A Comparative Analysis of Image Classification using the CIFAR10 Dataset COMP 6721 Applied Artificial Intelligence Final Report

GROUP- G(22) : Kalinga Swain 40226333 Nadira Anjum Nipa 40262312, Munj Bhavesh Nayak 40195590, Afif Bin Kamrul 40202952

Guided By: Professor Arash Azarfar

1. Abstract

Image classification is a prominent field within computer vision that entails the assignment of categorical labels to input images. This task is typically accomplished through the utilization of machine learning techniques, specifically designed to process and extract meaningful features from images. In this particular research endeavor, various machine learning and deep learning algorithms have been applied to classify images sourced from the CIFAR 10 dataset. A comprehensive comparative analysis of supervised, semi-supervised, and deep learning algorithms for image classification has been presented. The algorithms employed include Naive Bayes, Decision Tree, and Convolutional Neural Network (CNN). Performance evaluation metrics such as Precision, Recall, F1-Score, and Overall Accuracy have been utilized to assess and compare the performance of these algorithms. The findings indicate that Decision Tree outperforms Naive Bayes, while CNN surpasses all other algorithms in terms of classification performance. This research project serves as a valuable educational resource for individuals seeking to gain an understanding of the diverse algorithms employed in image classification.

2. Introduction

Image classification involves categorizing images into pre-defined groups or categories and is used in various applications like object recognition, facial recognition, autonomous driving, and medical image analysis. However, image classification is a complex task with several challenges in real-world scenarios. These challenges include dealing with a large and diverse image dataset that may not be clean or relevant. Finding the right balance between model complexity and generalization is also a challenge, along with selecting and interpreting appropriate evaluation metrics that have different assumptions, limitations, and trade-offs. The performance of image classification depends on effectively addressing these challenges. To improve the precision and effectiveness of image classification, numerous machine learning algorithms and deep learning techniques have been developed. Our project focuses on multiclass classification using the CIFAR 10 dataset. We explore different approaches and algorithms for image classification and present the performance of various models based on evaluation metrics such as Precision, Recall, F1-Score, and accuracy.

3. Literature Review

Numerous studies have been conducted on image classification. Prasant [2] utilized the Naive Bayes classifier to predict whether an individual's income exceeds 50K per year, focusing solely on the performance of this algorithm with numeric data. Bianca et al. [3] explored three classification models: logistic regression, random forest, and a convolutional neural network (CNN), presenting their respective results. Jagandeep [4] took a distinct approach by employing a semi-supervised method for image classification. Ebrahim [5] applied a CNN to the CIFAR 10 dataset and reported their findings. However, none of these works provided a comprehensive comparison of different algorithms encompassing supervised, semi-supervised, and deep learning methods. In our project, we have adopted diverse approaches for image classification, including supervised, semi-supervised, and deep learning techniques. We have applied various algorithms such as Naive Bayes, Decision Tree, and CNN, and conducted a comparative analysis of their performance.

4. Possible Methodology

The primary objective of our study is to train various classifiers for the task of CIFAR-10 image classification. This task falls under the domain of supervised and semi-supervised learning algorithms. Supervised learning involves training a model to approximate a function that maps input data to desired output labels, relying on labeled data

COMP-6721 PROJECT #*****

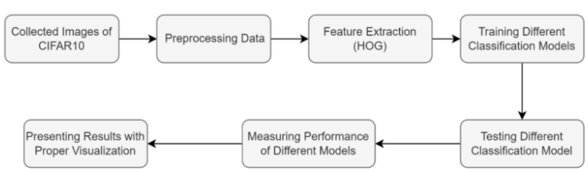
where both input features and corresponding outputs are provided. In contrast, semi-supervised learning focuses on leveraging a limited set of labeled data points to label the majority of unlabeled data points, allowing the model to generalize and make predictions for the unlabeled data.

For our project, we utilized the CIFAR-10 dataset, which is a widely recognized computer vision dataset. It consists of 60,000 color images, each with a fixed size of 32x32 pixels, distributed among 10 classes. There are 6,000 images per class, with the classes including airplane, automobile, bird, cat, dog, frog, horse, ship, and truck.

The image classification process encompasses several essential steps, namely dataset preparation, feature extraction, model training, classification, and model evaluation. These steps are illustrated in Figure 1 (reference).

To extract meaningful features from the images, we employed the Histogram of Oriented Gradients (HOG) technique, which is a commonly used method for object detection and image classification. HOG utilizes histograms to represent the distribution and orientations of gradients, computed as the derivatives of the image in the horizontal and vertical directions. Figure 2 (reference) illustrates the process, where the first image represents the original image, and the second image shows the result after applying the HOG technique.

Prior to feeding the data into the models, we partitioned the dataset into two subsets: a training set used for model training and a test set employed to evaluate the model's performance. The dataset was split using an 80:20 ratio, where 80% of the data was allocated to the training set, and the remaining 20% was allocated to the test set.



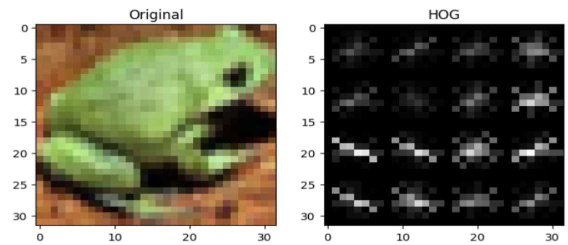
(a)

Figure 1. Methodology of Image Classification

5. Classification Algorithms

5.1. Supervised and Semi-supervised learning using Decision Tree

The Decision Tree (DT) algorithm is a classification method that recursively partitions a dataset based on the values of input features. It makes decisions at intermediate nodes and assigns class labels to the terminal nodes or leaves. DT is a non-parametric, efficient, and simple algorithm for classification tasks.



(a)

Figure 2. Feature Extraction from the image

In our supervised learning experiments, we trained a decision tree model for image classification using the CIFAR-10 dataset. The model achieved an accuracy of 31.4
Hyperparameter tuning involves searching for optimal values of parameters that govern the model's behavior. We varied the maximum depth, which limits the number of splits in the tree, and the number of estimators, which affects the ensemble size. By systematically evaluating different combinations of these hyperparameters on a validation set, we selected the best-performing model based on its accuracy.
The tuned decision tree model showed improved performance on the test set, demonstrating the effectiveness of hyperparameter tuning in optimizing the model for image classification. For the decision tree, we focused on tuning the following hyperparameters:

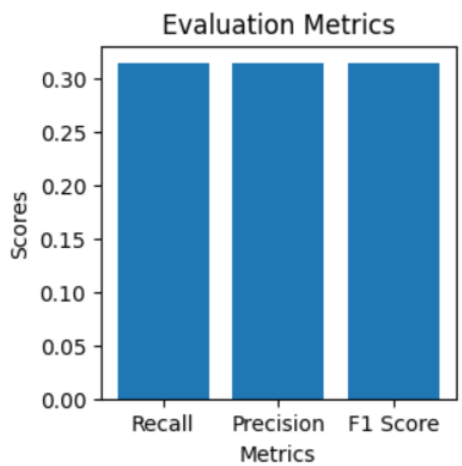
- **Max_depth**: This parameter determines the maximum depth of the decision tree. A higher value can lead to overfitting, while a lower value can result in underfitting. Through experimentation, we determined that a maximum depth of 200 produced the best results for our decision tree model.

- **Min_samples_split**: It specifies the minimum number of samples required to split an internal node. Increasing this value helps prevent overfitting by creating more generalized nodes. In our model, we set the minimum samples split to 100.

- **Min_samples_leaf**: This parameter sets the minimum number of samples required to be at a leaf node. A higher value promotes generalization, while a lower value allows for more specific leaves. We used a minimum samples leaf of 10 in our decision tree model.

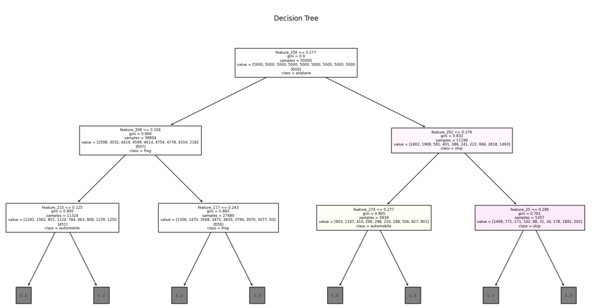
- **Max_features**: It determines the number of features to consider when searching for the best split. In our approach, we did not restrict the number of features and utilized all the available features from the images in our model.

- **Criterion**: This parameter specifies the function used to measure the quality of a split. We employed the "gini" criterion, which measures the Gini impurity.



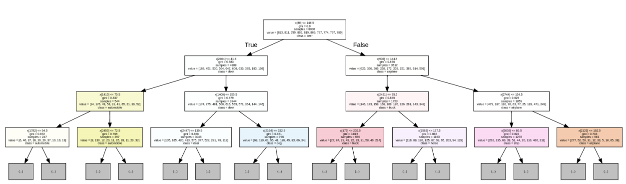
(a)

Figure 3. HOG Implementation Performance Metric



(a)

Figure 4. Generated Decision Tree after HOG Feature Extraction



(a)

Figure 5. Decision Tree for Supervised Learning

In our semi-supervised learning experiments, we observed a increase in the test set accuracy to 37.69% for the decision tree model. Semi-supervised learning involves assigning labels to unlabeled data points based on the knowledge gained from a limited number of labeled data points. For this model, we set the hyperparameters to a maximum depth of 5, a minimum samples split of 10, and a minimum samples leaf of 5. Additionally, we implemented a Random Forest (RF) classifier, which achieved an accuracy of 40.25% on the data.
In conclusion, our experiments demonstrated that hyperparameter tuning, along with the utilization of ensemble methods such as Random Forest, can improve the accuracy of the decision tree model for image classification tasks. The selection of appropriate hyperparameters is crucial in optimizing the performance of the decision tree algorithm, allowing it to achieve better results in the classification of images.

5.2. Gaussian Naive Bayes

The Naive Bayes (NB) classifier is a widely utilized supervised machine learning algorithm for classification tasks. It is a probabilistic classifier that applies Bayes' theorem. In our study, we implemented the Gaussian Naive Bayes classifier to evaluate its performance in image classification tasks. However, it is important to note that the NB classifier makes a strong assumption of feature independence in the input data, which may not always hold true in real-world scenarios. This assumption implies that each feature contributes independently to the probability of a particular class. However, in image classification, this assumption is often violated as there are dependencies and correlations between different image features.

Upon experimentation, we observed that the NB classifier did not yield satisfactory results in our image classification tasks. The accuracy achieved by the NB classifier was approximately 28%. This lower accuracy can be attributed to the limitations of the independence assumption, which restricts the classifier's ability to capture complex relationships and dependencies between image features. Consequently, the NB classifier may struggle to accurately classify images with interdependent features.

Therefore, while the NB classifier is a simple and computationally efficient algorithm, it may not be the most suitable choice for image classification tasks, where feature dependencies play a crucial role. Other more sophisticated classifiers that can capture the intricate relationships between image features, such as convolutional neural networks (CNNs), have demonstrated superior performance in image classification tasks.

5.3. Deep Neural Network

Deep learning algorithms are a class of machine learning models that leverage artificial neural networks to automatically learn hierarchical representations of data. Convolutional Neural Networks (CNNs) are a popular type of deep learning algorithm, particularly effective for tasks like image classification and computer vision. Unlike traditional methods that relied on manual feature engineering, CNNs revolutionized these tasks by automatically learning and extracting features from input data.

By utilizing multiple layers of interconnected neurons, deep learning algorithms can capture intricate patterns and representations from raw input data. CNNs, specifically designed for grid-like data such as images, employ convolutional layers to apply filters and capture local patterns and spatial relationships. These learned features are then combined through pooling layers, reducing dimensionality while retaining important information.

The power of deep learning lies in its ability to learn representations directly from raw data, eliminating the need for explicit feature engineering. Through the training process, deep learning models iteratively optimize their parameters, adjusting the weights based on the differences between predicted and true labels. This optimization process, known as backpropagation, enables the model to gradually improve its performance on the task at hand.

n our project, we focused on utilizing the ResNet18 architecture, which is a member of the ResNet (Residual Network) family. ResNet architectures are widely recognized for their effectiveness in training deep neural networks. ResNet-18, specifically, is composed of 18 layers, including convolutional layers, pooling layers, fully connected layers, and crucially, residual connections.

Residual connections are a key feature of ResNet architectures and address the challenge of vanishing gradients often encountered when training deep neural networks. The core component of ResNet-18 is the residual block, which consists of two consecutive convolutional layers followed by an element-wise addition operation that combines the block's output with the input to create a residual mapping. This skip connection allows the network to learn residual information, capturing the discrepancy between the desired output and the current prediction. By leveraging these residual connections, the optimization process becomes more manageable, leading to improved overall performance of the network.

By incorporating the ResNet18 architecture into our project, we aimed to benefit from its ability to effectively train deep neural networks. The use of residual connections in ResNet-18 helps mitigate the vanishing gradient problem and facilitates the training of deeper models. This architecture has demonstrated remarkable performance in various computer vision tasks, including image classification, object detection, and semantic segmentation.

In our project, optimizing the performance of our deep learning model was a crucial aspect. To achieve this, we employed stochastic gradient descent (SGD), which is a widely used optimization algorithm for training deep learning models. SGD iteratively updates the model's parameters by computing gradients on small batches of data, making it computationally efficient and suitable for large datasets. By using SGD, we aimed to improve the model's convergence and overall accuracy.

In addition to SGD, we conducted various hyperparameter tuning techniques to fine-tune our model. Hyperparameters are configuration settings that determine the behavior and performance of the model. They include parameters such as the learning rate, momentum, and weight decay, among others. By adjusting these hyperparameters, we aimed to find the optimal configuration that maximized the model's performance on our specific task. This involved experimenting with different combinations of hyperparameter values and evaluating their impact on the model's accuracy and convergence.

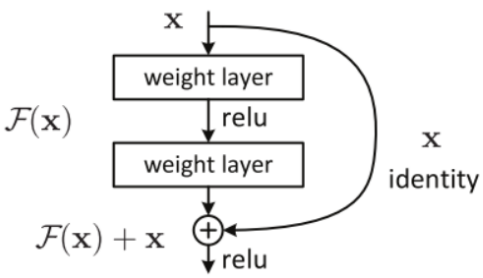
During the training process, we trained the ResNet18 model for a specific number of epochs. An epoch refers to a complete pass over the entire dataset during training. By iterating over the dataset multiple times, we aimed to enhance the model's ability to learn complex patterns and improve its accuracy. In our case, we trained the ResNet18 model for 10 epochs, allowing it to learn from the data and progressively refine its predictions.

By utilizing deep learning algorithms, specifically the ResNet18 architecture, and applying appropriate optimization techniques such as SGD and hyperparameter tuning, we sought to achieve superior performance in our image classification task. The ResNet18 architecture's ability to automatically learn hierarchical representations, combined with the optimization techniques employed, allowed our model to effectively capture intricate patterns in the data and make accurate predictions.

Overall, the utilization of deep learning algorithms, along with appropriate optimization strategies, plays a pivotal role in achieving enhanced performance in image classification tasks. Through the integration of advanced architectures and optimization techniques, we aimed to maximize the model's potential and improve its accuracy.

6. Results

In this project, we employed various image classification algorithms and conducted a comparative analysis of their performance. The dataset utilized in our study had preassigned labels corresponding to the image classes, which were utilized during the training phase of our classification algorithms. To evaluate the effectiveness of these algorithms, we employed a separate test set and assessed their performance using metrics such as accuracy, recall, precision, and F1-score. This comprehensive analysis allowed us to determine the most suitable classification algorithm for image classification tasks. Our implementation of the



(a)

Figure 6. A basic structure of Residual Network

classification models was carried out using Python's Scikit-learn library.

To gain insights into the classification results and facilitate visual interpretation, we utilized t-distributed Stochastic Neighbor Embedding (t-SNE) for dimensionality reduction and visualization. Additionally, we generated confusion matrices to gain a deeper understanding of the classification algorithm performance. The confusion matrix visually represents the predicted labels (Output Class) along the rows or x-axis and the true labels (Target Class) along the columns or y-axis. Correctly classified observations are represented by the diagonal cells, while misclassified observations are reflected in the off-diagonal cells. Figures 5 to 9 showcase the confusion matrices depicting the classification algorithm performances on the test data.

Based on our findings, we observed that the Convolutional Neural Network (CNN) exhibited the highest performance among the classification algorithms, achieving an overall accuracy of 79.57% (Figure 5). In the case of supervised learning, the Decision Tree (DT) algorithm achieved an overall accuracy of 31.42% (Figure 6), while the Naive Bayes (NB) algorithm attained an accuracy of approximately 28% (Figure 7). In the context of semi-supervised learning, the DT model achieved an accuracy of 26.65% (Figure 8), whereas the Random Forest (RF) model achieved an accuracy of 40.35% on the test set (Figure 9).

These results provide valuable insights into the performance of different image classification algorithms and guide the selection of suitable models for specific classification tasks.

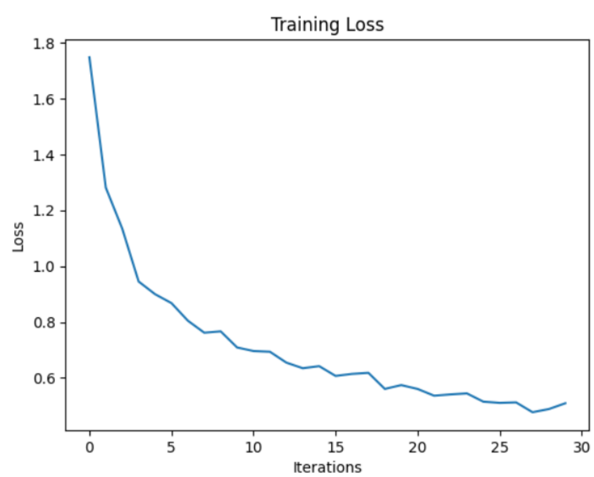
As expected, the Convolutional Neural Network (CNN) demonstrated superior performance in terms of Recall, Precision, and F1-score metrics, reaffirming its effectiveness in image classification tasks.

Our project presented a comprehensive analysis of multiple multiclass image classification algorithms, providing valuable insights for researchers interested in understanding the performance characteristics of these algorithms across diverse scenarios. This comparative overview serves as a valuable resource, enabling researchers to make informed decisions when selecting the most appropriate image classification algorithms for their specific requirements. The findings and observations presented in this study contribute to the existing body of knowledge in the field of image classification and offer guidance for future research endeavors.

```
Precision: 0.7998
Recall: 0.7986
F1 Score: 0.7985
Accuracy: 79.57%
Confusion Matrix:
[[809  11  31  17  24   3   2   9  69  25]
 [ 15 855   5   5   0   2   8   2  31  77]
 [ 41   6 770  38  56  21  39  22   4   3]
 [ 13   4  64 654  58 111  45  28  10  13]
 [ 14   1  55  44 798  18  28  33   6   3]
 [  5   4  41 193  31 653  15  50   3   5]
 [  6   4  43  50  24  16 843   6   3   5]
 [ 12   1  12  45  38  27   5 844   2  14]
 [ 35  14   9  14   6   0   2   6 901  13]
 [ 28  48   3   7   1   3   4  12  35 859]]
```

(a)

Figure 7. Performance of CNN (Supervised)

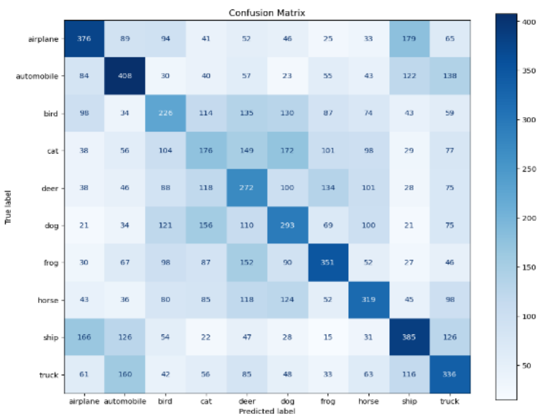


(a)

Figure 8. Loss-Iteration Graph of CNN

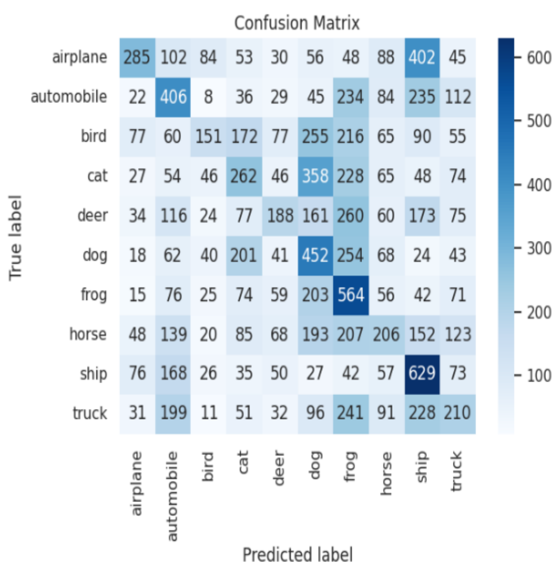
Table 1. Performance Metrics for Different Algorithms

Algorithm	Recall	Precision	F1-score	Accuracy
NB	0.2801	0.2930	0.2664	0.2794
DT (Super.)	0.2438	0.2425	0.2438	0.2424
DT (Super. HOG)	0.3142	0.3150	0.3141	0.3142
DT (Semi-super.)	0.4101	0.1504	0.1588	0.5281
CNN	0.7986	0.7998	0.7985	0.7957



(a)

Figure 9. Performance of DT (Supervised)



(a)

Figure 10. Performance after HOG Feature Extraction

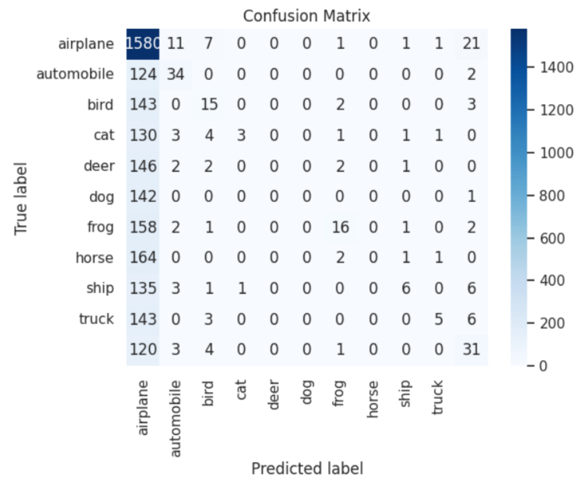
7. Analysis of the Performance of the Models

7.1. Decision Tree for Semi-Supervised learning

Accuracy: This model achieved an accuracy of 52.81%. It correctly predicted the labels for 31% of the samples in the dataset.

Precision: The precision of 15.04% indicates that when this model predicts a sample as positive, it is correct only 15.04% of the time.

Recall: The recall of 41.01% means that this model correctly identifies 41.01% of the positive samples in the



(a)

Figure 11. Performance of DT (Semi-supervised)

dataset.

F1 Score: The F1 score of 15.88% is the harmonic mean of precision and recall. It provides a balanced measure of the model's performance.

7.2. Decision Tree for Supervised Learning

Accuracy: This model achieved an accuracy of 24.24%, which is relatively lower than the other models.

Precision: The precision of 24.25% indicates that when this model predicts a sample as positive, it is correct only 24.25% of the time.

Recall: The recall of 24.38% means that this model correctly identifies 24.38% of the positive samples in the dataset.

F1 Score: The F1 score of 24.38% represents a balance between precision and recall, considering both aspects of the model's performance.

7.3. CNN

Accuracy: This model achieved an accuracy of 79.57%, which is significantly higher than the previous models.

Precision: The precision of 79.98% indicates that when this model predicts a sample as positive, it is correct 79.98% of the time.

Recall: The recall of 79.86% means that this model correctly identifies 79.86% of the positive samples in the dataset.

F1 Score: The F1 score of 79.85% represents a balance between precision and recall, considering both aspects of the model's performance.

Overall, the CNN model outperforms the other two models in terms of accuracy, precision, recall, and F1 score. It achieves the highest accuracy and provides a better balance between precision and recall. The Decision Tree with HOG (Semi-Supervised) model performs better than the Supervised Learning Decision Tree with Hyperparameter Tuning in terms of accuracy and precision, but it falls behind in terms of recall and F1 score.

8. References

1. https://developers.google.com/machine-learning/practica/image-classification#:~:text=Image%20classification%20is%20a%20supervised,them%20using%20labeled%20example%20photos.
2. https://www.kaggle.com/code/prashant111/naive-bayes-classifier-in-python.
3. https://medium.com/@bian0628/image-classification-cifar-10-dc1c23db46d5.
4. https://jagan-singhh.medium.com/semi-supervised-learning-19e431be16e.
5. https://medium.com/@ebrahimhaqbhatti516/convolutional-neural-network-on-cifar-10-dataset-1904560de3ac.
6. https://www.quantstart.com/articles/Beginners-Guide-to-Decision-Trees-for-Supervised-Machine-Learning/.
7. https://scikit-learn.org/stable/modules/tree.html.
8. https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-cifar-10-photo-classification/.
9. https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/.
10. https://towardsdatascience.com/4-pre-trained-cnn-models-to-use-for-computer-vision-with-transfer-learning-885cb1b2dfc.