# Assessing the aesthetic quality of photographs using generic image descriptors

Mohammed Ujjainwala
Chetan Chhabra
Ramkrishna Maheta

# Abstract

Objective is to automatically assess the aesthetic properties of images. In the past, this problem has been addressed by hand-crafting descriptors which would correlate with best photographic practices (e.g. “Does this image respect the rule of thirds?”) or with photographic techniques(e.g. “Is this image a macro?”). Instead of hand-crafting descriptors we are using generic image descriptors to assess aesthetic quality.

Macro image

# Local features in images

Local features refer to a pattern or distinct structure found in an image, such as a point, edge, or small image patch.

They are usually associated with an image patch that differs from its immediate surroundings by texture, color, or intensity

# Generic image descriptors

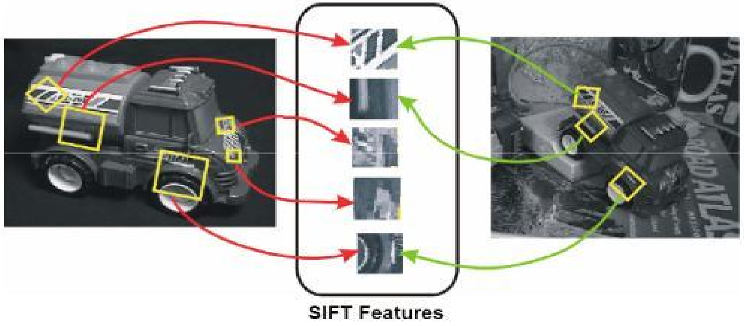
Generic image descriptors:-They contain low level descriptors which give a description about color,shape,regions,texture and motion.

Rather than trying to encode photographic rules explicitly, we encode them implicitly in generic content-based features such as the BOV. Each patch can tell us a lot about the local properties of an image (e.g. “Does this patch contain sharp edges?” or “Is the color of this patch saturated?”).

By aggregating patch-level information into an image-level BOV, one can take into account the global composition (e.g.“Do we have a mix of sharp patches and blurry ones?” or “Is there a dominant color or a mixture of colors in this image?”).

# Scale Invariant Feature Transform (SIFT) Descriptor

SIFT takes an image and transforms it into a collection of local feature vectors .
These feature vectors are distinctive and invariant to any scaling , rotation , or translation of the image.



# Steps of SIFT Algorithm

1. Approximate Keypoint Location .
2. Refining Keypoint Location .
3. Assigning Orientations.
4. Descriptors Computation for each point.

# Bag of Visual Words

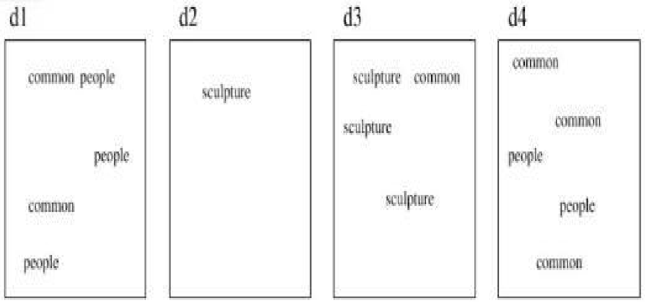
Bag of Visual Words , also called bag-of-feature is probably the most widely used technique.

Texture is characterized by the repetition of basic elements or textons

An unordered set of local patches are first extracted and described, for instance by SIFT descriptors

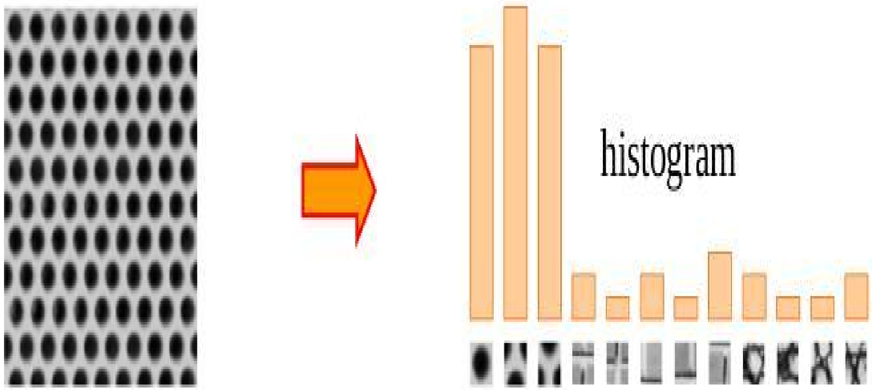
The set of local features extracted from a given image is then transformed into a fixed-length histogram representation by counting the number of local descriptors assigned to each visual word

# Bag of  Words



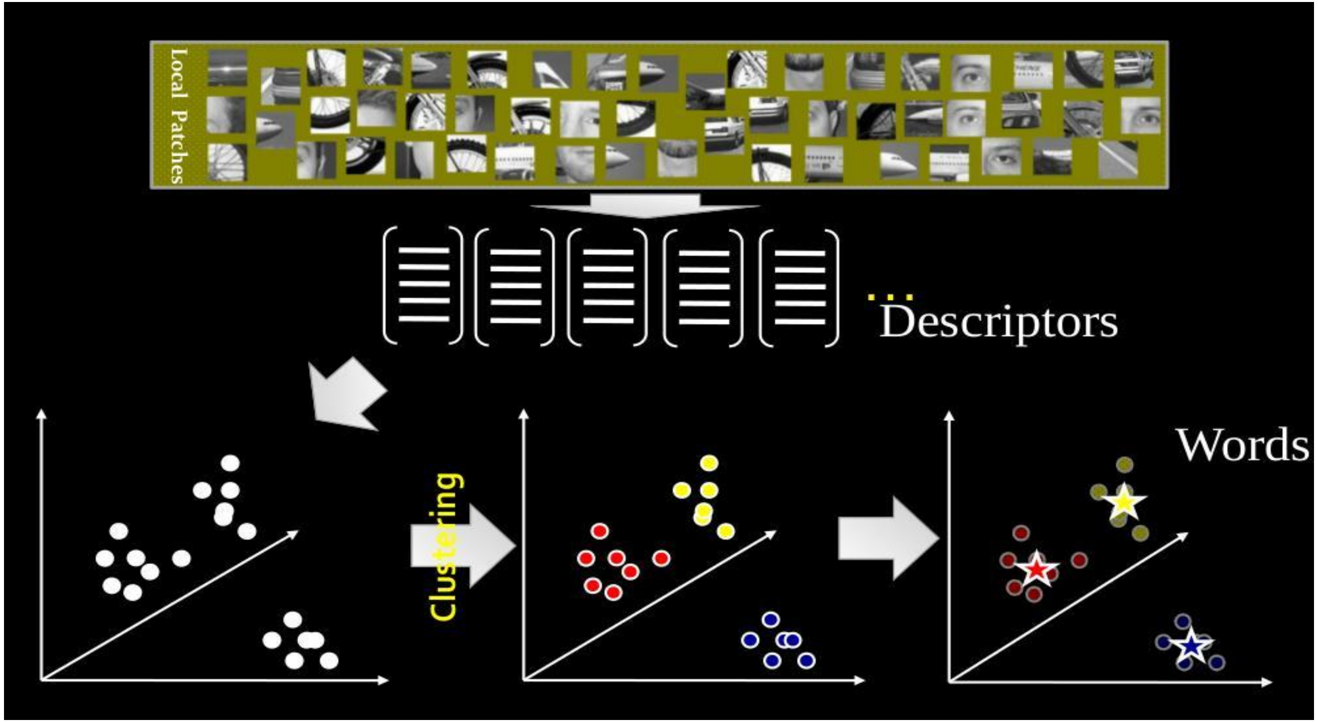

Bag-of-words

| | | | | |
|---|---|---|---|---|
| Common | 2 | 0 | 1 | 3 |
| People | 3 | 0 | 0 | 2 |
| Sculpture | 0 | 1 | 3 | 0 |
| ... | ... | ... | ... | ... |

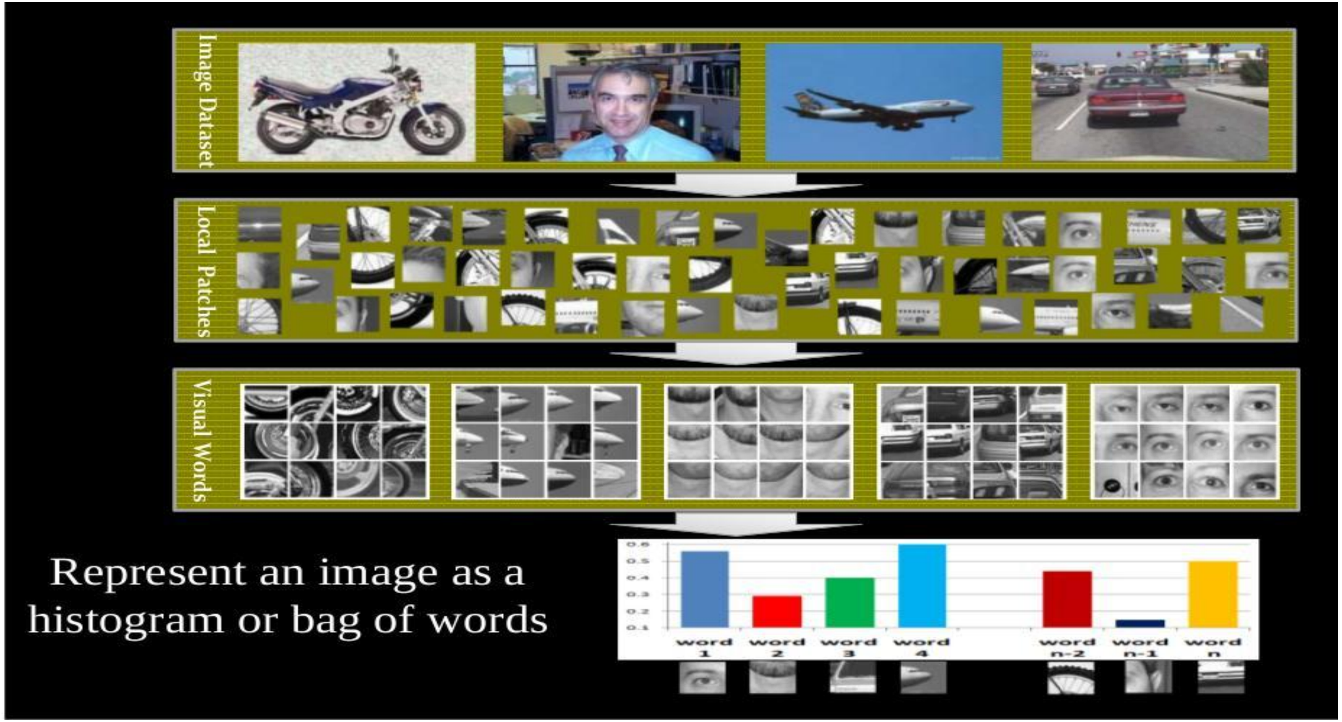


# Bag of Visual Words Model

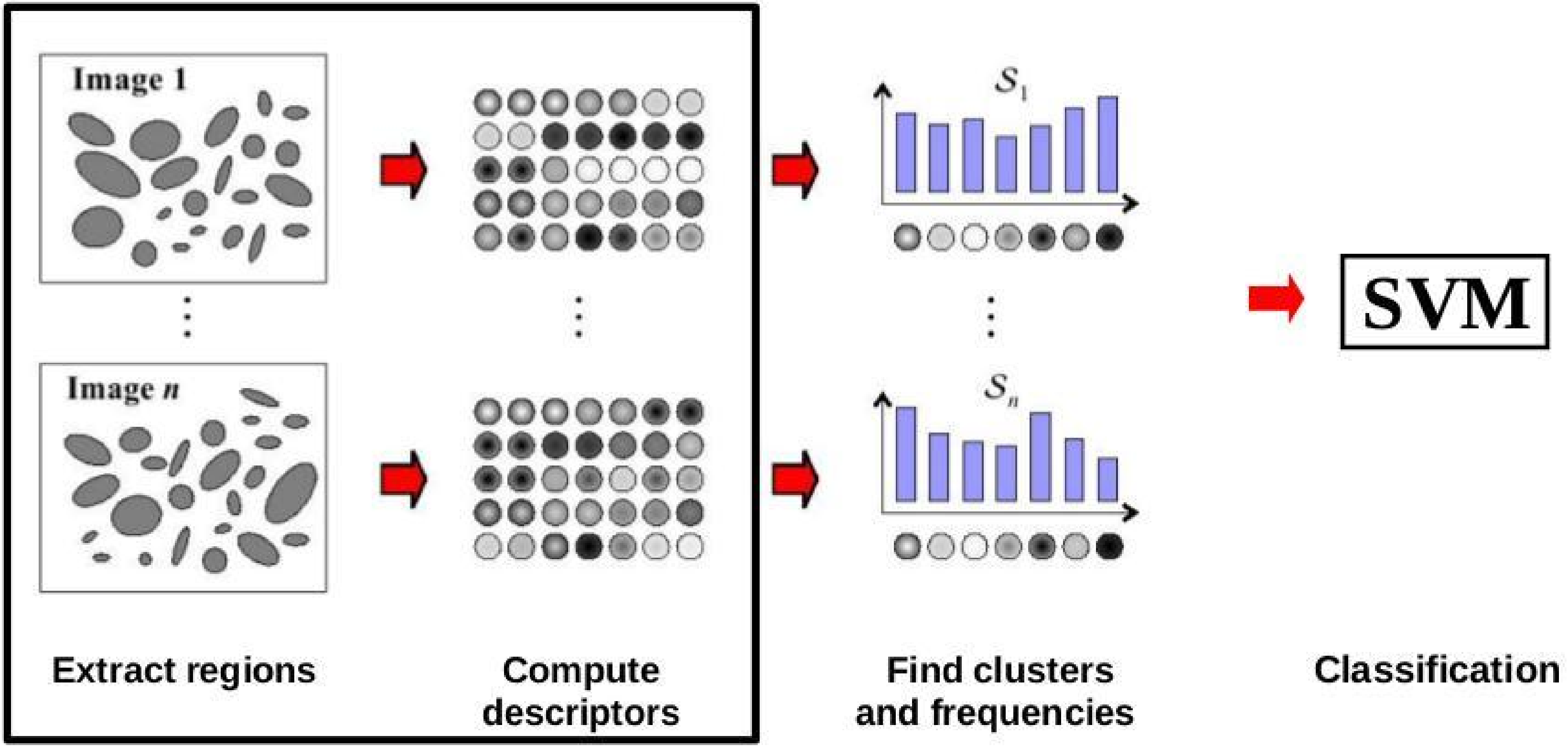
# Bag of Visual Words Model



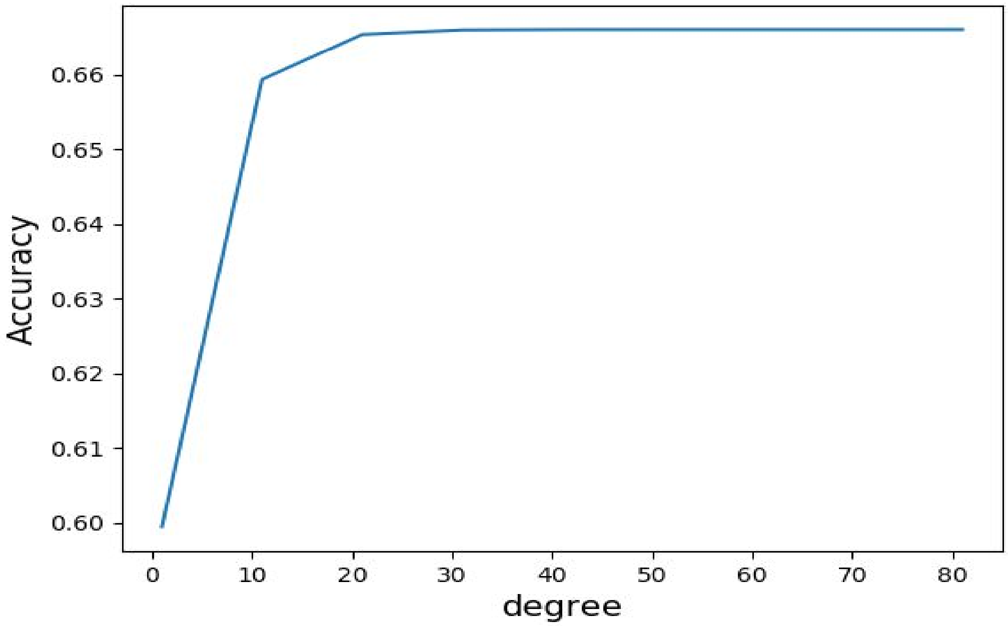
# Bag of Visual Words Model



# Bag of Visual Words Steps

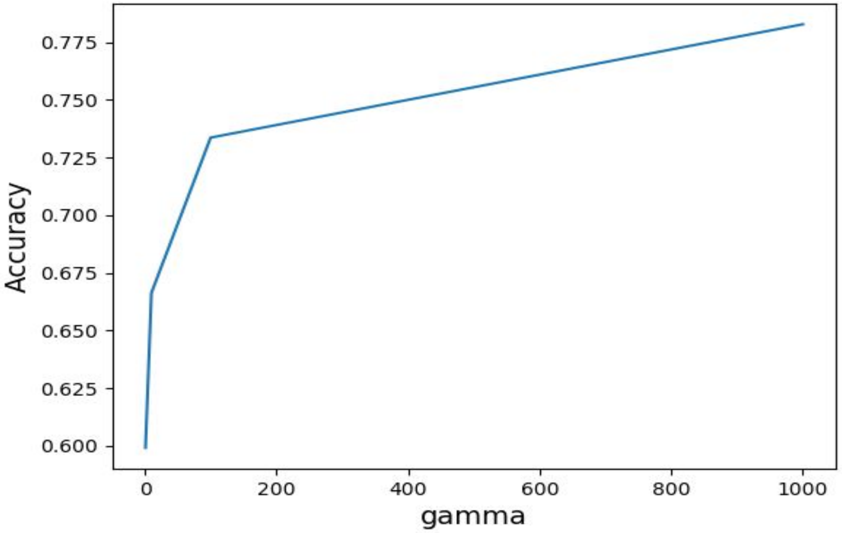


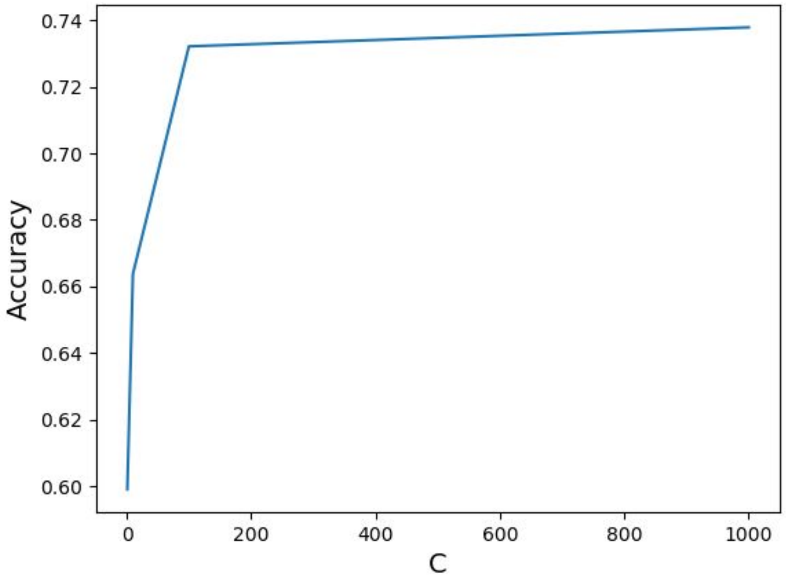
# Experiments : Using SVM




Kernel : Polynomial
Degree vs Accuracy Graph

# Experiments : Using SVM




Kernel : RBF
Gamma vs Accuracy

Kernel : RBF
C vs Accuracy

# Experiments : Using Neural Networks

```
dell@dell-Inspiron-5558:~/smaiproject$ python classifier2.py
0.578313253012
0.674698795181
0.719512195122
0.55421686747
0.55421686747
0.548780487805
0.55421686747
0.621951219512
0.542168674699
0.646341463415
Average acuracy  0.599441669115
dell@dell-Inspiron-5558:~/smaiproject$
```

No of Hidden Layers: 2
Neurons in Layer 1 : 25
Neurons in Layer 2 : 25
Accuracy : 59.94%

# Experiments : Using Neural Networks

```
dell@dell-Inspiron-5558:~/smaiproject$ python classifier2.py
0.614457831325
0.566265060241
0.682926829268
0.602409638554
0.542168674699
0.609756097561
0.626506024096
0.560975609756
0.578313253012
0.609756097561
Average acuracy  0.599353511607
dell@dell-Inspiron-5558:~/smaiproject$ 
```

No of Hidden Layers: 3
Neurons in Layer 1 : 25
Neurons in Layer 2 : 25
Neurons in Layer 3 : 50
Accuracy : 59.935%

# Experiments : Using Decision Tree