ForestKNN - Combining Random Forests and KNN

xAI-Proj-M: Master Project Explainable Machine Learning

Ali Ostadi[*]
Otto-Friedrich University of Bamberg
96049 Bamberg, Germany
ali.ostadi@stud.uni-bamberg.de

Andreas Franz Schwab[†]
Otto-Friedrich University of Bamberg
96049 Bamberg, Germany
andreas-franz.schwab@stud.uni-bamberg.de

Abstract

In an era where AI systems like ChatGPT consume electricity equivalent to powering 180,000 U.S. households daily (Gordon, 2024), efficient machine learning methods are crucial. This project enhances k-Nearest Neighbour (kNN) methods for image classification using vector-databases and latent space analysis. We explored the Cifar10 and Cifar100 (Krizhevsky, 2009), as well as the DermaMNIST and BreastMNIST datasets (Yang et al., 2021, 2023), visualized internal data clusters with tSNE plots and examined the existing neighborhood structures. Furthermore, we implemented a simple kNN in PyTorch and compared its performance to a simple trained Linear Layer. Our key contribution, ForestKNN, combines the idea of Random Forests and kNN, and includes multiple kNN classifiers using random subsets of samples and features, improving speed and accuracy over linear probing. Though resource-intensive, ForestKNN shows promise as a robust alternative for image classification. For detailed implementation and results, visit our Git Repository.

1 Introduction

The emergence of deep learning has led to unprecedented advances in various fields, including medical image analysis. This project seeks to explore the fundamental principles of deep learning and to leverage its potential in a practical setting. Our investigation is divided into three parts: understanding vector-databases and latent spaces, evaluating performance with traditional and advanced methods, and developing an enhanced kNN algorithm.

Image classification is a cornerstone of many AI applications, such as disease diagnosis, autonomous vehicles, and security systems. Traditional methods like k-Nearest Neighbour (kNN) offer simplicity and interpretability but struggle with high-dimensional data and large datasets. Improving these methods to handle modern, complex datasets is crucial for advancing practical AI applications.

At the centre of our project is the hypothesis that specific adaptations and refinements of deep learning techniques can significantly improve model performance on both simple and complex classification tasks . . .

Project structures . . .

1.1 Related Work

kNN has long been a popular choice due to its simplicity and ease of understanding. However, its performance can degrade with the increase in data dimensionality and size. Linear Probing,

[*] Degree: M.Sc. WI, matriculation #: 12345678

[†] Degree: M.Sc. AI, matriculation #: 2017990

which involves training a linear classifier on top of pre-trained features, has been suggested as a means to improve kNN performance (Alain and Bengio, 2016). Additionally, ensemble methods like Random Forests enhance model robustness and accuracy by combining multiple classifiers (Breiman, 2001). Research has also focused on latent space exploration to gain insights into data structures and relationships. Techniques like tSNE are used to visualize these latent spaces, revealing patterns and class overlaps that inform the development of more sophisticated models (van der Maaten and Hinton, 2008).

1.2 Contribution

This project aims to enhance kNN for image classification by integrating ensemble learning principles. We start by exploring vector-databases and latent spaces of several image datasets, including Cifar10, Cifar100, DermaMNIST, and BreastMNIST. By visualizing data distributions and examining neighborhood structures, we gain insights that guide our model development.

In the second phase, we implement a simple kNN in PyTorch and compare its performance against a Linear Layer trained with only a single layer. This comparison helps establish a baseline for further improvements.

Our key contribution, ForestKNN, combines the concepts of Random Forests and kNN. ForestKNN uses multiple kNN classifiers on random subsets of samples and features, aggregating their results through majority voting. This approach enhances accuracy, and once the optimal combination of parameters and methods is identified, it also reduces computation time.

2 Methods

2.1 Simple kNN

To evaluate the performance of the k-Nearest Neighbour (kNN) classifier, we implemented a custom kNN model using PyTorch, enabling the use of GPU acceleration. The kNN algorithm classifies a data point by identifying the k closest points in the training set and assigning the majority class among these neighbors. The kNN classifier is initialized with the number of neighbors k and the device (CPU or GPU) on which computations are performed. The training process involves storing the training data and corresponding labels as PyTorch tensors on the specified device. During prediction, the classifier computes pairwise distances between the input data and the training data using the Euclidean distance metric. The nearest neighbors are identified by sorting these distances. The class labels of the nearest neighbors are then used to predict the class of each input sample by taking the mode (most common label) of these labels.

2.2 Simple Linear Layer

The Linear Probing method involves training a simple linear classifier using the feature representations (embeddings) from the datasets. This method aims to improve classification by leveraging a single layer neural network directly on the embeddings. The Linear Layer is initialized with the input dimension, representing the feature size of the embeddings, and the number of classes in the dataset. The training process involves defining a loss function (Cross-Entropy Loss) and an optimizer (Adam). The model is trained by performing forward and backward passes to minimize the loss function over a series of epochs. The training data is fed into the linear layer, and the model parameters are updated to improve classification accuracy. Hyperparameter tuning is performed by training the model with various learning rates, batch sizes, and epochs. The performance on the validation set is monitored to identify the best set of hyperparameters.

2.3 Data Preparation and Evaluation

For loading the data, we used the given embeddings and labels from the Cifar10, Cifar100, DermaMNIST, and BreastMNIST datasets, stored in numpy files. For datasets without predefined validation sets, the training data was split into training and validation sets (80/20 split) to facilitate model evaluation and hyperparameter tuning. The data was normalized to ensure consistent feature scaling, crucial for the performance of distance-based algorithms like kNN. The training and evaluation process for the kNN model involved training the model using the training data and evaluating its performance on the validation set to identify the optimal number of neighbors k. The best k value was then used to evaluate the model on the test set. Performance metrics such as accuracy, precision, recall, and F1 score were calculated to assess the model's effectiveness. For the linear layer, various hyperparameter configurations were tested. The model's performance on the validation set was used to select the best hyperparameters. The final model was then evaluated on the test set, and the performance metrics were recorded.

3 Datasets

3.1 Cifar10

The CIFAR-10 dataset (Krizhevsky, 2009) consists of 60,000 32×32 color images in 10 different classes, with 6,000 images per class. These classes include airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. The dataset is split into a training set of 50,000 images and a test set of 10,000 images. Each image is labeled with one of the 10 classes, providing a robust benchmark for evaluating image recognition algorithms. The embeddings provided for this dataset have 768 dimensions. Some examples of the CIFAR-10 dataset can be seen in Figure 1.

Figure 1: Example images of the CIFAR-10 dataset.

3.2 Cifar100

Similar to CIFAR-10, the CIFAR-100 dataset (Krizhevsky, 2009) contains 60,000 32×32 color images, but it is divided into 100 classes, with 600 images per class. The classes are grouped into 20 superclasses, with each image labeled at two levels of granularity: a coarse label (superclass) and a fine label (class). The dataset is also divided into a training set of 50,000 images and a test set of 10,000 images. CIFAR-100 is more challenging than CIFAR-10 due to its larger number of classes and the finer granularity of the labels. The embeddings provided for this dataset have 768 dimensions. Some examples of the CIFAR-100 dataset can be seen in Figure 2.

Figure 2: Example images of the CIFAR-10 dataset.

3.3 DermaMNIST

The DermMNIST subset of the MedMNIST collection focuses on dermatological image classification, specifically for skin lesion analysis. This dataset includes 10,015 images, each resized to 3×28×28 pixels from their original higher resolution. The images are classified into seven categories: actinic keratoses, basal cell carcinoma, benign keratosis-like lesions, dermatofibroma, melanocytic nevi, melanoma, and vascular lesions. The training set contains 7,007 images, while the test set includes 2,008 images. The complexity of classifying various skin conditions makes DermMNIST a valuable dataset for developing and evaluating medical image classification algorithms. The embeddings provided for this dataset have 768 dimensions. Some examples of the DermMNIST dataset can be seen in Figure 3 (Yang et al., 2021, 2023).

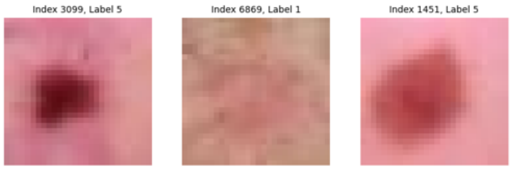

Figure 3: Example images of the CIFAR-10 dataset.

3.4 BreastMNIST

The BreastMNIST subset is another component of the MedMNIST collection, designed for breast ultrasound image classification. It comprises 780 images of breast lesions, categorized into three types: normal, benign, and malignant. Originally, the images were high-resolution, but for the purposes of this subset, they were resized to 28×28 pixels. The dataset is split into a training set of 546 images and a test set of 234 images. BreastMNIST presents a significant challenge due to the subtle visual differences between normal, benign, and malignant lesions, necessitating the use of sophisticated deep learning models. Some examples of the BreastMNIST dataset can be seen in Figure 4 (Yang et al., 2021, 2023).

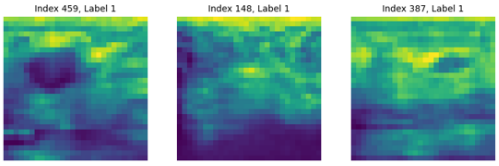

Figure 4: Example images of the CIFAR-10 dataset.

3.4.1 Headings: third level

Third-level headings should be in 10-point type.

Paragraphs There is also a `\paragraph` command available, which sets the heading in bold, flush left, and inline with the text, with the heading followed by 1 em of space.

4 Citations, figures, tables, references

These instructions apply to everyone.

4.1 Citations within the text

The `natbib` package will be loaded for you by default. Citations may be author/year or numeric, as long as you maintain internal consistency. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

The documentation for `natbib` may be found at

```
http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf
```

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{He_2016_CVPR} investigated\dots
```

produces

? investigated...

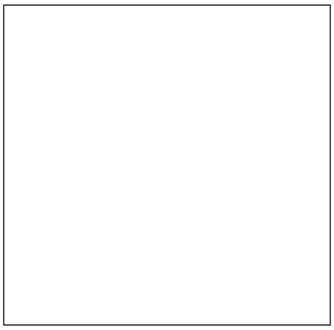

Figure 5: Sample figure caption.

For standard reference the command `\citep` is appropriate and produces (?). Multiple references can be cited e.g. with

```
\citep{Bengio_chapter2007,He_2016_CVPR,Hinton06,goodfellow2016deep}
```

yielding

(????)

If you wish to load the `natbib` package with options, you may add the following before loading the `ofu_xao_2022` package:

```
\PassOptionsToPackage{options}{natbib}
```

If `natbib` clashes with another package you load, you can add the optional argument `nonatbib` when loading the style file:

```
\usepackage[nonatbib]{ofu_xao_2022}
```

4.2 Footnotes

Footnotes should be used sparingly. If you do require a footnote, indicate footnotes with a number[3] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches.

Note that footnotes are properly typeset *after* punctuation marks.[4]

4.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction. The figure number and caption always appear after the figure. Place one line space before the figure caption and one line space after the figure. The figure caption should be lower case (except for first word and proper nouns); figures are numbered consecutively.

You may use color figures. However, it is best for the figure captions and the paper body to be legible if the paper is printed in either black/white or in color.

4.4 Tables

All tables must be centered, neat, clean and legible. The table number and title always appear before the table. See Table 1.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

[3] Sample of the first footnote.

[4] As in this example.

Table 1: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

Note that publication-quality tables *do not contain vertical rules.* We strongly suggest the use of the `booktabs` package, which allows for typesetting high-quality, professional tables:

`https://www.ctan.org/pkg/booktabs`

This package was used to typeset Table 1.

5 Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

6 Preparing PDF files

Please prepare submission files with paper size “A4,” and not, for example, “Letter”.

Fonts can be the main cause of problems. Your PDF file should only contain Type 1 or Embedded TrueType fonts.

6.1 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the `graphicx` package. Always specify the figure width as a multiple of the line width as in the example below:

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

See Section 4.4 in the graphics bundle documentation (`http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command when necessary.

7 Notation

This section provides a concise reference describing notation as used in the book by **?**. If you are unfamiliar with any of the corresponding mathematical concepts, **?** describe most of these ideas in chapters 2–4.

Numbers and Arrays

a	A scalar (integer or real)
$\boldsymbol{a}$	A vector
$\boldsymbol{A}$	A matrix
$\boldsymbol{\mathsf{A}}$	A tensor
$\boldsymbol{I}_n$	Identity matrix with n rows and n columns
$\boldsymbol{I}$	Identity matrix with dimensionality implied by context
$\boldsymbol{e}^{(i)}$	Standard basis vector $[0, \dots, 0, 1, 0, \dots, 0]$ with a 1 at position i
$\text{diag}(\boldsymbol{a})$	A square, diagonal matrix with diagonal entries given by $\boldsymbol{a}$
a	A scalar random variable
$\mathbf{a}$	A vector-valued random variable
$\mathbf{A}$	A matrix-valued random variable

Sets and Graphs

$\mathbb{A}$	A set
$\mathbb{R}$	The set of real numbers
$\{0, 1\}$	The set containing 0 and 1
$\{0, 1, \dots, n\}$	The set of all integers between 0 and n
$[a, b]$	The real interval including a and b
$(a, b]$	The real interval excluding a but including b
$\mathbb{A} \backslash \mathbb{B}$	Set subtraction, i.e., the set containing the elements of $\mathbb{A}$ that are not in $\mathbb{B}$
$\mathcal{G}$	A graph
$Pa_{\mathcal{G}}(\mathrm{x}_i)$	The parents of x_i in $\mathcal{G}$

Indexing

a_i	Element i of vector $\boldsymbol{a}$, with indexing starting at 1
a_{-i}	All elements of vector $\boldsymbol{a}$ except for element i
$A_{i,j}$	Element i, j of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{i,:}$	Row i of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{:,i}$	Column i of matrix $\boldsymbol{A}$
$\mathsf{A}_{i,j,k}$	Element (i, j, k) of a 3-D tensor $\boldsymbol{\mathsf{A}}$
$\boldsymbol{\mathsf{A}}_{:,:,i}$	2-D slice of a 3-D tensor
a_i	Element i of the random vector $\mathbf{a}$

Linear Algebra Operations

$\boldsymbol{A}^\top$	Transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A}^+$	Moore-Penrose pseudoinverse of $\boldsymbol{A}$
$\boldsymbol{A} \odot \boldsymbol{B}$	Element-wise (Hadamard) product of $\boldsymbol{A}$ and $\boldsymbol{B}$
$\det(\boldsymbol{A})$	Determinant of $\boldsymbol{A}$

Calculus

$\dfrac{dy}{dx}$	Derivative of y with respect to x
$\dfrac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\nabla_{\boldsymbol{x}} y$	Gradient of y with respect to $\boldsymbol{x}$
$\nabla_{\boldsymbol{X}} y$	Matrix derivatives of y with respect to $\boldsymbol{X}$
$\nabla_{\mathsf{\mathbf{X}}} y$	Tensor containing derivatives of y with respect to $\mathsf{\mathbf{X}}$
$\dfrac{\partial f}{\partial \boldsymbol{x}}$	Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ of $f : \mathbb{R}^n \to \mathbb{R}^m$
$\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x})$ or $\boldsymbol{H}(f)(\boldsymbol{x})$	The Hessian matrix of f at input point $\boldsymbol{x}$
$\int f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral over the entire domain of $\boldsymbol{x}$
$\int_{\mathbb{S}} f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral with respect to $\boldsymbol{x}$ over the set $\mathbb{S}$

Probability and Information Theory

$\mathrm{a} \bot \mathrm{b}$	The random variables a and b are independent
$\mathrm{a} \bot \mathrm{b} \mid \mathrm{c}$	They are conditionally independent given c
$P(\mathrm{a})$	A probability distribution over a discrete variable
$p(\mathrm{a})$	A probability distribution over a continuous variable, or over a variable whose type has not been specified
$\mathrm{a} \sim P$	Random variable a has distribution P
$\mathbb{E}_{\mathrm{x} \sim P}[f(x)]$ or $\mathbb{E} f(x)$	Expectation of $f(x)$ with respect to $P(\mathrm{x})$
$\mathrm{Var}(f(x))$	Variance of $f(x)$ under $P(\mathrm{x})$
$\mathrm{Cov}(f(x), g(x))$	Covariance of $f(x)$ and $g(x)$ under $P(\mathrm{x})$
$H(\mathrm{x})$	Shannon entropy of the random variable x
$D_{\mathrm{KL}}(P \Vert Q)$	Kullback-Leibler divergence of P and Q
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution over $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

Functions

$f : \mathbb{A} \to \mathbb{B}$	The function f with domain $\mathbb{A}$ and range $\mathbb{B}$
$f \circ g$	Composition of the functions f and g
$f(\boldsymbol{x}; \boldsymbol{\theta})$	A function of $\boldsymbol{x}$ parametrized by $\boldsymbol{\theta}$. (Sometimes we write $f(\boldsymbol{x})$ and omit the argument $\boldsymbol{\theta}$ to lighten notation)
$\log x$	Natural logarithm of x
$\sigma(x)$	Logistic sigmoid, $\dfrac{1}{1 + \exp(-x)}$
$\zeta(x)$	Softplus, $\log(1 + \exp(x))$
$\vert\vert \boldsymbol{x} \vert\vert_p$	L^p norm of $\boldsymbol{x}$
$\vert\vert \boldsymbol{x} \vert\vert$	L^2 norm of $\boldsymbol{x}$
x^+	Positive part of x, i.e., $\max(0, x)$
$\boldsymbol{1}_{\mathrm{condition}}$	is 1 if the condition is true, 0 otherwise

Sometimes we use a function f whose argument is a scalar but apply it to a vector, matrix, or tensor: $f(\boldsymbol{x})$, $f(\boldsymbol{X})$, or $f(\mathsf{\mathbf{X}})$. This denotes the application of f to the array element-wise. For example, if $\mathsf{\mathbf{C}} = \sigma(\mathsf{\mathbf{X}})$, then $\mathsf{C}_{i,j,k} = \sigma(\mathsf{X}_{i,j,k})$ for all valid values of i, j and k.

Datasets and Distributions

p_{data}	The data generating distribution
$\hat{p}_{\text{data}}$	The empirical distribution defined by the training set
$\mathbb{X}$	A set of training examples
$\boldsymbol{x}^{(i)}$	The i-th example (input) from a dataset
$y^{(i)}$ or $\boldsymbol{y}^{(i)}$	The target associated with $\boldsymbol{x}^{(i)}$ for supervised learning
$\boldsymbol{X}$	The $m \times n$ matrix with input example $\boldsymbol{x}^{(i)}$ in row $\boldsymbol{X}_{i,:}$

References

Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. Note that the Reference section does not count towards the page limit.

References

G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016. URL https://arxiv.org/abs/1610.01644.

L. Breiman. *Random Forests*. Springer, 2001.

C. Gordon. Chatgpt and generative ai innovations are creating sustainability havoc. *Forbes*, 2024. URL https://www.forbes.com/sites/cindygordon/2024/03/12/chatgpt-and-generative-ai-innovations-are-creating-sustainability-havoc/. Accessed: 2024-07-27.

A. Krizhevsky. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009. URL https://www.cs.toronto.edu/~kriz/cifar.html. Accessed: 2024-07-27.

L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. URL http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.

J. Yang, R. Shi, and B. Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021.

J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Declaration of Authorship

All final papers have to include the following ‘Declaration of Authorship’:

Declaration of Authorship

Ich erkläre hiermit gemäß § 9 Abs. 12 APO, dass ich die vorstehende Projektarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Des Weiteren erkläre ich, dass die digitale Fassung der gedruckten Ausfertigung der Projektarbeit ausnahmslos in Inhalt und Wortlaut entspricht und zur Kenntnis genommen wurde, dass diese digitale Fassung einer durch Software unterstützten, anonymisierten Prüfung auf Plagiate unterzogen werden kann.

Bamberg, July 28, 2024

(Place, Date)	(Signature)

Bamberg, July 28, 2024

(Place, Date)	(Signature)

Bamberg, July 28, 2024

(Place, Date)

(Signature)

A Appendix

Optionally include extra information (complete proofs, additional experiments and plots) in the appendix. This section will often be part of the supplemental material.