Comparing the performance of different classification methods on real datasets using lower dimensional embeddings

Mahya Qorbani
Georgia Insitute of Technology
mqorbani3@gatech.edu, GT ID: 903814605

Abstract

Image classification has been a hot topic for several decades. In this project, The objective is to evaluate the efficacy of various classification algorithms on a dataset comprising images of different bird species, segmented into five distinct classes. The study began with the implementation of the K-Nearest Neighbors (KNN) algorithm on the original, high-dimensional dataset to establish a baseline for classification performance. To explore the impact of dimensionality reduction on classification accuracy, Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) were applied to create lower-dimensional embeddings of the data. Subsequently, the KNN algorithm was employed on these reduced datasets. A comparative analysis was performed to assess the variations in KNN's performance with and without dimensionality reduction. In addition to KNN, a Convolutional Neural Network (CNN) model was applied to the original image data. The performance of the CNN model was found to be markedly superior to that of KNN, both on the original and dimension-reduced datasets. This highlighted CNN's robustness in handling complex, high-dimensional image data.

1. Introduction

In the realm of computational data analysis, the classification of high-dimensional data remains a significant challenge, particularly in the context of image recognition and categorization. The advent of machine learning and deep learning techniques has opened new avenues for efficient data processing and analysis, especially in fields requiring intricate pattern recognition, such as species classification in natural images [3] [6] [2]. This study delves into this challenge, focusing on the classification of bird species from image datasets.

The primary objective of this project is to evaluate and compare the effectiveness of traditional machine learning algorithms against more advanced deep learning techniques in classifying high-dimensional image data. The K-Nearest Neighbors (KNN) algorithm [4] , a staple in the machine learning domain, is selected for its simplicity and effectiveness in baseline classification tasks. To understand the impact of dimensionality reduction on classification accuracy, this study employs Principal Component Analysis (PCA) [1] and t-Distributed Stochastic Neighbor Embedding (t-SNE) [7], two widely used techniques for reducing the dimensions of high-volume data while preserving its essential characteristics.

In contrast to KNN, a Convolutional Neural Network (CNN) is chosen to represent deep learning methodologies. CNNs have been at the forefront of breakthroughs in image processing and classification [8] [5], owing to their ability to learn hierarchical representations and intricate patterns in image data. This project not only compares the performance of KNN and CNN on the original, high-dimensional datasets but also examines their effectiveness on the datasets transformed by PCA and t-SNE.

Additionally, the study explores the optimization of the CNN model by fine-tuning various hyperparameters, a critical aspect that often influences the performance of deep learning models. This process aims to ascertain the most effective configurations for the CNN when dealing with complex image data.

Through this comparative analysis, the project seeks to provide insights into the suitability and effectiveness of different classification methods for image-based datasets, particularly in the context of bird species classification. It aims to contribute to the broader understanding of how dimensionality reduction techniques can impact the performance of machine learning algorithms and to highlight the advantages of deep learning techniques in handling high-dimensional data.

2. Dataset

The dataset utilized in this study is the "BIRDS 525 SPECIES" dataset, sourced from Kaggle. This comprehensive dataset comprises a rich collection of bird images, encompassing a total of 525 distinct species. Each species is represented through high-quality, colored images, providing a diverse and extensive range of bird species imagery. The dataset is particularly well-suited for classification tasks in machine learning and deep learning due to its variety and volume. Dataset of 525 bird species, 84635 training images, 2625 test images(5 images per species) and 2625 validation images is a very high quality dataset where there is only one bird in each image and the bird typically takes up at least 50% of the pixels in the image. As a result even a moderately complex model will achieve training and test accuracies in the mid 90% range. Note that all images are original and not created by augmentation. All images are $224 * 224 * 3$ color images in jpg format. Data set includes a train set, test set and validation set. Each set contains 525 sub directories, one for each bird species.

3. Problem Definition

This project embarks on a comparative study of classification methods applied to a real-world dataset of bird species images. The dataset comprises images from five different bird species, posing a unique challenge due to the inherent complexity of natural images. This complexity is primarily attributed to the high dimensionality of the image data, where each image can be represented as a matrix I of pixel values. The classification task involves identifying a function $f : I \to C$, where C is the set of class labels corresponding to different bird species.

The first dimension of the problem involves assessing the performance of classification algorithms on the original, high-dimensional data. This aspect is crucial, as the effectiveness of any classification method like KNN largely depends on its ability to handle the high-dimensional feature space $F \subset R^n$, where n represents the number of features (pixels, in this case). KNN operates on the principle of feature similarity, where the classification of an unknown sample is inferred based on the majority class of its k nearest neighbors in this feature space. The performance metric here is classification accuracy, which is the proportion of correctly classified instances out of the total instances.

The second part of the problem investigates the effect of dimensionality reduction techniques, such as Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE), on classification performance. PCA reduces the dimensionality by transforming the original variables into a new set of uncorrelated variables (principal components), capturing the maximum variance in the data. Mathematically, PCA involves computing the eigenvectors and eigenvalues of the data covariance matrix and then projecting the data onto the space spanned by the top eigenvectors. On the other hand, t-SNE focuses on maintaining the local structure of the data in the reduced-dimensional space, making it particularly suitable for visualizing high-dimensional data in two or three dimensions. The transformed feature space through PCA and t-SNE is denoted as $F_{reduced} \subset R^m$ with $m < n$. The KNN classifier's performance in this reduced space is then compared against its performance in the original space.

In addition to these traditional methods, the study also incorporates a CNN model, a deep learning approach known for its prowess in image classification tasks. Unlike KNN, CNN automatically learns hierarchical feature representations from the data, which are then used for classification. The CNN model's effectiveness is quantified by its classification accuracy on the original dataset, and its performance is benchmarked against the results obtained from KNN. The CNN model's architecture, comprising layers with weights W and biases b, is optimized through backpropagation and gradient descent to minimize a loss function, typically the cross-entropy loss $L(W, b)$. The study also explores hyperparameter tuning in CNN to enhance its performance, examining parameters such as the number of layers, filter sizes, and learning rate.

4. Technical Approach

In this section, we delve into the mathematical formulation of each part of this study.

4.1. KNN

Given a dataset D with N data points $x_1, x_2, ..., x_N$, and a distance metric d, the "kin" of a data point x_i is determined by finding the set of points in D that are closest to x_i according to d. d is euclidean distance in this study. For a specified k, which represents the number of neighbors, the kin of x_i is the set $K(x_i)$ such that: $K(x_i) = \{x_j \in D \mid d(x_i, x_j)$ is among the k smallest distances from $x_i\}$. For each data point x_i, its classification is determined by applying a majority voting mechanism to the labels of the points in $K(x_i)$, and the most frequent label among these neighbors is assigned to x_i. k is equal to 5 in this project as we consider 5 different species of the birds. Each image ($x_i \in D$) has dimension of $64 * 64 * 3$, and we reshape it to a vector of size

$1 * 122888$ to be used in our KNN model. So, our train dataset has dimension of $843 * 12288$. We use class centroids, derived from the training phase, for labeling the test dataset.

4.2. Dimension Reduction

4.2.1 PCA

PCA is a statistical technique used to emphasize variation and bring out strong patterns in a dataset. It converts the original correlated features into a set of linearly uncorrelated variables known as principal components. These principal components are obtained in such a way that the first few retain most of the variation present in the original dimensions. The data is first standardized, especially important when the original features have different scales. This is achieved by subtracting the mean and dividing by the standard deviation for each feature. We compute the covariance matrix Σ to understand how the variables of the input data are varying from the mean with respect to each other. The covariance matrix is given by: $\Sigma = \frac{1}{n-1}(X - \bar{X})^T(X - \bar{X})$, where X is the matrix of input data, $\bar{X}$ is the mean vector, and n is the number of data points. The next step is to compute the eigenvalues and eigenvectors of the covariance matrix. These eigenvectors determine the directions of the new feature space, and the eigenvalues determine their magnitude. In essence, the eigenvectors represent the principal components. The eigenvalues are sorted in descending order, and the corresponding eigenvectors are aligned accordingly. The first few eigenvectors are selected based on the desired number of principal components, capturing most of the variance in the data. The final step involves projecting the original data onto the space spanned by the selected principal components. This results in a new dataset with reduced dimensions $F_{reduced}$. Utilizing the well-structured libraries in Python, specifically the Scikit-learn package, streamlined the process, allowing me to efficiently achieve the objectives of this section without manually performing each step. In this study, we retain 95% of the variation using PCA. Figure 1 shows original images alongside their PCA-transformed counterparts (refer to Figure 2). The image transformed using PCA, while lower in quality, effectively captures the key features of the original, particularly the distinct shape of the bird's body.

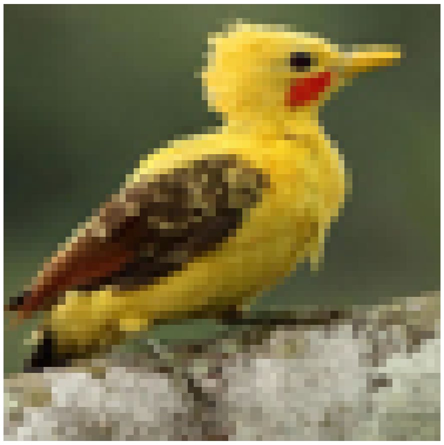

Figure 1. Original image before applying PCA

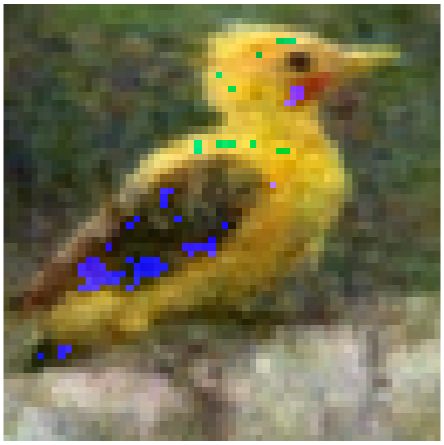

Figure 2. Transformed image after applying PCA

4.3. t-SNE

t-SNE is a non-linear technique used for dimensionality reduction and visualization of high-dimensional datasets. It differs from PCA in its approach, focusing on preserving the local structure of the data while embedding it into a lower-dimensional space. t-SNE starts by converting the Euclidean distances between data points in the high-dimensional space into conditional probabilities that represent similarities. These probabilities are then used to construct a similar distribution in a low-dimensional map. The optimization of t-SNE involves minimizing the Kullback-Leibler divergence between the two distributions with respect to the locations of the points in the map. This process allows t-SNE to capture complex nonlinear structures, which is particularly useful for datasets where the local relationships between data points are of interest. Due to the stochastic nature of t-SNE, it often reveals patterns and clusters in the data that are not apparent in linear dimensionality reduction techniques like PCA. The Python ecosystem, particularly the Scikit-learn library, provides efficient implementations of t-SNE, facilitating its integration into our workflow. By applying t-SNE to our dataset, we have successfully reduced its dimensions while maintaining the essential structures and relationships inherent in the original data. We experimented with using both two and three components in t-SNE for our study. We found that three components were more effective. However, t-SNE did not achieve clear separation of data in the lower dimension as we had hoped. In Figure 3, you can observe the 3D plot of these lower-dimensional embeddings of our images. A possible explanation for this outcome is the high similarity among our images. Even with dimensionality reduction, the images remain similar in the lower-dimensional space. This is likely because t-SNE aims to preserve the local structure of the data, meaning that if images are similar in the original high-dimensional space, they tend to remain similar after the reduction.

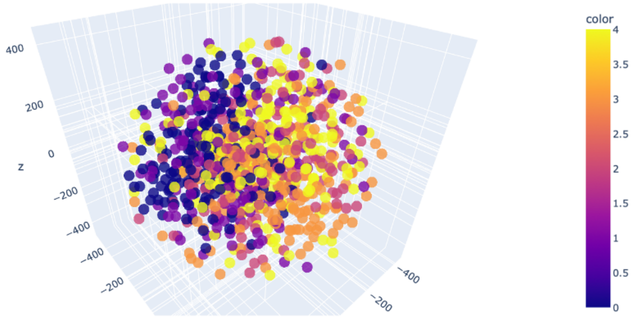

Figure 3. 3D plot of the lower-dimensional embeddings resulted from t-SNE

4.4. CNN

In our project, we utilized a custom-built Convolutional Neural Network named SimpleCNN for classifying images into 5 distinct bird species. The SimpleCNN model is defined in PyTorch and consists of a sequence of layers designed to extract and learn features from input images of size $64 \times 64 \times 3$. The first layer, ‘conv1‘, is a 2D convolutional layer with 32 filters of kernel size 3 and padding 1, designed to process the input RGB images. This is followed by two more convolutional layers, ‘conv2‘ and ‘conv3‘, with 64 and 128 filters respectively, each also using a kernel size of 3 and padding of 1. These convolutional layers are responsible for detecting increasingly complex features in the images. After each convolutional layer, a max-pooling layer (‘pool‘) with a size of 2×2 is applied, which reduces the spatial dimensions of the feature maps by half while retaining the most significant features. This reduction is crucial for managing computational complexity and improving the model’s efficiency. The output from the last pooling layer is flattened into a one-dimensional vector and passed through two fully connected layers (‘fc1‘ and ‘fc2‘). The first fully connected layer, ‘fc1‘, reduces the dimensionality from $128 \times 8 \times 8$ to 512, introducing a high-level abstraction of the image features. To prevent overfitting, a dropout layer with a dropout rate of 0.5 is used after ‘fc1‘, randomly zeroing out some of the features. Finally, the ‘fc2‘ layer maps these features to the number of classes, which corresponds to the different bird species in this study. The output of ‘fc2‘ represents the model’s predictions for each class. Through this architecture, SimpleCNN learns to discern intricate patterns specific to each bird species, leveraging both the depth and breadth of convolutional and pooling layers for effective feature extraction and classification.

5. Results

In this section, we provide an overview of the results obtained from our models and conduct a comparative analysis. Table 1 shows the performance of the KNN model applied to our dataset in various scenarios: following PCA, t-SNE with two and three components, and on the original images with different numbers of neighbors (2, 3, 6). Our analysis indicates that the optimal strategy involves using t-SNE with three components combined with KNN using 6 neighbors for distance calculations. This particular combination leads to the best results, attaining a 60% accuracy on the test dataset.

Table 2 and Table 3 present a clear comparison of our CNN model’s performance, organized by learning rate and optimizer. The analysis shows that the Adam optimizer slightly outperforms the AdamW optimizer on both training and test datasets. The best results of the CNN model using both Adam and AdamW optimizers are detailed in Table 2. Notably, we achieved a high accuracy of 92% on the test dataset, which is significantly better than the top result from the KNN model. Additionally, after experimenting with various learning rates, the optimal value was found to be 0.0005 (Table 2). Figure 4 clearly illustrates the accuracy of our CNN model on both training and test datasets across various learning rates.

Additionally, the Figure 5 provides a clear depiction of the CNN model’s loss values over each epoch in training process.

6. Conclusion and Future Steps

In our study, we evaluated the effectiveness of traditional classification techniques versus advanced deep

Table 1. Results of KNN with/ without different dimension reduction methods. tSNE2 (tSNE3) means tSNE method with two (three) components.

Model	Dim Reduction	n neighbors	Train Accuracy	Test Accuracy
KNN	PCA	2	91.10	52.00
KNN	PCA	3	72.95	44.00
KNN	PCA	6	62.87	44.00
KNN	tSNE2	2	75.20	48.00
KNN	tSNE2	3	71.17	52.00
KNN	tSNE2	6	65.00	48.00
KNN	tSNE3	2	81.96	52.00
KNN	tSNE3	3	76.39	44.00
KNN	tSNE3	6	65.12	60.00
KNN	None	2	89.79	56.00
KNN	None	3	74.13	48.00
KNN	None	6	62.51	44.00

Table 2. Best Results of CNN model on original images using Adam and AdamW

Model	learning rate	Optimizer	Train Accuracy	Test Accuracy
CNN	0.0008	AdamW	97.74	88.00
CNN	0.0005	Adam	99.88	92.00

Table 3. CNN Accuracy at Different Learning Rates using Adam

Index	Learning Rate	Train Acc (%)	Test Acc (%)
0	0.0002	99.762752	84.0
1	0.0005	99.881376	92.0
2	0.0008	96.915777	80.0
3	0.0010	99.288256	84.0
4	0.0030	43.653618	28.0
5	0.0060	48.635824	36.0
6	0.0090	96.322657	56.0
7	0.0100	42.467378	36.0
8	0.0300	23.368921	20.0
9	0.0600	23.368921	20.0
10	0.0900	23.368921	20.0
11	0.1000	23.368921	20.0
12	0.5000	16.963227	20.0

Table 4. Best models

Model	Train Accuracy	Test Accuracy
KNN, dim reduction = tSNE3, n neighbors = 6	65.12	60.00
CNN, learning rate = 0.0005, optimizer = Adam	99.88	92.00

learning approaches in image classification. We applied dimensionality reduction algorithms, such as PCA and t-SNE, to enhance the KNN model's performance. Our findings indicate that while t-SNE aids the KNN model in classification, the improvement is not substantial compared to using KNN on the original dataset. However, our CNN model exhibited exceptional performance in image classification, confirming that deep neural networks, particularly CNNs, are highly effective for this task. This is largely due to their ability to capture spatial relationships within images. We achieved impressive accuracies of 99.88% and 92% on the training and test datasets, respectively. For future steps, we can explore the application of other deep neural network models to our dataset. Additionally, expanding the scope to include more bird species, rather

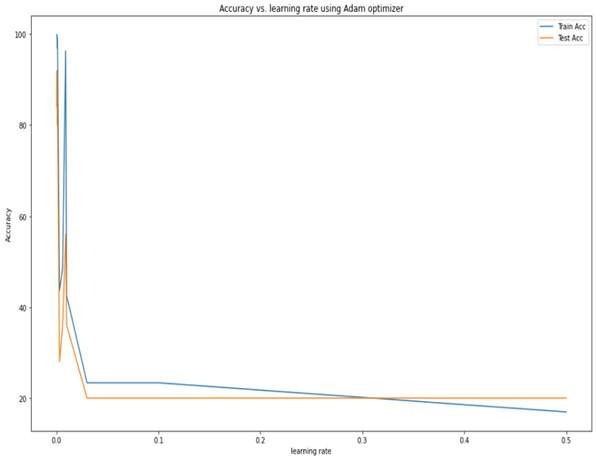

Figure 4. Accuracy of CNN model per various learning rates

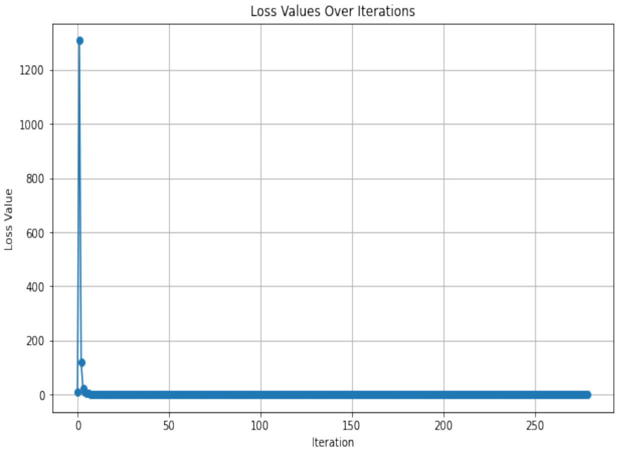

Figure 5. CNN Loss Values Over Iterations

than the current five, could lead to even more robust and accurate models for image classification tasks.

References

[1] Waldemar Ratajczak Andrzej Maćkiewicz. Principal components analysis (pca). Computers Geosciences, 1993. 1

[2] P. Anusha and K. ManiSai. Bird species classification using deep learning. 2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP), 2022. 1

[3] Mucahid Barstugan, Umut Ozkaya, and Saban Ozturk. Coronavirus (covid-19) classification using ct images by machine learning methods. 2020. 1

[4] Pádraig Cunningham and Sarah Jane Delany. k-nearest neighbour classifiers - a tutorial. ACM Computing Surveys, 54(6):1–25, jul 2021. 1

[5] A. Sufian F. Sultana and P. Dutta. Advancements in image classification using convolutional neural network. nternational Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), 2018. 1

[6] Wei B Zheng Y Han, Z. Breast cancer multi-classification from histopathological images with structured deep learning model. Sci Rep 7, 2017. 1

[7] Geoffrey Hinton Laurens van der Maaten. Visualizing data using t-sne. Journal of Machine Learning Research 9, 2009. 1

[8] Shanmugasundaram Hariharan Muthukrishnan Ramprasath, M.Vijay Anand. Image classification using convolutional neural networks. Journal of Machine Learning Research 9, 2018. 1