Name: Bhavya Samhitha Mallineni
USC ID: 6580252371
USC Email: mallinen@usc.edu
Submission Date: 03/29/2024

REPORT
HOMEWORK4

Problem 1: Texture Analysis

I. Abstract and Motivation :

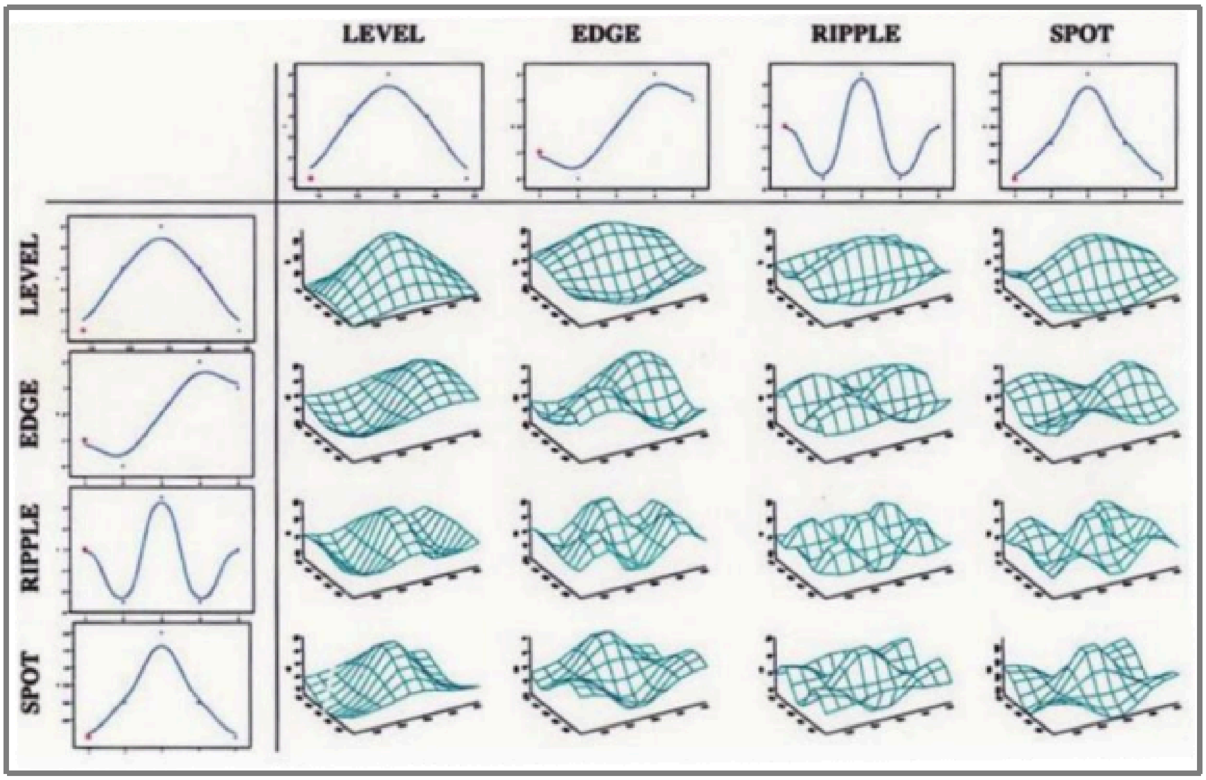
In image processing, texture refers to the spatial variations in pixel intensity, defining patterns repeated throughout an image. Texture analysis, a subset of image processing, involves quantifying and classifying these patterns to discern characteristics like smoothness or roughness. This analysis provides valuable insights into image content, applicable in fields such as medical imaging, geographic analysis, and automated inspection, where distinguishing textures aids in understanding and decision-making.

II. Approach and Procedure:

To identify image features, we employ the Laws filters energy method. This technique utilizes five 1D kernels that, when combined, generate 25 distinct 5x5 kernels. By convolving these kernels with the image and computing the average energy of the resulting convolutions, we extract features. Consequently, each image is characterized by 25 distinct features following Laws filtering. This process enables detailed feature analysis crucial for various image processing applications.

Name	Kernel
L5 (Level)	[1 4 6 4 1]
E5 (Edge)	[-1 -2 0 2 1]
S5 (Spot)	[-1 0 2 0 -1]
W5 (Wave)	[-1 2 0 -2 1]
R5 (Ripple)	[1 -4 6 -4 1]

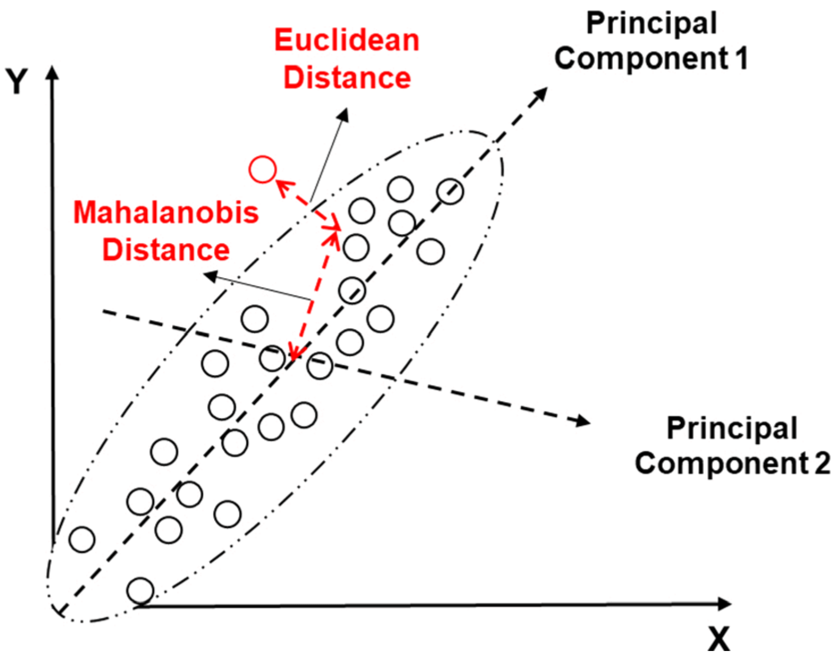
Above are the five kernels which are the L5 (Level), highlights areas of constant intensity, E5 (Edge) is structured for identifying edges, S5 (Spot) accentuates isolated points, W5 (Wave) seems tailored for capturing wave patterns, while R5 (Ripple) will be for detecting ripple-like oscillations.



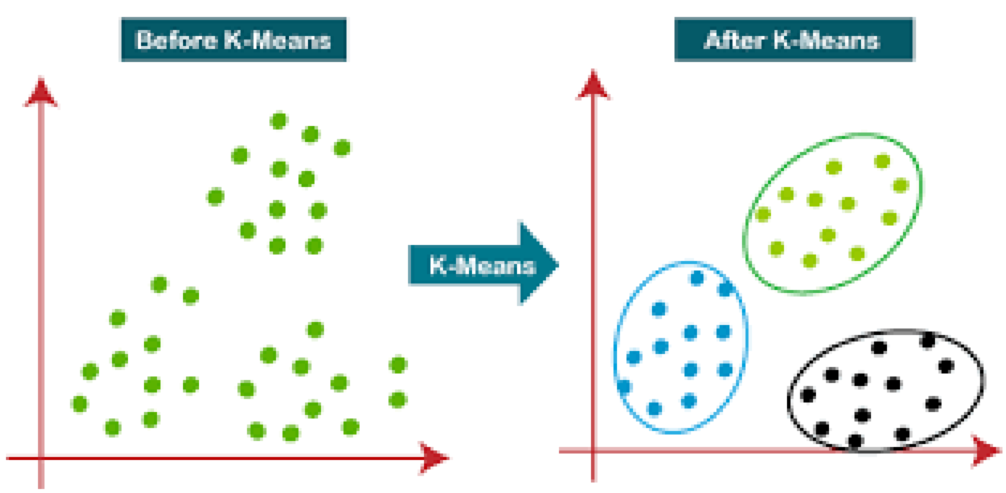
PCA:- Principal Component Analysis is a statistical method that applies an orthogonal transformation to transform a dataset of potentially correlated variables into a set of linearly uncorrelated variables known as principal components. This approach is primarily employed to reduce the dimensionality of data, thus making it easier to interpret without significantly losing information. PCA identifies key basis vectors that capture the greatest variance in the dataset. It begins by calculating the data's covariance matrix, followed by finding the eigenvalues and eigenvectors of this matrix. The eigenvectors create a new dimensional space, while the eigenvalues indicate their importance, allowing for the simplification of the dataset by selecting the most critical components. In our problem we are using pca and reducing the feature count to 3.

Now we have 2 feature vectors one with 25 vectors and the other with only 3 features. We then applied different classification methods.

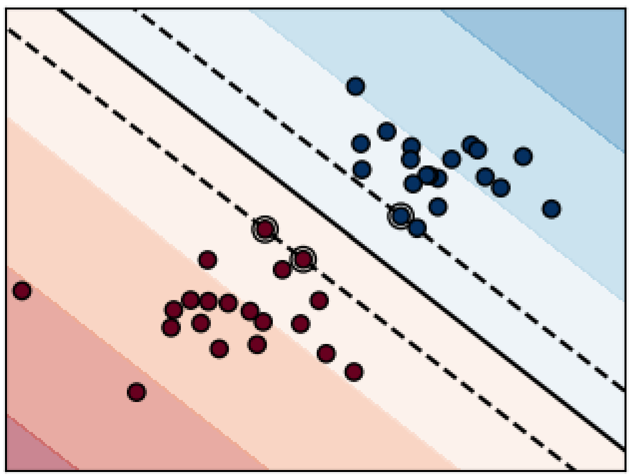
A. **Nearest Neighbors using Mahalanobis distance** is a variant of the k-Nearest Neighbors technique that incorporates the Mahalanobis distance for measuring similarity between instances. This distance metric considers the covariance among the variables to adjust for scale and correlation, thus providing a more accurate reflection of the distances in a multivariate space.



B. i) Unsupervised Learning (Kmeans): K-means is an unsupervised clustering algorithm designed to group n data points into K distinct clusters. Each cluster is defined by its centroid, which is the average position of all points in the cluster. The algorithm iteratively performs two key steps: it assigns each data point to its nearest centroid, forming *k* clusters, and then recalculates the centroids based on the current cluster memberships, optimizing the clusters' compactness. The process starts with the selection of *k* initial centroids and continues by alternating between assignment and recalibration of centroids until the centroids stabilize, meaning their positions do not change significantly. K-means is favored for its straightforward implementation and effectiveness in organizing large datasets into meaningful clusters. In our case $k = 4$.



C. i) Supervised Learning (SVM): Supervised learning is a type of machine learning where an algorithm is trained on labeled data, which means the algorithm learns from input data that has been tagged with the correct output. The goal is to enable the algorithm to make accurate predictions or decisions when it encounters new, unseen data by generalizing from the training set. It is used in applications such as image recognition, speech recognition, and forecasting. Support Vector Machine is a powerful supervised learning model used for classification and regression tasks. It works by finding the hyperplane that best separates different classes in the feature space. The best hyperplane is the one that has the maximum margin, which means it is farthest from the nearest training data points of any class. SVM can handle linear and nonlinear data and works well with high-dimensional data. It is effective in cases where the number of dimensions exceeds the number of samples, making it a robust choice for many machine learning challenges.



III. Experimental Results:

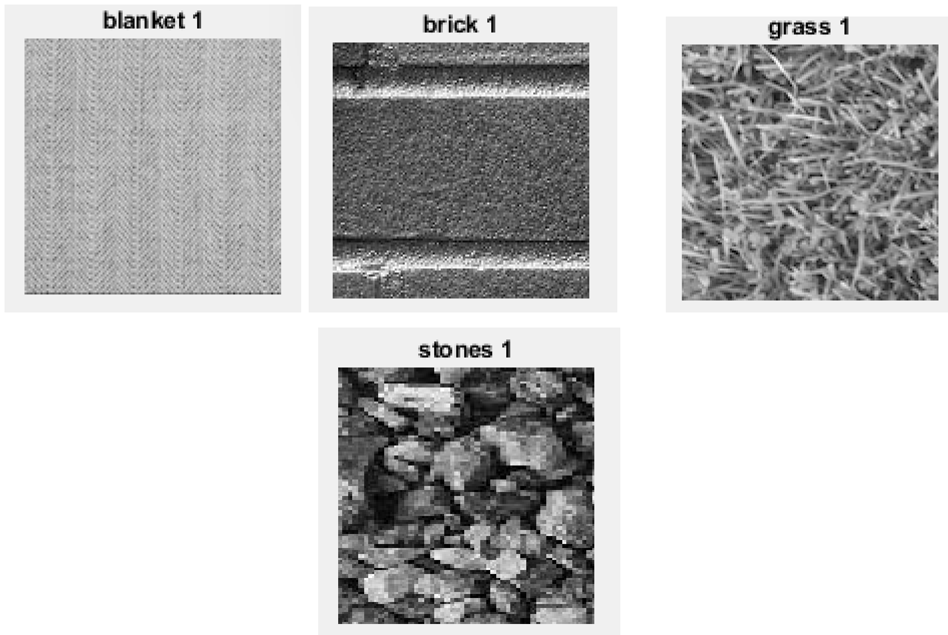


Figure 1: 4 types of Texture we are classifying

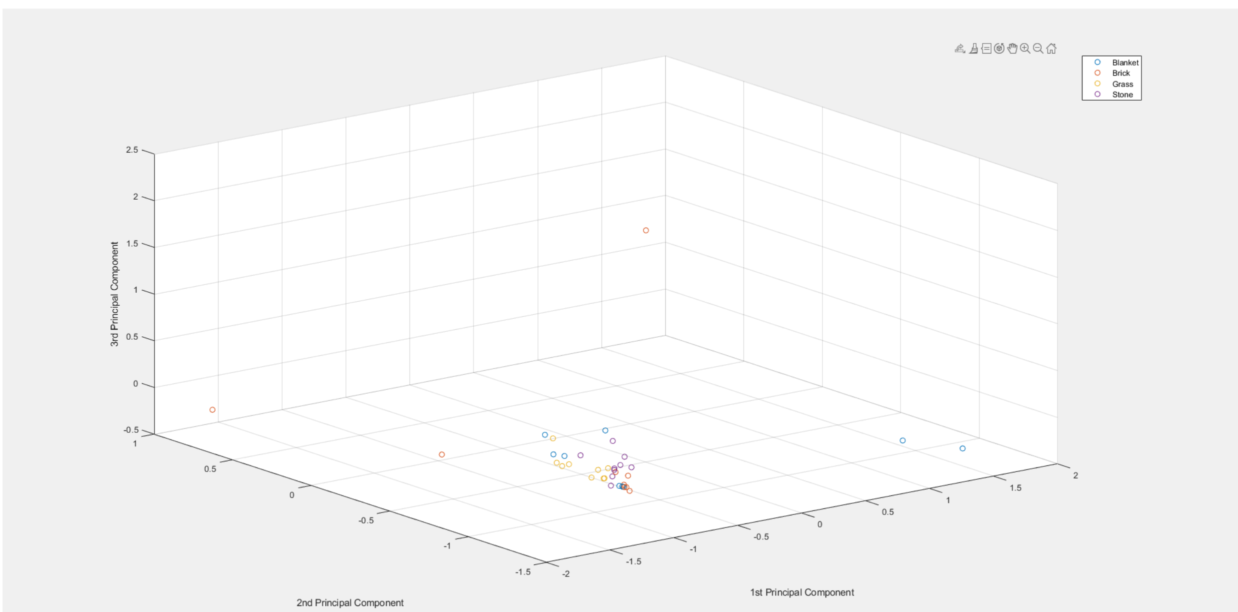


Figure 2: PCA Visualization

	Accuracy	Error
Nearest Neighbor	33.33%	66.6%
KMeans (25 features)	50%	50%

KMeans (3 features)	**58.33%**	**41.67%**
SVM (25 features)	**16.67%**	**83.33%**
SVM (3 features)	**41.67%**	**58.33%**

Table 1: Accuracies and Errors of texture Classification

IV. Discussion :

A) For this problem, the image was convolved with all 25 filters, resulting in 25 feature representations that were subsequently reduced to 3 features using PCA. The graph in the experimental results illustrates the reduction to 3D feature vectors. After calculating the energies of each image, the discriminant power was evaluated. It was observed that the strongest feature was at index 21, corresponding to the R5' * L5 convolution when the image was mirror-padded, and the weakest was at index 5, corresponding to the L5' * R5 convolution (indices 11 and 2, respectively, when zero-padded). From these observations, it is evident that distinguishing between textures such as blanket, grass, and stone was challenging. Although the brick texture showed some distinctive features, they were also quite similar to those of the other textures. Consequently, the accuracy in classifying these textures was notably low due to the minimal distances between them.

B) **Unsupervised Learning: Kmeans**
In the unsupervised learning segment of our study, we applied the K-means clustering algorithm, utilizing MATLAB's K-means function, to both the original feature space, which comprised 25 features, and a reduced feature space with only 3 features. Notably, the 'cityblock' distance metric outperformed other distance measures in terms of accuracy. When K-means was directly applied to the text data with four clusters, the accuracy was significantly low. However, training on a designated train set and then evaluating it showed an improvement in purity, leading to enhanced accuracy. Interestingly, between the original feature set of 25 features and the reduced feature space, the latter demonstrated superior performance, indicating that dimensionality reduction via PCA effectively distilled essential information, thereby improving the clustering outcome.The enhanced performance in the reduced feature space suggests that the reduction process managed to eliminate noise and redundant information, which often hampers the effectiveness of clustering algorithms. This highlights the importance of feature selection and dimensionality reduction in unsupervised learning, especially in complex datasets where the intrinsic patterns are not readily apparent. The success of the 'cityblock' distance metric also underscores the significance of choosing an appropriate distance measure tailored to the specific characteristics of the data and the objectives of the analysis. This approach paves the way for more nuanced and effective clustering strategies, capable of uncovering subtle patterns in data.

Supervised Learning: SVM

For constructing a support vector machine, a polynomial kernel was employed along with various solvers: SMO , ISDA, L1QP , and SGD . Among these, ISDA emerged as the most effective, leading to its selection for the final model. Notably, higher accuracy was achieved in a reduced 3-feature space compared to the 25-feature space. Although the overall accuracy remained modest, it is believed that augmenting the training dataset could significantly enhance the model's ability to delineate the proper support vectors and define the hyperplane more accurately. This underlines the importance of sufficient and relevant training data in optimizing the performance of support vector machines, particularly when employing complex kernels and sophisticated solvers in the model's construction.

In my observations comparing K-means and SVM, K-means outperformed SVM when images were classified based on the majority cluster. Conversely, SVM excelled in a PCA-reduced space. The provision of additional training and testing data could significantly enhance the models' accuracy in classification tasks. This suggests that the effectiveness of these algorithms is highly contingent on the dimensionality of the data and the volume of available data, highlighting the critical role of adequate data in achieving optimal classification performance.

Problem 2 : Texture Segmentation

V. Abstract and Motivation :

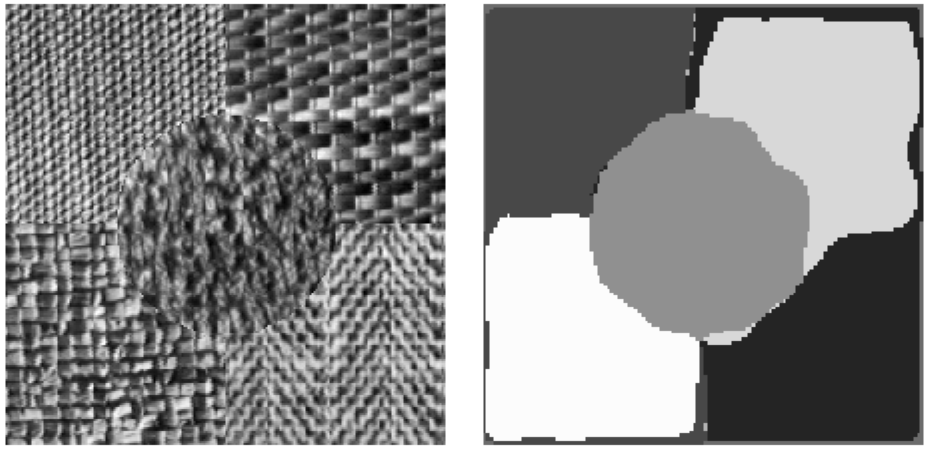
Texture segmentation plays a vital role in computer vision and image analysis, focusing on dividing an image into sections that differ in texture. This method is indispensable for the interpretation and study of images, as it allows for the identification of various objects and areas based on their texture characteristics. Texture involves the spatial distribution of colors or intensity in a segment of an image, offering essential insights into the surface's structural layout. The main challenge of texture segmentation is the variety and intricacy of textures present in images from the real world. To tackle this, algorithms for texture segmentation typically utilize texture features such as the density of edges, their orientation, or spectral components to distinguish between textures. Approaches including the use of Gabor filters, wavelet transformations, and deep learning techniques have been prominent for their ability to detail the complex aspects of textures.
This segmentation technique is pivotal in a range of applications, from medical imaging, where it aids in identifying different tissues, to the analysis of aerial imagery for classifying landscapes, and even in the quality control processes in manufacturing. The integration of machine learning, particularly through convolutional neural networks (CNNs), has markedly enhanced the precision and speed of texture segmentation, continuously driving forward research to improve its effectiveness and utility in various fields.

VI. Approach and Procedure:

Texture segmentation employs a variety of techniques to categorize and separate areas within an image based on their texture characteristics. These include statistical techniques that analyze pixel intensity distributions; model-based techniques, which apply predefined texture models; structural techniques, focusing on the arrangement of texture components; and machine learning approaches, especially deep learning with convolutional neural networks (CNNs), which excel in identifying intricate texture patterns through learned representations. Each strategy has its strengths, with statistical and model-based methods being straightforward and easy to interpret, whereas structural approaches and machine learning offer more robust solutions for dealing with complex textures, thereby improving the segmentation process's precision and effectiveness for a wide range of uses.

A)Basic Texture Segmentation: For this problem, we follow the same procedures as in Problem 1 to obtain the feature vectors. However, instead of calculating the average energy for the entire image, we determine the energy of each pixel based on varying window sizes. As the window size shifts, so does the energy associated with each pixel. Various energy window sizes were explored to optimize this process. Once the energy values are computed, the image is reshaped, and K-means clustering is applied. Given that the image contains five distinct segments, we choose to utilize five clusters for the segmentation process. After assigning each part of the image to a specific cluster, we allocate a unique gray level value to each cluster. This step is crucial for visually representing the segmentation outcome, allowing us to distinctly observe the segmented regions within the image. This approach enhances the segmentation's precision by adjusting the analysis to the local texture characteristics, leading to a more detailed and accurate segmentation.

B) Advanced Texture Segmentation: in order to get more better of an output we can use pca to reduce the dimentianality.Principal Component Analysis (PCA) plays a transformative role in texture segmentation by efficiently condensing the complexity of texture data. It reduces the dimensionality of the dataset while preserving the most significant features, thereby improving computational efficiency and maintaining the integrity of essential texture information. This dimensionality reduction facilitates a more streamlined segmentation process, enhancing the accuracy and speed of identifying distinct texture regions within an image. By focusing on the principal components that capture the most variance, PCA helps in distinguishing between varied textures with greater clarity, ensuring a more effective and precise segmentation. This approach not only simplifies the analytical process but also boosts the performance of texture segmentation algorithms.



Example of Texture Segmentation

VII. Experimental Results:

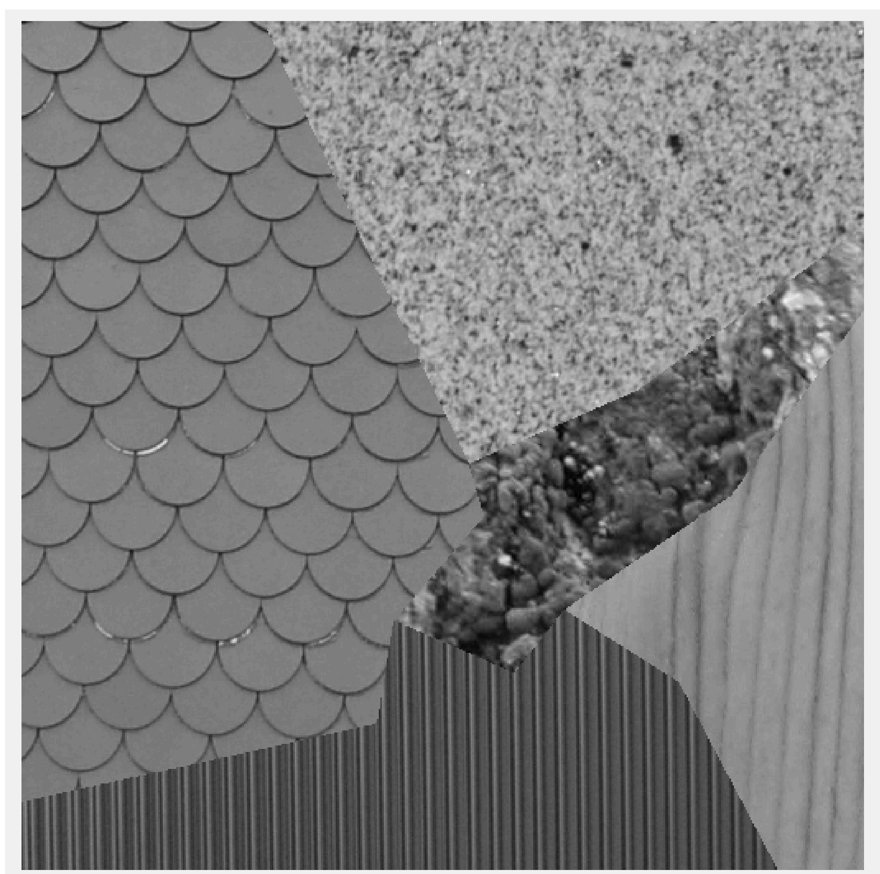


Figure 3: Original Mosaic Image

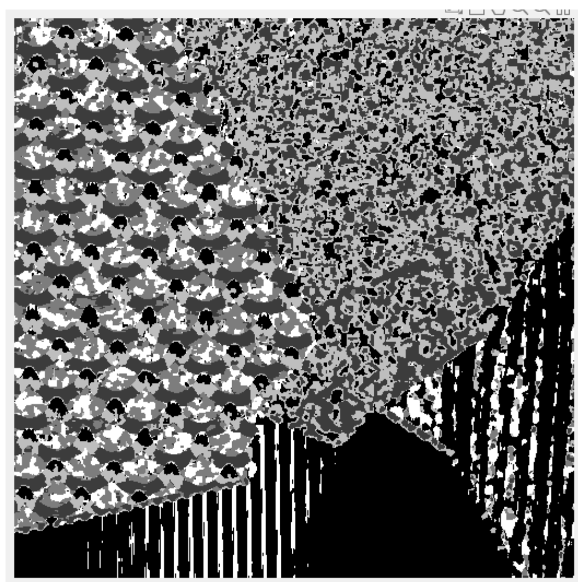


Figure 4: Window size = 5

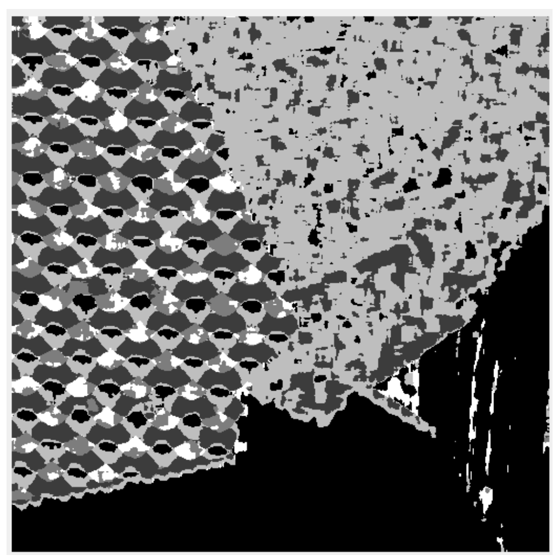


Figure 5: Window Size = 11

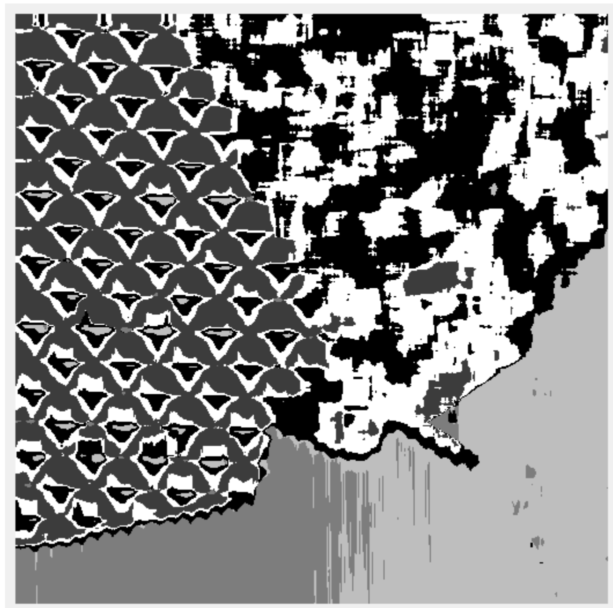


Figure 6: Window size = 21

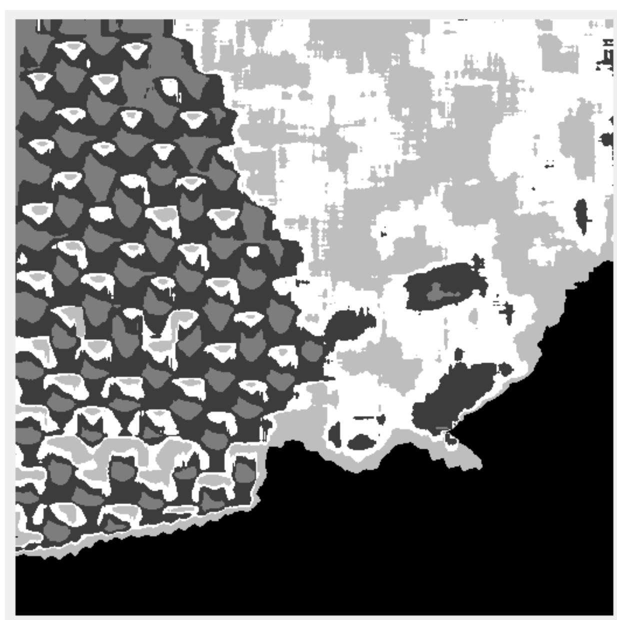


Figure 7: Window Size = 31

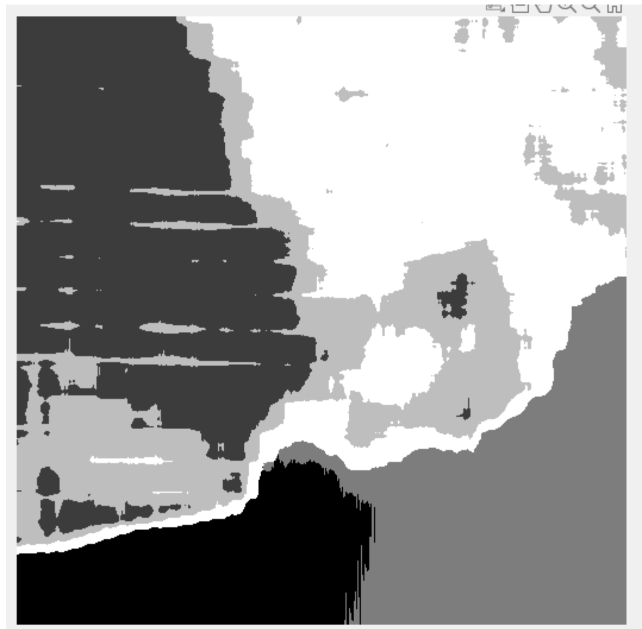


Figure 8: Window size = 55

Figure 9: window size= 85

Advanced Texture segmentation:

Window size = 55

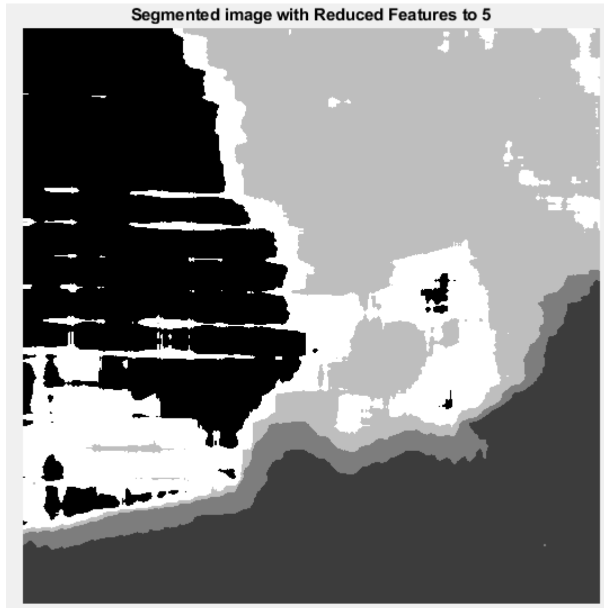


Figure 10: Reduced Features to 5

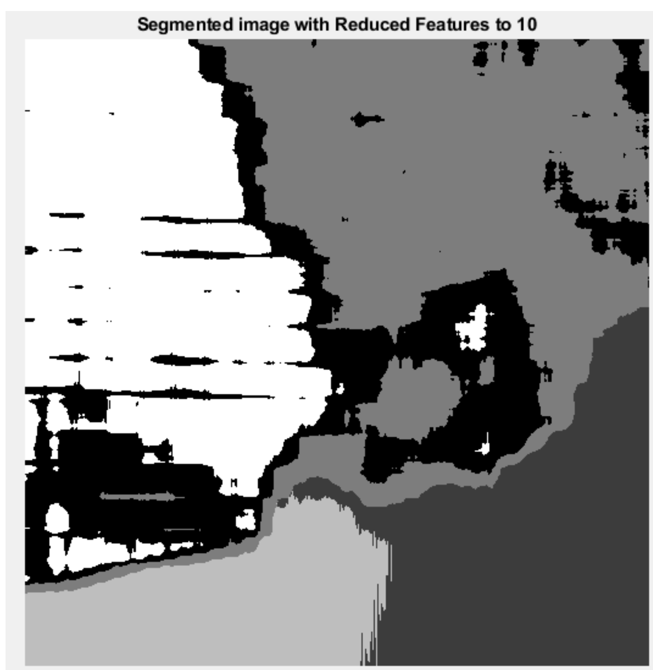


Figure 11: Reduced Features to 10

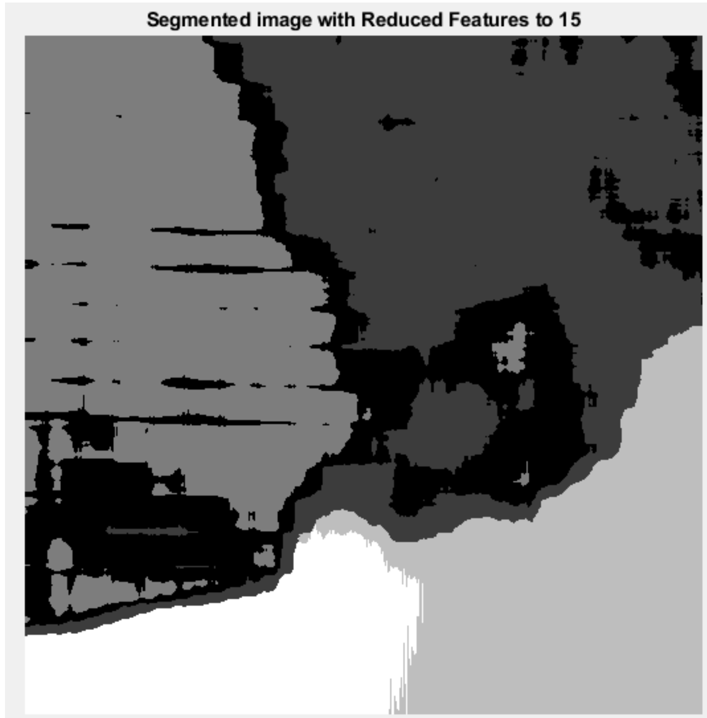


Figure 12: Reduced Features to 15

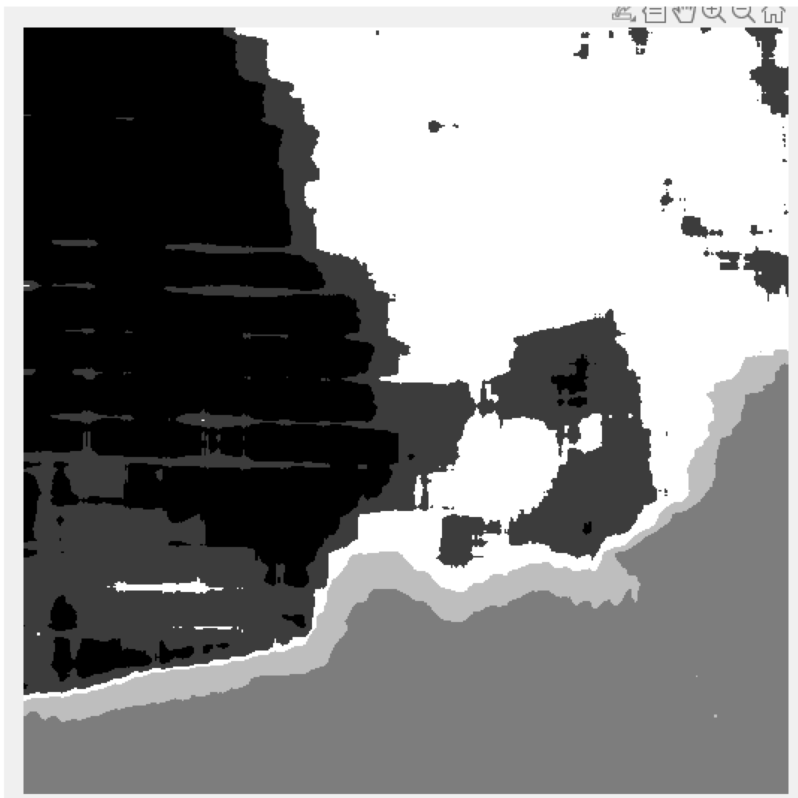


Figure 13: Reduced Features to 20

Window Size = 31

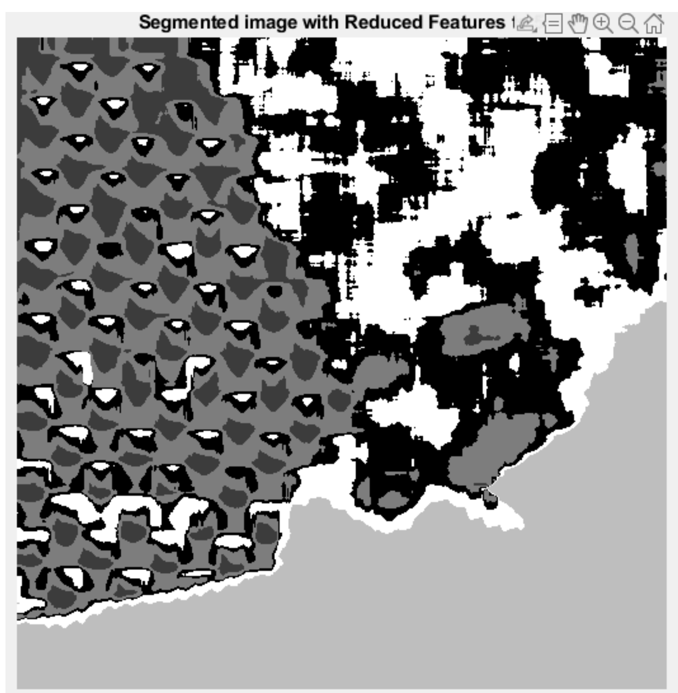


Figure 14: Reduced Features to 5

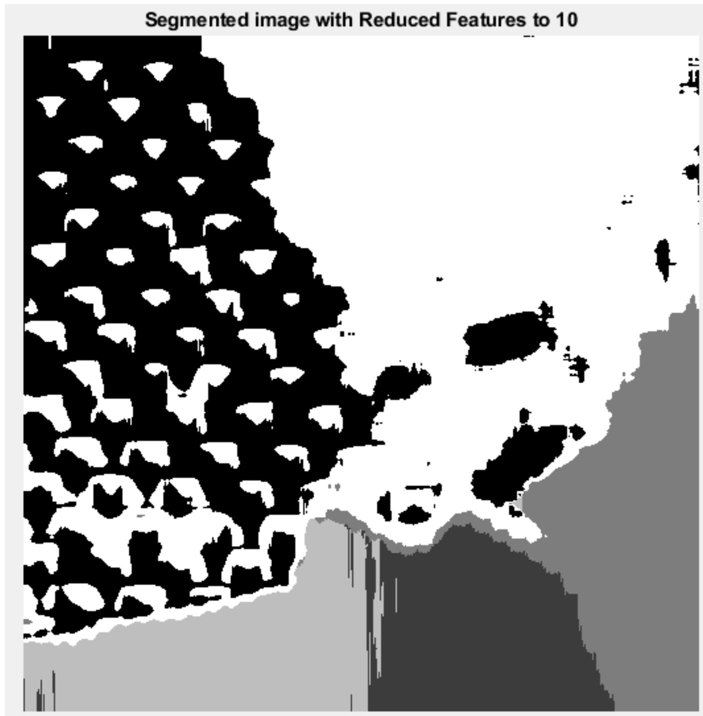


Figure 15: Reduced Features to 10

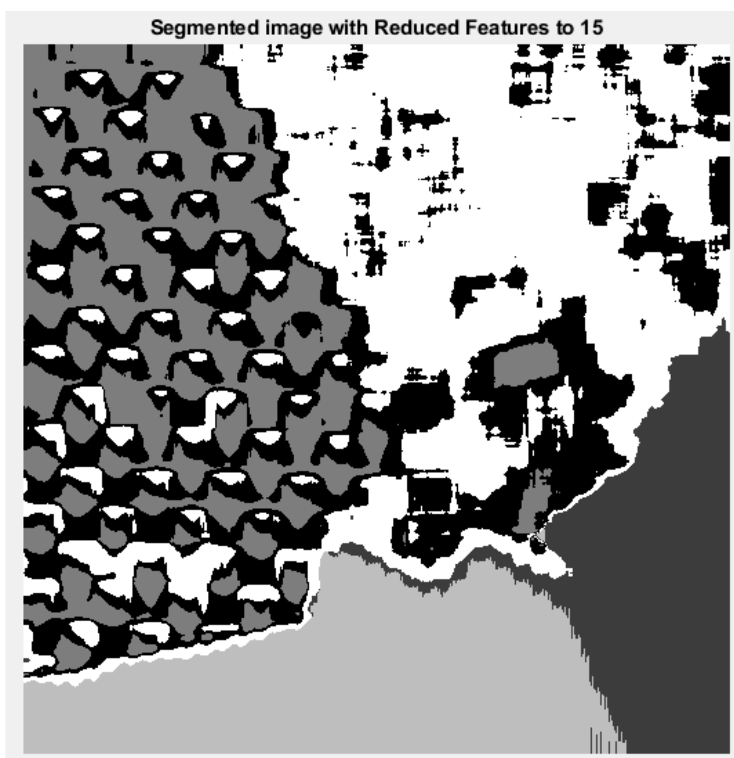


Figure 16: Reduced Features to 15

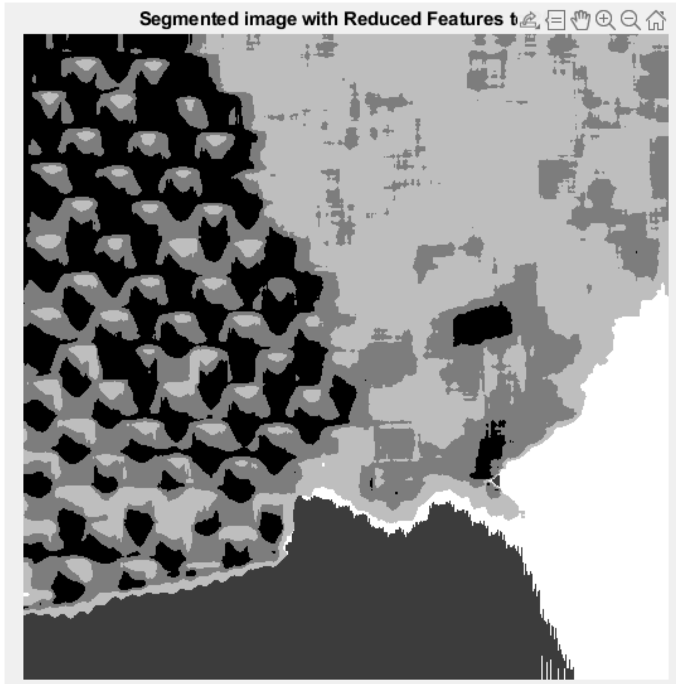


Figure 17: Reduced Features to 20

Window Size = 7

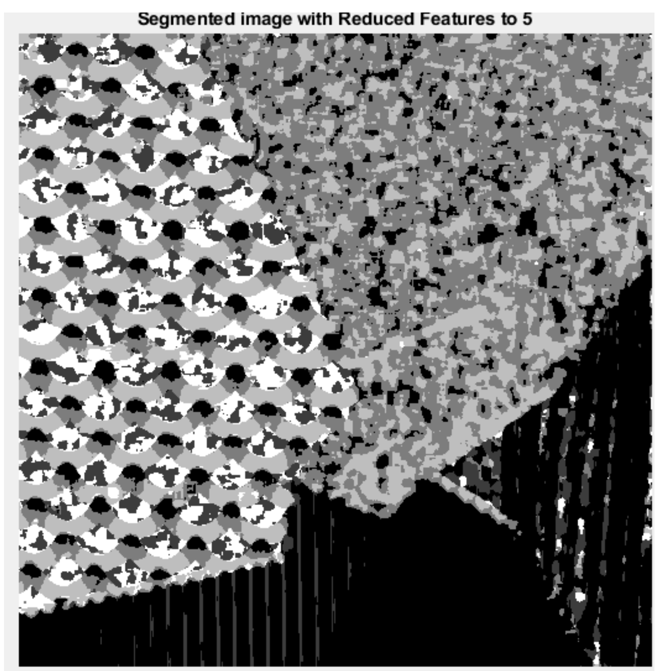


Figure 18:Reduced Features to Size 5

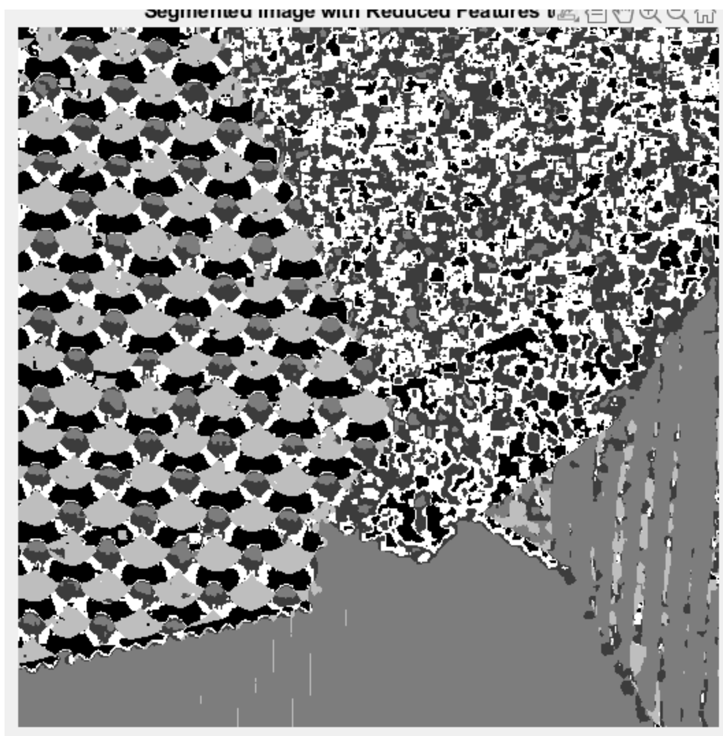


Figure 19: Reduced Features to Size 10

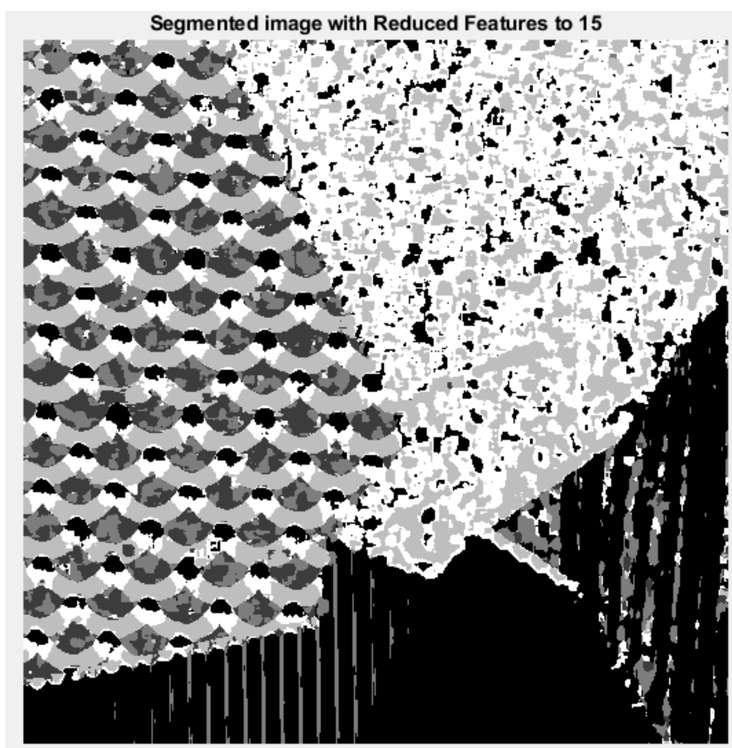


Figure 20: Reduced Features to 15

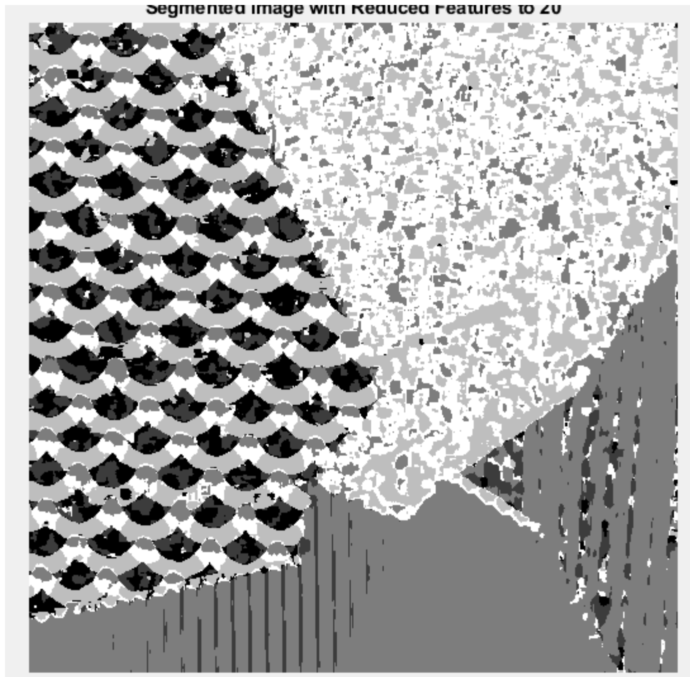


Figure 21:Reduced Features to 20

VIII. Discussion :

A) Basic Texture Segmentation: The segmentation process revealed a critical observation: the size of the window significantly impacts the results. As the window size increased, there was a noticeable reduction in noise, leading to segments that were more well-defined and easily distinguishable. Conversely, utilizing a smaller window size, such as 5, resulted in an extremely noisy image, compromising the segmentation's accuracy. In such cases, the clustering outcomes were erratic, with values varying randomly rather than consistently across different textures. However, as the window size expanded, this issue diminished, allowing for clearer and more precise segmentation. This pattern suggests that selecting an appropriate window size is crucial for optimizing texture segmentation, balancing between noise reduction and the preservation of essential texture details to achieve accurate classification and segmentation of the image's various parts.

B) Advanced Texture Segmentation:For this problem, I experimented with reducing the feature space to a minimum of 5 and a maximum of 20, and then tested these adjustments across different window sizes. Based on my observations, the reduction in feature space did not lead to a significant change in the outcomes. The results appeared to be similar to those achieved in Part A of this question. Ultimately, it seems that the window size and the texture of the image are the primary factors that influence the segmentation outcome. This finding underscores the importance of choosing the correct window size to capture the essential characteristics of the image's texture effectively. It also highlights the inherent challenge in texture segmentation, where the physical texture of the image plays a crucial role in determining the success of the segmentation process, more so than the mere adjustment of feature space dimensions.

Problem 3 : SIFT and Image Matching:

IX. Abstract and Motivation :

SIFT

The Scale-Invariant Feature Transform (SIFT) stands as a complex algorithm crafted for the identification and delineation of local image attributes, ensuring consistent matching across variations in scale, orientation, and lighting. Through detailed examination at various scales, SIFT identifies critical areas adorned with unique features. Following this, it determines their orientations and formulates descriptors that encapsulate the fundamental characteristics of each keypoint's local vicinity. This comprehensive process guarantees that the identified features are distinct and remain consistent across typical image alterations. The reliability and accuracy of SIFT render it vital for a plethora of computer vision applications, including recognizing objects, merging images for panoramic views, and constructing 3D models, where precise matching of features plays an essential role..

Image Matching :

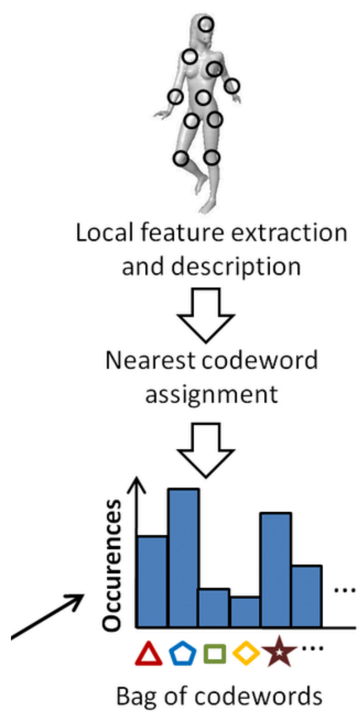
Image matching stands as a pivotal component in computer vision, crucial for pinpointing similar points or attributes across varying images. Such a method is essential in contexts like 3D modeling, where aligning points between images facilitates the perception of depth, and in surveillance systems, to identify objects or people across different scenarios. SIFT enhances image matching through its robust set of features, which are resilient to changes in scale, orientation, and light conditions, thus ensuring effective matching across varied representations of the same scene or item. Through the extraction of unique keypoints and their detailed description, SIFT enables accurate alignment even under demanding circumstances, including alterations in perspective, size, or illumination. This functionality positions SIFT as a key resource for achieving precise and durable image matching across numerous computer vision tasks.

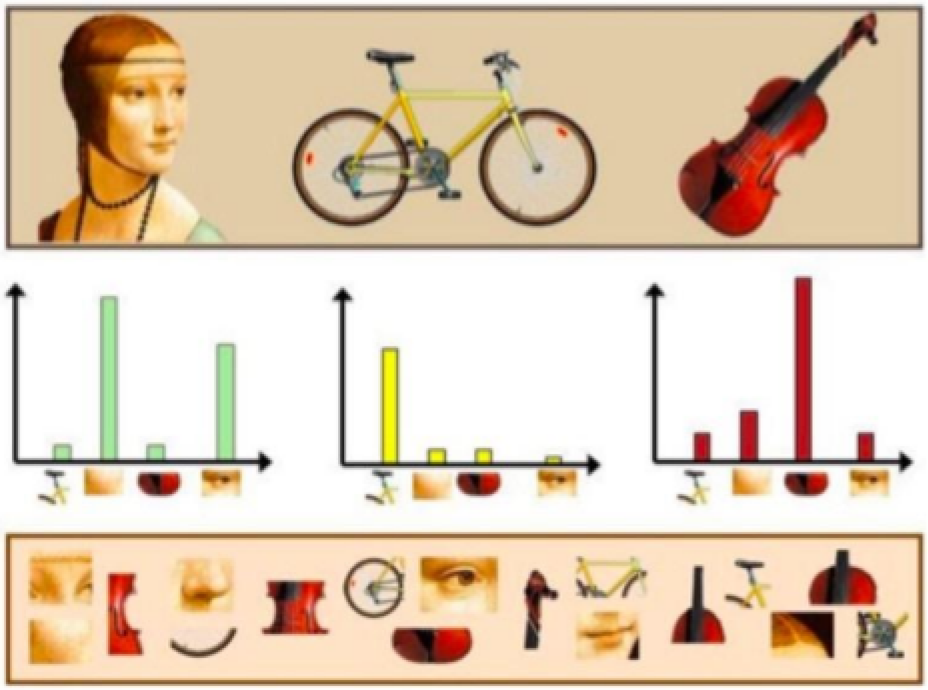
Bag of Words:

The Bag of Words (BoW) model, originally from text analysis, has been effectively adapted for image processing, where it functions as a method for encapsulating image features. In this model, images are analyzed as assortments of "visual words" by categorizing feature descriptors into a set vocabulary. The core process entails detecting significant features in an image, grouping these to form a visual lexicon, and subsequently encoding each image through a histogram that tracks the occurrence rate of each visual word. Such histograms offer a succinct yet informative snapshot of the image's essence, streamlining activities like image classification, search, and scene identification by easing the comparison and matching workflow.

Within the BoW framework, SIFT emerges as a pivotal element for extracting initial features. Thanks to the uniqueness and stability of SIFT descriptors against variations in size, orientation, and lighting, they are perfectly suited for crafting a dependable visual vocabulary. SIFT features, once extracted and categorized (typically via k-means clustering), contribute to forming this vocabulary. An image is then depicted through a histogram representing the frequency of these visual words, blending SIFT's detailed feature extraction and BoW's overarching image representation strategy. This synergistic approach leverages the accuracy of local feature identification with the overarching simplicity of image description, significantly boosting the efficiency of image processing operations.

Bag of Words description





X. Approach and Procedure:

Salient Point Descriptor: Answers for the questions are discussed below in the discussion section.

Image Matching: To perform image matching in our problem, we initially extracted features and descriptors utilizing the vl_sift function from the VLFeat toolbox. The VLFeat toolbox is a comprehensive library that offers a variety of computer vision algorithms, including those for image matching, object recognition, and texture analysis. Specifically, the vl_sift function implements the SIFT algorithm, which identifies and describes key points in images in a way that is invariant to scale, rotation, and translation. After extracting features, we identified the largest scale key point in Cat_1 and subsequently found the nearest neighbor corresponding to Cat_3. Utilizing the vl_ubcmatch function, we determined the SIFT pairs and plotted them. Similarly, this process was replicated for Cat_3 and Cat_2, Dog_1 and Cat_3, and Cat_1 and Dog_1. The vl_ubcmatch function matches these SIFT descriptors between two sets of features, identifying pairs based on the closest Euclidean distance of their descriptors, thereby enabling the detection of similar features across different images.

Bag Of Words: To develop a bag of words model, th e process begins by utilizing the SIFT to identify and extract key features and descriptors from images. This step involves aggregating SIFT features derived from four distinct images into a singular comprehensive array. Following this aggregation, the array, encompassing all extracted features, is subjected to clustering through the K-means clustering algorithm, resulting in the formation of eight distinct clusters. Subsequently, each original feature is associated with its respective cluster. Leveraging this cluster-based categorization, histograms are generated to represent the distribution of features across clusters. These histograms serve as a basis for evaluating and quantifying the level of similarity among the four images, thus facilitating a comparative analysis based on visual content.

XI. Experimental Results:

Cat 1 Vs Cat 3:

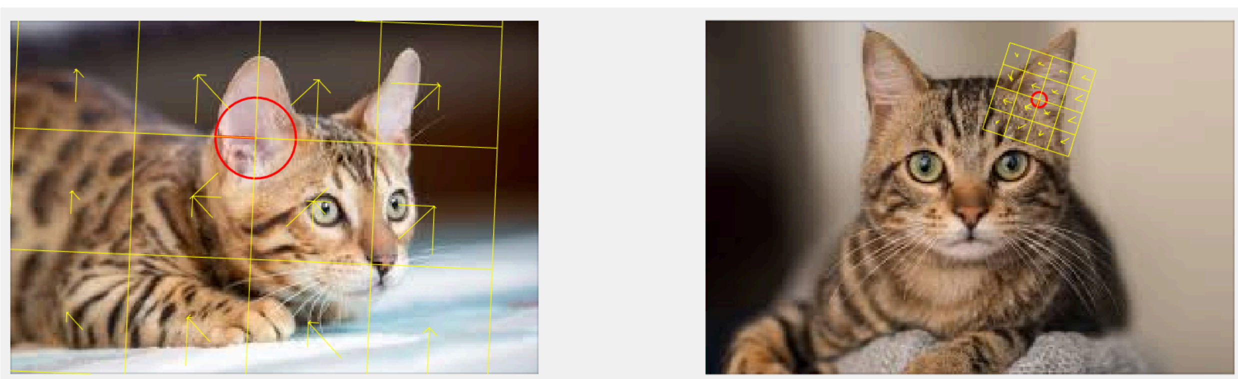


Figure 22: Highest Key Point in Cat_1 and matching point in Cat_3

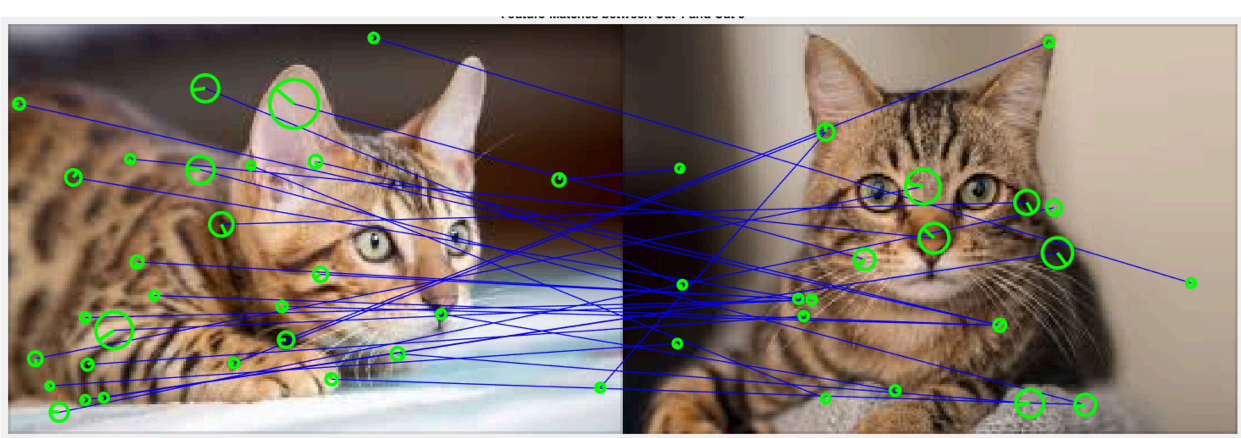


Figure 23: Matching SIFT Pairs Cat_1 and Cat_3

Cat 3 Vs Cat 1:

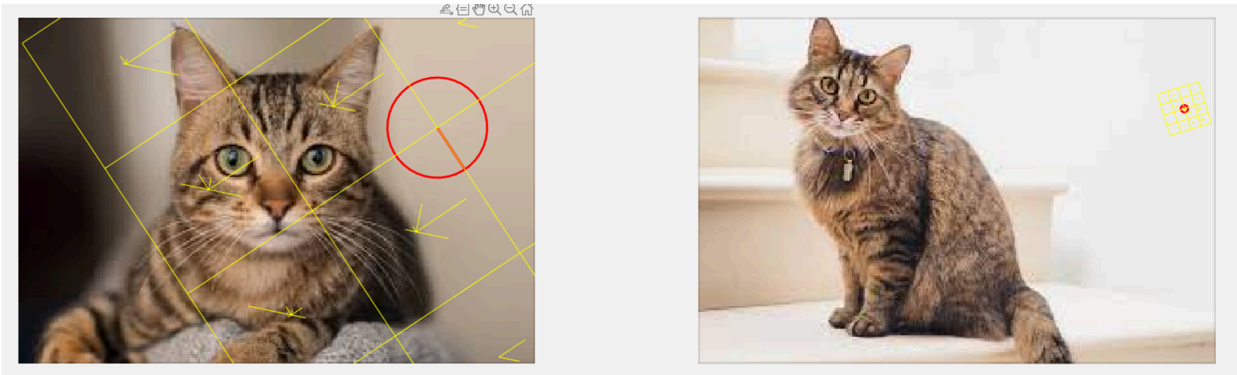


Figure 24: Highest Key Point in Cat_3 and matching point in Cat_2

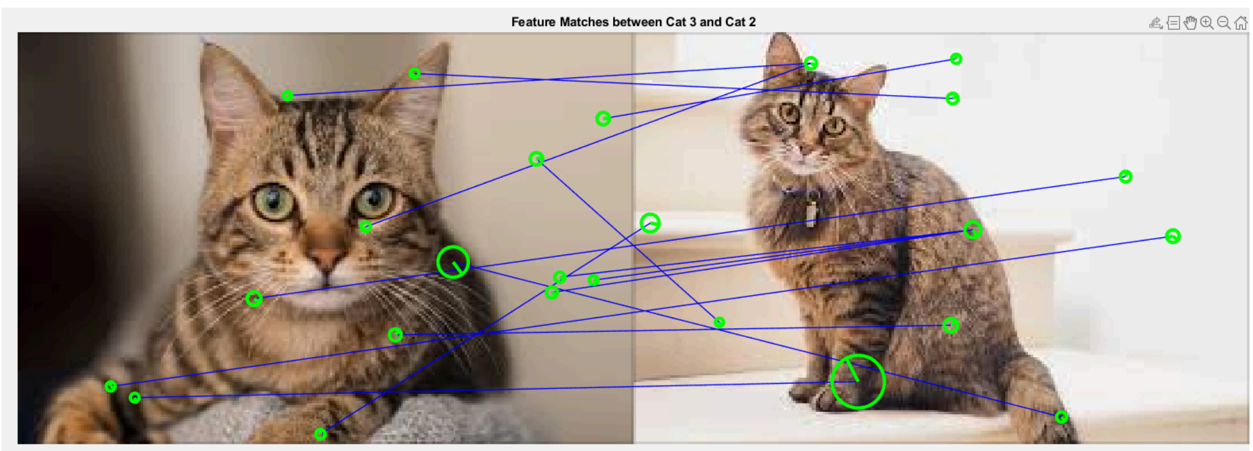


Figure 25: Matching SIFT Pairs Cat_3 and Cat_2

Dog 1 Vs Cat 3:

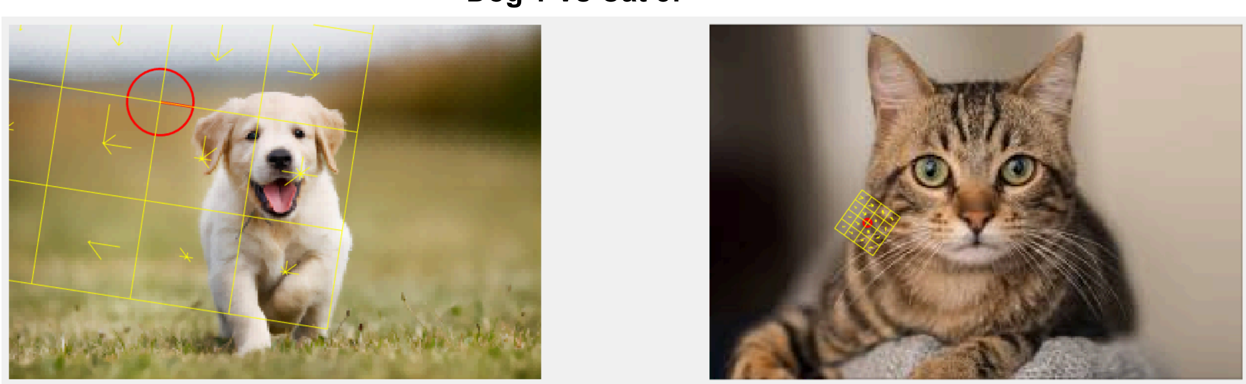


Figure 26: Highest Key Point in Dog_1 and matching point in Cat_3

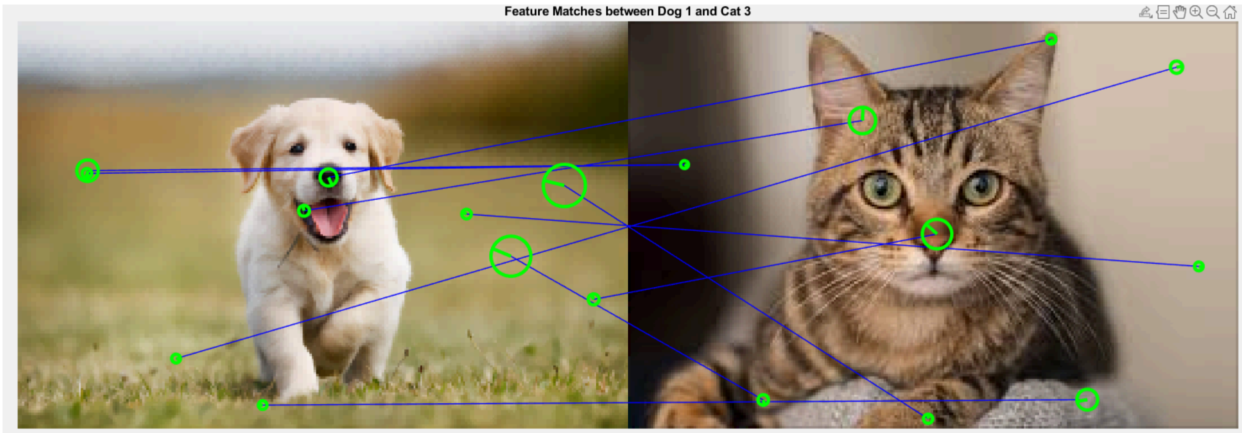


Figure 27: Matching SIFT Pairs Dog_1 and Cat_3

Cat 1 Vs Dog 1:

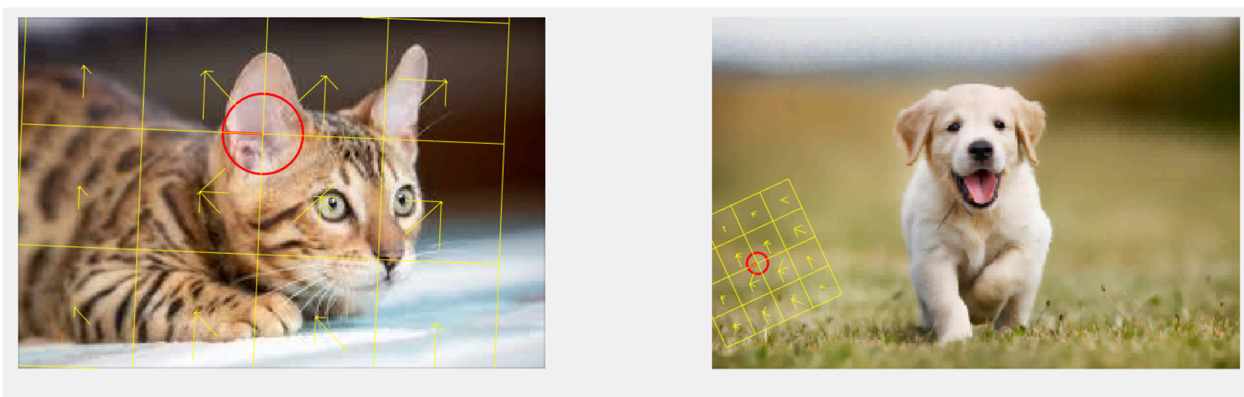


Figure 28: Highest Key Point in Cat_1 and matching point in Dog_1

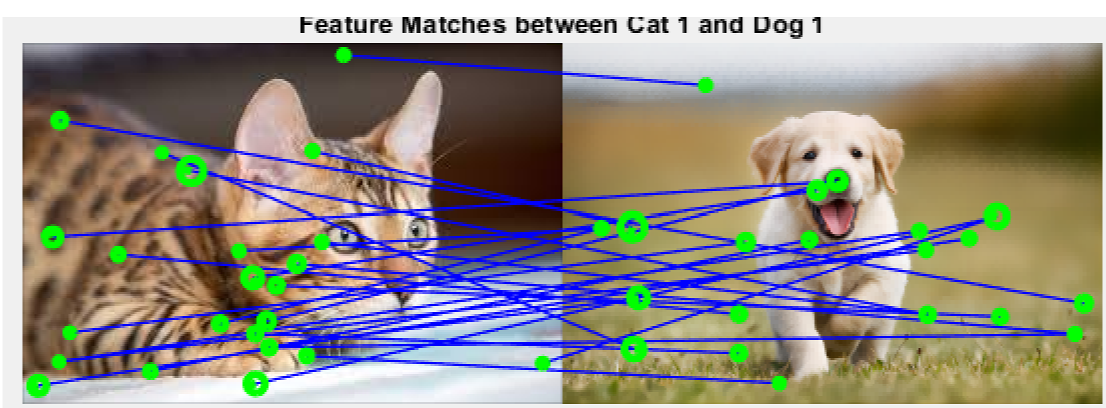


Figure 29: Matching SIFT Pairs Cat_1 and Dog_1

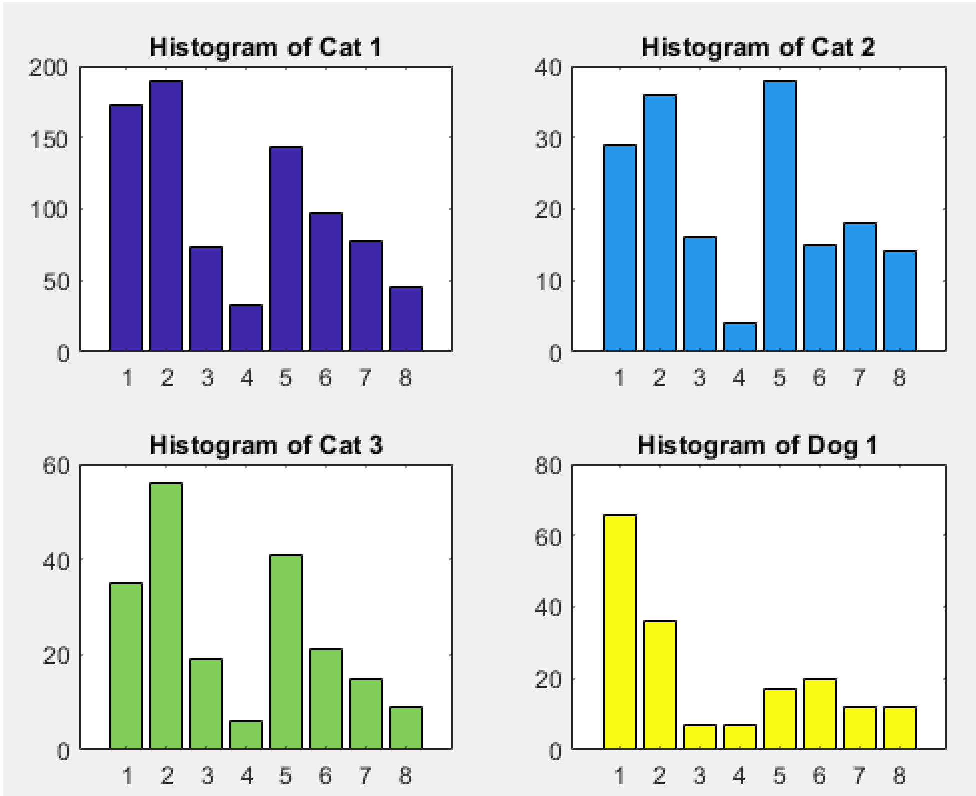


Figure 30: Frequency Histogram

Comparison with Cat3	Cat1	Cat2	Dog1
Similarity Index	37.65	77.128	39.928

XII. Discussion :

1. Salient Point Descriptor:

Q1. From the paper, it can be understood that SIFT is robust to translation, scaling,rotation and projective transformation.

Q2. Translation: SIFT's translation resilience stems from its method of identifying and characterizing key points through local gradient information, regardless of their position in the image. It employs a Difference of Gaussians technique to pinpoint consistent features at various scales, followed by the generation of descriptors derived from gradient configurations surrounding these keypoints. By concentrating on local features instead of their precise locations, the descriptors produced are naturally unaffected by translation.

Scaling: robustness to scaling is achieved by detecting keypoints at multiple scales using a pyramid of Gaussian-blurred images. By comparing these blurred images, stable features across scales are detected.

Rotation: invariance in rotation is by attributing a specific orientation to each detected keypoint, determined by the local image gradient's direction. This process aligns the keypoint descriptor with the predominant direction of the keypoint, rendering the descriptor unaffected by the rotation of the image.

Projective Transformation : As SIFT is invariant with respect to scale and rotation it automatically is robust to projective transformation.

Q3. In the keypoint descriptor phase of the SIFT algorithm, robustness to variations in illumination is significantly enhanced through the analysis and modification of local image gradients around keypoints. This process involves calculating gradients to capture the local structure and then adjusting these measurements to achieve invariance to several factors, including changes in light. A crucial step in this process involves normalizing the gradient-based feature vector to a unit length. Following this, the elements of the vector are capped at a maximum value of 0.2 to mitigate the effect of extreme variations in lighting, before the vector is normalized again to unit length. This series of operations ensures that the descriptor's sensitivity to illumination changes is substantially reduced, enhancing its reliability and stability across different lighting conditions.

Q4. Utilizing the Difference of Gaussian method rather than the Laplacian of Gaussians within the Scale-Invariant Feature Transform (SIFT) algorithm presents significant advantages, particularly in terms of computational efficiency and the effectiveness in pinpointing key points across images. By subtracting Gaussian-blurred images of slightly different scales, the DoG method efficiently mimics the LoG's functionality in detecting interest points but at a fraction of the computational cost. This approach not only speeds up the process but also ensures the algorithm remains effective in large-scale and real-time image analysis scenarios, where rapid processing without sacrificing accuracy is paramount.

Q5. SIFT's output vector has 128 features (size) in its original paper.

2. **Image Matching:**

Question 1 - The experimental results show the comparison between the cats and the dog was mapped. The features were found using SIFT and with the help of VLFeat toolbox. The key point for cat_1 was identified at position '202', located in the vicinity of the ears. Interestingly, upon identifying the closest neighbor in cat_3, it was observed that this point also resided in the ear area, demonstrating a consistent and accurate matching of features. This evidence of similarity in key points across different subjects suggests a degree of reliability in the results, indicating that the SIFT method can effectively identify and match corresponding anatomical features, thus affirming its utility in comparative analyses.

Question 2 - In my observation of the feature mapping of the SIFT pairs, I noticed that the matching pairs for the cat are the key points on the cat itself, whereas for the dog and the cat, the matching pairs usually involve a body point on the cat being mapped to a background point on the dog. Therefore, it can be concluded that cats do not match well with dogs.

3. **Bag of Words:**

Once the bags of words have been constructed and the histograms of the images have been generated, a close observation reveals that all three cats exhibit almost the same histogram shape, whereas the dog displays a distinct histogram curve. Upon calculating the similarity index, it was found that cat 3 is most similar to cat 2. However, this index calculates similarity for each bin, resulting in varied answers. Yet, if we rely on the visual representation of the histograms, it becomes apparent that the cats are more similar to each other than to the dog