Introduction To Machine Learning

Assignment 2: Applying ANN for Image Classification

Mazen Gaber 7467
Clara Samy 7491
Mohammed Emad Eldin 7786

I. INTRODUCTION

In this problem set we used PyTorch to train neural network-based image classifier using CIFAR-10 dataset to evaluate the model.The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, with 6000 images per class. The classes are completely mutually exclusive. Our report includes the following requirements:

1. Comments and documentation on model accuracy, precision, recall, F-score and Confusion matrix for the 3 hidden layer ANN

2. Comments and documentation on model accuracy, precision, recall, F-score and Confusion matrix for the transfer learning of 3 models (VGG16 ,GoogleNet ,Resnet50)

3. Bonus Comments and documentation on model accuracy, precision, recall, F-score and Confusion matrix for another ANN with 2 and 4 hidden layers

4. Comparison on all results of all models

5. Output for some test images for each model

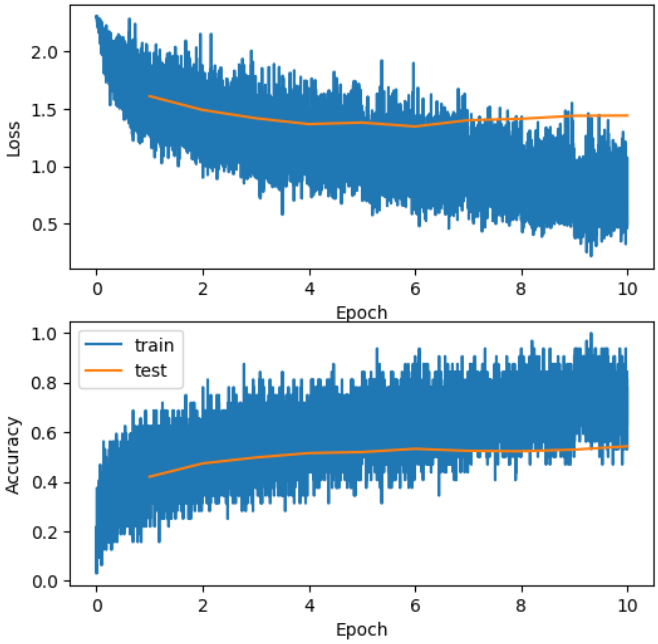
II. MODELS' DESCRIPTION

An Artificial Neural Network (ANN) is a computational model inspired by the human brain's neural structure. It consists of interconnected nodes (neurons) organized into layers. Information flows through these nodes, and the network adjusts the connection strengths (weights) during training to learn from data, enabling it to recognize patterns, make predictions, and solve various tasks in machine learning and artificial intelligence.We worked on building an ANN with 3 hidden layer model with first layer 4096 units , second layer 2048 units and third layer 512 units. We also worked on building another ANN with 2 and 3 hidden layers.
Convolutional Neural Networks (CNN) is a class of deep neural networks, most commonly applied to analyze visual imagery. It is used to accept unstructured and non-numeric data forms such as Image, Text, and Speech. Now when we think of a neural network we think about matrix multiplications but that is not the case with CNN. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

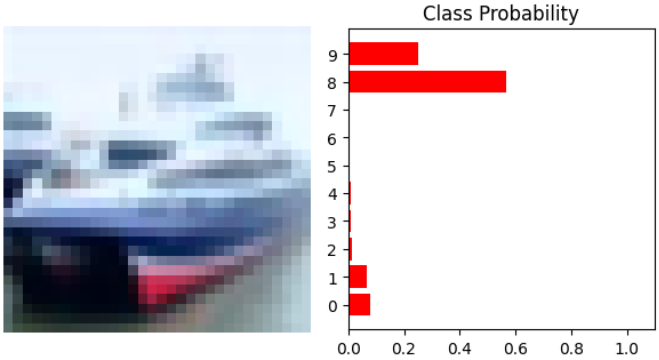
III. REQUIREMENTS RESULTS

A. ANN model: 3 hidden layers

Loss and accuracy graphs:



Class probability:
We choose a random image and analyzed the probability compared to each class and as we can see the highest probability goes to class 8(ship).



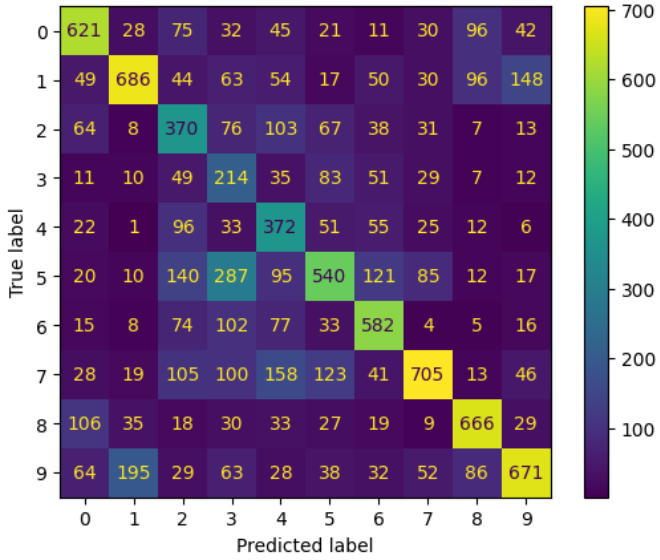
Classification Report:
The accuracy resulted in the report was 54%

```
#Model accuracy
report = classification_report(predicted.view(-1).cpu(),y_test.view(-1))
print(report)
```

[22] ✓ 0.0s

```
              precision    recall  f1-score   support

           0       0.62      0.62      0.62      1001
           1       0.69      0.55      0.61      1237
           2       0.37      0.48      0.42       777
           3       0.21      0.43      0.29       501
           4       0.37      0.55      0.44       673
           5       0.54      0.41      0.46      1327
           6       0.58      0.64      0.61       916
           7       0.70      0.53      0.60      1338
           8       0.67      0.69      0.68       972
           9       0.67      0.53      0.59      1258

    accuracy                           0.54     10000
   macro avg       0.54      0.54      0.53     10000
weighted avg       0.58      0.54      0.55     10000
```

Confusion Matrix:



Predicted Samples:
We choose a random batch of images from test data and reviewed the prediction of each image.The red labels were wrong predictions and the blue labels were right predictions.

B. CNN: GoogleNet

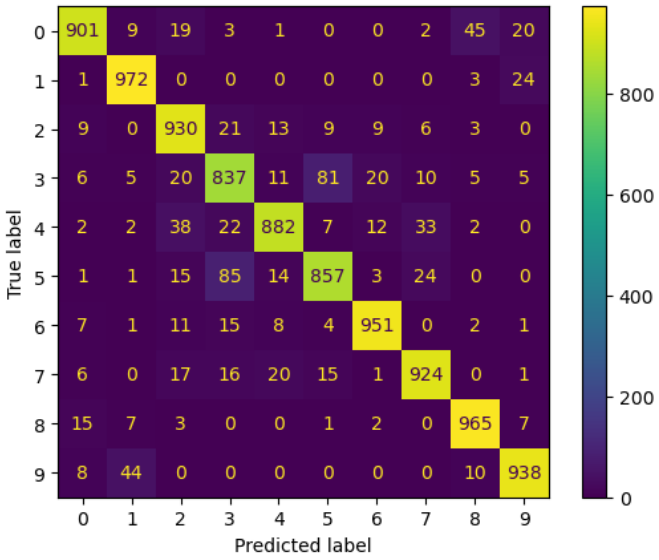
Classification Report:
The accuracy resulted in the report was 92%

```
#Model accuracy
report = classification_report(y_true,y_pred)
print(report)
```

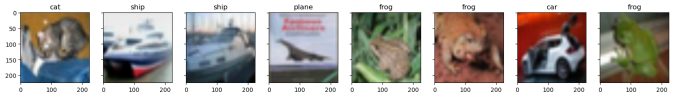
[31]

```
              precision    recall  f1-score   support

           0       0.94      0.90      0.92      1000
           1       0.93      0.97      0.95      1000
           2       0.88      0.93      0.91      1000
           3       0.84      0.84      0.84      1000
           4       0.93      0.88      0.91      1000
           5       0.88      0.86      0.87      1000
           6       0.95      0.95      0.95      1000
           7       0.92      0.92      0.92      1000
           8       0.93      0.96      0.95      1000
           9       0.94      0.94      0.94      1000

    accuracy                           0.92     10000
   macro avg       0.92      0.92      0.92     10000
weighted avg       0.92      0.92      0.92     10000
```

Confusion Matrix:

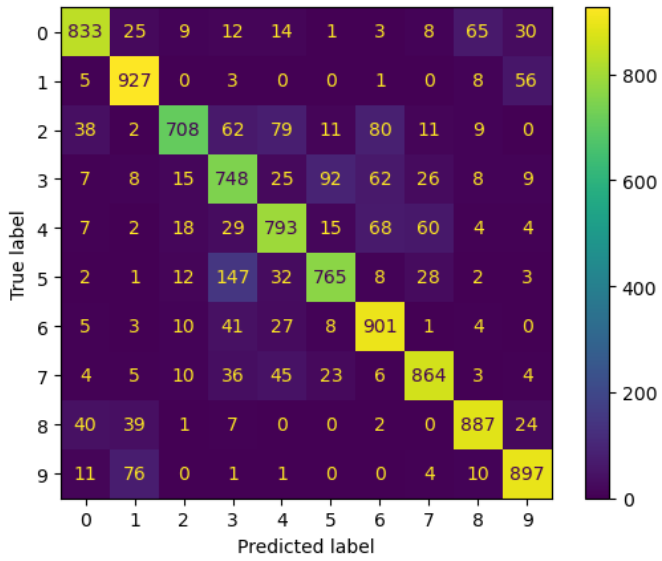


Samples:

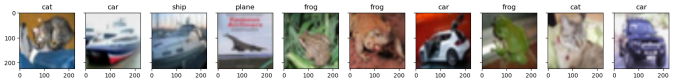


Consequently, GoogleNet provided the highest accuracy among all models in the project with relatively moderate time performance (nearly 70 mins)

Confusion Matrix:



Samples:



The VGG16 pretrained model gives the significantly higher accuracy comparing with other ANN Models with time elapsed nearly = 77 mins

C. CNN: VGG16

Classification Report:

The accuracy resulted in the report was 83%

Classification Report

```
report = classification_report(y_true,y_pred)
print(report)
```

[61]

	precision	recall	f1-score	support
0	0.88	0.83	0.85	1000
1	0.85	0.93	0.89	1000
2	0.90	0.71	0.79	1000
3	0.69	0.75	0.72	1000
4	0.78	0.79	0.79	1000
5	0.84	0.77	0.80	1000
6	0.80	0.90	0.85	1000
7	0.86	0.86	0.86	1000
8	0.89	0.89	0.89	1000
9	0.87	0.90	0.89	1000
accuracy			0.83	10000
macro avg	0.84	0.83	0.83	10000
weighted avg	0.84	0.83	0.83	10000

D. CNN: ResNet50

Classification Report:

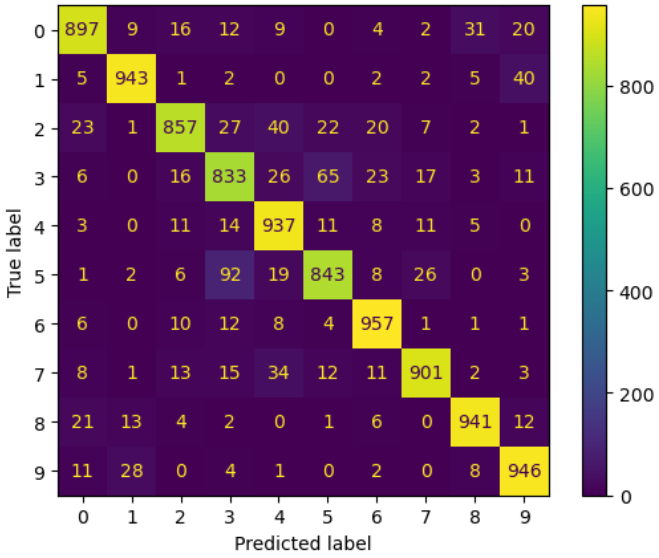
The accuracy resulted in the report was 91%

```
#Model accuracy
report = classification_report(y_true,y_pred)
print(report)
```

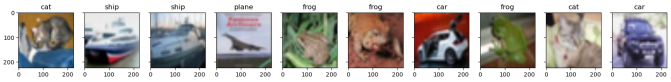
[100]

	precision	recall	f1-score	support
0	0.91	0.90	0.91	1000
1	0.95	0.94	0.94	1000
2	0.92	0.86	0.89	1000
3	0.82	0.83	0.83	1000
4	0.87	0.94	0.90	1000
5	0.88	0.84	0.86	1000
6	0.92	0.96	0.94	1000
7	0.93	0.90	0.92	1000
8	0.94	0.94	0.94	1000
9	0.91	0.95	0.93	1000
accuracy			0.91	10000
macro avg	0.91	0.91	0.91	10000
weighted avg	0.91	0.91	0.91	10000

Confusion Matrix:



Samples:

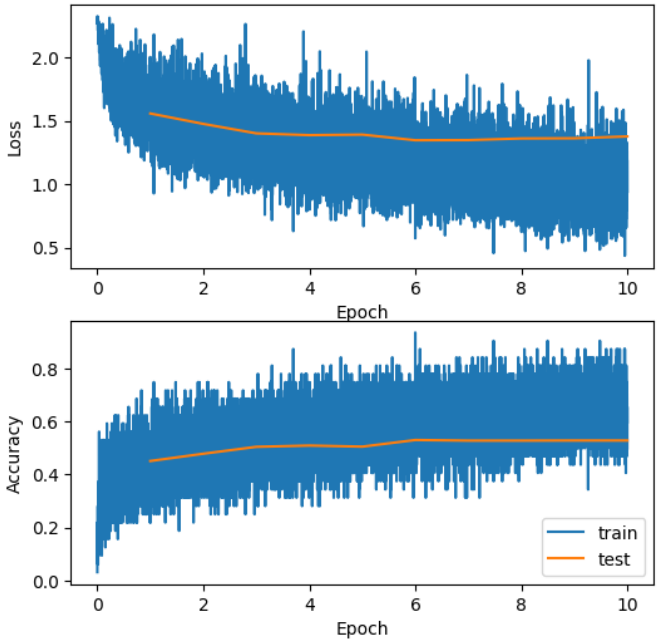


The RESNET50 Model provided an accuracy of 90.55% with time (67 mins)

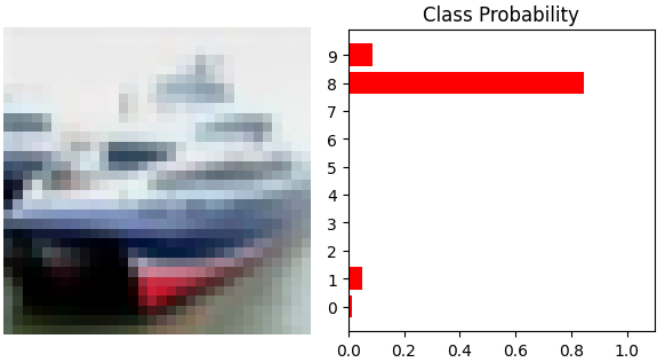
IV. BONUS RESULTS

A. *ANN model: 2 hidden layers*

Loss and accuracy graphs:



Class probability:
We choose a random image and analyzed the probability compared to each class and as we can see the highest probability goes to class 8(ship).



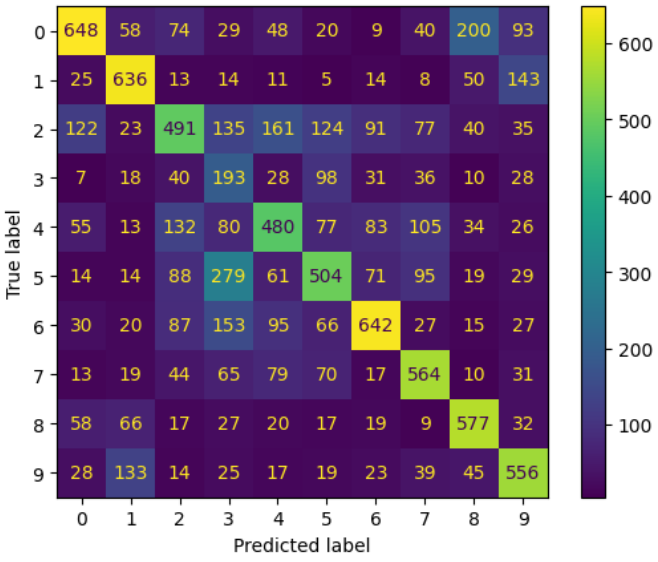
Classification Report:
The accuracy resulted in the report was 53%

```
#Model accuracy
report = classification_report(predicted.view(-1).cpu(),y_test.view(-1))
print(report)
```

[64] ✓ 0.0s

	precision	recall	f1-score	support
0	0.65	0.53	0.58	1219
1	0.64	0.69	0.66	919
2	0.49	0.38	0.43	1299
3	0.19	0.39	0.26	489
4	0.48	0.44	0.46	1085
5	0.50	0.43	0.46	1174
6	0.64	0.55	0.59	1162
7	0.56	0.62	0.59	912
8	0.58	0.69	0.63	842
9	0.56	0.62	0.59	899
accuracy			0.53	10000
macro avg	0.53	0.53	0.53	10000
weighted avg	0.55	0.53	0.53	10000

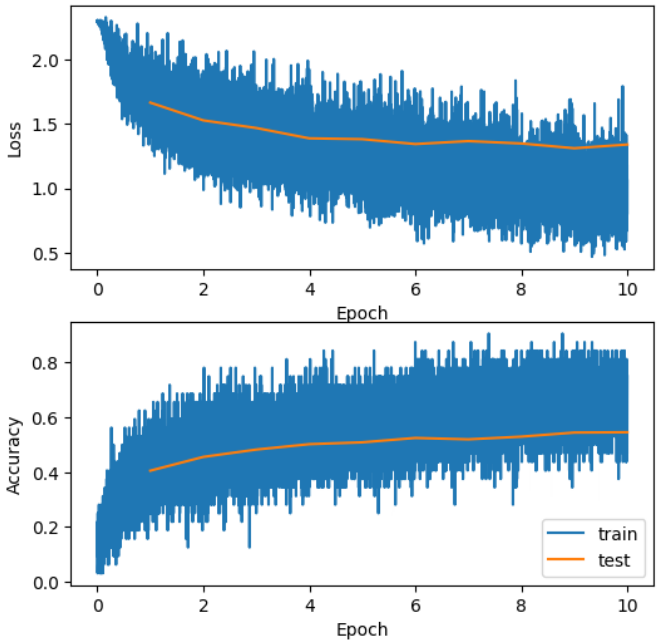
Confusion Matrix:



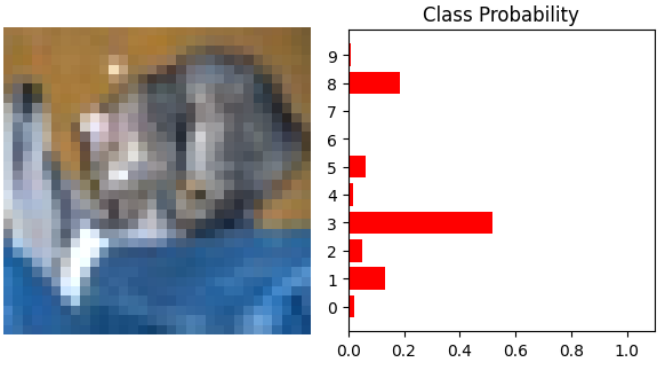
Predicted Samples:
We choose a random batch of images from test data and reviewed the prediction of each image.The red labels were wrong predictions and the blue labels were right predictions.

B. ANN model: 4 hidden layers

Loss and accuracy graphs:



Class probability:
We choose a random image and analyzed the probability compared to each class and as we can see the highest probability goes to class 3(cat).



Classification Report:
The accuracy resulted in the report was 55%

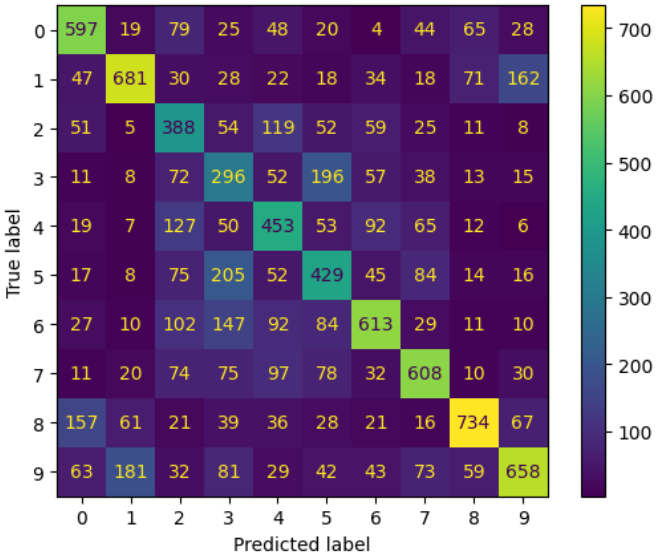
```
#Model accuracy
report = classification_report(predicted.view(-1).cpu(),y_test.view(-1))
print(report)
```

[82] ✓ 0.0s

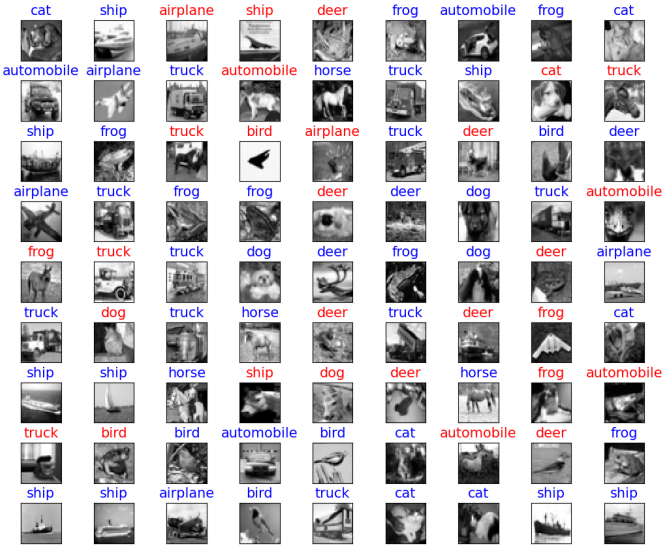
	precision	recall	f1-score	support
0	0.60	0.64	0.62	929
1	0.68	0.61	0.65	1111
2	0.39	0.50	0.44	772
3	0.30	0.39	0.34	758
4	0.45	0.51	0.48	884
5	0.43	0.45	0.44	945
6	0.61	0.54	0.58	1125
7	0.61	0.59	0.60	1035
8	0.73	0.62	0.67	1180
9	0.66	0.52	0.58	1261
accuracy			0.55	10000
macro avg	0.55	0.54	0.54	10000
weighted avg	0.57	0.55	0.55	10000

+ Code + Markdown

Confusion Matrix:



Predicted Samples:
We choose a random batch of images from test data and reviewed the prediction of each image.The red labels were wrong predictions and the blue labels were right predictions.



The ANN model with 4 hidden layers granted the highest accuracy (slightly higher than the model with 3 hidden layers) with overall 5 10 mins per model

V. OBSERVATIONS AND COMMENTS

A. THROUGH ANN MODELS:

It was observed that the model with 4 hidden layers recorded the highest (slightly) accuracy among the other models. But the model did not always provide the best accuracy, through different runs, sometimes the model with 3-hidden layers provided a higher accuracy. Moreover, it was noticed that the model with 2-hidden layers was somehow overfitting. But apparently, the models took less time than the CNN models.

B. CNN PRETRAINED MODELS:

By training different models (VGG16, GoogleNet, ResNet50) It was clearly discovered that they take much longer time than the previously-mentioned ANN models but with much higher accuracy, for e.g., GoogleNet recorded 91.57% accuracy while ANN-model with 4-hidden layers recorded 54.57%

C. Which is better for image classification ANN or CNN models?

With ANN, concrete data points must be provided. For example, in a model where we are trying to distinguish between dogs and cats, the width of the noses and length of the ears must be explicitly provided as data points.
When using CNN, these spatial features are extracted from image input. This makes CNN ideal when thousands of features need to be extracted. Instead of having to measure each individual feature, CNN gathers these features on its own.
Using ANN, image classification problems become difficult because 2-dimensional images need to be converted to 1-dimensional vectors. This increases the number of trainable parameters exponentially. Increasing trainable parameters takes storage and processing capability.
In other words, it would be expensive. Compared to its predecessors, the main advantage of CNN is that it automatically detects the important features without any human supervision. This is why CNN would be an ideal solution to computer vision and image classification problems.
Reference: https://viso.ai/deep-learning/ann-and-cnn-analyzing-differences-and-similarities/

VI. CONCLUSION

CNN models are more accurate and efficient when dealing with images as they have higher ability to detect features while ANN is better when dealing with data classification, in other words, CNN can follow a data augmentation method to provide same accuracy as ANN. But overall, CNN is a better fit for Cifar-10 dataset as it deals with images as there are multitude features to be extracted.

VII. COMPARISONS

Table I compares accuracy of all models.

Model	ANN			CNN (pretrained)		
	2 Hidden layers	3 Hidden layers	4 Hidden layers	VGG16	GoogleNet	ResNet50
Accuracy	52.91%	54.27%	54.57%	83.23%	91.57%	90.55%

TABLE I
ACCURACY TABLE

Table II compares precision of all models.

Model	ANN			CNN (pretrained)			Classes
	2 Hidden layers	3 Hidden layers	4 Hidden layers	VGG16	GoogleNet	ResNet50	
Precision	0.65	0.62	0.60	0.88	0.94	0.91	Airplane
	0.64	0.69	0.68	0.85	0.93	0.95	Automobile
	0.49	0.37	0.39	0.90	0.88	0.92	Bird
	0.19	0.21	0.30	0.69	0.84	0.82	Cat
	0.48	0.37	0.45	0.78	0.93	0.87	Deer
	0.50	0.54	0.43	0.84	0.88	0.88	Dog
	0.64	0.58	0.61	0.80	0.95	0.92	frog
	0.56	0.70	0.61	0.86	0.92	0.93	Horse
	0.58	0.67	0.73	0.89	0.93	0.94	Ship
	0.56	0.67	0.66	0.87	0.94	0.91	Truck
Average	0.53	0.54	0.55	0.84	0.91	0.91	

TABLE II
PRECISION TABLE

Table III compares recall of all models.

Model	ANN			CNN (pretrained)			Classes
	2 Hidden layers	3 Hidden layers	4 Hidden layers	VGG16	GoogleNet	ResNet50	
Recall	0.53	0.62	0.64	0.83	0.90	0.90	Airplane
	0.69	0.55	0.61	0.93	0.97	0.94	Automobile
	0.38	0.48	0.50	0.71	0.93	0.86	Bird
	0.39	0.43	0.39	0.75	0.84	0.83	Cat
	0.44	0.55	0.51	0.79	0.88	0.94	Deer
	0.43	0.41	0.45	0.77	0.86	0.84	Dog
	0.55	0.64	0.54	0.90	0.95	0.96	frog
	0.62	0.53	0.59	0.86	0.92	0.90	Horse
	0.69	0.69	0.62	0.89	0.96	0.94	Ship
	0.62	0.53	0.52	0.90	0.94	0.95	Truck
Average	0.53	0.54	0.54	0.83	0.92	0.91	

TABLE III

RECALL TABLE

Table IV compares F1-score of all models.

Model	ANN			CNN (pretrained)			Classes
	2 Hidden layers	3 Hidden layers	4 Hidden layers	VGG16	GoogleNet	ResNet50	
F1-score	0.58	0.62	0.62	0.85	0.92	0.91	Airplane
	0.66	0.61	0.65	0.89	0.95	0.94	Automobile
	0.43	0.42	0.44	0.79	0.91	0.89	Bird
	0.26	0.29	0.34	0.72	0.84	0.83	Cat
	0.46	0.44	0.48	0.79	0.91	0.90	Deer
	0.46	0.46	0.44	0.80	0.87	0.86	Dog
	0.59	0.61	0.58	0.85	0.95	0.94	frog
	0.59	0.60	0.60	0.86	0.92	0.92	Horse
	0.63	0.68	0.67	0.89	0.95	0.94	Ship
	0.59	0.59	0.58	0.89	0.94	0.93	Truck
Average	0.53	0.53	0.54	0.83	0.92	0.91	

TABLE IV

F1-SCORE TABLE