Noise-Resistant Deep Learning: A Novel Approach for Image Classification

A Project Report Submitted

for the Course

MA499 Project II

by

Abana Sidhu : 200123001

Khushi Mineeyar : 200123033

to the

DEPARTMENT OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, INDIA

April 2024

CERTIFICATE

This is to certify that the work contained in this project report entitled "Noise-Resistant Deep Learning: A Novel Approach for Image Classification" submitted by Abana Sidhu (Roll No.: 200123001) and Khushi Mineeyar (Roll No.: 200123033) to the Department of Mathematics, Indian Institute of Technology Guwahati towards partial requirement of Bachelor of Technology in Mathematics and Computing has been carried out by him/her under my supervision.

It is also certified that, along with literature survey, a few new results are established/computational implementations have been carried out/simulation studies have been carried out/empirical analysis has been done by the student under the project.

Turnitin Similarity: 18%

Guwahati - 781 039 (Dr. Rajen Kumar Sinha)

April 2024 Project Supervisor

ABSTRACT

The main aim of the project is to explore the integration of convolutional neural networks (CNNs) with the DECODE framework to enhance medicinal plant classification accuracy, particularly in managing noisy mislabeled datasets. Through the integration of DECODE with AlexNet, the study demonstrates significant mitigation of adverse effects from mislabeled data, ensuring learning from high-confidence samples. Comparative analysis under varying noise levels highlight DECODE's outperformance of traditional methods, maintaining higher accuracy. Additionally, experimentation with weighting functions reveals the efficacy of the negative exponential function in enhancing model performance under high noise levels. This research underscores the importance of addressing data quality issues for high accuracy in plant classification, with the DECODE framework offering a promising solution to challenges posed by noisy mislabeled datasets, thereby improving the reliability of automated plant classification systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent times, there has been a notable rise in the success of deep convolutional neural networks for tasks like image classification. Current training methods presume clean datasets for model training, which perform well in benchmark settings but often fall short in real-world scenarios due to the expensive and labor-intensive nature of gathering ample clean data.

On the other hand, collecting labeled samples from Internet, e.g., retrieving images from image search engine like Google using target keyword, turns out to be a convenient and economical way but unfortunately, Web images are very noisy in practice. The significant drop in accuracy, even with a minor 5% to 10% label noise, underscores the critical need for robust label noise-resistant classification in data-driven deep learning methods [2].

1.1 Motivation and Contribution

The significant drop in accuracy, even with a minor 5% to 10% label noise, underscores the critical need for robust label noise-resistant classification in data-driven deep learning methods.

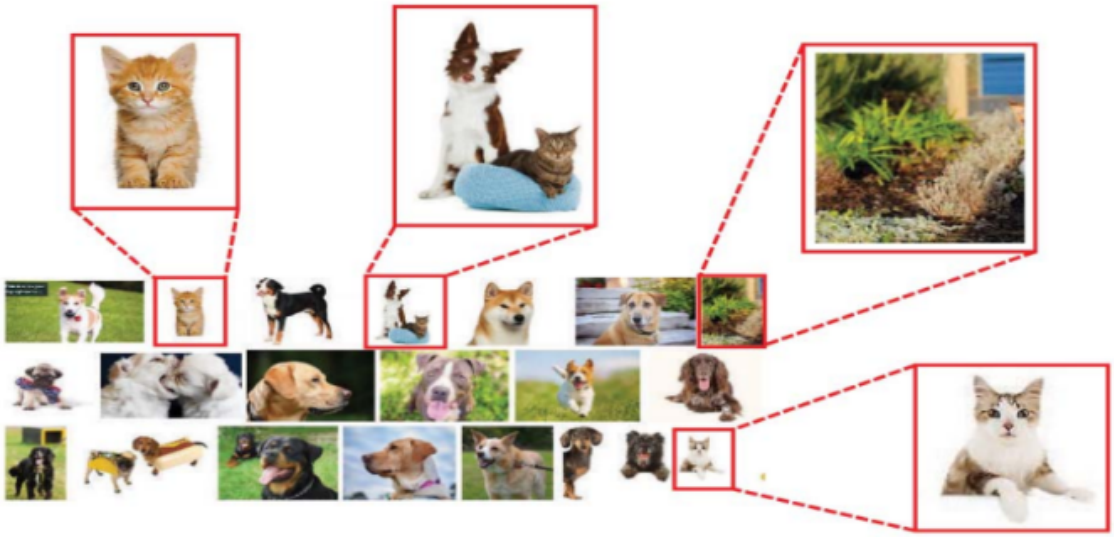

Figure 1.1: *'Dog' keyword search in Google Images surprisingly shows non-dog images, such as cats and backgrounds, in top-ranked results.*

DEep COnfiDEnce network (DECODE) [3] is a novel approach to tackle this issue. It incorporates a confidence evaluation strategy that assesses the likelihood that a sample is mislabeled based on the distribution of mislabeled data. With this confidence measure, weights are assigned to individual samples such that less attention is paid to data with low confidence, which is more likely to be noisy and more weights are assigned to high-confidence samples, that are more likely to have correct labels [**?**]. Thus the approach enhances the accuracy of deep models trained on noisy data . DECODE is versatile and can be seamlessly integrated with various popular CNN architectures like LeNet, AlexNet and DenseNet.

1.2 Recap of Phase 1 Work

The MNIST dataset of handwritten digits serves as a foundational benchmark for image classification models. However, the real-world applicability of such

models often requires handling noisy or mislabeled data. We introduced the DECODE framework, aimed at enhancing the robustness of deep learning models when confronted with noisy training data. LeNet, one of the earliest convolutional neural networks, was selected for its simplicity and historical significance in the development of deep learning models [5].

1.2.1 Experiment Settings

We evaluated the classification error rate on the MNIST test set, where the performance metric was defined by the proportion of misclassified examples. Label noise rates (nr) were varied from 5% to 40%, to create training sets of differing noise levels. LeNet was trained on these noisy datasets and tested on the original clean test set to assess noise robustness.

1.2.2 Implementation Details

The DECODE framework was implemented using the Caffe deep learning library, with all models trained from scratch to avoid bias from pre-trained weights. A learning rate of 1×10^{-4} and a weight decay of 5×10^{-5} were used, with a batch size of 128. Initial training was conducted without confidence evaluation for the first $1,000$ iterations to establish a base level of feature detection before introducing the noise handling mechanism of DECODE.

1.2.3 Effectiveness Verification

The LeNet model's performance was heavily affected by the presence of noise, with a significant increase in error rates as the noise levels were raised. In contrast, DECODE maintained a consistently lower error rate across all levels of noise, demonstrating its efficacy in managing label noise. Notably, at a noise

rate of 10%, the standard LeNet model's performance decreased drastically, while DECODE exhibits a negligible change in error rate.

Noise Rate	Original	DECODE
0%	0.92%	1.46%
10%	5.35%	1.51%
20%	12.95%	1.73%
30%	19.06%	1.82%
40%	31.22%	2.75%

Table 1.1: Classification Rate Comparison

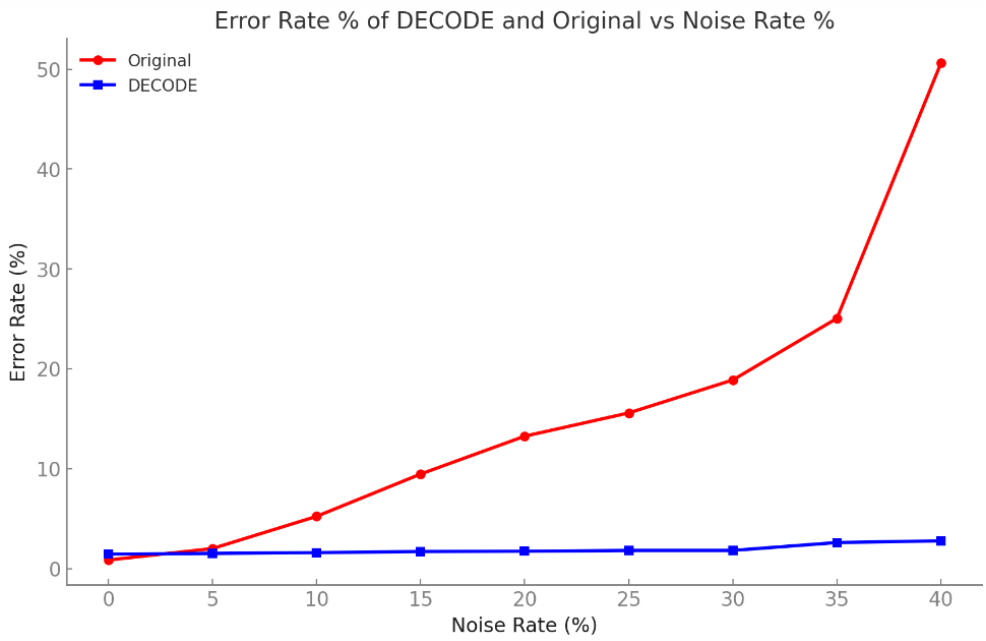

Figure 1.2: *Classification error rate comparison on MNIST dataset with LeNet model.*

1.2.4 Summary

DECODE moderates the impact of boundary cases by assigning lower weights to training examples near the decision boundary. This is based on the assumption that such examples are more likely to be mislabeled due to their ambiguity. By attenuating their influence, DECODE aims to prevent the model from

overfitting to potentially noisy labels, thereby enhancing overall robustness. However, this cautious approach can slightly reduce the model's performance on datasets like MNIST, where boundary examples are generally reliable and contribute valuable information for distinguishing between classes.

1.3 Application of DECODE on Medicinal Plants

CNNs are pivotal in automating plant classification, involving steps such as image processing and feature extraction, crucial for identifying species. However, these models face challenges in accurately representing features like leaf shape. Historically, recognizing medicinal plants has been vital for Ayurvedic medicine, done by hand. Modern needs for large-scale identification have led to using CNNs for more efficient, automated leaf classification, which includes image capture, noise filtering, resizing, and feature analysis.

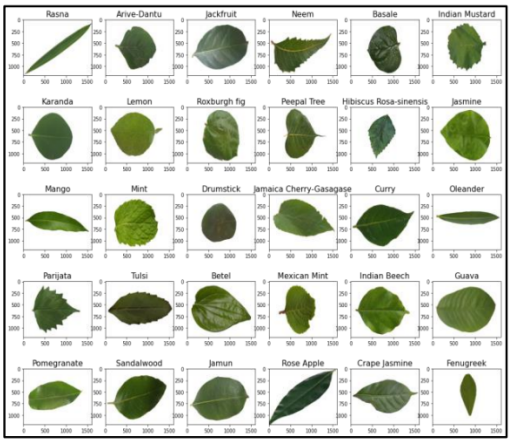

Figure 1.3: *Figure showing the images of the Indian origin leaf species.*

The success of CNNs in image classification tasks is well-documented [7], but their performance can degrade when faced with label noise, a common issue

in real-world data collection. DECODE tackles this problem by introducing a confidence evaluation module, which assesses the likelihood of mislabeling in the training data. We aim at incorporating the DEep COnfiDEnce network (DECODE) into a convolutional neural network (CNN) such as AlexNet which would substantially enhance its capability for classifying images of medicinal plants, particularly in scenarios where the training dataset contains noisy labels.

This report is structured as follows: Chapter 2 introduces the DECODE framework, a novel strategy to mitigate the effects of noisy data in image classification. Chapter 3 discusses the integration of DECODE with AlexNet for classifying medicinal plants, emphasizing methodology and data handling. Chapter 4 analyzes the impact of different weighting functions on model accuracy in noisy environments. Finally, Chapter 5 summarizes the report, highlighting the effectiveness of DECODE in improving classification accuracy amidst label noise.

Chapter 2

Deep Confidence Network

Specifically, considering the distribution of incorrectly labeled data, we integrate a confidence evaluation module capable of assessing the likelihood that a given sample is mislabeled. Utilizing this confidence measure, we implement a weighting strategy to assign varied weights to individual samples. This ensures that the model downplays its focus on low-confidence data, which is often indicative of noise. This approach enhances the robustness of the deep model against label inaccuracies.

2.1 Problem Defintion and Notation

The problem at hand involves working with a set of training data, denoted as X, which comprises elements x_i, where each x_i consists of raw image data (such as RGB pixel values, represented as r_i, an observed label (l_i) for training, and a true label ($\tilde{l}_{\mathrm{i}}$) which remains unknown during the training process. The primary objective is to train a model, denoted as M, using this noisy training dataset X in order to predict the true label of a test image.

To elaborate further, when employing a model and feeding it an input im-

age, we refer to the intermediate output as f_i. This study specifically focuses on addressing uniform random label noise. In the context of an image x_i, we establish that the probability of the observed label (l_i) being equal to the true label ($\tilde{l}_i \mid x_i$) is ρ, while the probability of the observed label being any other class (c) in the label set (L) is (1 - ρ)/(C -1), where C represents the total number of classes in the training dataset, and signifies the unknown probability that the observed label is accurate during training.

2.2 Observation

We initiate the model training process by setting the noise rate (nr) to 0.1 and train the model .During training, we extract the features of each image by obtaining the output of the final pooling layer in the model.This feature extraction step plays a critical role in enabling the model to learn and discriminate between different characteristics.

In the initial iterations of the model, the model does not tightly fit the data; instead, it concentrates on capturing the broad characteristics of the images, enabling it to distinguish between different categories. This approach mirrors the way humans tend to begin by focusing on general features when learning to distinguish between object categories. Consequently, there might be cases where outliers and boundary data samples on the boundary are misidentified.

As training progresses and the model gains experience, reaching up to 10,000 iterations, it becomes proficient at discerning fine and subtle features that are crucial for distinguishing categories. Just as in human learning, where individuals first grasp general attributes before noticing intricate details, deep Convolutional Neural Networks (CNNs) exhibit a similar behaviour.

Research findings corroborate this behaviour in deep CNNs. These networks tend to prioritise learning simple and fundamental patterns initially, then gradually adapt to fit unusual or outlier data. This sequential learning strategy proves effective in gradually improving the model's recognition capabilities.

2.3 DECODE vs. Other Frameworks

Unique Framework: Unlike non-deep methods for handling noise, DECODE is a deep learning framework. It introduces a novel confidence evaluation module and can seamlessly integrate with various deep CNN architectures.

Distinct Strategy: DECODE's strategy differs by not requiring a clean dataset as auxiliary information. It uses the fully-connected (fc) layer output for analysis, offering richer data insights and accommodating intra-class diversity more effectively.

Versatile Application: DECODE can be applied to a wide range of settings and existing architectures without making specific assumptions about noise distribution. This adaptability distinguishes it from approaches with limited applicability based on noise assumptions. Additionally, DECODE's performance during training with noisy data is empirically supported, enhancing its credibility.

2.4 Framework and Approach

We propose a novel approach where we assign varying weights to different samples based on their confidence in being correctly labelled by the current model. Samples with lower confidence, likely to have noisy labels, receive smaller weights. We then backpropagate the weighted loss to adjust the model pa-

rameters. DECODE is versatile and can be integrated into various deep CNN architectures.

Most existing architectures generally follow a convolution-fully connected (fc) design, with many convolution layers initially and a few fully connected layers towards the end. In our approach, we use the output of the layer just before the classification layer as the feature representation (e.g., pool3 layer in DenseNet or fc7 layer in AlexNet) for each sample (f_i).

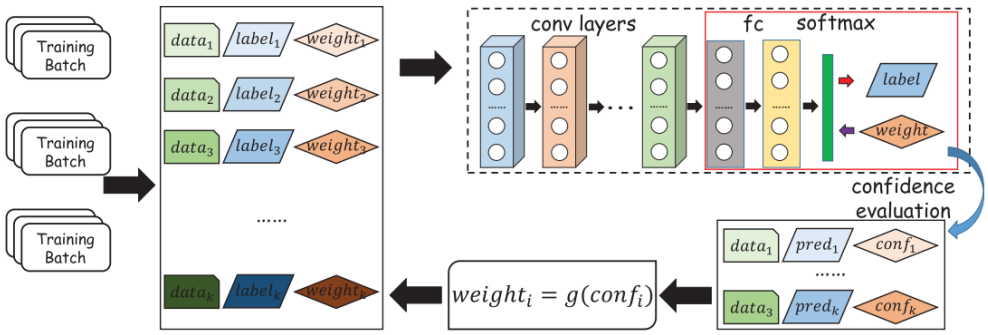

Figure 2.1: *DECODE method for training deep CNN models with noisy labels.*

To determine the confidence of each training sample, we rely on the clustering hypothesis. This hypothesis posits that data from the same class should form clusters and exhibit similarity, a fundamental concept in various machine learning techniques, including semi-supervised learning. DECODE leverages this hypothesis and employs a set of clustering centres (k_j^c) for each class (c ϵ L), with the number of centres (K_c) allowing for flexibility. While the simplest case involves one cluster centre per class, real-world scenarios often benefit from multiple cluster centres for each class, considering the inherent complexity of class structures.

$$v(x_i) = \sum_{c \in C} \sum_{j=1}^{K_c} f\left(\max(0, d(f_i, k_j^{l_i}) - d(f_i, k_j^c) + \lambda)\right). \quad (1)$$

We employ a distance measure (e.g., Euclidean or cosine distance, denoted as d(·,·)) and introduce a non-negative margin parameter (λ). Our objective is to identify abnormal samples that deviate from the clustering hypothesis, meaning their distance to their own class centre surpasses the distance to centres of other classes. To quantify these deviations, we use a counter function $f(x)$ that keeps track of violations. This function can be a hard count, denoted as $f(x) = 1$ for true or 0 otherwise, or a soft count, $f(x) = x$, which accounts for the degree of violation.

A critical aspect is the evaluation of $v(x_i)$, where a larger value indicates a higher likelihood of mislabeling because the sample is closer to centres of other classes instead of its own observed label's centre. One might argue that a straightforward strategy involves considering the softmax output (p_i) of a sample. However, our approach in Eq. (1) offers an advantage over this method. Eq. (1) takes into account the distribution structure of the training data, whereas the softmax-based approach focuses solely on individual sample information. As a result, Eq. (1) utilises more comprehensive information, rendering its predictions more reliable.

Based on the violation factor, the confidence (conf_i) is computed as a monotonically decreasing function. DECODE utilizes Eq. (2) to normalize the confidence to the range [0, 1]:

$$conf_i = \exp(-av(x_i)) \,, \tag{2}$$

where α is a factor to control the decay rate.

After normalizing the confidence ($conf_i$), we directly set the training weight (w_i) as $conf_i$. When dealing with a training batch $B = x_i$ for $i = 1, ..., b$, the weighted loss is defined as follows:

$$\text{Loss}(B, \varphi) = \sum_{i=1}^{b} w_i \text{loss}(x_i, l_i, \varphi), \tag{3}$$

where ϕ represents the current model, and loss is a loss function (e.g., cross-entropy).

With a training weight, the influence of noise can be effectively suppressed so that the model is more robust.

The learning algorithm comprises two key parts:

Training the Deep Network: Utilized stochastic gradient descent to minimize Equation (3) in a standard deep learning fashion. This was accomplished with existing deep learning tools like Caffe.

Training the Confidence Evaluation Module: After training the deep network, the confidence evaluation module is updated. This module calculates the violation factor and computes the confidence values which is crucial for the DECODE algorithm's effectiveness.

$$k_j^c = k_j^c - \frac{\tau \sum_{i=1}^{b} \delta(J(x_i) = j \wedge l_i = c) w_i \frac{\partial d(f_i, k_j^c)}{\partial k_j^c}}{1 + \sum_{i=1}^{b} \delta(J(x_i) = j \wedge l_i = c)} \ , \tag{4}$$

where $\delta(z)$ is an indicator function which is 1 if z is true or 0 otherwise and τ is a tiny step size.

As the centres should focus more on reliable samples, here we also take take the weight w_i into consideration. Here $\frac{\partial d(f_i, k_j^c)}{\partial k_j^c}$ is the partial derivative of the distance measure to k_j^c. We have considered the Euclidean distance. We have the option to employ alternative distance or similarity measures such as cosine and inner-product similarity. However, our investigation reveals that Euclidean distance performs slightly better. Below, we provide a summary of the training algorithm for DECODE.

Pseudocode for Training DECODE

Input: Training data λ, model structures, parameters;

Output: A deep CNN model;

Initialize the model with random values;

Inorder to get the initial cluster centers k_j^c run k-means clustering on image features for each class;

repeat

  Select a training batch B;

  Forward B through the model and get image feature f;

  Compute violation factor $v(x_i)$ using f_i by Eq. (1);

  Compute $conf_i$ and training weight w_i by Eq. (2);

  Compute the weighted loss by Eq. (3);

  Back-propagate the weighted loss by SGD;

  Update centers k_j^c by Eq. (4);

until Convergence or maximum iterations;

Return the obtained model;

Chapter 3

Implementation Details and Application

CNNs are pivotal in automating plant classification [6], involving steps such as image processing and feature extraction, crucial for identifying species [4] However, the real-world applicability of such models often requires handling noisy or mislabeled data. We introduce the DECODE framework, aimed at enhancing the robustness of deep learning models when confronted with noisy training data. AlexNet is a pioneering convolutional neural network (CNN) used primarily for image recognition and classification tasks.

3.1 Methodology

When integrating DECODE with AlexNet, each image of a medicinal plant would pass through AlexNet's deep layers, extracting features and predicting labels. Concurrently, the confidence evaluation module of DECODE would appraise these predictions, weighing them based on the assessed confidence. This dual-path system enables AlexNet to focus on learning from the most reliable samples, decreasing the influence of potentially incorrect labels. By dynami-

cally adjusting attention to samples according to their assessed label quality, DECODE effectively makes AlexNet more robust to the inaccuracies inherent in large-scale, real-world datasets. This integration could prove especially beneficial in the field of medicinal plant classification, where the distinction between species is often nuanced and training data can be imperfect.

3.2 Data Preparation

To effectively train and assess our model, it is crucial to leverage datasets with diverse characteristics and ample coverage. In our case, we utilized a dataset sourced from Kaggle [1], which offered a robust collection of images for our analysis. Initially, this dataset encompassed images representing 12 different plant species. However, to focus our study on medicinal plants, we carefully curated a subset by selecting 5 specific medicinal plant species from the Kaggle dataset. This curation process resulted in a total of 1150 images for our analysis. Furthermore, to enhance computational efficiency and streamline model training, the selected images were resized to 224 x 224 dimensions.
The images were split between training, test, validating and prediction datasets for model training and testing purposes.

As depicted in Figure 3.1, the sample dataset consists of samples for each plant for the purpose of analysis. The medicinal plant species selected are:

1. Citrus Limon (Lemon)
2. Ocimum Tenuiflorum (Basil/Tulsi)
3. Pongamia Pinnata (Indian Beech)
4. Punica Granatum (Pomegranate)
5. Syzygium Cumini (Jamun)

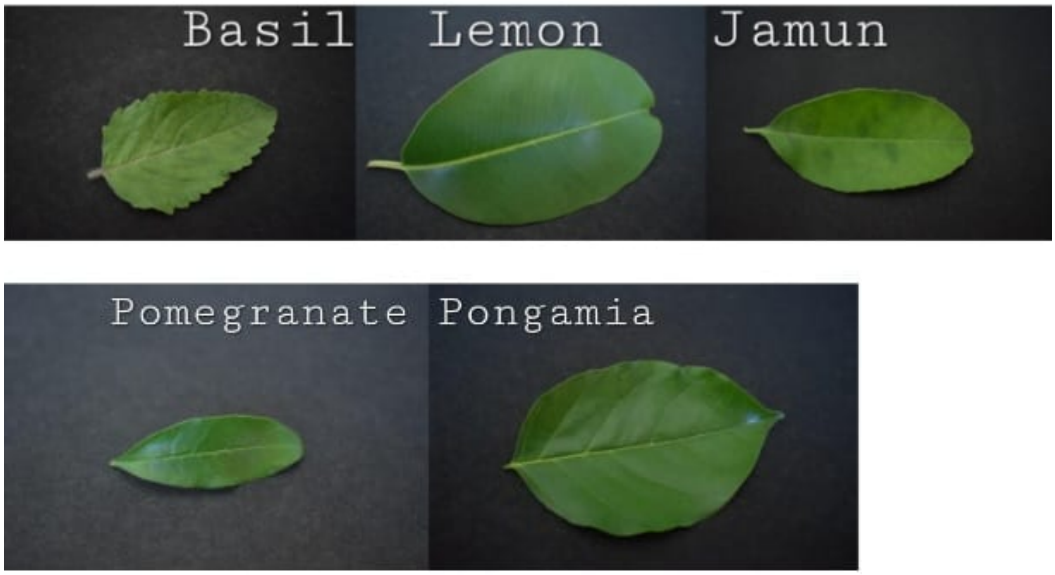

Figure 3.1: *Medicinal plant species in the curated dataset*

3.3 Experiment Settings

We evaluate the classification error rate on the medicinal plants dataset, where the performance metric is defined by the proportion of misclassified examples. Specifically, the classification error rate on the test set is used as the metric, which is defined as:

$$\text{Error Rate} = \frac{1}{n_t} \sum_{i=1}^{n_t} I(p_i \neq \tilde{l}_i) \ ,$$

where n_t is the number of test samples, $\tilde{l}_i$ is the true label of the i-th sample, and p_i is the predicted label by the model.

The indicator function $I(x)$ equals 1 if x is true and 0 otherwise. Our goal is to evaluate models trained with noisy data. However, it is important to note that the benchmark datasets are almost noiseless. Therefore, we manually add label noise to them. In particular, we set a noise rate (n_r). Given a training sample with label $\tilde{l}_i$, its observed label l_i, which is actually used for training, is sampled with probability $p(l_i = \tilde{l}_i) = 1 - n_r$ and $p(l_i = c) = \frac{n_r}{C-1}$, $\forall c \in L \setminus \tilde{l}_i$

where $C = |L|$. Label noise rates (n_r) are varied from 0% to 40%, to create training sets of differing noise levels. AlexNet is then trained on these noisy datasets and tested on the original clean test set to assess the robustness of the model to noise.

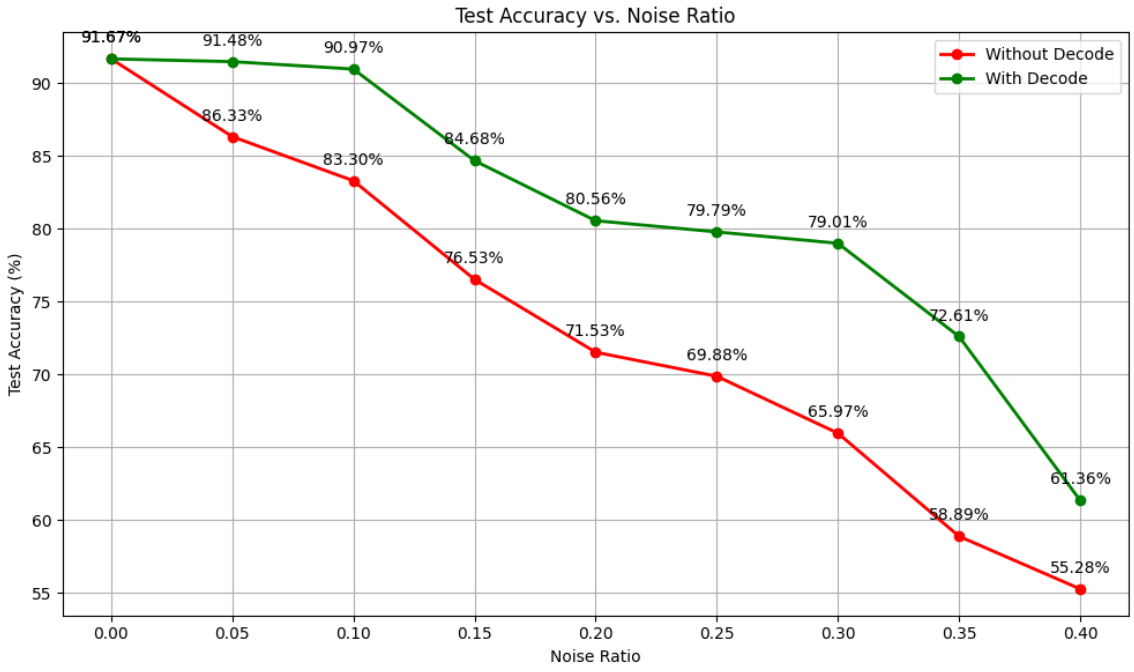

Figure 3.2: *Classification accuracy rate comparison on medicinal dataset with AlexNet model.*

3.4 Implementation Details

The DECODE framework is implemented and all models trained from scratch to avoid bias from pre-trained weights. A learning rate of 5×10^{-4} and a weight decay of 5×10^{-2} are used, with a batch size of 48. Initial training is conducted without confidence evaluation for the first 25 iterations to establish a base level of feature detection before introducing the noise handling mechanism

of DECODE.

3.5 Performance Validation

The AlexNet model's performance is heavily affected by the presence of noise, with a significant increase in error rates as noise levels rise. In contrast, DECODE maintains a consistently lower error rate across all levels of noise, demonstrating its efficacy in managing label noise. Notably, at a noise rate of 20%, the standard AlexNet model's performance decreases drastically, while DECODE exhibits a negligible change in error rate.

Noise Rate	**Original**	**DECODE**
0%	91.67%	91.67%
5%	86.33%	91.48%
10%	83.30%	90.97%
15%	76.53%	84.68%
20%	71.53%	80.56%
25%	69.88%	79.79%
30%	65.97%	79.01%
35%	58.89%	72.61%
40%	55.28%	61.36%

Table 3.1: Classification Rate Comparison

The figures below offer a comparative analysis of the results obtained from AlexNet, both with and without the implementation of DECODE framework when subjected to a noise rate of 10% . It is observed that the model with the DECODE framework exhibits superior performance, accurately predicting 9 out of 10 images. In contrast, the version of AlexNet without decode functionality demonstrates slightly reduced accuracy, correctly identifying 8 out of 10 images under the same conditions. This discrepancy underscores the efficacy of the DECODE framework in enhancing model robustness and accuracy in the presence of label noise.

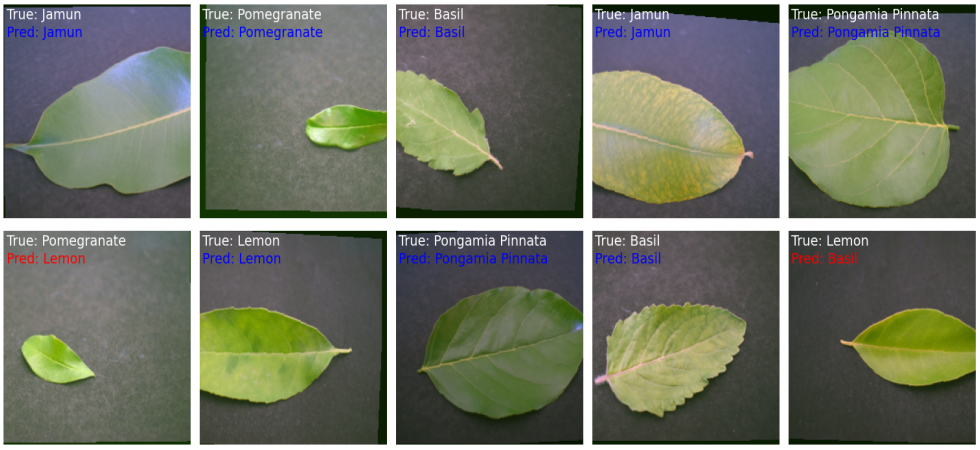

Figure 3.3: *Correctly predicted images by the Original model with 10% noise rate*

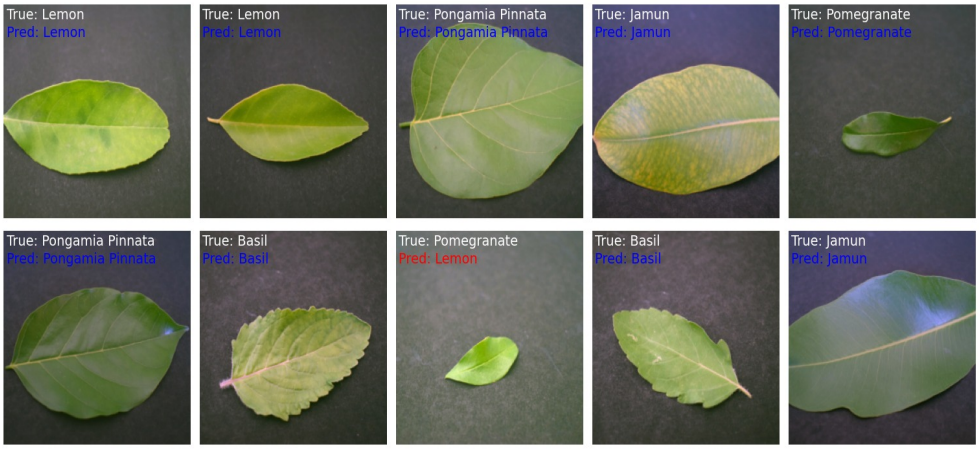

Figure 3.4: *Correctly predicted images by the DECODE model with 10% noise rate*

3.6 Conclusion

DECODE adjusts the influence of boundary cases by assigning lower weights to training examples situated near the decision boundary. This strategy rests on the premise that such examples are more susceptible to mislabeling due to their inherent ambiguity. By mitigating their impact, DECODE endeavors to safeguard the model against overfitting to potentially erroneous labels, thereby bolstering its overall robustness.

Nevertheless, this cautious approach may result in a slight reduction in the model's performance on datasets where boundary examples are typically reliable and contribute valuable information for distinguishing between classes. Thus DECODE with AlexNet is particularly suitable for medicinal plant classification tasks, as it helps mitigate mislabeling of similar plant leaves, thereby enhancing the accuracy and reliability of the classification process.

Chapter 4

Experiment and Analysis

4.1 Introduction

In machine learning, the integrity of a classifier often hinges on the assumption that samples from the same class will cluster closely together [8]. Real-world data however often contains anomalies — samples that, contrary to expectation, lie closer to the centers of foreign classes rather than their own. Such samples are said to have a high violation factor $v(x)$ and their identification is critical for improving the robustness of classification models. To systematically manage these aberrant samples, a measure of confidence is employed, which is inversely related to the violation factor. The confidence measure serves a dual purpose:

1. It acts as a normalizing agent constraining values between 0 and 1,

2. It also functions as a weighting mechanism, w_i during the training phase.

Consequently, by setting the training weight equal to the confidence, $w_i = conf_i$ [3] we ensure that the influence of each data point on the learning process is proportional to its adherence to the clustering hypothesis, thus enhancing the model's sensitivity to well-clustered data while diminishing the impact of outliers.

4.2 Experimental Setup

We conducted an experiment on medicinal plants dataset augmented with 20% noise to simulate mislabeling. Each data point's weight was calculated using one of the three functions — **reciprocal, logarithmic, and negative exponential**, with the weight being equal to the confidence .These weights were then used during the training of the AlexNet model with DECODE.

The training process spanned over around 150 epochs to allow the model to adequately learn and adapt to the weighted dataset. The primary metric for evaluation was the validation accuracy, measured at the end of each epoch.

Table 4.1 compares the characteristics of the different weighting functions.

Feature	**Negative Exponential**	**Logarithmic Scaling**	**Reciprocal Function**
Formula	$f(x) = \exp(-\alpha v(x))$	$f(x) = \frac{1}{\log(\alpha v(x)+\beta)}$	$f(x) = \frac{1}{1+\alpha v(x)}$
Decrease Rate	Rapid decrease	Gradual decrease	Moderate decrease
Behavior as $v(x)$ increases	Weight decreases rapidly, very sensitive to increases in $v(x)$, especially when $v(x)$ is small	Weight decreases slowly, less sensitive to increases in $v(x)$, may become undefined $v(x)$=0	Weight decreases more uniformly, is never undefined, does not decrease to zero as quickly as the exponential
Parameter Adjustment	α controls rate of decrease	α and β adjust sensitivity and ensure positivity	α controls sensitivity to $v(x)$ changes
Best use case	Suitable for aggressively penalizing mislabeled data	Suitable for a more conservative approach, penalizing mislabeled data more gently	Suitable for a balanced approach, penalizing mislabeled data without diminishing their weight too rapidly

Table 4.1: Comparison of Weighting Functions for Mislabeled Data

4.3 Results

According to the trends observed in the validation accuracies, it becomes clear that the negative **exponential function** outperformed its counterparts. Despite a modest beginning, its strict approach to down-weighting the mislabeled samples showed increasing returns, as evidenced by the steady and notable rise in accuracy. This suggests that the function's aggressive penalization of potential mislabels was ultimately beneficial, helping the model to disregard deceptive data and focus on learning the correct patterns.

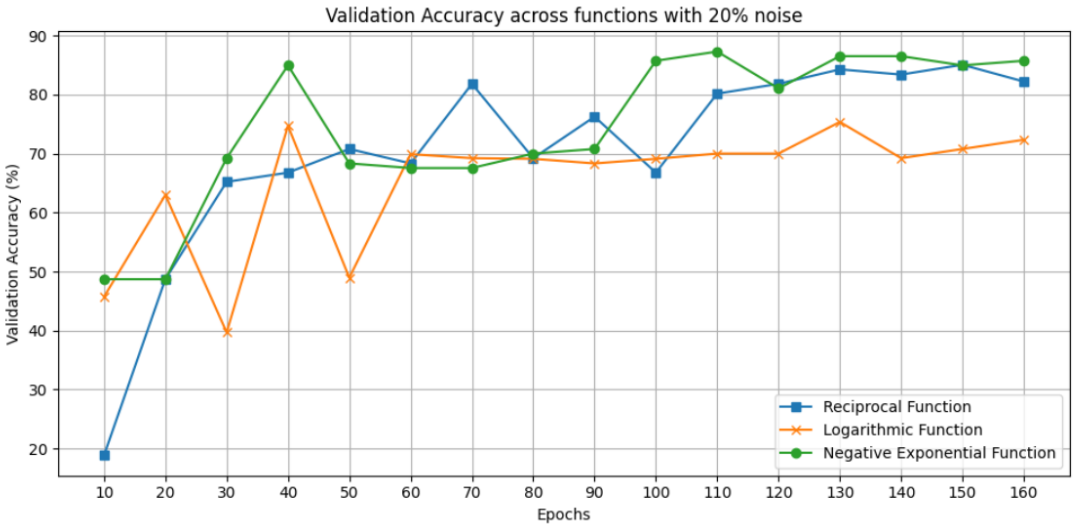

Figure 4.1: *Validation accuracy rate comparison across different functions*

The **reciprocal function**, while ultimately improving, showed a higher degree of initial fluctuation in validation accuracy, which is likely a result of its more conservative approach to penalization. This less aggressive strategy may require additional epochs for the model to adjust and learn effectively from the noisy data.

The **logarithmic function**, striking a balance between the two, demonstrated steadier performance but did not reach the validation accuracy highs of the

negative exponential function. It suggests that its moderate penalization was not as adept at handling high noise levels in the dataset.

4.4 Conclusion

In light of the experimental data, the **negative exponential function** has demonstrated the strongest potential for dealing with mislabeled data. Its ability to significantly boost model performance in the presence of noisy data underscores its appropriateness as a weighting mechanism, particularly in scenarios where rigorous penalization of mislabels is integral to maintaining high levels of classification accuracy.

Chapter 5

Concluding Remarks

In this report, we have detailed the integration of convolutional neural networks (CNNs) with the DECODE framework for enhanced plant classification accuracy, emphasizing its efficacy in managing noisy mislabelled datasets. We started by providing a brief overview of the model's architecture and highlighting some of the previous works in the paper by us.

We integrated the DECODE framework with AlexNet, which revealed how we can significantly mitigate the adverse effects of mislabeled data, thereby ensuring that learning is predominantly influenced by high-confidence, reliable samples. In the course of this research, a deliberate introduction of noise into the medicinal plants dataset served as a critical test for evaluating the resilience and effectiveness of convolutional neural networks, particularly when augmented with the DECODE framework.

We did a comparative analysis of model performances under varying levels of artificially introduced noise which showcased that while the model experienced a degradation in accuracy with increased noise levels, the implementation of DECODE, significantly outperformed traditional approaches. This combination not only demonstrated a remarkable resilience to noise but also underscored

the efficacy of the DECODE framework to maintain higher accuracy levels.

We also conducted an experiment comparing the weighting functions —reciprocal, logarithmic, and negative exponential to refine the model's ability to discern and prioritize valuable data during the training process. Among these, the negative exponential function emerged as particularly effective in enhancing model performance under conditions of high noise levels, highlighting its utility in aggressively filtering out mislabeled samples to focus on learning accurate data patterns.

Overall, this research contributes valuable insights into the application of CNNs for plant classification, emphasizing the necessity of addressing data quality issues to achieve high classification accuracy. The DECODE framework, combined with CNNs, presents a compelling approach to overcome the challenges of noisy mislabelled datasets, demonstrating significant potential for improving the reliability and efficiency of automated plant classification systems.

Bibliography

[1] Plant leaves for image classification. https://www.kaggle.com/datasets/csafrit2/plant-leaves-for-image-classification, 2020. Accessed: Feb 5, 2024.

[2] X. Chen and A. Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1431–1439, December 2015.

[3] Guiguang Ding, Yuchen Guo, Kai Chen, Chaoqun Chu, Jungong Han, and Qionghai Dai. Decode: Deep confidence network for robust image classification. volume 28, Aug 2019.

[4] Nayana G. Gavhale and A.P. Thakare. Identification of medicinal plant using machine learning approach. volume 07, page 1116, Jul 2020.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. volume 86, pages 2278–2324, November 1998.

[6] Yang Liu, Yating Li, Yanjie Zhao, and Xitai Na. Image classification and recognition of medicinal plants based on convolutional neural network. In *2021 IEEE 21st International Conference on Communication Technology (ICCT)*, pages 1431–1439, Tianjin, China, October 2021. IEEE.

[7] Trung Nguyen Quoc and Vinh Truong Hoang. Medicinal plant identification in the wild by using cnn. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 25–29. IEEE, 2020.

[8] X. J. Zhu. Semi-supervised learning literature survey. Technical report, Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, 2005.