

# Assignment 3

Name(s): Ziyue Tang

NetID(s): ziyet3

## Part 1: Self-supervised Learning on CIFAR10

### 1) Rotation training

*Report the hyperparameters you used to train your model. Discuss any particular implementation choices which caused significant performance increases.*

Hyperparameters:

Optimizer: Adam

Criterion: Cross Entropy Loss

Learning rate: 0.01

Number of epochs: 45

Accuracy: 78.75 %

### 2) Fine-tuning late layers

*Report the hyperparameters you used to fine-tune your model. Compare the performance between pre-trained model and randomly initialized model.*

Hyperparameters:

Optimizer: Adam

Criterion: Cross Entropy Loss

Learning rate: 0.01

Number of epochs: 20

Accuracy for the fine-tuned pre-trained model: 72.40 %

Accuracy for the fine-tuned randomly initialized model: 46.14 %

The performance of the pre-trained model is much better than randomly initialized model.

### 3) Fully supervised learning

*Report the hyperparameters you used to fine-tune your model. Compare the performance between pre-trained model and randomly initialized model. Discuss anything you find interesting comparing fine-tuning the late layers only in section (2) and fine-tuning the whole model in section (3).*

Hyperparameters

Optimizer: Adam

Criterion: Cross Entropy Loss

Learning rate: 0.01

Number of epochs: 20

Accuracy for the supervised training on the pre-trained model: 79.52 %

Accuracy for the supervised training on the randomly initialized model: 82.61 %

The performance of the randomly initialized model is even a bit higher than the pre-trained model.

Something I found interesting:

1. Fine-tuning the late layers ends up much smaller test accuracy than fine-tuning the whole model.
2. In section (2) the test accuracy changes very stable and slow. But for the section (3), test accuracy changes drastically during early epochs.

#### 4) Extra credit

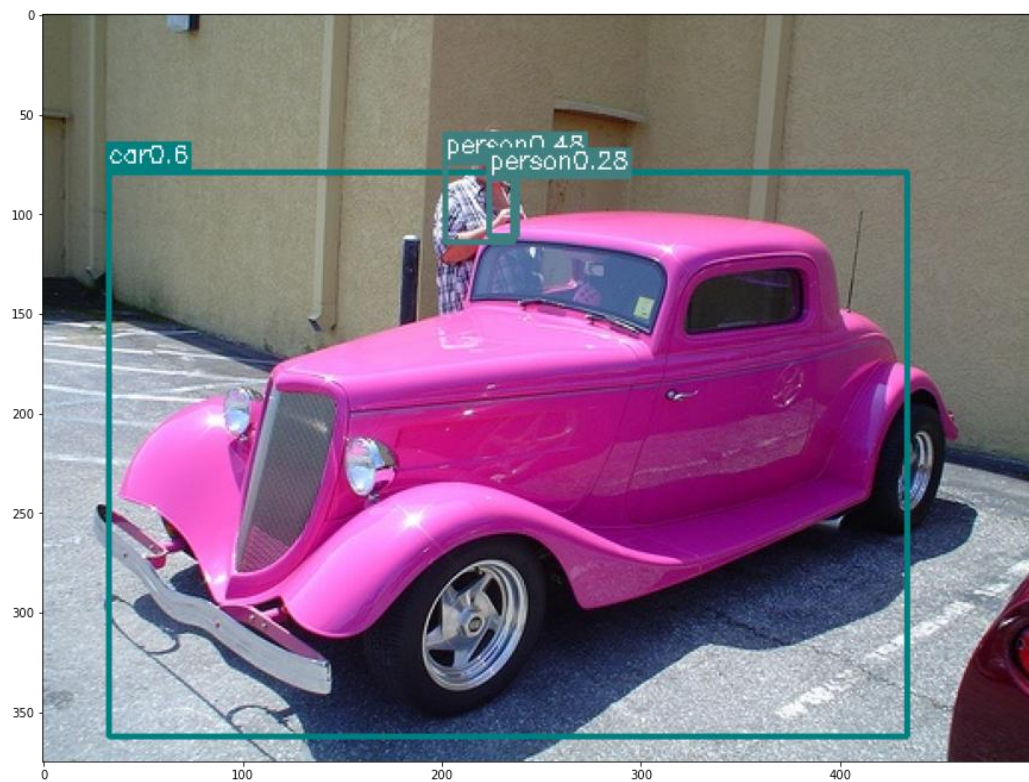
## Part-2: Object Detection by YOLO

1. My best mAP value on Kaggle : 0.44649
2. Did you upload final CSV file on Kaggle: Yes
3. My final loss value : 0.941
4. What did not work in my code(if anything): The kernel always died when trying to predict. When trying to predict. And I ran this code to solve the problem:  
`os.environ["KMP_DUPLICATE_LIB_OK"] = "True"`. I can't not run the training on the gcp because it always get stucked at "`pred = net(images)`". So, I have to rent another VM myself. And in order to get CUDA available, I ran this in the terminal "`pip install torch==1.13.0+cu116 torchvision==0.14.0+cu116 torchaudio==0.13.0 --extra-index-url https://download.pytorch.org/whl/cu116`". Also, the file path in section "Cell added to get intermediate mAP values for students" is not correct, have have to add "checkpoint/" before the file name. Also, the given code didn't specify "`img_root`", and I had to do it for it.
5. Sample Images from my detector from PASCAL VOC:









**Extra Credit for YOLO :**