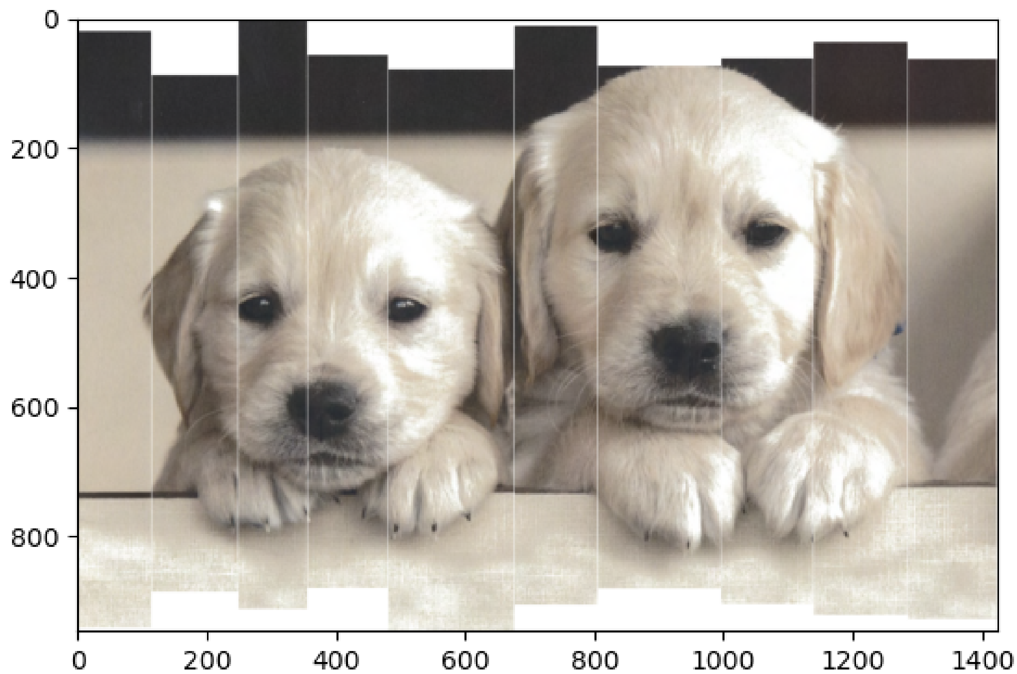
$\therefore \Sigma V^T x = 0.$

$\therefore \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} x = 0 \quad \therefore \quad x = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$

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.