CIFAR-10 Image Classification using Convolutional Neural Networks (CNNs)

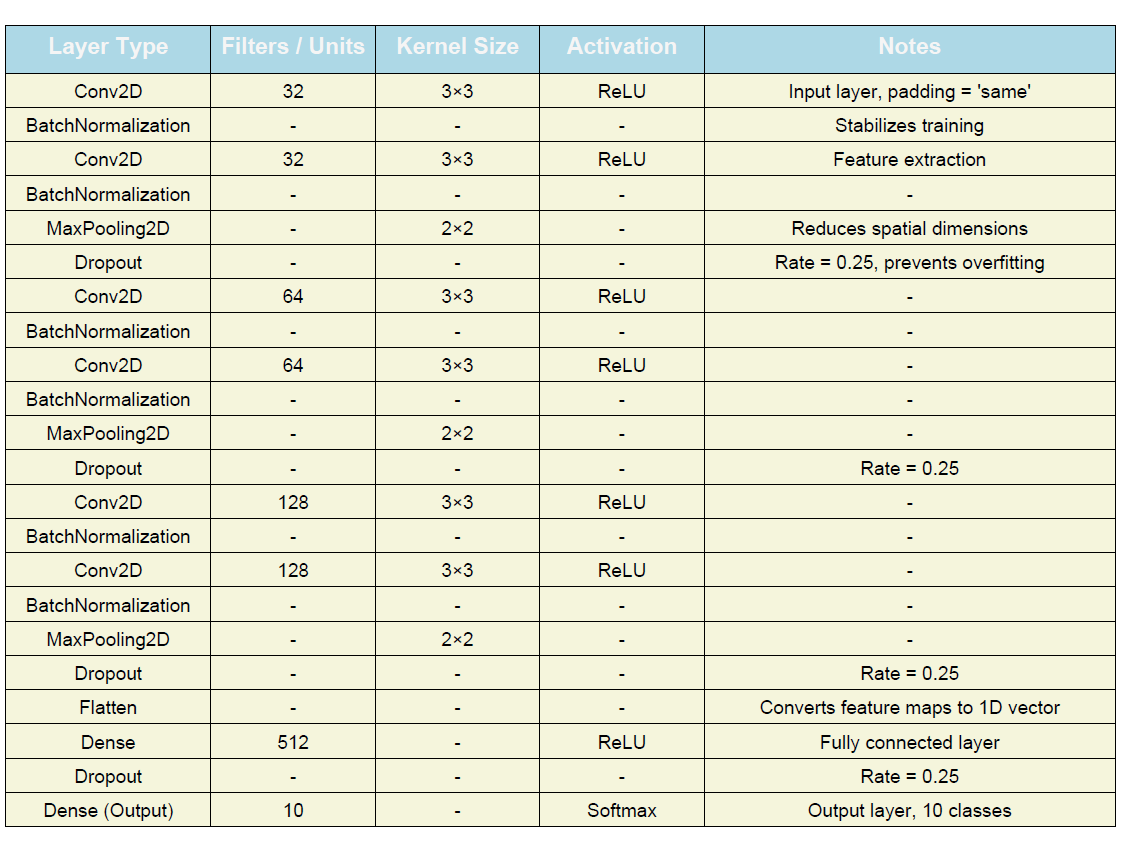
Abstract

Convolutional neural network (CNN) is a powerful tool that can be used in many applications of machine learning. This report demonstrates the effectiveness of using a CNN to classify images in the CIFAR-10 dataset. Convolutional Neural Network (CNN) is a type of Deep Neural Networks (DNN) that contains multiple layers such as Conv layers, integration layer and fully integrated layer. r ability to automatically and adaptively extract hierarchical structures from images helps them recognize and understand complex spatial patterns. This makes CNNs a strong and useful tool for analyzing images. In this work, the ten standard object categories from the CIFAR-10 dataset were considered, and CNN was trained to classify these images. During training and testing, the model's performance was measured using factors like accuracy and validation loss. These measurements help determine how well the chosen model worked.  This project uses a Convolutional Neural Network (CNN) to classify images from the CIFAR-10 dataset. The dataset includes 60,000 images, each 32x32 pixels in colour, and they are divided into 10 categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The CNN was built using the Keras and TensorFlow tools. It uses several layers like convolutional layers, batch normalization, pooling layer, dropout, and dense layers.  
To help the model perform better on different types of images, data augmentation techniques were used.  
These techniques include flipping images horizontally and slightly shifting their positions.  The trained model achieved a test accuracy of around 89.42%, showing how effective CNNs can be for classifying small images. The evaluation included loss and accuracy charts, a confusion matrix, and a classification report.

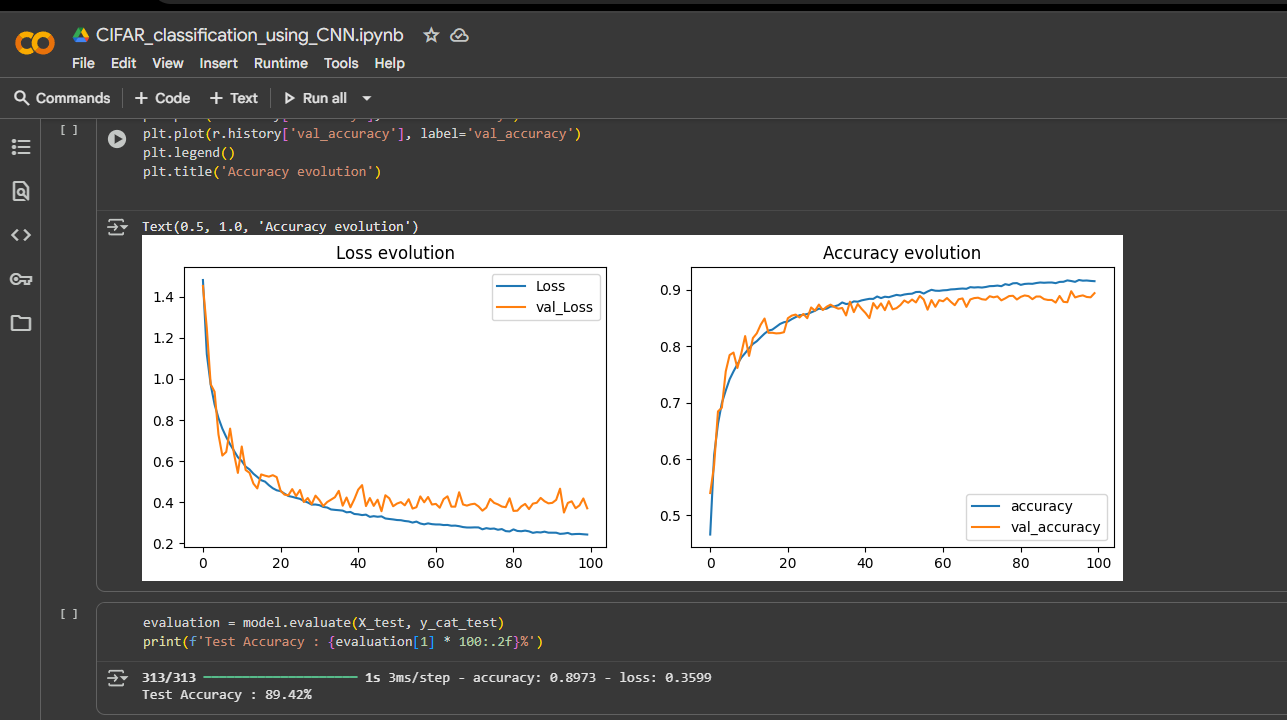
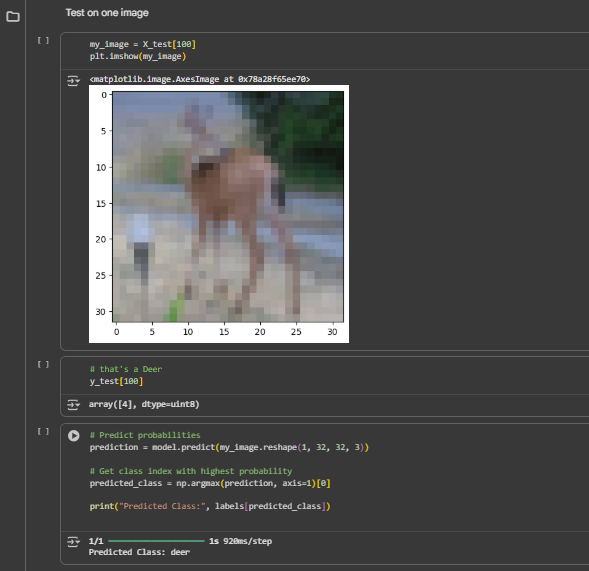
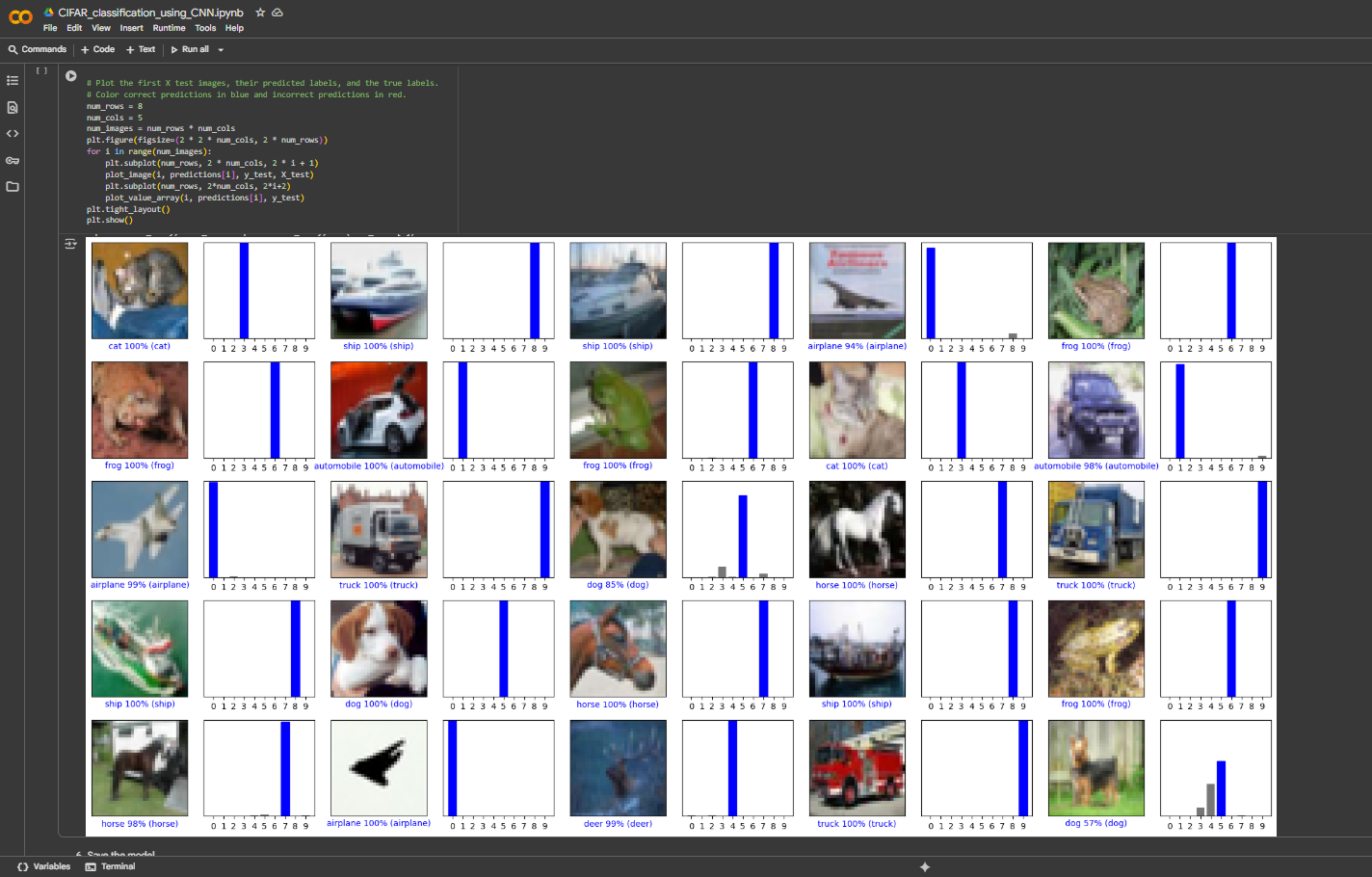
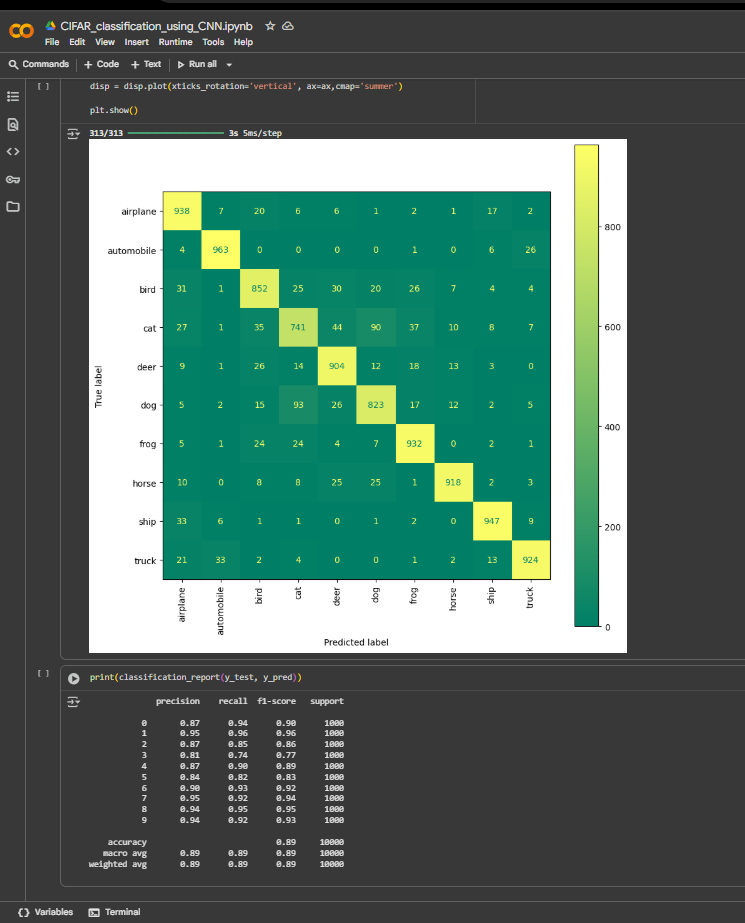
Image classification is a very important issue in digital image analysis. CNN is type of Deep Neural Networks (DNN) that contains multiple layers such as Conv layers, integration layer and fully integrated layer. Convolutional neural networks (CNNs) are widely used in pattern-and picture- recognition problems as they have a number of advantages compared to others strategies. CIFAR-10is a very popular computer vision database. This database is well read in it many types of in-depth study of object recognition. This database contains 60,000 images separated by 10 target classes, each a section containing 6000 images of 32 \* 32 shapes. This database contains images of low-resolution (32 \* 32), which allows researchers to experiment with new algorithms. Image classification is a fundamental task in computer vision that involves assigning a label or category to an image. Convolutional Neural Networks (CNNs) have become the state-of-the-art method for image classification due to their ability to automatically learn features from raw pixel values. In this project, we aim to build a CNN model that can accurately classify images in the CIFAR-10 dataset.

Convolutional neural network (CNN) is a powerful tool that can be used in many applications of machine learning. This paper demonstrates the effectiveness of using a CNN to classify images in the CIFAR-10 dataset.

Introduction  
  
**Image** **classification** is a **key** **task** in **computer** **vision**.It **has** **many** **uses** in **real** **life**, like **helping** **self**-**driving** **cars**, **supporting** **medical** **diagnosis**, **using** **cameras** for **security**. The CIFAR-10 **dataset** is **commonly** **used** to **test** **machine** **learning** **models** on **small** **images**.  
  
**CNNs** are **especially** **good** for **image** **data** since they can **automatically** **find** **features** like **edges**, **textures**, and **parts** of **objects** through a **process** **called** **convolution**.  
This **project** **shows** how to **build**, **train**, and **evaluate** a CNN for **classifying** **images** from the CIFAR-10 **dataset**. It **also** **shows** how to **look** at **intermediate** **results** and **performance** **metrics**.  
  
Methodology  
  
**1.Dataset Loading**  
  
The CIFAR-10 **dataset** was **imported** from tensorflow.keras.datasets.  
  
- **Training** **set**: 50,000 **images**  
- **Test** **set**: 10,000 **images**  
- Each **image** is 32x32 **pixels** with three **colour** **channels** (**red**, **green**, **blue**)

**2.Data Preprocessing**  
  
- Pixel **values** were **scaled** between 0 and 1  
- Labels were **converted** to one-**hot** **encoding** **using** **to\_categorical** to **match** the **categorical** **crossentropy** **loss** **function**  
  
**3.Data Augmentation**  
  
To **improve** the **model**’s **ability** to **generalize**:  
  
- The **ImageDataGenerator** was **used** to **apply** **random** **horizontal** **flips**, and **random** **shifts** in **width** and **height** (10% each)  
- This **helps** **create** **more** **varied** **training** **data** without **collecting** **new** **images**  
  
**4.CNN Model Architecture**  
  
A **Sequential** CNN **model** was **built** with these **layers**:  
  


**5.Model Compilation**  
  
- Optimizer: Adam  
- **Loss** **function**: Categorical Crossentropy  
- **Metrics**: Accuracy  
  
**6.Model Training**  
  
- **Trained** for 100 **epochs** with a **batch** **size** of 32  
- Validation was **done** **using** the **test** **set**  
- EarlyStopping was **optionally** **used** to **avoid** **overfitting**

Results  
  
**1.Training and Validation Curves**  
  
- **Loss** Curve: Both **training** and **validation** **loss** **decreased** **steadily** as the **model** **trained**  
- **Accuracy** Curve: **Training** **accuracy** **reached** around 89.42%, and **validation** **accuracy** **reached** around 89.73%  
- **Observation**: The **model** didn’t **overfit** **much** **due** to **dropout** and **data** **augmentation**  
  
  
  
**2.Confusion Matrix**  
  
- The **confusion** **matrix** was **created** **using** sklearn.metrics.**ConfusionMatrixDisplay**  
- **Correct** **predictions** are **shown** along the **diagonal**  
- Mistakes **often** **happened** between **similar** **classes**, like **cat** and **dog**, or **automobile** and **truck**  
  
  
**3.Classification Report**  
  
- **Metrics**: Precision, Recall, F1-**Score** for each **class**  
- The **model** **had** **high** **precision** and **recall** for **classes** like **frog**, **ship**, and **airplane**  
- Some **confusion** **occurred** for **classes** with **similar** **features**  
  
**4.Sample Predictions**  
  
- **Random** **test** **images** were **visualized** along with their **predicted** and **true** **labels**  
- **Correct** **predictions** are **shown** in **blue**, **incorrect** in **red**  
- **Probability** **bar** **charts** were **displayed** for all 10 **classes** per **image**  
  
  
   
Conclusion  
  
The CNN **successfully** **classified** CIFAR-10 **images** with **strong** **performance**.  
Data **augmentation** and **dropout** **helped** **prevent** **overfitting**. A **simple** CNN **achieved** **good** **performance** as a **baseline**, and **more** **complex** **models** or **transfer** **learning** could **help** **improve** **accuracy** **further**. **Future** **work** could **involve** **experimenting** with **models** like ResNet or VGG, or **tuning** **hyperparameters**.  
  
References  
  
- Krizhevsky, A., & Hinton, G. (2009).  
**Learning** **multiple** **layers** of **features** from **tiny** **images**. CIFAR Dataset.  
- Chollet, F. et al. (2015).  
Keras: Deep Learning for **Humans**.  
- TensorFlow Documentation: https://www.tensorflow.org  
- **Scikit**-**learn** **Documentation**: https://**scikit**-**learn**.org  
  
This **version** is **ready** to be **exported** as a **PDF** with **figures** **included** for:  
  
- **Training**/**validation** **curves**  
- **Confusion** **matrix**  
- Sample **image** **predictions**