THE UNIVERSITY OF SYDNEY

School of Information Technologies
Faculty of Engineering & IT

ASSIGNMENT/PROJECT COVERSHEET - GROUP ASSESSMENT

Unit of Study: COMP5318

Assignment name: COMP5318 -Machine Learning and Data Mining Assignment 2

Tutorial time: PRAC 02 MON 20:00 **Tutor name:** James Collins

DECLARATION

We the undersigned declare that we have read and understood the *University of Sydney Student Plagiarism: Coursework Policy and Procedure*, and except where specifically acknowledged, the work contained in this assignment/project is our own work, and has not been copied from other sources or been previously submitted for award or assessment.

We understand that understand that failure to comply with the *Student Plagiarism: Coursework Policy and Procedure* can lead to severe penalties as outlined under Chapter 8 of the *University of Sydney By-Law 1999* (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

We realise that we may be asked to identify those portions of the work contributed by each of us and required to demonstrate our individual knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

Project team members

Student name	Student ID	Participated	Agree to share	Signature
1. RUIXIAN LIU	510127572	Yes	Yes	RUIXIAN LIU
2. FEIFAN GAO	510583158	Yes	Yes	
3.		Yes / No	Yes / No	
4.		Yes / No	Yes / No	
5.		Yes / No	Yes / No	
6.		Yes / No	Yes / No	
7.		Yes / No	Yes / No	
8.		Yes / No	Yes / No	
9.		Yes / No	Yes / No	
10.		Yes / No	Yes / No	

COMP 5318 Assignment 2 report

Feifan Gao, Ruixian Liu

November 6, 2021

Contents

1 Abstract

The main objective of this report is to classify the CIFAR-100 dataset using five different classification algorithms and analyze the results by comparing them to find which is the best algorithm. The classification algorithms used are AdaBoost, Convolutional Neural Networks, K-nearest neighbors algorithm, Naive Bayes algorithm, and RandomForest.The introduction describes the data set used, the importance of the problem, and an overview of the classification methods used. This report then presents a previously used algorithm, explaining how it works on the dataset and how it differs from the method used in this experiment. The report is followed by a description of the different classification methods and comparing the different methods with multiple data. Convolutional Neural Networks (CNN) was the best performing classification algorithm on the CIFAR-100 dataset.

2 Introduction

2.1 Dataset description

CIFAR-100 was collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton[KH+09]. The CIFAR - 100 dataset consists of 60,000 32 x 32 color images of 100 categories. There are 50,000 training images and 10,000 test images, and each class contains 600 images. There are 500 training images and 100 testing images in each class. One hundred classes in CIFAR-100 are divided into 20 super classes. Each image has a ”fine” label (the class it belongs to) and a ”coarse” label (the superclass it belongs to).

$$P(\theta|\mathbf{D}) = P(\theta)\frac{P(\mathbf{D}|\theta)}{P(\mathbf{D})} \tag{1}$$

$$D(A,B) = \sqrt{(a-b)^2 + (a-b)^2 + ... + (a-b)^2} \tag{2}$$

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x') \tag{3}$$

$$\alpha_m = \frac{1}{2}\ln\left(\frac{1-\epsilon_m}{\epsilon_m}\right) \tag{4}$$

2.2 Importance of the problem

This project aims to classify the image dataset into 100 classes by finding the best model algorithm. The purpose of this project is to test students’ proficiency in related algorithms of machine learning, as well as their analytical skills of various parameters and related data.

Project developers can gradually master various open-source classifier libraries by classifying the image data sets, which deepen their understanding of various classification algorithms. Through the training of the CIFAR-100 data set, developers can find a suitable algorithm for image recognition.

2.3 Overview of the method

This project adopts K-Nearest Neighbor, Naïve Bayes, Random Forest, AdaBoost, Convolutional Neural Network (CNN), and other algorithms. The performance of each classification algorithm is evaluated according to the accuracy, precision, recall, and confusion matrix. KNN and Naïve Bayes belong to the basic algorithm, Random forest and AdaBoost are ensembles of the basic algorithm, and CNN is a classification algorithm of neural networks. This progression reveals the evolution of machine learning algorithms from simple to complex. By evaluating their performance, this study will describe to readers the outstanding contribution of neural networks to the image recognition field.

3 Previous work

This report in this section will present two previously used algorithms proposed in the literature or reports for classifying the CIFAR-100 dataset and describe how it differs from the best algorithm used in this experiment. Support vector machines (SVMs) and their extensions algorithms are used to perform classification tasks on the CIFAR-100 dataset. SVMs are widely used to solve classification and regression tasks because of their excellent performance[LLLH17]. However, because of the high complexity of SVMs and their training time, it is not possible to use SVMs to classify large data sets. By dividing the problem into smaller subproblems and reducing the time and memory space complexity, the experimental team successfully used the SVM for training and classifying the CIFAR-100 dataset[HSH19]. The CNN is more suitable than the SVM algorithm for analyzing and classifying high-dimensional, high-complexity data in the image class and requires only a simple normalization of the data before training can begin. In contrast, the SVM algorithm requires a method such as PCA to reduce the dimensionality of the data before it can be trained. The main reason for this difference is that the two algorithms work fundamentally differently. CNN's extract features, whereas SVMs only map their input to some high-dimensional space, thus revealing differences between classes. This difference makes the CNN algorithm more suitable for deep structures, whereas SVMs are more suitable for surface-level structures.

4 Methods

4.1 Data preprocessing Methods

Data preprocessing is the first and one of the most important sections of machine learning, which include data cleaning, normalization, and feature extraction. Feature extraction is the process of removing less relevant features and selecting a representative subset of the data attributes.

4.1.1 Normalization

Normalizing features of a range is also known as min-max scaling. It involves rescaling the range of features to fit the specified value. Since the range of values in raw data is often wide such as the RPG range in this dataset is [0,255], objective functions typically fail or slow if they do not have the necessary normalization. This is because the range of values can affect the calculation of the distance between two attribute points.

By applying the min-max scaling technique, we normalize the image dataset within the [0,1] interval. Normalization will speed up the convergence process of some algorithms like SVM.

4.1.2 PCA

The principal component analysis is a programming language that is commonly used in data analysis and machine learning. It projects each data point onto the first few principal components of a graph. By applying PCA, the dataset dimension can be reduced but 85% of the information is saved. Each element in CIFAR-100 dataset is a 32Pixel x 32Pixel x 3(RGB) image. After dimensional reduction, only 46 principal components are left, which significantly reduces the complexity of the dataset. See Figure 1.

4.2 Classification Models

Classification techniques are widely used in many fields like fraud detection in credit card transactions, spam mail detection, and most importantly, image recognition. There are many classifiers such as Knn, Decision Tree, linear classifiers, boosting & bagging, and neural networks. Classification is a supervised machine learning class in which datasets are labeled just like the CIFAR-100 example.

4.2.1 K-Nearest Neighbor

The K-Nearest Neighbor (KNN) algorithm is simple and easy to understand. KNN finds the historical data point closest to the new data by measuring the Euclidean distance between the data sets and then

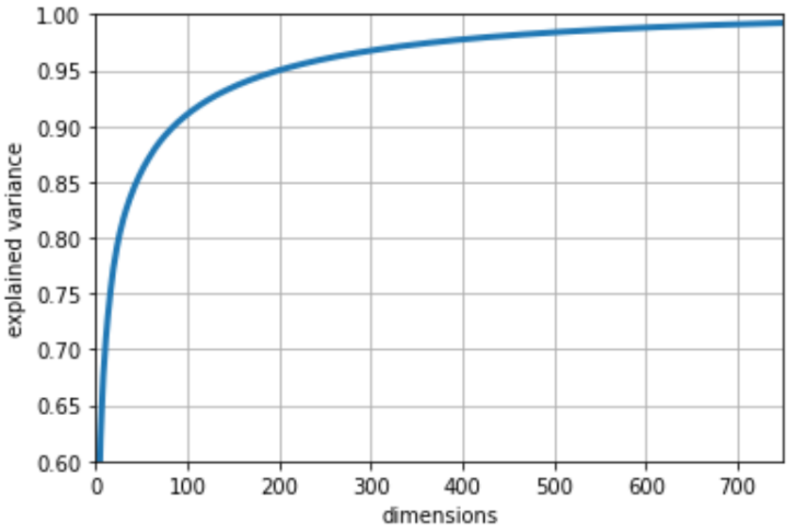


Figure 1: PCA dimensional estimation

assigns the data to the training data class with the closest distance. The number of nearest neighbor data (N) can be selected from 1 to sqrt(number of attributes). N = 10 is a frequently used value, but since CIFAR-100 has one hundred classes, in this case, choosing a small value of N will improve the performance of the model.

$$D(A, B) = \sqrt{(a-b)^2 + (a-b)^2 + ... + (a-b)^2} \tag{5}$$

4.2.2 Naïve Bayes

The reason why Naïve Bayes is called naive is its assumption:

1. attributes are independent
2. attributes are equally important

Since the CIFAR-100 dataset is an image set, background pixels are not as crucial as other pixels. However, by dimensional reduction, most background noise is reduced, feature selection will improve the performance.

$$P(\theta|\mathbf{D}) = P(\theta)\frac{P(\mathbf{D}|\theta)}{P(\mathbf{D})} \tag{6}$$

4.2.3 Random Forest

A single algorithm may be biased by its restriction. Random forest aggregate base algorithm with randomly selected attributes. This project aggregates 500 base decision trees while each tree will gradually extend from the root to the leaf nodes according to the selected random features. After all trees are generated, the 500 decision trees vote for the final prediction.

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x') \tag{7}$$

4.2.4 AdaBoost

Boosting is another ensemble classification theorem. AdaBoost[SF13] is one of the most well known boosting algorithms. Adaboost sets weight for all training data. After each boosting iteration, the weight of misclassified data will rise and others will decrease. In the next iteration, higher weight data are more likely to be included or even be selected multiple times. Thus, each iteration slightly improves the weakness of the last iteration. At the same time, the weight of each classifier with lower accuracy will be granted a lower weight, in the testing phase, these classifiers will perform a weighted vote. α_m is the weight of a base classifier, ϵ_m is the error rate of such base classifier.

$$\alpha_m = \frac{1}{2} \ln \left(\frac{1 - \epsilon_m}{\epsilon_m} \right) \tag{8}$$

4.2.5 Convolutional Neural Network

In this project, we build a ten layers Convolutional Neural Network based on Chris's work[Ver20]. The main reason for choosing a convolutional neural network eliminates extensive computational work associated with features extraction. Instead, it can efficiently learn and apply the corresponding features through a large pool of samples without any feature extraction pre-processing. For The CIFAR-100 dataset, skipping the feature extraction step will save developers time and effort while retaining all dataset information.

The Convolutional Neural Network of this project includes three CONV(Convolutional) layers that each followed by a pooling layer. We choose "ReLu" as the activation function of each CONV layer and "Adam" as the optimizer widely used. After that, we apply three dense layers to generate the actual classification based on the features extracted by CONV layers.

5 Experiment

5.1 Hardware and software specifications of the computers

Processor: Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz 2.81 GHz

RAM: 32.0 GB

Software: Google Colab

Runtime:
K-Nearest Neighbor: In this experiment, the model was trained with values of k from 1 to 20 to find the most suitable value of k. Therefore, it took a long time, about 15 minutes.As can be seen from the images below, this experiment will define the range of k as 1-20 and graph the accuracy to find the most appropriate value of k. See Figure 2.

Naïve Bayes: About 1 minute

Random Forest: About 5 minutes

AdaBoost: Using a hybrid model using Adaboost and Naïve Bayes, the training time is longer, 7 minutes 20s

Convolutional Neural Network: This experiment used a ten-layer convolutional model for training, and the training time was approximately 50 minutes. The figure below lists each of the layers used. See Figure 3.

5.2 Comparison of experimental results on different indicator

5.2.1 Accuracy

Accuracy is one of the metrics used to assess the performance of a model and is used to indicate the score that the model correctly predicts and is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

Accuracy comparison: See Figure 4.

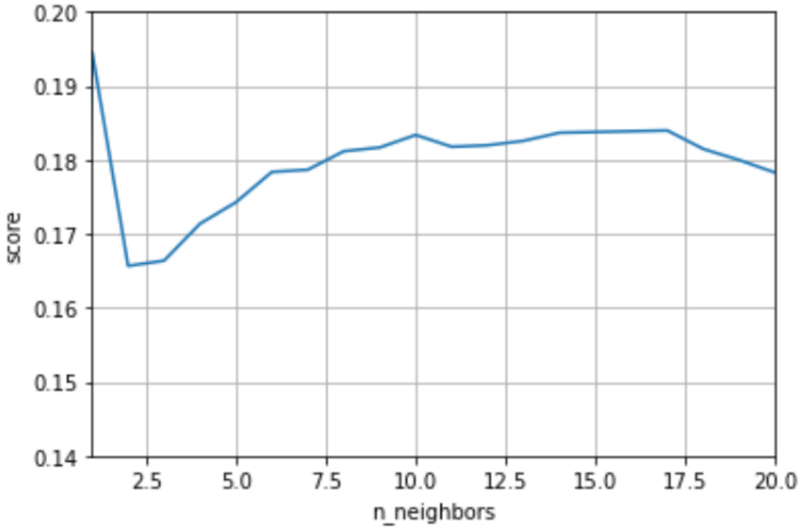
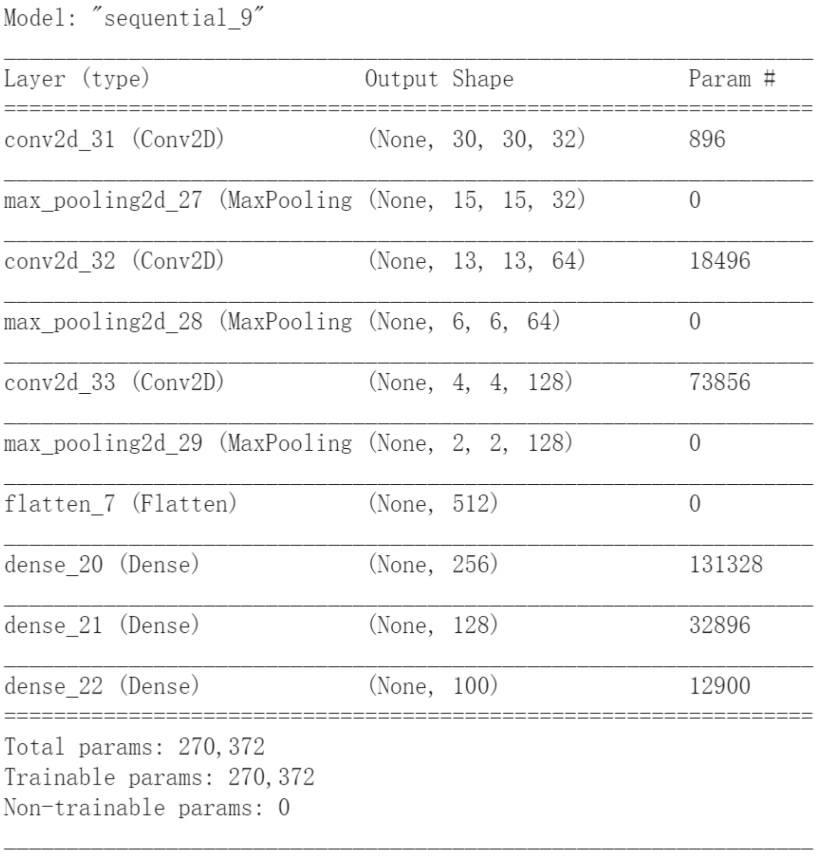


Figure 2: knn N tuning



```
Model: "sequential_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_31 (Conv2D)           (None, 30, 30, 32)        896
_________________________________________________________________
max_pooling2d_27 (MaxPooling (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_32 (Conv2D)           (None, 13, 13, 64)        18496
_________________________________________________________________
max_pooling2d_28 (MaxPooling (None, 6, 6, 64)          0
_________________________________________________________________
conv2d_33 (Conv2D)           (None, 4, 4, 128)         73856
_________________________________________________________________
max_pooling2d_29 (MaxPooling (None, 2, 2, 128)         0
_________________________________________________________________
flatten_7 (Flatten)          (None, 512)               0
_________________________________________________________________
dense_20 (Dense)             (None, 256)               131328
_________________________________________________________________
dense_21 (Dense)             (None, 128)               32896
_________________________________________________________________
dense_22 (Dense)             (None, 100)               12900
=================================================================
Total params: 270,372
Trainable params: 270,372
Non-trainable params: 0
_________________________________________________________________
```

Figure 3: CNN summary

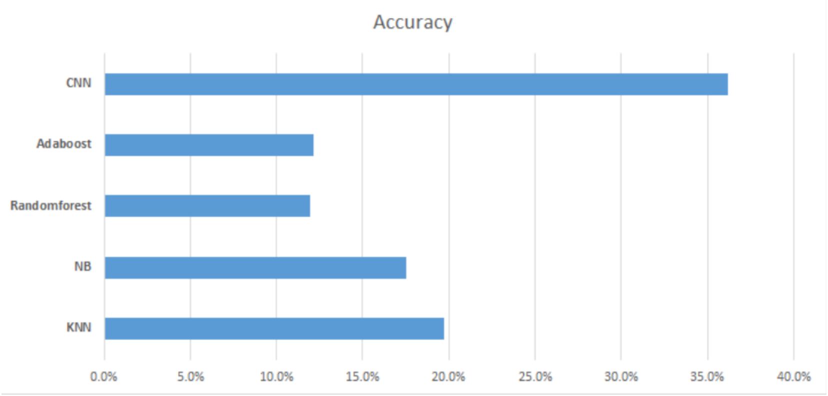


Figure 4: Accuracy comparison

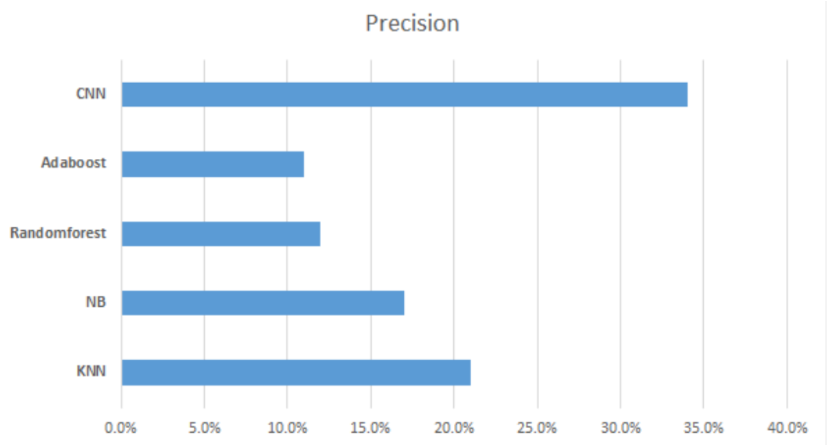


Figure 5: Precision Comparison

5.2.2 Precision

Accuracy is also an important metric used to assess the performance of the model, and it is used to indicate the percentage of the data that the model predicts to be positive that is actually positive. The function is:

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

Precision Comparison: See Figure 5.

It is clear from the graph that the CNN has a higher precision rate, which means that more of the positive samples predicted by the CNN are actually positive.

5.2.3 Recall

Recall refers to the percentage of samples that are positive out of those that are correctly predicted.The function is:

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

Recall comparison: See Figure 6.

From the graph we can see that the CNN algorithm has a higher Recall.

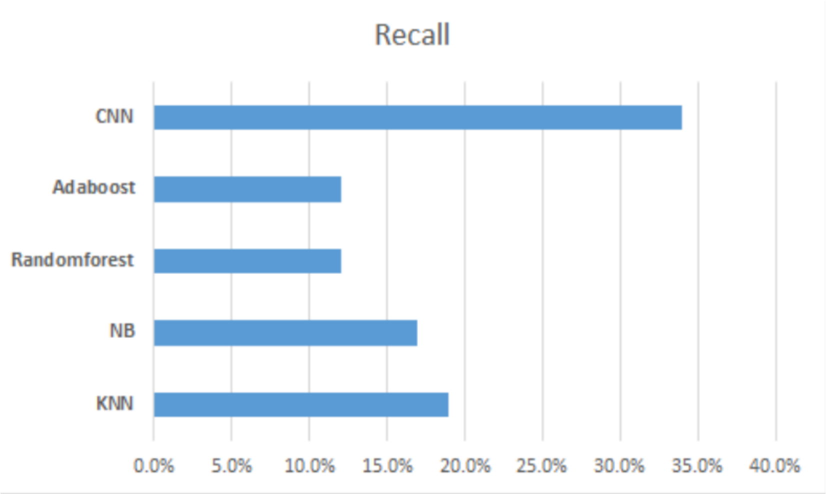


Figure 6: Recall comparison

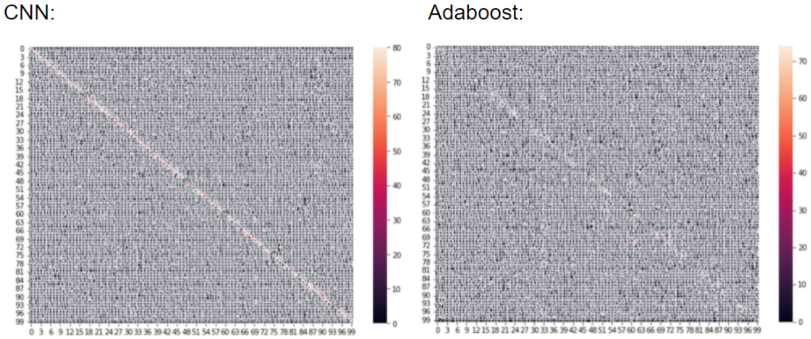


Figure 7: Confusion matrix of CNN and Adaboost

5.2.4 Confusion matrix

The confusion matrix is a table that is typically used to describe the performance of a classification model on a set of test data. As the dataset for this experiment has 100 categories, it was chosen to present the confusion matrix as a heat map so that the performance of each method could be compared more visually [Mar20]:

See Figure 7, 8, and 9.

As can be seen in these heat maps, there is a more pronounced line in the heat maps of KNN and CNN compared to the other algorithms, which proves that these two algorithms have a better performance in classifying CIFAR-100 more clearly.

5.2.5 Conclusion

By comparing the accuracy, precision, regression, and confusion matrix of each classification algorithm, the Convolutional Neural Network is undoubtedly the best performing algorithm, as shown in the summary performance chart above, where it comes out way ahead in each of the compared methods.

5.2.6 Reflection

One of the first reflections from this experiment is the issue of overfitting. Although CNNs have better performance and higher accuracy than other classification algorithms, 50 epochs is clearly too much, so that the accuracy is in a state of little change later on, which flattens out and wastes more time.Another counterpoint is tuning, which was not done too many times on some methods in this experiment and could be one of the possible reasons for the lower accuracy of the other classification algorithms.

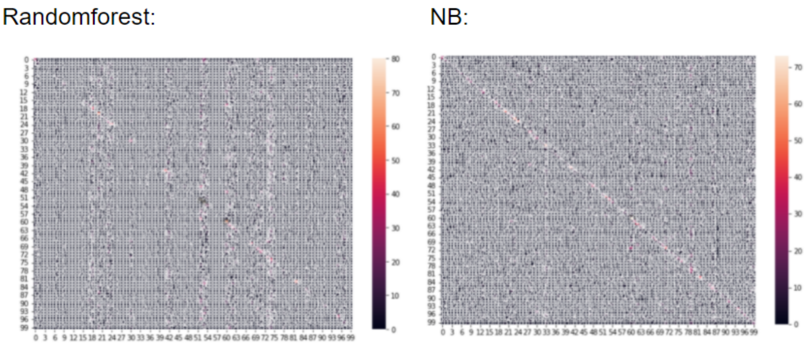


Figure 8: Confusion matrix of Random Forest and Naive Bayes

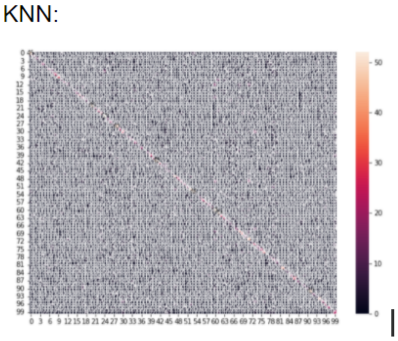


Figure 9: Confusion matrix of KNN

6 Conclusion

In summary, this experiment used the KNN, CNN, Adaboost, Random forest, and NB classification algorithms to classify the CIFAR-100 dataset. The performance of each algorithm was evaluated using accuracy, precision, recall, and confusion matrix, and the CNN algorithm was considered the best algorithm in this experiment as it was in the top position in all the metrics. However, one of the shortcomings of the CNN algorithm is that the training time is relatively slow. Hence, a recommendation for future work drawn from this experiment is to avoid the risk of overfitting, leading to long computing times and even reduced accuracy.

7 Appendix

7.1 Contribution

See Table 1.

7.2 How to run the code

Step1: Download the CIFAR-100 dataset from the website https://www.cs.toronto.edu/ kriz/cifar.html and unzip the folder 'cifar-100-python' into the same directory as the two ipynb files.
Step2: Since this experiment was conducted on colab, the count function of colab was used to store

Table 1: Contribution Table

Name	Unikey	SID	SID
FEIFAN GAO	fgao2584	510583158	50%
RUIXIAN LIU	rliu0898	510127572	50%

the dataset, and the code shown in the image below can be commented or deleted before running the code (both2 ipynb file)

```python
from google.colab import drive
drive.mount('/content/drive')
```

Step3: Changing the paths of the three load datasets in the image.

```python
# load data
train = unpickle('./drive/MyDrive/cifar-100-python/train')
X_train = train.get(b'data')
y_train = train.get(b'fine_labels')

name = load_labels_name('./drive/MyDrive/cifar-100-python/meta')

test = unpickle('./drive/MyDrive/cifar-100-python/test')
X_test = test.get(b'data')
y_test = test.get(b'fine_labels')
print(X_test.shape)
```

Remove '/drive/MyDrive' from the three paths (train,meta,test) (keep the leading '.') , so that the data should load properly for both files.
Step4: The 2 ipynb files should be able to run properly.

References

[HSH19] Hayder Hasan, Helmi ZM Shafri, and Mohammed Habshi. A comparison between support vector machine (svm) and convolutional neural network (cnn) models for hyperspectral image classification. In *IOP Conference Series: Earth and Environmental Science*, volume 357, page 012035. IOP Publishing, 2019.

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[LLLH17] Peifeng Liang, Weite Li, Donghang Liu, and Jinglu Hu. Large-scale image classification using fast svm with deep quasi-linear kernel. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1064–1071. IEEE, 2017.

[Mar20] Kevin Markham. Simple guide to confusion matrix terminology, Feb 2020.

[SF13] Robert E Schapire and Yoav Freund. Boosting: Foundations and algorithms. *Kybernetes*, 2013.

[Ver20] Christian Versloot. How to build a convnet for cifar-10 and cifar-100 classification with keras?, Apr 2020.