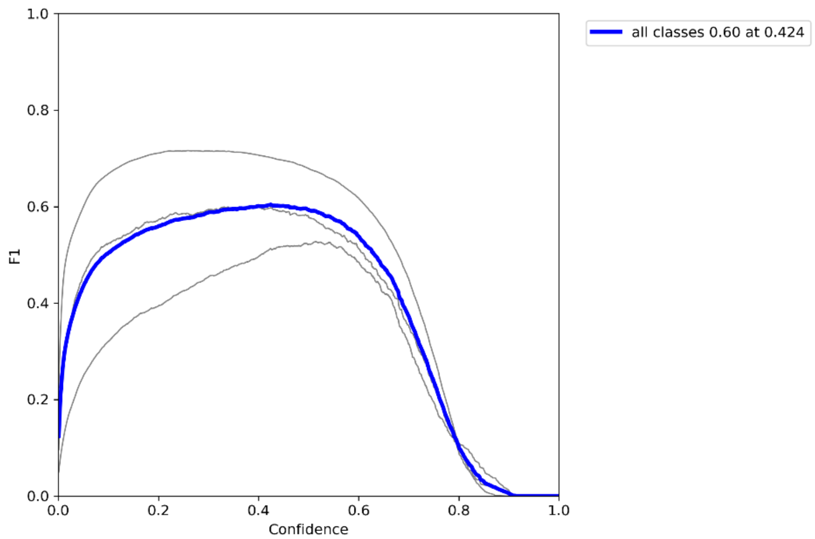
CV_HW4 Handling Domain Shift 107081028 常安彥

Q1. Train Valid Split

1. Original implement: mAP = 0.6, F1 = 0.6

   Simply split data according to the index, which means training set contains 40 images of camera-173 and 140 images of camera-398, and valid set is composed of 20 images of camera-398. **Validation was based on camera-398.**

Camera	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
170							
	all	1200	1761	0.482	0.51	0.456	0.274
	car	1200	1504	0.801	0.651	0.744	0.409
	bus	1200	210	0.581	0.495	0.548	0.368
	truck	1200	47	0.063	0.383	0.076	0.045
495							
	all	1200	2227	0.631	0.67	0.651	0.375
	car	1200	1950	0.686	0.687	0.675	0.373
	bus	1200	55	0.888	0.727	0.81	0.529
	truck	1200	222	0.32	0.595	0.468	0.222
410							
	all	1200	2331	0.524	0.614	0.492	0.285
	car	1200	2005	0.85	0.629	0.725	0.418
	bus	1200	15	0.139	0.533	0.174	0.131
	truck	1200	311	0.583	0.678	0.577	0.307
511							
	all	1200	2294	0.401	0.57	0.427	0.257
	car	1200	2077	0.78	0.56	0.695	0.374
	bus	1200	86	0.29	0.593	0.336	0.235
	truck	1200	131	0.132	0.557	0.249	0.163
398							
	all	1200	2353	0.719	0.682	0.701	0.469
	car	1200	2056	0.745	0.722	0.721	0.417
	bus	1200	104	0.878	0.556	0.705	0.504
	truck	1200	193	0.534	0.767	0.678	0.488
173							
	all	1200	1991	0.808	0.76	0.789	0.513
	car	1200	1680	0.925	0.709	0.865	0.532
	bus	1200	171	0.853	0.877	0.863	0.625
	truck	1200	140	0.646	0.692	0.639	0.383
ALL							
	all	1200	12957	0.647	0.589	0.6	0.37
	car	1200	11272	0.819	0.613	0.722	0.41
	bus	1200	641	0.687	0.53	0.595	0.416
	truck	1200	1044	0.435	0.624	0.485	0.282



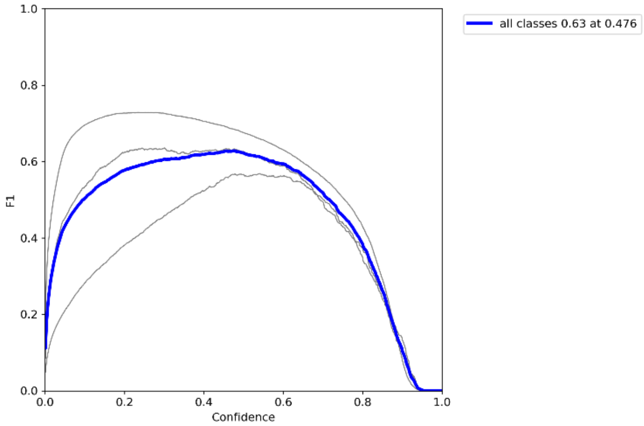
2. Separate Respectively: mAP = 0.66, F1 = 0.63

First, I separate into training and valid data on both camera-173 and 398. Therefore, training set contains 36 images of c-173 and 144 images of c-398, while valid set contains 4 images of c-173 and 16 images of c-398. In this case, validation was based on both cameras.

According to the result below, the mAP of camera-173 increased. Furthermore, the performance of the whole dataset significantly increased. Although I assumed the recall of camera-173 would have been better, it didn't. Based on the high performance of the 'car' class with other classes performing badly, I thought that this issue could be dealt as label imbalance in following questions.

Camera	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
170							
	all	1200	1761	0.516	0.535	0.512	0.334
	car	1200	1504	0.855	0.668	0.784	0.45
	bus	1200	210	0.636	0.598	0.643	0.477
	truck	1200	47	0.059	0.34	0.109	0.075
495							
	all	1200	2227	0.609	0.763	0.682	0.444
	car	1200	1950	0.644	0.737	0.664	0.376
	bus	1200	55	0.867	0.831	0.843	0.667
	truck	1200	222	0.318	0.721	0.539	0.288
410							
	all	1200	2331	0.594	0.745	0.697	0.478
	car	1200	2005	0.847	0.644	0.763	0.47
	bus	1200	15	0.324	0.8	0.576	0.512
	truck	1200	311	0.611	0.791	0.751	0.451
511							
	all	1200	2294	0.59	0.477	0.546	0.353
	car	1200	2077	0.882	0.446	0.735	0.422
	bus	1200	86	0.553	0.535	0.604	0.455
	truck	1200	131	0.333	0.45	0.298	0.183
398							
	all	1200	2353	0.722	0.636	0.711	0.487
	car	1200	2056	0.751	0.676	0.73	0.434
	bus	1200	104	0.887	0.529	0.774	0.581
	truck	1200	193	0.528	0.705	0.629	0.445
173							
	all	1200	1991	0.793	0.72	0.806	0.553
	car	1200	1680	0.946	0.687	0.895	0.574
	bus	1200	171	0.857	0.804	0.873	0.66
	truck	1200	140	0.576	0.67	0.649	0.424
ALL							
	all	1200	12957	0.702	0.584	0.663	0.435
	car	1200	11272	0.83	0.583	0.743	0.442
	bus	1200	641	0.757	0.544	0.673	0.512
	truck	1200	1044	0.519	0.624	0.573	0.35

F1 score: 0.63



3. Camera Balance: mAP = 0.66, F1 = 0.61

Next, I tried to balance the data size of each camera in order to balance the weights and importance of images of each camera. Therefore, I took the same amount of data from camera-173 and camera –398, both 36 as training and 4 as validation.

The result of camera-173 increased a little, but the overall performance did not go well. Since the size of the data shrinks, the performance went bad.

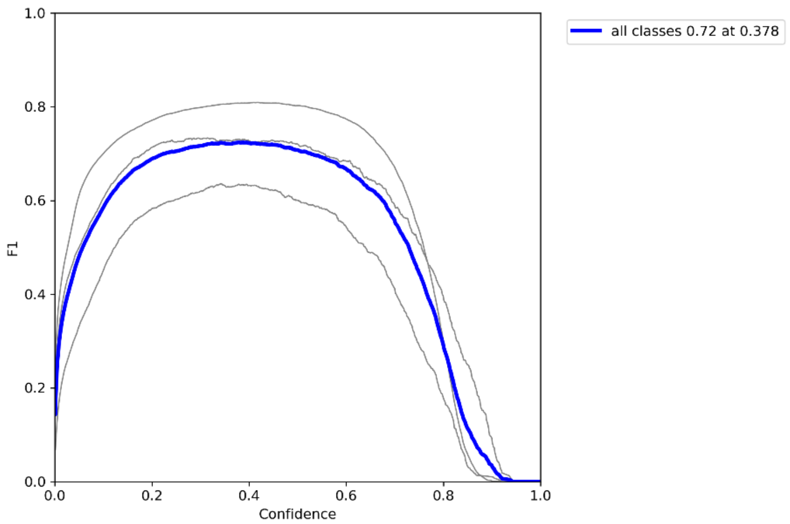
Camera	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
170							
	all	1200	1761	0.491	0.597	0.518	0.338
	car	1200	1504	0.823	0.709	0.796	0.459
	bus	1200	210	0.592	0.657	0.642	0.473
	truck	1200	47	0.057	0.426	0.117	0.082
495							
	all	1200	2227	0.634	0.683	0.67	0.434
	car	1200	1950	0.685	0.697	0.672	0.384
	bus	1200	55	0.84	0.673	0.828	0.644
	truck	1200	222	0.377	0.68	0.509	0.273
410							
	all	1200	2331	0.594	0.736	0.696	0.471
	car	1200	2005	0.849	0.63	0.767	0.474
	bus	1200	15	0.315	0.8	0.572	0.493
	truck	1200	311	0.616	0.778	0.749	0.447
511							
	all	1200	2294	0.603	0.455	0.544	0.35
	car	1200	2077	0.893	0.421	0.735	0.426
	bus	1200	86	0.58	0.523	0.6	0.438
	truck	1200	131	0.336	0.42	0.296	0.185
398							
	all	1200	2353	0.692	0.651	0.716	0.487
	car	1200	2056	0.739	0.694	0.743	0.45
	bus	1200	104	0.854	0.529	0.776	0.562
	truck	1200	193	0.482	0.731	0.629	0.45
173							
	all	1200	1991	0.792	0.717	0.818	0.57
	car	1200	1680	0.949	0.679	0.9	0.582
	bus	1200	171	0.846	0.807	0.876	0.67
	truck	1200	140	0.581	0.664	0.677	0.458
ALL							
	all	1200	12957	0.699	0.574	0.664	0.436
	car	1200	11272	0.835	0.571	0.751	0.451
	bus	1200	641	0.748	0.538	0.669	0.505
	truck	1200	1044	0.513	0.613	0.572	0.353

Q2. Select Images

1. Random Sampling: mAP = 0.729, F1 = 0.72

   Random Sample from the whole dataset which is composed of images from 6 different cameras. The result is much better than Q1, because the training data contains images from all cameras, which provide images for testing.

Camera	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
170							
	all	1200	1761	0.69	0.618	0.636	0.407
	car	1200	1504	0.885	0.82	0.89	0.499
	bus	1200	210	0.696	0.843	0.789	0.567
	truck	1200	47	0.489	0.191	0.229	0.155
495							
	all	1200	2227	0.672	0.754	0.699	0.426
	car	1200	1950	0.697	0.846	0.734	0.423
	bus	1200	55	0.751	0.767	0.842	0.579
	truck	1200	222	0.569	0.649	0.52	0.276
410							
	all	1200	2331	0.641	0.806	0.76	0.493
	car	1200	2005	0.809	0.79	0.823	0.477
	bus	1200	15	0.234	0.933	0.647	0.519
	truck	1200	311	0.881	0.694	0.811	0.483
511							
	all	1200	2294	0.682	0.744	0.737	0.493
	car	1200	2077	0.782	0.88	0.89	0.504
	bus	1200	86	0.757	0.94	0.902	0.683
	truck	1200	131	0.509	0.412	0.418	0.291
398							
	all	1200	2353	0.706	0.694	0.72	0.488
	car	1200	2056	0.722	0.78	0.767	0.466
	bus	1200	104	0.833	0.527	0.701	0.509
	truck	1200	193	0.563	0.777	0.692	0.489
173							
	all	1200	1991	0.862	0.722	0.837	0.56
	car	1200	1680	0.898	0.851	0.924	0.576
	bus	1200	171	0.879	0.765	0.852	0.614
	truck	1200	140	0.808	0.55	0.733	0.49
ALL							
	all	1200	12957	0.706	0.743	0.729	0.468
	car	1200	11272	0.781	0.836	0.822	0.481
	bus	1200	641	0.703	0.76	0.759	0.546
	truck	1200	1044	0.635	0.634	0.607	0.376

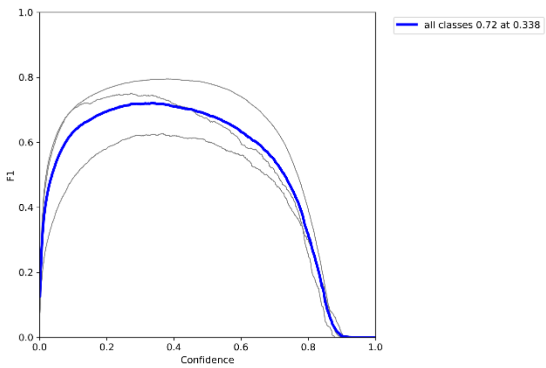


2. Elimination of small bounding boxes & little information: mAP = 0.732, F1= 0.72

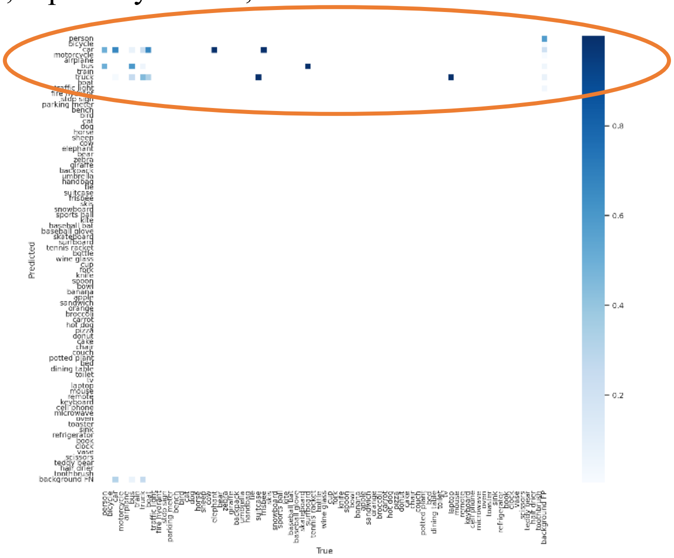
First of all, I figured out that there were some small bounding boxes in several images, which might be too small to provide enough information for the model to learn. So, I analyzed the area of all bounding boxes and calculated the standard deviation. Based on the analyzing, I set a threshold, 0.004, and accumulated the number of labels of each image that had a bounding box with area larger than the threshold.

Images with more big enough boxes had the higher priority to be selected, which I assumed as images providing more useful information. Then, based on the sorted image paths, split into training and valid set with the method implemented on Q1.

Camera	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
170							
	all	1200	1761	0.64	0.627	0.64	0.416
	car	1200	1504	0.892	0.812	0.888	0.531
	bus	1200	210	0.718	0.835	0.803	0.563
	truck	1200	47	0.312	0.234	0.228	0.155
495							
	all	1200	2227	0.662	0.762	0.68	0.397
	car	1200	1950	0.67	0.857	0.722	0.427
	bus	1200	55	0.752	0.771	0.787	0.489
	truck	1200	222	0.563	0.658	0.532	0.276
410							
	all	1200	2331	0.697	0.765	0.801	0.534
	car	1200	2005	0.829	0.775	0.82	0.486
	bus	1200	15	0.457	0.8	0.783	0.621
	truck	1200	311	0.806	0.72	0.798	0.494
511							
	all	1200	2294	0.805	0.697	0.758	0.514
	car	1200	2077	0.856	0.84	0.891	0.522
	bus	1200	86	0.856	0.901	0.935	0.713
	truck	1200	131	0.703	0.351	0.449	0.306
398							
	all	1200	2353	0.682	0.662	0.704	0.47
	car	1200	2056	0.703	0.778	0.754	0.467
	bus	1200	104	0.827	0.461	0.689	0.502
	truck	1200	193	0.516	0.746	0.668	0.44
173							
	all	1200	1991	0.786	0.801	0.827	0.544
	car	1200	1680	0.864	0.883	0.926	0.594
	bus	1200	171	0.803	0.784	0.818	0.576
	truck	1200	140	0.691	0.736	0.736	0.461
ALL							
	all	1200	12957	0.707	0.737	0.732	0.469
	car	1200	11272	0.763	0.827	0.805	0.484
	bus	1200	641	0.755	0.736	0.77	0.543
	truck	1200	1044	0.602	0.649	0.62	0.38



According to the overall result above, the performance only increased a little compared with random sampling. Then, based on the confusion matrix below, a huge proportion of other labels were mis-predicted into the three major labels, car, bus and truck, especially car. So, I assumed the issue as data imbalance.

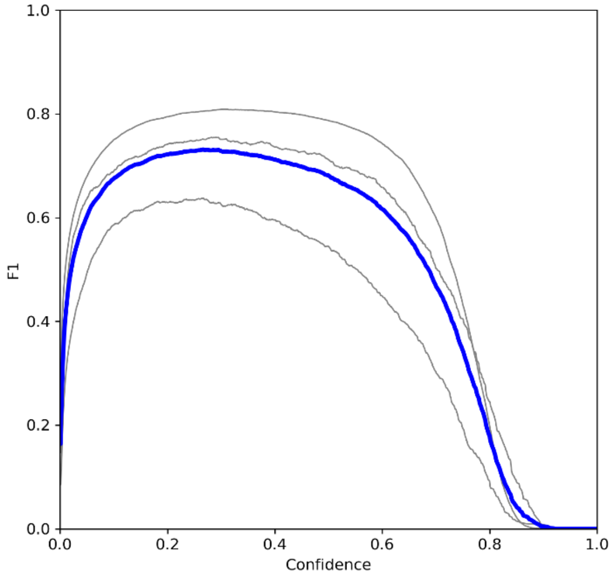


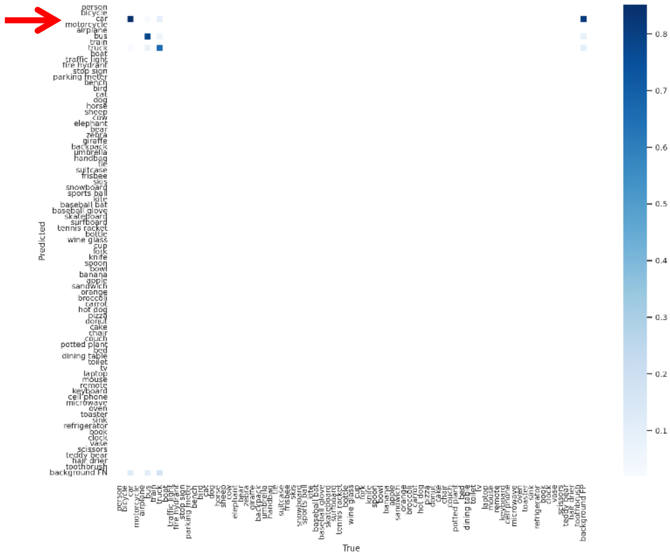
3. Emphasize ‘Bus”: mAP = 0.738, F1 = 0.73

After implementing emphasis on different labels, I figured out that emphasizing the class, ‘bus,’ may help a lot on classifying. Therefore, I modified the sorting method, and made images with ‘bus’ labels higher priority. More images with bus on them would be included to the training and valid data.

Camera	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
170							
	all	1200	1761	0.698	0.663	0.66	0.412
	car	1200	1504	0.844	0.847	0.893	0.517
	bus	1200	210	0.7	0.91	0.809	0.546
	truck	1200	47	0.55	0.234	0.277	0.174
495							
	all	1200	2227	0.599	0.787	0.683	0.4
	car	1200	1950	0.619	0.884	0.717	0.414
	bus	1200	55	0.671	0.891	0.824	0.543
	truck	1200	222	0.508	0.585	0.506	0.243
410							
	all	1200	2331	0.639	0.803	0.731	0.452
	car	1200	2005	0.826	0.788	0.829	0.485
	bus	1200	15	0.209	0.933	0.519	0.365
	truck	1200	311	0.881	0.689	0.846	0.506
511							
	all	1200	2294	0.719	0.645	0.721	0.464
	car	1200	2077	0.839	0.87	0.908	0.516
	bus	1200	86	0.773	0.674	0.839	0.613
	truck	1200	131	0.545	0.389	0.415	0.265
398							
	all	1200	2353	0.691	0.741	0.748	0.486
	car	1200	2056	0.691	0.825	0.775	0.467
	bus	1200	104	0.81	0.616	0.76	0.516
	truck	1200	193	0.572	0.782	0.708	0.476
173							
	all	1200	1991	0.841	0.79	0.861	0.559
	car	1200	1680	0.864	0.887	0.933	0.587
	bus	1200	171	0.829	0.877	0.884	0.591
	truck	1200	140	0.829	0.607	0.767	0.5
ALL							
	all	1200	12957	0.711	0.756	0.738	0.459
	car	1200	11272	0.769	0.846	0.818	0.481
	bus	1200	641	0.703	0.807	0.766	0.518
	truck	1200	1044	0.661	0.615	0.631	0.377

According to the result, the performance was the best so far, and according to the confusion matrix, the misprediction issue turned well.





4. Conclusion

	Random	Elimination	Bus
mAP	0.729	0.732	0.738
F1	0.72	0.72	0.73
Precision	0.706	0.707	0.711
Recall	0.743	0.737	0.756

The precision of simply elimination method was the highest because it was the dataset composed of the most labels. On the other hand, the bus emphasizing method outperformed on the other scores.

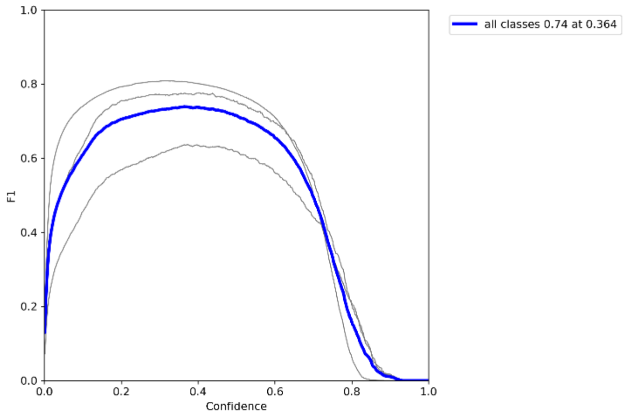
Q3. Unlabeled Images

1. Pseudo label: mAP = 0.747, F1 = 0.73

First, I predicted the labels of the 1200 unlabeled images of Q3. With the 'detect' python file and the best model trained on Q2, I created the txt labels and boxes of the whole Q3 dataset. Then, I trained the Q3 model based on the 200 selected images and these 1200 Q3 images.

According to the result below, the performance significantly increased, because the dataset augmented a lot. Semi-supervise learning can gather dataset with less cost and enhance the robustness of the model.

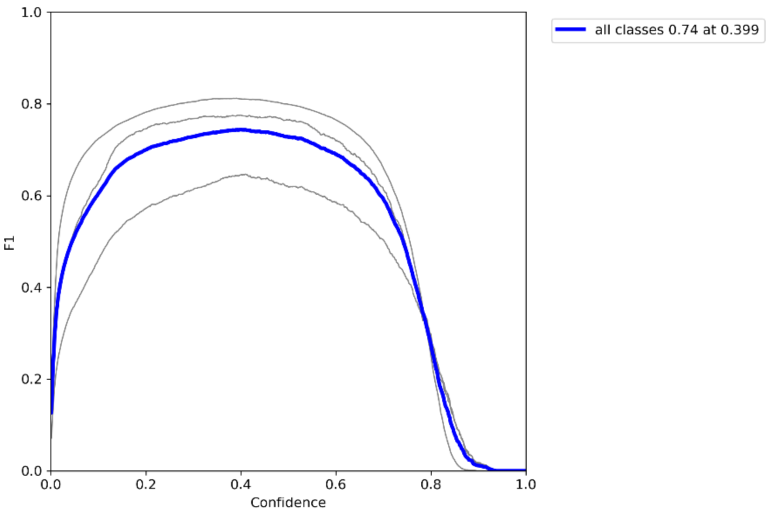
Camera	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
170							
	all	1200	1761	0.605	0.697	0.636	0.387
	car	1200	1504	0.842	0.861	0.897	0.517
	bus	1200	210	0.655	0.91	0.808	0.525
	truck	1200	47	0.319	0.319	0.203	0.12
495							
	all	1200	2227	0.682	0.768	0.716	0.394
	car	1200	1950	0.706	0.789	0.736	0.409
	bus	1200	55	0.794	0.855	0.87	0.529
	truck	1200	222	0.546	0.662	0.543	0.244
410							
	all	1200	2331	0.697	0.78	0.727	0.44
	car	1200	2005	0.885	0.691	0.817	0.46
	bus	1200	15	0.304	0.933	0.499	0.352
	truck	1200	311	0.903	0.717	0.866	0.509
511							
	all	1200	2294	0.826	0.666	0.748	0.471
	car	1200	2077	0.938	0.777	0.899	0.505
	bus	1200	86	0.91	0.907	0.951	0.666
	truck	1200	131	0.63	0.313	0.395	0.241
398							
	all	1200	2353	0.72	0.666	0.705	0.442
	car	1200	2056	0.735	0.747	0.753	0.437
	bus	1200	104	0.874	0.538	0.725	0.491
	truck	1200	193	0.551	0.713	0.638	0.398
173							
	all	1200	1991	0.773	0.843	0.851	0.527
	car	1200	1680	0.836	0.883	0.929	0.572
	bus	1200	171	0.848	0.883	0.889	0.597
	truck	1200	140	0.636	0.762	0.735	0.413
ALL							
	all	1200	12957	0.732	0.748	0.747	0.449
	car	1200	11272	0.82	0.793	0.819	0.47
	bus	1200	641	0.751	0.8	0.791	0.521
	truck	1200	1044	0.623	0.649	0.632	0.356



2. Transfer Learning: mAP = 0.752, F1=0.74

Next, I implemented transfer learning based on the weight trained on Q2. In this case, I could straightly enhance the robustness of the Q2 best model with these Q3 images, and more quickly train a better model.

Camera	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
170							
	all	1200	1761	0.675	0.629	0.638	0.42
	car	1200	1504	0.916	0.808	0.896	0.526
	bus	1200	210	0.725	0.868	0.812	0.591
	truck	1200	47	0.384	0.213	0.206	0.144
495							
	all	1200	2227	0.676	0.765	0.713	0.448
	car	1200	1950	0.693	0.825	0.74	0.438
	bus	1200	55	0.779	0.836	0.86	0.616
	truck	1200	222	0.555	0.634	0.539	0.291
410							
	all	1200	2331	0.758	0.67	0.736	0.495
	car	1200	2005	0.902	0.655	0.827	0.496
	bus	1200	15	0.457	0.733	0.515	0.417
	truck	1200	311	0.915	0.621	0.868	0.573
511							
	all	1200	2294	0.843	0.674	0.757	0.523
	car	1200	2077	0.917	0.791	0.902	0.519
	bus	1200	86	0.919	0.918	0.957	0.754
	truck	1200	131	0.693	0.313	0.411	0.298
398							
	all	1200	2353	0.742	0.646	0.714	0.491
	car	1200	2056	0.74	0.74	0.764	0.475
	bus	1200	104	0.887	0.529	0.724	0.532
	truck	1200	193	0.6	0.668	0.655	0.467
173							
	all	1200	1991	0.835	0.793	0.859	0.6
	car	1200	1680	0.892	0.871	0.93	0.598
	bus	1200	171	0.87	0.864	0.888	0.686
	truck	1200	140	0.744	0.643	0.759	0.517
ALL							
	all	1200	12957	0.733	0.755	0.752	0.502
	car	1200	11272	0.806	0.815	0.823	0.495
	bus	1200	641	0.751	0.801	0.792	0.591
	truck	1200	1044	0.641	0.649	0.64	0.419



3. Freeze Backbone: mAP = 0.748, F1=0.74

Finally, I implemented backbone freezing when transfer learning, and tried freezing 1, 2 and 3 layers. When implementing transfer learning, backbone freezing could decrease the amount of computation and time consumption.

According to the result, although the training time did decrease, the model couldn't outperform the previous one.

Camera	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
170							
	all	1200	1761	0.685	0.637	0.63	0.407
	car	1200	1504	0.869	0.843	0.891	0.493
	bus	1200	210	0.71	0.876	0.783	0.574
	truck	1200	47	0.474	0.191	0.215	0.154
495							
	all	1200	2227	0.743	0.702	0.724	0.444
	car	1200	1950	0.72	0.78	0.747	0.435
	bus	1200	55	0.909	0.745	0.883	0.628
	truck	1200	222	0.6	0.581	0.541	0.268
410							
	all	1200	2331	0.75	0.774	0.763	0.498
	car	1200	2005	0.926	0.723	0.863	0.505
	bus	1200	15	0.397	0.933	0.58	0.475
	truck	1200	311	0.928	0.666	0.847	0.513
511							
	all	1200	2294	0.809	0.703	0.759	0.516
	car	1200	2077	0.829	0.825	0.878	0.477
	bus	1200	86	0.86	0.919	0.946	0.767
	truck	1200	131	0.738	0.366	0.452	0.303
398							
	all	1200	2353	0.691	0.703	0.72	0.49
	car	1200	2056	0.705	0.805	0.755	0.457
	bus	1200	104	0.879	0.49	0.707	0.519
	truck	1200	193	0.49	0.813	0.698	0.493
173							
	all	1200	1991	0.794	0.827	0.846	0.583
	car	1200	1680	0.857	0.902	0.934	0.588
	bus	1200	171	0.86	0.864	0.871	0.659
	truck	1200	140	0.667	0.714	0.734	0.502
ALL							
	all	1200	12957	0.745	0.735	0.748	0.486
	car	1200	11272	0.813	0.808	0.836	0.487
	bus	1200	641	0.778	0.761	0.78	0.583
	truck	1200	1044	0.644	0.637	0.63	0.388