CS3315 Final Project
CIFAR10 Image Classification
Christopher Clark
9 Dec 2022

Introduction

The project was selected based on current thesis research. My thesis research is in generating adversarial examples with minimal perturbations capable of fooling neural networks. My current work is with MNIST (Modified National Institute of Standards and Technology) data and the next step is to run the algorithm for generating adversarial examples on the CIFAR10 (Canadian Institute For Advanced Research) dataset. The research was focused on building, training, and comparing different machine learning models on their ability to classify CIFAR10 data based on accuracy and loss. The hypothesis is that a convolutional neural network (CNN) will be the best model.

Selection of Data

The CIFAR10 dataset is a collection of 60,000 32 x 32 pixel images that have been classified by people with 10 classification labels: airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. This dataset is well organized and is used often in machine learning research. There was not data cleaning required. The only modification was to normalize the dataset pixel values to be between zero and one, vice zero and 255. There is no missing and malformed data due to the nature of this dataset for academic research.

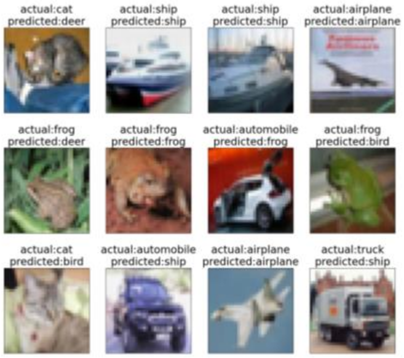


Fig 1. Example of results after training kNN

Methods

Four machine learning models were created and trained on CIFAR10: k-Nearest Neighbors (kNN), Support Vector Machine (SVM), Decision Tree (DT), and CNN. Each model was trained with the CIFAR10 training data using the Keras dataset test/train split. Grid Search was used with kNN, SVM, and DT models to determine the best hyperparameters for each based on test and training accuracy as applicable.

Several combinations of different numbers of convolutional layers were used to try to find the best CNN model. Number of layers ranged from 16 to 134. It was expected that adding additional layers (with batch normalization and pooling) would result in greater test accuracy and loss.

Results

k-Nearest Neighbors

Using GridSearch, the best k value was determined to be k=5. Figure 2 shows that k=12 was slightly better, but that is because the values in GridSearch were multiples of 5 and did not include 12 as a parameter. This would only produces a slightly better model. The best test accuracy for kNN was 34% as indicated in both Figures 2 and 3.

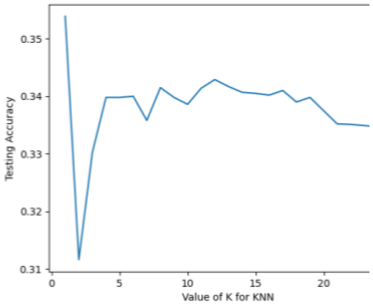
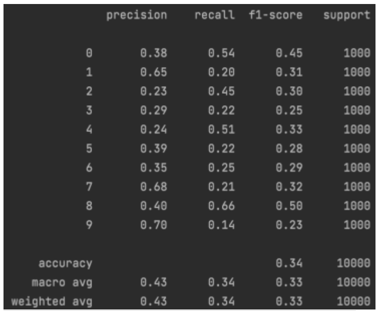


Fig 2. kNN Test Accuracy

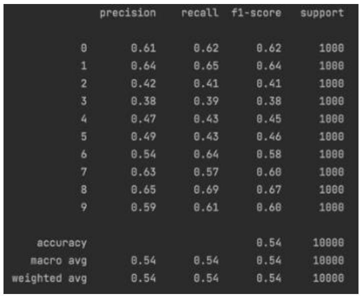


```
              precision    recall  f1-score   support

           0       0.38      0.54      0.45      1000
           1       0.65      0.20      0.31      1000
           2       0.23      0.45      0.30      1000
           3       0.29      0.22      0.25      1000
           4       0.24      0.51      0.33      1000
           5       0.39      0.22      0.28      1000
           6       0.35      0.25      0.29      1000
           7       0.68      0.21      0.32      1000
           8       0.40      0.66      0.50      1000
           9       0.70      0.14      0.23      1000

    accuracy                           0.34     10000
   macro avg       0.43      0.34      0.33     10000
weighted avg       0.43      0.34      0.33     10000
```

Fig 3. kNN Confusion Matrix

Support Vector Machine

It was very difficult to generate a lot of data for the SVM model. This is due to the complex nature of CIFAR10 images and the use of the rbf kernel. This model performed better than kNN as indicated in the confusion matrix in Figure 4. SVM had a 54% test accuracy.

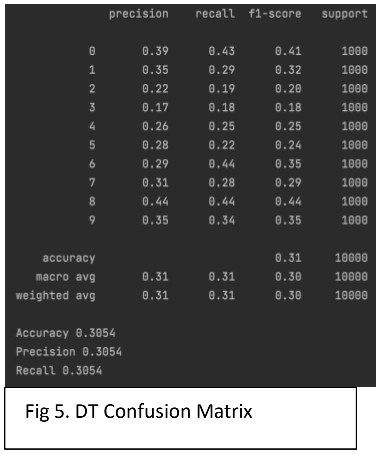


```
              precision    recall  f1-score   support

           0       0.61      0.62      0.62      1000
           1       0.64      0.65      0.64      1000
           2       0.42      0.41      0.41      1000
           3       0.38      0.39      0.38      1000
           4       0.47      0.43      0.45      1000
           5       0.49      0.43      0.46      1000
           6       0.54      0.64      0.58      1000
           7       0.63      0.57      0.60      1000
           8       0.65      0.69      0.67      1000
           9       0.59      0.61      0.60      1000

    accuracy                           0.54     10000
   macro avg       0.54      0.54      0.54     10000
weighted avg       0.54      0.54      0.54     10000
```

Fig 4. SVM Confusion Matrix

Decision Tree

GridSearch was used with the Decision Tree model to determine the best max_depth value was 10. The Decision Tree had the lowest performance out of all of the models as seen in Figure 5 with a test accuracy of 31%.



```
              precision    recall  f1-score   support

           0       0.39      0.43      0.41      1000
           1       0.35      0.29      0.32      1000
           2       0.22      0.19      0.20      1000
           3       0.17      0.18      0.18      1000
           4       0.26      0.25      0.25      1000
           5       0.28      0.22      0.24      1000
           6       0.29      0.44      0.35      1000
           7       0.31      0.28      0.29      1000
           8       0.44      0.44      0.44      1000
           9       0.35      0.34      0.35      1000

    accuracy                           0.31     10000
   macro avg       0.31      0.31      0.30     10000
weighted avg       0.31      0.31      0.30     10000

Accuracy 0.3054
Precision 0.3054
Recall 0.3054
```

Fig 5. DT Confusion Matrix

Convolutional Neural Network

The CNN had the best performance as anticipated with a test accuracy of 83% and test loss of 0.77. The best CNN was determined by trying different layers and finding the model with the highest training/test accuracy and lowest training/test loss. As additional layers were added, the model accuracy and loss became highly erratic. The much smaller model, with only 16 layer, had the best performance as seen in Figure 6.

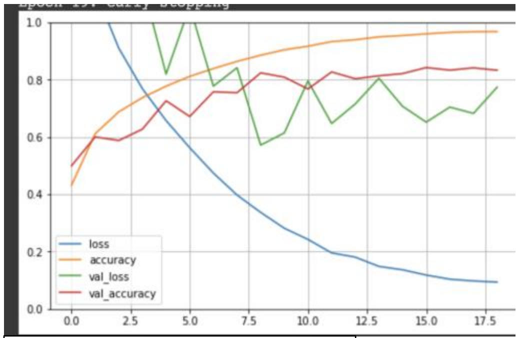


Fig 6. CNN Training/Test Output

This model was used with the A* Search algorithm for generating adversarial examples with examples of successful adversarial examples shown in figure 7. The algorithm generated 100 adversarial examples with the following outcome:

- Average L_2 = 1.16
- Attack Success: 83 %
- Total Run Time: 3.3 hr

These values will likely be improved when using a model with lower test loss and high accuracy.

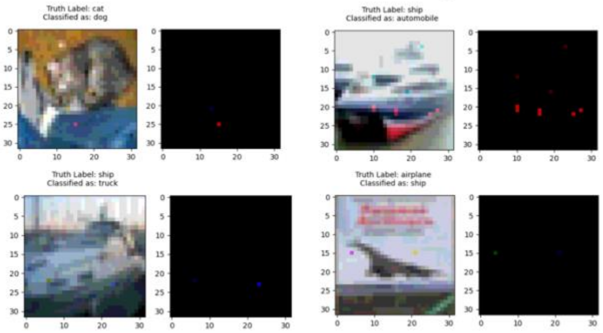


Fig 7. Successful Adversarial Examples

Discussion

Model	Validation Accuracy
CNN	0.83
SVM	0.54
kNN	0.34
Decision Tree	0.31

The CNN model was in fact the best model for classifying CIFAR10 data as shown in the following table. While this was the best model used in the project, the test loss is not as low as would be expected. During a literature review, the Wide Residual Network (ResNet) is the preferred model for classifying CIFAR10 data. This model includes a skip output where the output from one layer is both passed as input to the next layer and also skips over the next layer to be passed as input to the following layer. The expected model performance for a ResNet can be seen in figure 8. This model has the idealized test accuracy and loss for CIAF10 data classification.

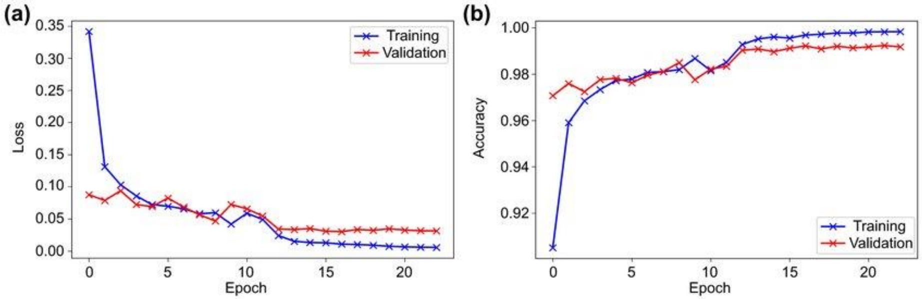


Fig 8. Naushad, Raoof & Kaur, Tarunpreet & Ghaderpour, Ebrahim. (2021). Deep Transfer Learning for Land Use and Land Cover Classification: A Comparative Study. Sensors. 21. 8083. 10.3390/s21238083.

Summary

The convolutional model provides the best model for CIFAR10 image classification. The inclusion of the skip connection allows larger CNN models to perform in-line with what is needed for the next step of the thesis research.

Github link: https://github.com/Christopher-d-clark5/CS3315_Final_Project

References

1. Naushad, Raoof & Kaur, Tarunpreet & Ghaderpour, Ebrahim. (2021). Deep Transfer Learning for Land Use and Land Cover Classification: A Comparative Study. Sensors. 21. 8083. 10.3390/s21238083.