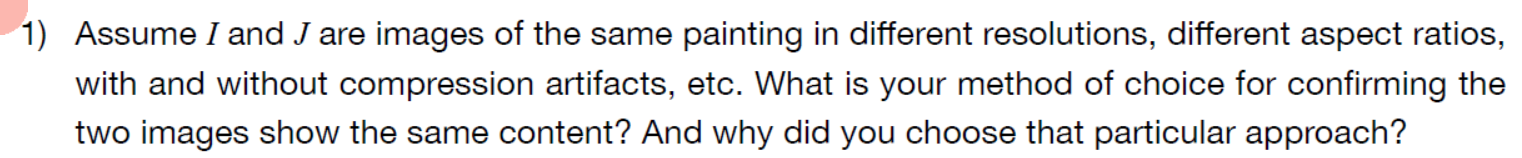
**Interview Challenge:**

**Computer Vision & Machine Learning for Image Retrieval Systems**

**Image Matching**

**The core functionality of Image Retrieval is based on the ability to identify matching images.**

****

*Approach 1: Non-ideal case (e.g. photographs of the painting)*

*If scale is different or perspective distortion is likely to be in at least one of the images, a more sophisticated approach will be necessary. For example, a Key Point Matching strategy based on Scale Invariant Feature Transform (SIFT) could be exploited, which is already implemented in opencv and its patent expired in the year 2020.*

*Since Key Points Matching (KPM) technique preserves structure information, a grey scale conversion previous to key point detection could be a good option. If color-structure information is relevant to the comparison, it is a good idea to apply the KPM technique to chrominance (Cb and Cr) and luminance (Y) channels rather than RGB channels. Where the latter is equivalent to the previous gray-scale conversion.*

*A very simple metric that can be used is the sum of distances between n best matches (those satisfying a ratio test, sorted and truncated). If good matches are less than n, assign an infinite metric. The larger the parameter n, the finer the resolution in the metric but it is more likely to have an infinite value.* ***See file image\_matching\_1.ipynb in repo \*\*\* for an draft of the implementation.***

*Once the metric is correctly working, a threshold should be defined using Signal Detection Theory on a labeled dataset, not necessarily large. The threshold could be selected by minimizing a cost function dependent on the cost of False Positives and of False Negatives. If these costs are not provided, the threshold value that maximizes informedness (Youden's J statistic) could be used.*

*Approach 2: Ideal case (e.g. versions of the same image available in different web pages)*

*Assuming the painting has the same scale in both images (the same part of the painting spans the same image proportion), and there is no aspect ratio distortion, translations, nor perspective distortion, i.e. images are centered plane projections; the following processing steps could be applied:*

*1. Crop both images in both dimensions to the smaller dimension. This solves the issue of aspect ratio difference.*

*2. Resample both images to a resolution coarser than that of the image with coarsest one. This reduces the computational cost of subsequent steps and allows comparison with classic metrics. Possibly, a resolution as low as 256x256 is good enough to distinguish classical painting, but finer resolution could be necessary to distinguish imitations. If one of the images has a resolution low enough, could be better to down-sample only the image with the finest resolution.*

*3. If one of the images could be in grey-scale, convert both images to grey-scale.*

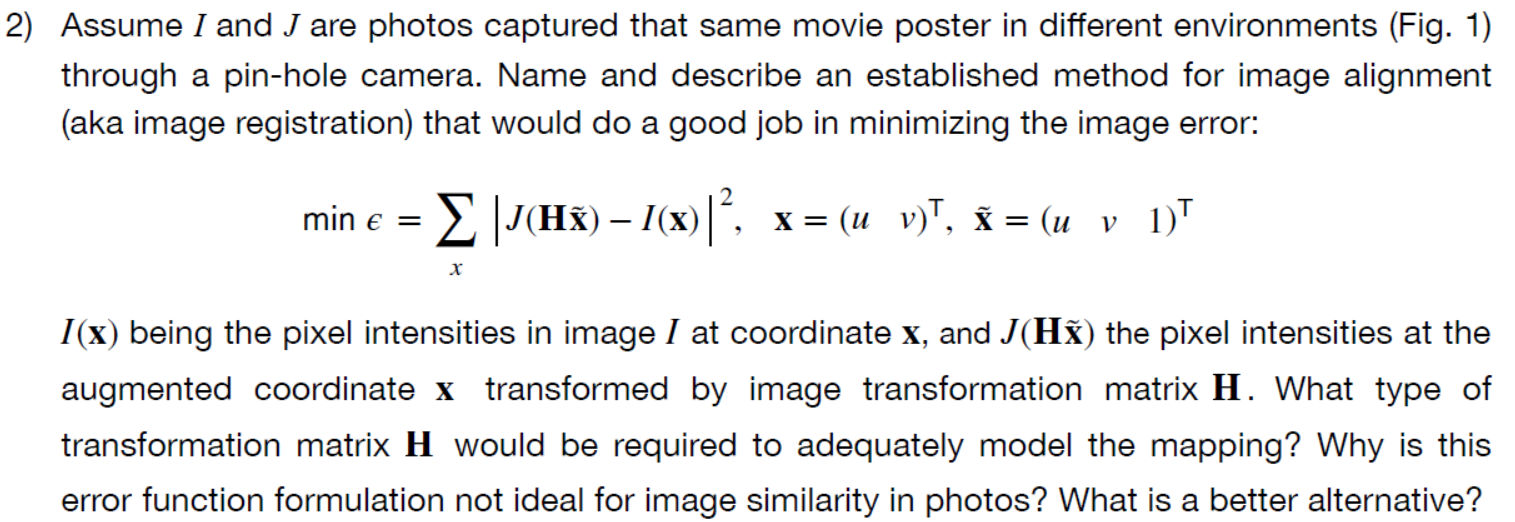
*4. If the images are to be compared with distance metrics, e.g. as mean square error, at least one of the images should be intensity rescaled in order to match the intensity scale of the other image. If the images are to be compared with correlation metrics, e.g. dot product, intensity re-scaling is not necessary.*

*If centering is not guaranteed, the central part of one image (e.g. 1/3 by 1/3) could be used as template for template matching in the other image.*

*A quicker approach could be measuring the correlation between histograms for each color channel. This does not require the 1-3 preprocessing steps. However, this approach is likely to confuse different paintings of the same artistic style, due to the loss of structure information. An obvious example is the case of the Chiaroscuro, which presents histograms with high frequencies for dark colors.*

Hybrid approach

Apply Approach 2 to images preprocessed using Approach 1.

****

*A method for minimizing the image error could be Stochastic Gradient Descent. However, since the cost function is quadratic, a second-order method as Newton’s could be more efficient.*

*In order to model the mapping, the transformation matrix should be a 2x3 matrix defining a (6 DOF) 2D affine transform (scale, rotation, 2 translations, and 2 shears). This is the best option for the definitions given for the coordinates and augmented coordinates. However, a full 2D affine transform cannot model perspective distortions as foreshortening or depth.*

*It is much better to use a homography (aka perspective transform), which has 8 DOF.*

*The proposed error function is not good for image similarity in photos because of two reasons:*

* *It is computationally very expensive because it uses all the pixels, which could lead to millions of terms.*
* *The error function has low signal-to-noise ratio because a significant number of pixels do not actually match between the images. This is because of several reasons:*
  + *When images J and I are photos, they are different perspective projections of a hypothetical plane image. So, it is inevitable that one image has pixels (near the boundaries) that do not correspond to any of the pixels in the other image. Note that perspective projection is likely to produce shear and the domain of images is rectangular.*
  + *In a real-world case as the one shown in Fig. 1, it could happen that the images are not projections of the same plane image, but of similar images.*
  + *Compression artifacts affect the matching pixels differently in the images, even in the ideal case of using the same compression method, parameters and resolution.*

*Many of the problems of matching pixels could be mitigated with a pre-processing involving down-sampling of the images. However, it is a better alternative to* ***match key points.*** *That is, the perspective transform, i.e. a 3x3 matrix with 8 DOF, is found by minimizing an error function that compares locations of key points in image I with the projected locations of the matching key points of image J; besides outliers are iteratively removed. The outliers are those matching key points with large reprojection error (distance). This technique is named as RANdom Sample Consensus (RANSAC). A draft of the implementation is given in file* ***image\_matching\_2.ipynb of the repo \*\*\*.***