EEEM071 Coursework Report

ADVANCED TOPICS IN COMPUTER VISION AND DEEP LEARNING

Anitha Kugur Parameshwarappa (ak02923)

Abstract

Vehicle re-identification is an active research area in intelligent transportation systems (ITS) that involves identifying a specific vehicle across non-overlapping images in a multi-camera network. This task is challenging due to multiple factors, including similarity between classes, variations within classes, and changes in viewpoint, and spatiotemporal ambiguity. The development of effective vehicle re-id technologies has the potential to provide valuable insights into traffic flow patterns and vehicle behavior in urban environments. There have been several research efforts to address the challenges associated with vehicle re-id. Deep learning-based methods such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and Siamese networks are commonly used in vehicle re-identification tasks. These methods have been shown to be effective in capturing and learning relevant features from images and improving the performance of the vehicle re-identification task. This report is a study of various Resnet18 architecture being used for vehicle re-identification on the VeRi dataset.

1. Introduction

Vehicle re-identification (vehicle re-ID) is a computer vision task that involves recognizing the same vehicle across multiple cameras or time frames. Its aim is to match a query vehicle image with images of the same vehicle in the gallery. This has various practical applications such as surveillance, traffic monitoring, and law enforcement. However, vehicle re-ID can be challenging due to variations in vehicle appearance caused by factors like changes in viewpoint, lighting, weather conditions, and occlusions. Vehicle identification is a critical and fast-growing research area in Intelligent Transportation Systems (ITS) with significant potential for further development and innovation. Vehicle re-identification using deep learning has improved significantly in recent years as a result of advances in computer vision and deep learning techniques. Deep learning models can learn complicated patterns and attributes from photos, making them ideal for vehicle re-identification tasks. improve transportation efficiency, safety, and security. One of the most popular approaches for vehicle re-identification is using Convolutional Neural Networks (CNNs). CNNs can learn spatial hierarchies of features that can be used to identify vehicles across multiple non-overlapping camera views. These features are learned through multiple convolutional layers that extract low-level to high-level features from images.

1.1. Dataset

The dataset used for this analysis is called VeRi having 50,000 images of 776 vehicles in 24 hours which are in a distance of 1 sq. km to the nearby 20 cameras. The dataset is split into train, query, test images. With train set having 37,778 images, gallery set having 11,579 images, query set having 1,678 images.

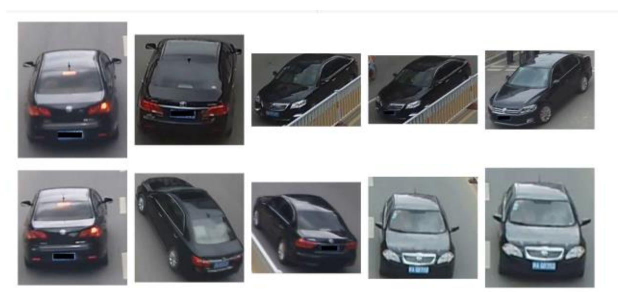

Figure 1. First 10 images from the training set

Vehicle re-identification is a process that involves matching a query image against a gallery set of images. The gallery set contains previously captured images in a surveillance camera network. The main objective is to find the image in the gallery set that is most similar to the query image.

2.Model Selection

Selecting an appropriate model for image classification can be influenced by several factors, including the dataset's size and complexity, the desired accuracy, and the computational resources available. It is crucial to pick a model that achieves a balance between performance and

computational efficiency based on the task's specific requirements and available resources.
The 3 pre-trained models that were chosen in this experimentation are,

1.mobilenet_v3_small
2.resnet18
3.regnet_x_400mf.

3. Data Augmentation

Data augmentation is a technique used in machine learning to expand the size of a dataset by generating new training examples from existing ones. The aim of data augmentation is to enhance the performance of a machine learning model by increasing the diversity and size of training data. There are several techniques available for data augmentation.

The implemented data augmentation techniques were Random Erase, Color Jitter, and Color Augmentation which are all data augmentation techniques commonly used in computer vision tasks.

Random Erase: Random Erase is a data augmentation technique that randomly removes rectangular patches from an image during training. This can help to prevent the network from overfitting to specific features in the training set and encourage it to learn more robust and generalizable representations. The size, aspect ratio, and color of the erased patch are randomly chosen, and the erased region is filled with either zeros or random pixel values.

Color Jitter: Color Jitter is a data augmentation technique that randomly adjusts the brightness, contrast, saturation, and hue of an image during training. This can help to improve the network's ability to generalize to different lighting conditions and color distributions. The magnitude of the jitter is randomly chosen within a specified range, and the jitter is applied independently to each color channel.

Color Augmentation: Color Augmentation is a data augmentation technique that applies a color transformation to an image during training. This can help to improve the network's ability to recognize objects under different color conditions, such as changes in lighting or camera settings. The color transformation can be chosen to simulate various lighting conditions, such as daylight, tungsten, or fluorescent, or to simulate color distortions, such as color cast or color noise.

Figure 2. Shows the training set images after data augmentation

	Model	Rank 1	Rank 2	Rank 3	Rank 4
1.	Mobilenet_v3_small	82.9%	92.2%	95.1%	97.1%
2.	Regnet_x_400mf	84.4%	93.1%	96.6%	98..4%
3.	Resnet18	85.3%	93.6%	96.2%	97.7%

The above table gives the results obtained from the three models after the introduction of data augmentation techniques.

Mobilenet_v3_small with data augmentation and adam as an optimizer yielded these results.

```
Results ----------
mAP: 53.4%
CMC curve
Rank-1  : 83.9%
Rank-5  : 93.7%
Rank-10 : 96.0%
Rank-20 : 97.4%
```

So, we are selecting the Restnet18 as best model to tune the hyper-parameters and also to experiment with the architecture of the CNN.

4. Experimentation on Resnet18 architecture

ResNet-18 is a popular CNN architecture that uses residual connections to train very deep neural networks with improved performance. Its use of residual connections allows it to overcome the vanishing gradient problem, and it has been often used in computer vision tasks such as image classification. It consists of several blocks, each of which contains multiple convolutional layers with batch normalization and ReLU activation, followed by a residual connection. The first block downsamples the input feature map by a factor of two, while subsequent blocks maintain the same spatial resolution.

Addition of LeakyReLU activation function in place of ReLu and an extra layer (layer 5) to the ResNet18 architecture so as to increase the complexity of the model which can improve the model's capacity to capture complex patterns in the data have been introduced. The activation function is crucial in adding non-linearity to the output of a neuron, enabling the neural network to learn intricate patterns in the data. LeakyReLU is a variant of the conventional ReLU activation function. The LeakyReLU function introduces a small slope for negative input values, which can potentially improve the performance of deep neural networks. It addresses the issue of "dying ReLU", where some neurons in the network become inactive and stop learning during training.

Activation function	LeakyReLU - output of batch-normalization layer 1 in the forward function.
New Layer Addition	Layer 5 with 1 block, 512 output channels, and a stride of 1.
Optimizer	Adam
Data Augmentation	True
No. of epochs	20

```
Model size: 15.897 M

Results ----------
Acc 97.66 (97.74)
mAP: 54.5%
CMC curve
Rank-1  : 85.4%
Rank-5  : 93.6%
Rank-10 : 96.4%
Rank-20 : 97.9%
```

5. Hyper Parameters Tuning

Hyperparameter tuning is an important step in the machine learning workflow, as it can significantly improve the performance and generalization of a model. It is important to carefully select the hyperparameters to tune and choose an appropriate tuning method based on the available resources and problem characteristics. The hyperparameters such as learning rate (default=0.0003), learning rate scheduler(default= multi-step) and the step size (default= [20 40]) are adjusted in this experimentation keeping the optimizer same.

Optimization algorithms (default = amsgrad) are used in machine learning to update the parameters of a model during training in order to minimize the loss function and improve the performance of the model on the training data. There are many optimization algorithms available, each with its own strengths and weaknesses. Some of the optimizers which are used are Stochastic Gradient Descent (SGD), Adam, RMSProp, and Amsgrad (default). Experimentation was done with optimizers like Adam and SGD which yielded in high loss and less training accuracy. The result after using 'SGD' as an optimization algorithm is given as below.

```
Results ----------
Acc 8.20 (10.70)

mAP: 25.6%
CMC curve
Rank-1  : 54.9%
Rank-5  : 72.3%
Rank-10 : 80.9%
Rank-20 : 88.0%
```

The suitable one other than amsgrad was adam optimizer.

The learning rate controls the step size during optimization. It determines how quickly the model learns and converges. A high learning rate may cause the loss to diverge, while a low learning rate may cause the model to learn very slowly.

During training, the learning rate scheduler is utilized to modify the learning rate of a model. It is capable of gradually decreasing the learning rate as training advances, which can potentially accelerate the convergence of the model and mitigate overfitting. There are several types of learning rate schedulers available, including StepLR, MultiStepLR, and CosineAnnealingLR. The StepLR is being used here.

The step size is a crucial hyperparameter to consider when implementing a learning rate scheduler as it determines the number of epochs after which the learning rate is decreased. By adjusting the step size, we can control the rate at which the learning rate is reduced during training. A smaller step size may lead to a more gradual reduction in the learning rate, potentially improving the model's accuracy, but may also increase the training time. In contrast, a larger step size may reduce the training time but could also result in a suboptimal solution. Therefore, it's essential to carefully tune the step size based on the specific requirements of the task and the available resources. size may cause the learning rate to decrease too quickly, while a larger step size may cause the learning rate to decrease too slowly.

Learning rate	0.0005
Learning Rate scheduler	Single-step
Step size	10
No. of epochs	20

The results were found to be as given below,

```
Acc 98.05 (97.61)
Results ----------
mAP: 57.5%
CMC curve
Rank-1  : 85.9%
Rank-5  : 94.7%
Rank-10 : 96.7%
Rank-20 : 98.7%
```

Preserving the same architecture resnet18 with an additional layer 5 and without changing the activation function. Label smoothing, num_instances, and margin are all important hyperparameters in metric learning for deep learning, and their values can significantly affect the performance and generalization of the model. Optimal values for these hyperparameters are typically determined through cross-validation.

Label smoothing is a regularization technique that can improve the generalization of a model by preventing overfitting. Num_instances is a hyperparameter used in triplet loss, a popular loss function in metric learning. It specifies the number of positive and negative examples to sample for each anchor example, and increasing this value can improve performance but may also increase the computational cost.

In metric learning tasks, triplet loss is a commonly used loss function that compares the distances between anchor, positive, and negative examples in the embedding space. The margin is a hyperparameter in triplet loss that controls the minimum distance between the anchor and the negative example relative to the distance between the anchor and the positive example. A larger margin can help the model learn more robust embeddings, but it may also result in slower convergence during training. Therefore, selecting the appropriate margin value is crucial for achieving optimal performance in metric learning tasks.

Label-smooth	True
Optimizer	Adam
Learning rate	0.0003
margin	0.4
Num-instances	5
Learning Rate scheduler	Multi-step
Step size	[20 40]

Results ----------
mAP: 53.2%
CMC curve
Rank-1 : 84.3%
Rank-5 : 92.8%
Rank-10 : 95.5%
Rank-20 : 97.2%

6. Results

The figures show the query image and the resulting images obtained from the model.

Figure 3. Query image-1

Figure 4. Visualization of result for Query image-1

Figure 5. Query image-2

Figure 6. Visualization of result for Query image-2

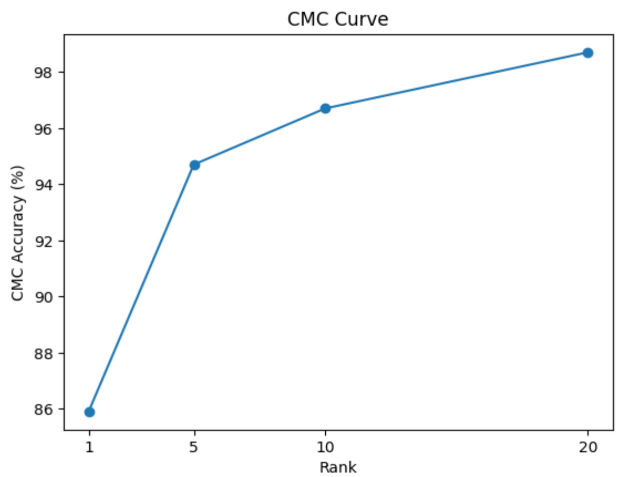

Figure 7. Visualisation of CMC curve

7. Conclusion

The focus of this study was to apply CNNs for vehicle re-identification and explore the effectiveness of different architectures such as MobileNet_v3_small, regnet_x, resnet18. Different hyperparameters were explored and hyperparameters were tuned to find out their influence on accuracy and different data augmentation techniques such as random-erase, color-jitter, and color-augmentation were experimented on the model. The study also examined the impact of changing the architecture of standard Resnet18 and using different activation function (LeakyReLU) and tuning its hyper-parameters. Furthermore, the visualization results demonstrate that the query image matching the resulting gallery set of images from the model.

References

[1] Ye, Mang, et al. "Deep learning for person re-identification: A survey and outlook." IEEE transactions on pattern analysis and machine intelligence 44.6 (2021): 2872-2893.

[2] Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., & Hoi, S. C. (2021). Deep learning for person re-identification: A survey and outlook. IEEE transactions on pattern analysis and machine intelligence, 44(6), 2872-2893.

[3] Deng, Jianhua, et al. "Trends in vehicle re-identification past, present, and future: A comprehensive review." Mathematics 9.24 (2021): 3162.

[4] Deng, J., Hao, Y., Khokhar, M. S., Kumar, R., Cai, J., Kumar, J., & Aftab, M. U. (2021). Trends in vehicle re-identification past, present, and future: A comprehensive review. Mathematics, 9(24), 3162.

[5] Deng, J., Hao, Y., Khokhar, M.S., Kumar, R., Cai, J., Kumar, J. and Aftab, M.U., 2021. Trends in vehicle re-identification past, present, and future: A comprehensive review. Mathematics, 9(24), p.3162.