Assignment Report #2

Student name: *Carin Samy/7481 —— Hassan Tamer/ 7405*

Due date: *August 24, 2023*

ANN(1):. 3 hidden layers ANN

1)Pre Processing:. In pre processing after several attempts 2 techniques were chosen for this network

a)Data augmentation. Random rotation by 10 degrees and random horizontal flips were the 2 data augmentation techniques used

b)Normalization .

2) Architecture: . This NN had 3 hidden layers as follows
Layer 1 (input size , 4096) and the input size corresponds to flattening the input (32*32*3) = 3072
Layer 2 (4096 , 2048)
Layer 3 (2058 , 512)
Layer 4 (512 , 10)

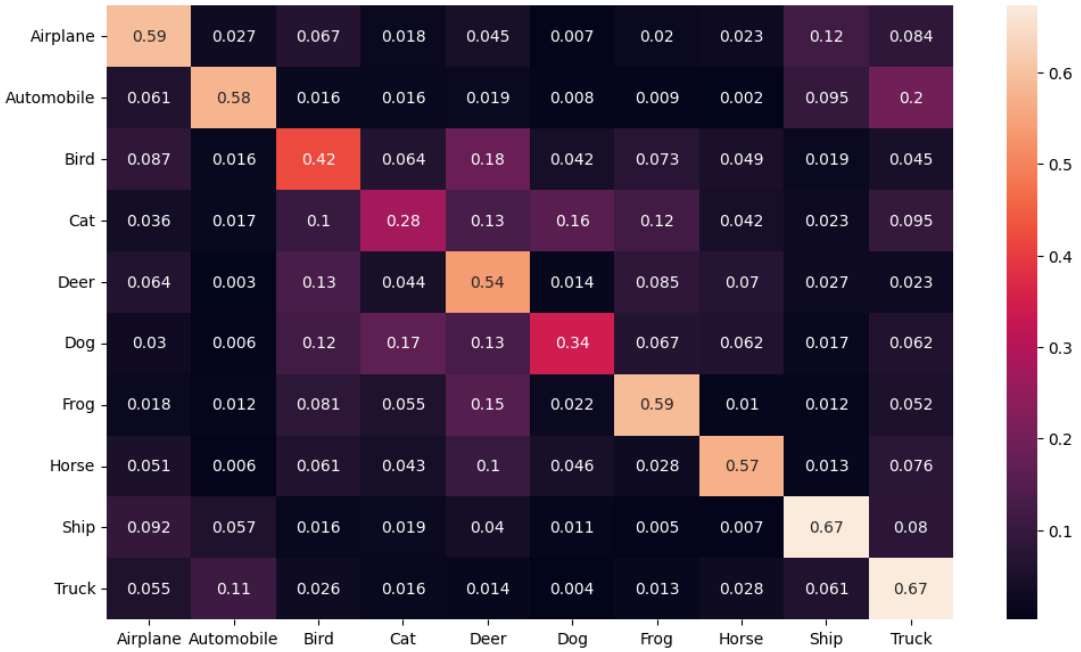
3) Hyperparameter:
. Epochs : 25
Batch size = 128
Learning rate = 0.001
Loss : cross entropy
Optimizer: Adam
Activations : leaky ReLU

4) Accuracy:. Training acc → Got 37406 from 50000 with accuracy 74.81

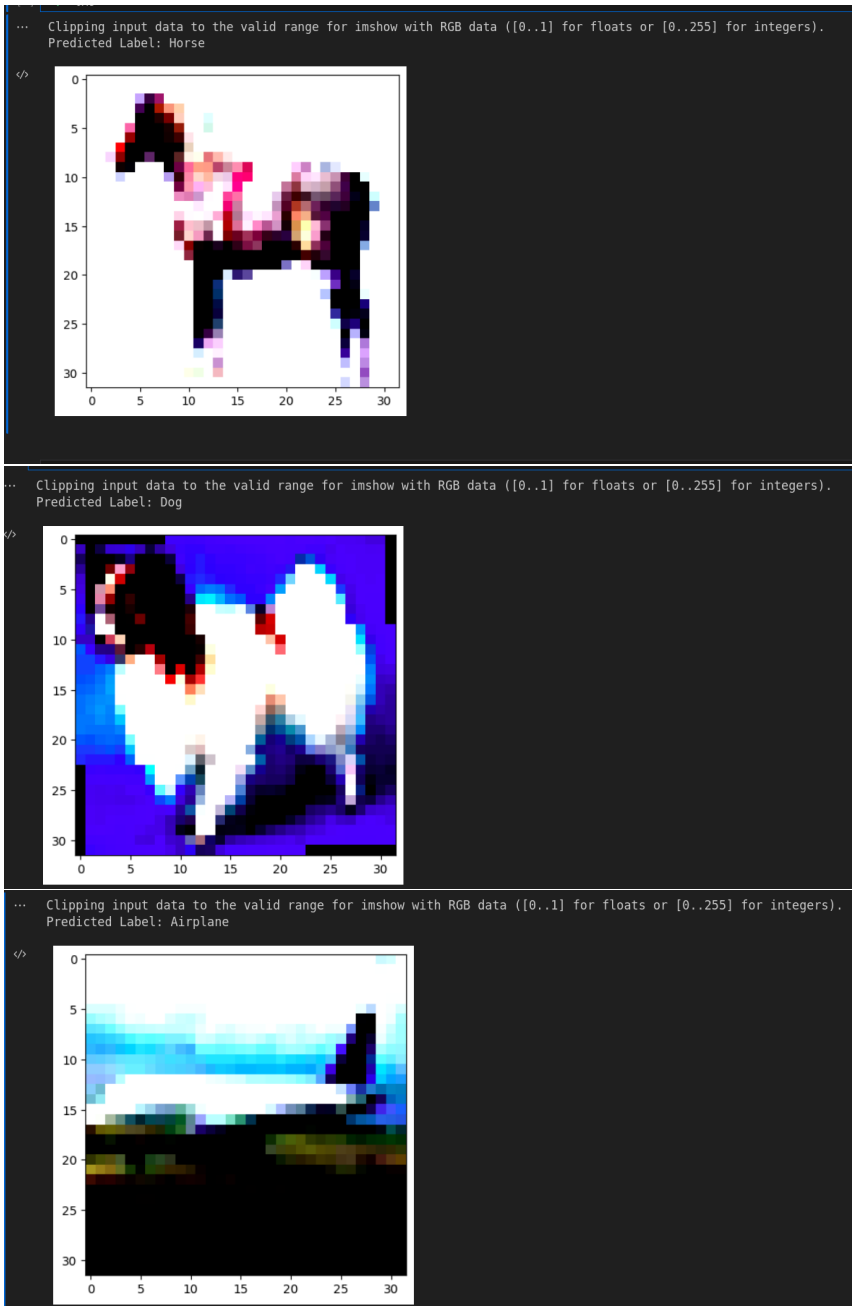
ANN (1) :Outputs 1-Accuracy

```
Accuracy = 52.49%
Classification Report:
              precision    recall  f1-score   support

    Airplane       0.55      0.59      0.57      1000
  Automobile       0.69      0.58      0.63      1000
        Bird       0.40      0.42      0.41      1000
         Cat       0.38      0.28      0.32      1000
        Deer       0.40      0.54      0.46      1000
         Dog       0.52      0.34      0.41      1000
        Frog       0.58      0.59      0.59      1000
       Horse       0.66      0.57      0.61      1000
        Ship       0.64      0.67      0.65      1000
       Truck       0.48      0.67      0.56      1000

    accuracy                           0.52     10000
   macro avg       0.53      0.52      0.52     10000
weighted avg       0.53      0.52      0.52     10000
```

ANN (1) :Outputs 2-Confusion matrix



ANN (1) :Outputs 3- Output for some test images for each model:



Transfer Learning

Transfer learning is the reuse of a pre-trained model on a new problem. It's currently very popular in deep learning because it can train deep neural networks with comparatively little data. This is very useful in the data science field since most real-world problems typically do not have millions of labeled data points to train such complex models.

Using GoogleNet architecture.

1)Pre Processing. Googlenet is designed to input 224x224 pictures so this resize transformation was done along with all the previous transformations

2)Architecture. All layers were frozen except 'inception5b' layer at the end Along with the unfrozen layers 2 layers were added

0.1. Layer 1 (1024,512).

0.2. Layer2 (512,10)
. And a ReLU activation function

3)HyperParameters
. Epochs = 20
Batch size = 256
Learning rate = 0.001
loss : cross entropy
Optimizer : Adam

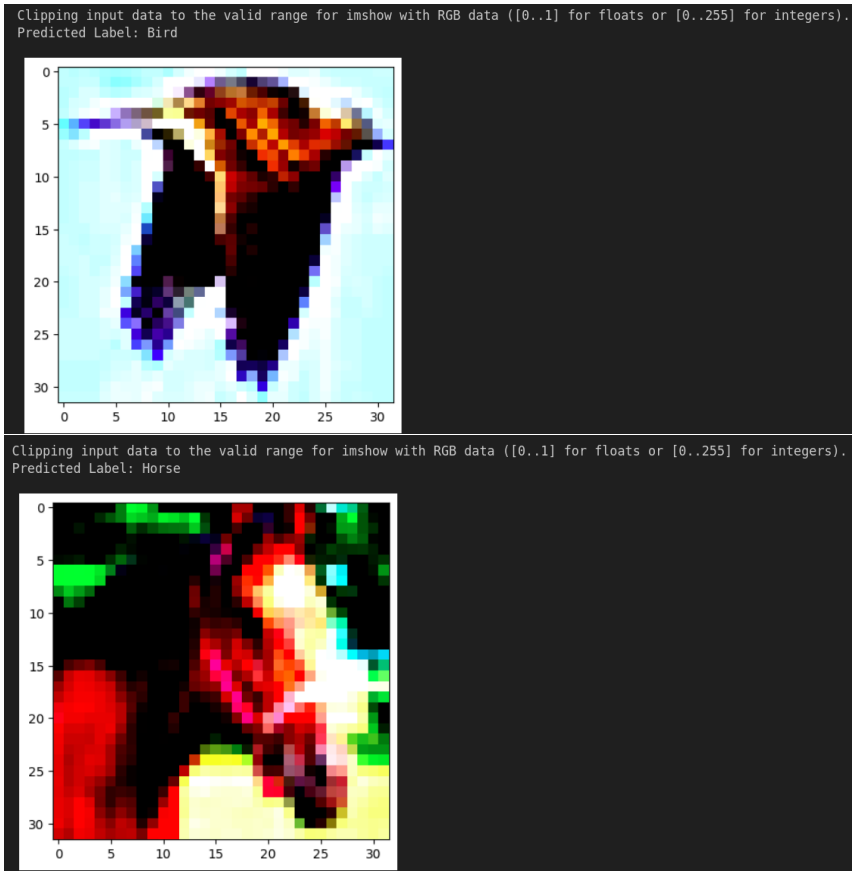
4) Accuracy. Training acc → Got 46909 from 50000 with accuracy 93.82

GoogleNet(1) :Outputs 1-Accuracy

```
Accuracy = 88.84%
Classification Report:
              precision    recall  f1-score   support

    Airplane       0.93      0.84      0.88      1000
  Automobile       0.93      0.97      0.95      1000
        Bird       0.87      0.89      0.88      1000
         Cat       0.78      0.83      0.80      1000
        Deer       0.84      0.89      0.86      1000
         Dog       0.86      0.82      0.84      1000
        Frog       0.91      0.89      0.90      1000
       Horse       0.91      0.90      0.91      1000
        Ship       0.95      0.91      0.93      1000
       Truck       0.91      0.94      0.92      1000

    accuracy                           0.89     10000
   macro avg       0.89      0.89      0.89     10000
weighted avg       0.89      0.89      0.89     10000
```

GoogleNet(1) :Outputs 2-Confusion matrix

GoogleNet(1) :Outputs 3-Output for some test images for each model.



Using ResNet50 architecture.

1)Pre Processing. Along with the previous pre processings multiple data augmentation techniques were added

a)Random crop
b)Random brightness changes

2)Architecture. All layers were frozen except 'layer4' layer at the end Along with the unfrozen layers 1 layer was added

Layer 1 (2048,10).

3)HyperParameters
. Epochs = 20
Batch size = 128
Learning rate = 0.001
loss : cross entropy
Optimizer : Adam

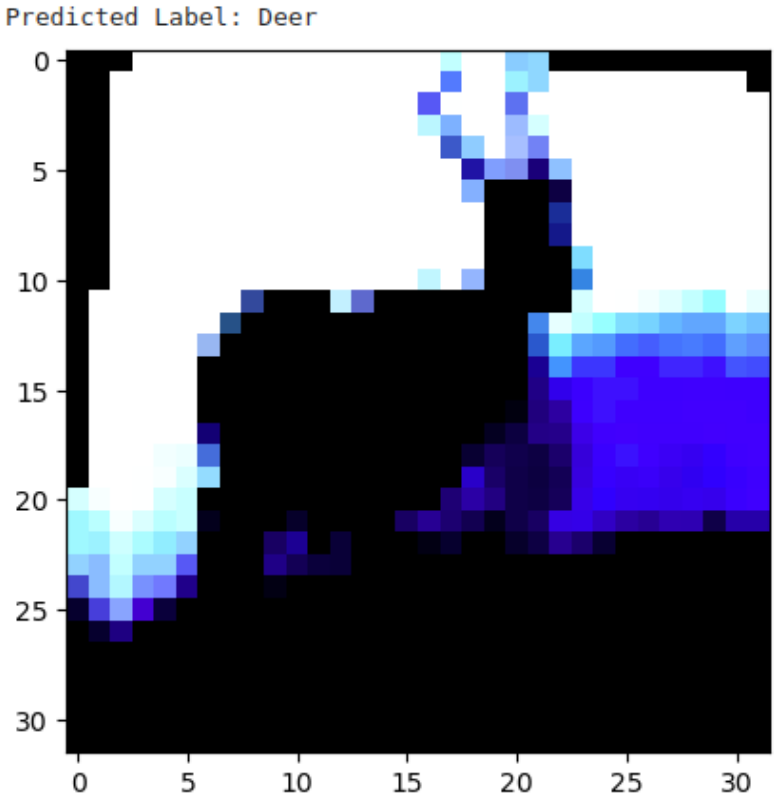
4) Accuracy. Training acc → Got 49369 from 50000 with accuracy 98.74

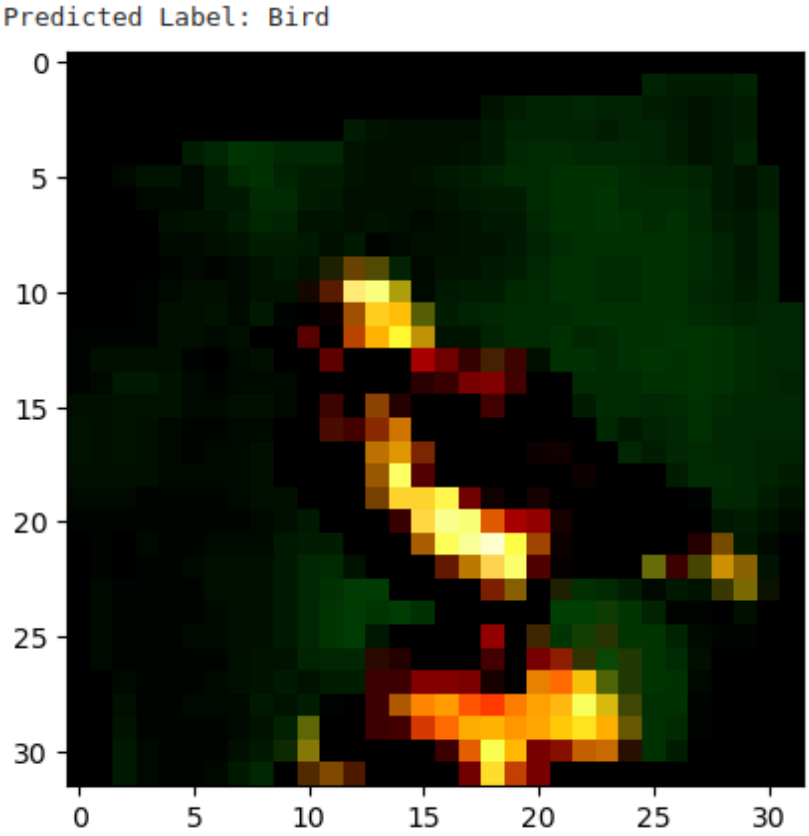
ResNet50(2) :Outputs 1-Accuracy

```
Accuracy = 92.15%
Classification Report:
              precision    recall  f1-score   support

    Airplane       0.91      0.95      0.93      1000
  Automobile       0.97      0.95      0.96      1000
        Bird       0.91      0.89      0.90      1000
         Cat       0.83      0.87      0.85      1000
        Deer       0.91      0.92      0.91      1000
         Dog       0.90      0.85      0.88      1000
        Frog       0.97      0.94      0.95      1000
       Horse       0.92      0.95      0.94      1000
        Ship       0.96      0.95      0.95      1000
       Truck       0.95      0.94      0.95      1000

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

ResNet50(2) :Outputs 2-Confusion matrix

ResNet50(2) :Outputs 3-Output for some test images for each model.





Using VGG16 architecture.

1)Pre Processing. Previous pre processing were used

2)Architecture. All layers were frozen except 'avgpool' and 'classifier' layers at the end Along with the unfrozen layers 5 layer was added

Layer 1 (4096,2048).

Layer 2 (2048,1024).

Layer 3 (1024,512).

Layer 4 (512,256).

Layer 5 (256,10).

3)HyperParameters
. Epochs = 7
Batch size = 128
Learning rate = 0.001
loss : cross entropy
Optimizer : Adam

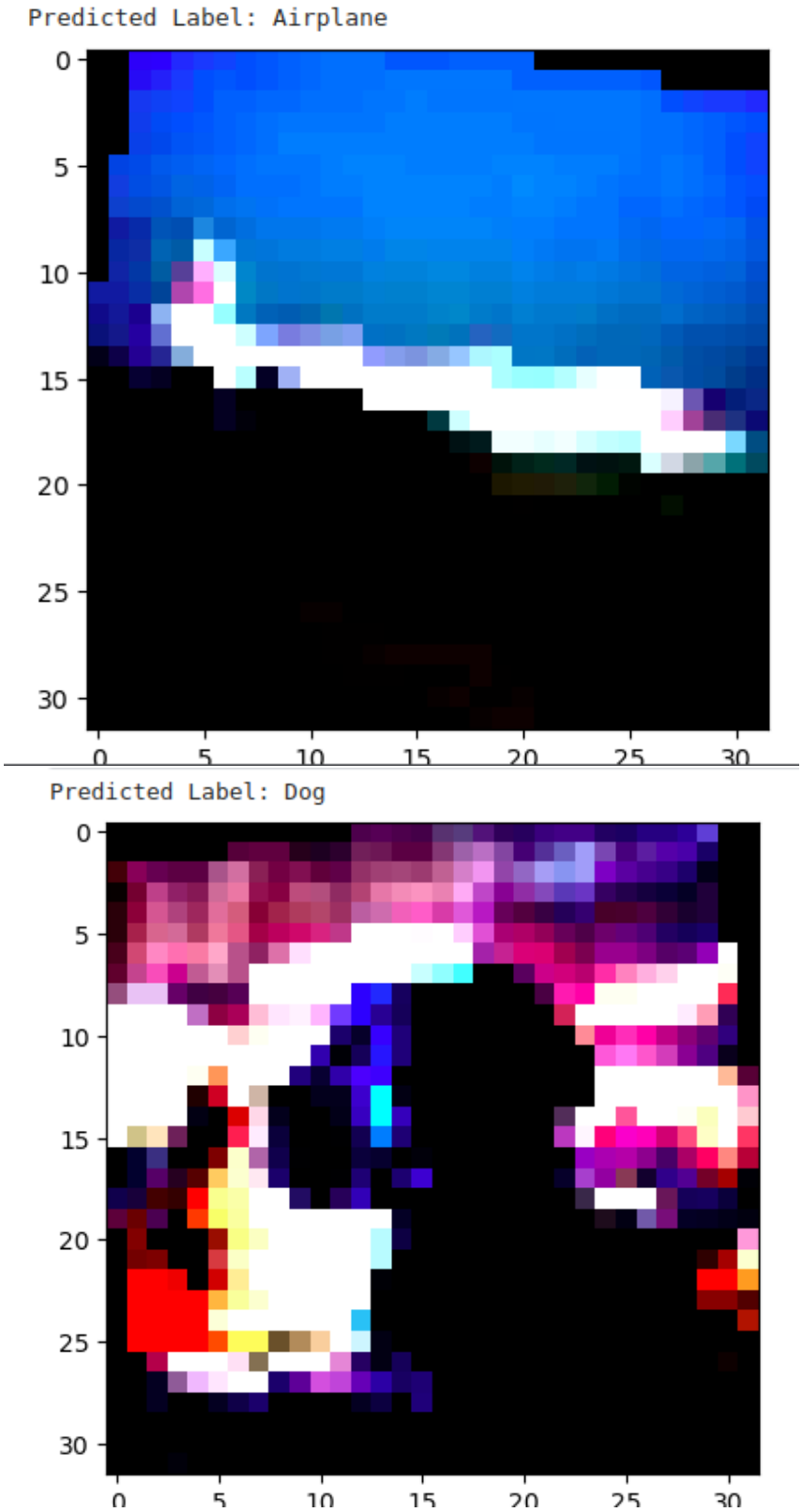
4) Accuracy. Training acc Got 44670 from 50000 with accuracy 89.34

VGG16(3) :Outputs 1-Accuracy

```
Accuracy = 87.36%
Classification Report:
              precision    recall  f1-score   support

    Airplane       0.84      0.91      0.87      1000
  Automobile       0.91      0.95      0.93      1000
        Bird       0.85      0.86      0.85      1000
         Cat       0.74      0.79      0.76      1000
        Deer       0.81      0.91      0.86      1000
         Dog       0.85      0.80      0.82      1000
        Frog       0.94      0.88      0.91      1000
       Horse       0.93      0.88      0.90      1000
        Ship       0.96      0.87      0.91      1000
       Truck       0.95      0.89      0.92      1000

    accuracy                           0.87     10000
   macro avg       0.88      0.87      0.87     10000
weighted avg       0.88      0.87      0.87     10000
```

VGG16(3) :Outputs 2-Confusion matrix

VGG16(3):Outputs 3-Output for some test images for each model.



1. **[BONUS PART]**

ANN(2):. 2 hidden layers ANN

1)Pre Processing:. In pre processing after several attempts 2 techniques were chosen for this network

a)Data augmentation. Random rotation by 10 degrees and random horizontal flips were the 2 data augmentation techniques used

b)Normalization .

2) Architecture: . This NN had 3 hidden layers as follows

Layer 1 (input size , 2048). ,and the input size corresponds to flattening the input (32*32*3) = 3072

Layer 2 (2048 , 1024).

Layer 3 (1024 , 10).

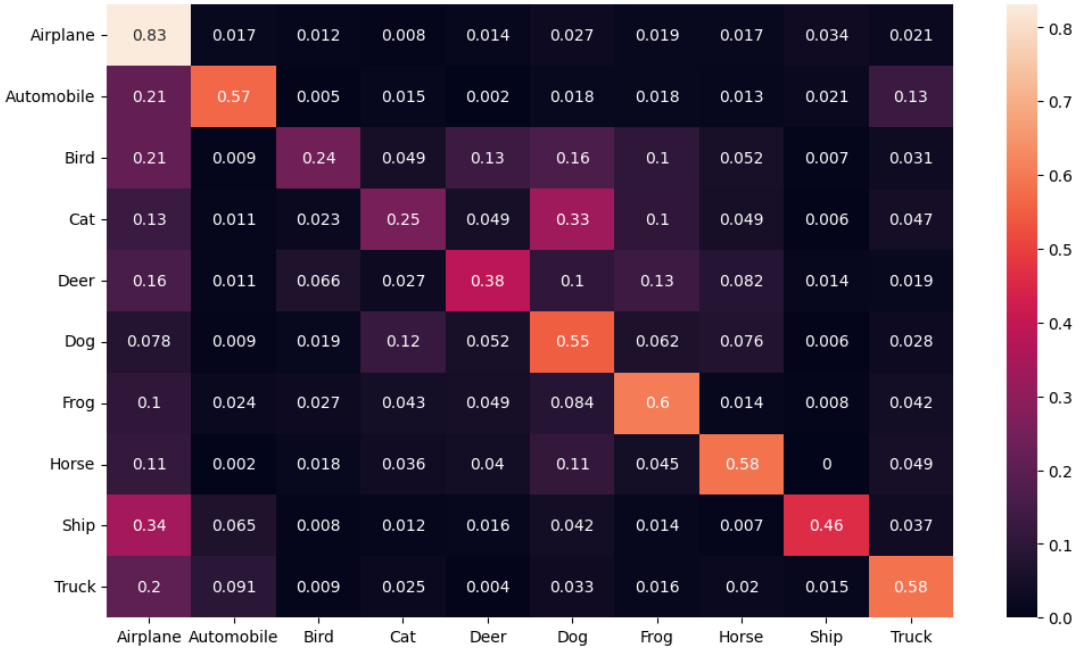
3) Hyperparameter:
. Epochs : 25
Batch size = 128
Learning rate = 0.001
Loss : cross entropy
Optimizer: Adam
Activations : leaky ReLU

4) Accuracy:. Training acc → Got 35096 from 50000 with accuracy 70.19

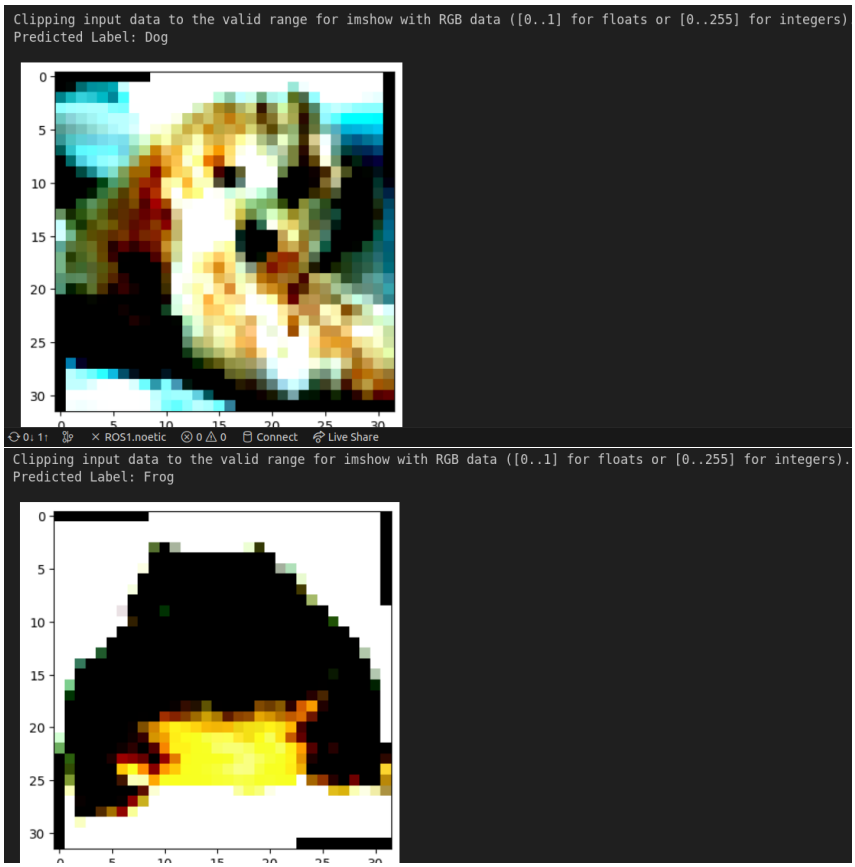
ANN (2) :Outputs 1-Accuracy

```
Accuracy = 50.64%
Classification Report:
              precision    recall  f1-score   support

    Airplane       0.35      0.83      0.49      1000
  Automobile       0.70      0.57      0.63      1000
        Bird       0.57      0.24      0.34      1000
         Cat       0.43      0.25      0.32      1000
        Deer       0.52      0.38      0.44      1000
         Dog       0.38      0.55      0.45      1000
        Frog       0.54      0.60      0.57      1000
       Horse       0.64      0.58      0.61      1000
        Ship       0.81      0.46      0.59      1000
       Truck       0.59      0.58      0.59      1000

    accuracy                           0.51     10000
   macro avg       0.55      0.51      0.50     10000
weighted avg       0.55      0.51      0.50     10000
```

ANN (2) :Outputs 2-Confusion matrix



ANN (2) :Outputs 3- Output for some test images for each model:



ANN(4):. 4 hidden layers ANN

1)Pre Processing:. In pre processing after several attempts 2 techniques were chosen for this network

a)Data augmentation. Random rotation by 10 degrees and random horizontal flips were the 2 data augmentation techniques used

b)Normalization .

2) Architecture: . This NN had 3 hidden layers as follows

Layer 1 (input size , 2048). and the input size corresponds to flattening the input (32*32*3) = 3072

Layer 2 (2048 , 512).

Layer 3 (512 , 256).

Layer 4 (256 , 10).

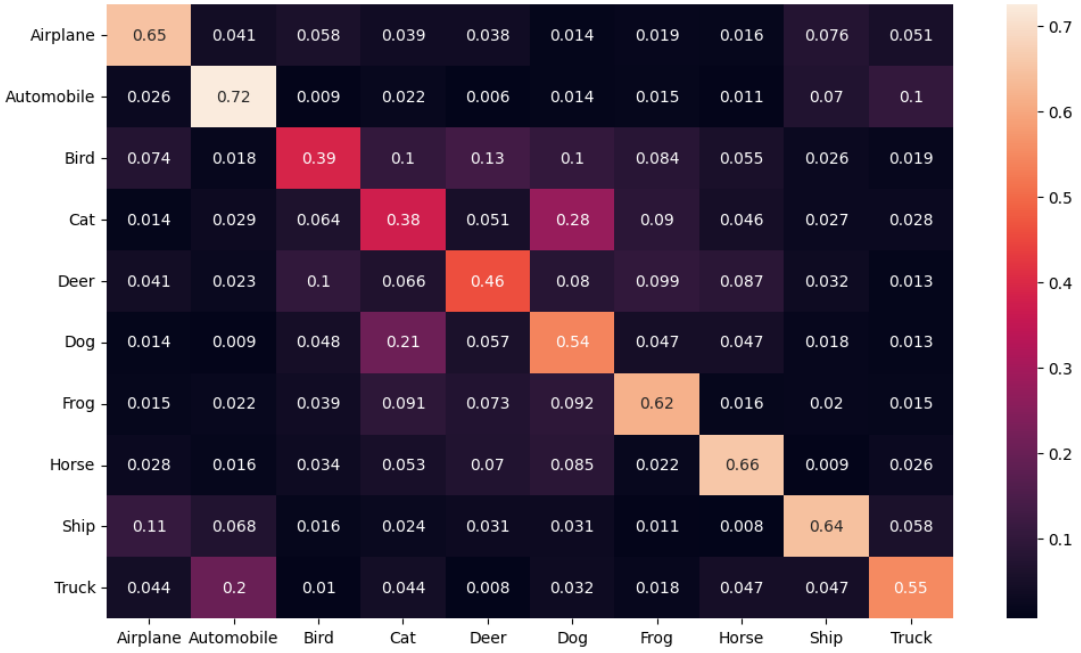
3) Hyperparameter:
. Epochs : 30
Batch size = 128
Learning rate = 0.001
Loss : cross entropy
Optimizer: Adam
Activations : leaky ReLU

4) Accuracy:. Training acc → Got 40137 from 50000 with accuracy 80.27

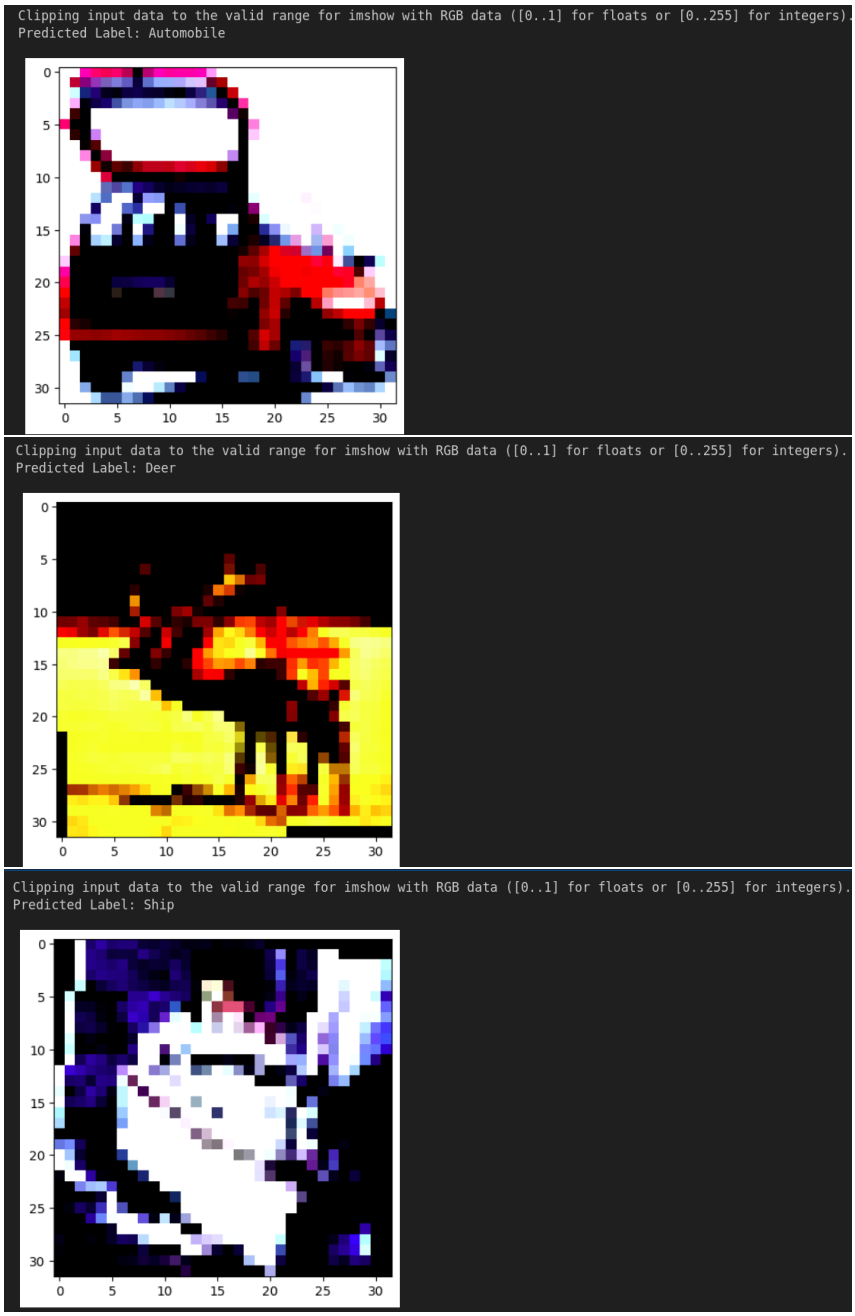
ANN (4) :Outputs 1-Accuracy

```
Accuracy = 56.02%
Classification Report:
              precision    recall  f1-score   support

    Airplane       0.64      0.65      0.64      1000
  Automobile       0.63      0.72      0.67      1000
        Bird       0.51      0.39      0.44      1000
         Cat       0.37      0.38      0.37      1000
        Deer       0.50      0.46      0.48      1000
         Dog       0.43      0.54      0.48      1000
        Frog       0.60      0.62      0.61      1000
       Horse       0.66      0.66      0.66      1000
        Ship       0.66      0.64      0.65      1000
       Truck       0.63      0.55      0.59      1000

    accuracy                           0.56     10000
   macro avg       0.56      0.56      0.56     10000
weighted avg       0.56      0.56      0.56     10000
```

ANN (4) :Outputs 2-Confusion matrix



ANN (4) :Outputs 3- Output for some test images for each model:



2. Comparison Between ANN1,ANN2 and ANN3

Point Of Comparison	ANN1	ANN2	ANN4
Training Accuracy	74.81	70.19	80.27
Testing Accuracy	52.94	50.64	56.02
Hidden Layers	Three	Two	Four

From these outputs we can see that ANN1 gives high training accuracy and highest testing accuracy.

3. Comparison Between VGG16,GoogleNet and ResNet50

Point of Comparison	VGG16	GoogleNet	ResNet50
Training Accuracy	89.34	93.82	98.74
Testing Accuracy	87.36	88.84	92.15

From these outputs we can see that ResNet50 gives the highest training accuracy and highest testing accuracy.

4. Comparison Between Neural networks using transfer learning and not using it

From previous tables we conclude that Transfer learning gives the higher training accuracy and higher testing accuracy than normal neural network
So if you have a complex and deep neural network that requires a lot of computational power and time to train, then you may prefer to use transfer learning.