Data Analytics Final Project: Image Classification

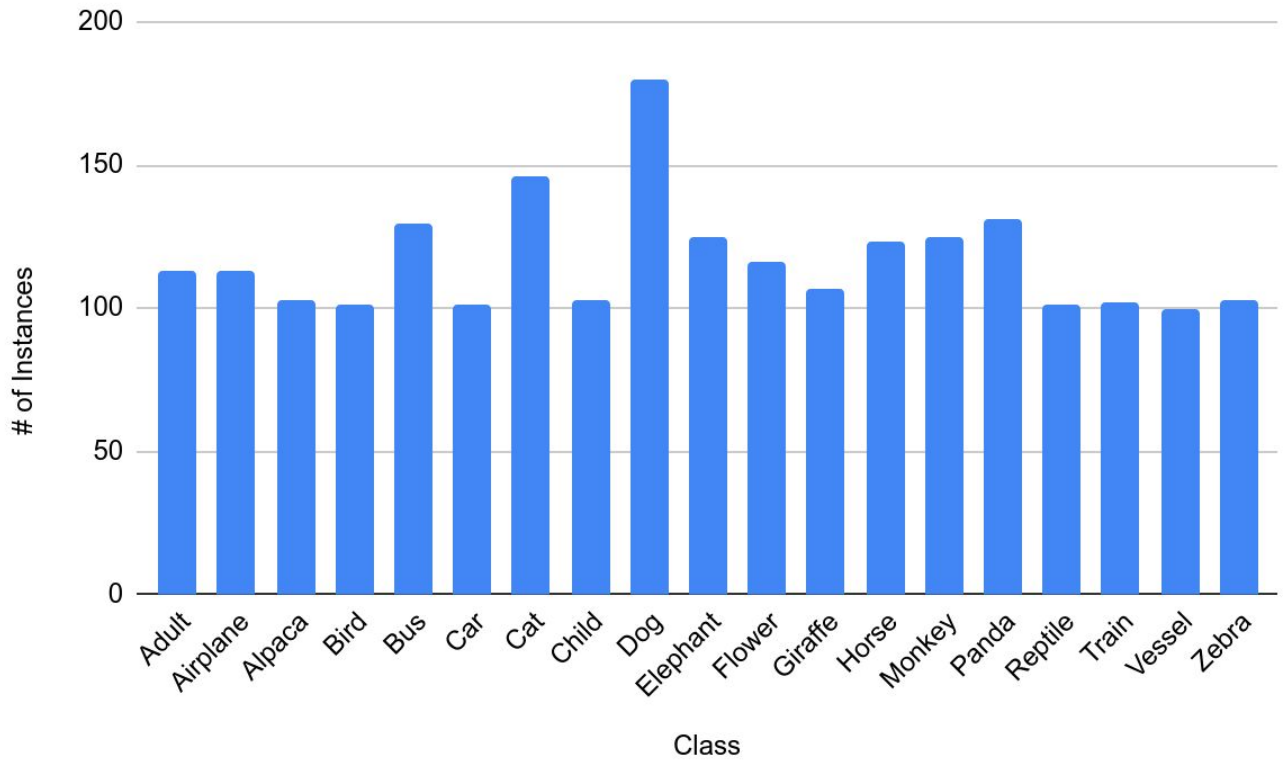
Ahmed Shahabaz
U 8941-5490

UNIVERSITY of SOUTH FLORIDA

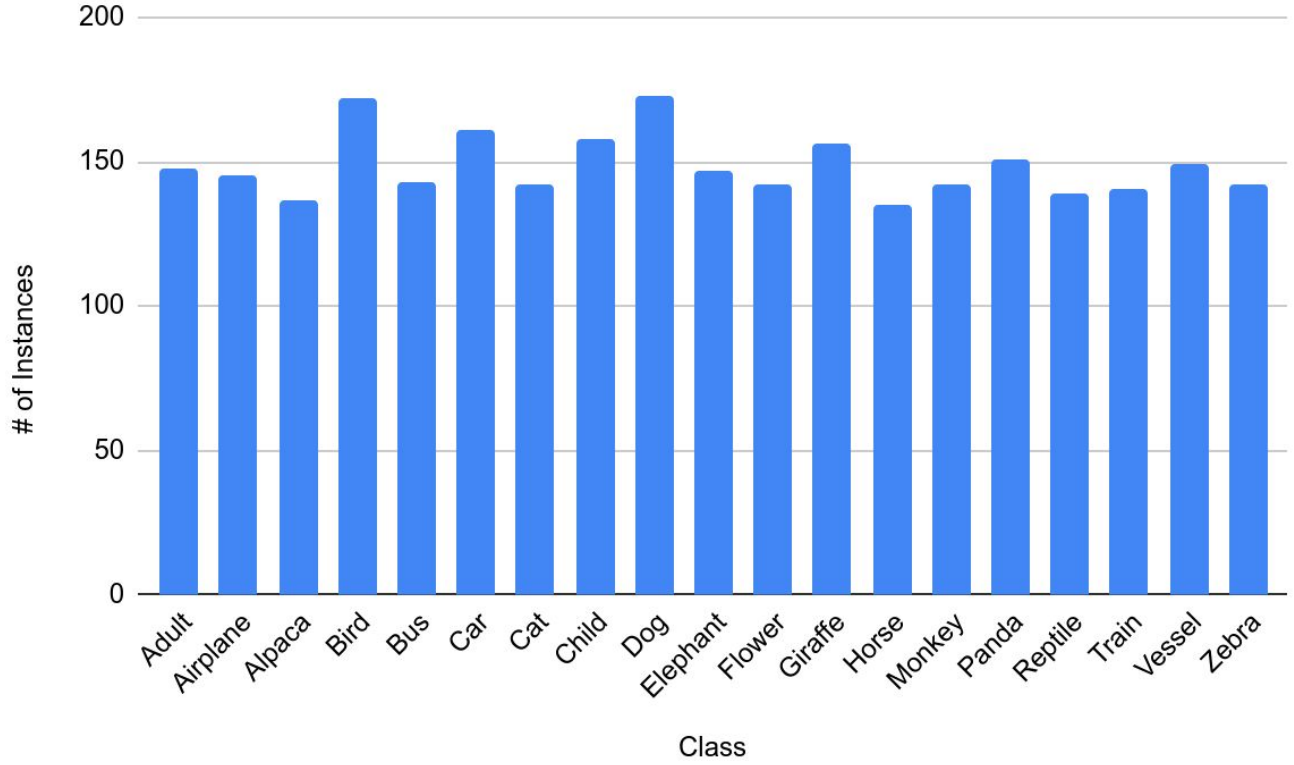
Problem Statement

- Classify test images correctly
- Steps:
  - Input pre-processing
  - Feature extraction - (CNN)
  - Classification
- Challenges:
  - Imbalanced dataset
  - Different lighting conditions in images

Dataset (train) distribution



Dataset (train) distribution using [torch.utils.data.WeightedRandomSampler]



Dataset Statistics

```
Data loader in train mode!
Number of classes: 19.
Train Data size: 2823.
Validation Data size: 647.
 ---------- --------- ----------


Data loader in test mode!
Number of classes: 19.
Test Data size: 570.
 ---------- --------- ----------
```

Problems of using *torch.utils.data.WeightedRandomSampler*

- Unbalanced data issue is solved using oversampling
  - Images are repeated for classes with lower number of images
  - So model might overfit to those classes. As model sees same sample multiple times instead of only once in an epoch
    - Solution: Use Data Augmentation/Image Transformation
- Another approach can be weighted loss

Different data augmentations

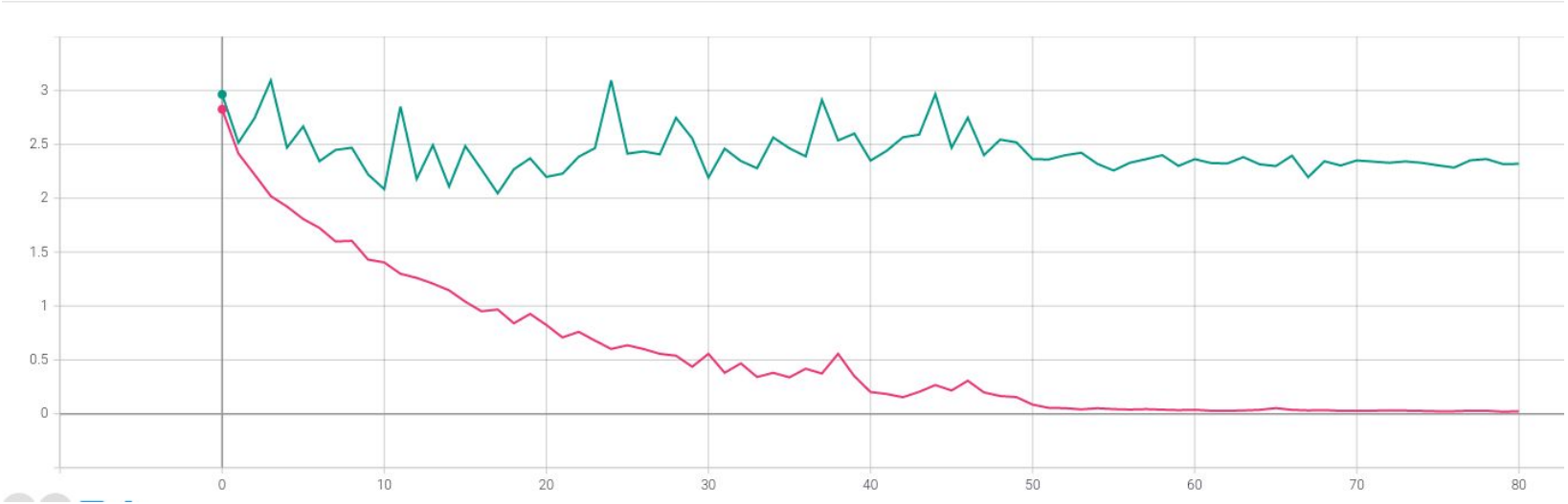
```
if mode == "train" or mode == "Train":
    transforms_list = [
    transforms.Resize((args.image_size)),
    transforms.CenterCrop(args.crop_size),
    transforms.RandomHorizontalFlip(),
    transforms.RandomVerticalFlip(),
    #transforms.RandomAutocontrast(),
    #transforms.RandomEqualize(),
    transforms.RandomApply(torch.nn.ModuleList([
        transforms.ColorJitter(),
        transforms.GaussianBlur(15),
        ]), p = 0.3),
    transforms.RandomRotation(20),
    transforms.ToTensor(),
    transforms.Normalize((0.485, 0.456, 0.406),(0.229, 0.224, 0.225))
    ]

else:
    transforms_list = [
    transforms.Resize((args.image_size)),
    transforms.CenterCrop(args.crop_size),
    transforms.ToTensor(),
    transforms.Normalize((0.485, 0.456, 0.406),(0.229, 0.224, 0.225))
    ]
```

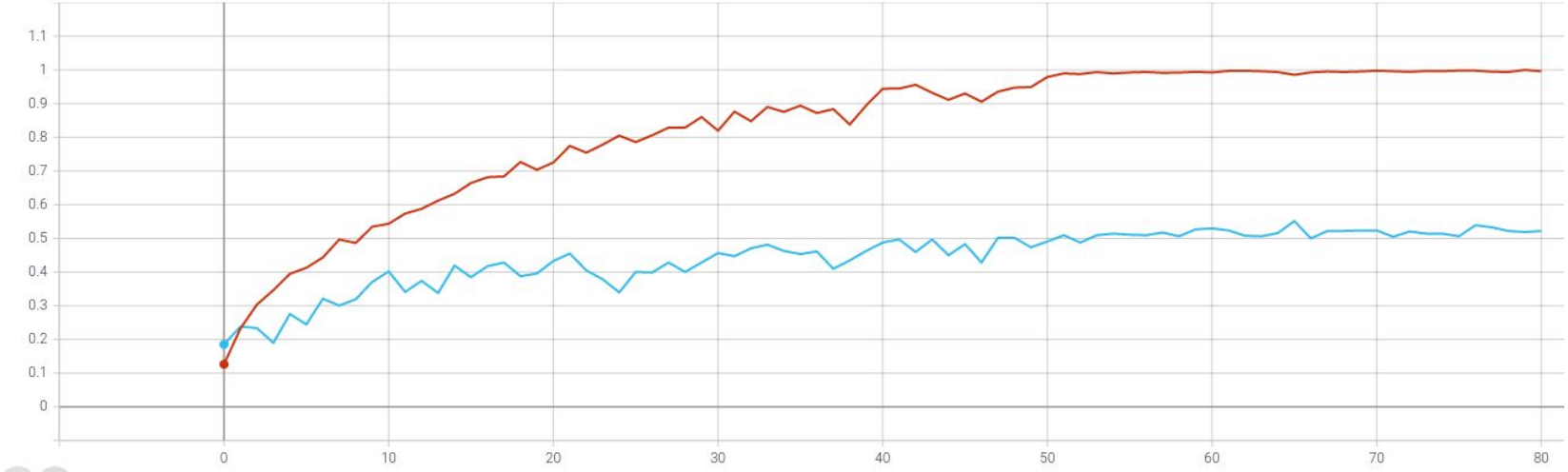
Addressing overfitting

- Early Stopping
  - If no change in validation loss for a few epochs then training stops.
- Different regularization
  - Weight decay
    - Multiply the sum of squares of weights (parameter) with another smaller number. This number is called ***weight decay*** or wd.
    - Prevents weights getting too large
    - wd = 1e-6 worked better for our case
  - Drop out
    - Randomly drops some of the neurons of a layer
    - Only tried 0.5

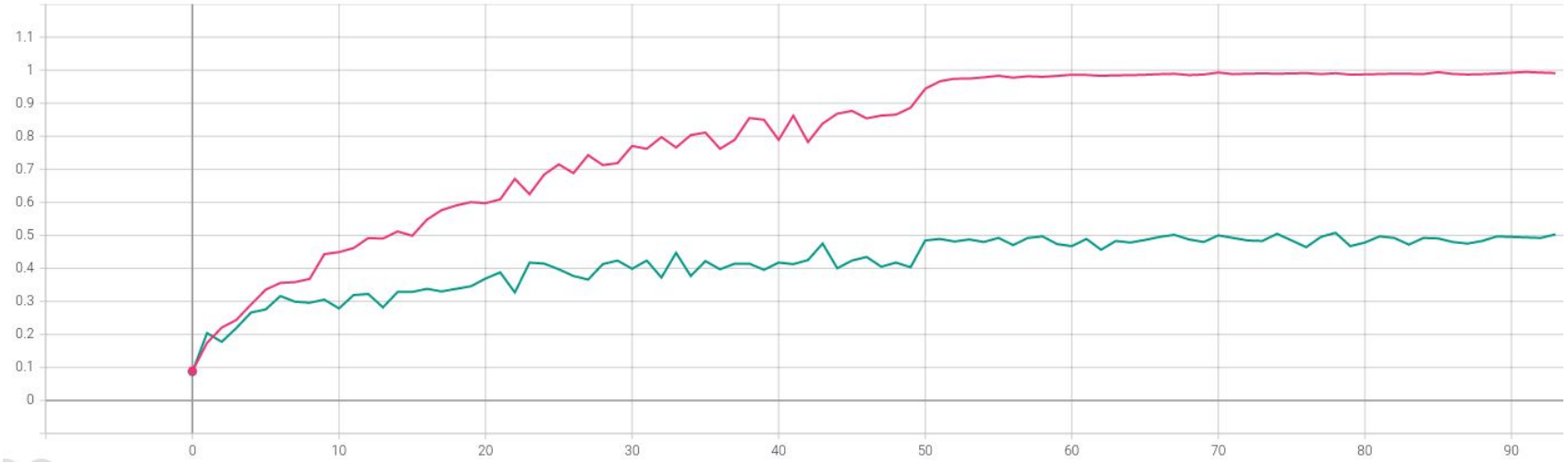
ResNet18: Loss w/o pre-training (no lr. scheduler)



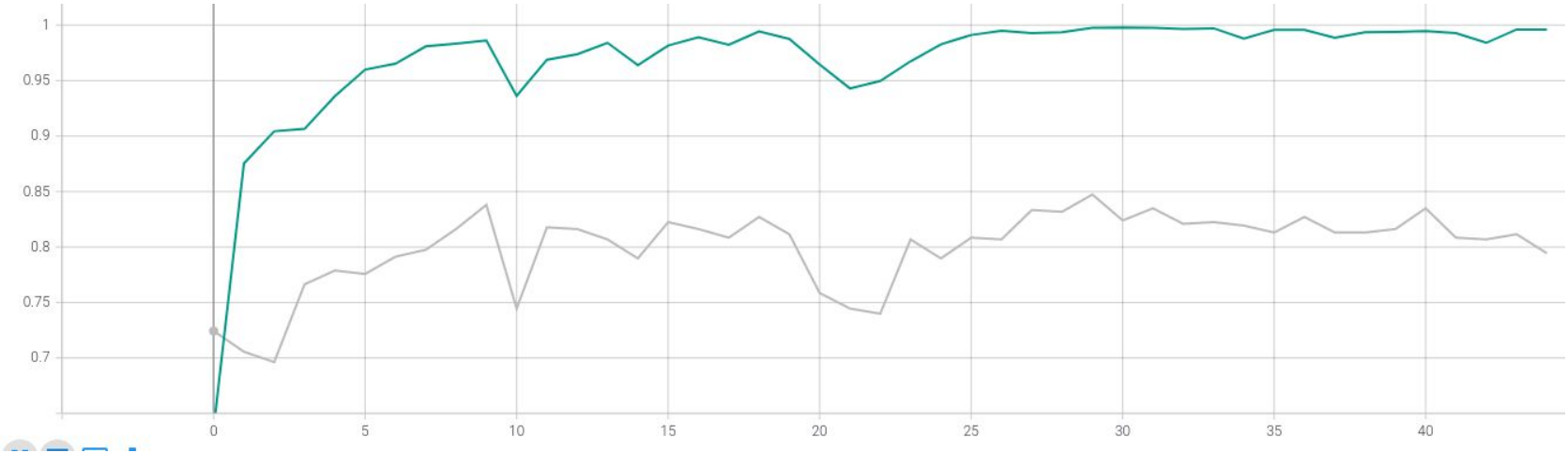
ResNet18: Accuracy w/o pre-training (no lr. scheduler)



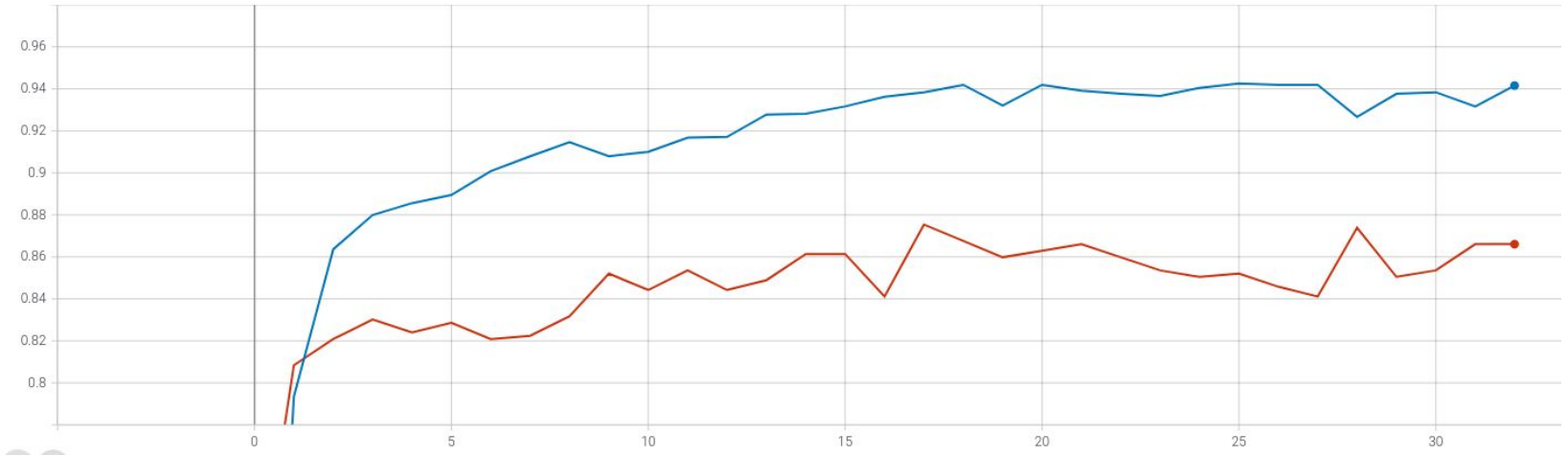
ResNet34: Accuracy w/o pre-training (lr. scheduler = 50)



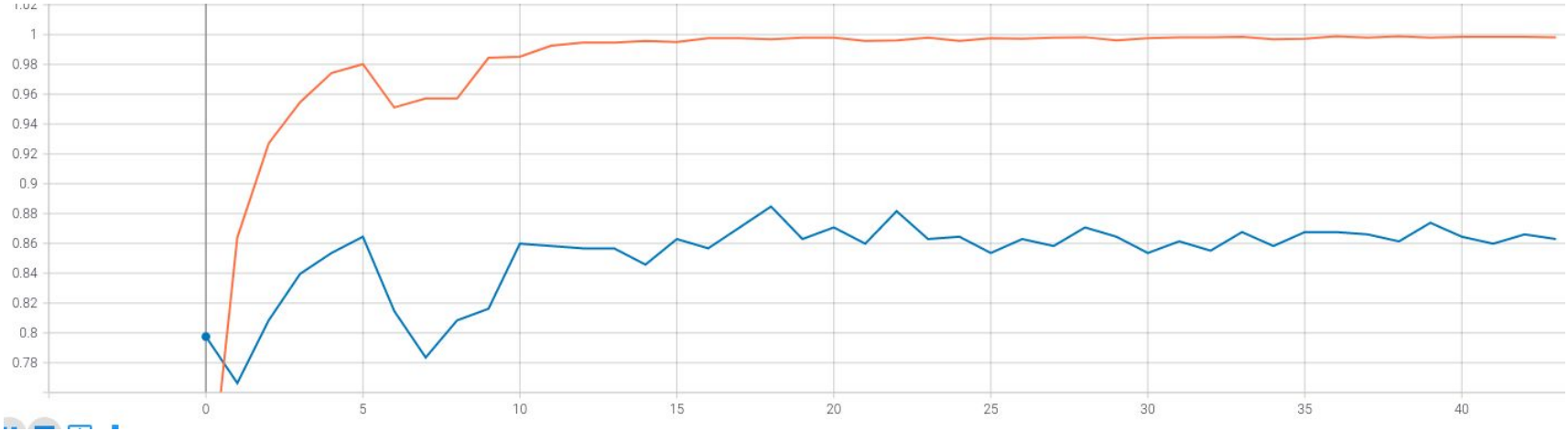
ResNet34: Accuracy with pre-training (no lr. scheduler)



ResNet34 (fc layer): Accuracy with pre-training (lr. scheduler = 15)



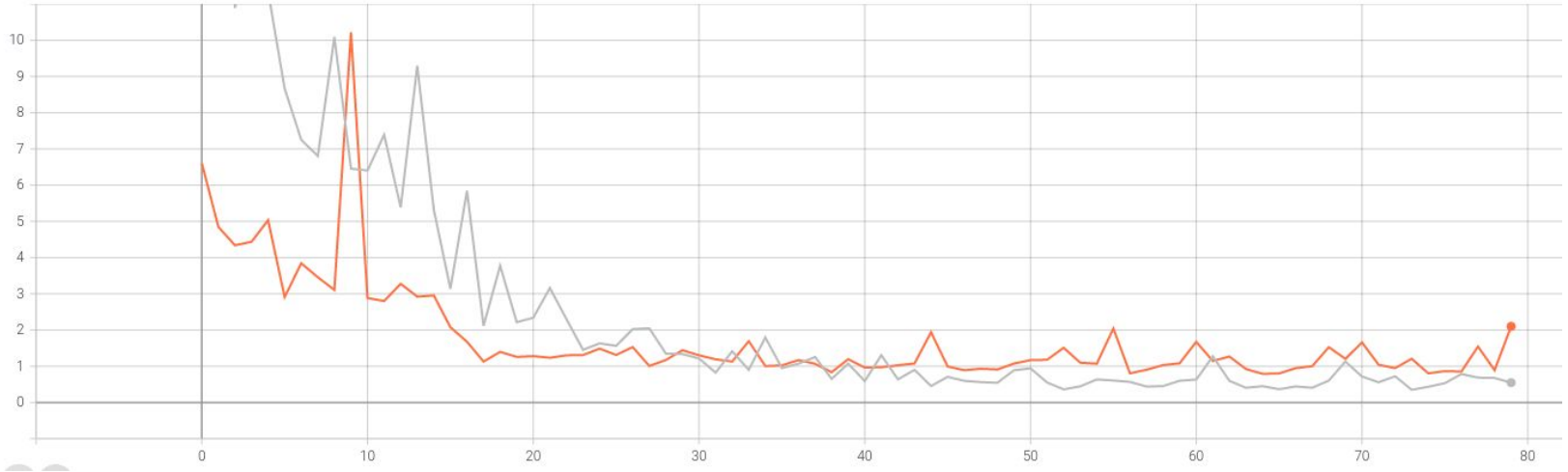
ResNet18: Accuracy with pre-training and (lr. scheduler = 10)



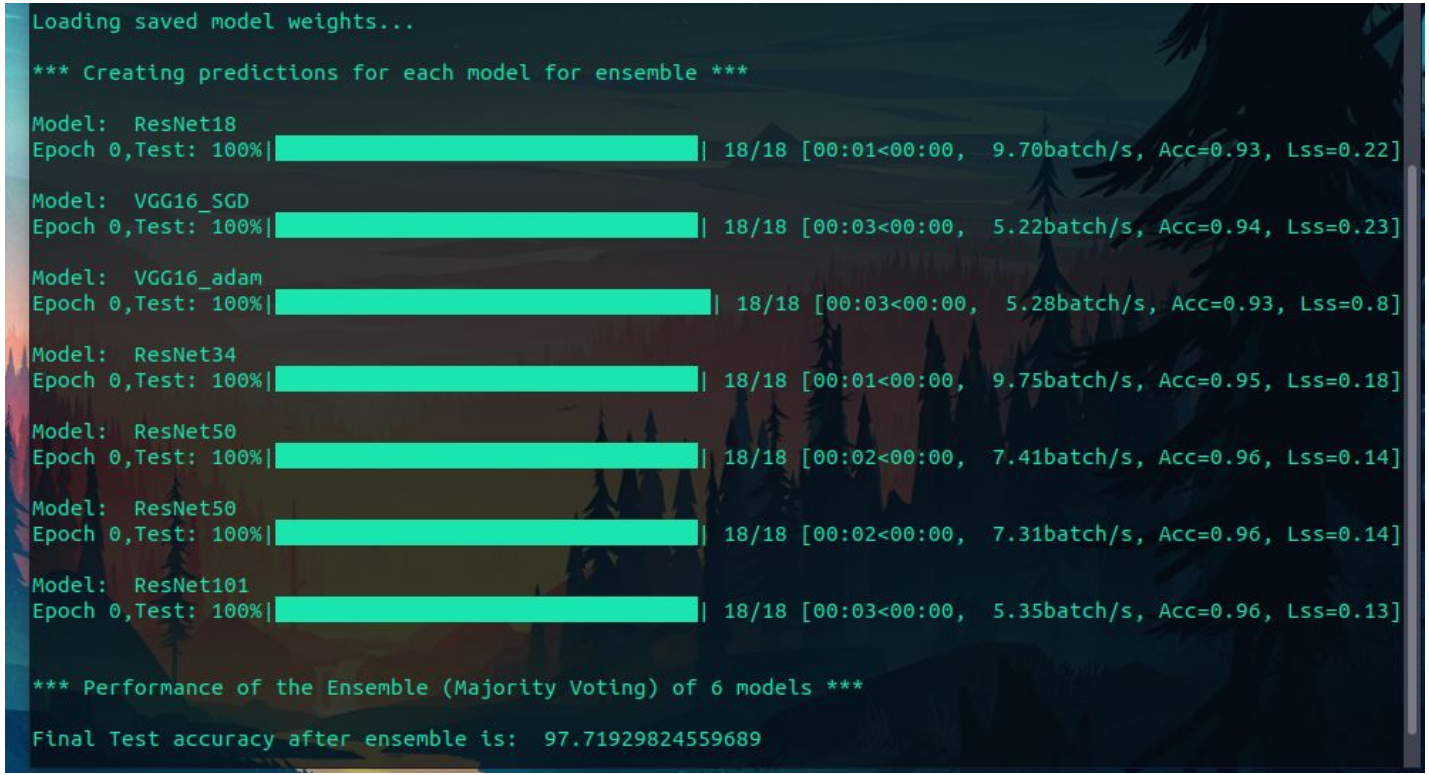
Comments on results of Deep Learning

- ResNet34, ResNet50, ResNet101 and VGG16 with pre-trained weights are better
  - Only the fc layer of the ResNets and the classification layer of VGG16 was trained
  - Using lr. scheduler = 15 gives better performance
- Overfits easily to ResNet18
- Optimizer used: SGD

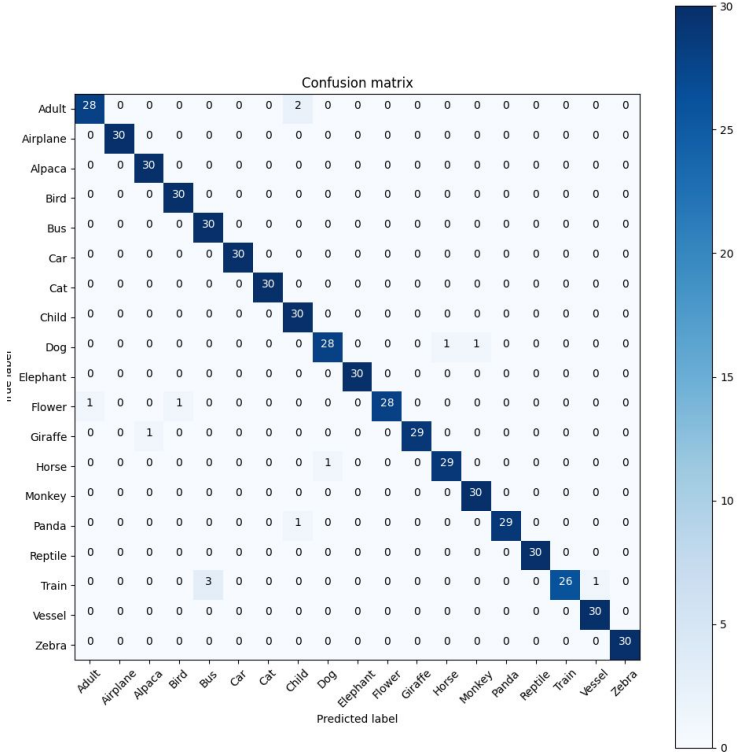
VGG16 (classification layer): Loss using ADAM optimizer (lr. scheduler = 15)



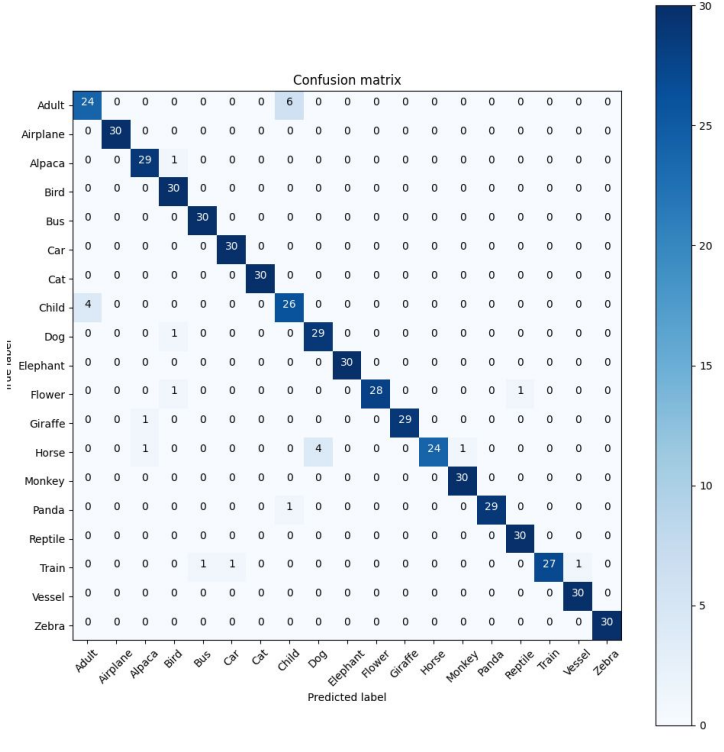
Test Results



Confusion matrix of ensemble vs ResNet101 (fc layer)



CM of ensemble of deep learners



CM of ResNet101

RandomForest

- RandomForestClassifier from sklearn.ensemble

n-estimators	50	100	200	300	500
Accuracy	0.17543	0.21754	0.20350	0.22280	0.22807

Support Vector Machine (SVM)

- sklearn implementation of SVM

SVM Kernel	Linear	RBF
Accuracy	0.13333	0.18070

Conclusion

- Deep learning algorithms outperformed traditional algorithms
- Ensemble of multiple deep learners was better than any single one of them
- Incase of deep learning transfer learning worked much better than training from scratch
  - Which says our training dataset was not large enough for training a deep learning algorithm from scratch

Thank You

UNIVERSITY of SOUTH FLORIDA