~~Motivation (5 points) (a) Briefly state the nature of your work and why you chose it. (b) What specific question, goal, or task did you try to address related to structure in the data (e.g. the clusters you found)?~~

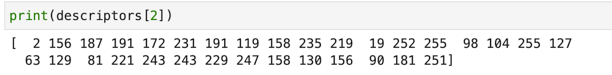
Goal: Our goal was to extract features manually such that the features have the same properties of a convolution filter, namely location invariance, scale invariance and orientation invariance. This was necessary because the faces of all people in images were not exactly vertical. Some people were looking to their left, some to their right and most were looking straight. Also, all the images did not have an identical field of view / scale. Some images were likely shot at a higher optical zoom level so part of their hair or ears was cut off whereas some others did not. Hence, it was necessary to have features that were location, scale and orientation invariant.

Task: Feature Extraction: We extracted 300 features from each image using SIFT and ORB algorithm implementations within OpenCV. A feature is defined as a *keypoint* with a corresponding *descriptor*. *Keypoints* are the most distinctive points of interest in an image that can be used to compare images and perform tasks such as image alignment, registration and object tracking. A *descriptor* is a vector of numbers that describes the visual appearance and properties of the key point. Descriptors can be used to compare and match key points across different images.

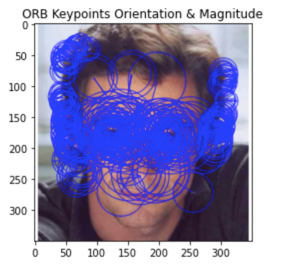
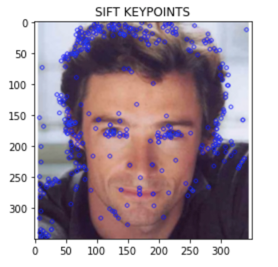
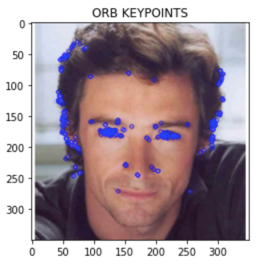
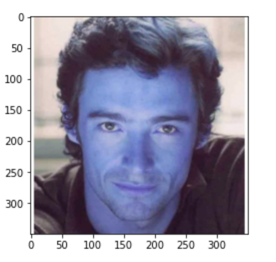
Following is an example of a keypoint showing its 3 properties - pixel location of the keypoint on the image, size of keypoint in pixels and the angle/orientation of the keypoint.



Following is an example of the descriptor vector corresponding to the above keypoint.



The features were visualized as seen below.



● Data Source (5 points): ~~Describe the properties of the dataset (or data API service) you used. Be specific. Your information at a minimum should include but not be limited to~~:

● ~~where the datasets or API resource is located~~,

● ~~what formats they returned/used,~~

● ~~what were the important variables contained in them,~~

● ~~how many records you used or retrieved (if using an API),~~ and

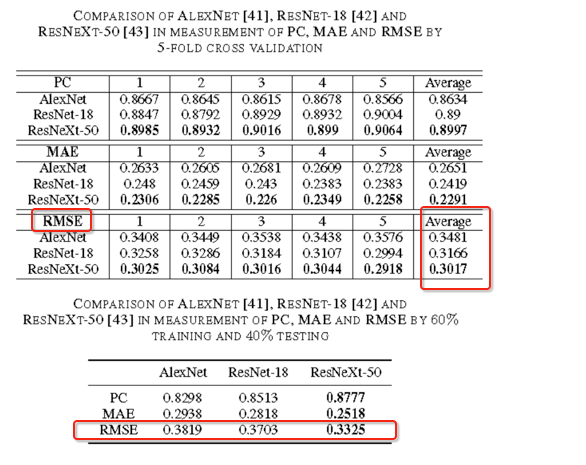
● ~~what time periods they covered (if there is a time element)~~

~~For example, if you downloaded data or used API services, you should state the specific URLs to those files or resources in a way that is trivial for the instructor to retrieve them if needed.~~

We used a multi-paradigm facial beauty prediction dataset named *SCUT-FBP5500* released by the *Human Computer Intelligent Interaction Lab* of *South China University of Technology.* The SCUT-FBP5500 dataset has 5500 frontal faces with diverse properties (male/female, Asian/Caucasian, ages) and diverse labels (facial landmarks, beauty scores in 5 scales, beauty score distribution), which allows computational models with different facial beauty prediction paradigms, such as appearance-based/shape-based facial beauty classification/regression/ranking models

The dataset can be downloaded from https://github.com/HCIILAB/SCUT-FBP5500-Database-Release in the form of a zipped folder. The most important variable in the dataset was the target variable (label) namely the Beauty Score. All the images are labeled with beauty scores ranging from 1 to 5 by 60 volunteers and 86 facial landmarks are also located to the significant facial components of each images.

*Human Computer Intelligent Interaction Lab has* used *AlexNet, ResNet-18,* and *ResNeXt-50* as the benchmarks of the SCUT-FBP5500 dataset, and evaluated the benchmarks on various measurement metrics, including: Pearson correlation (PC), maximum absolute error (MAE), and root mean square error (RMSE). We took note of these metrics to compare our models’ performance to these benchmarks. The benchmark RMSE on the test set ranged from 0.3325 to 0.3819, equating to a MSE of 0.665 (for ResNEXt-50) to 0.7638 (for AlexNet)



---------------------------------- ENDS SO FAR ----------------------------------------

● Unsupervised Learning Methods (20 points):

● Briefly describe the workflow of your source code, the learning methods you used, and the feature representations you chose.

● How did you tune parameters?

● What challenges did you encounter and how did you solve them?

Feature Extraction: 32 Gabors + 9

Dimensionality Reduction: Bag of Visual Features, PCA  
Clustering using KMeans

● Unsupervised Evaluation (10 points)

● What interesting relationships or insights did you get from your analysis?

● What didn't work, and why?

● To summarize your findings, include at least two visualizations (chart, plot, tag cloud, map or other graphic) that summarize your analysis.

Skeleton of project:

Unsupervised learning  
Feature Extraction Techniques Used:

* SIFT (fullform) and ORB (full form) to extract upto 300 feature vectors from images - location invariant...mimic convolution filters in CNN. Each feature vector has size/length >5? properties/attributes like orientation, radius, … circles che.. So each image will be represented as a matrix of upto 300 column vectors
* 32 Gabor Filters to detect various frequencies in the image.
* 9 Edge detection filters like Canny, Prewitt, Sobel, Robers….list all names

Dimension reduction using PCA:

The dataset was transformed into the new feature space above. If u do PCA in original space, only 33% variance is explained by 122500 (350 \* 350 pixels) features whereas PCA done on dataset transformed into Gabor filter space achieving 85% variance on 1st component and 77% on dataset transformed into 9 edge detector space. We reduced dimensionality using both 32 Gabors and Edge detectors from 36 GB to 10 GB This proves blindly doing dimensionality reduction in transformed feature space produces much more useful features than in the original feature space of 122500 features.

Feature Extraction using using Bag of Visual Features

vocabularies and represent each image as a frequency histogram of features that are in the image.

The first step to build a bag of visual words is to perform feature extraction by extracting descriptors from each image in our dataset.

Feature representation methods deal with how to represent the patches as numerical vectors. These vectors are called feature descriptors.

A good descriptor should have the ability to handle the intensity, rotation, scale and affine variations to some extent.

One of the most famous descriptors is Scale-invariant feature transform (SIFT) and another one is ORB.

Upto 300 feature vectors \* 4400(?) = 631264 ORB feature vectors (only done on training set produced 631264 feature vectors. We put them in a bag and tried to cluster them into 500 groups.

K-Means clustering was done on ORB feature vectors to find 500 clusters. Then each image is represented as a weighted sum of these 500 feature vectors. The weights were used to find most frequent/important features/clusters/topics. Histogram.

Supervised Learning using bag of words

* Random Forest and SVR were run on 500 bag of visual features (each feature being a location invariant ORB vector) and gender. Mean Squared Error on Random Forest was 0.37 and on SVR was 0.31. SVR performed better.
* PassiveAgreesiveClassifier was used to include all features including 32 Gabors, 9 edge detectors, bag of 500 visual features, gender, produced a classification accuracy of 30%

1. ANN
2. CNN
3. Dummy all average MSE
4. Interpretation and R2