ResNet is all you need:

Vehicle type classification using ResNet50 Cascor and ResNet50 MLP

Mengqi He
Research School of Computer Science,
Australian National University,
Canberra ACT 0200, Australia
{u6630774@anu.edu.au}

Abstract. transportation is one of the largest segments that can benefit from the AI city. Intelligent transportation system(ITS) can get it deep learning solution to deal with traffic task. To help to build an ITS, we build specific neuron networks to do the vehicle type classification. In this paper, we mainly use the idea from the Residual neural network(ResNet), Cascade neural network (Cascor) to construct the ResNet50-MLP and the ResNet50-Cascor. The classification task is done on the vehicle -x dataset, which have 1362 types of vehicle. The result shows that, with the help of the ResNet, the ResNet50-MLP has a significantly better performance than the ResNet50-Cascor, and it reaches an accuracy of 0.71, while it takes less time than the ResNet50-Cascor.

Keywords: Deep Learning,Object classification,Multi-class classification,Cascor,ResNet

1 Introduction

AI city significantly attracts the industry and the government since it is efficient. While there is already a great amount of the surveillance camera in the city, the government find that the transportation is one of the largest segments that can benefit from this according to the AI city. And the intelligent transportation system(ITS), as a global phenomenon, attracting worldwide interest from transportation experts, the automobile industry, and governments(Figueiredo et al., 2001). Hence, the traffic solution from deep learning is given to have a good contribution to ITS(Veres and Moussa, 2020). Classification of vehicles can have a significant application in ITS which can do analyzing traffic, checking for fraud, tracking targets, and other security applications(Siddiqui, Mammeri and Boukerche, 2015)

For the vehicle Classification task, some of the researchers have already reached a 0.97 accuracy on an 11 categories dataset using a self-design Residual neural network(ResNet) which contain 18 layers (Jung et al., 2017)And in their comparison, we found that other method which uses other variants of ResNet also have state of art performance on this task.

In this paper, we attempt to do a vehicle classification NN to be suitable and robust for the tasks in the ITS. To try to improve the performance of the ResNet on the current NN, we consider the Cascade neural network, which can determine the number of hidden units according to the complexity of the task(Gedeon, 2021). And instead of testing on a dataset that has 11 categories to classify, we want to find a dataset that can provide us enough categories to do a fine-grained classification to be robust for future application. Therefore, we choose the vehicle-x dataset from ANU and NVIDIA, which contains 1362 vehicle types and contains more than 75000 images, which we will discuss in more detail in the dataset section.

2 Method

2.1 ResNet

The main idea of ResNet is to add the direct connection channel into the network, which is a bit similar to Highway Network. The previous network structure is to do a nonlinear transformation on the input, while the Highway Network allows retaining a certain proportion of the output of the previous Network layer. And the idea of ResNet is very similar to that of the Highway Network, allowing raw input information to be passed directly to later layers. In this way, the neural network at this layer does not need to learn the entire output, it just learns the residual output of the previous network. ResNet uses multiple parameter layers to learn the representation of residuals between input and output, rather than using parameter layers to directly try to learn the mapping between input and output as a general CNN network (such as AlexNet /VGG, etc.) does. And this is why this call ResNet.Fig.1 shows the idea of a ResNet block in the ResNet.

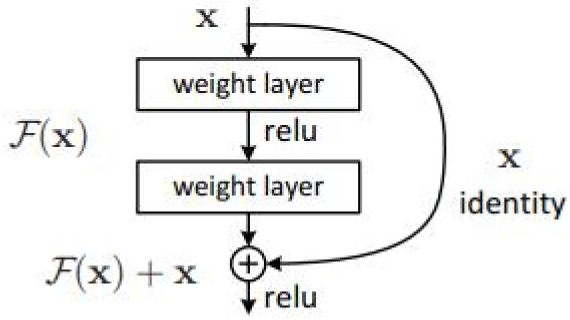

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig.1. ResNet block(He, Zhang, Ren and Sun, 2016)

2.2 ResNet 50-MLP

Resnet 50 is the ResNet which 48 Convolution layers along with a max-pooling and an average pooling layer. It has five stages, of which stage 0 is simple and can be considered as the pre-processing layer of input. The next four stages are composed of bulk and are relatively similar in structure. Stage 1 contains 3 bottlenecks and the remaining 3 stages include 4, 6, and 3 bottlenecks, respectively. The bottleneck layers are used here to reduce the computation complexity, which uses 1x1 convolution. As for why stage 0 doesn’t have bottleneck layers. After that five stages, it should connect with a pooling layer.

However, to let the ResNet50 be suitable for our current task, we choose to connect the last stage of it to the self define multilayer perceptron(MLP). It has just one same dimension hidden layer which has the same dimension as the input layer, and it follows with a leaky_ReLU and a fully connected layer that from 2048 to classification dimension. Finally, connect it to the softmax to do multi-class classification. The overview of the ResNet50-MLP is shown in Fig.2.

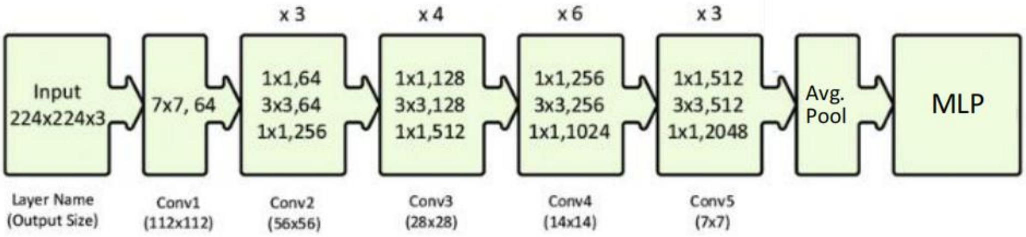


Fig.2. ResNet50-MLP(Mahmood et al., 2020)

The reason why sigmoid is not using here is that for the current multi-class classification task, sigmoid will cause exploding gradient if the input has a large absolute value. The ReLU is also been deprecated since it might let the gradient be 0 and lead to the ‘dead’ neuron when the input value is less than 0.

2.3 ResNet 50-Cascor

Cascor starts without any hidden neuron as a NN and it's all the output layer straightway connected with the input layer. It creates one neuron per time to the network and those hidden neurons created gradually are called the cascade neuron. Those cascade neurons will be connected to the input neurons, the previously hidden neurons, and the output neurons. It will train until the loss of the current network be stable, and then it generates the connection between those candidate hidden neurons and the previous neurons. Once the candidate neuron is added to the current architecture, the weight connected to the input layer will be frozen. And the train will be finalized after meeting the specific requirement. In this way, it can get a better correlation between the candidate neurons and the residual error of the net, which should have a great pair with the ResNet. Fig. 3 shows how the architecture of the Cascor.

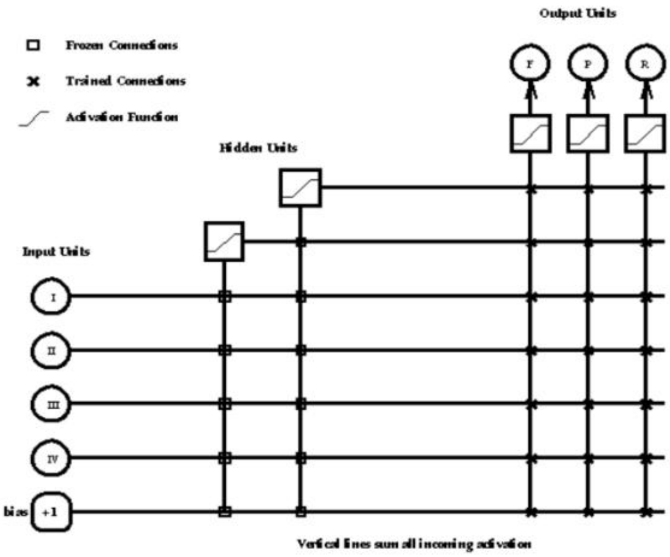


Fig 3. The Architecture of the Cascor(Gedeon, 2021)

To construct the Cascor version of the ResNet50, we remove the last layer of the ResNet 50 and connect it to the Cascor NN directly. The Coscor NN we have original designs for the fashion-mnist dataset. Although the fashion-mnist dataset has 28x28 for each image which has 784 dimensions and can be modified to 2048 dimension according to the ResNet50 's output in the Average pooling layer, it just has 10 classes to do the classification, which is much lower than our needs,1362 classes. Hence, the good performance of Cascor on the fashion-mnist of it doesn't mean it will not be granted to have a result on our task as well. Considered we have a ResNet50 before connected to the Cascor, we decided to increase the original classified classes of this Cascor to 1362 and connect it to the ResNet 50 we mentioned above. But due to the time complexity of the cascor is significant, we choose to connect it to a 5 neurons Cascor and call it ResNet50-Cascor. Fig 4 shows the architecture of the ResNet50-Cascor.

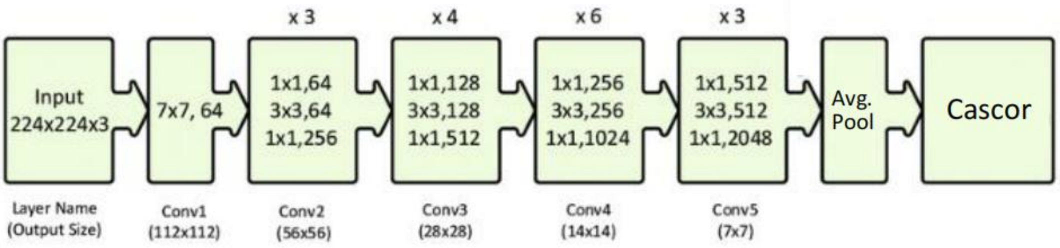


Fig 4. ResNet50-Cascor(Mahmood et al., 2020)

3 Dataset

We choose the vehicle-x dataset which is originally generated from the Australian National University and NVIDIA, It is used in the 2021 CVPR AI city Challenge and Alice Challenge. They use the graph engine to automatically process the image which directly edits the source domain of the content in the image and this reduces the difference between the practical images and the image they produced(Yao et al., 2021). This dataset contains 45438 images for training,15142 images for testing, and 14936 images for validation, where all the images are jpg files with a size of 256x256. And the file name indicates the labeling format for the dataset, which is "id_cam_num.jpg". For instance, take the first image in the train set as the example,’00001_c001_2723.jpg’ means it is an image for a vehicle that id is 00001, captured by the 001 camera and counting as the 2723rd images.

To demonstrate the statistic distribution of the dataset, we plot the histogram and we can see that it looks like a normal distribution, most of the class in train set contains 30 - 35 images, while most of the class in the test set contains 10 - 14 images and the validation set have the similar distribution as the test set. Those histograms are shown in Fig 5.

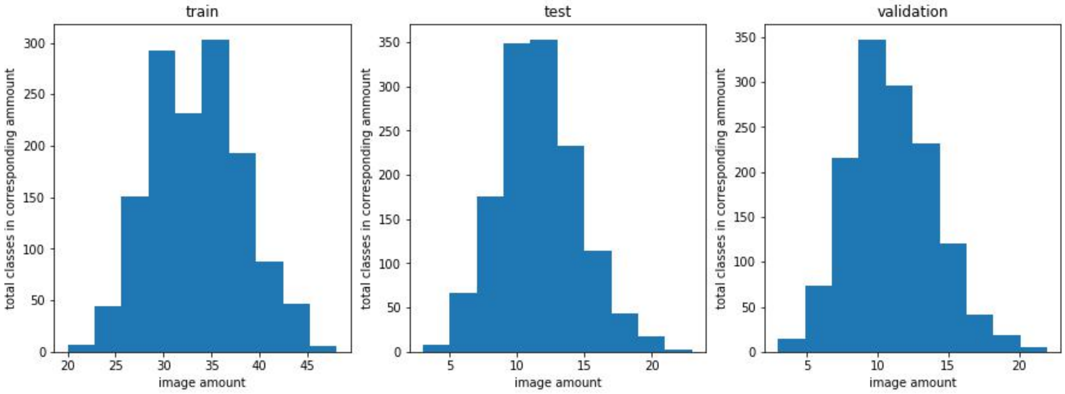


Fig.5. Data distribution in the vehicle-x dataset

4 Experiment

4.1 Data Pre-processing

The size of the original image in the vehicle x is 256x256, but this can not work on our two ResNet-based networks, so we resize all of the images to 224x224. Although we don’t need the validation set in our experiment, to prepare for the next step of work, we apply the first 49 layers of the ResNet50 on all of these three sets and save the result in the CSV file. As for the target label, we set up it by the extract them from the images’ names and then add them to the last column of those three CSV files. Before the data are input into the model, we do Z-score standardization to ensure the convergence speed and performance.

4.2 Metrics

Accuracy is one of the metrics we choose to evaluate the performance of the models.TP is the number of the true positive result and the TN is the true negative result. As for the FP and the FN, the F represents the false, the remaining part has the same meaning as before. We can also consider the denominator to directly be the total number of the image in that epoch.And the loss here we use is the cross-entropy loss, which is suitable for the multi-class classification.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

4.3 Experiment process

In the Experiment, since apply this complex task on this vehicle-x dataset needs significant computing resources and both of these two networks are base on the ResNet 50, we change our strategy to apply the Model to the data. We use to save the computing result of the first 49 layers of the ResNet50 and then apply the following MLP and the Cascor layer rather than directly including the MLP and the Cascor layer.

Hence, we first set up cross-entropy loss as our loss and the Adaptive Moment Estimation(Adam) as our optimizer since it combines the Momentum and RMSProp. And then in both the ResNet50-MLP and the ResNet50-Cascor, do the data pre-processing. After that, we apply the following MLP and Casor layer respectively on those results. Finally, we display the loss and the accuracy per 10 epochs in both of them.

For the ResNet50-Cascor, due to the limitation of the computation resource we have, we restrict the epoch of maximizing correlation to 4 and the max neuron to 5. Since the Cascor layer just has 5 neurons, to have a better comparison, for the ResNet50-MLP, we set both ResNet-MLP-2048 and ResNet-MLP-5 which respectively have 2048 hidden neurons and 5 hidden neurons.

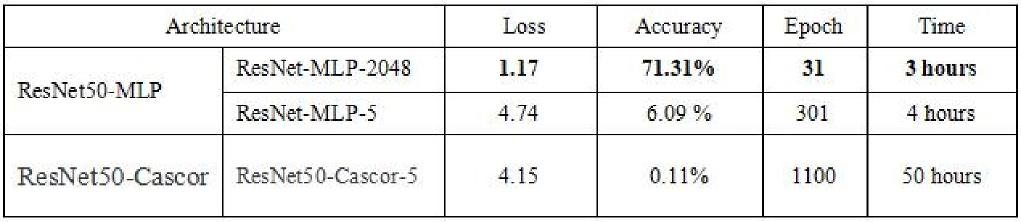
5 Results and Discussion

The result of both the ResNet50-MLP and the ResNet50-Cascor are list in Table 1, where all the results are round up to 2 decimal places. The ResNet50-MLP has ResNet-MLP-2048 and ResNet-MLP-5 while the ResNet50-Cascor just have 5 neurons version ResNet50-Cascor-5 due to the limitation of the computation resource we have.The Loss and the Accuracy we mention in Table 1 are the test version instead of the train, and the Epoch is the training epoch while the time is the time excluding the data pre-processing.

As we can see in Table 1, we can see that the ResNet-MLP-2048 have the best performance on this specific task in all of the ResNet50 based neuron network we have a test. It has 1.17 loss and 71.31% accuracy with just 31 epoch and just 3 hours of training. As for the ResNet50-Cascor, while it just contains 5 hidden neurons,it has 0.11% accuracy while take over two days to do the computation, although the loss of if is small enough as 4.15. By comparison. although the ResNet50-MLP-5 just has 5 hidden layers as well, after 301 times of training, it has an accuracy that reaches to 6.09%, and the loss of it is 4.74.

Resnet50-Cascor has poor performance in this experiment, which might be cause by two main reasons. Firstly,the max correlation compute time we give is just 4 times,it might need far more than 4 time to do the correlation. Secondly,the activation function hyperbolic tangent might not suitable for the current data,while it requires exponentiation operation, and the calculation cost is high; Again, the gradient disappears, because it goes to 0 on both sides.

Table 1.the performance for the different Architecture on Vehicle-x



Architecture		Loss	Accuracy	Epoch	Time
ResNet50-MLP	ResNet-MLP-2048	**1.17**	**71.31%**	**31**	**3 hours**
	ResNet-MLP-5	4.74	6.09 %	301	4 hours
ResNet50-Cascor	ResNet50-Cascor-5	4.15	0.11%	1100	50 hours

Hence, we can explore more detail from those two ResNet-MLP models.As shown in Fig 6, the train accuracy and the the test accuracy of the ResNet-MLP-2048 increase the fastest in the period from epoch 1 to epoch 11,reach about 100% of the training accuracy and about 70% of test accuracy.And it reach 100% train accuracy and 71.31% testing accuracy at epoch 31.As for the train loss,it start with 139.58 and reduce to 0.13 at the epoch 31.And the ResNet50-MLP-5 have a similar trend,although it just reach to a train accuracy of 9.84% and a test accuracy of 6.09%.As for the training loss,it start with 166.7 and finally reduce to 97.56.The performance of the ResNet50-MLP-5 is not that good,but considered it as a 1362 class classification and we just use 5 hidden neuron in the MLP layer,it is not bad as a low cost choice.And we actually also test that once we increase the amount of the hidden neuron in its MLP layer,the performance will be improved according to the amount of the neuron we add in here.

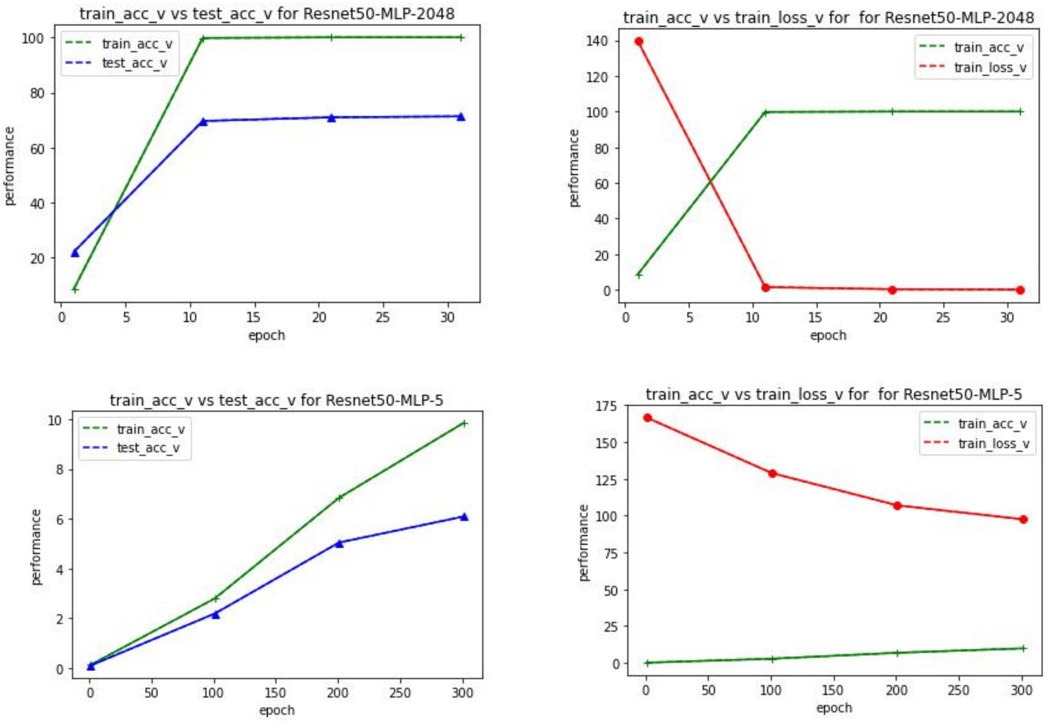
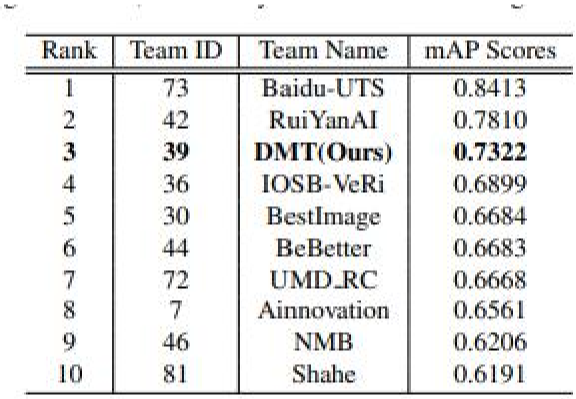


Fig.6. broken line graph of for the ResNet-MLP

By comparison, we find that there are already some of the researchers who have done this task using other models in the competition, and they evaluate the result through the mAP. And they list the top 10 results those teams get in that competition, where the highest model from the Baidu-UTS reaches about 0.8413 for mAP on this task. Table 2 below is that list.

Table 2. Competition results of AICITY20 Track2.(He et al., 2021)



Rank	Team ID	Team Name	mAP Scores
1	73	Baidu-UTS	0.8413
2	42	RuiYanAI	0.7810
3	**39**	**DMT(Ours)**	**0.7322**
4	36	IOSB-VeRi	0.6899
5	30	BestImage	0.6684
6	44	BeBetter	0.6683
7	72	UMD_RC	0.6668
8	7	Ainnovation	0.6561
9	46	NMB	0.6206
10	81	Shahe	0.6191

6 Conclusion and Future Work

After introducing the Resnet and the Cascor in this article, we combine them as Resnet50-Cascor tests its classification performance on the vehicle-x dataset. And to have a comparison, we use a Resnet50-MLP which combines the Resnet 50 and the simple MLP. However, the testing result gives us an unexpected and surprising result, the Resnet50-Cascor has really bad performance and costs much more time than the ResNet-MLP, about 10 times of it, and it even has less accuracy than the Resnet50-MLP-5 which has 5 neurons as well. But that might be caused by the incorrect setup of the Resnet50-Cascor such as the max correlation times and the activation function. As for the Resnet50-MLP, it reaches 71.31% accuracy and it reaches a great level even compare to the top 10 teams in the AI city 2021 competition(although they use the mAP as the metric).

In the future, there are three things we plan to do to extend our work. Firstly, we will try to set up the

ResNet50-Cascor in a better way to get better performance on that model. Secondly, although our ResNet50-MLP model does have a good performance on this vehicle-x dataset, we want to do some adjustments to it and test this model on another dataset in the related area to check if this model is robust on this kind of fine-grained vehicle classification task. Last but not least, we get more ideas from other state-of-art models in AICity 2020 competition such as the model generated by the Baidu-UTS group to let it have a better performance on this task.

Reference

1. Figueiredo, L., Jesus, I., Machado, J., Ferreira, J. and Martins de Carvalho, J., 2001. Towards the development of intelligent transportation systems. ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585), [online] Available at: <https://www.researchgate.net/publication/224073158_Towards_the_Development_of_Intelligent_Transportation_Systems> [Accessed 31 May 2021].
2. Gedeon, T., 2021. Cascade correlation NNs. [online] ANU. Available at: <https://wattlecourses.anu.edu.au/pluginfile.php/2558398/mod_resource/content/6/NN9_cascor%2Bcasper.pdf> [Accessed 27 April 2021].
3. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), [online] Available at: <https://ieeexplore.ieee.org/document/7780459> [Accessed 31 May 2021].
4. He, S., Luo, H., Chen, W., Zhang, M., Zhang, Y., Li, H., Wang, F. and Jiang, W., 2021. Multi-Domain Learning and Identity Mining for Vehicle Re-Identification. [online] Arxiv.org. Available at: <https://arxiv.org/pdf/2004.10547v2.pdf> [Accessed 31 May 2021].
5. Jung, H., Choi, M., Jung, J., Lee, J., Kwon, S. and Jung, W., 2017. ResNet-Based Vehicle Classification and Localization in Traffic Surveillance Systems. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/document/8014863> [Accessed 31 May 2021].
6. Mahmood, A., Ospina, A., Bennamoun, M., An, S., Sohel, F., Boussaid, F., Hovey, R., Fisher, R. and Kendrick, G., 2020. Automatic Hierarchical Classification of Kelps Using Deep Residual Features. Sensors, [online] 20(2), p.447.
7. Siddiqui, A., Mammeri, A. and Boukerche, A., 2015. Towards Efficient Vehicle Classification in Intelligent Transportation Systems. Proceedings of the 5th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, [online] Available at: <https://www.researchgate.net/publication/301464073_Towards_Efficient_Vehicle_Classification_in_Intelligent_Transportation_Systems> [Accessed 31 May 2021].
8. Veres, M. and Moussa, M., 2020. Deep Learning for Intelligent Transportation Systems: A Survey of Emerging Trends. IEEE Transactions on Intelligent Transportation Systems, [online] 21(8), pp.3152-3168. Available at: <https://ieeexplore.ieee.org/abstract/document/8771378> [Accessed 31 May 2021].
9. Yao, Y., Zheng, L., Yang, X., Naphade, M. and Gedeon, T., 2021. Simulating Content Consistent Vehicle Datasets with Attribute Descent. [online] arXiv.org. Available at: <https://arxiv.org/abs/1912.08855> [Accessed 31 May 2021].