CP322: Mini Project 3

Alexandros Ioannou, Duc Nguyen Minh, Florian Novak, Saumya Patel

I. Abstract

In this project, we addressed image classification problems by developing different Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) and comparing their performances. Therefore, we used the CIFAR-10 dataset, consisting of a pre-defined training set with 50,000 labeled, mutually exclusive images of 10 different categories of images. After the pixel inputs, we developed and trained several models and compared their accuracies and losses on the unseen test set. In the end, we find that the CNN does a way better job of classifying images than ANN, which we mostly relate to the ability of CNNs to extract features out of the image with its convolutional layers.

II. Introduction

Image Recognition and Classification is a trending topic in current technologies, with the newest Language Processing Models like ChatGPT 4.0 starting to be able to recognize and describe content from given input pictures. Underlying all these trends is the so-called “computer vision”, meaning machines that are able to identify and interpret input image data. This computer vision can be created in different ways, two of them, namely ANNs and CNNs, are compared and tested in the following. In our project, we created different image classifiers and compared their performances on classifying unseen images. For this, we loaded the CIFAR-10 dataset with 50,000 labeled training images and 10,000 testing images. Every image was 32x32 pixels large and came with 3 different values for its color, namely its RGB values. We developed different Neural Networks to compare how well they perform in classifying the images. On the one hand, we created ANNs, to see how well a deep neural network can actually classify images. On the other hand, we created CNNs, which are seen as gold-standard for image classification in the AI field. With all its convolutional and pooling layers, we found that the CNN did a much better job in classifying the images, reaching an test-accuracy of nearly 80% compared to 63% of the ANN.

III. Related Work

In prior work related to image classification tasks with Neural Networks across different industries, all found that CNN is the better fit for this purpose compared to ANN (Kareem et al. 2021, Shah et al. 2023, Xing & Wang 2019). It was consistently in the finding that CNNs outperformed ANNs in image classification tasks, which also aligns with our key findings. This superiority is attributed to CNNs' deeper structures and specialized feature extraction capabilities, which are more adept at handling image data. While ANNs have their applications, these studies recommend CNNs for image classification due to their efficiency and accuracy in processing complex visual information.

IV. Dataset

The used CIFAR-10 dataset is a dataset consisting of 60,000 labeled images from ten different classes, evenly distributed with 6,000 images per class. The list of the classes can be seen in the appendix. Every image in the sets is mutually exclusive for one category, making it a single-item classification task. The CIFAR-10 dataset comes with an integrated training set with the size of 50,000 images and a testing set of 10,000 images. In both of these sets, the ten categories are evenly balanced. Every image consists of 32x32 pixels and 3 numerics for the RGB value for the corresponding pixels. Before being able to work with this data, we preprocessed the RGB values to be between 0 and 1 with the process of normalization. This preprocessing makes it easier for our models to get trained by the input values.

V. Proposed Approach

We developed several different ANNs and CNNs to find the best-fitting model for our classification task.

ANN:

For the ANN architecture, we began by flattening the layers, transforming the input images from a 3D shape (32x32 pixels with 3 color channels) into 1D vectors, since the subsequent layers (Dense Layers) require 1D input. Then follow three hidden layers. The hidden layers consist of 512, 256, 128 neurons respectively and all of them use the ReLu activation function to handle non-linearities. Finally, the output layer is a dense layer that has 10 neurons, corresponding to the 10 output classes. A BatchNormalization layer was also used during our experiment to standardize the inputs to a layer for each mini-batch, but we excluded it since in our case it worsened the models accuracy.

- Input layer: Processes 32x32x3 input images.
- Hyperparameters: Learning Rate(0.0001), Optimizer(Adam), Batch size(32), Episodes(20).
- 3 Hidden layers (512, 256, 128 neurons for each layer).
- Output Layer: 10 neurons for 10 classes

CNN:

For the CNN architecture, we developed a hierarchical convolutional layer strategy, trying to find the best model in regards to the trade-off between overfitting, out-of-sample accuracy and complexity. We found our best model by progressively increasing filter complexity (32, 64, 128) to extract increasingly abstract features from the 32x32x3 input images. Batch normalization follows each convolutional layer to stabilize learning, while max pooling layers reduce spatial dimensions, enhancing feature robustness. To mitigate overfitting, dropout layers were strategically placed in the network. The architecture culminates in a flattening process and dense layers, transforming the multi-dimensional feature maps into a vector space for classification into 10 distinct categories, which then classifies the input image into one of the 10 given categories. For the best accuracy of the model, hyper-parameter tuning was necessary.

- Input layer: Processes 32x32x3 input images.

- Hyperparameters: Learning Rate(0.0001), Optimizer(Adam), Batch size(32), Episodes(20).
- Convolution layers
    - First layers: 32 filters (3x3), ReLU activation.
    - Second layers: 63 filters (3x3), ReLu activation.
    - Third + fourth layers: 128 filters (3x3), ReLu activation, 'same' padding.
- Batch normalization: applied after each convolutional layer to stabilize learning.
- Max pooling layers: reduce spatial dimensions, enhancing feature robustness.
- Dropout layers: mitigate overfitting, rate set at 0.5.
- Flattening process: transforms 2D feature map into a 1D vector.
- Dense layer: one layer with 64 neurons, final output layer with 10 neurons for classification

Our CNN model gets better at spotting different features in images by using more complex filters step by step. Then we added batch normalization and max pooling to make the learning more better and stable. We also apply dropout layers at certain points to stop the model from overfitting. Model ends with a dense layer, which helps in sorting the images into one of 10 groups. Fine-tuning details like learning rate, optimizer, batch size, epochs was really important. We chose these settings to make sure our model doesn't overfit and still does a good job at classifying images correctly.

VI. Results

When comparing our best ANN with the best CNN, we see that the CNN significantly outperforms the ANN in classifying image data when looking at the test losses and test accuracy, meaning how the model performs on new, unseen images. We have also tested our models with different learning rates (0.01,0.001,0.0001,0.00001) and found the optimal ones to be 0.0001 for both models, see Figures 7 & 8.

ANN: (Figure 1)

- Peak accuracy after 20 epochs: 63.80%
- Best Validation Accuracy: 53.66%
- Corresponding Loss: Approx. 1.3293
- The ANN shows signs of overfitting (Figure 2 & 3), which is shown from the confusion matrix with lower recall and precision in certain classes (bird, cat, deer, dog). This means that while ANN learned the training data well, it does not generalize this learning to new data as efficiently.
- For the ANN, classes like "dog", "deer" and "truck" have higher recall, suggesting the ANN was able to identify these categories more frequently but not as accurately.

CNN: (Figure 4)

- Peak accuracy after 20 epochs: 80.21%
- Best Validation Accuracy: 75.98%
- Corresponding Loss: Approx. 0.7028
- The CNN performs well across all classes (Figure 5 & 6), with no signs of overfitting. The precision and recall values across the classes are higher and more balanced. The precision and recall values across the classes are higher and more balanced, indicating that CNN learned to generalize better than ANN. CNNs are inherently more suitable for image data due to their ability to capture spatial hierarchies in data.

- The slightly lower recall values in certain classes for the CNN suggest that while the model generalizes well, it might be facing some learning difficulties. This could possibly be improved with fine-tuning of regularization techniques, such as adjusting dropout rates or adding weight decay.
- The CNN performs well across all classes, with particularly high scores in “automobile” and “ship”. However, it seems to struggle slightly with “cat” and “dog” classes.

VII. Discussion and Conclusion

This report presents a comprehensive comparison between ANNs and CNNs for image classification using the CIFAR-10 dataset. Our findings reinforce the prevailing view: CNNs are substantially more effective than ANNs in image classification tasks. Our CNN model achieved a peak accuracy of 80.21% and a validation accuracy of 75.98%, significantly outperforming the ANN, which only achieved a peak accuracy of 63.80% and a validation accuracy of 53.66%. The ANN showed overfitting issues and struggled to generalize its learning to new data, while the CNN showed high precision and recall across classes, indicating generalization capabilities. However, the CNN did face some difficulties in certain classes, like “cat” and "dog", indicating areas for improvement. Adjustments in regularization techniques, such as tuning the dropout, might enhance its performance. While the ANN did face challenges in generalizing this specific data, they remain useful for other applications like pattern recognition, text classification, and time-series prediction, where data is more uniform. This report showcases CNN's dominance in image classification while acknowledging the utility of ANNs for other applications.

VIII. Statement of Contributions

- **Alexandros Ioannou:** Implemented ANN, tested and plotted different learning rates for models, contributed to the writeup.
- **Duc Nguyen Minh:** Contributed to Write Up. Refinement of CNN for Enhanced Accuracy.
- **Florian Novak:** Loading and Preprocessing of Data. Development of CNN and according evaluation graphics. Contributed to Write Up.
- **Saumya Patel:** Write up

IX. References

1. Kareem, S., Hamad, Z. J., & Askar, S. (2021). An evaluation of CNN and ANN in prediction weather forecasting: A review. *Sustainable Engineering and Innovation*, *3*(2), 148-159.
2. Shah, A., Shah, M., Pandya, A., Sushra, R., Sushra, R., Mehta, M., ... & Patel, K. (2023). A Comprehensive Study on Skin Cancer Detection using Artificial Neural Network (ANN) and Convolutional Neural Network (CNN). *Clinical eHealth*.
3. Xin, M., & Wang, Y. (2019). Research on image classification model based on deep convolution neural network. *EURASIP Journal on Image and Video Processing*, *2019*, 1-11.

X. Appendix

Figure 1: Accuracy and Loss of ANN

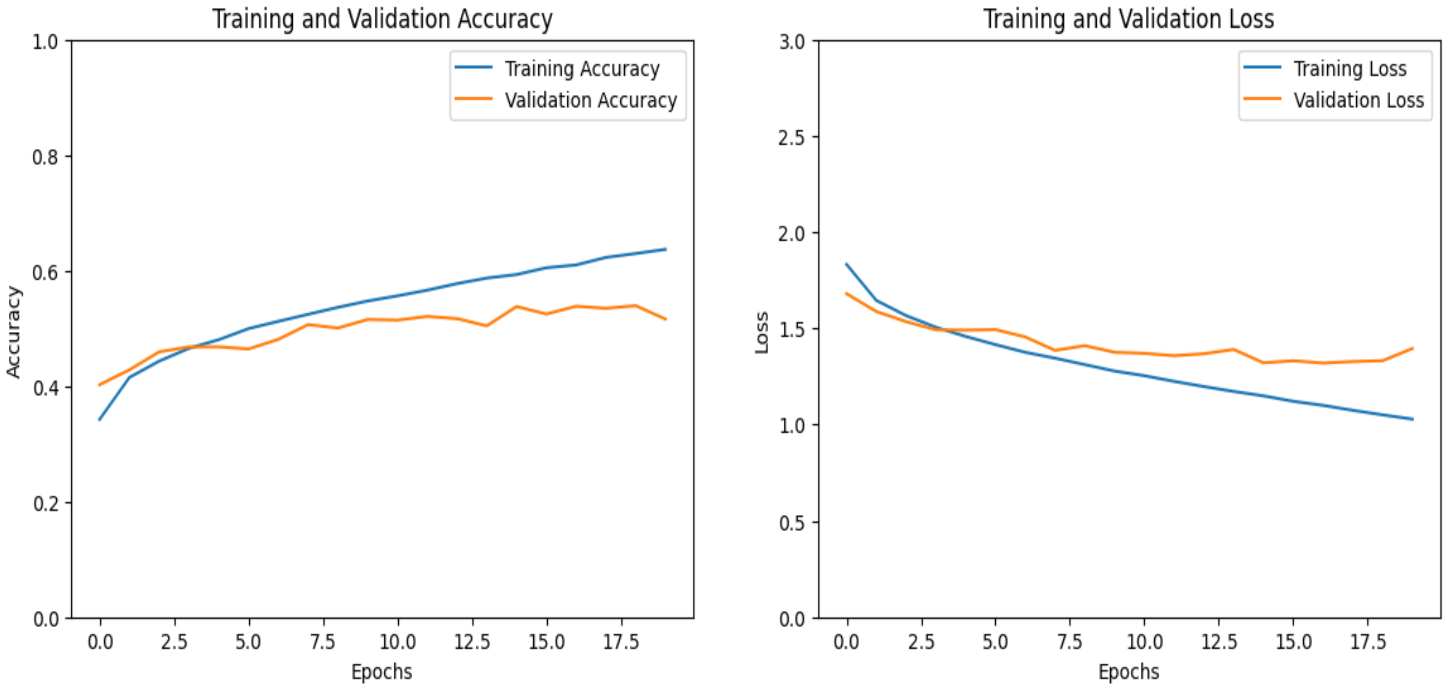

Figure 2: Confusion Matrix ANN

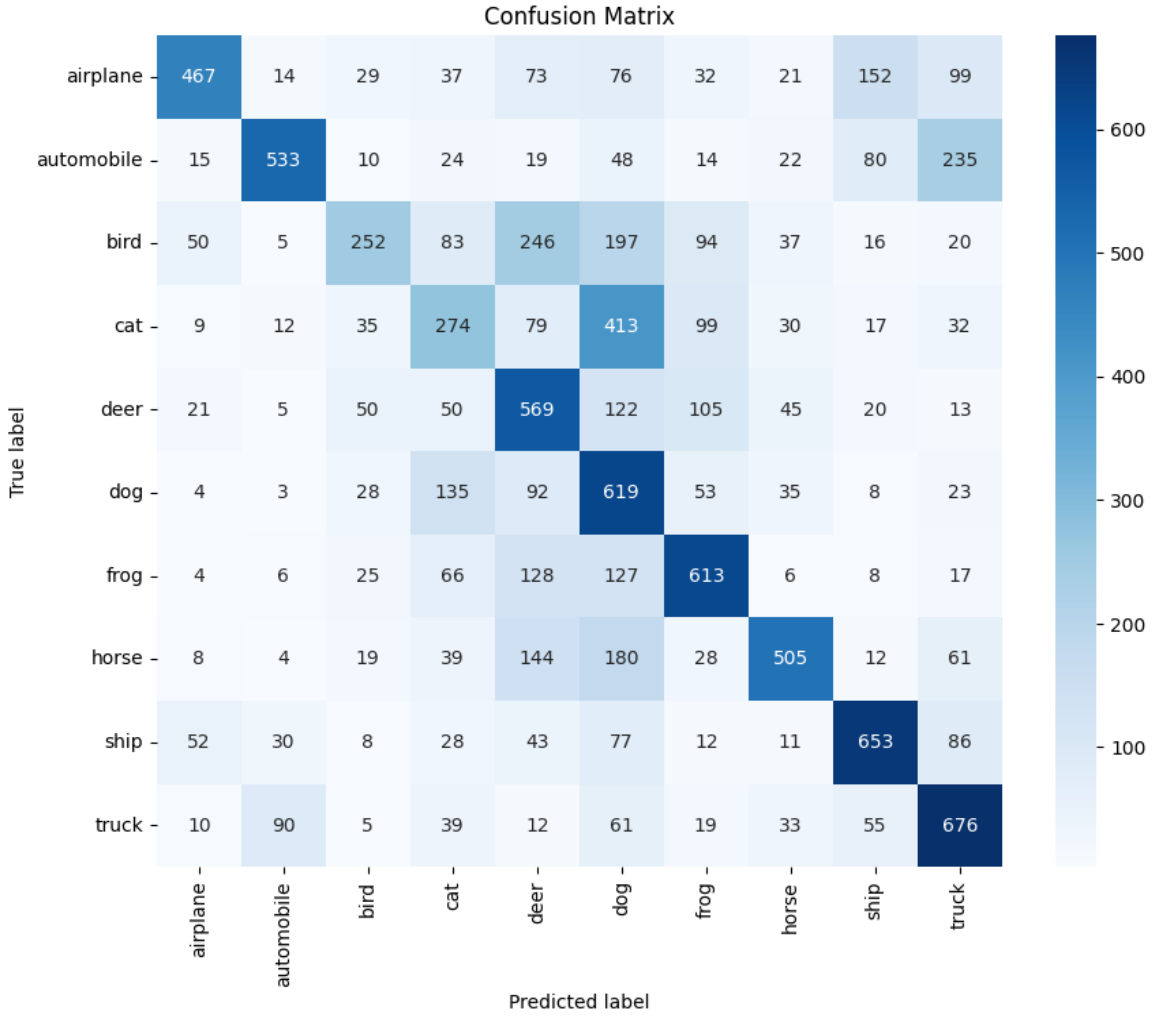

Figure 3: Precision, Recall and F-Score for all Categories ANN

	precision	recall	f1-score	support
airplane	0.73	0.47	0.57	1000
automobile	0.76	0.53	0.63	1000
bird	0.55	0.25	0.34	1000
cat	0.35	0.27	0.31	1000
deer	0.40	0.57	0.47	1000
dog	0.32	0.62	0.42	1000
frog	0.57	0.61	0.59	1000
horse	0.68	0.51	0.58	1000
ship	0.64	0.65	0.65	1000
truck	0.54	0.68	0.60	1000
accuracy			0.52	10000
macro avg	0.55	0.52	0.52	10000
weighted avg	0.55	0.52	0.52	10000

Figure 4: Accuracy and Loss of CNN

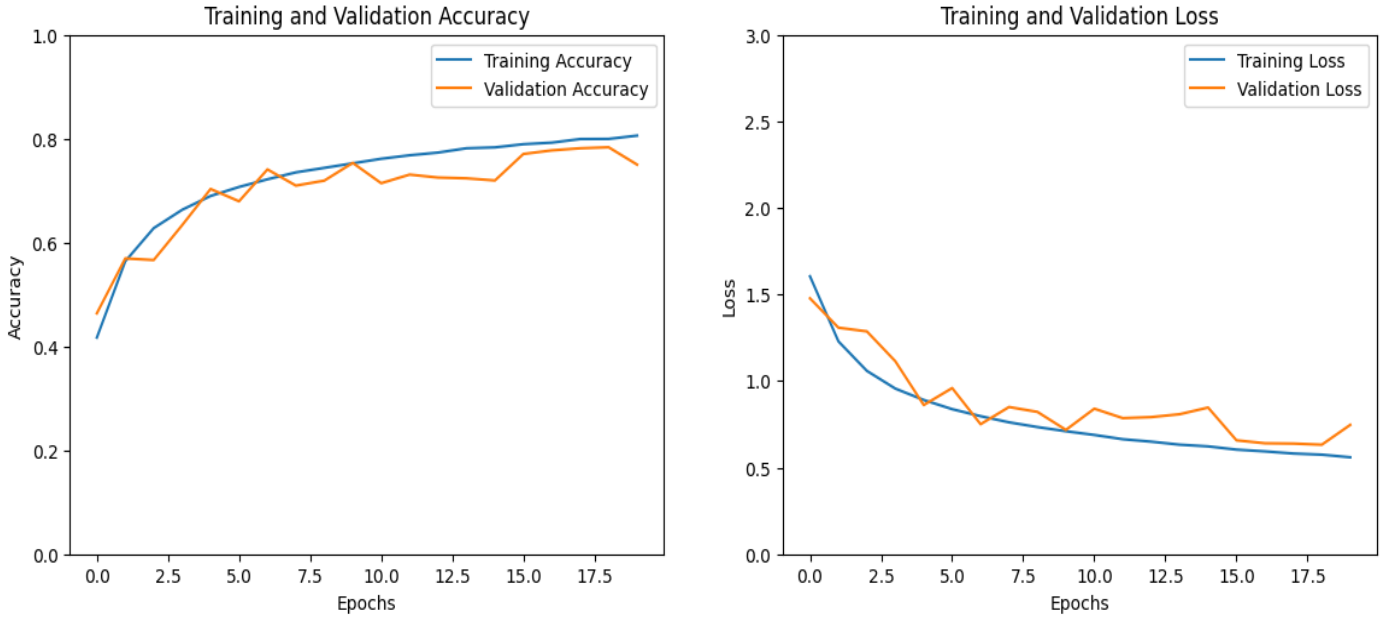

Figure 5: Confusion Matrix CNN

Figure 6: Precision, Recall and F-Score for all Categories CNN

	precision	recall	f1-score	support
airplane	0.69	0.87	0.77	1000
automobile	0.93	0.88	0.90	1000
bird	0.49	0.81	0.61	1000
cat	0.64	0.51	0.57	1000
deer	0.81	0.59	0.69	1000
dog	0.73	0.60	0.66	1000
frog	0.86	0.75	0.80	1000
horse	0.82	0.81	0.81	1000
ship	0.83	0.88	0.85	1000
truck	0.91	0.80	0.85	1000
accuracy			0.75	10000
macro avg	0.77	0.75	0.75	10000
weighted avg	0.77	0.75	0.75	10000

Figure 7

Model accuracy for different lr's on ANN(after 5 epochs for each lr due to computational time) - best 0.0001

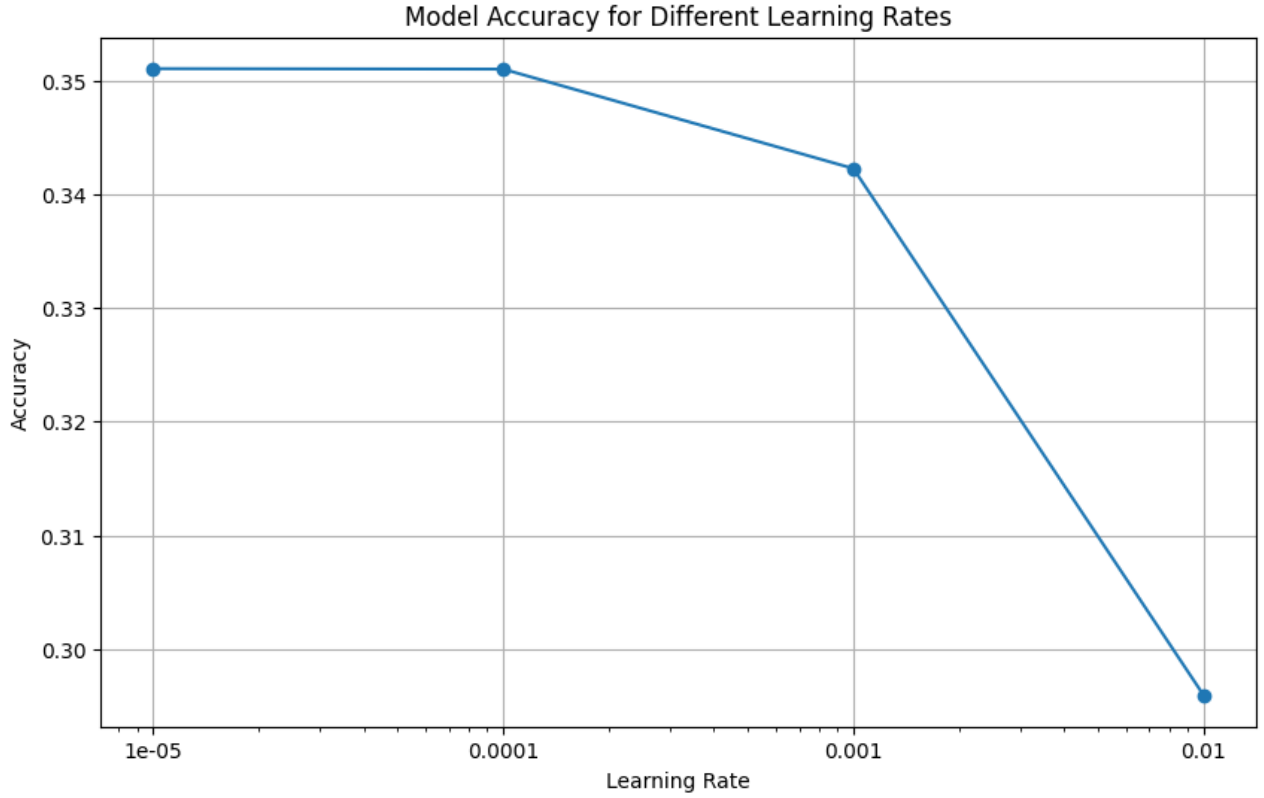

Figure 8

Model accuracy for different lr's on CNN(after 5 epochs for each lr due to computational time) - best 0.0001