**EEEM071 Advanced Topics in Computer Vision and Deep Learning**   
**Coursework Assignment (Spring 2024)**  
**Vehicle Re-identification**

**NAME**: Rajnish Kumar

**URN**: 6846168

**Baseline**: The default settings already provided for you in the code base.

**Hyperparameters**: The parameters that are not learnable, and you can set before starting the training.

|  |
| --- |
| **Important instructions:**   1. Regarding log files:    1. Please submit your own log files from your codebase.    2. All log files are automatically watermarked and are unique to each run. Do not change the log file structure.    3. All log files should be named based on the section number and the question number. (log\_{Section\_num}\_{Question\_num}.txt) For example, a log file generated for an experiment corresponding to question 2 in section 1, should be named as “log\_1\_2.txt” 2. Regarding word limit:    1. Please ensure your answers do NOT exceed the word limit.    2. Going beyond the word limit will be penalized.    3. Content within Tables/graphs/log files is not counted in the word limit. 3. Regarding *presentation and clarity* of your answers:    1. In addition to the scientific content, you will also be assessed on the presentation and clarity of the writing.    2. This accounts for 10 marks. Following criteria to be taken into consideration:       1. Figures should be well presented. The axis and the markings should be easily readable.       2. The writing should be clear, grammatically correct, and the ideas come across easily to the reader.       3. It is important to use tables if you’re discussing results across different values of hyperparameters. |

**NOTE**: Please read all questions carefully before attempting to answer. The below three sections account for 90 marks and (as mentioned above) 10 marks are allotted for *presentation and clarity* of the overall report.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Section 1 – Familiarity with the provided code. [40 marks]** **Max. 200 words.**   1. Run the code using the default settings. **[20 marks]**    1. Provide evidence in terms of the log file.  |  |  |  | | --- | --- | --- | | Model | Results | log file | | mobilenet\_v3\_small | mAP: 45.0%  CMC curve  Rank-1 : 81.1%  Rank-5 : 91.0%  Rank-10 : 94.5%  Rank-20 : 96.6% |  |  * 1. Discuss the training and evaluation process followed by implications of the observed performance using an appropriate metric.  Max. 100 words.   The training process involves feeding the labeled data into the deep neural model to calculate the loss function and update the model parameters by using backpropagation. Our main goal is to minimize loss function so that our model can represent dataset features. The training process also includes fine-tuning hyperparameters for optimization.  To evaluate the model’s accuracy mAP (Mean Average Precision) and CMC (Cumulative Matching Characteristics) are used. The mAP for the mobilenet\_v3\_small model is below average 45% while the CMC curve Rank1 is 81.1% which shows that the model can correctly identify rank1 vehicles most of the time.   1. Apply another CNN variant (that is not provided in the default settings). [**10 marks]**    1. Provide evidence in terms of the log file.  |  |  |  | | --- | --- | --- | | Model | Results | Log file | | ResNet50 | mAP: 49.3%  CMC curve  Rank-1 : 82.2%  Rank-5 : 91.8%  Rank-10 : 94.2%  Rank-20 : 96.6% |  |  * 1. Critically discuss and contrast the results with what observed in question 1 above.  Max. 50 words.   Resnet50 shows average performance in mAP metrics and high scores in the CMC curve. For surveillance and identification, these results are unsatisfactory. In comparison with mobilenet\_v3\_small, Resnet50 is more consistent and robust with the same default setting and is able to retrieve information better than other models.   1. Apply one more neural network architecture (say, a transformer variant). [**10 marks]**    1. Provide evidence in terms of the log file.  |  |  |  | | --- | --- | --- | | Model | Results | Log File | | Vgg16 | mAP: 22.4%  CMC curve  Rank-1 : 58.5%  Rank-5 : 71.5%  Rank-10 : 78.7%  Rank-20 : 84.6% |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  |  | CMC |  |  |  | | Model | mAP | Rank-1 | Rank-5 | Rank-10 | Rank-20 | | Mobilenet\_v3\_small | 45.00% | 81.10% | 91.00% | 94.50% | 96.60% | | Resnet 50 | 49.30% | 82.20% | 91.80% | 94.20% | 96.60% | | Vgg16 | 22.40% | 58.50% | 71.50% | 78.70% | 84.60% |  * 1. Critically discuss and contrast the results with what observed questions 1 and 2 above. Max. 50 words.   Vgg16 shows bad performance in mAP and the CMC curve metrics. During training, model was able to achieve high accuracy but on test set accuracy was very low. This shows model is overfitting. In comparison with mobilenet\_v3\_small and Resnet50, Vgg16 performed worse and was not able to retrieve correct information. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Section 2 – Dataset preparation and Augmentation experiments. [25 marks]** **Max. 250 words.**   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | ResNet 50 | Augmentation |  |  |  |  |  | Log File | |  |  | mAP | Rank-1 | Rank-5 | Rank-10 | Rank-20 |  | |  | No aug | 49.30% | 82.20% | 91.80% | 94.20% | 96.60% | Default | |  | Random Erase | 52.40% | 83.10% | 92.00% | 95.10% | 97.30% |  | |  | Color Jitter | 51.90% | 83.20% | 92.30% | 95.50% | 97.10% |  | |  | Color agu | 48.20% | 81.60% | 91.80% | 94.80% | 97.20% |  | |  | All aug | 51.00% | 82.10% | 92.30% | 95.50% | 97.70% |  |  1. Apply any one data augmentation technique (for example, “crop”). Discuss the results in comparison when no data augmentation is employed, i.e. the default configuration in the provided code. **[10 marks]** Max. 100 words.   Random erase is a data augmentation technique that erases parts of the image during the training process so that the model can learn only important features of data and discourage learning noise and irrelevant features. With the introduction of random erase, the performance of ResNet50 was improved. It was also observed that during training model accuracy remained below 100% most of the time. This shows that random erase acts as a regularizer and prevents overfitting. Sometimes random erase caused loss (Xent) to increase but it improves model performance on testing data.   1. Apply two different augmentations in isolation (for example, only “blurring” or only “horizontal flip” etc.) and discuss the implications of each augmentation and analyze the results in comparison when no data augmentation is employed. Highlight any improvement or drop in overall score. **[10 marks]** Max. 100 words.   Color Jitter and color augmentation are data augmentation techniques used to diversify the dataset by randomly adjusting contrast, saturation, brightness, hue, and color. Color jitter has a positive impact on the test scores of Resnet50. It introduces color properties to the model because of which model was able to generalize well on different color settings and reduce overfitting. On the other hand, color augmentation slightly decreased the test scores of Resnet50. Color augmentation has wider modification than color jitter. Too much augmentation sometimes leads to added noise because of which model is not able to generalize well.   1. Combine augmentation techniques employed in questions 1 and 2 above (for example, “crop” + “blurring” + “vertical flip”). Highlight any improvement or drop in overall score. **[5 marks]** Max. 50 words.   Combining all the augmentation random erase, color jitter, and color augmentation the accuracy of ResNet50 has slightly improved from default settings. Accuracy has dropped during training because of the regularization effect of these augmentations but models have become more robust and able to generalize well on testing datasets. |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Section 3 – Exploration of Hyperparameters [25 marks]** **Max. 250 words.**   1. Exploration of Learning Rate (LR). **[10 marks]**    1. Experiment with 4 values of LR (in addition to the default value).  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Resnet50 | Learning rate | mAP | Rank-1 | Rank-5 | Rank 10 | Rank20 | Properties of Xent | Log files | |  | 0.1 | 3.10% | 4.10% | 12.80% | 20.60% | 32.30% | No convergence and oscillations |  | |  | 0.005 | 16.40% | 36.10% | 54.80% | 65.70% | 75.40% | No convergence |  | |  | 0.0003 | 49.30% | 82.20% | 91.80% | 94.20% | 96.60% | convergence |  | |  | 0.00007 | 59.90% | 87.50% | 95.00% | 96.80% | 97.90% | Convergence |  | |  | 1.00E-05 | 45.40% | 75.60% | 86.70% | 91.50% | 95.00% | Slow convergence |  |  * 1. Discuss the effects observed on overall performance. Max. 100 words.   Learning rate is a hyperparameter that manages how much model parameters are updated after every iteration during training process. In Resnet50 large learning rates like 0.1 and 0.005 cause oscillations and there was no convergence of loss function. The mAP and CMC scores were very low for both training and testing process. On the other hand, 0.0003 has stability and convergence but scores were average. For Learning rate 0.00007 convergence is fast and stable and accuracy scores are best. Lr-0.00001 had very slow convergence and needs more iterations to increase the accuracy.   1. Exploration Batch sizes. **[10 marks]**    1. Fixing the best LR value from the experiments in question 1 above, experiment with 4 different values of the BS (in addition to the default value).  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | ResNet 50 lr=0.00007 |  |  |  |  |  |  |  | | batch size | mAP | Rank-1 | Rank-5 | Rank-10 | Rank-20 | Properties of Xent | Log file | | 16, 50 | 59.40% | 88.30% | 94.90% | 96.60% | 98.20% | Too Noisy gradient and slow convergence, computational cost |  | | 32, 75 | 61.10% | 88.60% | 95.10% | 96.80% | 98.40% | Noisey gradient and slow convergence,  Computation cost. |  | | 64, 100 | 59.90% | 87.50% | 95.00% | 96.80% | 97.90% | Convergence | Default value | | 84, 100 | 59.00% | 87.70% | 94.50% | 97.40% | 98.70% | Fast convergence |  | | 128,150 | 59.00% | 85.50% | 93.90% | 96.20% | 98.00% | Fast and gradual convergence |  |  * 1. Discuss the effects observed on overall performance. Max. 100 words.   Batch size is a hyperparameter that affects how many samples are processed before updating the parameters of a model. Extreme batch sizes can also lead to overfitting and underfitting in a model. Small training batch sizes 16 and 32 have slow convergence, noisy gradient (oscillations), and parameters are updated frequently. On the other hand, large batch size like 84 and 128 cause fast convergence because weights are updated less frequently. There is little effect of these batch sizes on mAP and CMC except batch size 32 because noisy gradient had produced a regularization effect and prevented overfitting.   1. Exploration of the optimizer. **[5 marks]**    1. Fixing the best LR value and best Bath Size value from the experiments in question 1 and 2 above, respectively, experiment with changing the optimizer to SGD. (use PyTorch’s internal class)  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | ResNet50 |  |  |  |  |  |  | | lr=0.00007bs=  32, 75 | mAP | Rank-1 | Rank-5 | Rank-10 | Rank-20 | Log | | SGD | 26.30% | 53.20% | 70.70% | 78.00% | 84.70% |  | | amsgrad | 61.10% | 88.60% | 95.10% | 96.80% | 98.40% | Default |  * 1. Discuss the effects observed on overall performance. Max. 50 words.   The mAP and CMC scores are very low for SDG optimizers. During training model loss remains high and training accuracy scores are very low. It can be concluded that the model is underfitting with SGD optimizer. This may be caused by noise or hyperparameters need to be modified. |