Master in Computer Vision *Barcelona*

Image Classification

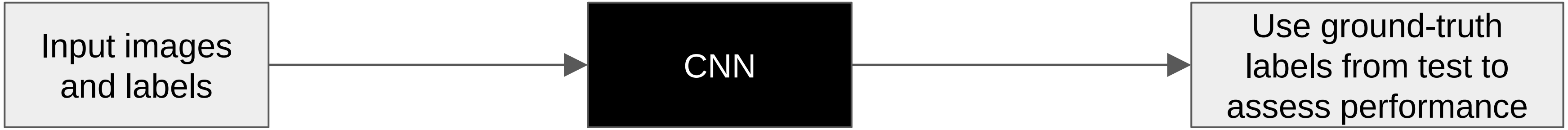
Ramon Baldrich

Artificial Intelligence in Computer Vision

Machine learning for image classification:

Data driven methods: Deep Convolutional Networks: 3 sessions

- From hand-crafted to learnt features
- Fine tuning of pre-trained CNNs
- Training a CNN from scratch

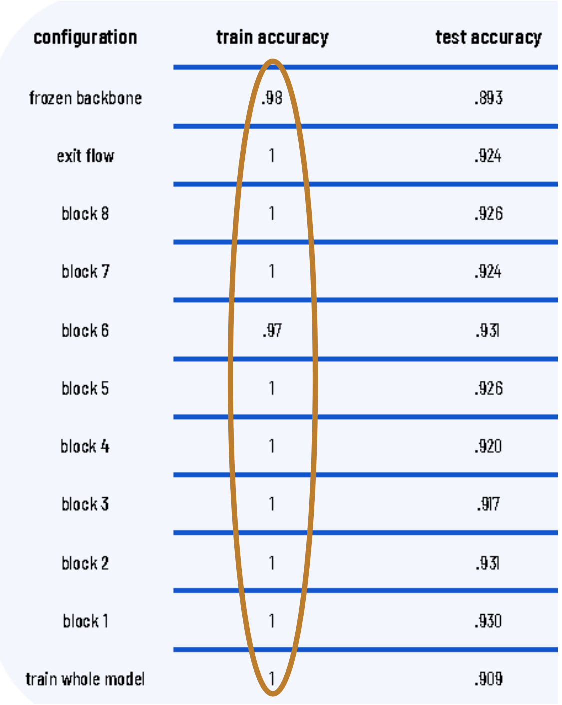


Input images and labels

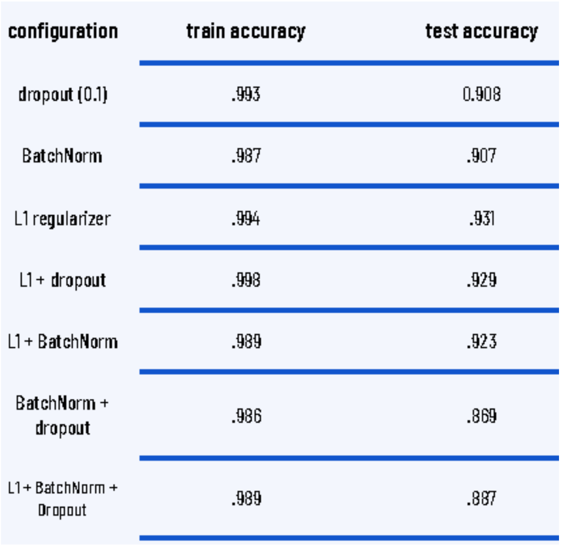
CNN

Use ground-truth labels from test to assess performance

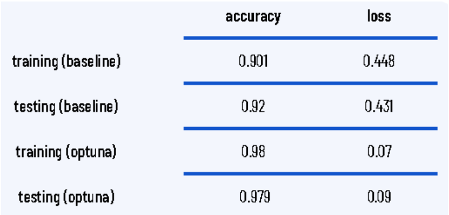
Xception 0.92



configuration	train accuracy	test accuracy
frozen backbone	.98	.893
exit flow	1	.924
block 8	1	.926
block 7	1	.924
block 6	.97	.931
block 5	1	.926
block 4	1	.920
block 3	1	.917
block 2	1	.931
block 1	1	.930
train whole model	1	.909



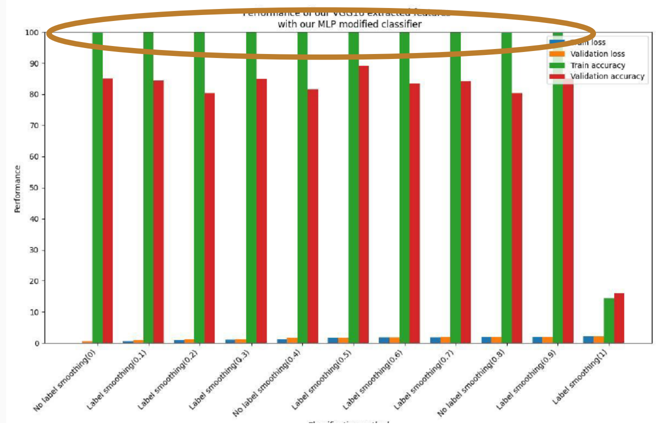
configuration	train accuracy	test accuracy
dropout (0.1)	.993	0.908
BatchNorm	.987	.907
L1 regularizer	.994	.931
L1 + dropout	.998	.929
L1 + BatchNorm	.989	.923
BatchNorm + dropout	.986	.869
L1 + BatchNorm + Dropout	.989	.887



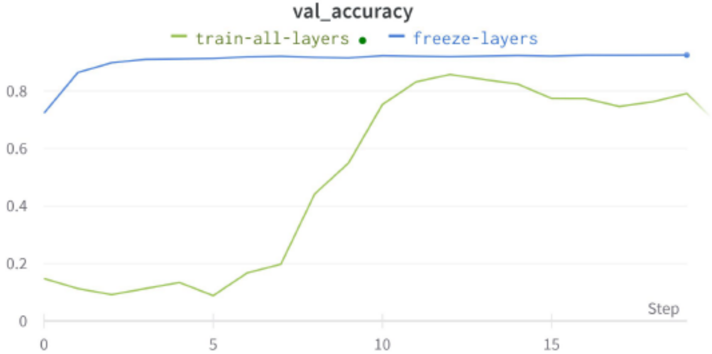
	accuracy	loss
training (baseline)	0.901	0.448
testing (baseline)	0.92	0.431
training (optuna)	0.98	0.07
testing (optuna)	0.979	0.09

We saw how in general it is better to finetune some of the feature extractor's layers on our dataset, instead of keeping it entirely frozen. Nevertheless, it is not advisable to totally unfreeze it.

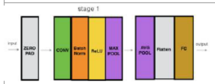
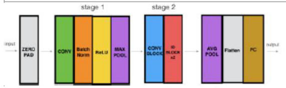
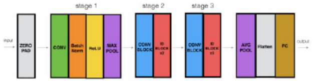
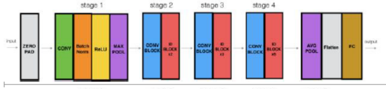
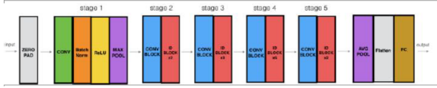
VGG16   0.89


Performance of our VGG16 extracted features
with our MLP modified classifier
Train loss
Validation loss
Train accuracy
Validation accuracy
Performance
100
90
80
70
60
50
40
30
20
10
0
No label smoothing(0)
Label smoothing(0.1)
Label smoothing(0.2)
Label smoothing(0.3)
No label smoothing(0.4)
Label smoothing(0.5)
Label smoothing(0.6)
Label smoothing(0.7)
No label smoothing(0.8)
Label smoothing(0.9)
Label smoothing(1)

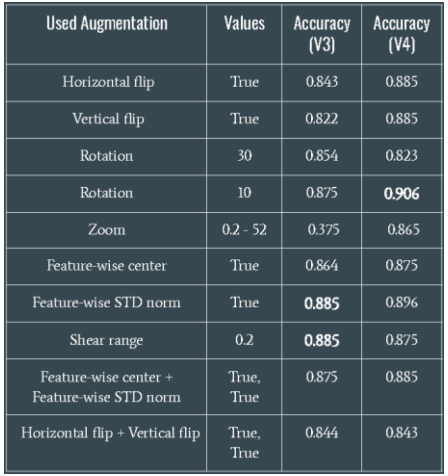
ResNet50 ??



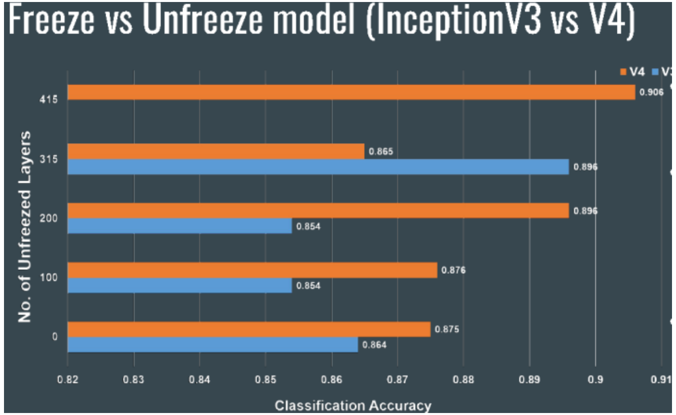
Better use train/val loss

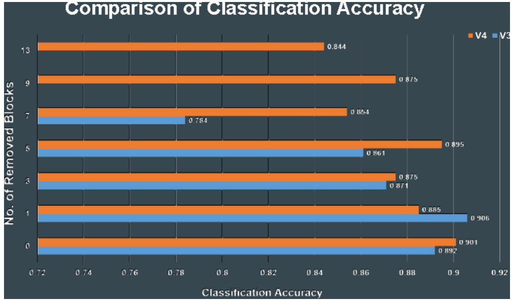
Test accuracy after training			Added **Dropout** after every conv	
Dropped Blocks	Accuracy	Loss	Accuracy	Loss
 stage 1 input ZERO PAD CONV Batch Norm ReLU MAX POOL AVG POOL Flatten FC output	0.5717	1.2997	0.5992	1.2404
 stage 1 stage 2 input ZERO PAD CONV Batch Norm ReLU MAX POOL CONV BLOCK ID BLOCK x2 AVG POOL Flatten FC output	0.7954	0.6399	0.8020	0.6324
 stage 1 stage 2 stage 3 input ZERO PAD CONV Batch Norm ReLU MAX POOL CONV BLOCK ID BLOCK x2 CONV BLOCK ID BLOCK x3 AVG POOL Flatten FC output	0.8899	0.3545	0.8824	0.3846
 stage 1 stage 2 stage 3 stage 4 input ZERO PAD CONV Batch Norm ReLU MAX POOL CONV BLOCK ID BLOCK x2 CONV BLOCK ID BLOCK x3 CONV BLOCK ID BLOCK x5 AVG POOL Flatten FC output	0.9380	0.2162	**0.9419**	0.2098
 stage 1 stage 2 stage 3 stage 4 stage 5 input ZERO PAD CONV Batch Norm ReLU MAX POOL CONV BLOCK ID BLOCK x2 CONV BLOCK ID BLOCK x3 CONV BLOCK ID BLOCK x5 CONV BLOCK ID BLOCK x2 AVG POOL Flatten FC output	0.9318	**0.2031**	0.9274	0.2067

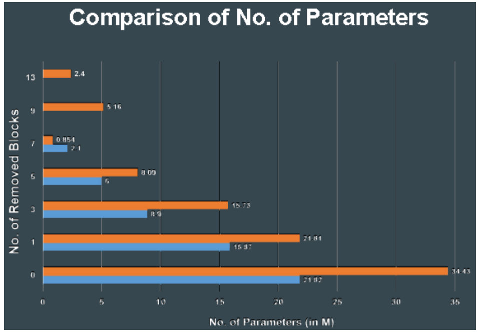
Inception v3   0.85-ish



Used Augmentation	Values	Accuracy (V3)	Accuracy (V4)
Horizontal flip	True	0.843	0.885
Vertical flip	True	0.822	0.885
Rotation	30	0.854	0.823
Rotation	10	0.875	**0.906**
Zoom	0.2 - 52	0.375	0.865
Feature-wise center	True	0.864	0.875
Feature-wise STD norm	True	**0.885**	0.896
Shear range	0.2	**0.885**	0.875
Feature-wise center + Feature-wise STD norm	True, True	0.875	0.885
Horizontal flip + Vertical flip	True, True	0.844	0.843

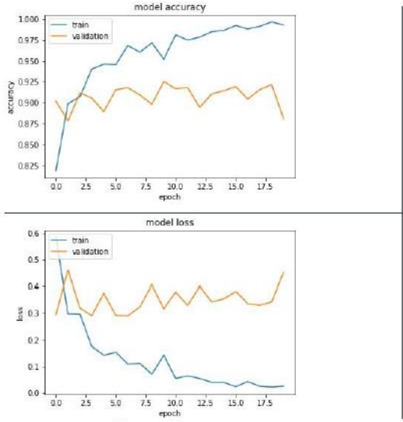

Freeze vs Unfreeze model (InceptionV3 vs V4)


Comparison of Classification Accuracy


Comparison of No. of Parameters

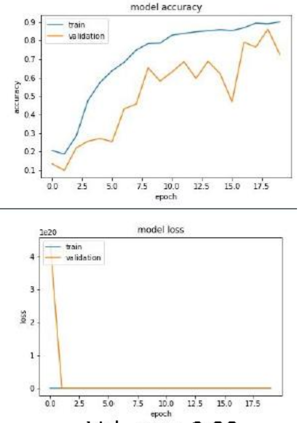
InceptionResnetV2 0.92

All frozen until last one



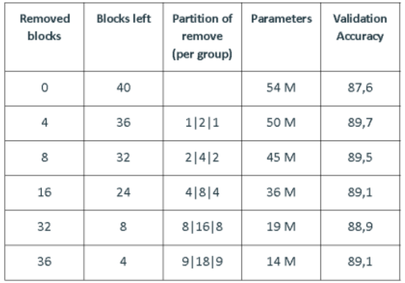
Val. acc.: 0.92

No one frozen

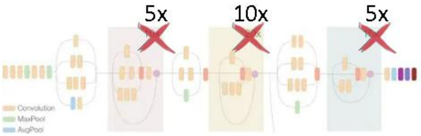


Val. acc.: 0.88

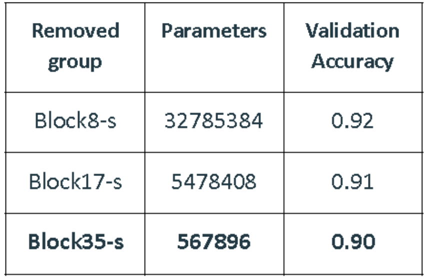
Removing blocks method 1:



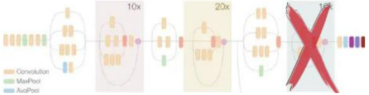
Removed blocks	Blocks left	Partition of remove (per group)	Parameters	Validation Accuracy
0	40		54 M	87,6
4	36	1|2|1	50 M	89,7
8	32	2|4|2	45 M	89,5
16	24	4|8|4	36 M	89,1
32	8	8|16|8	19 M	88,9
36	4	9|18|9	14 M	89,1



Removing blocks method 2:



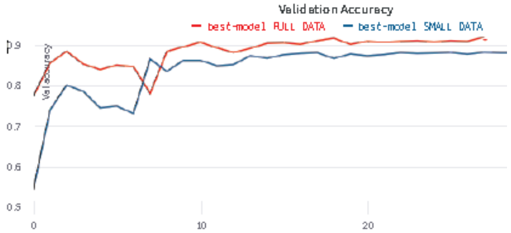
Removed group	Parameters	Validation Accuracy
Block8-s	32785384	0.92
Block17-s	5478408	0.91
Block35-s	**567896**	**0.90**



We achieved a validation accuracy of 89% with a network having 500k parameters (108 times less parameters than the original network) using only 400 training images

MobileNet 0.91

Same accuracy removing 2 blocks

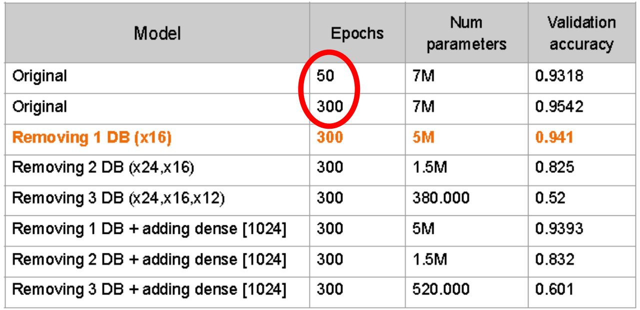


~88% Validation Accurracy on small dataset

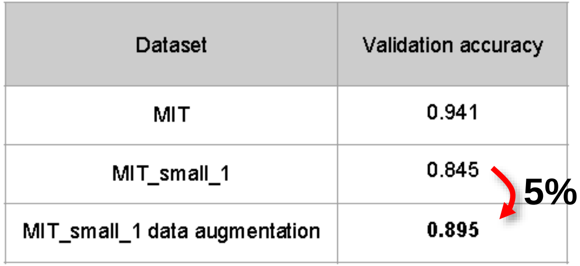
Focus on the message

https://wandb.ai/mcv-m3-g6/task_1_sweep/reports/Slides-M3-Week-4-Team-6--VmlldzozNDM0NTY1?accessToken=poxugew27pus1iukp4o5e39l53d8lrf0by30c6tugqfirps0dnin4rwq7gz2tp6w

DenseNet121   0.917

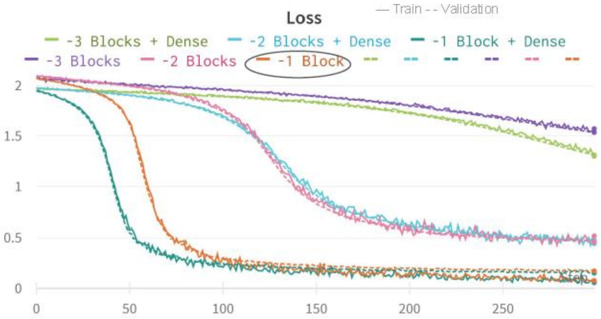


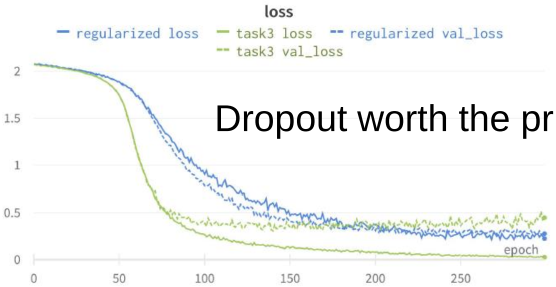
Model	Epochs	Num parameters	Validation accuracy
Original	50	7M	0.9318
Original	300	7M	0.9542
Removing 1 DB (x16)	**300**	**5M**	**0.941**
Removing 2 DB (x24,x16)	300	1.5M	0.825
Removing 3 DB (x24,x16,x12)	300	380.000	0.52
Removing 1 DB + adding dense [1024]	300	5M	0.9393
Removing 2 DB + adding dense [1024]	300	1.5M	0.832
Removing 3 DB + adding dense [1024]	300	520.000	0.601



Dataset	Validation accuracy
MIT	0.941
MIT_small_1	0.845
MIT_small_1 data augmentation	**0.895**

5%!!

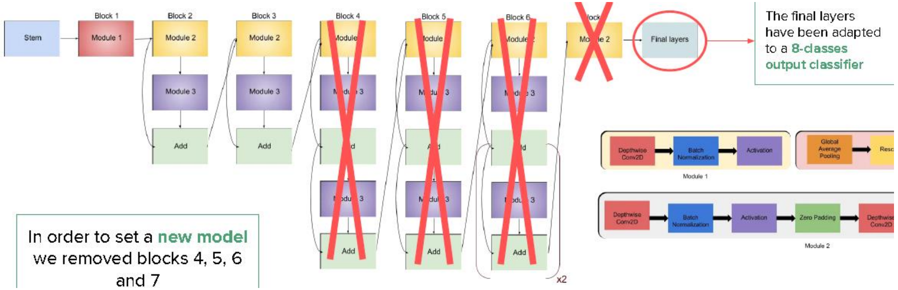




Dropout worth the price

NasNetMobile 0.913

EfficientNetB0   0.886

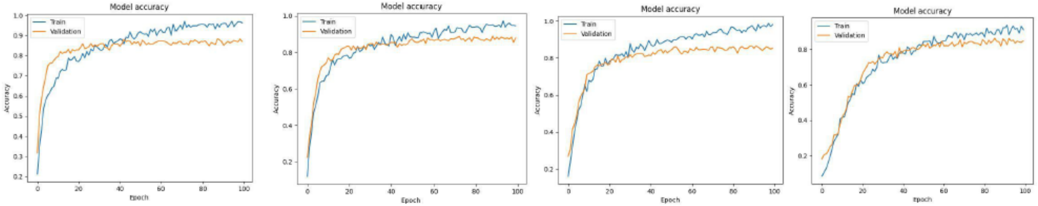


The final layers have been adapted to a **8-classes output classifier**

In order to set a **new model** we removed blocks 4, 5, 6 and 7

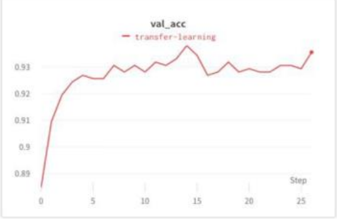
0.855 on small dataset

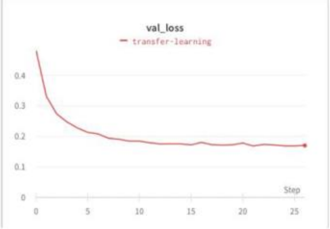
8 Batch		16 Batch		32 Batch		64 Batch	
Train acc	Val acc	Train acc	Val acc	Train acc	Val acc	Train acc	Val acc
96.24%	87.00%	94.50%	87.88%	98.09%	85.37%	91.07%	84.89%



ConvNextTiny   0.934

Overfits?





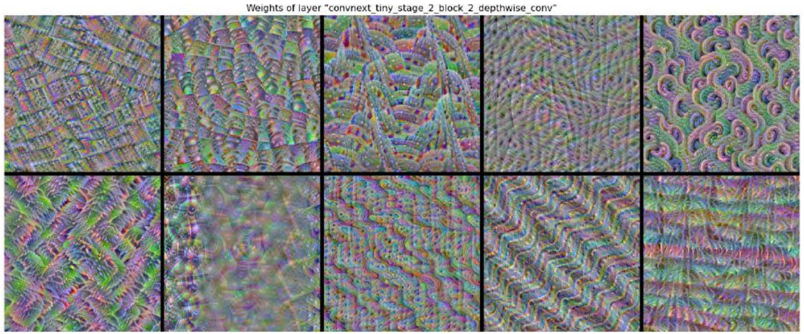
Training set	Accuracy
MIT_small_1	0.9165
MIT_small_2	0.9318
MIT_small_3	0.9335
MIT_small_4	**0.9419**
MIT_training(transfer learning)	0.9343

Data aug.

After scaling down our search space the following data augmentation parameter values resulted in models with high accuracies:

- Height shift: <0.06
- Horizontal flip: True
- Rotation: <10 deg
- Shear shift: <0.28
- Vertical flip: False
- Width shift: <0.13
- Zoom: <0.1

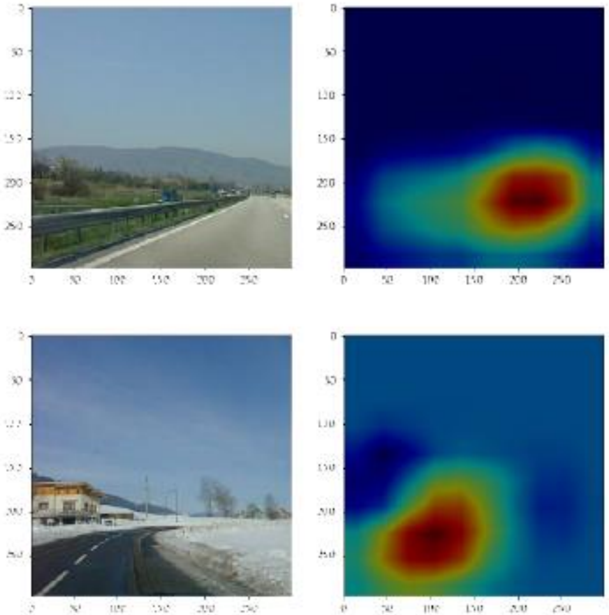
D.A depends on the architecture


Weights of layer "convnext_tiny_stage_2_block_2_depthwise_conv"

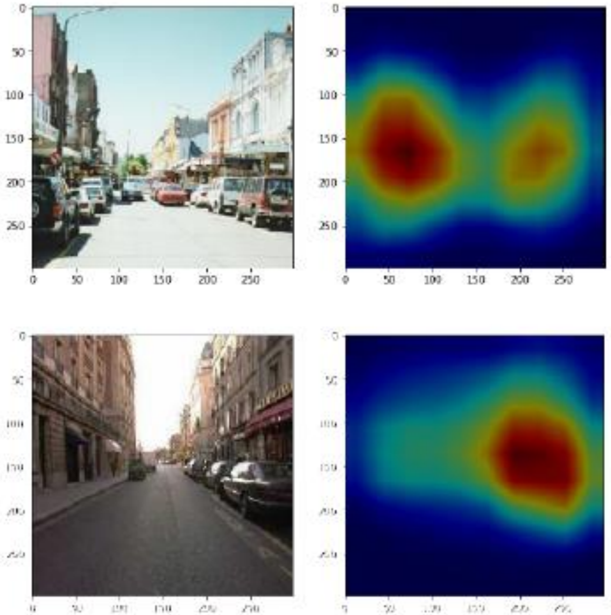
Group	grade
1	10
2	6
3	9
4	8
5	10
6	8
7	10
8	9
9	9
10	9

EXTRA: Interpretability - Class Activation Maps

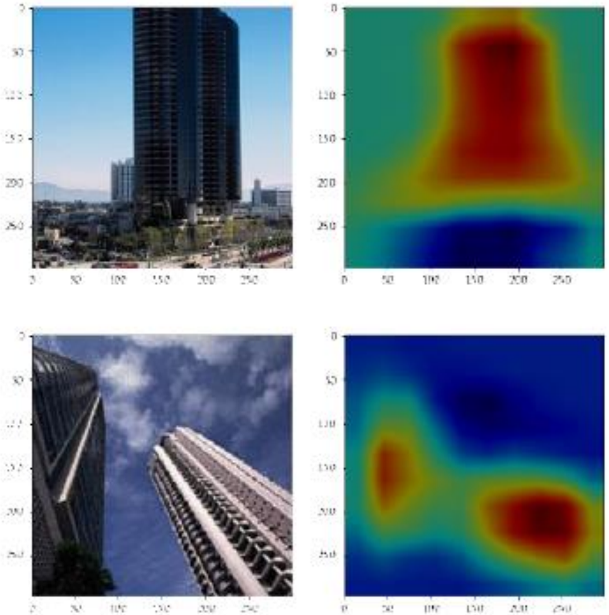
Correct highway



Correct street



Correct tallbuildings

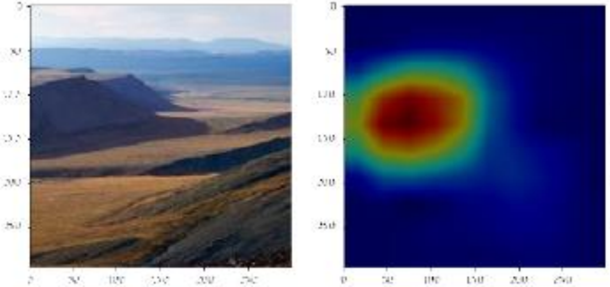


EXTRA: Interpretability - Errors analysis

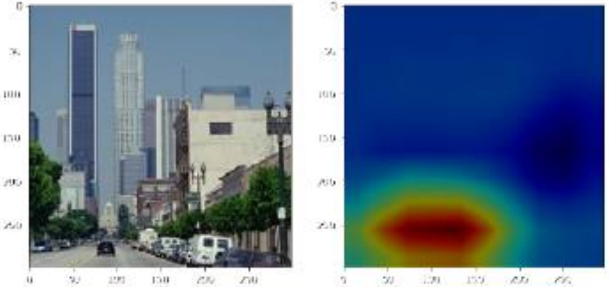
(Ground truth => **Network prediction**)

to the appearance of key features of other classes in the image. This is **very clear with opencountry**.

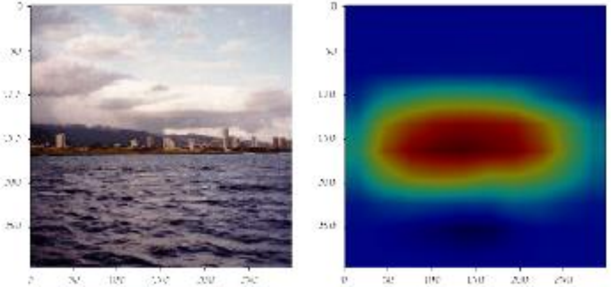
Opencountry => **Mountain**



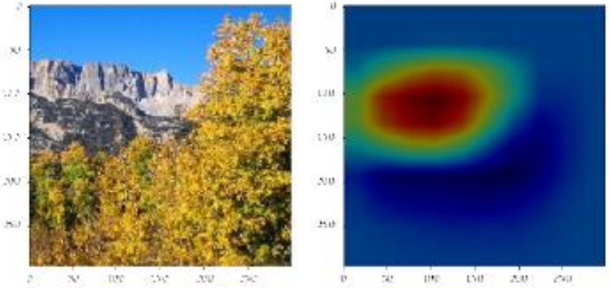
Tallbuilding => **Highway**



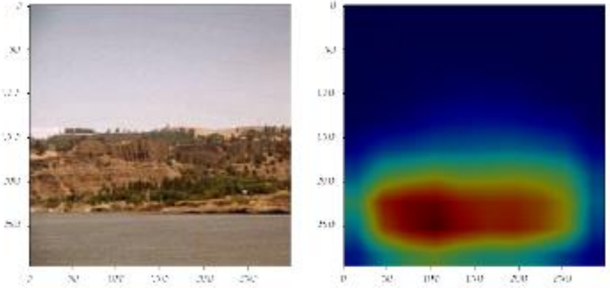
Coast => **Tallbuilding**



Forest => **Mountain**



Opencountry => **Coast**



Opencountry => **Inside_city**