Analysis and Evaluation of Feature Detection and Classification Methods for Image Classification & Scene Recognition

100286321
University Of East Anglia

100443924
University Of East Anglia

Abstract

In the ever-evolving field of computer vision, the accuracy of image classification and scene categorisation is of great importance. This report provides a review of existing literature on current methodologies in feature detection and classifier algorithms for image classification. Furthermore, by critically analysing effectiveness of these techniques, we explore the advancements and challenges in the domain. Following the literature review, we apply the theoretical insights gained into practice using the SUN database, which comprises 3,000 images across various scenes, in order to analyse the efficacy of benchmark image classification algorithms, specifically the 'Color Indexing' method by Swain and Ballard, and the 'tiny image' feature. Through systematic testing and modification of algorithm parameters, such as color space and histogram quantisation, as well as adjustments to the k-nearest neighbour classifier, we aim to identify how tweaking these parameters affect scene recognition accuracy. Our findings offer insights into the strengths and limitations of current approaches while identifying factors influencing image categorisation effectiveness..............

Keywords

Computer Vision, Feature detection, Image Classification and categorisation

1 Introduction

The need to classify images and recognise scenes with high accuracy and efficiency is crucial within the domain of computer vision. Various applications, such as autonomous driving, medical imaging, environmental monitoring, and security surveillance heavily rely on these processes [Szeliski 2022]. Image classification involves assigning labels to images based on the content or primary objects they contain [LeCun et al. 2015], while scene recognition encompasses a broader perspective of identification and characterisation of image based on the context depicted. This not only includes recognising individual elements but also understanding their collective arrangement and overall ambiance they create, thereby providing a comprehensive label for the scene as a whole. For example, distinguishing a forest containing trees from a beach containing coconut trees or a kitchen with a dining table from a living room with a coffee table [Zhou et al. 2017]. These process involve detection and interpretation of patterns within the images, which continues to pose a significant challenges within the academic community.

In this paper we concentrate on two vital elements of image classification and scene recognition: feature detection and classifiers. Feature detection involves identifying information (such as points, edges, or objects) from images for differentiating one image from another [Lowe 2004], and classifiers apply algorithms to assign categorical labels to images based on the detected features [Bishop and Nasrabadi 2006]. By exploring fundamental and recent works, the review aims to delve into the methodologies, and effectiveness of various feature detection and classification techniques. This review looks at foundational theories and modern advancements, comparing different methods, and delves into their potential challenges and drawbacks in real world scenarios.

Subsequently, the report delves into applying Color Indexing method described by Swain and Ballard (S&B) on the SUN database. This section analyses how algorithm parameters can be varied to try and arrive at the best overall classification performance.................

2 Literature Review

2.1 Feature Detection

As previously discussed, feature detection involves identifying information (features) such as edges, corners, and textures from an image. Once, relevant features are detected, descriptions are created in a way that allows for efficient matching and recognition. Features serve as the starting

ACM, Computer Vision,
2024. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

point for computer vision algorithms, and the choice of features significantly impacts the effectiveness of subsequent algorithms used for classification [Lowe 2004; Mikolajczyk and Schmid 2005]. This section reviews key developments, methodologies, and impacts of feature detection, providing insights into its evolution and application in current research.

2.1.1 Early Approaches and Evolution:

Early work in feature detection can be traced back to methods focusing on extracting simple attributes like edges and shapes. Canny [1986] work on edge detection, has been fundamental for outlining boundaries within images, this algorithm has been instrumental for identifying shapes and orientations. While methods such as these have been effective in structured environments, they struggle with real world complex scenarios, as they have limited ability in capturing variance in texture and is prone to noise. Infact **Canny's algorithm** can be fooled by noise caused from low light or compression, which can lead to false edges or blur real ones, affecting accuracy. This is because of the algorithms tade-off between sensitivity (the ability to detect true edges) and specificity (the ability to reject noise and false edges), and finding the right balance between these two factors can be challenging. Furthermore, the algorithm involves multiple steps such as Gaussian smoothing, gradient calculation, suppression, and hysteresis thresholding. Each of these steps incurs computational overhead, making the algorithm relatively slow when using it for very large images or real-time applications [Muthukrishnan and Radha 2011].

SIFT: As the field progressed with better computational abilities, and the need to develop more robust detectors being capable of handling challenges of diverse image such as, changes in lighting, scale and rotation. Approaches like Scale-Invariant Feature Transform (SIFT) by Lowe [2004] marked a significant milestone. SIFT identifies keypoints/features that are distinctive and remain invariant across scale, rotation, and illumination changes. Unlike Canny's agorithm that focuses on edges, which can change significantly with scale and orientation, SIFT excels at finding features that remain consistent. Moreover, SIFT not only detects keypoints but also generates a descriptor capturing the local gradient around the keypoints. This capability greatly enhances the task of image matching and object recognition, setting a new standard for feature detection methods.

The SIFT algorithm can be summarised into 4 basic steps. First is to use the Difference of Gaussian (DoG) to find potential keypoints based on variation in in image intensity. Second, it refines these initial detections, eliminating unreliable ones with low contrast. Third, is to assign specific orientation to remaining keypoints based on the surrounding image's gradients. Finally, it generates a unique descriptor for each keypoint, capturing the local image properties like gradient magnitude and direction. This descriptor essentially acts as a fingerprint for the key point, greatly enhancing detection and matching tasks. However, similar to Canny's algorithm, all of these steps are computationally complex, which is a major drawback especially for real-time applications.

2.1.2 Further advancements and Analysis:

Following SIFT, a whole host of feature detector and descriptor algorithms emerged, such as Speeded Up Robust Features (SURF) [Bay et al. 2006] and Oriented FAST and Rotated BRIEF (ORB) [Rublee et al. 2011], that improved upon computational efficiency, robustness, accuracy and speed of SIFT. SURF provided a fast scale-and-rotation-invariant alternative to SIFT, which was the most computationally intensive component of SIFT, and ORB further accelerated feature detection by combining modified FAST detection and BRIEF descriptors, making it suitable for real-time applications by improving speed without substantial loss in performance.

SURF: (Speeded-Up Robust Features) uses box filters to approximate the DoG, using integral images to compute over different scales more efficiently. The point of interest is identified using Hessian matrix-based BLOB detector, that finds orientation on the basis of wavelet responses in the horizontal and vertical direction using Gaussian weights. The feature descriptor divides the neigbourhood around a keypoint into subregions and uses wavelet responses for representation. This increases the robustness of the descriptor. SURF also uses the sign of the Laplacian to differentiate bright blobs on dark backgrounds and vice versa, thus allowing for faster feature matching by only comparing features of the same contrast type [Ke and Sukthankar 2004].

ORB: (Oriented FAST and Rotated Brief) combines the FAST keypoint detector with the BRIEF descriptor, to enhance the performance [Calonder et al. 2010]. ORB selects top key points using the FAST method, refined with the Harris corner measure for corner detection. Subsequently, since FAST is a rotation variant and only returns a rotation, ORB computes orientation using the intensity centroid of the patch around the corner. The vector from the corner to the centroid gives orientation in this process, hence improving on invariance. To overcome with BRIEF's in-plane rotation limitations, ORB applies a rotation matrix based on the orientation of each patch and steers the BRIEF descriptors so that the consistency is retained despite the rotation.

Comparative analyses have been very critical in understanding the strengths and shortcomings of these descriptors. Mikolajczyk and Schmid [2005] laid down a foundational performance evaluation criteria that showed how different factors, including repeatability and robustness, affect the utility of feature detection algorithms under varied image transformations. Their work has been influential in highlighting the trade-offs between computational demand versus speed and accuracy, and provides a guide to selection of an appropriate feature detector based on the on extraction requirements of an application.

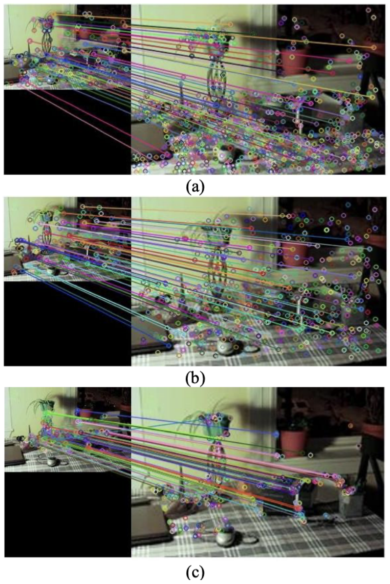
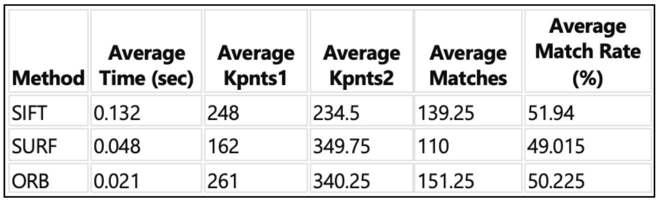


Figure 1: The matching of the original image with its scaled image using: (a) SIFT (b) SURF (c) ORB. Photograph by [Karami et al. 2017], via Proceedings of the 2015 Newfoundland Electrical and Computer Engineering Conference

Furthering this discussion, Karami et al. [2017] provide a comparative study on "Image matching using SIFT, SURF, BRIEF, and ORB: Performance comparison for distorted images," which provides a direct comparison between these method for different kinds of distortions, such as scaling, rotation, and noise. Their results as seen in table 2, show that SIFT generally remains the superior by providing the highest matching rates, while ORB is the quickest due to its computational efficiency. Both ORB and SURF outperform SIFT in computational speed a better option for task that require fast and real time processing, however with some trade-offs in matching performance under harsh distortions. Moreover, figure 2 shows how the descriptors work, in ORB the features tend to be concentrated around objects located at the center of the image. On the other hand, in SURF and SIFT the features are more evenly distributed across the entire image. These assessments highlight how appropriate feature detectors and descriptors, can lead to more refined and tailored implementations in many applications based on their requirement.



Method	Average Time (sec)	Average Kpnts1	Average Kpnts2	Average Matches	Average Match Rate (%)
SIFT	0.132	248	234.5	139.25	51.94
SURF	0.048	162	349.75	110	49.015
ORB	0.021	261	340.25	151.25	50.225

Figure 2: Averaged performance comparison of SIFT, SURF, and ORB for under various distortions. Results are averaged across scaling, shearing, fish-eye distortion, and salt and pepper noise. Data by [Karami et al. 2017], via Proceedings of the 2015 Newfoundland Electrical and Computer Engineering Conference

2.1.3 Scene recognition, Beyond bag of Features and Spatial Pyramid matching:

These traditional methods we have discussed for feature detection have advanced image classification tasks, however they focus on detecting individual features without considering their spatial relationships within the image. This approach is also known as the 'Bag of Features' (BoF) model, and it struggles with scene recognition tasks where the arrangement and context of features play a critical role [Sivic and Zisserman 2003]. For instance, an image of a beach might contain similar individual features to an image of a desert (such as sand), but the spatial arrangement of these features (such as the presence of water, sky, and horizon) helps distinguish between the two scenes.

To address these limitations, the "Beyond Bags of Features" concept incorporates spatial relationships among features. A key advancement in this domain is Spatial Pyramid Matching (SPM), proposed by Lazebnik et al. [2006], extends the BoF approach by partitioning the image into multiple, hierarchical grids, and computing feature histograms for each segment, preserving spatial information. By considering features at multiple levels of granularity and maintaining their spatial hierarchy, SPM significantly enhances the descriptive power of feature representations, leading to more accurate and robust scene categorisation.

This paves the way for more sophisticated and contextually aware computer vision solutions.

2.1.4 Challenges and Integration with Deep Learning:

The transition from traditional feature detection to its application in scene recognition highlights the importance of contextual understanding within images. Despite the development in this field, there remains challenges for traditional feature extraction methods with handling occlusion (when an object is hidden or blocked from view by another object), or due to sensor limitations, and dynamic changes of the scene [He et al. 2016a; Zhou et al. 2014].

In response, the integration of feature detection with deep learning, particularly through Convolutional Neural Networks (CNNs), has revolutionized scene recognition by automating feature extraction and enabling adaptation to diverse visual data [Krizhevsky et al. 2017]. This paradigm shift towards CNNs has merged feature detection and classification into a one cohesive framework, significantly improving scene recognition tasks. As we delve into classifiers, we will further explore the pivotal role of CNNs, which excel in both classification and feature extraction, thereby addressing intricate challenges in scene recognition and exploring new benchmarks algorithms in computer vision.

2.2 Classifiers

Classifiers are the main decision tools involved in categorising images into specific classes or scenes based on the extracted features, thereby facilitating image understanding and interpretation. Their application extends across various domains, from basic image classification to complex scene recognition tasks. This section reviews classifiers for image classification and scene recognition, explores traditional and contemporary approaches, and addresses inherent challenges.

2.2.1 Traditional Classifiers

k-Nearest Neighbors (kNN):

This is a non-parametric method of classifying images based on the majority labels among the k closest points in the feature space. However, though intuitive and simple, the method performs poorly in high-dimensional data classification as found in image processing [Cover and Hart 1967]. This algorithm will be furthered discussed in greater detail in the next section where we implement tiny images and colour histogram feature extractor with KNN for scene recognition.

The K-means classifier:

partitions a dataset into k distinct clusters to minimise within-cluster variances, aiding significantly in image segmentation [Kanungo et al. 2002]. The algorithm assigns each data point iteratively to the nearest cluster center, updating the centers based on the current assignments until convergence. This method, was further enhanced by the K-means++ technique proposed by Arthur and Vassilvitskii [2007] for ensuring efficient selection of initial centers improving the quality of clustering and also reducing the computational time. However, the biggest disadvantage of the K-means algorithm is its sensitivity to the initial placement of cluster centers and its tendency to produce poor clustering results with non-spherical data distributions with varying cluster sizes.

Support Vector Machines (SVM):

SVM's is principled upon identifying the hyperplane that best separates the different classes in a feature space. It has been proven efficient for binary classification and is extended for multi-class problems. The main characteristics of SVMs are that they generalise very well and are highly robust to overfitting, provided there is a clear margin of separation in high-dimensional spaces. However, it is sensitive to parameter tuning, requiring the selection of an appropriate kernel and adjustment of its parameters which can be computationally quite heavy if large image datasets are used [Cortes and Vapnik 1995]. The choice of these parameters significantly affects performance leading to varying classification accuracy and generalisation ability. Despite these challenges, the impact of SVMs in image classification has been profound. For instance, Chapelle et al. [2002] showcased the use of SVMs in digit and object recognition, highlighting the model's robustness and efficiency. Similarly, in the domain of face recognition, Guo et al. [2001] demonstrated SVMs' capability to distinguish between faces and non-faces with high accuracy, leveraging the high-dimensional feature space typical of facial images. Subsequently, landmark studies by Csurka et al. [2004] used SVM's with bag-of-features models in the recognition of scenes and objects, which had great results and has been used as a benchmark for other studies. Their work underscored SVMs' ability to effectively handle diverse and complex image data.

Decision Trees and Random Forests:

Decision trees and random forests apply simple decision rules computed from feature data. While decision trees derive their rules from individual features, random forests build an ensemble of them. Ensemble methods mitigate overfitting, which is highly relevant for image data space as it improves overall classification accuracy [Breiman 2001; Liaw and Wiener 2002]. The efficacy of Random

Forests in handling the high-dimensional nature of image data lies in their inherent diversity. Through training on the random data subsets and features, these models are more resilient noise and variability. Additionally, randomness provides protection against overfitting making these models well suited for image analysis [Liaw and Wiener 2002]. Belgiu and Drăguţ [2016] and Rodriguez-Galiano et al. [2012] have demonstrated the enhanced performance of Random Forests in remote sensing and land-use classification. Their studies reveal that Random Forests efficiently manage the complex, high-dimensional spatial and spectral data characteristic of satellite imagery and aerial photography. Specifically, they mention the algorithm's ability to discern various land cover types by analysing the importance of different spectral bands, which is crucial for accurately categorising land areas. Additionally, Rodriguez-Galiano et al. [2012] highlighted the improvement in classification accuracy when incorporating spatial data, which aids in resolving ambiguities in areas where land uses are spectrally similar but spatially distinct. This added detail underscores the capability of Random Forests in leveraging both the spectral and spatial dimensions of remote sensing data to enhance image classification outcomes.

Naive Bayes Classifier (NBC):

It is a simple but effective classifiers known for its efficiency, particularly in classification of text and preliminary analyses of images. It relies on the assumption of feature independence, and therefore is a computationally efficient approach, however it less effective to deal with complex image data due to its nature of interdependent features [Domingos and Pazzani 1997]. Nevertheless, NBCs have found use in basic image classification tasks where high-dimensional data does not seriously violate the feature independence assumption. While this assumption rarely holds in real images due to the spatial correlations, NBCs can still provide baseline performance and are computationally efficient, making them suitable for initial analyses or applications with limited computational resources where speed is prioritised over accuracy [Rish et al. 2001]. NBC has been used for tasks such as face recognition, while NBC provides a valuable baseline for these object detection tasks, scene recognition typically necessitate more complex approaches that require spatial information from scenes where features are not independent [Maturana et al. 2009].

2.2.2 Deep Learning Approaches:

Deep learning represents a paradigm shift in image classification and scene recognition. Unlike traditional methods that rely on specific feature extractors and ML classifiers, deep learning algorithms learn hierarchical feature representations directly from data enabling models to automatically discover the representations needed for feature detection and classification from raw images [LeCun et al. 2015]. As previously mentioned Deep learning integrates feature extraction and classification into a unified framework, particularly Convolutional Neural Networks (CNNs), automate this process by learning optimal features during training and using these learned features for classification. This end-to-end learning significantly enhances the model's ability to handle complex visual data and improves overall accuracy in scene recognition tasks [Krizhevsky et al. 2017].

Convolutional Neural Networks (CNN):

CNN lie at the heart of the current deep learning revolution, particularly in image classification and scene recognition. Their structure comprises of multiple layers that automatically facilitate feature extraction through applying convolutional filters to input data, thereby not requiring manual engineering [LeCun et al. 2015]. The architecture of CNNs is designed to mirror the human visual system processers, where layers detect simple features like edges and textures, and higher layers interpret more complex features like shapes or objects [He et al. 2016a]. The layers include: Convolutional Layers: Utilise learnable filters to scan the input image, creating activation maps that highlight detected features like edges and patterns, optimising feature detection through training [Krizhevsky et al. 2017]. Pooling Layers: Reduce data dimensions and computational load, enhancing feature invariance to scale and orientation, and mitigating overfitting risks [Scherer et al. 2010]. Fully Connected Layers: Integrate all features, linking them to specific image classes, and use a softmax function in the final layer to determine class probabilities, enabling precise image classification [Szegedy et al. 2015].

Region-based Convolutional Neural Networks (R-CNNs):

Region-based Convolutional Neural Networks (R-CNNs) are a big leap in deep learning for object detection and scene recognition in static scenarios. Unlike conventional CNNs that process the whole image at once, R-CNN focuses more on identifying and analysing different regions of the image, hence is particularly suitable to perform tasks like object detection and understanding complex scenes. R-CNN runs through a set of stages, first, it uses the selective search algorithm to obtain potential object region proposals, which are then resized and fed forward into a CNN, which extracts features within that region and passes the features to a set of classifiers (typically SVMs) designed to determine presence and class of objects within them. Moreover, the approach leverages a regression model to refine the detected bounding boxes of the objects for localisation and accuracy [Girshick et al. 2014].

While the original R-CNN framework marked a breakthrough in object detection, its running was grossly inefficient, it ran the CNN over each region proposal independently making it not best for real time videos. Subsequent improvements in both Fast R-CNN and Faster R-CNN versions have addressed these issues by making the architecture lightweight—most importantly, including the region proposal mechanism in the network. This makes a huge stride in processing speed, reaching near real-time object detection [Ren et al. 2015]. Together, R-CNNs and the later applications combined have come to offer a really strong solution to these problems with extremely detailed processing of images and recognition of scenes, suitable from object localization and recognition in autonomous driving to surveillance. Collectively, R-CNNs and their evolved versions provide a robust solution for detailed image analysis and scene recognition, excelling in applications requiring accurate object localisation and identification, such as autonomous driving to surveillance.

.

3 Methodologies, Experiments, and Results

In Image classification, we employ Tiny image feature extraction for compact representations and Color Histogram analysis to quantify color distributions across predefined No. of bins, enhancing classification accuracy.

3.1 Tiny image feature extraction

It involves resizing images to a small fixed size, followed by vectorization. This compact representation aids in fast feature comparison and recognition tasks [Torralba et al. 2008]. For the Tiny image, the following parameters:
Output size specifies an image's pixel dimensions, which are important for custom resolutions in applications such as web design and mobile apps. Resizing to a smaller output size can improve loading speed but may affect image detail [He et al. 2016b].
Color spaces are fundamental in image classification, offering distinct representations for color information.
RGB is a color model for digital images, representing each pixel's intensity of red, green, and blue light [Krizhevsky et al. 2012].
LAB (CIELAB) color space, as defined by CIE, offers perceptually uniform color representations through lightness (L) and two color-opponent dimensions (A and B) and is widely employed in color correction and image editing for its intuitive adjustment capabilities owing to its perceptual uniformity [Sharma et al. 2005]
YCBCR is a color space optimized for digital imaging and video compression. It separates luminance (Y) from chrominance components (Cb-Chroma-blue and Cr-Chroma-red), maximizing bandwidth for color transmission [Gonzalez and Faisal 2019].
CMYK color model in color printing uses four ink colors (cyan, magenta, yellow, and black) to produce a wide range of colors [Sharma 2003].
Grayscale Conversion approaches entail decreasing image color depth to grayscale, emphasizing luminance or brightness, and retaining as many perceptual details as feasible. This method frequently includes a weighted sum of the RGB values or complex algorithms that incorporate human perception to better maintain the contrast and details present in the original color image. [Zeger et al. 2021].
Lastly, **HSV** color model is favored for color texture categorization over the RGB color space because it more accurately describes color details, intensity, and brightness. The mathematically thorough conversion from RGB to HSV demonstrates its ability to characterize the hue plane for feature extraction, which is necessary for correct image analysis and categorization [Chang et al. 2010].
Quantization levelspecifies the amount of intensity/color levels that an image can have. Lowering it decreases image size for storage and transmission, but it may cause quality loss and artifacts such as banding [Zhou et al. 2018].
Normalization modifies the range of pixel intensity levels, which is important in image processing for uniform data comparison. It ensures that input photos have the same influence on machine learning and computer vision models [Deng and Yu 2014].
We created a **Tiny img feature extraction function** with adjustable parameters to extract features from tiny images. The options include output sizes ranging from 2x2 to 256x256, color spaces such as RGB, LAB, HSV, YCBCR, grayscale, and CMYK, quantization levels of 0,4, 8, 16, and 32, and normalization enabled. Using this function, we can analyze how different parameters affect image classification accuracy, allowing for informed decision-making and the development of feature extraction methods.

3.2 Color histogram feature extraction

From the research paper [Swain and Ballard 1991], color histogram provides a concise representation of color distribution in images, making it useful for content analysis, retrieval, and categorization. The Number of Bins and Distance Matrix are important parameters for extracting features from color histograms, in addition to Color Space and Normalization, as explained in Tiny image feature extraction. Color histogram-based approaches in computer vision are resilient and efficient, providing vital insights into image features and facilitating automated analysis and interpretation.

No. of Bins: Color representation granularity in a histogram is determined by the number of bins. Adding more bins increases information but requires additional processing [Zhang and Lu 2016].

Distance Matrix: Measures color histogram similarity, which is important for image retrieval. The metrix used (Euclidean, Manhattan, or Chi-square) impacts performance [Ahmed et al. 2019].

To address this, we created a **Color Histogram color space feature extraction** function with customizable settings. The settings include the number of bins (4-512), color spaces (RGB, LAB, HSV, YCBCR, grayscale, and CMYK), and normalizing set to true and for Distance matrix -Eucidenan distance by pdist2, Matlab's default library. Using this function, we can analyze how different parameters affect image classification accuracy, allowing for informed decision-making and feature extraction optimization.

3.3 Nearest Neighbour Classifier Function

In addition, we developed a nearest-neighbor classification function to determine the maximum accuracy. This classifier uses the largest value of k as input. It then iterates over all odd numbers from 1 to the provided maximum k, calculating accuracy for each k and saving the results as a k accuracy table. For example, if the maximum k value is 21, it checks and records the accuracy of 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, and 21. The function also saves the best accuracy and its matching k value to the workspace, which is presented in the command line.

3.4 For Tiny Image: Experiments and Test results

Based on our experiments with Tiny image feature extraction using the nearest neighbor classifier, we discovered considerable differences in classification accuracy across different parameter settings, which are consistent with previous research findings.

In the RGB color space, we got the highest accuracy of 34.6% using a [4x4] output size with no quantization. This conclusion aligns with [Torralba et al. 2008], which highlights the significance of compact representations in improving classification accuracy.

In the LAB color space, we achieved the maximum accuracy of 31.3% with an output size of [8x8] and no quantization.

Our experiment in the YCBCR color space achieved a maximum accuracy of 33.5% with an output size of [4x4] and a quantization level of 32.

In contrast, experiments with grayscale images presented challenges, with accuracies ranging from 21.3% to 10.1%.

In the HSV color space, we achieved a maximum accuracy of 34.6% using a [4x4] output size and no quantization.

Finally, our testing in the CMYK color space yielded the second-highest accuracy of 34.5% with an output size of [4x4] and no quantization.

Continuing from the previous experiment results, we maintained two parameters constant across tests to ensure a concentrated investigation of the impact of color space selection and k-value change on classification accuracy. We kept the quantization level (0) and output size unchanged (specified values for each color space). This method allowed us to isolate the impacts of other variables, reducing confounding factors and confirming that observed differences in accuracy were due solely to variances in color space and k-value.

Figure 3 Line Plot for Comparing Accuracies Across Color Spaces with Output Sizes

- This line graph compares accuracies among color spaces at a quantization level of 0 and k-value of 19, with varying output sizes.
- Each line represents a color space, allowing for a visual comparison of accuracies across different output sizes within that space.
- The graph isolates the impact of output size change on accuracies within specific color spaces, allowing for targeted study and decision-making.

Findings from the figure 3 describe as:

The RGB and HSV color spaces have the best accuracies near (0.35) with a [4x4] output size, whereas YCBCR has consistently good accuracies (0.3 to 0.325). Grayscale consistently produces the lowest accuracies (0.1 to 0.2). LAB has good accuracy (0.3) at [4x4], but declines with bigger output sizes. This demonstrates the effect of color space and output size on classification accuracy in image processing.

Figure 4: Line Plot for Comparing Accuracies Across

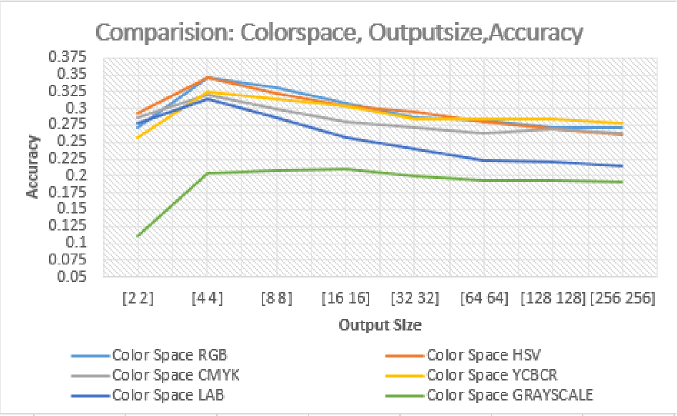


Figure 3: Comparison between Color spaces, Output Sizes, Acucracies

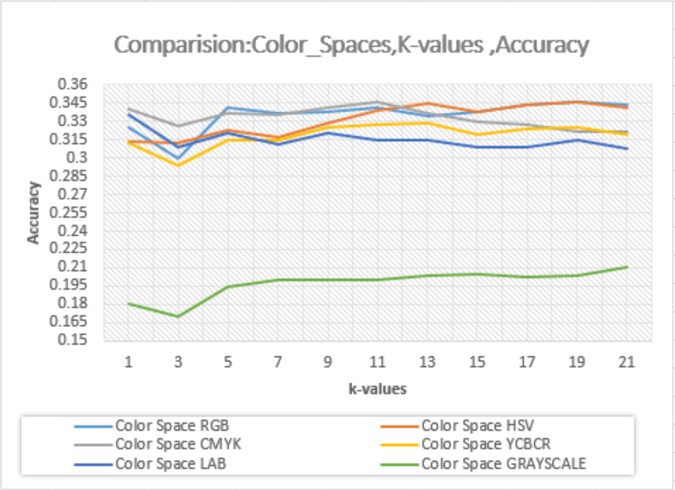


Figure 4: Comparison between Color space, K values, accuracies

Color Spaces with K-values

- This line plot shows how accuracies vary in different color spaces with a fixed output size of [4x4] and a quantization level of 0.
- Each line represents a separate color space, allowing for direct comparison of accuracy trends with varying k-values.
- Consistent quantization level and output size isolate the impact of k-value fluctuation on accuracies within each color space, allowing for focused analysis and decision-making.

Finding from the figure 4 describe as:
Both RGB and HSV color spaces reach the highest accuracy (0.346) at k-value 19. At k-value 3, RGB color space accuracy approaches 0.3, surpassing HSV's 0.315 accuracy.

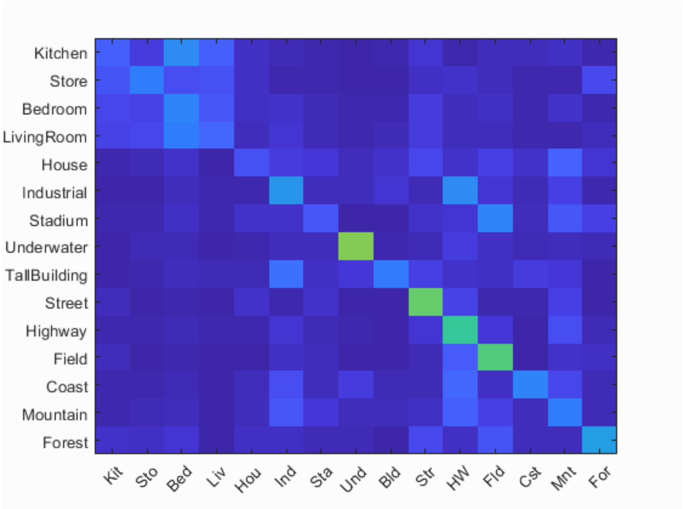


Figure 5: Tiny Image Confusion Matrix

Figure 3 shows that grayscale begins with the lowest accuracy (0.18), decreases to around 0.165, and then climbs to around 0.21. Other color spaces have accuracies ranging from 0.3 to 0.345 for various k-values. This investigation shows how classification accuracies vary among color spaces and k-values.

Combining the conclusions from both figures 3,4:
RGB and HSV color spaces consistently achieve high accuracy near to (0.35) with a [4x4] output size, quantization level 0, and k-value 19, while YCBCR maintains good accuracy (0.3 to 0.325) under comparable settings. Grayscale consistently produces the lowest accuracies (0.1-0.2) in both instances. LAB has fair accuracy (0.3) at [4x4], quantization level 0, but decreases with increasing output sizes. Other color spaces range from 0.3 to 0.345 across various k-values. This highlights how color space, output size, and k-value affect classification accuracy in image processing tasks.

Finally we cross-validate our experiment with **confusion matrix as seen in figure 5.**

Confusion Matrix for Tiny image Interpretation:
This confusion matrix uses shading to indicate how many times a classification model's predictions match the actual labels. True positive predictions are often shown diagonally, with the anticipated class matching the true class. Areas in this matrix that match to Underwater (both predicted and true class) are lighter, indicating a higher number of true positives for that class.

In contrast, the dark squares in the non-diagonal positions indicate a lesser number of instances when the predicted class did not match the true class, also known as false positives. For example, the dark square at the junction of Forest (true class) and Street (predicted class) indicates that the model mistakenly identified a Forest as a Street in few or no instances.

The general pattern indicates that the model works well in some classes but confuses particular situations, such as Mountain, Forest, and Coast, which have lighter squares off the diagonal.

3.5 For Color Histogram: Experiments and Test results

In our examination, we experimented with numerous parameter combinations across color spaces. Our findings are consistent with previous studies in the field, demonstrating the usefulness of specific color spaces for image classification tasks.

In the RGB color space, we observed promising results with the highest accuracy achieved at 0.315, accompanied by a k value of 15 and 8 bins.

Moving to the LAB color space, our experiments revealed a peak accuracy of 0.247 with a k value of 21 and 4 bins.

In the HSV color space, we achieved a maximum accuracy of 0.356 with a k value of 15 and 8 bins.

Exploring the YCBCR color space, our analysis yielded a peak accuracy of 0.229 with a ’k’ value of 7 and 8 bins.

Lastly, in the CMYK color space, we obtained a maximum accuracy of 0.302 with a ’k’ value of 7 and 32 bins.

Our methodical experimentation advances our understanding of parameter optimization in image classification, drawing on prior expertise in the field. From the above experiments, we further created two graphs as follows

Figure 6 Line Plot for comparing accuracies across Color Spaces with Number of Bins (Constant: K-value = 15) Figure 6 illustrates how the number of bins influences image categorization accuracy across color spaces while maintaining a constant k value. Optimizing image classification methods requires identifying the best color space for a specific number of bins. It can help pick parameters for classification jobs by identifying color spaces that are more sensitive to variations in the number of bins.

Finding from the figure 6 describe as:

HSV initially has the highest accuracy around 0.350 with 8 bins but experiences a decrease with more number of bins.

RGB and CMYK show accuracies ranging from 0.29 to 0.20, decreasing with an increase in the number of bins.

YCBCR consistently demonstrates the lowest accuracy, especially noticeable at 512 bins.

These findings highlight the necessity of selecting the right number of bins for each color space in order to maximize image classification accuracy.

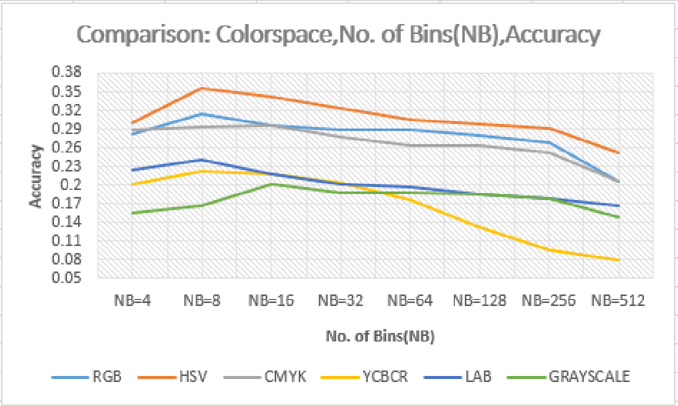


Figure 6: Comparison: Color space, No of Bins(NB), accuracies

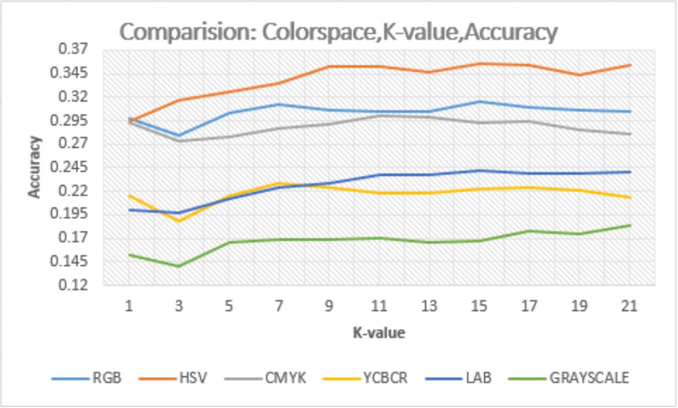


Figure 7: Comparison: Color space, K-value,Accuracy

Figure 7 Line Plot for Comparing accuracies across Color Spaces with K-values (Constant:Number of Bins = 8).

Figure 7 shows how the k value affects image categorization accuracy across different color spaces while keeping the number of bins constant. Observing the performance of multiple color spaces with varying k values allows us to discover which is more resilient or susceptible to cluster number changes. This information helps determine the optimal k value for each color space, improving classification accuracy and computational efficiency.

Findind fron the figure 7 describe as:

- HSV Color Space has the highest accuracy, just above 0.345, with a k value of 15. Accuracy gradually improves from k values 1-21, surpassing other color spaces.
- RGB and CMYK Color Spaces: Accuracy increases with k values, but decreases slightly near k = 21. The trend follows that of HSV, with a little decrease at higher k values.
- YCBCR and LAB color spaces follow the same trend as RGB and CMYK, with accuracies increasing with k values. However, they have poorer accuracies than HSV, RGB, and

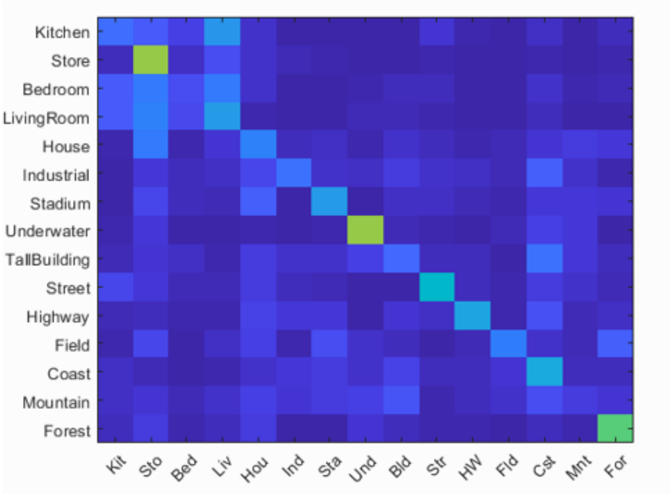


Figure 8: Color Histogram Confusion Matrix

CMYK.

• Grayscale Color Space: Initially displays the lowest accuracies for various k values. However, accuracies continuously grow from the initial k value until k = 21.

These findings highlight the efficiency of several color spaces in image classification tasks, with HSV demonstrating the highest accuracy and a clear trend of improvement with higher k values. However, RGB and CMYK exhibit similar tendencies, although YCBCR and LAB fare quite poorly. Furthermore, the progressive increase in grayscale accuracy indicates that it has the ability to improve with larger k values.

Combining the conclusions from both figures 6,7: In conclusion, parameter optimization is critical for image classification. HSV works best with 8 bins at first, but falls as the number of bins increases, whereas RGB and CMYK exhibit decreasing accuracy. YCBCR constantly delays, and grayscale could improve with greater k settings. Tailored parameter selection is critical for improving accuracy across color spaces.

Finally we cross-validate our experiment with **confusion matrix as seen in figure 8**.

Confusion Matrix for Color histogram Interpretation:

The presented confusion matrix depicts an image recognition model's classification performance across multiple environmental categories.

Key observations include a high rate of correct predictions for 'Underwater' images, indicating that the model is particularly effective at identifying this category.

However, there is notable confusion between 'Mountain' and 'Forest' categories, suggesting that the model struggles to distinguish between these two classes.

Other environments like 'Kitchen', 'Store', and 'Bedroom' show moderate levels of correct classification.

Overall, while the model shows competence in certain areas, the misclassifications between similar environmental categories highlight potential areas for model refinement.

4 Conclusion

Our analysis of image classification methods found that using Color Histograms with the HSV color space resulted in the highest accuracy (35.6%). The peak performance was observed with a k-value of 15.

In comparison, the Tiny Image technique achieved the highest accuracy of 34.6% utilizing the RGB color space, with a [4x4] output size and no quantization. Tiny Image gives a small representation, however it may not capture color changes as well as Color Histograms.

Based on our findings, the Color Histogram technique using the HSV color space appears as the best option for image classification jobs that need exact color differences. This emphasizes the need of using proper feature extraction methods and color spaces to improve classification accuracy in a variety of applications.

References

Eman Ahmed, Alexandre Saint, Abd El Rahman Shabayek, Kseniya Cherenkova, Rig Das, Gleb Gusev, Djamila Aouada, and Bjorn Ottersten. 2019. A survey on Deep Learning Advances on Different 3D Data Representations. arXiv:1808.01462 [cs.CV]

David Arthur and Sergei Vassilvitskii. 2007. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (New Orleans, Louisiana) *(SODA '07)*. Society for Industrial and Applied Mathematics, USA, 1027–1035.

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. Springer, 404–417.

Mariana Belgiu and Lucian Drăguţ. 2016. Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing* 114 (2016), 24–31.

Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*. Vol. 4. Springer.

Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.

Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. 2010. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer, 778–792.

John Canny. 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8, 6 (1986), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

Jun-Dong Chang, Shyr-Shen Yu, Hong-Hao Chen, and Chwei-Shyong Tsai. 2010. HSV-based Color Texture Image Classification using Wavelet Transform and Motif Patterns. *Journal of Computers* 20 (01 2010).

Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. 2002. Choosing multiple parameters for support vector machines. *Machine learning* 46 (2002), 131–159.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20 (1995), 273–297.

T. Cover and P. Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13, 1 (1967), 21–27. https://doi.org/10.1109/TIT.1967.1053964

Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. 2004. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, Vol. 1. Prague, 1–2.

Li Deng and Dong Yu. 2014. Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing* 7, 3–4 (2014), 197–387. https://doi.org/10.1561/2000000039

Pedro Domingos and Michael Pazzani. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning* 29 (1997), 103–130.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 580–587.

Rafael Gonzalez and Zahraa Faisal. 2019. *Digital Image Processing Second Edition.*

Guodong Guo, Stan Z Li, and Kap Luk Chan. 2001. Support vector machines for face recognition. *Image and Vision computing* 19, 9-10 (2001), 631–638.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016a. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016b. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90

Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. 2002. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (2002), 881–892. https://api.semanticscholar.org/CorpusID:12003435

Ebrahim Karami, Siva Prasad, and Mohamed Shehata. 2017. Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726* (2017).

Yan Ke and Rahul Sukthankar. 2004. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Vol. 2. IEEE, II–II.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, Vol. 25.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.

S. Lazebnik, C. Schmid, and J. Ponce. 2006. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. 2169–2178. https://doi.org/10.1109/CVPR.2006.68

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

Andy Liaw and Matthew Wiener. 2002. Classification and Regression by randomForest. *R News* 2, 3 (2002), 18–22. http://CRAN.R-project.org/doc/Rnews/

David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision* 60, 2 (nov 2004), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Daniel Maturana, Domingo Mery, and Alvaro Soto. 2009. Face recognition with local binary patterns, spatial pyramid histograms and naive Bayes nearest neighbor classification. In *2009 International Conference of the Chilean Computer Science Society*. IEEE, 125–132.

K. Mikolajczyk and C. Schmid. 2005. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 10 (2005), 1615–1630. https://doi.org/10.1109/TPAMI.2005.188

Ranjan Muthukrishnan and Miyilsamy Radha. 2011. Edge detection techniques for image segmentation. *International Journal of Computer Science & Information Technology* 3, 6 (2011), 259.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).

Irina Rish et al. 2001. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, Vol. 3. Citeseer, 41–46.

Victor Francisco Rodriguez-Galiano, Bardan Ghimire, John Rogan, Mario Chica-Olmo, and Juan Pedro Rigol-Sanchez. 2012. An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS journal of photogrammetry and remote sensing* 67 (2012), 93–104.

Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*. 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544

Dominik Scherer, Andreas Müller, and Sven Behnke. 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In *International conference on artificial neural networks*. Springer, 92–101.

Gaurav Sharma. 2003. *Digital Color Imaging Handbook.* CRC Press.

Gaurav Sharma, Wencheng Wu, and Edul N. Dalal. 2005. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application* 30, 1 (2005), 21–30. https://doi.org/10.1002/col.20070 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/col.20070

Sivic and Zisserman. 2003. Video Google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*. 1470–1477 vol.2. https://doi.org/10.1109/ICCV.2003.1238663

Michael J. Swain and Dana H. Ballard. 1991. Color indexing. *International Journal of Computer Vision* 7 (1991), 11–32. https://api.semanticscholar.org/CorpusID:8167136

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

Richard Szeliski. 2022. *Computer vision: algorithms and applications*. Springer Nature.

Antonio Torralba, Rob Fergus, and William T. Freeman. 2008. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 11 (2008), 1958–1970. https://doi.org/10.1109/TPAMI.

2008.128

Ivana Zeger, Sonja Grgic, Josip Vukovic, and G. Sisul. 2021. Grayscale Image Colorization Methods: Overview and Evaluation. *IEEE Access* PP (08 2021), 1–1. https://doi.org/10.1109/ACCESS.2021.3104515

L. Zhang and H. Lu. 2016. Deep cross-modal hashing for image-text retrieval. *IEEE Transactions on Image Processing* 25, 12 (2016), 5750–5761.

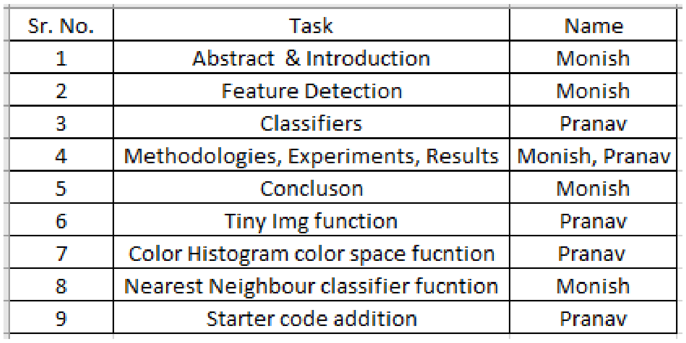
Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence* 40, 6 (2017), 1452–1464.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2018. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 6 (2018), 1452–1464. https://doi.org/10.1109/TPAMI.2017.2723009

Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. 2014. Learning deep features for scene recognition using places database. *Advances in neural information processing systems* 27 (2014).

5 Appendix

Work Distribution Table:



Sr. No.	Task	Name
1	Abstract & Introduction	Monish
2	Feature Detection	Monish
3	Classifiers	Pranav
4	Methodologies, Experiments, Results	Monish, Pranav
5	Concluson	Monish
6	Tiny Img function	Pranav
7	Color Histogram color space fucntion	Pranav
8	Nearest Neighbour classifier fucntion	Monish
9	Starter code addition	Pranav

Figure 9: Work Distribution Table

Received 19 March 2024