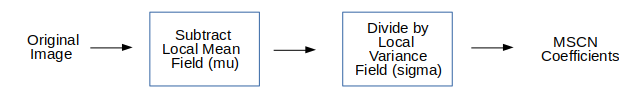
Objective

The objective is to classify images into bad quality or good quality. The approach intensively revolves around doing feature engineering using pixels of images and hence, use these features as input for classification algorithm.

Theory

1. The distribution of pixel values of good quality or natural image varies from that of distorted/bad quality images. After normalization, if we observe the distribution – it represents normal distribution for natural images and otherwise for bad quality/distorted images. The more the deviation from normal distribution more is the distortion in an image.

(One way to normalize image is called MSCN Mean Subtracted Contrast Normalization)



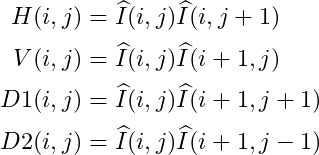
Where,

Local Mean Field – Gaussian Blur of the original image

Local Variance Field – Gaussian Blur of the square of difference of original image and mean

1. Another clearly observable difference between natural and distorted images is the difference between neighbourhood pixel values.

(To measure this, pairwise product of MSCN image with a shifted version of MSCN image. Orientations used here are - Horizontal, Vertical, Left Diagonal, Right Diagonal)



Approach

Predict

Feature Vectors

Extracting Natural Scene Statistics

Original Image

Oversample

Feature Engineering

Using these 5 images( MSCN & 4 pairwise product), a feature vector of size 36\*1 is created, description of which can be found here:

* Shape & Variance by fitting Generalized Gaussian Distribution to MSCN Image.
* Shape, mean, left variance, right variance of horizontal pairwise product by fitting Asymmetric Generalized Gaussian Distribution
* Shape, mean, left variance, right variance of vertical pairwise product by fitting Asymmetric Generalized Gaussian Distribution
* Shape, mean, left variance, right variance of left diagonal pairwise product by fitting Asymmetric Generalized Gaussian Distribution
* Shape, mean, left variance, right variance of right diagonal pairwise product by fitting Asymmetric Generalized Gaussian Distribution

Model Building

* Oversampling is done using SMOTE technique to make classes balanced
* Stratified KFold(k=10) is used for cross validation and Xgboost model is built.

Result

* Accuracy- 93% in training data, 90-92% in validation data