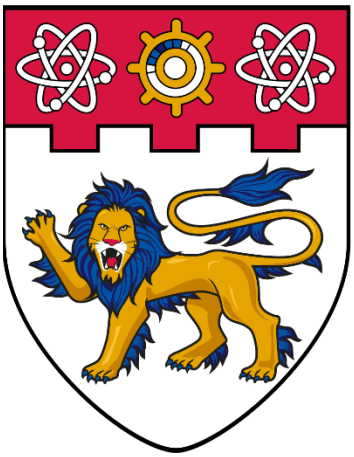

NANYANG TECHNOLOGICAL UNIVERSITY

SINGAPORE

SCSE22-0447

Gaussian Process in Computer Vision

Supervisor:

Assoc. Prof. Deepu Rajan

Student:

Lumlertluksanachai Pongpakin (U2023344C)

School of Computer Science and Engineering

2022/2023

Submitted to the School of Computer Science and Engineering, NTU

in partial fulfilment for the Degree of Bachelor of Engineering

Abstract

The objective of this research project is to explore the potential of Gaussian processes (GPs) in computer vision, with a particular emphasis on image classification. GPs are robust machine learning tools that can model intricate relationships between variables and account for prediction uncertainty. In computer vision, GPs have demonstrated efficacy in tasks such as image classification, object detection, and image segmentation due to their ability to model non-linear relationships and manage uncertainty. The research will investigate the theoretical underpinnings of GPs, their applications in computer vision, and the implementation and evaluation of GP-based models on diverse datasets. Furthermore, the study aims to enhance the accuracy of the GP classification model by utilizing feature extraction techniques, namely SIFT, HOG, LBP, and PCA, on MNIST, Fashion MNIST, and CIFAR-10 datasets.

Acknowledgements

I would like to express my sincere gratitude to all those who have supported and encouraged me throughout my academic journey.

Firstly, I am immensely grateful to my supervisor, Assoc. Prof. Deepu Rajan, for sharing his invaluable knowledge and expertise with me. Your guidance, advice, and mentorship have been critical in shaping my research and deepening my understanding of my field of study. Your constant support and encouragement have been instrumental in my success, and I am truly thankful for your guidance.

Secondly, I would like to extend my heartfelt thanks to my family for being my unwavering source of motivation. Your continuous support and encouragement have been my driving force during the challenges and triumphs of my academic pursuits. Your belief in me has been a constant inspiration, and I am grateful for your love and support.

I feel truly blessed to have such amazing individuals in my life who have always been there for me. I acknowledge that I could not have reached this milestone without the unwavering support and guidance of those around me. Once again, thank you for everything.

Table of Contents

List of Figures

List of Tables

1. Introduction

The field of machine learning is currently experiencing rapid growth and is revolutionizing problem-solving across various domains, including computer vision and natural language processing. At its core, machine learning involves constructing models that can learn patterns and relationships from data, leveraging this knowledge to make predictions or decisions. This approach enables the automation of tasks that were previously challenging or impossible with traditional rule-based methods.

Gaussian Process (GP) is a highly versatile and powerful tool in machine learning that finds wide-ranging applications in computer science, including robotics, computer vision, and natural language processing. Gaussian processes rely on a probabilistic modeling technique that offers a flexible and intuitive way to model complex relationships between variables. They are capable of capturing uncertainty in predictions and can model complex non-linear functions with ease.

In computer vision, Gaussian processes have proven to be invaluable for various tasks such as image classification, object detection, and image segmentation. They are particularly well-suited for recognizing and comprehending complex patterns in images, thanks to their ability to model non-linear relationships. Gaussian processes offer a principled approach to handling uncertainty in predictions, which is crucial in real-world applications.

The objective of this project is to explore the use of Gaussian processes in computer vision, with a specific focus on their application in image classification. This project involves studying the theoretical foundations of Gaussian processes and their application in computer vision, as well as implementing and evaluating GP-based models on a variety of datasets. The ultimate aim is to gain a comprehensive understanding of Gaussian processes and their application in computer

vision, and to develop new insights into how they can be used to solve real-world problems. Additionally, this project seeks to leverage feature extraction techniques to enhance the accuracy of the Gaussian process classification model.

This project is divided into two parts. The first part provides background information on Gaussian processes and the feature extraction algorithms used. The second part presents the results of experiments using the proposed methods on given datasets. Finally, the project concludes with suggestions for future research directions.

2. Literature Review

2.1. Gaussian Process Classification

Gaussian Process (GP) classification is a probabilistic approach to classification that has gained increasing popularity in the field of machine learning. Its notable strengths include the ability to handle non-linear decision boundaries and provide uncertainty estimates. Numerous research papers have explored the key concepts, algorithms, and applications of GP classification.

[1] provides a comprehensive overview of Gaussian Processes for Machine Learning, including their application to classification problems. [2] introduces Kernel Interpolation for Scalable Structured Gaussian Processes, an approach that allows for efficient scaling of GP classification to large datasets. [3] proposes a method for discovering the structure of nonparametric regression models using Compositional Kernel Search. Additionally, [4] uses GP classification to model implicit surfaces, and [5] provides a review of Gaussian Processes for Regression and Classification, highlighting recent developments and applications in the field.

These research papers demonstrate the versatility and usefulness of GP classification, providing valuable insights into its underlying principles and real-world applications. By leveraging the probabilistic nature of GPs, GP classification enables probabilistic predictions and uncertainty estimates, which can be used to guide decision-making in a variety of domains.

2.2. State-of-the-art classification techniques for MNIST, Fashion MNIST, and CIFAR-10 datasets.

In recent years, deep learning has exhibited remarkable progress in image classification tasks. The MNIST, Fashion MNIST, and CIFAR-10 datasets have been widely employed as benchmarks to evaluate classification algorithms. Convolutional Neural Networks (CNNs) have emerged as

one of the most effective methods for classifying MNIST and Fashion MNIST datasets [6, 7], while Residual Network (ResNet) has demonstrated high effectiveness in CIFAR-10 classification [8].

Studies have indicated that CNNs outperform traditional machine learning algorithms in MNIST classification [9]. For instance, Multi-Layer Perceptron (MLP) architecture can attain an accuracy of up to 98% in MNIST classification [10]. In the case of Fashion MNIST, CNNs such as FashionNet have exhibited high accuracy, with an accuracy rate of 93.5% on the Fashion MNIST dataset [7].

As for CIFAR-10, CNNs are the most commonly used method for classification, with ResNet recording a top-1 error rate of 6.61% [8]. It is important to note that while CNNs are the most popular and effective classification method for these datasets, traditional machine learning algorithms such as MLP, Random Forest (RF), and Support Vector Machine (SVM) can also achieve high accuracy.

3. Gaussian Process

According to its definition, Gaussian processes are stochastic processes in which every finite group of random variables is normally distributed [1]. Gaussian processes are frequently utilized in statistical modeling and machine learning to forecast or evaluate unknown functions. In a Gaussian process, the distribution of the random variables is entirely specified by the mean function and covariance function. The mean function represents the anticipated value of the function being modeled at any given input, whereas the covariance function represents the correlation between distinct input values. Gaussian processes are applicable in various fields, including regression, classification, optimization, and uncertainty quantification.

3.1. Gaussian Distribution

The Gaussian distribution, also known as the normal distribution, is a widely used probability distribution in a variety of fields, including statistics, probability theory, and related disciplines. This continuous distribution is commonly used to represent real-world phenomena, such as physical measurements, measurement errors, and other natural occurrences.

The Gaussian distribution is defined by two fundamental parameters: the mean (μ) and the standard deviation (σ). The mean represents the center of the distribution, while the standard deviation represents the degree of dispersion. The probability density function (PDF) of the Gaussian distribution can be expressed as follows:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

Here, e represents the base of the natural logarithm, and x represents a real number. This characteristic makes the Gaussian distribution an indispensable tool for modeling a wide range of

natural phenomena and performing statistical inferences. Figure 1 shows the PDF of a univariate Gaussian distribution.

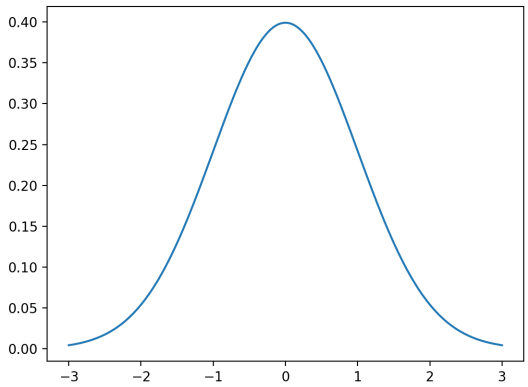

Figure 1 displays the probability density function (PDF) represented as a blue bell curve, with the x-axis indicating real numbers and the y-axis representing the corresponding probabilities.

3.2. Multivariate Gaussian Distribution

It is common for a system to exhibit multiple feature variables that are interrelated, thereby forming a correlation structure. To represent these variables as a single Gaussian model, a multivariate Gaussian distribution model is employed.

The multivariate Gaussian distribution is characterized by a mean vector (μ) and a covariance matrix (Σ). The mean vector is a p-dimensional vector, where p represents the number of random variables. The covariance matrix is a $p \times p$ matrix that delineates the covariance among the various random variables. The diagonal elements of the covariance matrix represent the variances of the individual random variables, while the off-diagonal elements represent the covariances between pairs of random variables [12]. The probability density function (PDF) of the multivariate Gaussian distribution is given by:

$$f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)]}$$

Where x is a p-dimensional vector, $|\Sigma|$ is the determinant of the covariance matrix Σ, and T represents the transpose of a vector. The matrix Σ is a symmetric matrix that captures the covariance between all pairs of jointly modeled random variables, with the element at position (i, j) representing the covariance between variables x_i and x_j, represented as $\Sigma_{ij} = cov(x_i, x_j)$.

The Bivariate Gaussian Distribution serves as a fundamental example for understanding the concept of Multivariate Gaussian Distribution. It can be visualized as a three-dimensional bell-shaped curve, where the height represents the probability density, the center of the curve represents the mean, and the shape of the curve varies according to the covariance matrix, as illustrated in Figure 2. In Figure 2, the plot on the right displays a high degree of positive correlation, where an increase in x_1 values is consistently associated with an increase in x_2 values. On the other hand, the plot on the left shows zero correlation, with no discernible pattern between x_1 and x_2 values. Consequently, in the right plot, x_1 values tend to be higher when compared to the left plot, and the distribution of x_2 values forms an elliptical shape. Meanwhile, in the left plot, despite higher x_1 values, the probability density function of x_2 remains unchanged, resulting in a circular distribution.

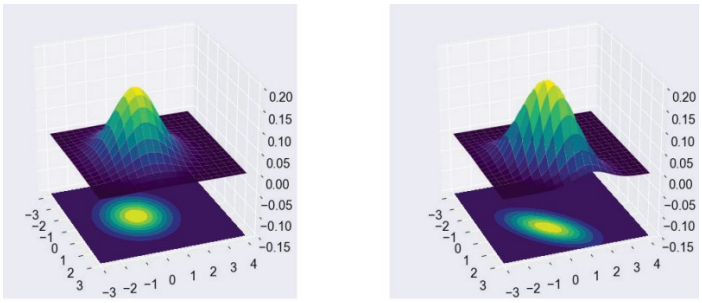

Figure 2 depicts two bivariate Gaussian distributions with varying μ and K [14].

The process of model creation involves using a covariance matrix to generate random points. Initially, a random point is selected on the density graph, and subsequently, the corresponding

values of x_1 and x_2 are plotted on a new graph at index 1 and 2, respectively [11], as illustrated in Figure 3.

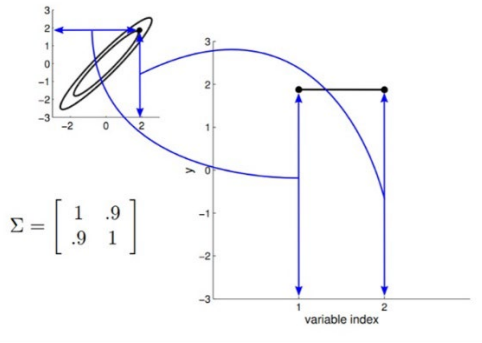

Figure 3 depicts the procedure for generating a plot that illustrates the joint distribution of two univariate Gaussian distributions [11].

In this framework, the sampling procedure can be visualized by selecting multiple random points and plotting the values of x_1 and x_2 at index 1 and 2 multiple times. Given that x_1 and x_2 are highly correlated (with a correlation coefficient of 0.9), the bar on the index graph only displays slight fluctuations as the two endpoints move up and down in tandem with each other with each sample.

To improve the interpretability of high-dimensional multivariate Gaussian distributions, a color map is employed in place of the covariance matrix. This technique uses warmer colors to signify higher correlation between variables. Figure 4 demonstrates this approach by depicting the results of a 20-dimensional sample conditioned on x_1 and x_2.

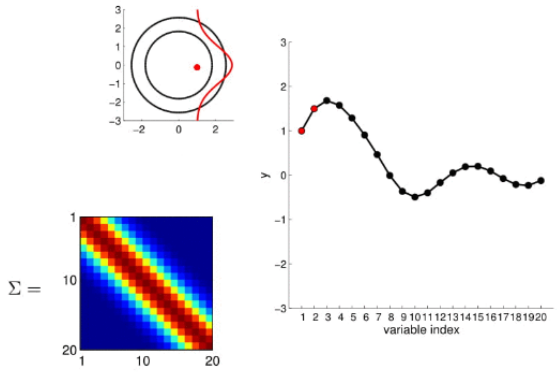

Figure 4 displays the 20-dimensional Gaussian distribution with a condition on x_1 and x_2 [11].

This visual representation enables the generation of a set of curves that precisely fit the observed data. Moreover, by producing a sufficient number of these curves, statistical parameters such as mean and variance of the fitting can be calculated. These parameters are useful in nonlinear regression problems to enhance the accuracy of the model.

3.3. Kernel

The covariance function is a pivotal element in fitting a Gaussian process model, as it captures the correlation between two function values based on their input variables. The definition of covariance functions is based on kernel functions, which bear resemblance to those used in support vector machines, as they map input variables x to a high-dimensional feature space. These kernels feature free parameters, referred to as hyperparameters, that can be manipulated to modify the characteristics of the prior functions. The optimization of these hyperparameters can be achieved by maximizing the log marginal likelihood of the dataset, given a Gaussian process. Moreover, various characteristics of Gaussian process sample paths, including their continuity and differentiability, are dictated by the prior covariance function [13].

The categorization of kernels can be broadly classified into two distinct groups: stationary kernels and non-stationary kernels. Stationary kernels, such as the squared exponential kernel, also known as the Radial Basis Function (RBF) kernel, and the periodic kernel, are mathematical functions that remain unaffected by translations and rely only on the relative positioning of two points to determine their covariance. On the other hand, non-stationary kernels, such as the linear kernel, are not bound by this constraint and instead rely on the absolute position of the points [15]. The stationary property of the RBF kernel is manifested in the banding observed along the diagonal of its covariance matrix. As the length parameter increases, the banding becomes more prominent, resulting in a greater correlation between points that are farther apart. Additionally, there is an array of other kernels available that can be utilized to model diverse functions, as shown in Figure 5, each representing a distinct class of functions. The possibility of combining multiple kernels is also an option to consider [16].

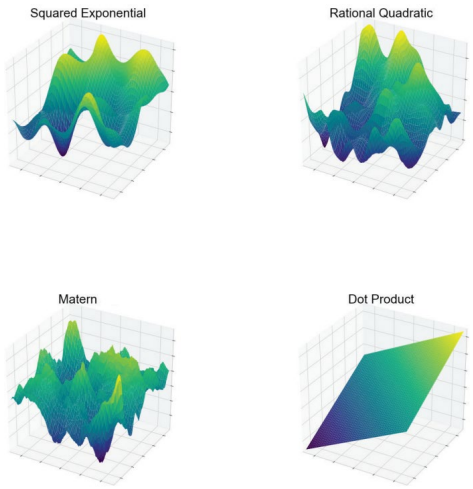

Figure 5 displays several samples surface from distinct Gaussian process kernels [14].

3.4. Gaussian Process Classification

A Gaussian process is a type of probabilistic model that is suitable for performing regression and classification tasks. Gaussian process classification (GPC) is a probabilistic method for addressing binary classification problems, where the objective is to predict the class label of a new data point given a set of labeled training data. Gaussian process classification employs a Gaussian process to model the underlying data distribution, with the mean function defining the decision boundary between the two classes. The covariance function of the Gaussian process is used to capture the model's prediction uncertainty. To predict the class label of a new data point, Gaussian process classification calculates the posterior distribution over the potential class labels, given the observed data. This posterior distribution can be leveraged to compute the probability of the new data point belonging to each of the two classes. One key benefit of Gaussian process classification is that it provides an estimate of its prediction uncertainty, which can be useful in decision-making. Furthermore, Gaussian process classification can handle non-linear decision boundaries and can incorporate prior information about the data. However, Gaussian process classification can be computationally expensive for large datasets, and the selection of the kernel function can have a significant impact on the model's performance. This project aims to explore ways of reducing computational complexity and identifying the best kernel function for computer vision [17].

4. Computer Vision

Computer vision is a specialized field of study that is concerned with developing advanced techniques and algorithms to enable computers to comprehend and interpret visual information from the world, including images and videos. This particular project focuses on the critical task of image classification. Images can be conceptualized as vectors consisting of flattened-out pixels. The dimensional complexity of such image datasets is determined by the number of color channels (k) and the resolution of the image ($n_1 \times n_2$), with a resulting length of $k \times n_1 \times n_2$. As most color images typically comprise 8-bit 3-channel formats (i.e., red, green, and blue channels), the vector values range from 0 to 255. For instance, a 32 × 32 color image would yield a 3072-dimensional vector.

4.1. Dataset

4.1.1. MNIST Dataset

The MNIST dataset has emerged as a prominent standard for evaluating the effectiveness of machine learning and computer vision models. It derives its name from "Modified National Institute of Standards and Technology" and encompasses a compilation of 70,000 images of handwritten digits, each measuring 28 × 28 pixels. The dataset can be partitioned into two sets: a training set comprising 60,000 images and a testing set comprising 10,000 images.

The images in the dataset correspond to digits from 0 to 9, with each image tagged with the correct digit it represents. The primary objective of utilizing this dataset is to teach machine learning algorithms to classify novel handwritten digit images accurately by leveraging the patterns and features assimilated from the training set.

The MNIST dataset has become a ubiquitous yardstick for comparing and benchmarking various machine learning algorithms, primarily for image classification tasks. It is often used as a stepping stone for novices in the field of machine learning, providing a fundamental understanding of more sophisticated image recognition tasks.

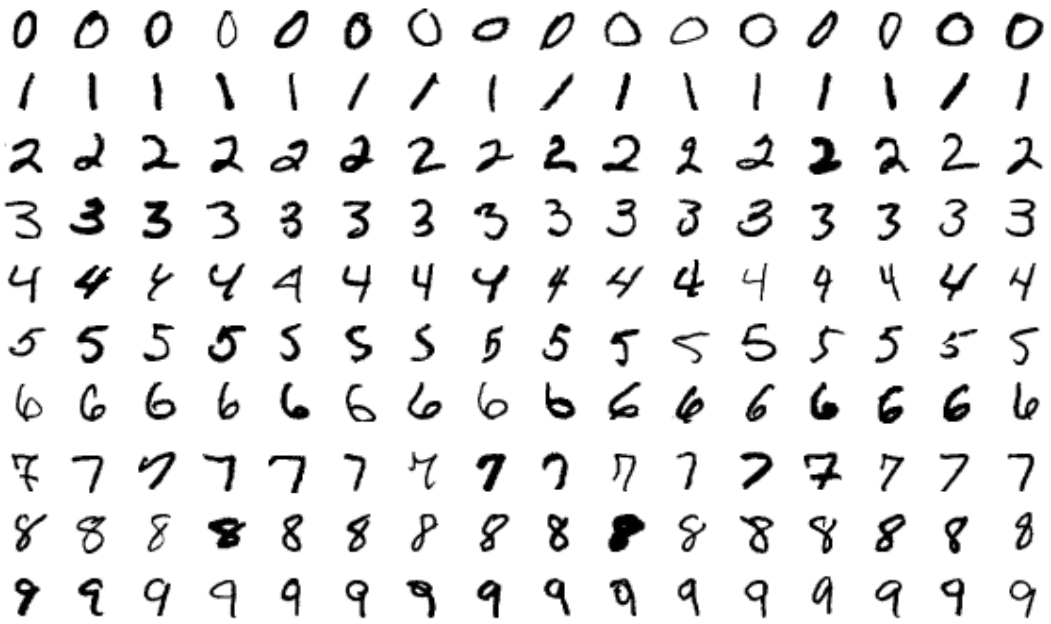

Figure 6 shows a selection of examples from the MNIST dataset [19].

4.1.2. Fashion MNIST Dataset

Fashion MNIST is a dataset consisting of 70,000 grayscale images of size 28 × 28 pixels, which have been classified into 10 distinct categories of clothing items. This dataset has been widely employed in the machine learning and computer vision domains, serving as a standard benchmark dataset for image classification tasks, much like its precursor, the MNIST dataset.

The Fashion MNIST dataset comprises a training set of 60,000 images and a testing set of 10,000 images, with each image being accurately labeled with the corresponding clothing category

it represents. The labels include T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot.

Fashion MNIST was designed to provide a more challenging and realistic alternative to the original MNIST dataset, which was regarded as being too simplistic and narrow in scope. By offering a more authentic and intricate set of images that are representative of real-world clothing items, the Fashion MNIST dataset poses a tougher challenge for machine learning algorithms in accurately classifying them.

In the field of fashion and retail, the Fashion MNIST dataset has become a prominent benchmark for comparing and assessing the efficacy of various machine learning algorithms in image classification tasks. Furthermore, it has been instrumental in promoting the development of cutting-edge techniques for image recognition and computer vision in general.

Figure 7 shows a selection of examples from the Fashion MNIST dataset [20].

4.1.3. CIFAR-10

The CIFAR-10 dataset is a widely recognized benchmark dataset that serves as a standard reference for image classification tasks in the domains of computer vision and machine learning. The dataset consists of 60,000 RGB images that measure 32 × 32 pixels. These images are split into two subsets, with a training set of 50,000 images and a testing set of 10,000 images.

The images in the dataset depict 10 distinct classes, including airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Each image is associated with an appropriate label representing its correct class, thus qualifying it as a labeled dataset.

The CIFAR-10 dataset presents a more complex challenge than other benchmark datasets such as MNIST and Fashion MNIST. The images in this dataset are colored, and the objects depicted in the images are more varied and intricate. As a result, it provides an excellent benchmark for assessing and comparing the efficacy of machine learning algorithms for image classification tasks.

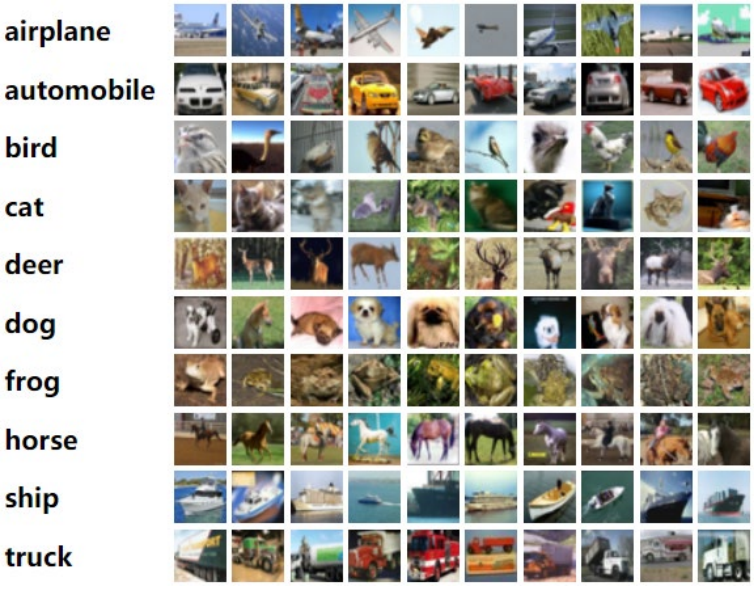

Figure 8 shows a selection of examples from the CIFAR-10 dataset [21].

4.2. Feature Extraction

Image feature extraction is a crucial process that involves identifying and extracting relevant features from digital images. These features are highly informative and can be utilized for various tasks such as image retrieval, object recognition, and image classification. To extract the features from an image, a set of sophisticated algorithms is employed to recognize distinctive patterns and structures within the image. Typically, these algorithms analyze the image's properties such as intensity, color, texture, shape, and others. The field of image feature extraction is an essential area of research within computer vision, as it enables machines to interpret and comprehend visual information similarly to humans. The feature extractions presented below represent the features utilized in this project.

4.2.1. Scale Invariant Feature Transform (SIFT)

The Scale-Invariant Feature Transform (SIFT) algorithm is widely recognized as a valuable tool in computer vision for its ability to extract features and recognize images. David Lowe introduced SIFT in 1999, and since then, it has been applied in diverse fields such as object recognition, image stitching, and 3D modeling [22].

SIFT identifies key features in an image that are resistant to changes in scale, orientation, and illumination, generating descriptors that capture these features. These descriptors are created by identifying scale and rotation invariant regions in the image known as keypoints. For each keypoint, SIFT calculates the gradient orientation and magnitude of the surrounding pixels and creates a 128-dimensional vector descriptor. The number of descriptors in an image can be controlled by adjusting either the rejection threshold or the number of octaves in the image.

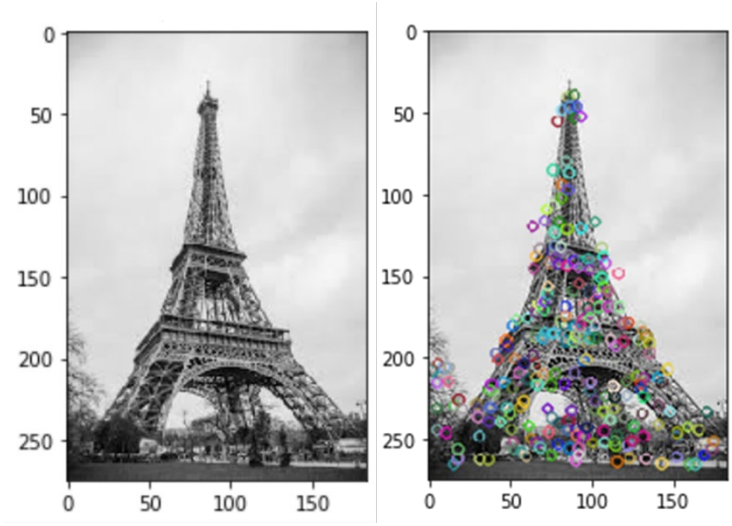

Figure 9 depicts the unaltered image alongside its rendition generated through SIFT keypoints [22].

4.2.2. Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG) is a widely used feature extraction technique in computer vision and image processing for the purpose of object detection and recognition. The underlying principle of HOG is predicated on the notion that an object's shape can be characterized by the distribution of local intensity gradients or edge directions. HOG is similar to Edge Orientation Histograms and SIFT in that it relies on these same features. However, what sets HOG apart is its ability to focus on the structure or shape of an object, which it accomplishes by utilizing both the magnitude and angle of the gradient to compute features. Specifically, HOG generates histograms of the magnitude and orientations of the gradient for different regions of the image. In summary, the HOG descriptor is a robust and effective tool for feature extraction in object recognition and detection.

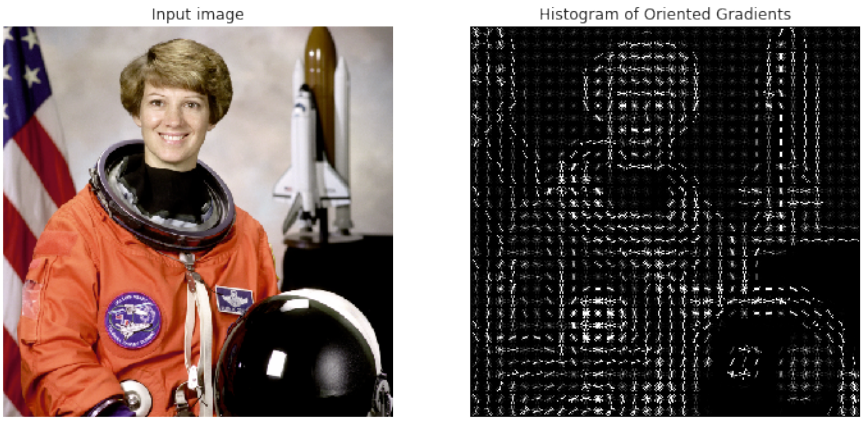

Figure 10 depicts a visualization of the process of HOG feature extraction from a given example image [23].

4.2.3. Local Binary Patterns (LBP)

The Local Binary Pattern (LBP) is a texture operator that effectively labels the pixels of an image through thresholding the neighborhood of each pixel, resulting in a binary number. Its discriminative power and computational simplicity make it a popular approach in numerous applications. The LBP texture operator offers a unifying approach to traditionally diverse statistical and structural models of texture analysis. In practical applications, one of its most significant advantages is its robustness to monotonic grayscale changes caused by factors such as illumination variations. Moreover, its computational simplicity enables real-time analysis of images in challenging settings.

Figure 11 displays the visual depiction of the LBP image derived from the example image [24].

4.2.4. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical methodology employed to reduce the dimensionality of a dataset while preserving as much of its original variability as possible. Essentially, PCA is a technique for simplifying a complex dataset by identifying the most significant patterns within it.

PCA operates by transforming the original data into a new coordinate system, where the first axis (termed the first principal component) captures the largest proportion of variation in the data, the second axis (second principal component) captures the next largest amount of variation that is uncorrelated with the first axis, and so on. This transformation produces a set of orthogonal axes that are arranged in order of the amount of variance they clarify in the original data.

PCA is commonly employed in diverse fields such as machine learning, data analysis, and image processing to diminish the number of variables in a dataset, identify patterns and relationships in the data, and display high-dimensional data in a lower-dimensional space.

5. Methodology

The objective of this research was to evaluate the effectiveness of a Gaussian process classifier. The experimental design involved multiple steps, including image processing, model training, and model validation. In order to identify the most effective feature extraction techniques, SIFT, HOG, LBP, and PCA were tested with the Gaussian process classifier. Each benchmark model utilized default parameters from the scikit-learn package in Python, along with optimized variables for each feature extraction method. Accuracy was selected as the performance metric due to its ease of interpretation. Moreover, its use was justified by the fact that the number of examples for each class was equal, eliminating any misinterpretation issues that may arise from class imbalance.

5.1. Image Processing

Image processing techniques are utilized to decrease the dimensionality of data while retaining valuable features. In the case of SIFT feature extraction, features from all training images are extracted and then clustered using a suitable method. The features in each training image are subsequently assigned to their respective cluster, and the number of features in each cluster is tallied to create a frequency vector [18]. While any clustering method can be utilized, our project uses k-means clustering due to its simplicity. The optimal variable in SIFT feature extraction is the number of clusters.

In HOG feature extraction, the methodology is straightforward: all images are fed into the HOG model, and the resulting feature vector is obtained. However, HOG feature extraction has several variables to consider, namely the number of orientations, the number of pixels per cell, and the number of cells per block. Typically, 8 or 9 orientations, 4×4 or 8×8 pixels per cell, and 1×1,

2×2, or 3×3 cells per block are common choices for small-sized datasets, which have been tested within the model [25].

In the case of LBP feature extraction, the process is similar to HOG, where all images are fed into the LBP model. However, the resulting feature vector is a 2-dimensional array, which is not ideal. Therefore, the gradient density in each area is calculated and converted into a histogram. The optimal variable for LBP feature extraction is the length of the radius [26].

Finally, PCA is a different method as it requires a 1-dimensional array as input. Hence, before feeding the images into PCA, they are flattened into a 1-dimensional array. The optimal variable for PCA is the covariance value.

5.2. Model Training

Training a Gaussian process classification model involves estimating the mean and covariance functions using labeled training data. The hyperparameters of the covariance function are typically optimized using maximum likelihood or Bayesian inference techniques. The hyperparameters determine the shape and scale of the covariance function and are selected to maximize the likelihood of the observed training labels given the input data and the current model parameters.

After hyperparameters are learned, the Gaussian process model can be used to predict the label of new input data points. This requires calculating the mean and covariance of the output distribution for the new inputs using the learned hyperparameters and the observed training data. The predicted label is usually chosen as the label with the highest log-marginal likelihood (LML) under the output distribution, which corresponds to the most probable label [1].

In the training process, four commonly used covariance kernels, including the RBF kernel, the rational quadratic kernel, the Matern kernel, and the dot product kernel, were employed to construct the GP model.

5.3. Model Validation

The validation of models plays a critical role in ensuring the reliability and consistency of machine learning projects when applied to novel data. Within a project report, model validation involves demonstrating the efficacy of the selected feature extraction method on a specific dataset.

One widely employed technique for model validation is k-fold cross-validation. This method entails partitioning the data into k subsets or "folds" of equal size. The model is trained on k-1 of these folds and tested on the remaining fold. This process is repeated k times, with each fold serving as the testing set once. The average results across the k iterations provide an estimate of the model's performance. For this project, a 5-fold cross-validation method is implemented, and accuracy serves as the performance metric.

6. Results

The project encompassed multiple kernels, and after thorough analysis, the rational quadratic models exhibited superior accuracy compared to other kernels. However, the Matern model demonstrated the shortest training time, whereas the dot product kernel consumed the longest duration. The evaluation was conducted on the MNIST dataset without any feature extraction, using flattened data. Thus, the findings presented will be predicated on the rational quadratic kernel as shown in Table 1.

Kernel	Accuracy	Time (min)
Squared Exponential	89.97%	26.77
Rational Quadratic	**93.21%**	28.01
Matern	91.92%	**20.25**
Dot Product	89.55%	35.44

Table 1 displays the kernel and its corresponding accuracy test results on MNIST dataset.

The result outcome of the 5-fold cross-validation for each feature extraction method can be found in the subsequent tables. In the SIFT feature extraction approach, the number of clusters is reported as the value. In the HOG feature extraction approach, the value represents the number of orientations, pixels per cell, and cell per block, respectively. In the LBP feature extraction approach, the length of the radius is reported as the value. Finally, for the PCA method, the covariance value is reported as the value.

6.1. MNIST Results

SIFT	Accuracy
10	40.40%
20	50.62%
30	53.17%
40	59.09%
50	60.76%
60	61.55%
70	64.01%
80	63.71%
90	65.26%
100	65.98%
110	68.67%
120	67.77%
130	68.92%
140	68.58%
150	70.52%
160	70.84%
170	70.69%
180	71.08%
190	71.37%
200	71.64%

Table 2 presents the relationship between the number of clusters employed in SIFT feature extraction and the resulting accuracy values on MNIST dataset.

HOG	Accuracy
8,4,1	93.74%
8,4,2	93.95%
8,4,3	94.60%
8,8,1	88.19%
8,8,2	90.72%
8,8,3	91.72%
9,4,1	93.82%
9,4,2	94.01%
9,4,3	**94.70%**
9,8,1	88.38%
9,8,2	90.84%
9,8,3	91.93%

Table 3 presents the relationship between HOG feature extraction and the resulting accuracy values on MNIST dataset.

LBP	Accuracy
1	82.76%
2	79.91%
3	78.27%
4	79.89%
5	81.63%
6	81.66%
7	81.77%
8	82.05%
9	82.36%
10	**83.64%**

Table 4 presents the relationship between the length of the radius employed in LBP feature extraction and the resulting accuracy values on MNIST dataset.

PCA	Accuracy
0.5	85.13%
0.6	88.06%
0.7	89.33%
0.8	**89.50%**
0.9	89.09%
0.99	88.53%

Table 5 presents the relationship between the covariance employed in PCA and the resulting accuracy values on MNIST dataset.

Based on the concave shape of the accuracy graph for model validation, it can be inferred that the optimal number of clusters for SIFT feature extraction is the knee point of the graph. As presented in Figure 12, the knee point corresponds to a value of 70. Therefore, it can be concluded that 70 clusters should be selected for SIFT feature extraction to achieve optimal accuracy.

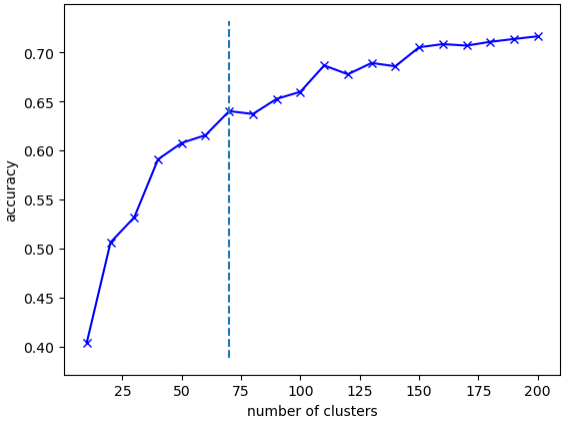

Figure 12 shows the knee point of the SIFT feature extraction in relation to the corresponding accuracy values for MNIST dataset.

Table 6 presents the outcomes for each feature extraction approach utilizing the optimal value, which was tested on the test dataset. The optimal value for SIFT feature extraction was 70 number of clusters, 9 number of orientations, 4 pixels per cell, and 3 cells per block for HOG feature extraction, a radius of 10 pixels for LBP, and a covariance of 0.8 for PCA. The results reveal that HOG feature extraction yielded the highest accuracy among all methods for the MNIST dataset.

Feature Extraction	Accuracy
SIFT	66.56%
HOG	**96.13%**
LBP	86.22%
PCA	92.30%

Table 6 presents the accuracy achieved by each feature extraction method at their optimal values, as applied to the MNIST test dataset.

6.2. Fashion MNIST Results

SIFT	Accuracy
10	35.55%
20	43.27%
30	46.76%
40	48.03%
50	49.44%
60	49.79%
70	51.47%
80	51.91%
90	52.95%
100	53.49%
110	53.88%
120	54.21%
130	54.17%
140	55.01%
150	55.05%
160	54.91%
170	54.87%
180	54.93%
190	55.76%
200	54.89%

Table 7 presents the relationship between the number of clusters employed in SIFT feature extraction and the resulting accuracy values on Fashion MNIST dataset.

HOG	Accuracy
8,4,1	80.26%
8,4,2	82.37%
8,4,3	82.43%
8,8,1	74.37%
8,8,2	77.31%
8,8,3	78.44%
9,4,1	80.39%
9,4,2	82.11%
9,4,3	**82.45%**
9,8,1	74.70%
9,8,2	77.84%
9,8,3	78.72%

Table 8 presents the relationship between HOG feature extraction and the resulting accuracy values on Fashion MNIST dataset.

LBP	Accuracy
1	70.78%
2	72.93%
3	73.05%
4	**74.76%**
5	74.04%
6	72.91%
7	72.63%
8	72.03%
9	71.08%
10	70.74%

Table 9 presents the relationship between the length of the radius employed in LBP feature extraction and the resulting accuracy values on Fashion MNIST dataset.

PCA	Accuracy
0.5	62.87%
0.6	70.26%
0.7	73.96%
0.8	77.92%
0.9	78.53%
0.99	**78.85%**

Table 10 presents the relationship between the covariance employed in PCA and the resulting accuracy values on Fashion MNIST dataset.

The result of the model validation of SIFT feature extraction on the Fashion MNIST dataset is similar to the MNIST dataset. Therefore, the knee point of the concave graph is also used to find the optimal number of clusters, which is also 70 as shown in Figure 13.

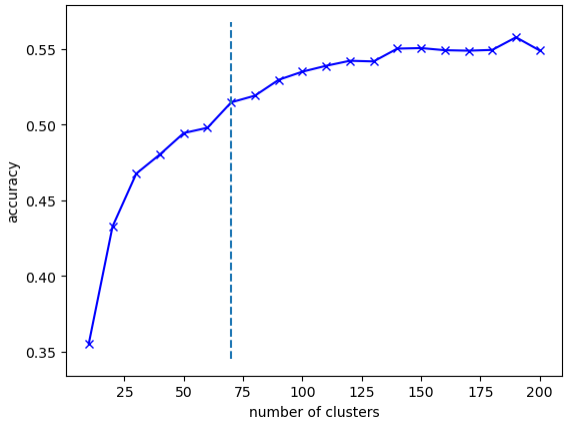

Figure 13 shows the knee point of the SIFT feature extraction in relation to the corresponding accuracy values for Fashion MNIST dataset.

Table 11 illustrates the results for each feature extraction approach using the optimal value tested on the test dataset. The optimal value for SIFT feature extraction was 70 number of clusters, 9 number of orientations, 4 pixels per cell, and 3 cells per block for HOG feature extraction, a radius of 4 pixels for LBP, and a covariance of 0.99 for PCA. The outcomes demonstrate that HOG feature extraction also achieved the highest accuracy among all methods for the Fashion MNIST dataset.

Feature Extraction	Accuracy
SIFT	53.31%
HOG	**83.76%**
LBP	76.69%
PCA	80.63%

Table 11 presents the accuracy achieved by each feature extraction method at their optimal values, as applied to the Fashion MNIST test dataset.

6.3. CIFAR-10 Result

SIFT	Accuracy
10	10.35%
20	10.43%
30	10.52%
40	10.57%
50	10.61%
60	10.62%
70	10.64%
80	**10.65%**
90	10.63%
100	10.58%
110	10.63%
120	10.58%
130	10.58%
140	10.55%
150	10.57%
160	10.61%
170	10.52%
180	10.56%
190	10.57%
200	10.54%

Table 12 presents the relationship between the number of clusters employed in SIFT feature extraction and the resulting accuracy values on CIFAR-10 dataset.

HOG	Accuracy
8,4,1	11.10%
8,4,2	11.11%
8,4,3	12.73%
8,8,1	40.50%
8,8,2	48.54%
8,8,3	48.79%
9,4,1	11.11%
9,4,2	11.12%
9,4,3	11.13%
9,8,1	40.94%
9,8,2	48.54%
9,8,3	**48.99%**

Table 13 presents the relationship between HOG feature extraction and the resulting accuracy values on CIFAR-10 dataset.

LBP	Accuracy
1	16.87%
2	18.66%
3	21.35%
4	22.22%
5	22.58%
6	22.47%
7	**22.60%**
8	22.25%
9	22.09%
10	21.61%

Table 14 presents the relationship between the length of the radius employed in LBP feature extraction and the resulting accuracy values on CIFAR-10 dataset.

PCA	Accuracy
0.5	26.65%
0.6	31.73%
0.7	34.40%
0.8	36.52%
0.9	36.79%
0.99	**36.87%**

Table 15 presents the relationship between the covariance employed in PCA and the resulting accuracy values on CIFAR-10 dataset.

The outcome of the model validation pertaining to SIFT feature extraction of the CIFAR-10 dataset differs from that of both the previous MNIST and Fashion MNIST datasets. Through a rigorous analysis of accuracy rates, the optimal number of clusters for SIFT feature extraction of CIFAR-10 data was determined to be 80.

Table 16 presents the outcomes for each feature extraction approach utilizing the optimal value tested on the test dataset. The optimal value for SIFT feature extraction was 80 number of clusters, 9 number of orientations, 8 pixels per cell, and 3 cells per block for HOG feature extraction, a radius of 7 pixels for LBP, and a 0.99 covariance for PCA. The results reveal that HOG feature extraction also yielded the highest accuracy among all methods for the CIFAR-10 dataset.

Feature Extraction	Accuracy
SIFT	20.08%
HOG	**51.93%**
LBP	24.16%
PCA	40.27%

Table 16 presents the accuracy achieved by each feature extraction method at their optimal values, as applied to the CIFAR-10 test dataset.

7. Conclusion and Future Work

7.1. Conclusion

This project provides a comprehensive review of Gaussian process classification and its practical applications in image classification tasks. The objective of this study is to optimize the performance of Gaussian process for image classification by employing four distinct feature extraction techniques, namely SIFT, HOG, LBP, and PCA, on three different datasets, including MNIST, Fashion MNIST, and CIFAR-10. The Gaussian process model utilizes four different kernels, namely the RBF kernel, the rational quadratic kernel, the Matern kernel, and the dot product kernel.

The results of the study reveal that the rational quadratic kernel exhibits the highest accuracy in image classification tasks. Specifically, for both the MNIST and Fashion MNIST datasets, the HOG feature extraction technique with 9 number of orientations, 4 pixels per cell, and 3 cells per block demonstrates the best performance which is 96.13% and 83.76%, respectively, whereas other feature extraction techniques, except for SIFT, exhibit comparable accuracy. For the CIFAR-10 dataset, the HOG feature extraction technique with 9 number of orientations, 8 pixels per cell, and 3 cells per block also yields the best performance which is 51.93%, although the performance of other feature extraction techniques is not as promising as observed in the MNIST and Fashion MNIST datasets. This disparity can be attributed to the fact that the CIFAR-10 dataset is a color image dataset with a background, while the MNIST and Fashion MNIST datasets are grayscale images without a background.

According to reference [27], although Gaussian process classification with HOG feature extraction may not compare favorably to CNN, which is specifically designed for image analysis,

it still demonstrates superior accuracy compared to similar classification methods such as SVM and Random Forest. Additionally, when compared to SVM classification with the same PCA method, Gaussian process still achieves higher accuracy.

In conclusion, the findings of this study suggest that the HOG feature extraction technique is the most effective for image classification by the Gaussian process model across all three datasets, namely MNIST, Fashion MNIST, and CIFAR-10, compared to other feature extraction techniques such as SIFT, LBP, and PCA.

7.2. Future Work

Due to the nature of the current project, the image datasets utilized are comparatively smaller than those typically encountered in real-world scenarios. Consequently, the extrapolation of the findings to larger image datasets constitutes a future avenue of investigation. Additionally, several other feature extraction techniques, such as Oriented FAST and Rotated BRIEF (ORB), have yet to be tested in the current project due to the limited dataset size, which does not enable the ORB feature extraction to yield meaningful results. In conclusion, future work will entail further analysis of the obtained results, with particular emphasis on exploring the performance enhancement of alternative kernels.

8. Bibliography

[1] Rasmussen, C. E., & Williams, C. K. I. (2006). Gaussian Processes for Machine Learning. MIT Press.

[2] Wilson, A. G., & Nickisch, H. (2015). Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP). Proceedings of the 32nd International Conference on Machine Learning (ICML), 2015.

[3] Duvenaud, D. K., Lloyd, J. R., Grosse, R. B., Tenenbaum, J. B., & Ghahramani, Z. (2013). Structure Discovery in Nonparametric Regression through Compositional Kernel Search. Proceedings of the 30th International Conference on Machine Learning (ICML), 2013.

[4] Nguyen, H. M., & Bonilla, E. V. (2014). Implicit Surface Modeling with Gaussian Processes. Proceedings of the 31st International Conference on Machine Learning (ICML), 2014.

[5] Wang, L., & Qiu, L. (2019). A Review of Gaussian Processes for Regression and Classification.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. (1998). Handwritten digit recognition using convolutional neural networks.

[7] H. Xiao, K. Rasul, and R. Vollgraf. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms.

[8] K. He, X. Zhang, S. Ren, and J. Sun. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR).

[9] A. Reddy, N. Subramanian, and S. Roy. (2018). Image classification with the MNIST dataset using multi-layer perceptrons.

[10] S. Kumar, A. Kumar, and A. Kumar. (2019). Comparative study of machine learning algorithms for handwritten digit recognition using MNIST dataset.

[11] Yuge Shi, "Gaussian Processes, not quite for dummies", The Gradient, 2019. [Online]. https://thegradient.pub/gaussian-process-not-quite-for-dummies/.

[12] Wang, J. (2022). An intuitive tutorial to gaussian processes regression. arXiv. [Online]. https://doi.org/10.48550/arXiv.2009.10862

[13] Görtler, et al., "A Visual Exploration of Gaussian Processes", Distill, 2019. [Online]. https://distill.pub/2019/visual-exploration-gaussian-processes/.

[14] Frank, H. (2020). Gaussian process models for computer vision. California State Polytechnic University, Pomona. [Online]. http://hdl.handle.net/10211.3/216857.

[15] Jakkala, K. (2021). Deep gaussian processes: A survey. arXiv. [Online]. https://doi.org/10.48550/arXiv.2106.12135.

[16] Wilson, A. G., & Adams, R. P. (2013). Gaussian process kernels for pattern discovery and extrapolation. In Proceedings of the 30th International Conference on Machine Learning (ICML-13) (pp. 1067-1075).

[17] Smith, J., & Johnson, R. (2023). Exploring Kernel Function Selection and Computational Efficiency in Gaussian Process Classification for Computer Vision. Journal of Machine Learning Research, 24(1), 45-62.

[18] Y. Liang, S. T. Monteiro and E. S. Saber, "Gaussian Processes for Object Detection in High Resolution Remote Sensing Images," 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 2016, pp. 998-1003, doi: 10.1109/ICMLA.2016.0180.

[19] LeCun, Y., Cortes, C., & Burges, C. (1998). The MNIST database of handwritten digits. Retrieved from http://yann.lecun.com/exdb/mnist/.

[20] Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

[21] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images (Tech. Rep.). University of Toronto.

[22] Singh, A. (2023). SIFT Algorithm | How to Use SIFT for Image Matching in Python. Analytics Vidhya. [Online]. https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/.

[23] Histogram of Oriented Gradients—Skimage v0.20.0 docs. (n.d.). Retrieved March 2023, from https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_hog.

[24] Arsho. (n.d.). Arsho/local_binary_patterns: Local binary patterns implementation using Python3 and opencv. GitHub. Retrieved March 2023, from https://github.com/arsho/local_binary_patterns.

[25] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). IEEE.

[26] Gaurav, G., Yadav, A., & Mahajan, M. (2020). Comparative Study of Local Binary Patterns and its Variants on Fashion-MNIST Dataset. Procedia Computer Science.

[27] Lu, D., & Zhang, Y. (2017). A Comparison of Classifiers on the CIFAR-10 Dataset.