This Could Be Anywhere

Winning Geoguessr using Convolutional Neural Networks

August Baggesen Hilger, Catrine Trudslev og Emma Kirchhoff

GitHub: https://github.com/890tsugua/AppliedML2026
Data (Google Drive): https://drive.google.com/drive/folders/1SV1wbInxP3pNCU2XCWl14cqe2YSgLwjH?usp=sharing

UNIVERSITY OF COPENHAGEN

/
2
0
2
6

Where did Troels go on vacation?

The dataset

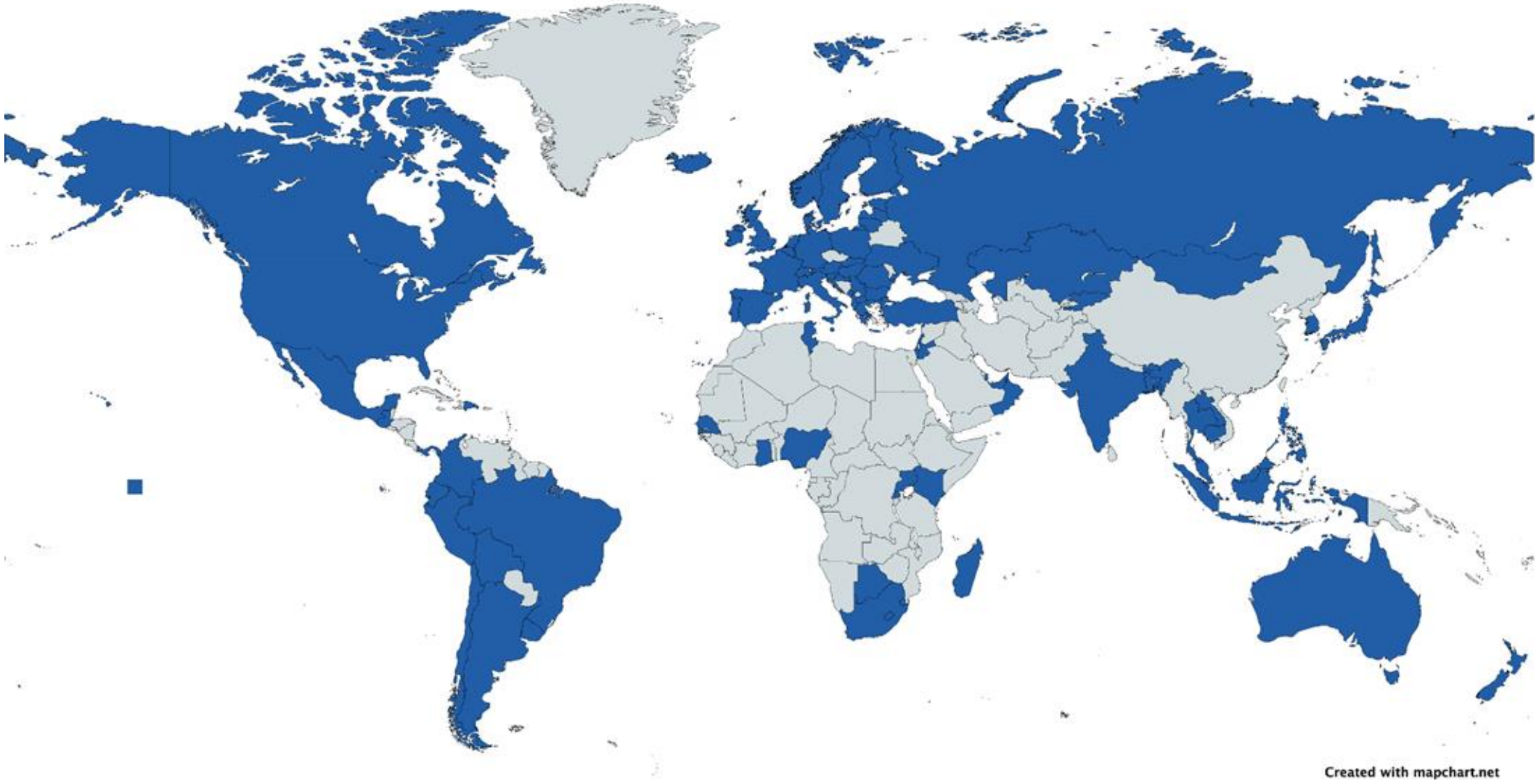
- Images from around the world
- Scraped from Geoguessr*

- ~ 300 images per country
- 1600 x 900 image size

- ~ 60 GB
- Two folders: test and train (80/20)

*Credit Pepijn van Wijk (deboradum on GitHub):
https://huggingface.co/datasets/deboradum/GeoGuessr-countries

Countries in the dataset



Example images

Colombia

Denmark

South Korea

Preparing the dataset

Huggingface database

~ 60 GB

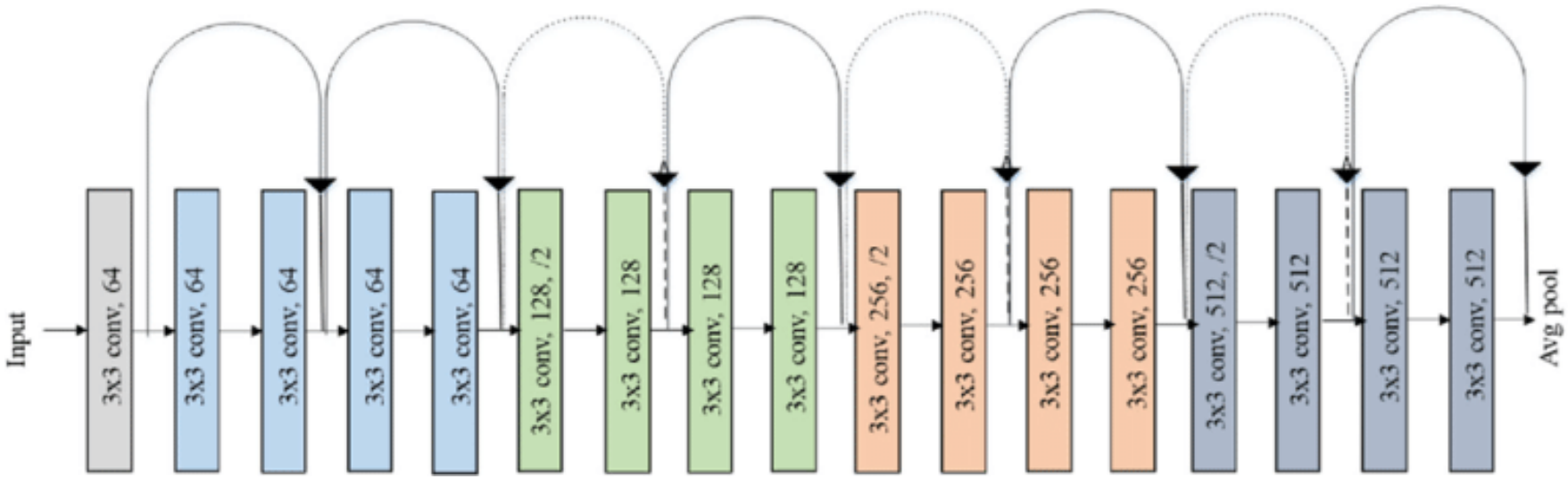
Compress 1600x900 → 500x281 and zip (!) locally

~ 1 GB

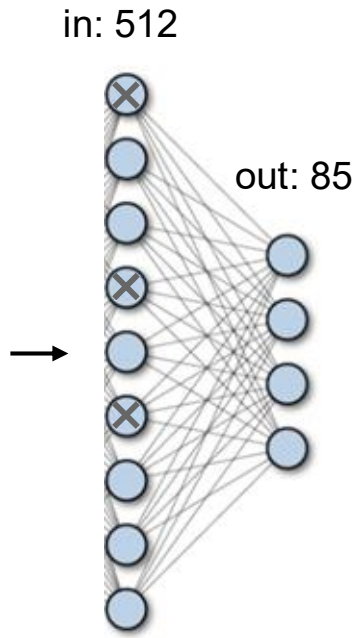
Load from Drive into Colab and unzip each session

- We used the same Google account across PCs
- Gave us a space to share files, results and access to good GPU
- Would mostly edit in VScode and push to Github, then clone repo into Colab.

Pretrained ResNets



Resnet18
(4 trainable layers)



FC classification layer
(dropout = 0.3)

Loss function: **Cross-entropy loss**

Data augmentation

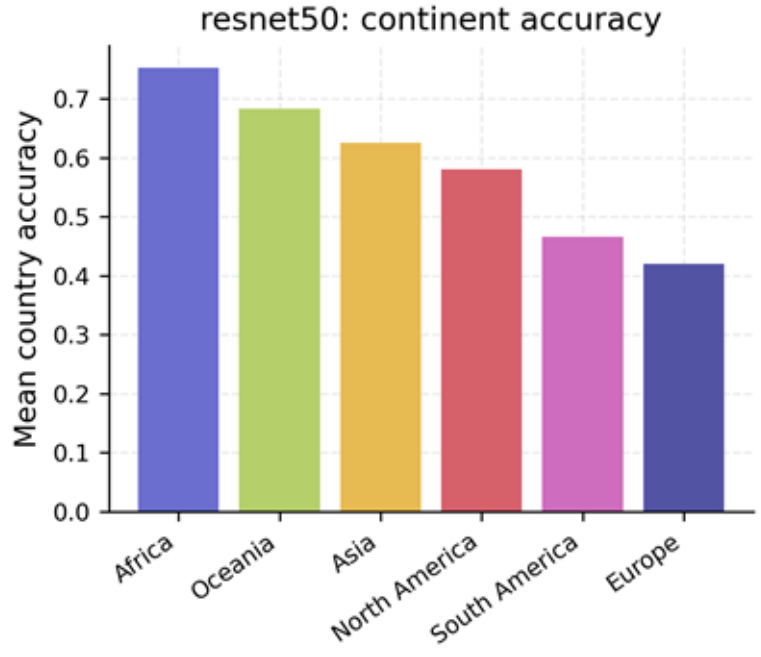
- Artificially create new data samples for training
- Involves random cropping, rotation, color jitter
- Random data augmentation → better performance.
- Pre-computed augmentation → severe overtraining

Somewhere in Albania…

GPUmaxxing

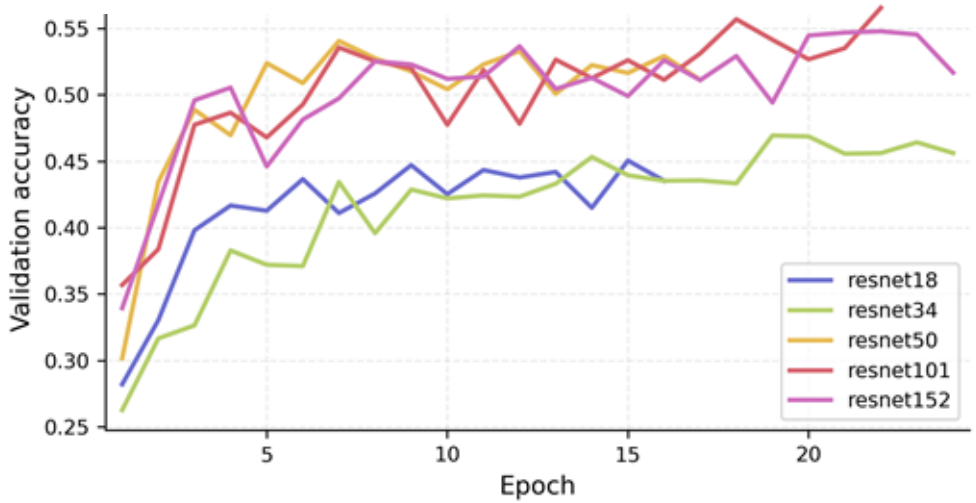
- Colab gives you 100 Compute Units. Use wisely!
  - Do *not* choose a more powerful GPU than necessary. Consider how heavy your task is and check with nvidia-smi.
  - Usage starts already from connection, *not* when you start running code.
- CPU bottleneck: Image preprocessing / augmentation can be expensive.
  - Think about your transforms
- Dataloader settings:
  - num_workers = 8 (!)
  - prefetch_factor = 2 or 4

Initial results with dropout and simple augmentation



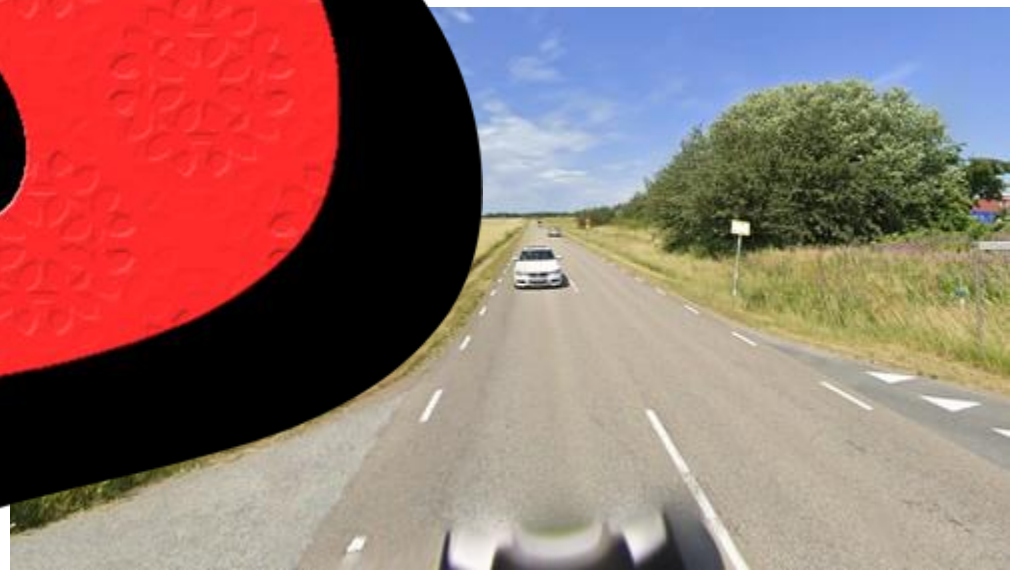
ResNet50 Test Accuracy	
Top 1	Top 5
0.541	0.838

- Europe hard. Unique countries easy.
- Dataset imbalance and image artefacts



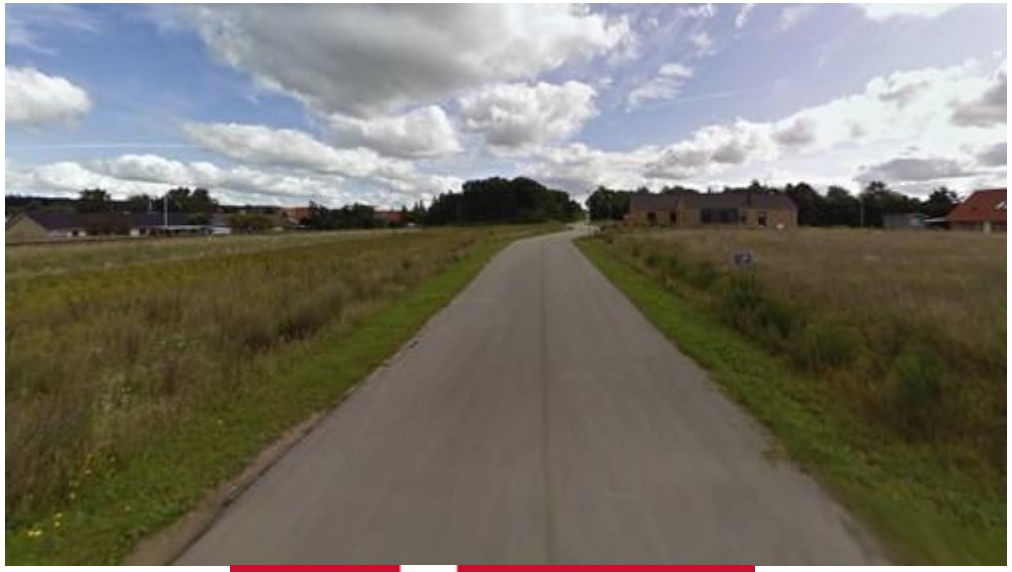
Why cluster countries?

Motivation



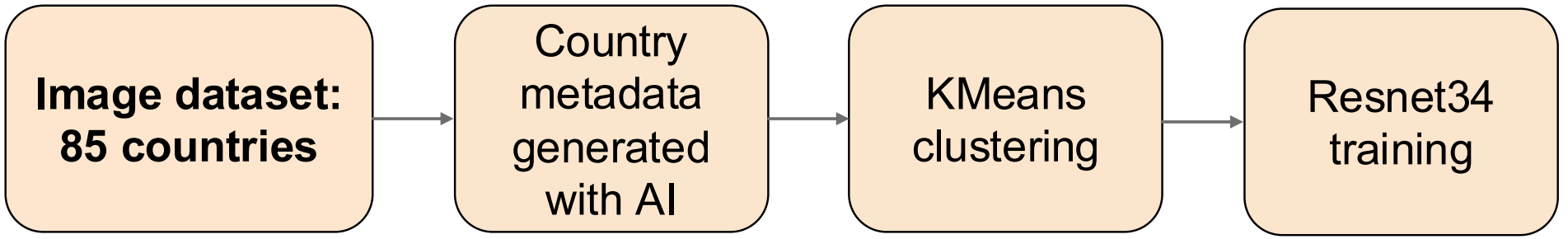
Why cluster countries?

Motivation



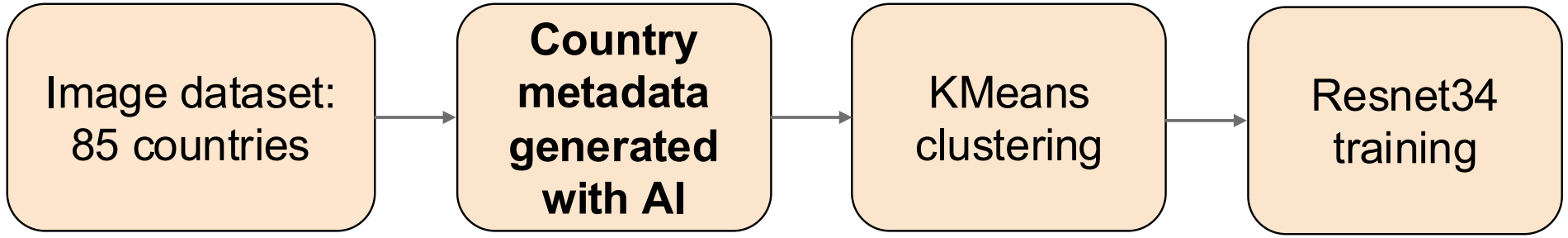
Cluster Based Country Image Classification

Setup



Cluster Based Country Image Classification

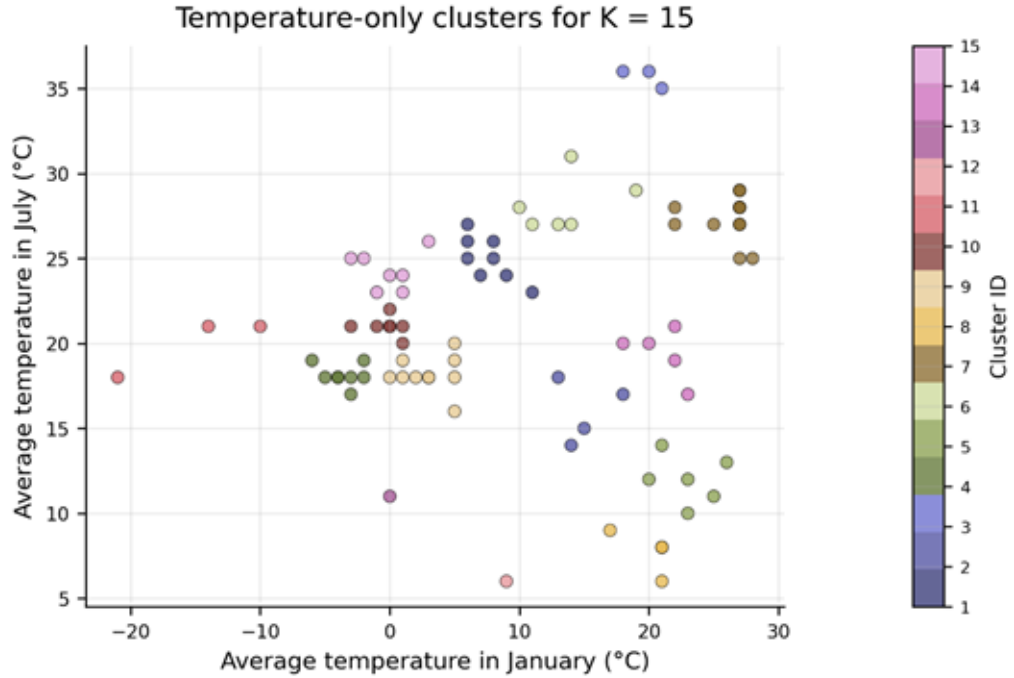
Setup



Cluster Based Country Image Classification

Setup

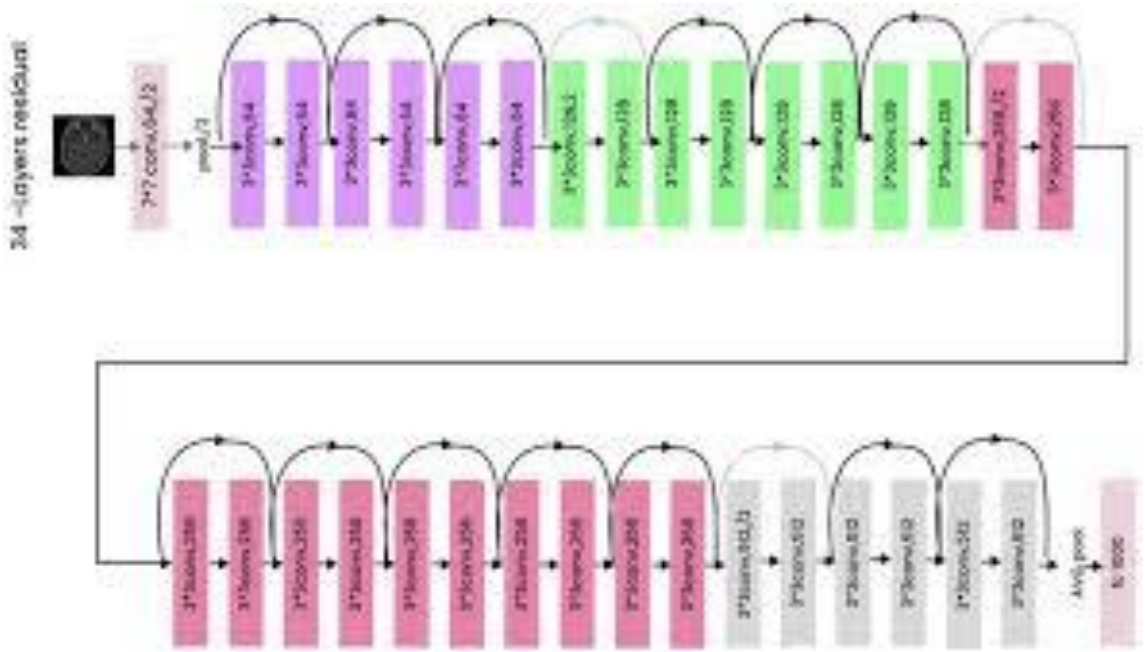
Image dataset: 85 countries → Country metadata generated with AI → **KMeans clustering** → Resnet34 training



Cluster Based Country Image Classification

Setup

Image dataset: 85 countries → Country metadata generated with AI → KMeans clustering → **Resnet34 training**



Cluster Based Country Image Classification

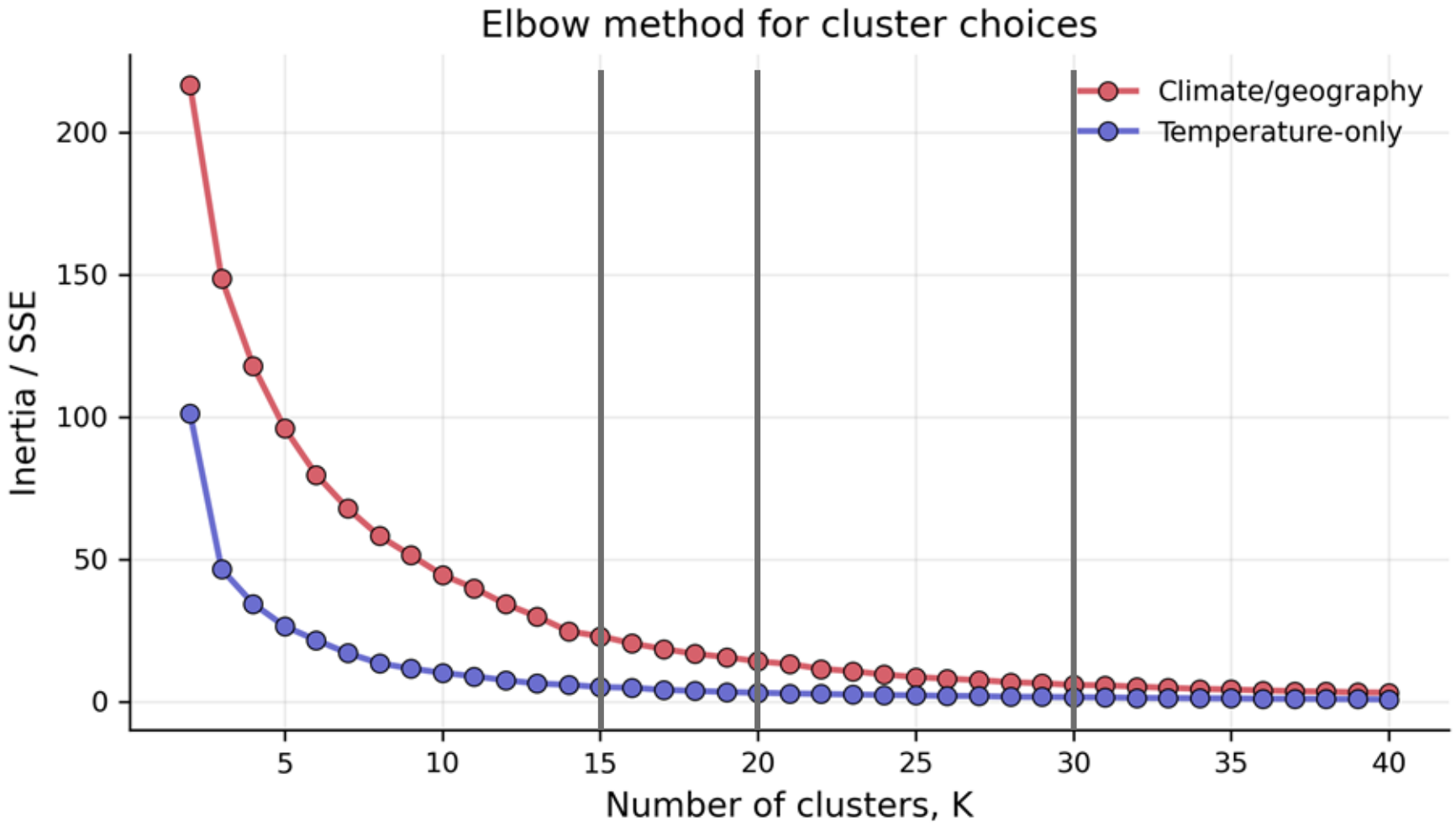
Two sets of features used for clustering

- **Temperature-only based clustering:**
  - Jan, July temperature (2 features)

- **Climate/Geography based clustering:**
  - Jan, July temp. + latitude and longitude (4 features)

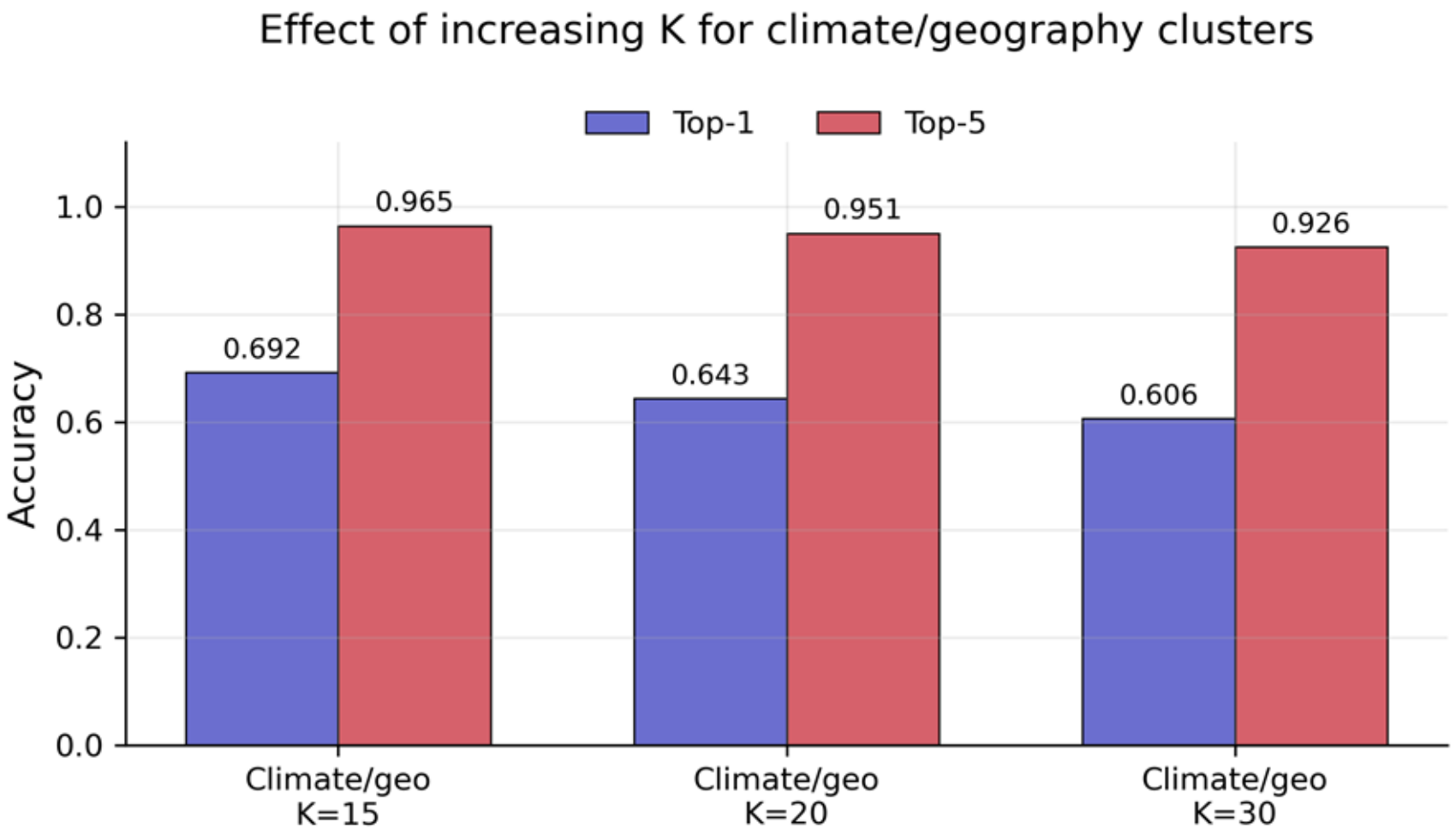
Cluster Based Country Image Classification

Choosing the number of clusters



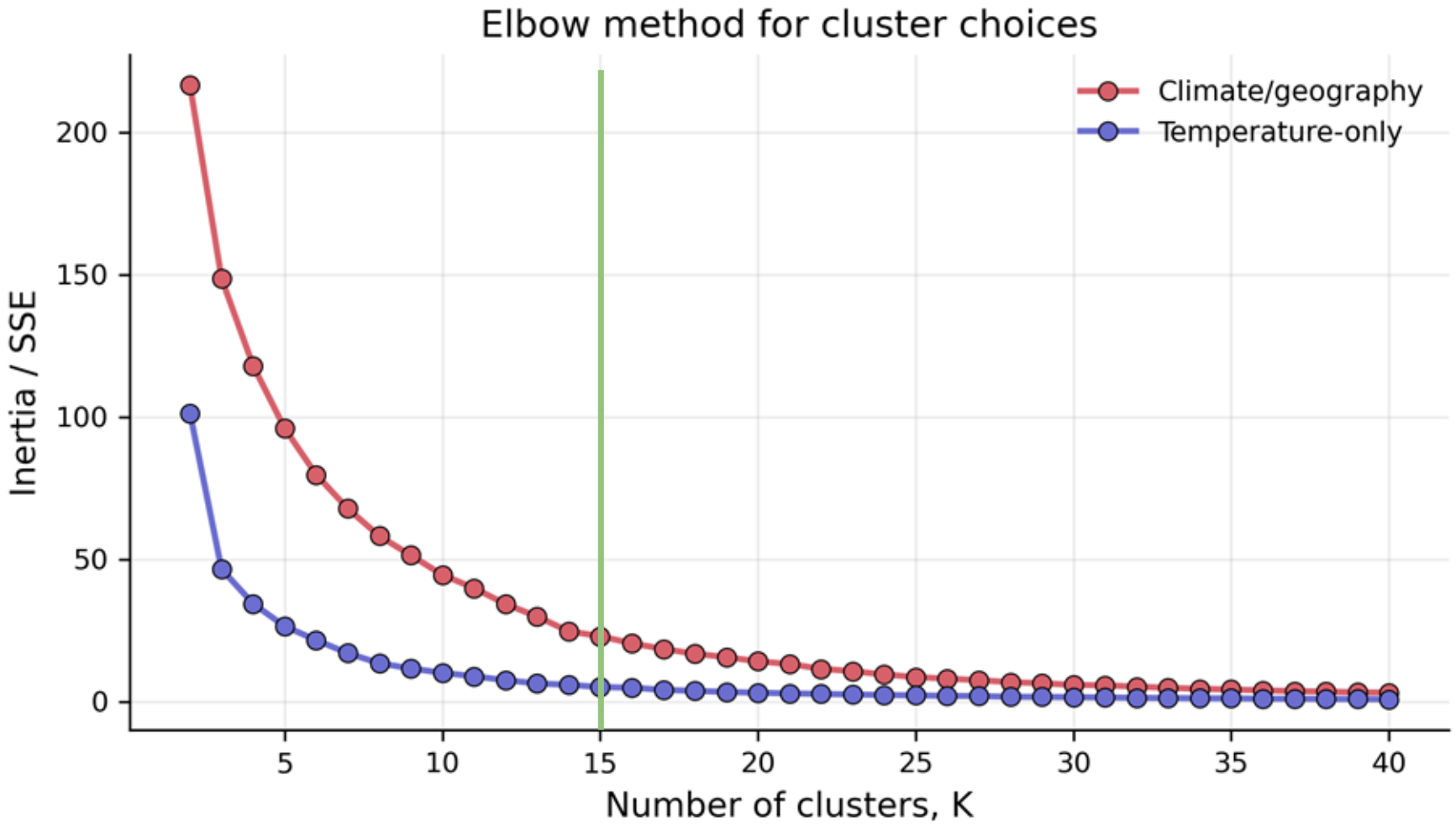
Predicting clusters from images using ResNet-34

Accuracy of prediction decreases with number of clusters

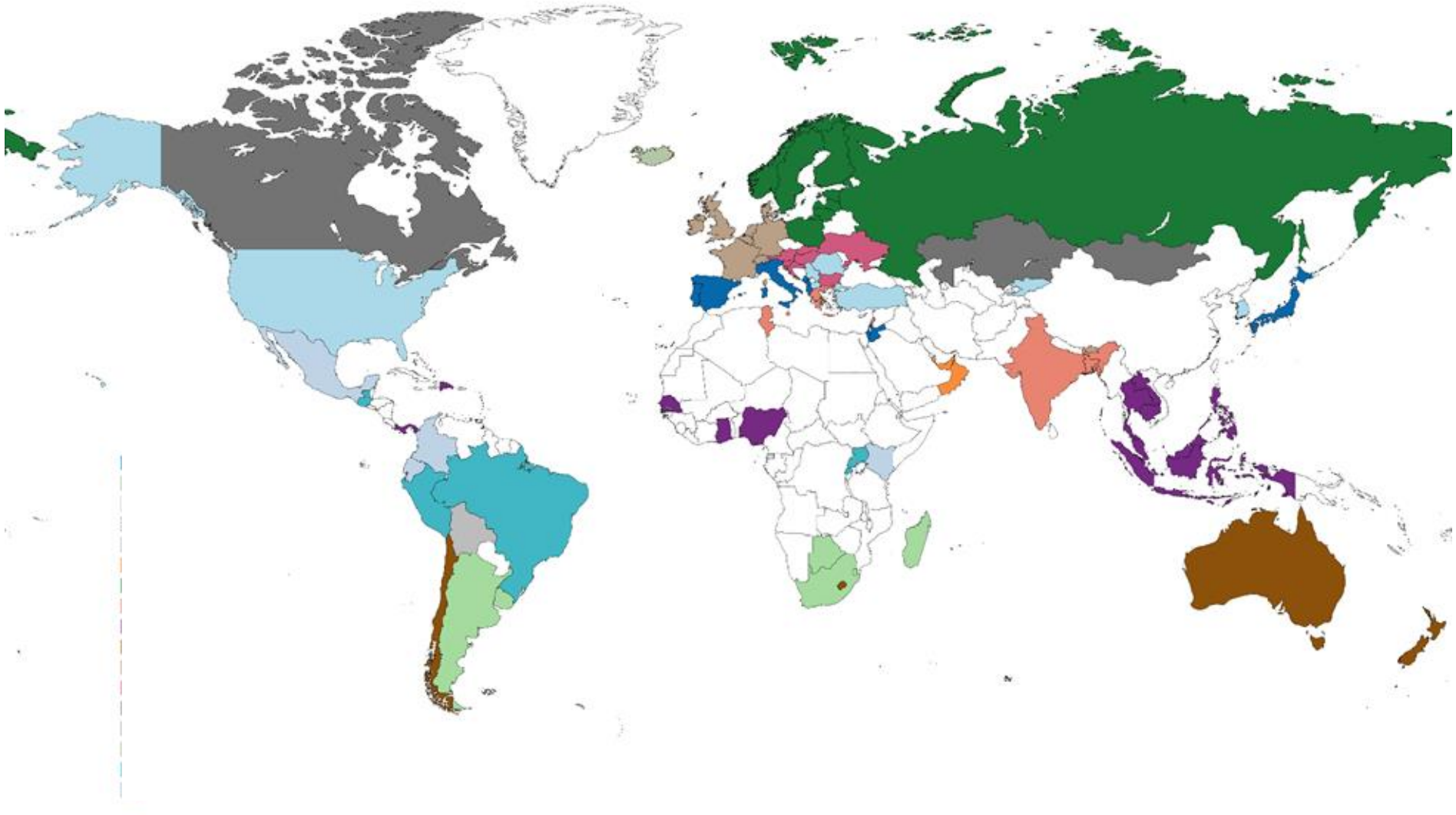


Cluster Based Country Image Classification

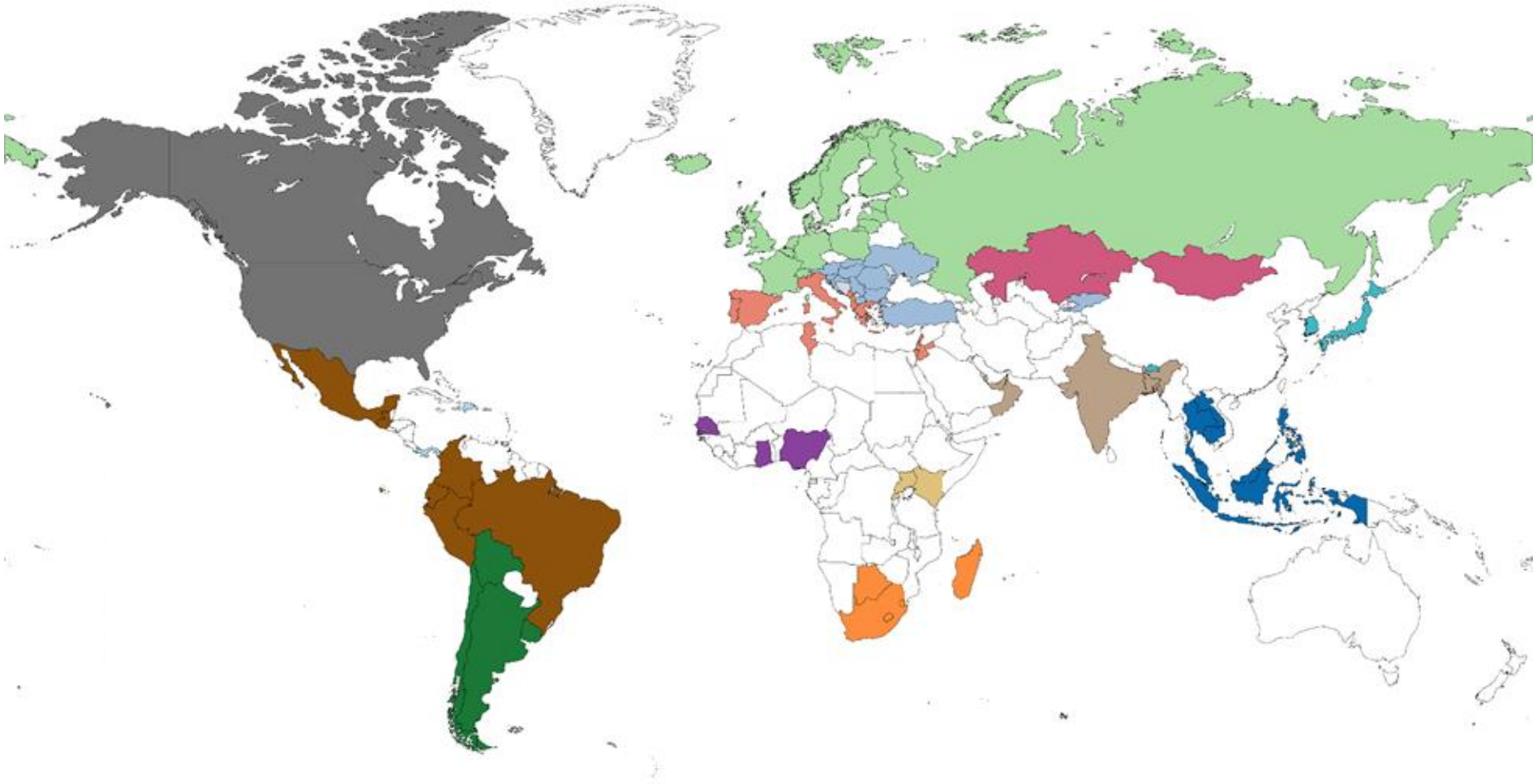
We will continue with 15 clusters



Temperature-only: Clusters are spread geographically

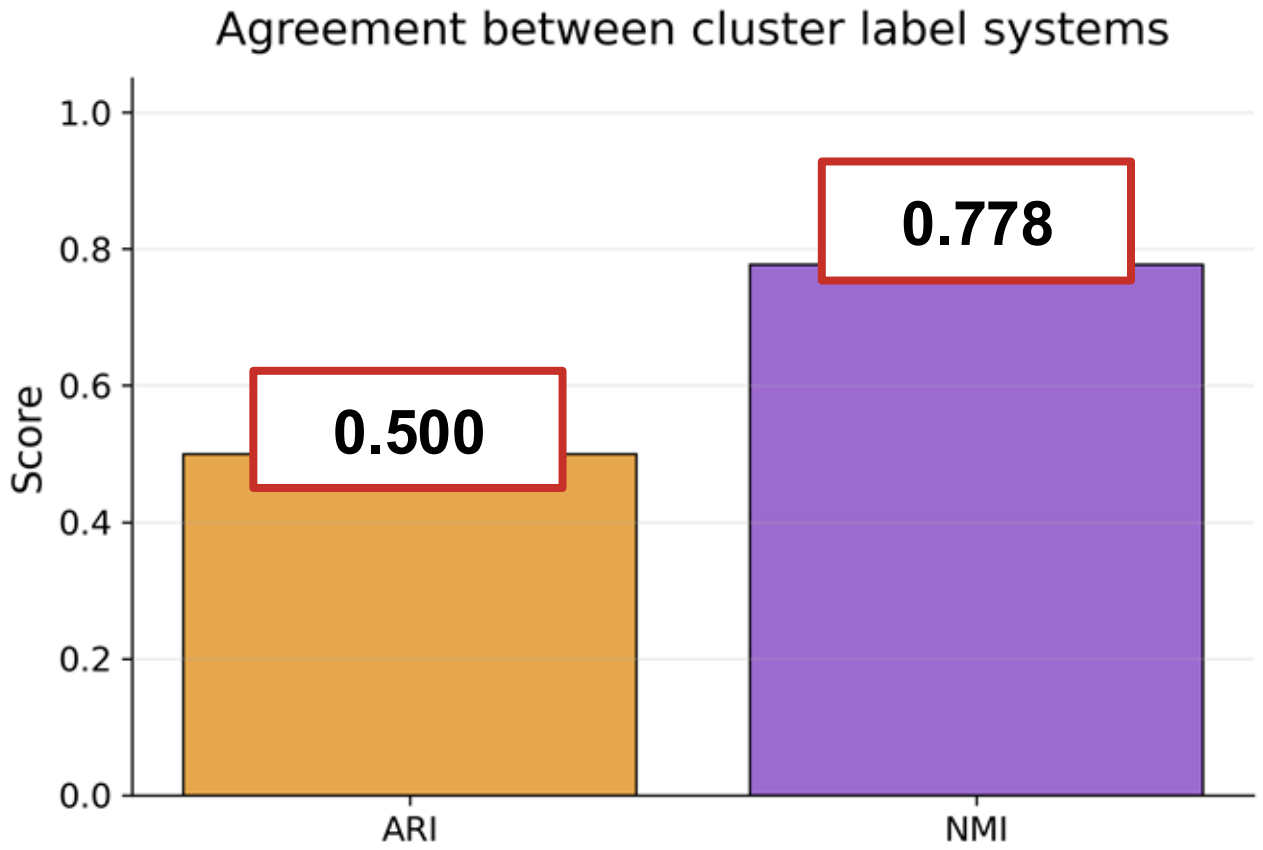


Climate/Geography: Clusters are grouped geographically



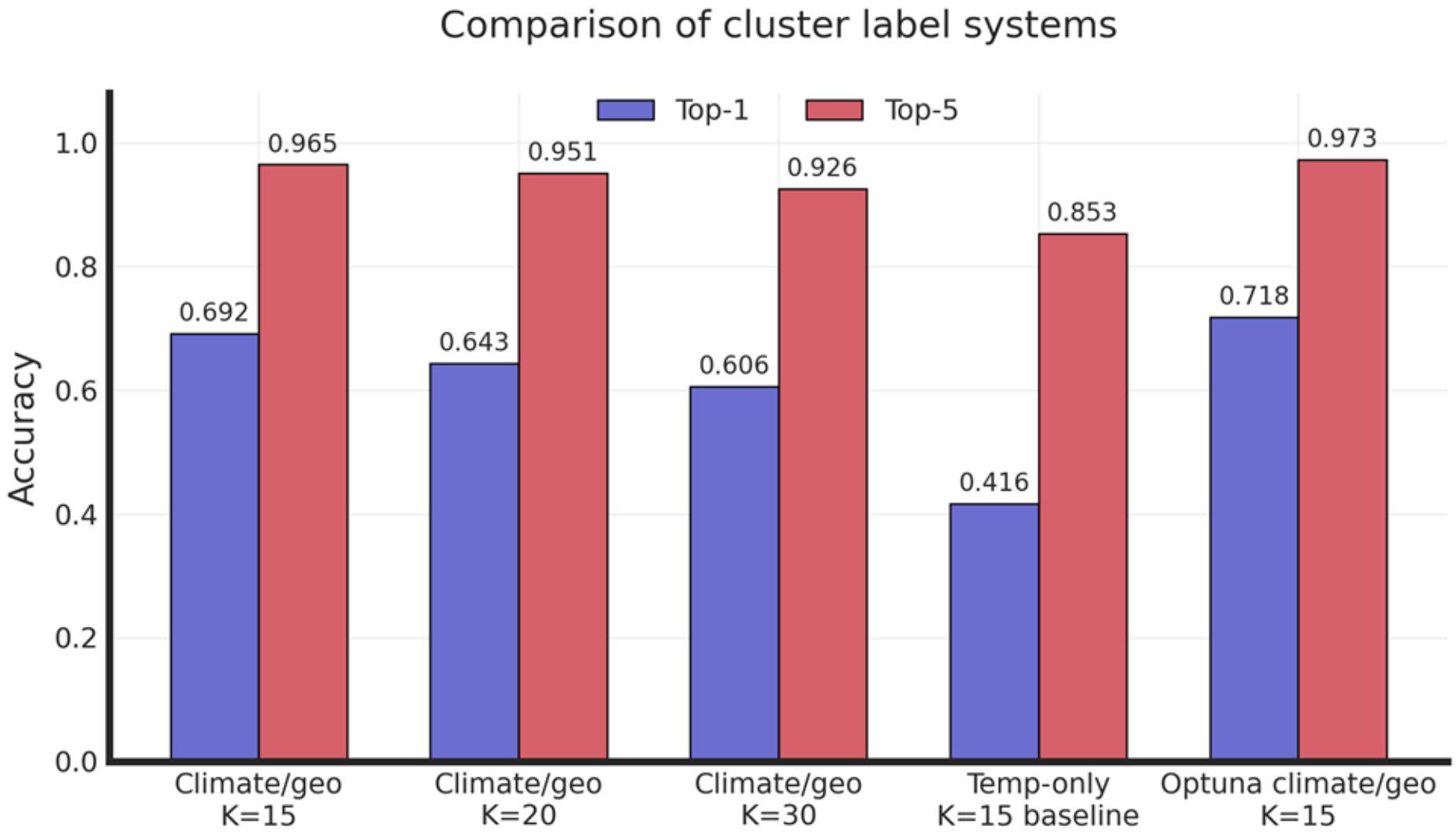
Comparing clustering for temperature-only vs. climate/geography

Adjusted Rand Index (ARI) and Normalized Mutual information (NMI)



Predicting clusters from images using ResNet-34

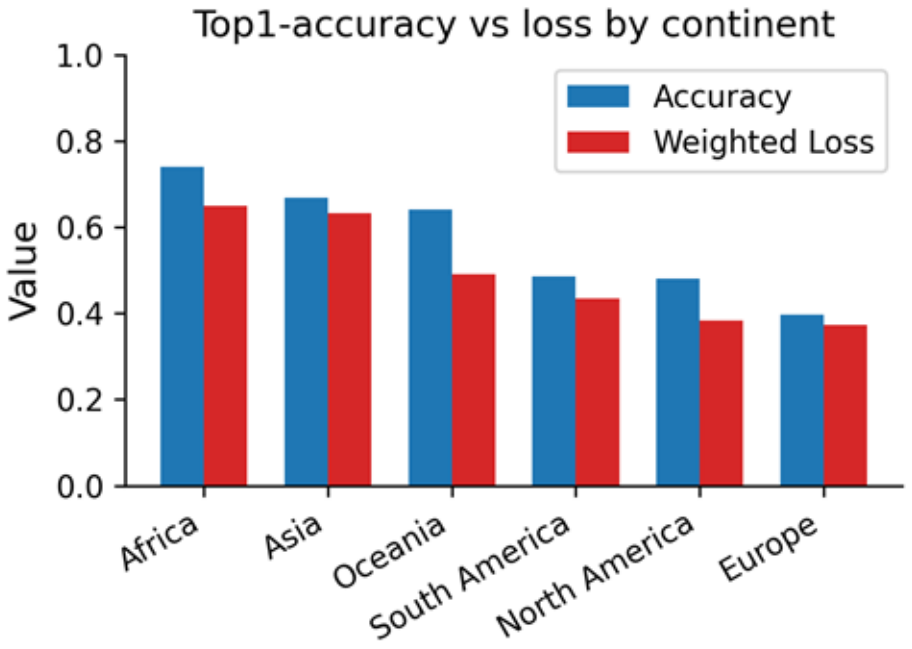
Prediction works best on climate/geography based clusters. Optuna helps.

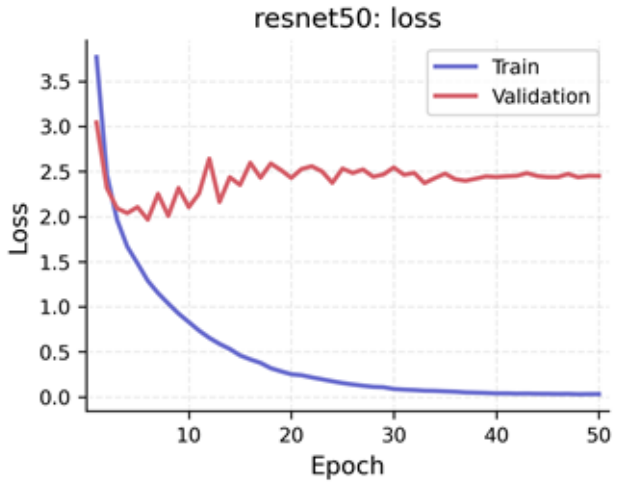


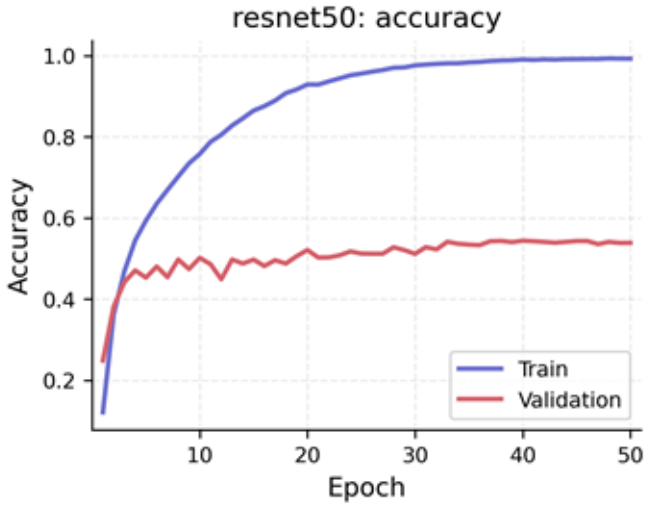
Adding weighted cross-entropy loss function

Weight = 1.5 on European countries does not improve prediction on Europe

$$L = -\sum_{k=1}^{K} y_k \log(p_k)$$







ResNet50 Test Accuracy	
Top 1	Top 5
0.518	0.817

Label Smoothing

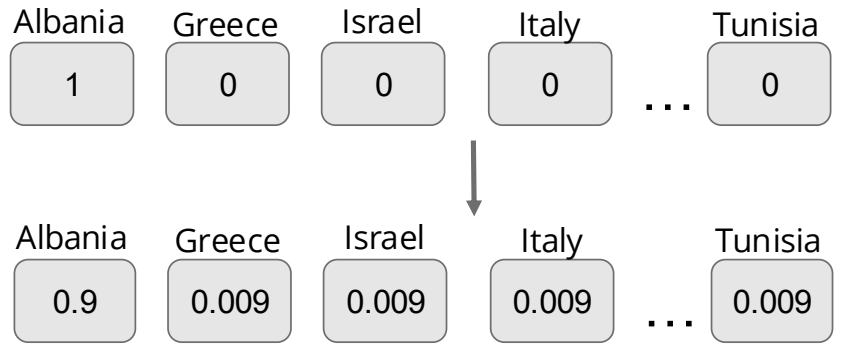
$$L = -\sum_{k=1}^{K} y_k \log(p_k)$$

The k=15 clusters was used
With prob 0.9

ResNet50 Test Accuracy	
Top 1	Top 5
0.525	0.816

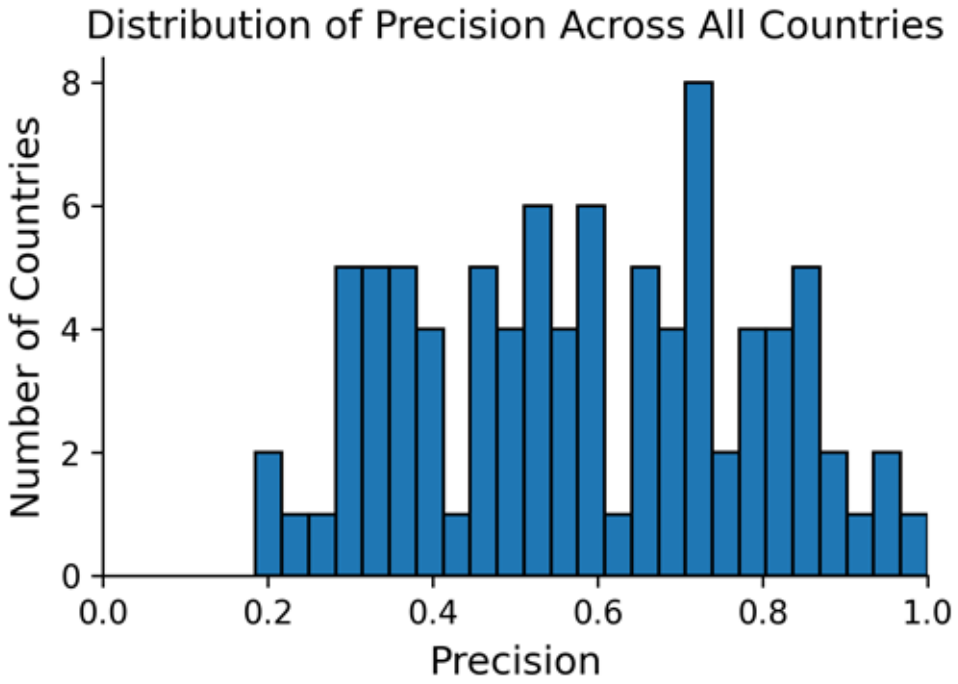
Example (Albania in Cluster 5)

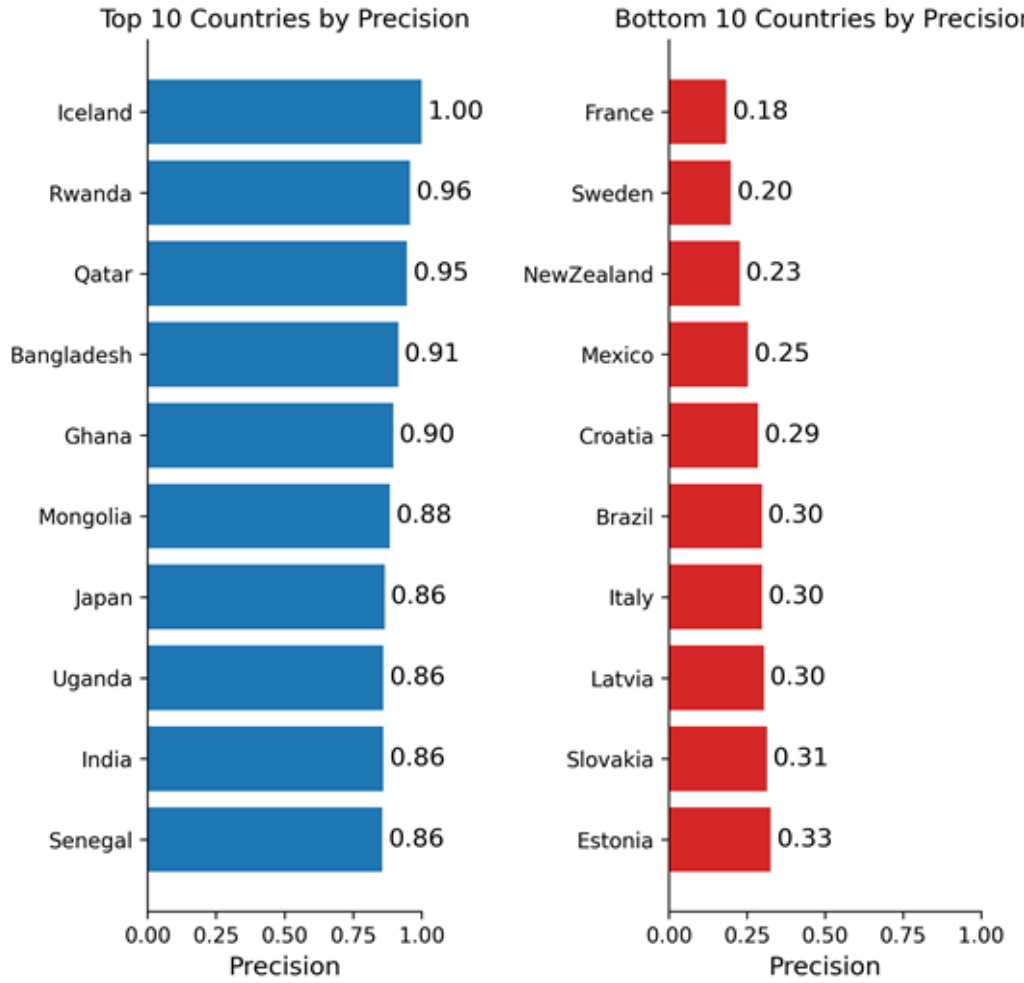
Cluster 5
Albania, Greece, Israel, Italy, Jordan, Lebanon, Malta, Montenegro, Portugal, Spain, Tunisia



- Clusters were probably too large, so smoothing probability was distributed across too many countries.
- As a result, smoothed labels were small-valued and almost identical to normal one-hot labels.

Accuracy of prediction still highly depends on the country





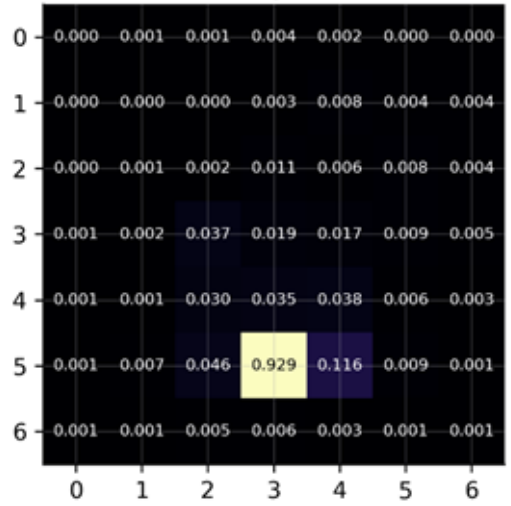
Dataset pollution: Car artefacts

- Car hood artefacts on many images
- Example images from Ghana

Network has learned the car hoods of specific countries…

Original

Patch importance



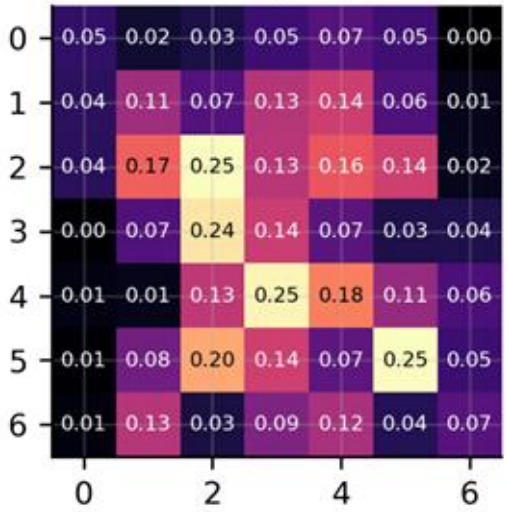
Patch importance overlay

… but not always!

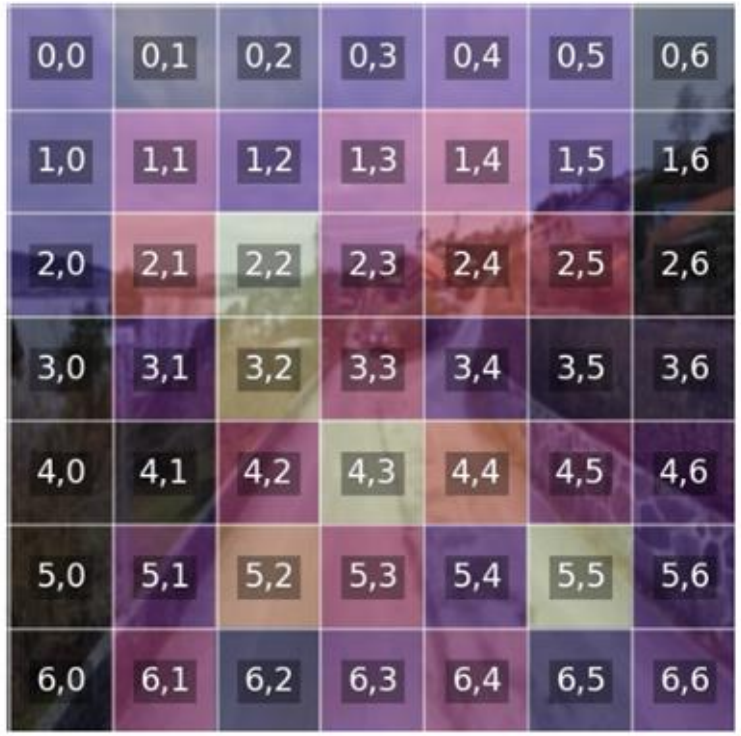
Correctly predicted Norway's cluster without car artefact.

Original

Patch importance

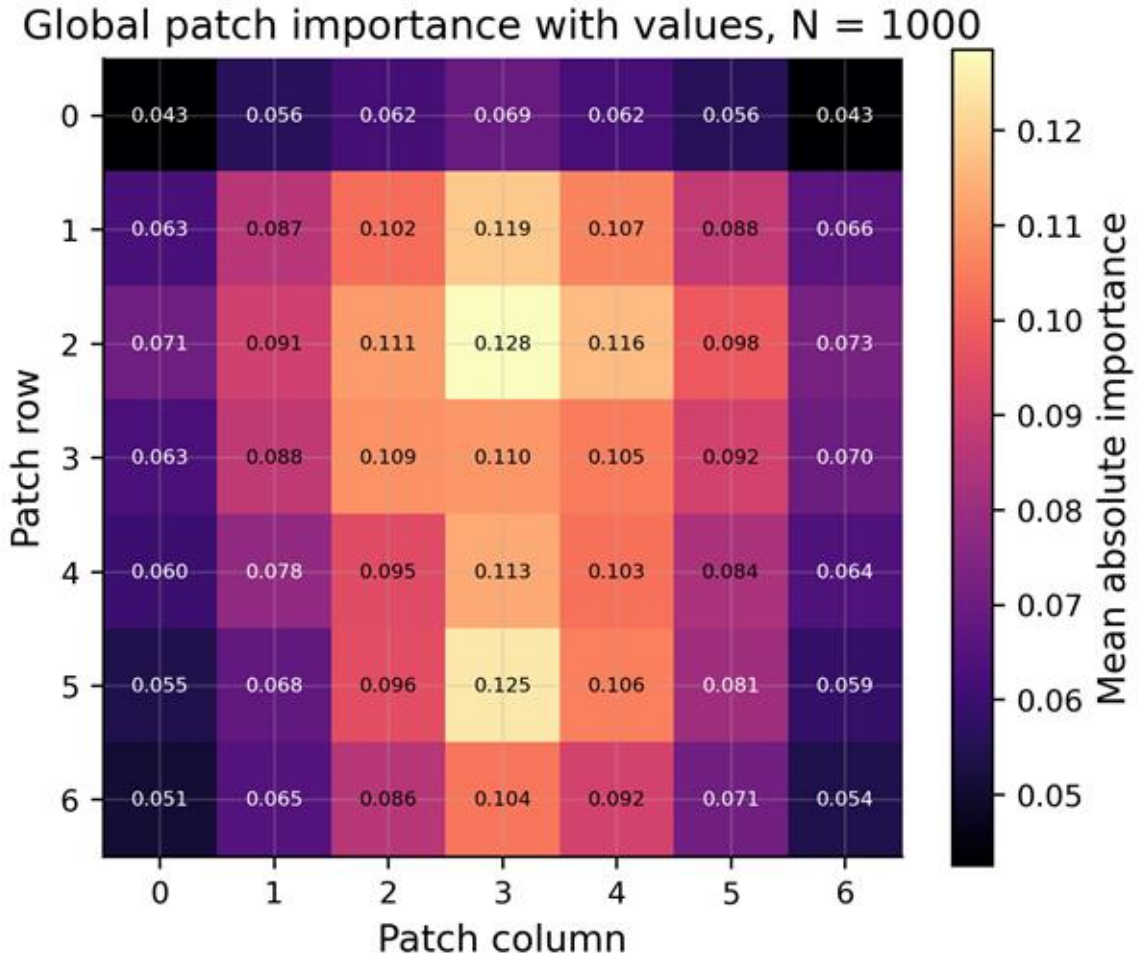


Overlay + patch IDs



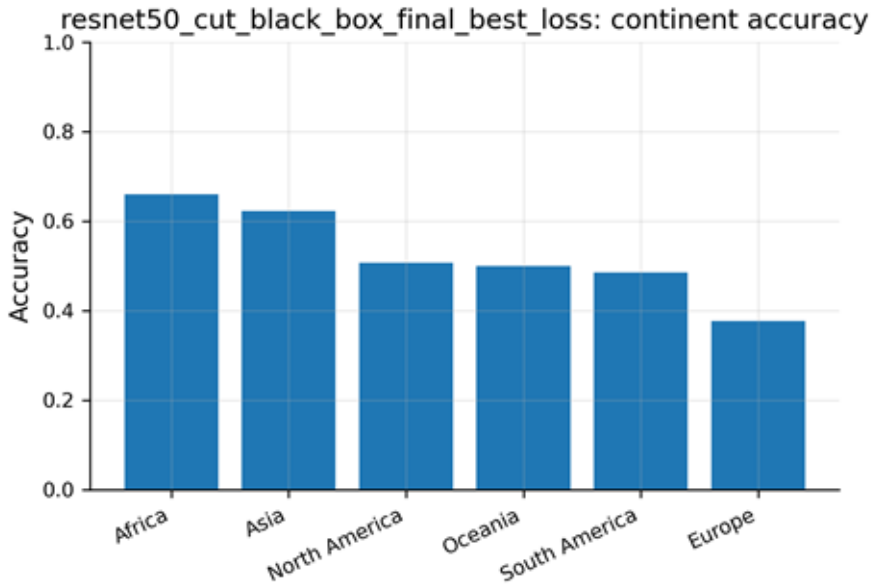
Prediction relies not only on car artefacts

… but definitely relies heavily on car artefacts

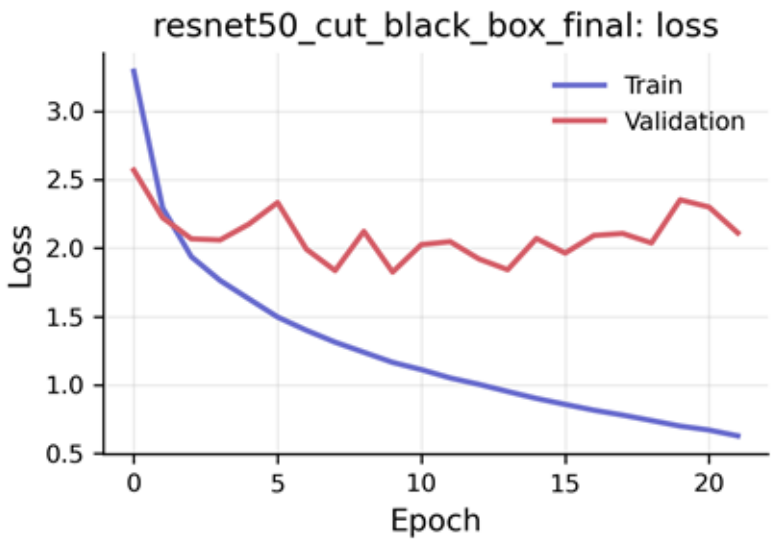


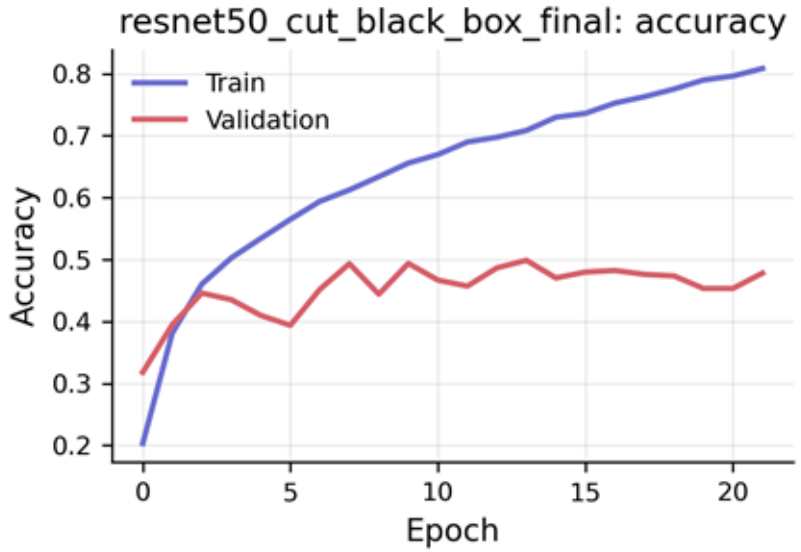
Cutting out the cars: Inserting a black box

Accuracy decreased for unique countries after cutting out cars



ResNet50 Test Accuracy	
Top 1	Top 5
0.502	0.813

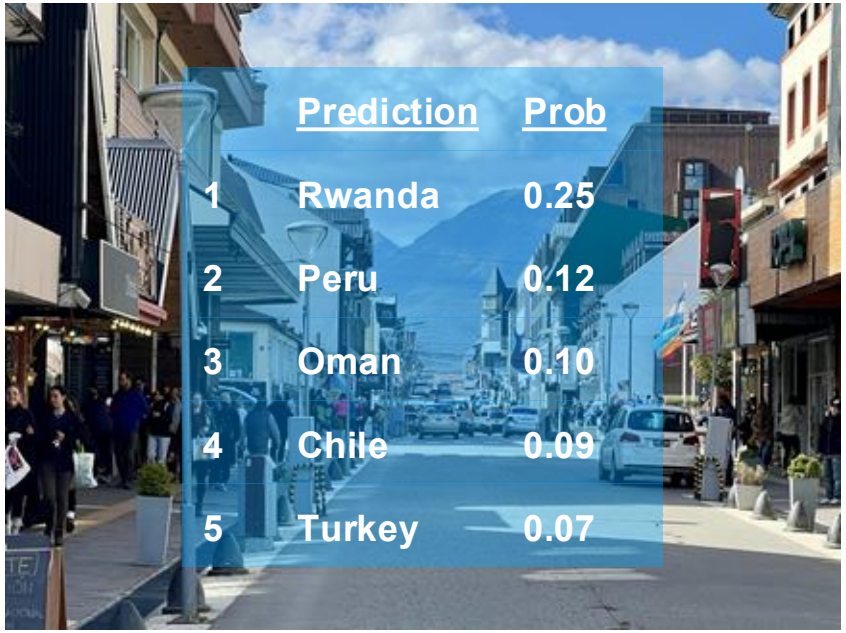




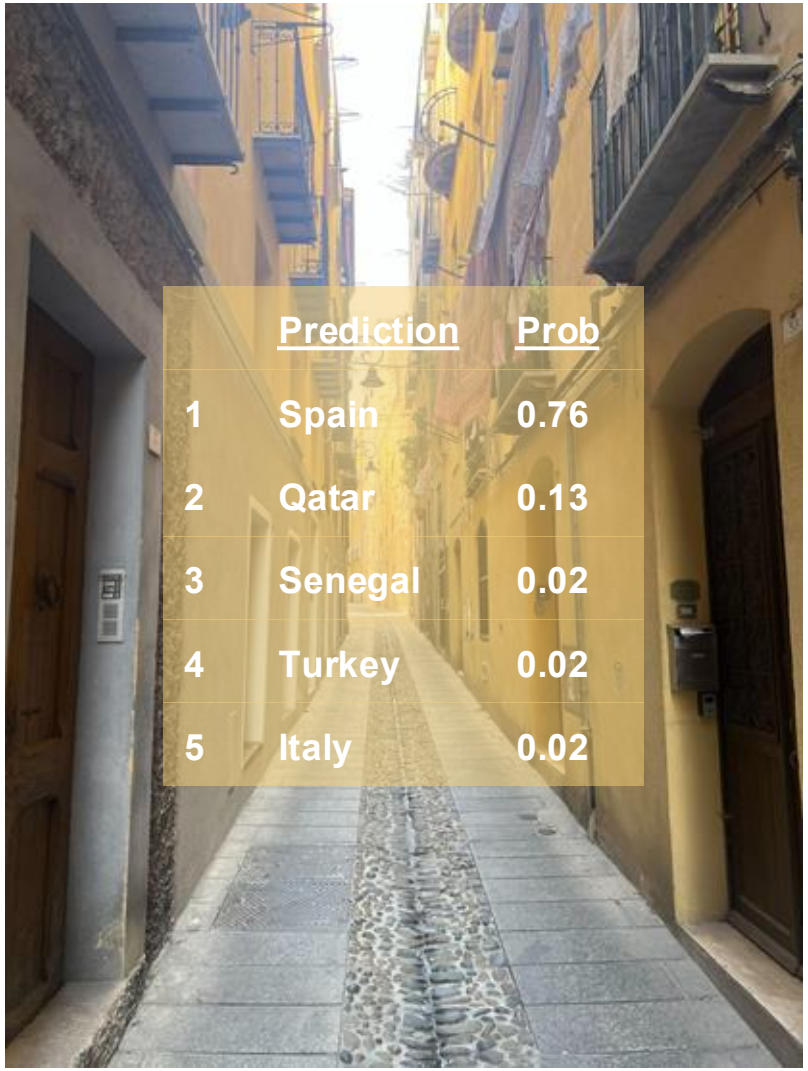
Where did Troels go on vacation?

Where did Troels go on vacation?

	Prediction	Prob
1	Lebanon	0.24
2	Italy	0.14
3	Spain	0.13
4	Turkey	0.10
5	Greece	0.05



	Prediction	Prob
1	Rwanda	0.25
2	Peru	0.12
3	Oman	0.10
4	Chile	0.09
5	Turkey	0.07



	Prediction	Prob
1	Spain	0.76
2	Qatar	0.13
3	Senegal	0.02
4	Turkey	0.02
5	Italy	0.02

Conclusion

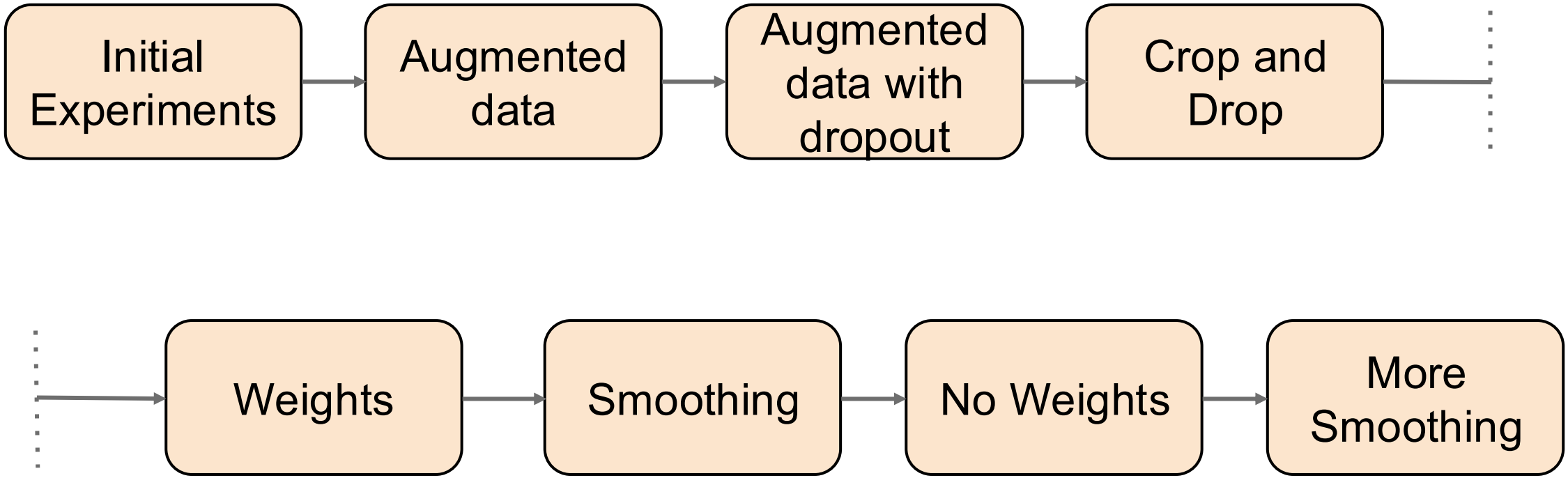
- Neural networks will follow the path of least resistance (car artefacts)
- On-the-fly random data augmentation is crucial for performance
- Class weighting and label smoothing did not improve prediction
- Cluster-based setup reduced complexity and improved separability
- … and don't waste time connecting to Colab GPU if you are not running!!

Future directions:

- Bigger data set (5 million images) with coordinates and more (https://osv5m.github.io/)
- *Really* deep and long training to learn details of road signs etc.

APPENDIX

Model Development Timeline



1. Augmentation and initial optimization

Optuna on ResNet34 with 4 trainable layers
20 Optuna trials, 5 epochs each

Results:

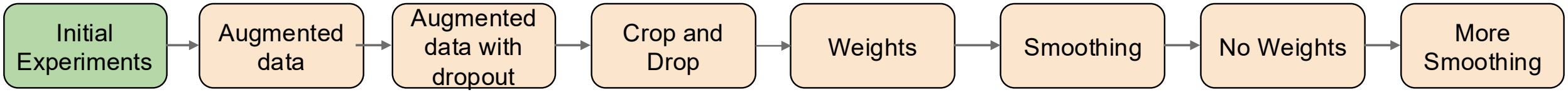
Best validation loss: 3.703

Best hyperparameters:
- Learning rate: 0.0019
- Weight decay: 9.26e-6

Same HP was used to train a ResNet50 model

ResNet50 Test Accuracy	
Top 1	Top 5
0.487	0.779

Test Accuracy
After training the best model is used to predict countries on the test data.
The accuracy on this is the test accuracy

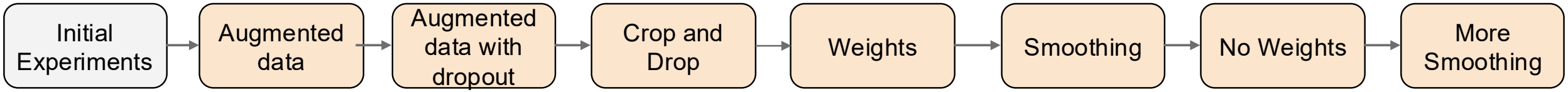


CPU Bottleneck

- Data augmentation caused a CPU bottleneck for smaller resnets and few trainable layers.
- Crop + Resize was the most expensive operation, due to interpolation.
- So we just cropped a 224x224 random region from our compressed 500x281 images.

- JPG compression helped reduce size of dataset substantially.
- JPG → tensor conversion was just done on-the-fly during training, since it was relatively cheap.

- Used torch.profiler, nvidia-smi and timing the dataloaders to investigate CPU bottleneck.



Augmentation data

We tried to avoid the CPU completely by augmenting data once and for all, so it could be loaded straight onto RAM. We knew it could have an effect on performance. Same HP were used.

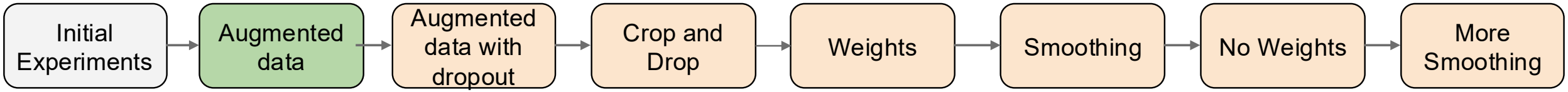
The conclusion was clear: High degree of overfitting, data augmentation is key for performance!

It makes sense: Our dataset is not huge, only ~300 datapoints per class.

We introduced dropout to reduce overfitting and decided to move forward with on-the-fly augmentation: Crop, color jitter and rotation.

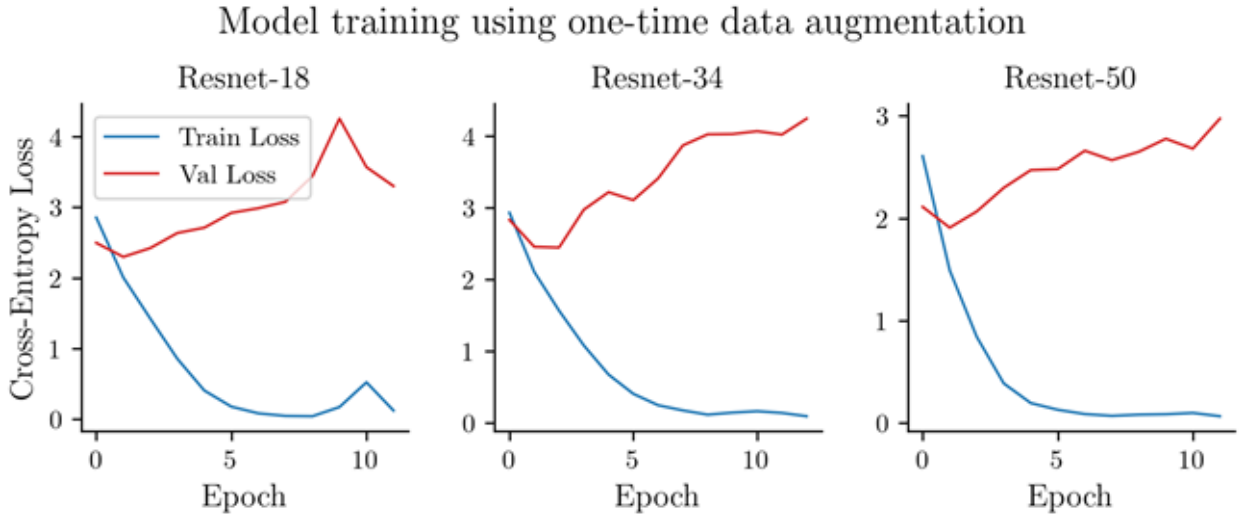
ResNet50 Test Accuracy	
Top 1	Top 5
0.464	0.744

Top 5 Accuracy
The model gives a probability to every country. If the true country is in the top 5 highest probabilities it is in the top 5 accuracy.



Random data augmentation is key for performance

- Each image were cut into three regions.
- 75,000 images for instant loading, instead of on-the-fly augmentation.
- Pre-computed augmentation → severe overtraining



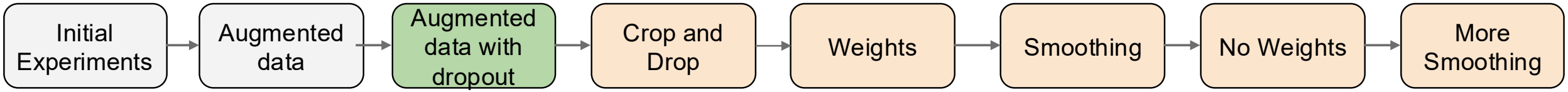
Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → Weights → Smoothing → No Weights → More Smoothing

Augmentation data + Dropout

ResNet50 Test Accuracy	
Top 1	Top 5
0.473	0.757

- Tried to reduce overfitting using dropout
  Used dropout rate = 0.3

- Test accuracy did not improve significantly.
- Overfitting was still observed during training.



Augmentation on the go + Dropout

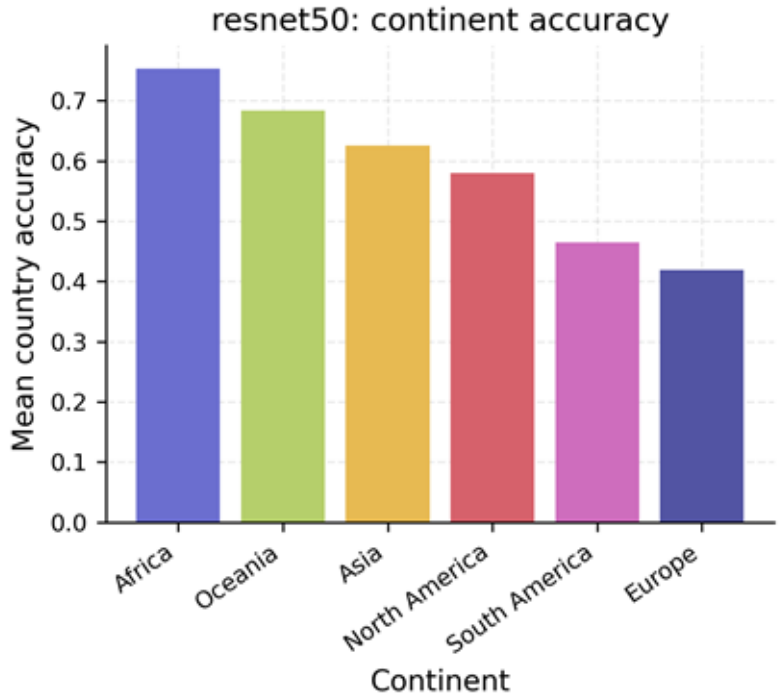
ResNet50 Test Accuracy	
Top 1	Top 5
0.541	0.838

Did not work as hoped
Went back to model where we cropped along the way and now also included dropout

HP parameters from initial optimization was used as

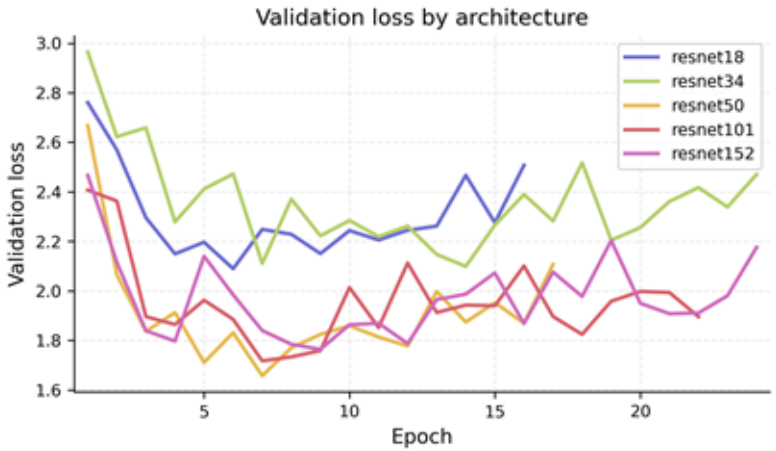
Conclusion:

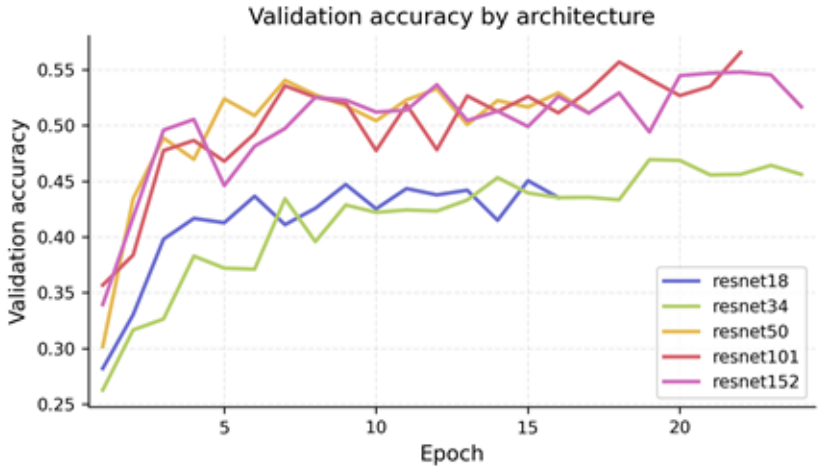
Dropout is good
European countries is hard to predict

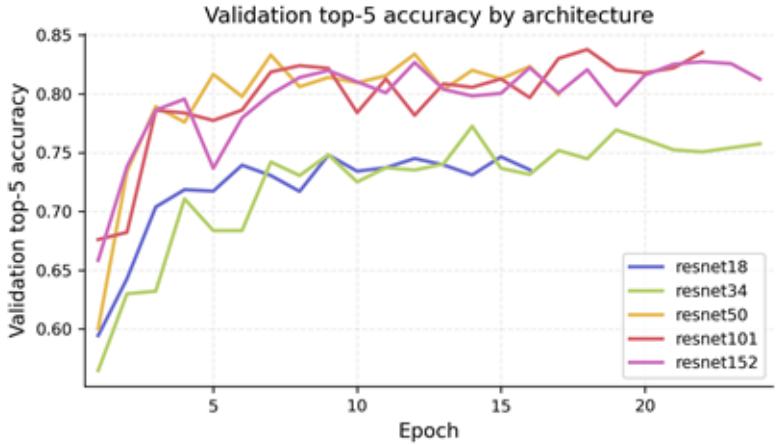


Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → Weights → Smoothing → No Weights → More Smoothing

Augmentation on the go + Dropout - Training plots







Observations:

- Validation accuracy improved quickly during the first epochs for all architectures.
- Larger models generally achieved higher validation accuracy and top-5 accuracy.

- ResNet50 showed strong and stable performance while remaining significantly cheaper to train than ResNet101 and ResNet152.
- ResNet50 was therefore selected as the main model for the later experiments due to its balance between performance, stability, and computational efficiency.

Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → Weights → Smoothing → No Weights → More Smoothing

Adding Class Weights

ResNet50 Test Accuracy	
Top 1	Top 5
0.518	0.817

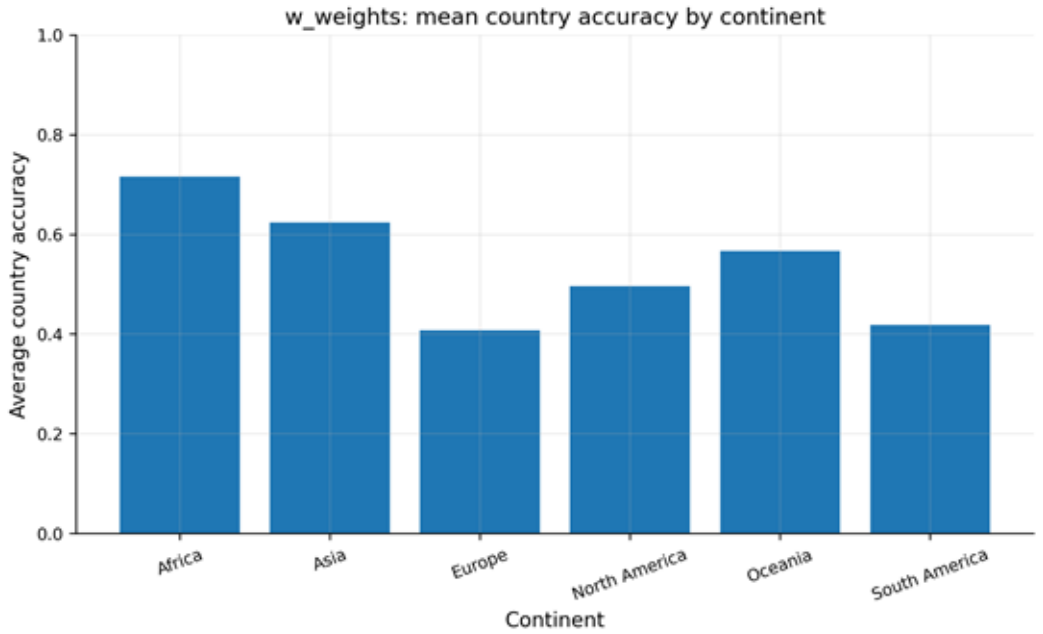
For European countries the CrossEntropy loss was multiplied with 1.5

Initial try:

HP parameters from initial optimization was used as

Results:

- Accuracy got a bit worse
- Avg accuracy for Europe did not improve



Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → **Weights** → Smoothing → No Weights → More Smoothing

Adding Class Weights - Optimization

ResNet50 Test Accuracy	
Top 1	Top 5
0.542	0.838

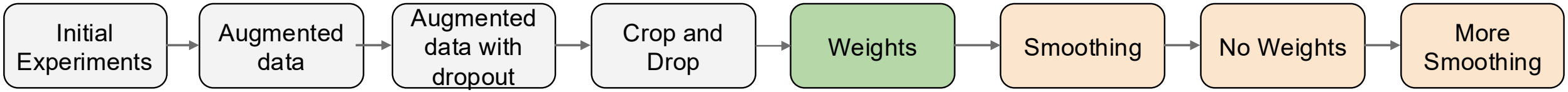
Grid search:

Model	Batch Size	Learning rate	Weight decay	Trainable layers	Best val loss	Best Val accuracy	Test top 1	Test top 5
ResNet50	256	0.001	0.001	1	1.803	0.502	0.518	0.820

And then Optuna

Model	Batch Size	Learning rate	Weight decay	Trainable layers	Test top 1	Test top 5
ResNet50	256	0.00087	0.00001	1	0.542	0.838

Seemed like weights are improving the results



With Class Weights - Optimization Loop

ResNet50 Test Accuracy	
Top 1	Top 5
0.518	0.817

Motivation:

To evaluate which ResNet is best

First tried with ResNet50's HP but the results seemed biased

model	test_top1	test_top5
resnet18	0.4467833936	0.7631320357
resnet34	0.4646862149	0.7654927969
resnet50	0.5107220411	0.8170371652
resnet101	0.5075742602	0.8101514578
resnet152	0.5221325755	0.8152665496

Execution was looping over ResNet(34, 50, 101, 152)

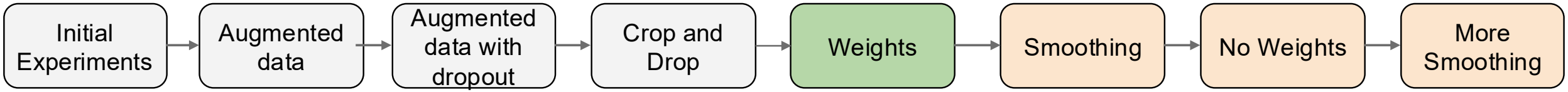
Tested multiple ResNet models with weighted loss and Optuna-tuned:

- learning rate: `1e-4 – 2e-3`
- weight decay: `1e-6 – 1e-4`
- trainable layers: `0, 1, 2`

Then retrained each model with the best settings and evaluated both:

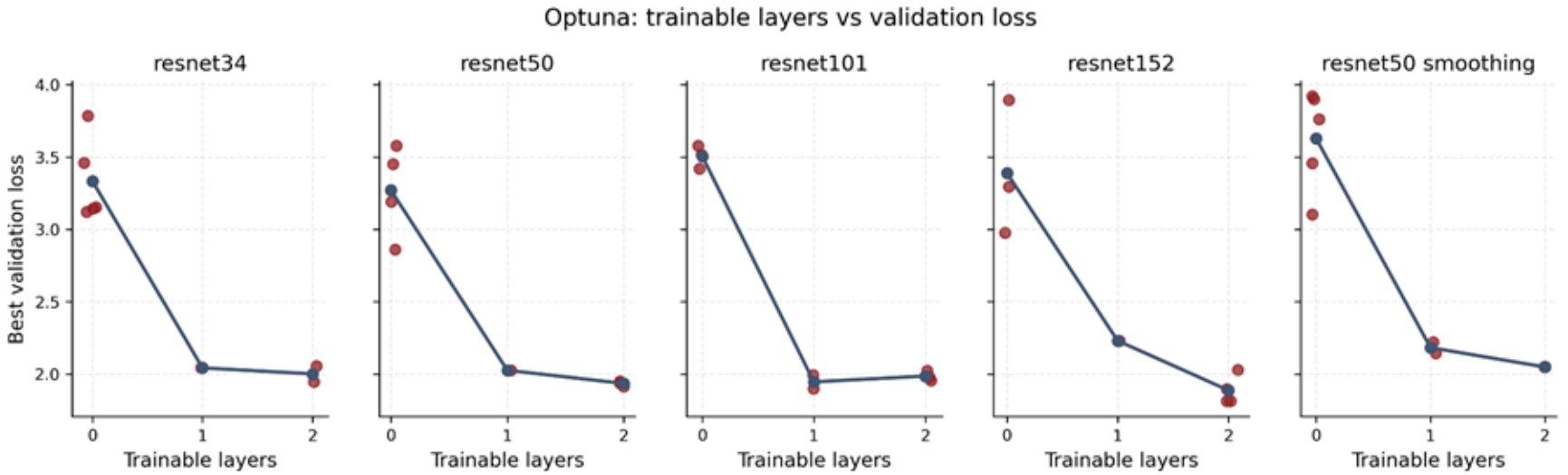
- best validation loss checkpoint
- best validation accuracy checkpoint

Saved test accuracy, top-5 accuracy, confusion matrices, predictions, and continent accuracy for both a best loss model and a best accuracy model.



With Class Weights - Optuna Results

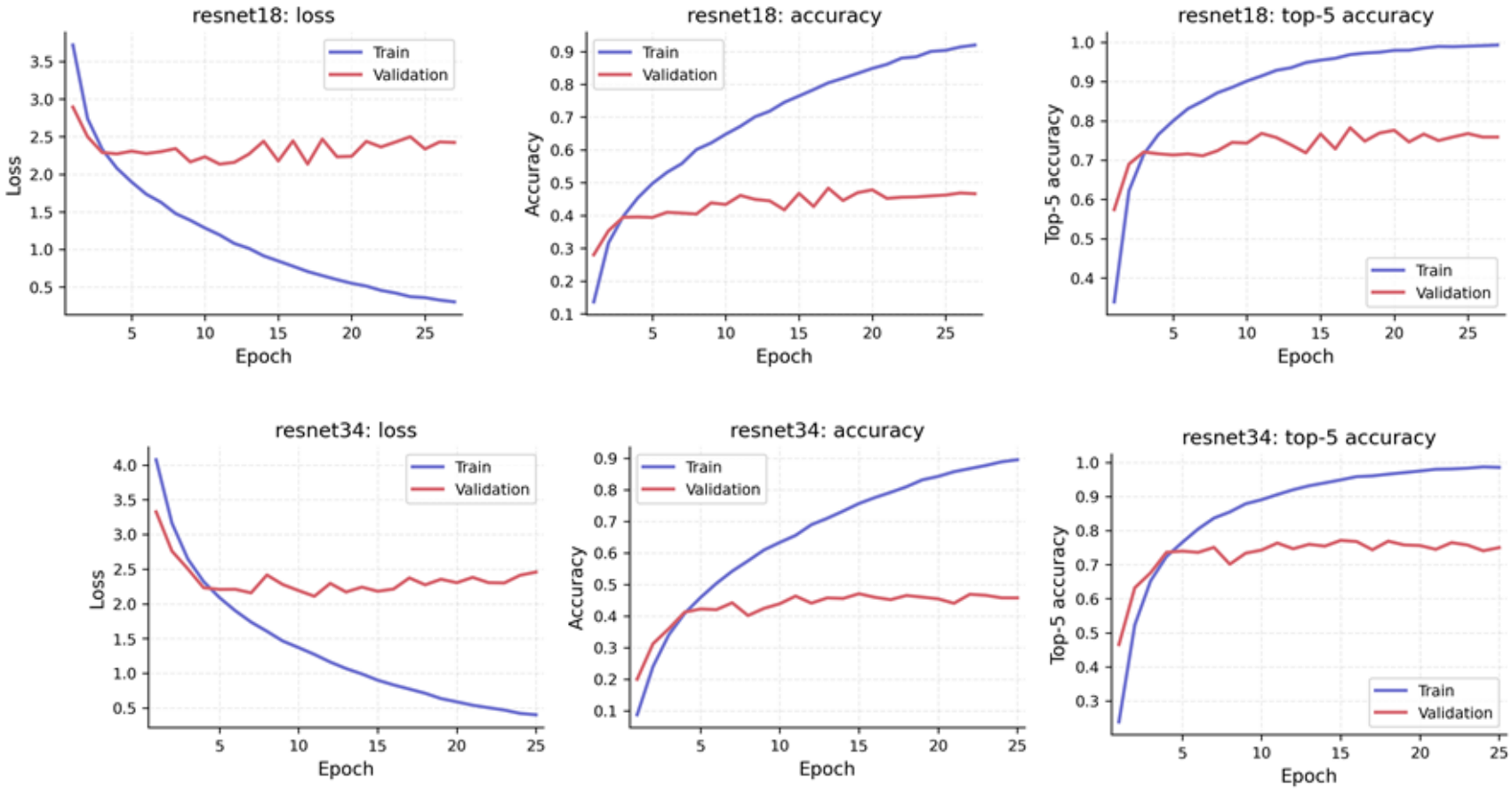
model	best_val_loss	learning_rate	weight_decay	trainable_layers
resnet101	1.8983	7.55e-04	1e-06	1
resnet152	1.8118	9.98e-04	1e-05	2
resnet34	1.9447	1.23e-04	5e-05	2
resnet50	1.9145	3.95e-04	1e-06	2
resnet50 smoothing	2.0483	2.32e-04	1e-06	2



The previous optuna suggested that one layer was best. Which was why only [0, 1, 2] was tested.
In hindsight, trainable layer setting 3 should likely also have been tested.

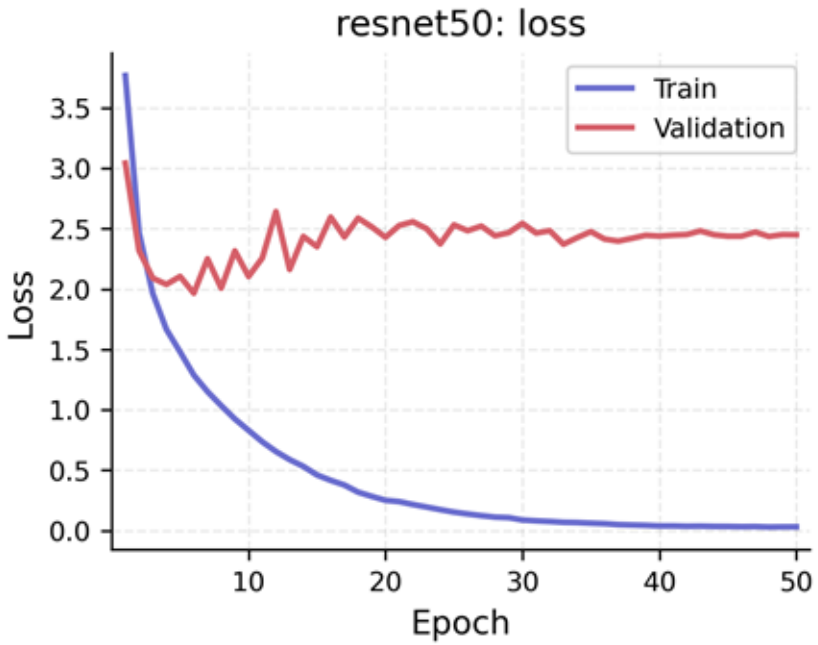
Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → **Weights** → Smoothing → No Weights → More Smoothing

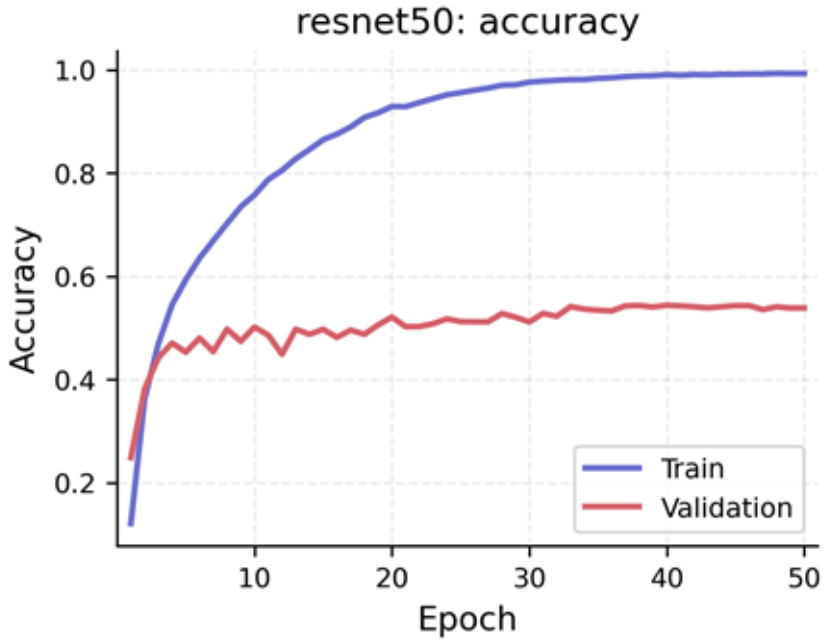
With Class Weights - ResNet(18 + 34)

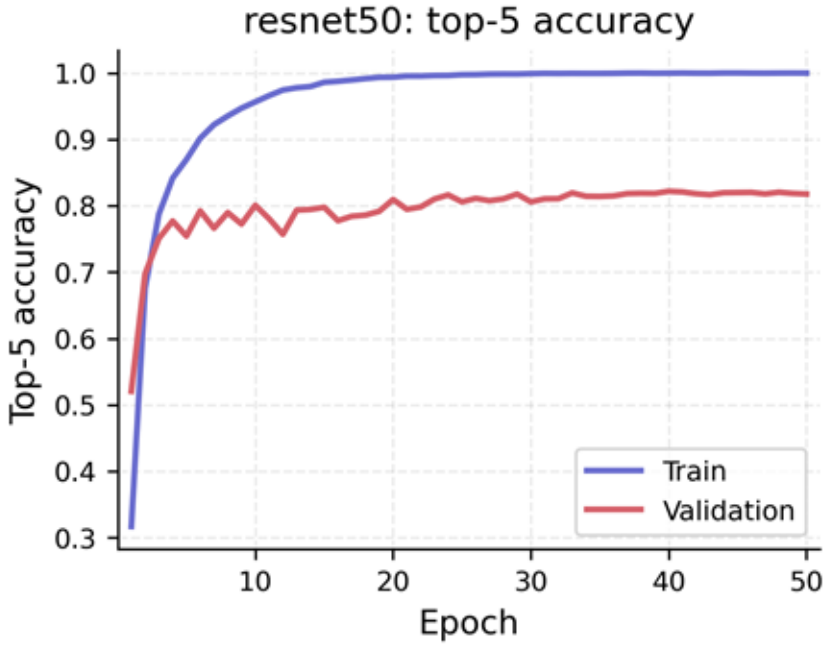


Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → **Weights** → Smoothing → No Weights → More Smoothing

With Class Weights - Resnet 50

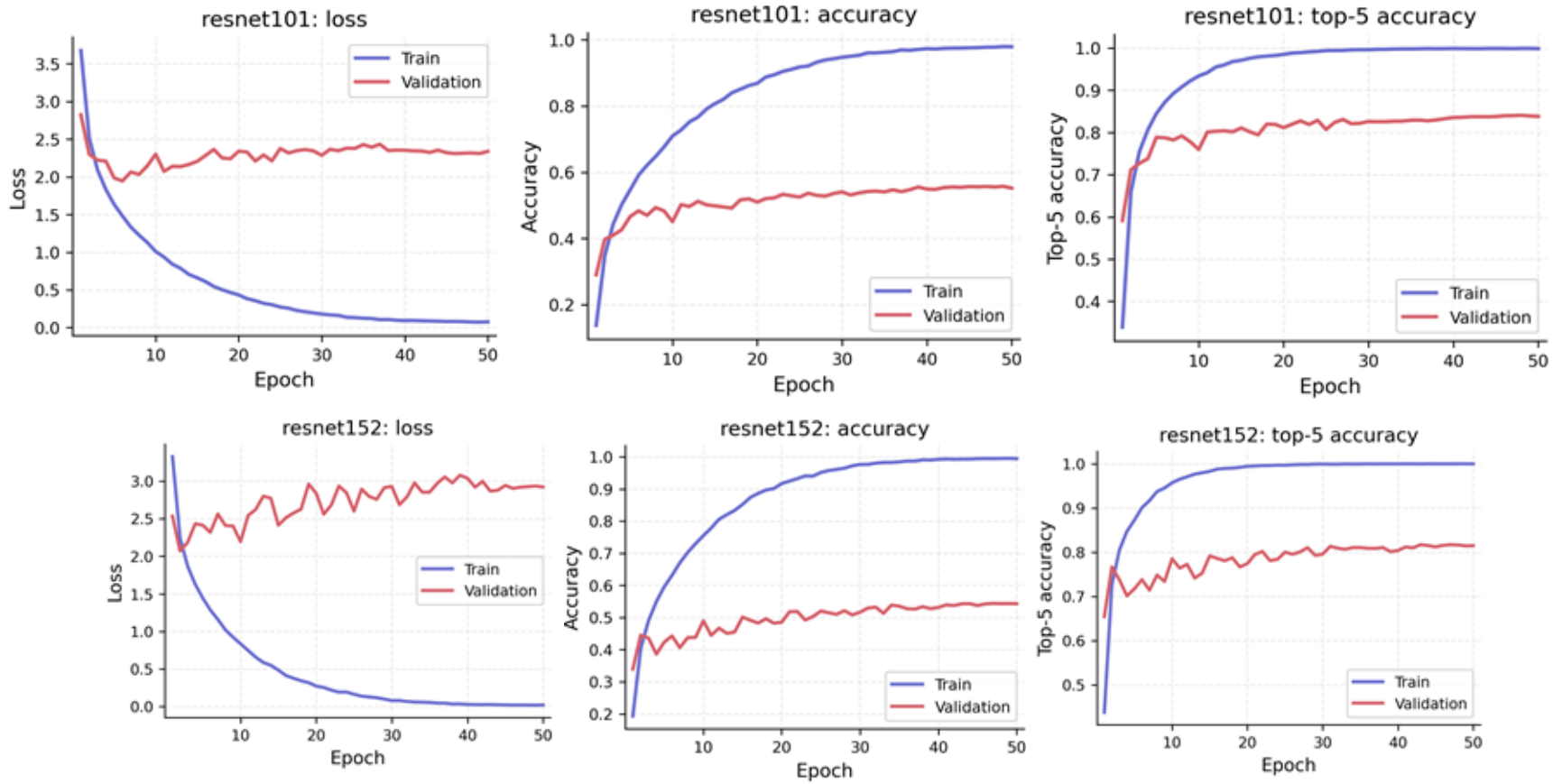






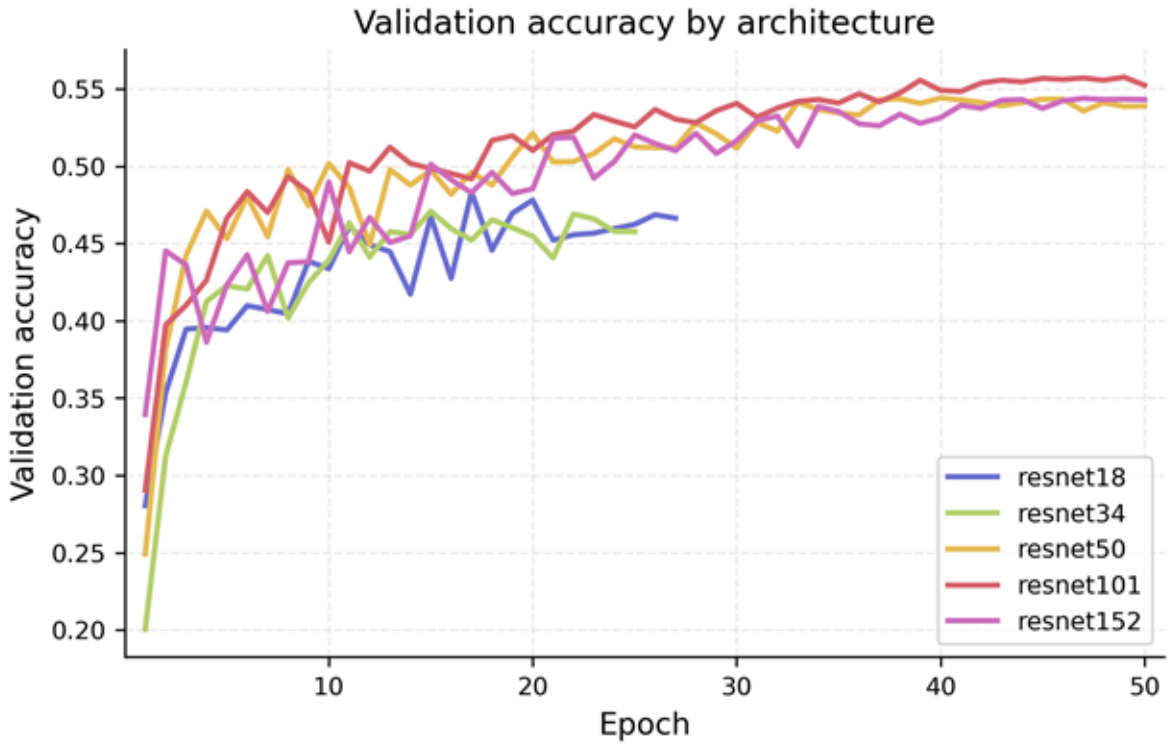
Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → **Weights** → Smoothing → No Weights → More Smoothing

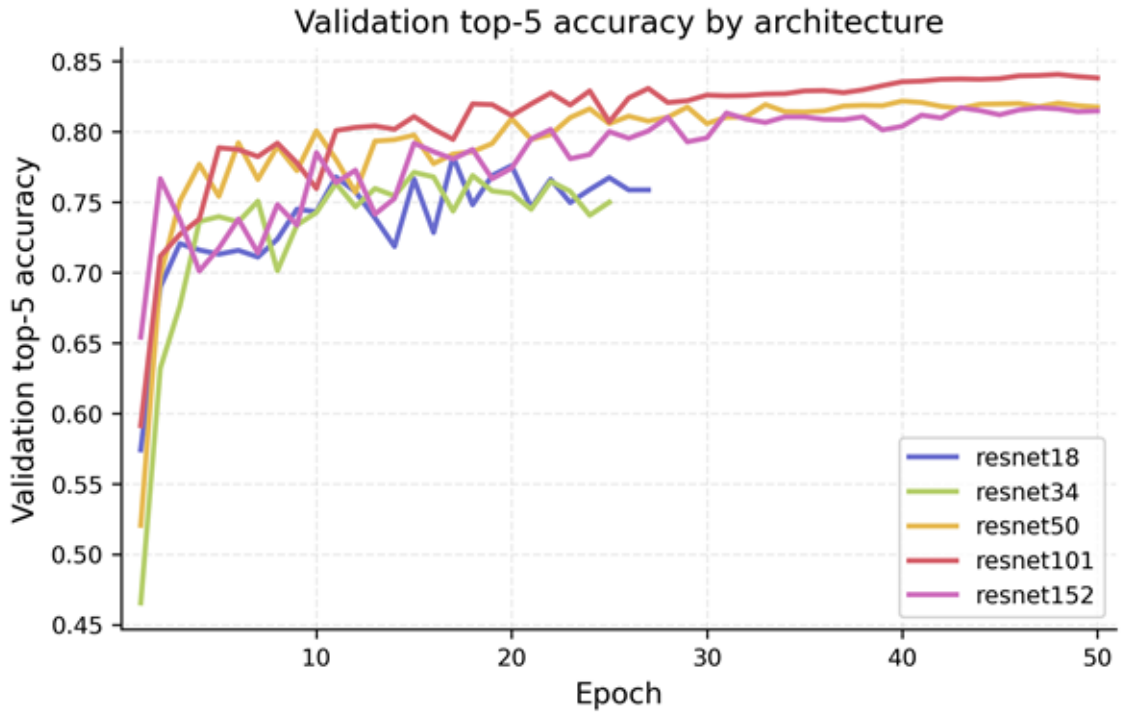
With Class Weights - ResNet(101+152)



Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → **Weights** → Smoothing → No Weights → More Smoothing

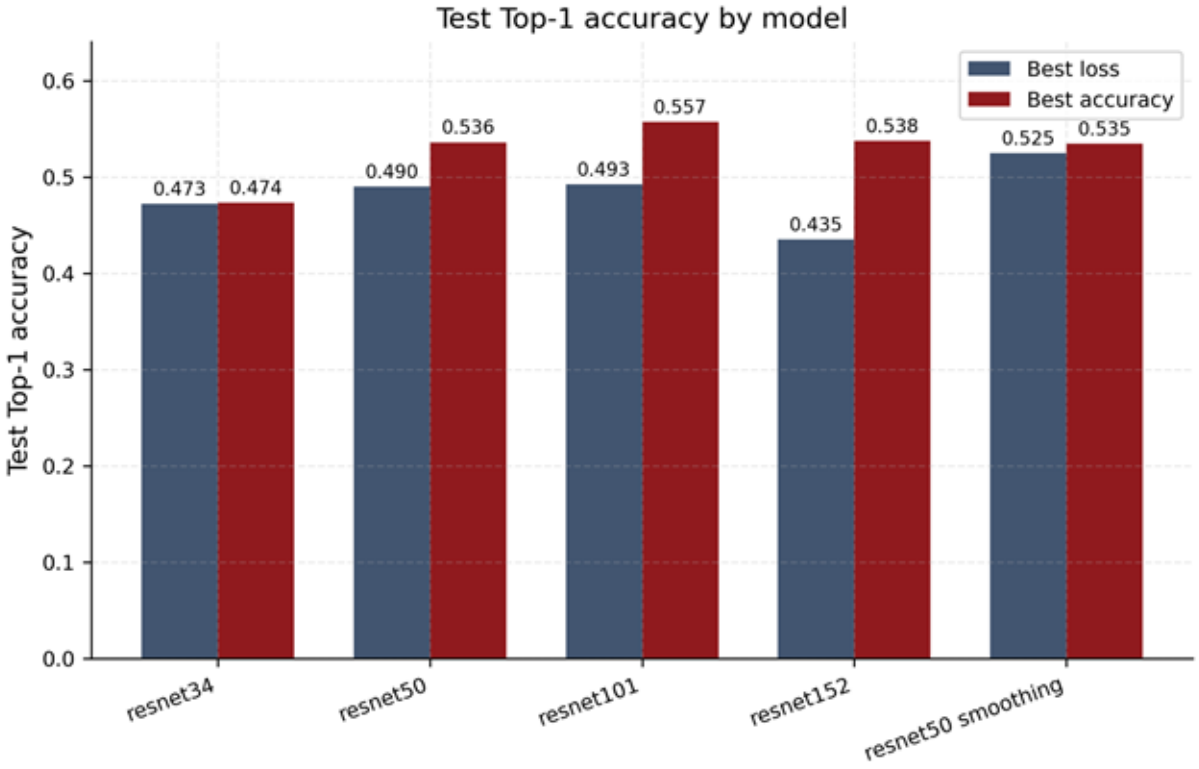
With Class Weights - Accuracy for all RestNet

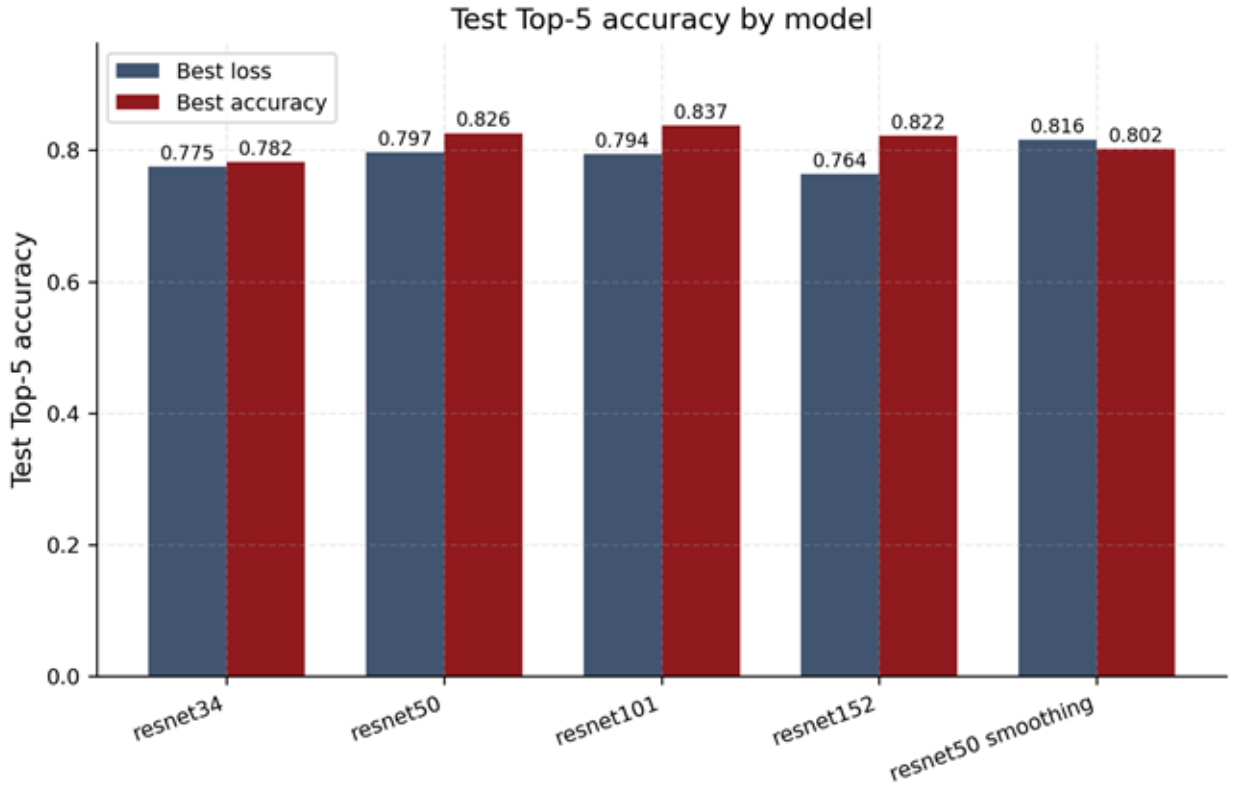




Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → **Weights** → Smoothing → No Weights → More Smoothing

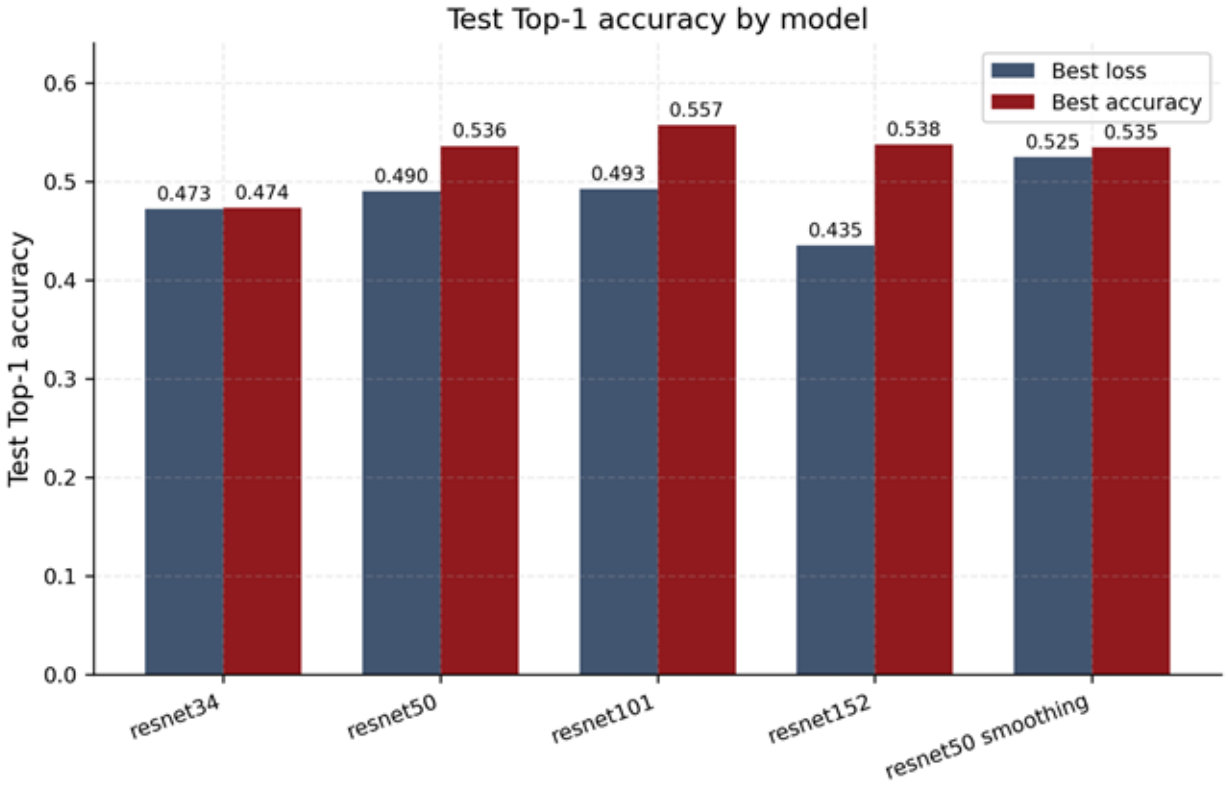
With Class Weights - Test Accuracy for all RestNet





Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → **Weights** → Smoothing → No Weights → More Smoothing

With Class Weights - Conclusion



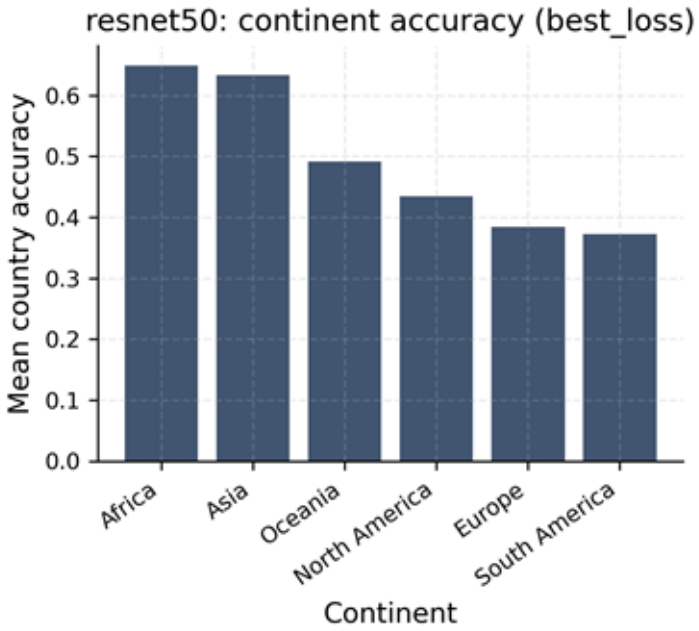
From these results it is clear that when training the the accuracy keeps increasing while loss quickly finds minima.

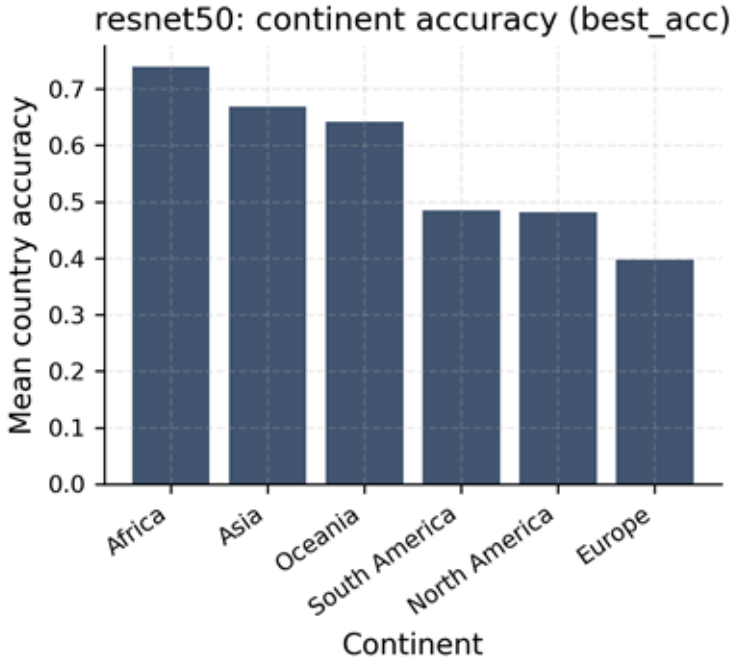
The Crop and Drop did not experience this. But we did not do HP tuning on this, which maybe could explain the differences.

However it could also be because Europe is hard to learn which penatalizes the loss a lot.

Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → **Weights** → Smoothing → No Weights → More Smoothing

With Class Weights - Conclusion





Here we see that using the best validation loss checkpoint results in relatively low average accuracy across all continents compared to Crop and Drop.

Using the best validation accuracy checkpoint instead gives much better test accuracy for all continents except Europe.

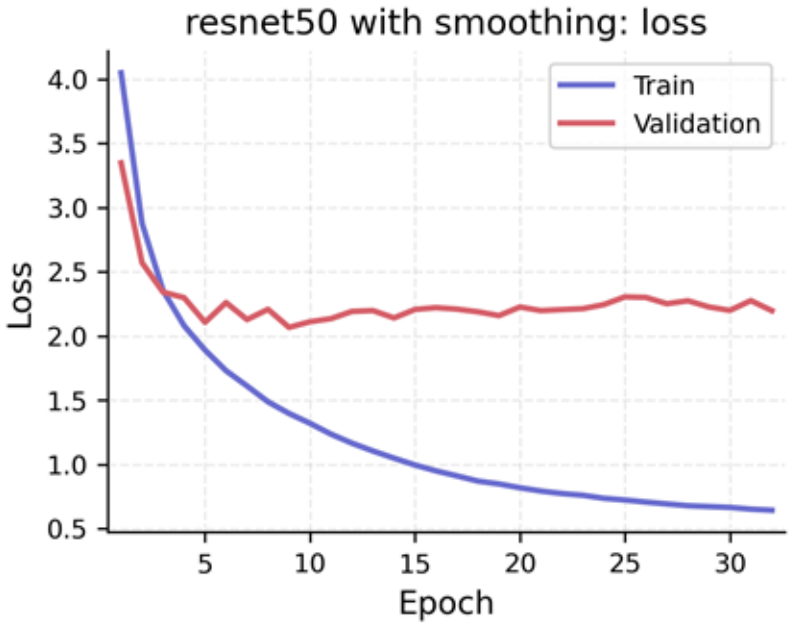
This suggests that the weighted loss may no longer be a reliable metric for selecting the best model.

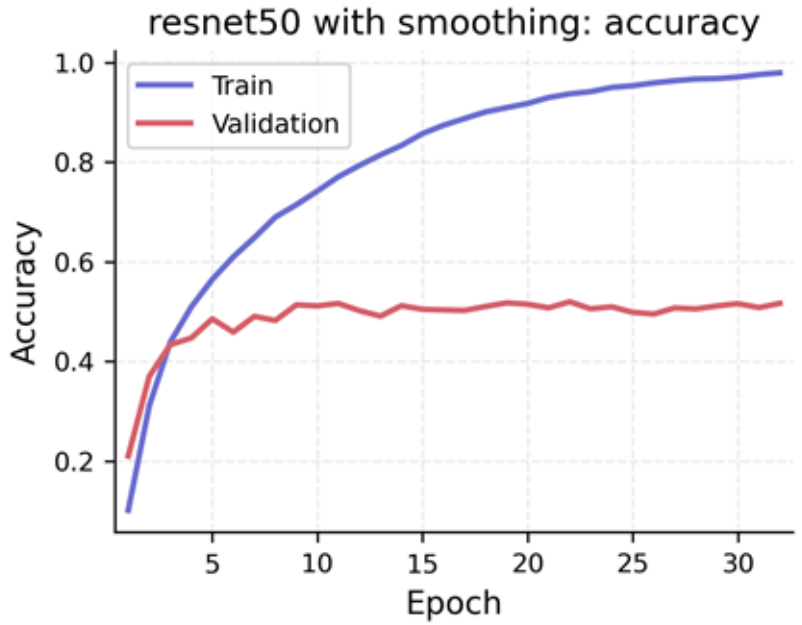
Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → **Weights** → Smoothing → No Weights → More Smoothing

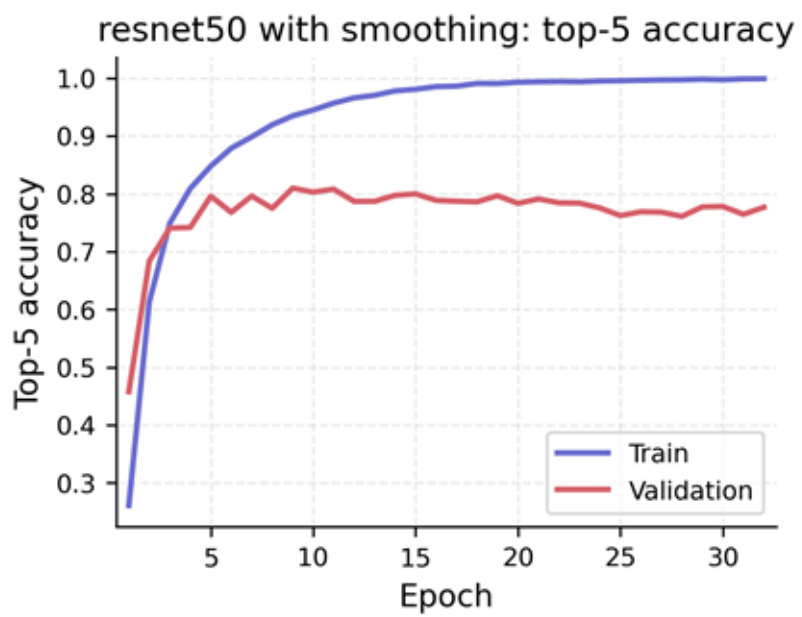
Smoothing with ResNet50

Added smoothing by using cluster (k=15) as explained in slide X.
Optuna was run like with the weights

ResNet50 Test Accuracy	
Top 1	Top 5
0.525	0.816

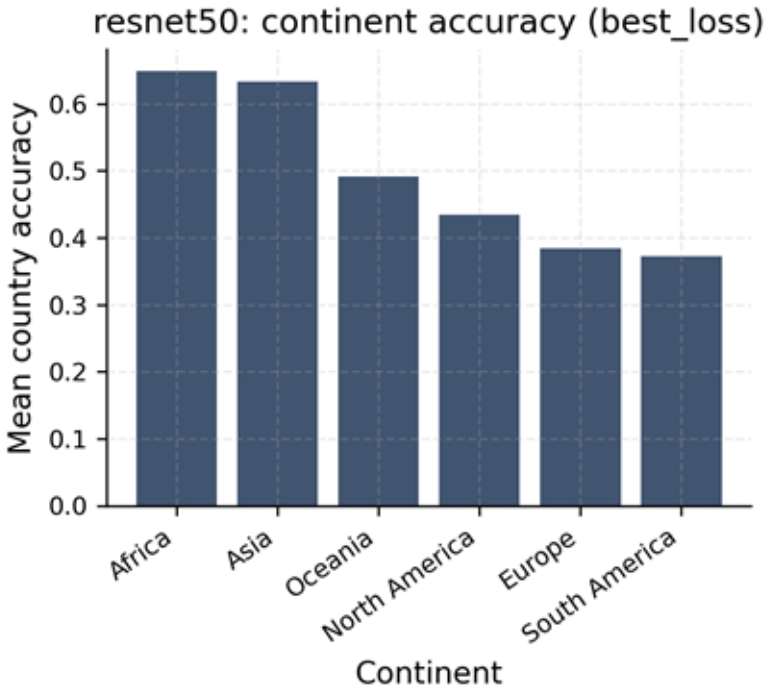


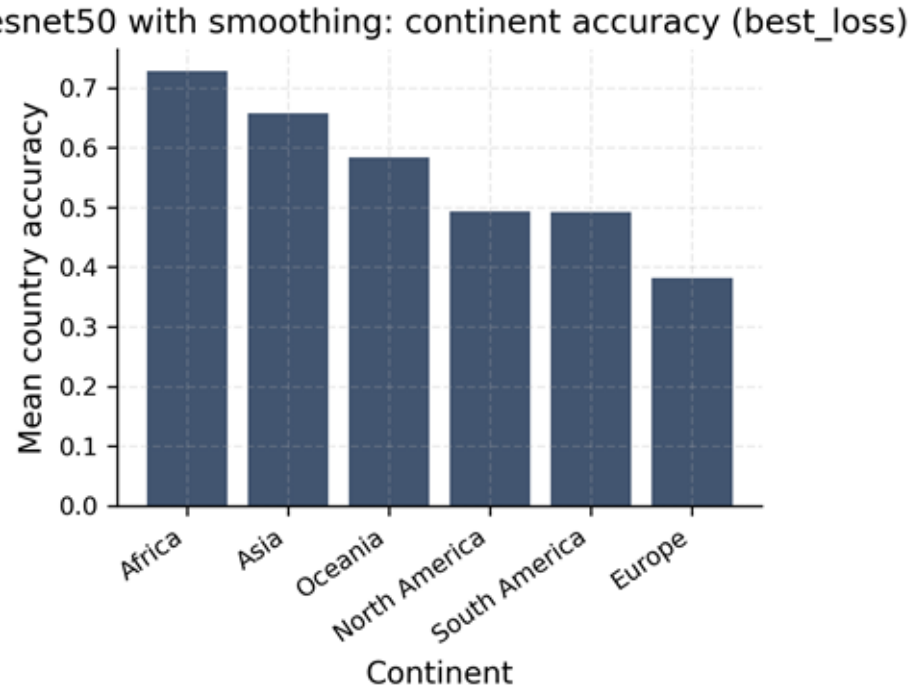




Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → Weights → Smoothing → No Weights → More Smoothing

Smoothing with ResNet50 - weights vs smoothing



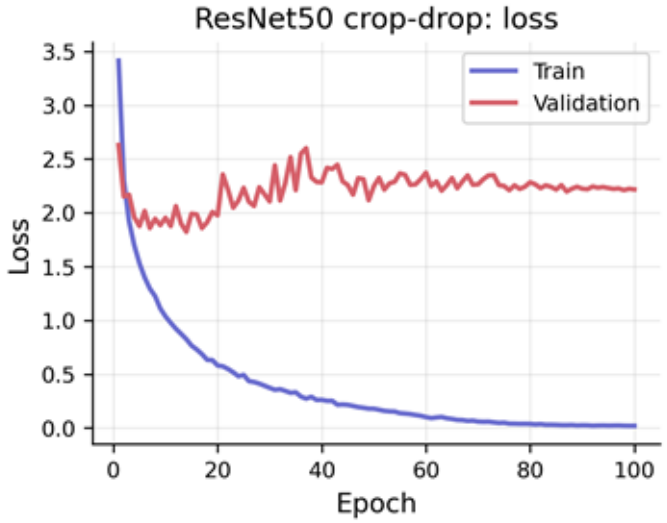


Here it becomes obvious that when using weights the best loss model does not predict as well.
Also using smoothing does not change the results a lot from using Crop_and_Drop

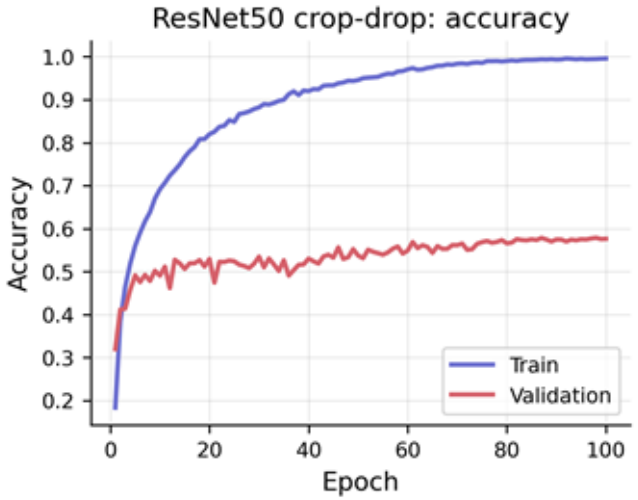
Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → Weights → Smoothing → No Weights → More Smoothing

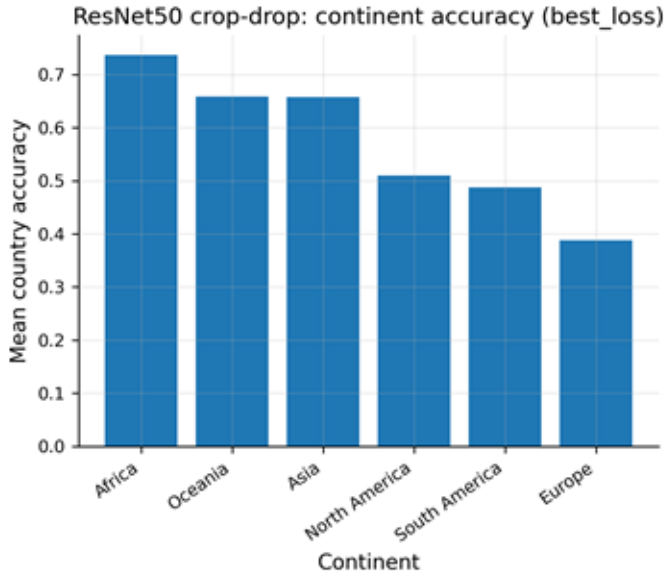
Removing Weights - Initial

- Back to the crop-and-drop model
- Used hyperparameters from the previous grid search.
- Validation accuracy continued improving despite increasing validation loss.
- This suggests that the effect may not only be caused by class weights, but also by dataset imbalance and some classes overfitting earlier than others.



ResNet50 Test Accuracy	
Top 1	Top 5
0.518	0.817





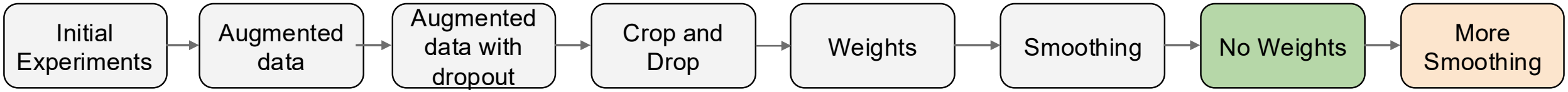
Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → Weights → Smoothing → No Weights → More Smoothing

Removing Weights - HP tuning

HP tuning with optuna for
ResNet50 and ResNet 101
8 trials of 15 epoch
Training with 50 epoch and
early stopping 8

ResNet50 Test Accuracy	
Top 1	Top 5
0.499	0.804

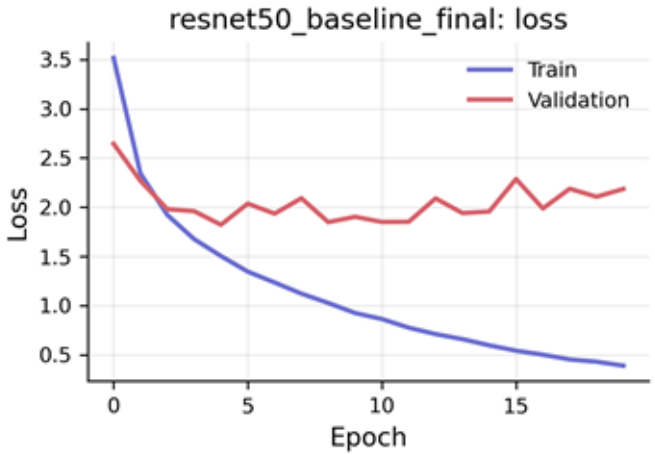
experiment	model	checkpoint_type	test_top1	test_top5	learning_rate	weight_decay	trainable_layers
baseline	resnet50	best_loss	0.4989179671	0.8046429157	0.0008449129418	5.00E-06	1
baseline	resnet50	best_acc	0.5453472137	0.8235294223	0.0008449129418	5.00E-06	1
baseline	resnet101	best_loss	0.5374778509	0.8310053349	0.0004069368009	5.00E-05	3
baseline	resnet101	best_acc	0.5508558154	0.8347432613	0.0004069368009	5.00E-05	3

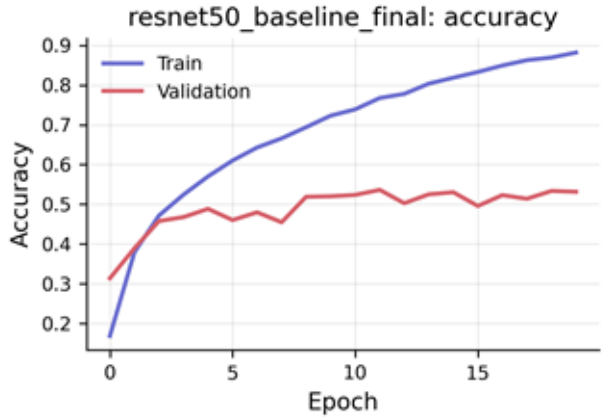


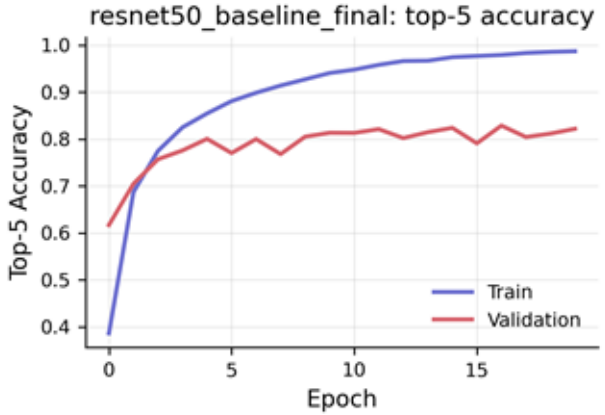
Removing Weights - HP tuning Results ResNet50

- The results suggests that the hyperparameter search should have been better. However the search was limited due to compute constraints.
- Earlier experiments with class weights required a large amount of compute resources, which reduced the remaining budget for further tuning.
- Validation accuracy again continued improving while validation loss increased.
- This suggests that the mismatch between loss and accuracy may not only be caused by class weights, but could also be related to dataset imbalance and uneven learning across classes.

ResNet50 Test Accuracy	
Top 1	Top 5
0.499	0.804



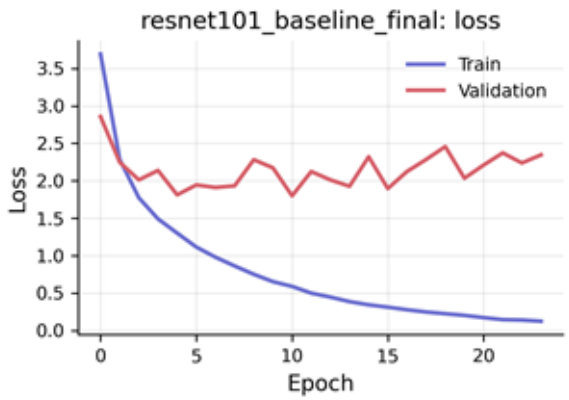


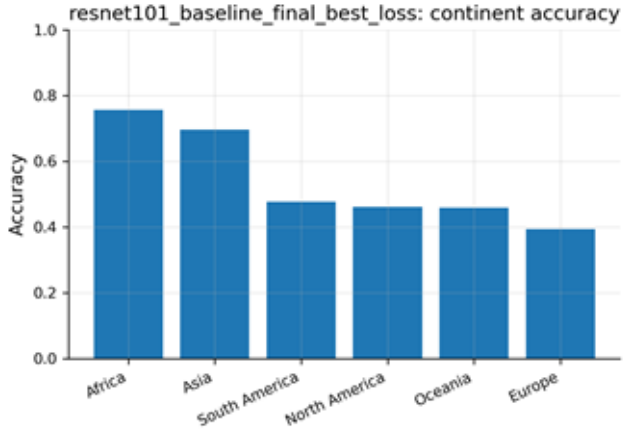


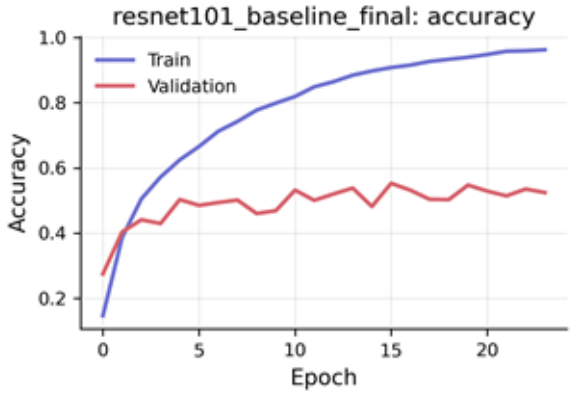
Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → Weights → Smoothing → No Weights → More Smoothing

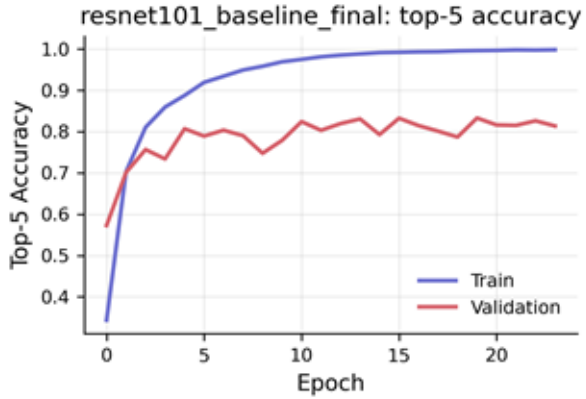
Removing Weights - HP tuning Results ResNet101

ResNet101 Test Accuracy	
Top 1	Top 5
0.537	0.831









- Here the accuracy improved
  - The HP search was properly better
- Validation accuracy again continued improving while validation loss increased.

Initial Experiments → Augmented data → Augmented data with dropout → Crop and Drop → Weights → Smoothing → No Weights → More Smoothing

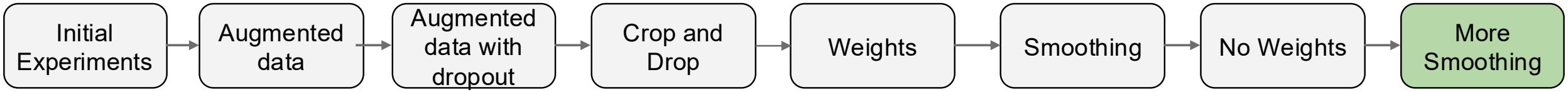
Trying Smoothing again

Different prob and Clusters
Did not improve

- Maybe not the best to use clusters. Another experiment could be to for each country smooth with the 3 closest countries (it seems like too many countries becomes a problem)

Model	Cluster k	Correct Prob	Test top 1	Test top 5
ResNet50	30	0.9	0.486	0.790
ResNet50	30	0.8	0.509	0.789
ResNet50	15	0.9	0.525	0.816
ResNet50	15	0.8	0.486	0.785

Cluster 15 and 30 made from temp and geo data



What have we learned from these “experiments”

Findings

- The best performing model was the crop-and-drop ResNet50 model with initial HP and the tuned ResNet50 model with weights (not loop).
- Dataset imbalance appears to affect training significantly.
- Some countries may begin overfitting earlier than others, causing validation loss to increase while validation accuracy still improves.
- Hyperparameter optimization was important before comparing models fairly.

Class weights

- Adding higher weights to Europe did not improve performance.
- This may be because Europe was already difficult to predict, causing the optimization to focus too strongly on those classes.
- The chosen weight value (1.5) may also have been too large.
- Different weighting strategies could still be explored further.

Label smoothing

- Cluster-based label smoothing did not improve performance.
- One possible issue is that some clusters contained too many countries, weakening the learning signal.
- A potentially better approach could be smoothing each country only toward its 2–3 closest neighboring countries.

Dataset limitations

- While the model is far from perfect, a top-1 accuracy around 0.5 is still substantially better than random guessing across 85 countries.
- The dataset may simply be too imbalanced to achieve much higher performance.
- The training images mainly consisted of road scenes and may not have been diverse enough.
- Cars and road infrastructure may therefore have had a strong influence on predictions.

Future work

- Explore alternative weighting strategies and smaller class weights.
- Use geographically nearest-neighbor smoothing instead of large clusters.
- Improve dataset balance and image diversity.
- Investigate how much cars and road-specific features influence predictions.

What have we learned from these “experiments”

Workflow reflections

- It was important to save all experiments and intermediate results.
- Earlier models needed to be revisited several times, which was only possible because checkpoints, histories, and evaluation files had been stored systematically.
- Some expensive compute usage could likely have been avoided with a clearer experimental plan and by testing potential issues earlier in the process.
- A large amount of compute was spent on experiments with class weights and smoothing before it became clear that the main issue might instead be related to dataset imbalance and the relationship between validation loss and validation accuracy.
- Additional compute units could probably also have been saved by not optimizing all ResNet architectures in parallel from the beginning.
- A potentially better workflow would have been to focus primarily on ResNet50 first, understand the optimization behavior and training dynamics, and then later extend the workflow to the remaining architectures.

Why cluster countries?

Country level classification is difficult because many countries can look visually siimiar.

Instead of predicting 85 individual countries, we grouped countries into broader labels based on metadata,
Here:

- Temperature only
  - Average January temperature
  - Average July temperature
- Climate/geography
  - Average January temperature
  - Average July temperature
  - Absolute Latitude (Because distance from the equator is more relevant for climates structure than north/south direction)
  - Longitude

This reduces the number of classes and allows us to test whether regional climate/geography structure is easier for the CNN to learn.

How the clusters were made

- We used country-level metadata for each of the 85 countries
- Two feature sets were compared:
  - Temperature only: Average January and July temperature
  - Climate/geography: Average January temperature, July temperature, absolute latitude and longitude
- Latitude was converted to absolute latitude, because distance from the equator is more relevant for climate than weather a country is north or south
- Features were standardized before KMeans, so one feature did not dominate only because of its scale

Meta data was used only to group countries into cluster labels

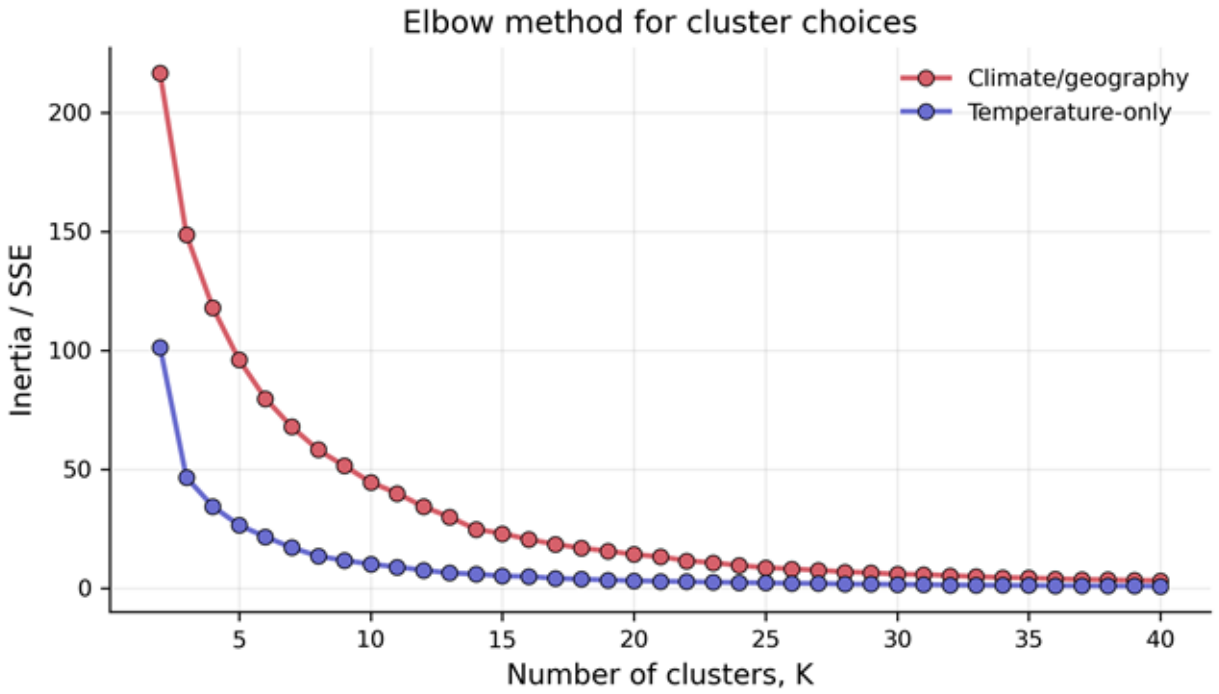
- The CNN did not receive latitude, longitude or temperature as input
- The model only saw images
- Therefore, good cluster performance means that visual image features contain information related to climate/geography

Elbow method / choosing K

The elbow plot was used as a guideline for choosing the number of clusters.
For temperature only clustering, the curve flattens quite early. This suggest that a smaller value of (K) could probably have been sufficient, because the data only constraints two features; Jan and july temp.

However we still used (K=15) for temperature-only clustering in the comparison, because the main climate/geography model was also tested with (K=15). This made the comparison between the two label systems more controlled.

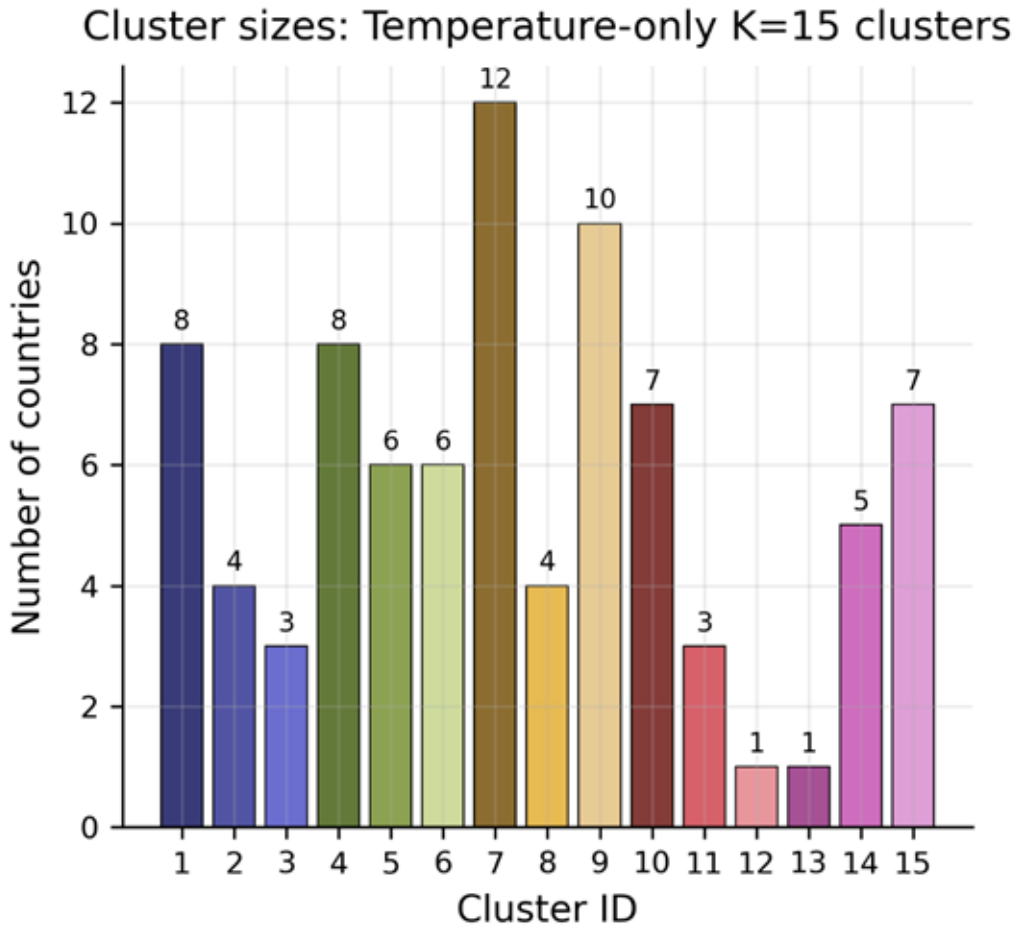
So, one limitation is that temperature-only clustering might have been better evaluated with fewer clusters. But for our main comparison, we kept (K) fixed to focus on the effect of **adding geography**

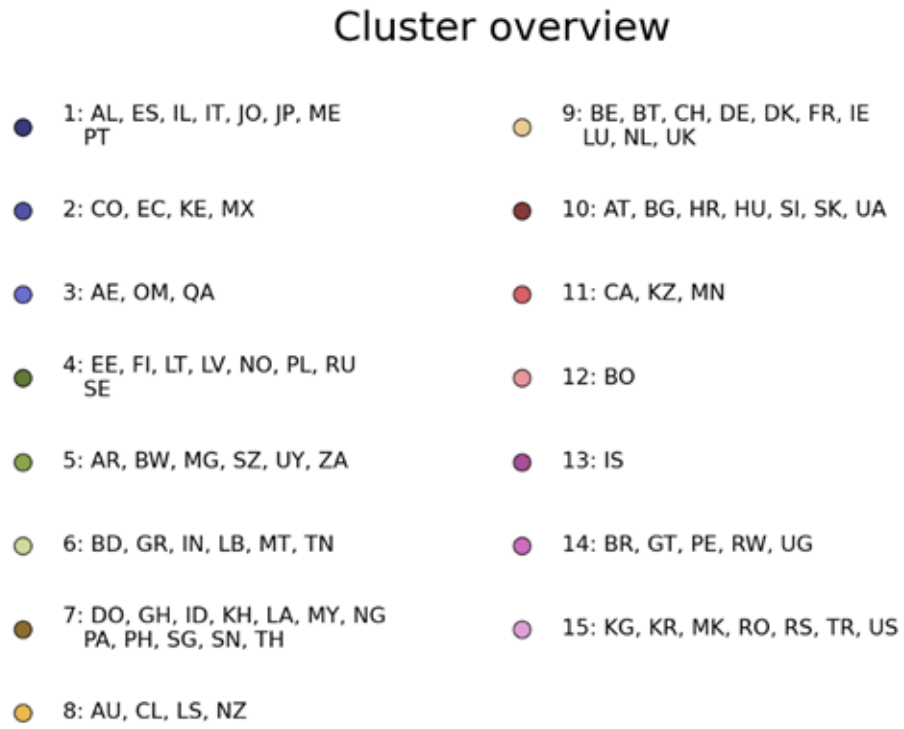


Temperature-only cluster size overview

Countries clustered using only January and July temperature. The plot shows the size of each cluster, while the overview on the right shows which countries are assigned to each group.

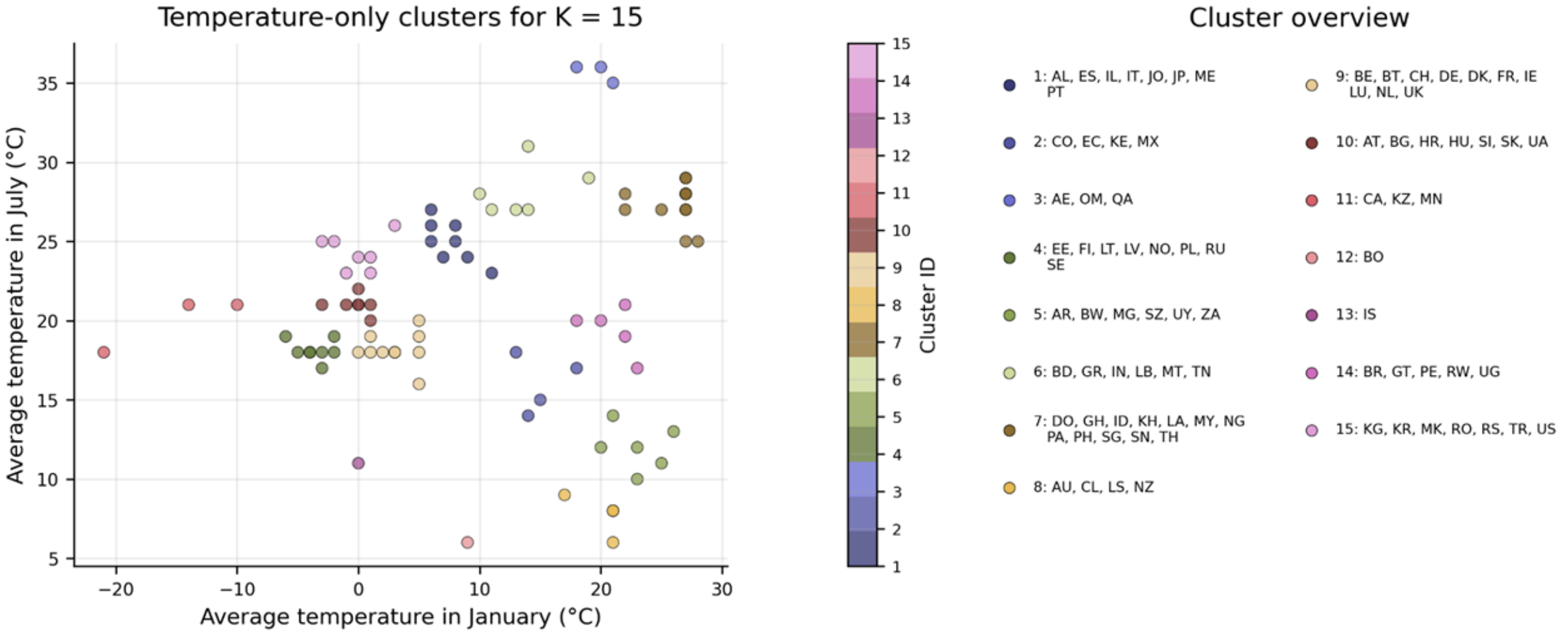
The cluster sizes are uneven, some temp. Patterns are shared by many countries, while others are more specific.





Temperature only clusters in feature space

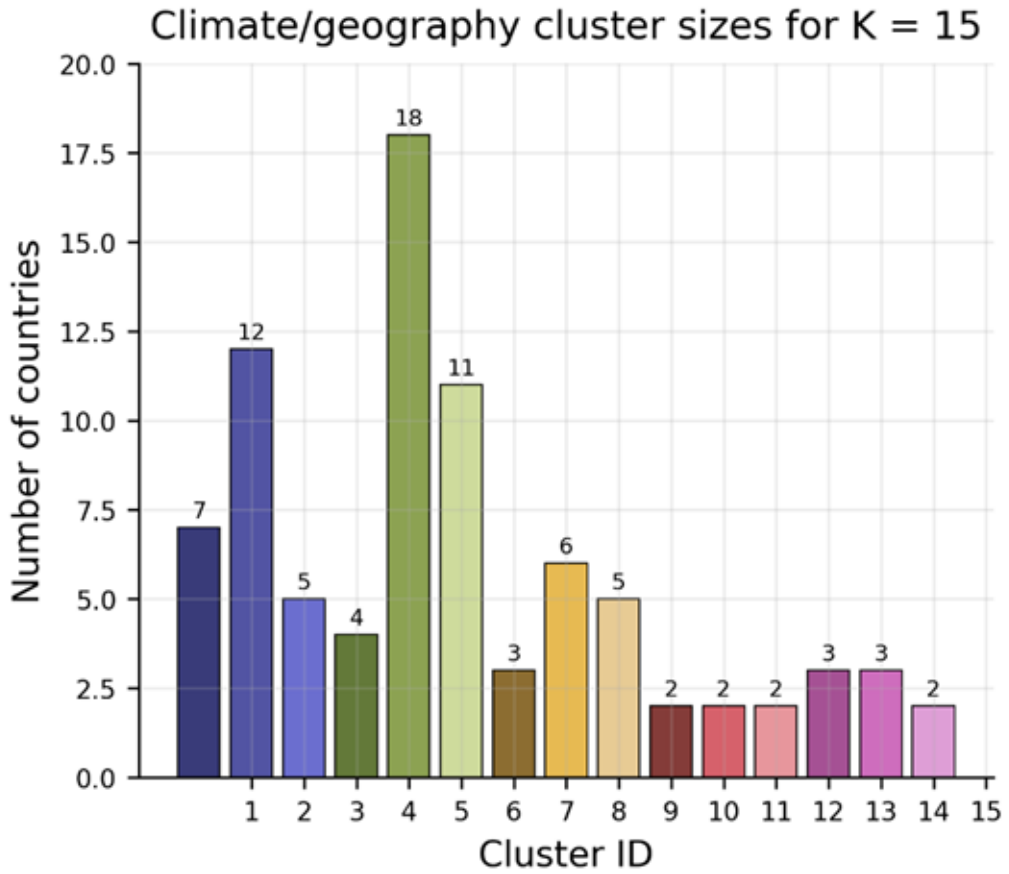
Each point represents one country. The color indicates the assigned temperature-only cluster. Countries with similar seasonal temperature patterns are grouped together, even if they are geographically far apart

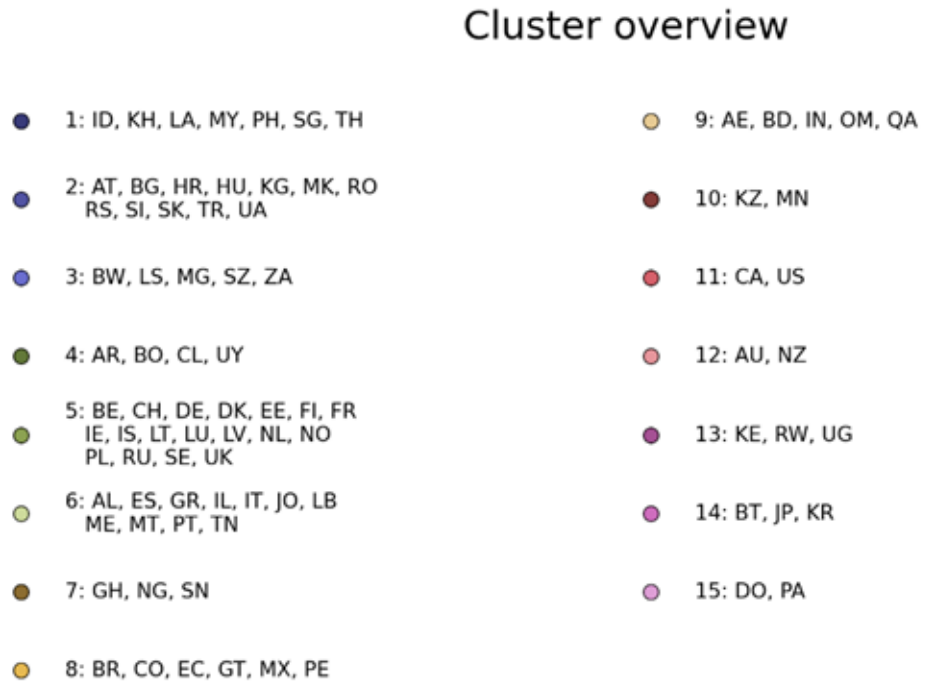


Climate/geography cluster size overview

Countries clustered using latitude, longitude, Jan.- and July temp. The plot shows the size of each cluster, while the overview on the right shows which countries are assigned for each group.

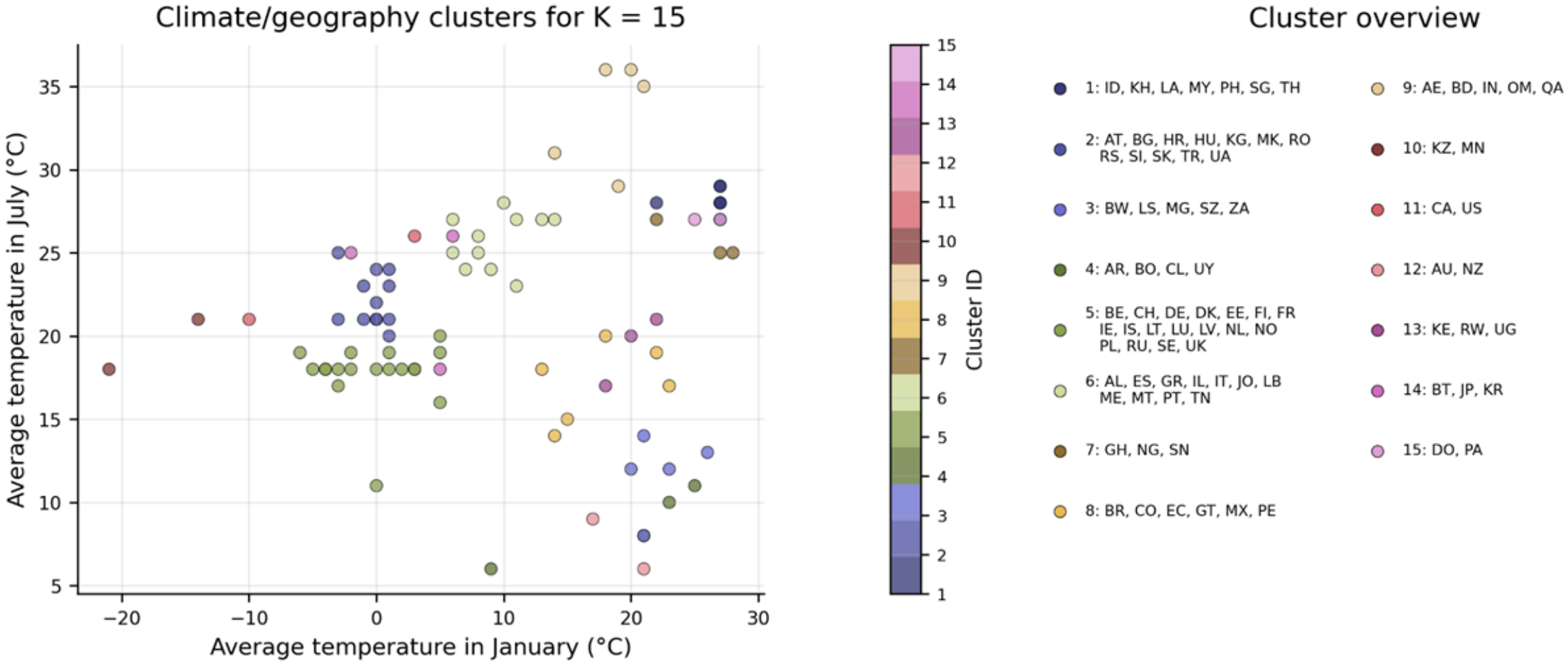
Adding geographic information changes the grouping compared to temp.-only clustering. The cluster sizes are still uneven.





Climate/geography only clusters in feature space

Each point is still plotted by Jan and July temperature, but the clusters were created using both temp. and geography. This means countries with similar temperatures can now belong to different clusters if they differ in latitude or longitude.



Uneven cluster size overview

The cluster size plots show how many countries are assigned to each cluster.

Uneven cluster sizes are important because large clusters may constrain more countries and more visual variation, while small clusters may be more specific but less representative.

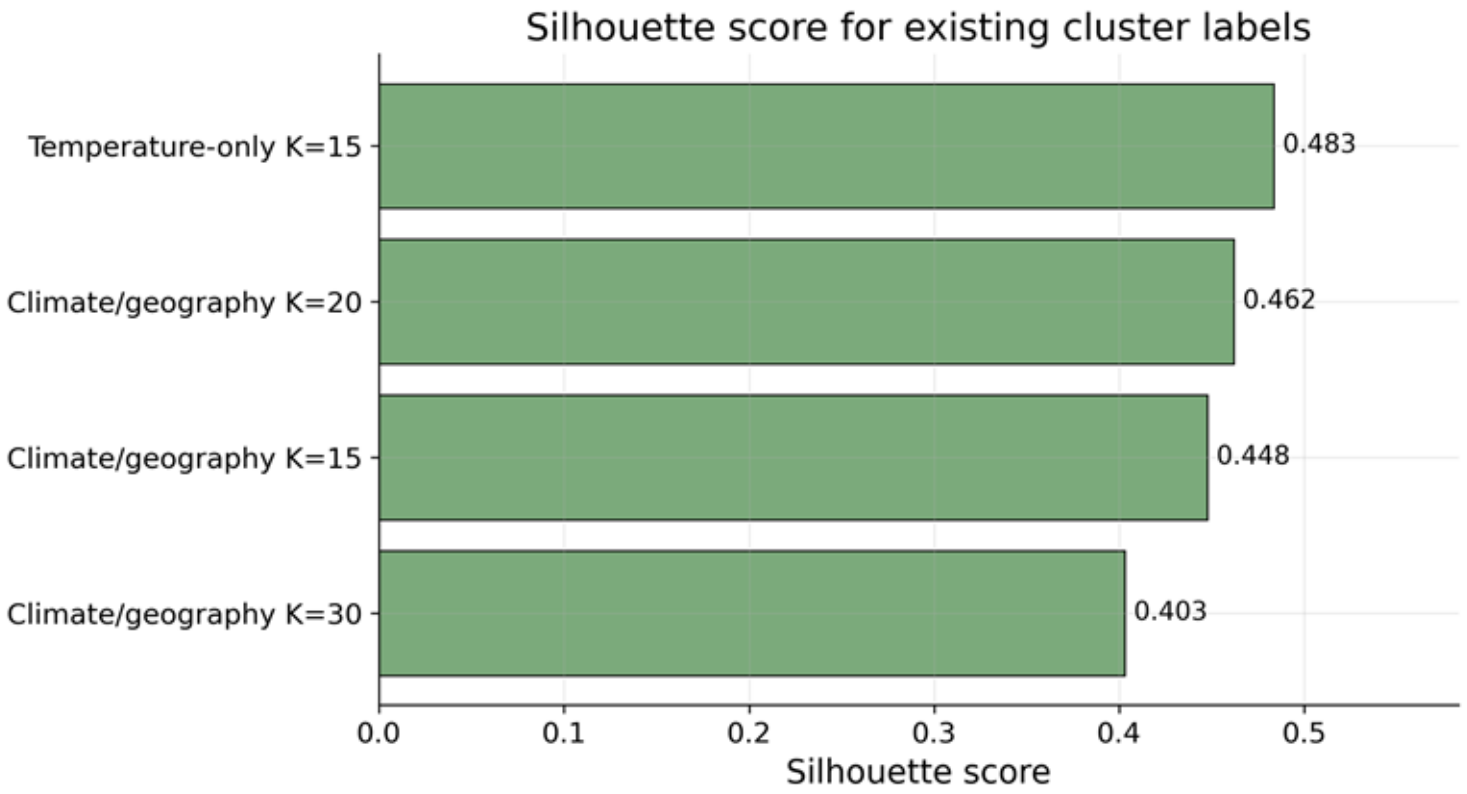
This also means that larger clusters may dominate the training loss -> smaller clusters can be harder to learn and may have lower recall/F1-score. This is why we also compare F1 score and, not only accuracy.

Comparing clustering

Silhouette score for cluster label systems

Silhouette score is used to measure how well separated the clusters are. It compares how close each country is to

A higher score indicates more compact and better separated clusters. **The temperature-only clustering obtains the highest silhouette score,** suggesting that countries are more clearly separated when only January and July temperatures are used. However, the climate/geography clusters may still be more meaningful for image classification, because they also include spatial information through latitude and longitide



Comparing clustering

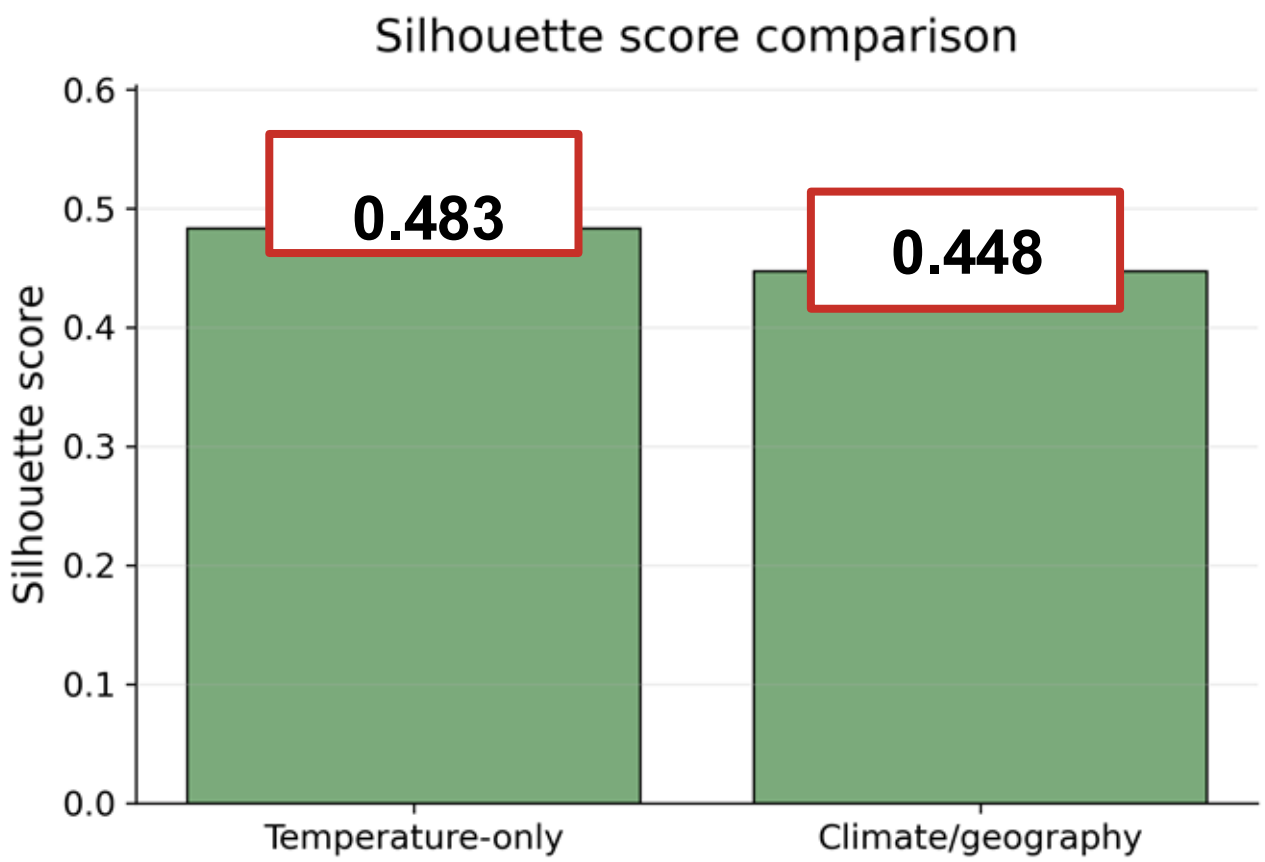
Silhouette score comparison

Higher score = more compact

and clearly separated clusters

Temp only > Climate/geography

Meaning that the temperature-only is more separated, which can might be because there's less features to assign, and it does not necessarily make it better.

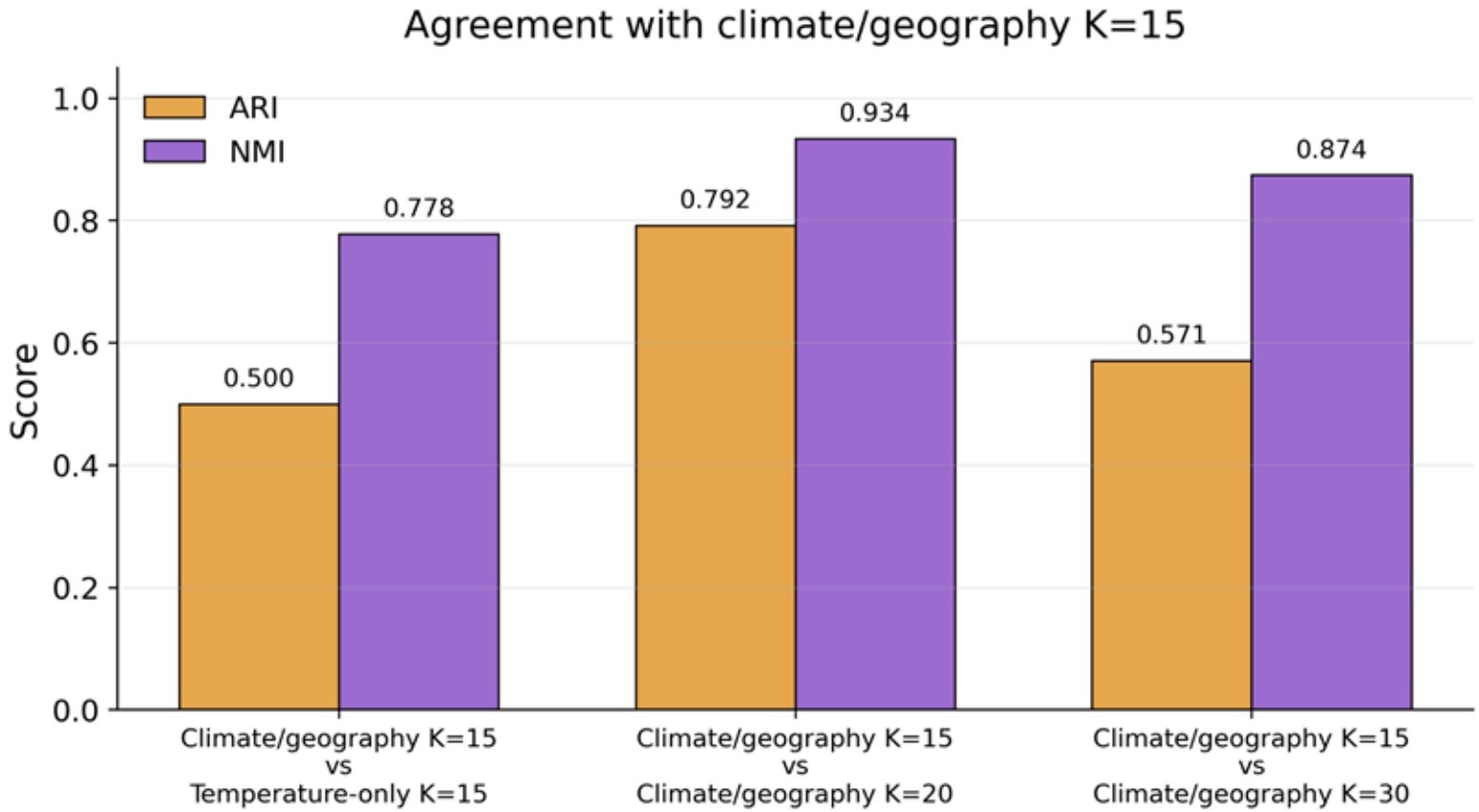


Cluster - Agreement with climate/geography K=15

ARI and NMI

ARI and NMI compare how similar two cluster label systems are. Here, the climate/geography clustering with K=16 is used as the reference and compared with the other clusterings.

ARI measures exact agreement in how countries are assigned to clusters, adjusted for random agreement. NMI measures how much information the two clusterings share.

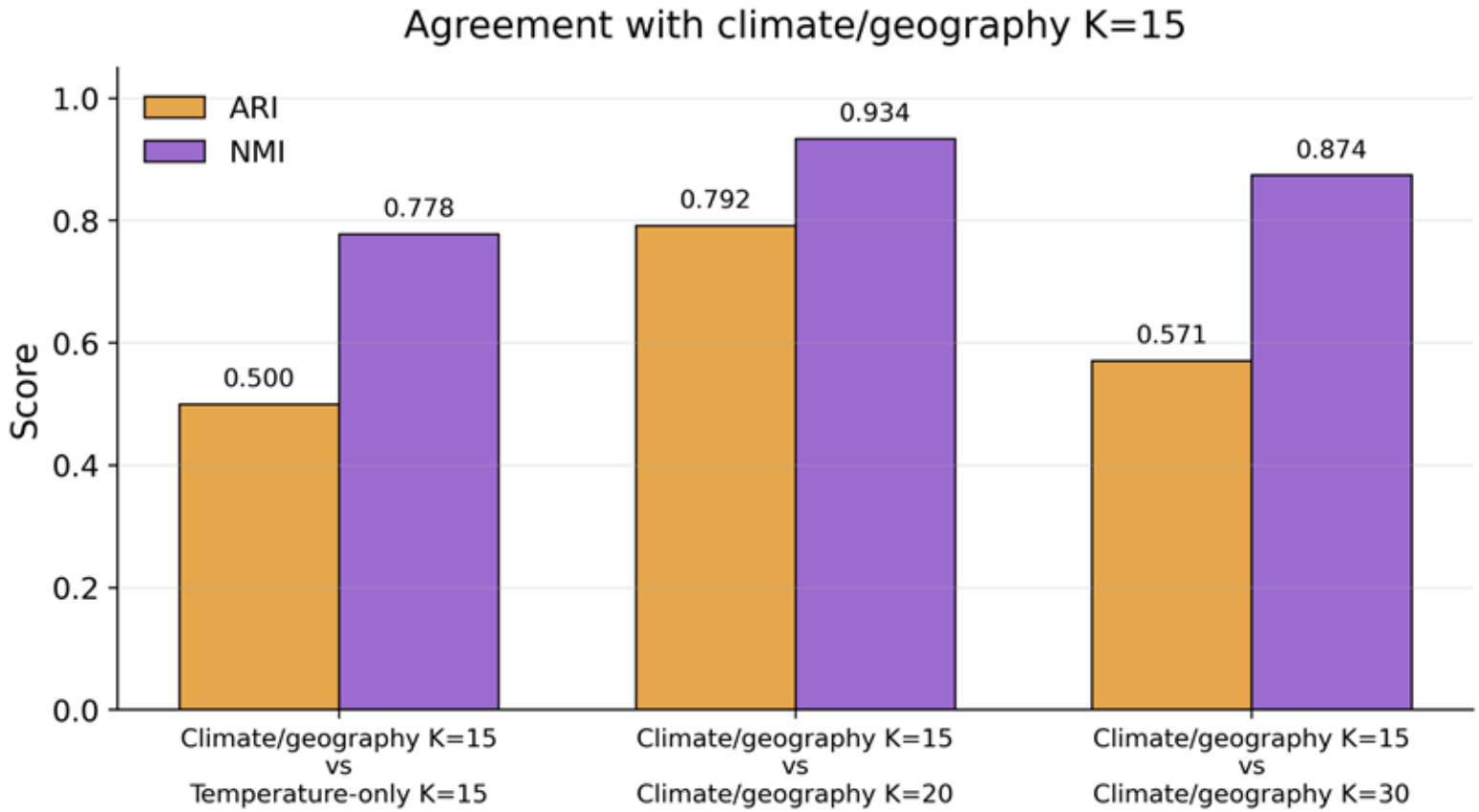


Cluster - Agreement with climate/geography K=15

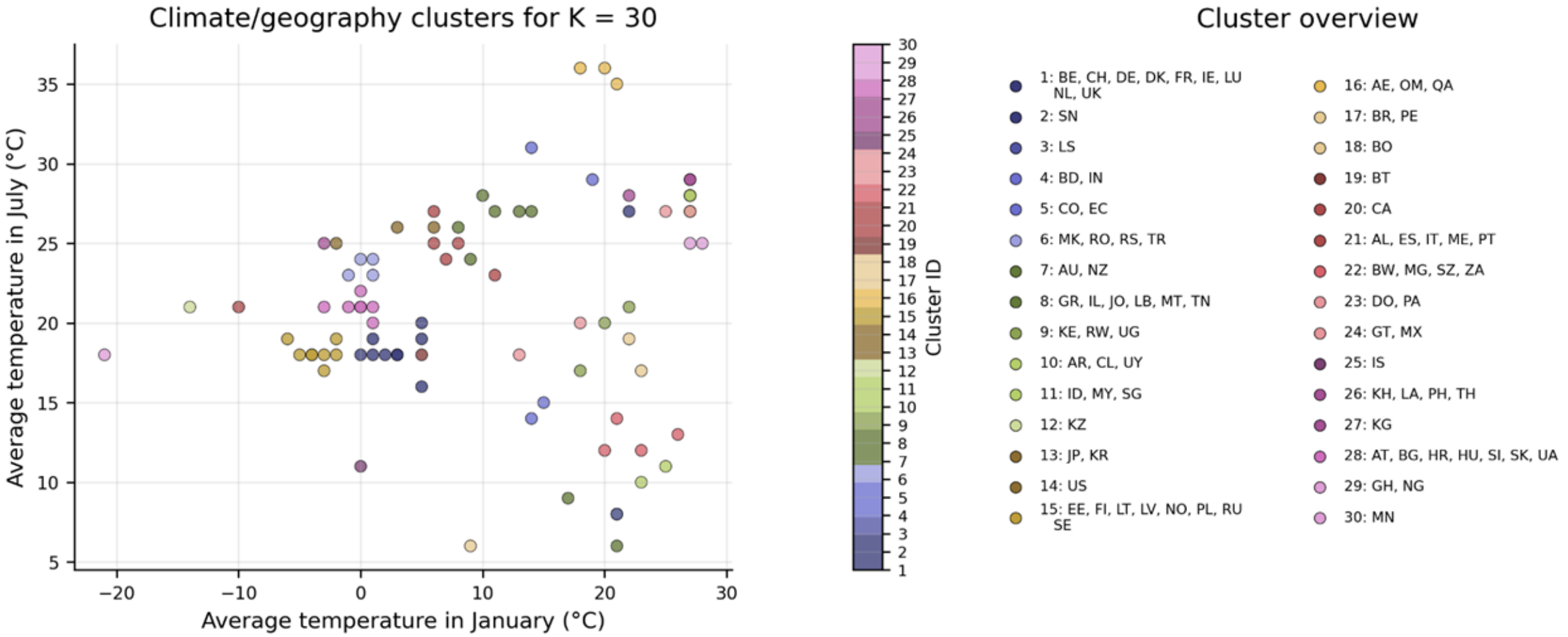
ARI and NMI

The climate/geography K=20 clustering has the strongest agreement with K=15 which is expected because both are based on the same climate/geography features. The temperature only clustering has lower

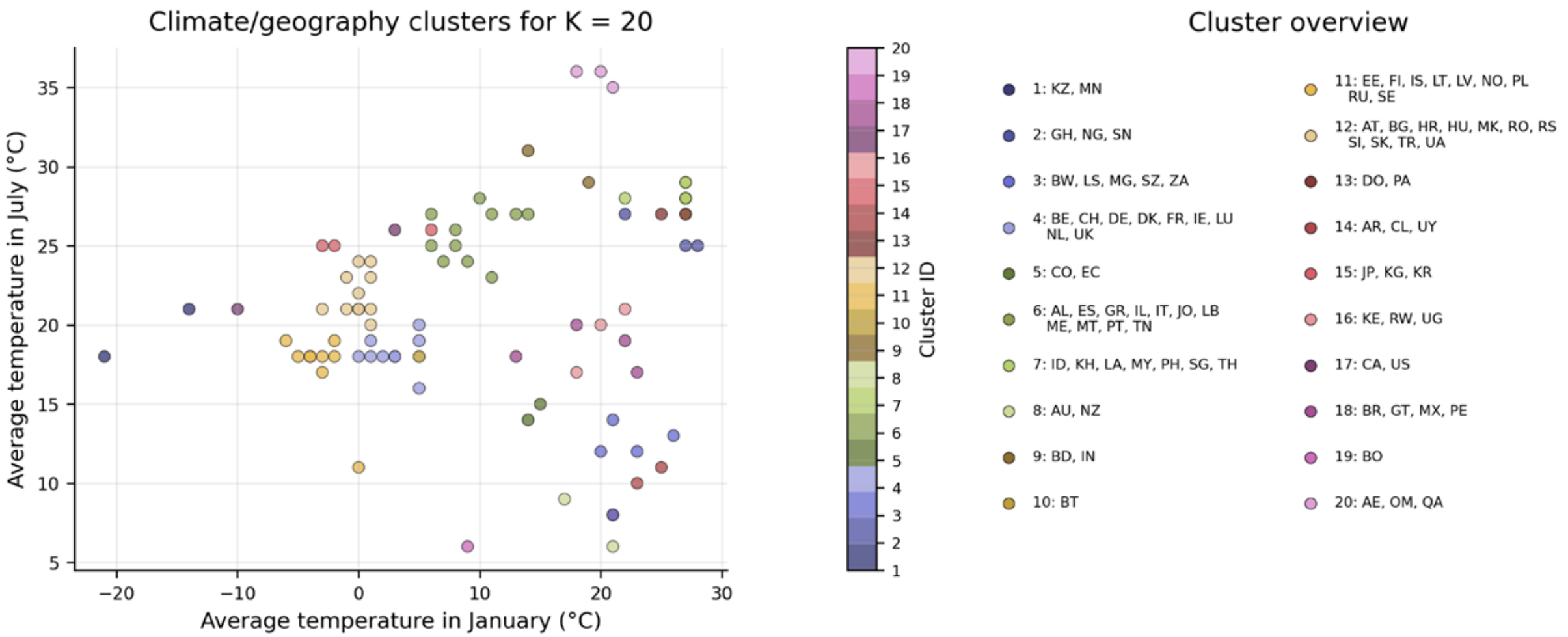
ARI but still relatively high NMI, meaning that temperature explains part of the structure, but does not give exactly the same country groupings.



Cluster for K=30



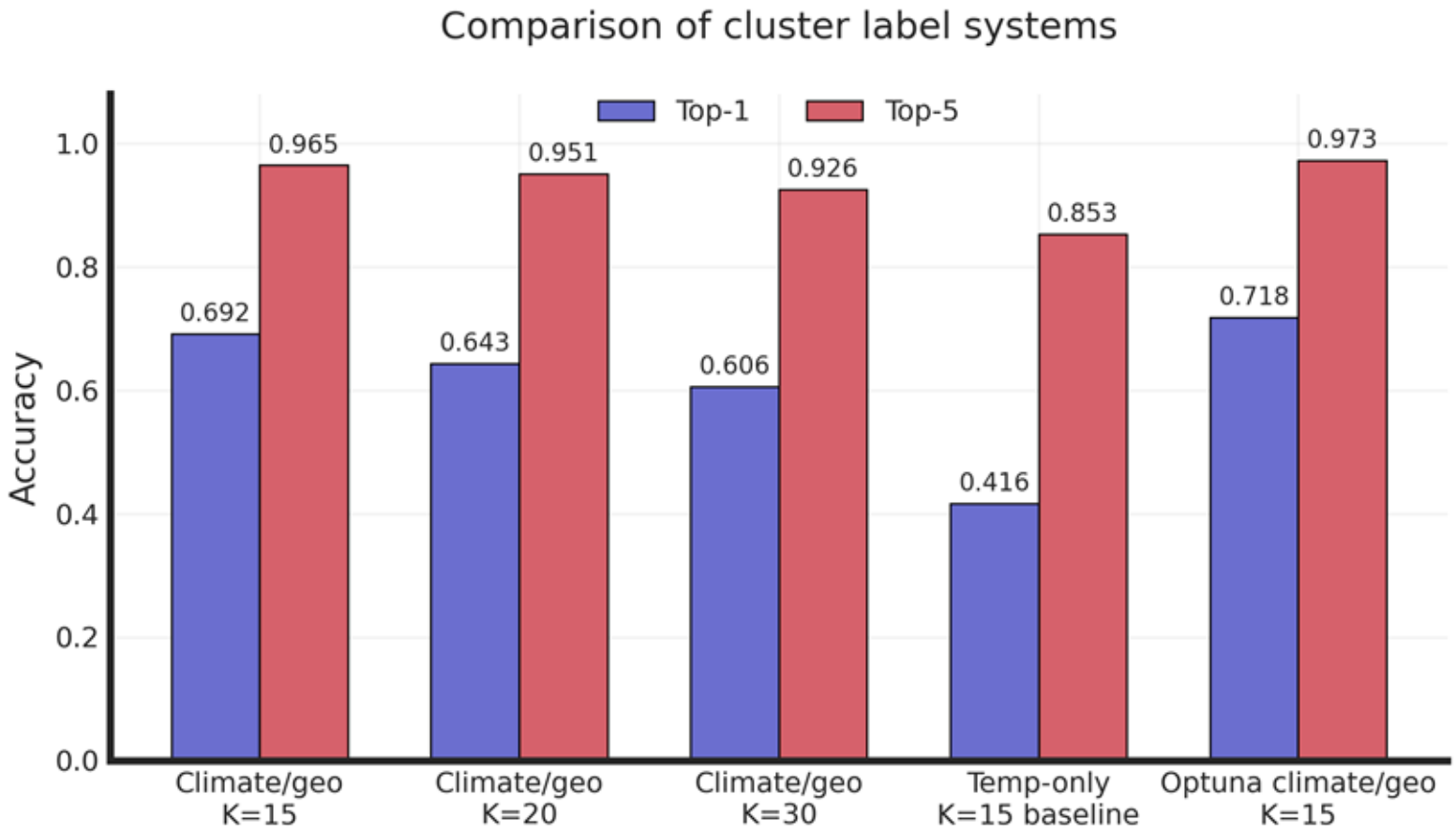
Cluster for K=20



Overvies of cluster models

Different bbales systems used for ResNet-34 training

All the models use ResNet-34 and take images as input.
The difference is how the output labels are defined
K: number of clusters/classes
Two different cluster methods
Optuna: optimized training hyperparameters

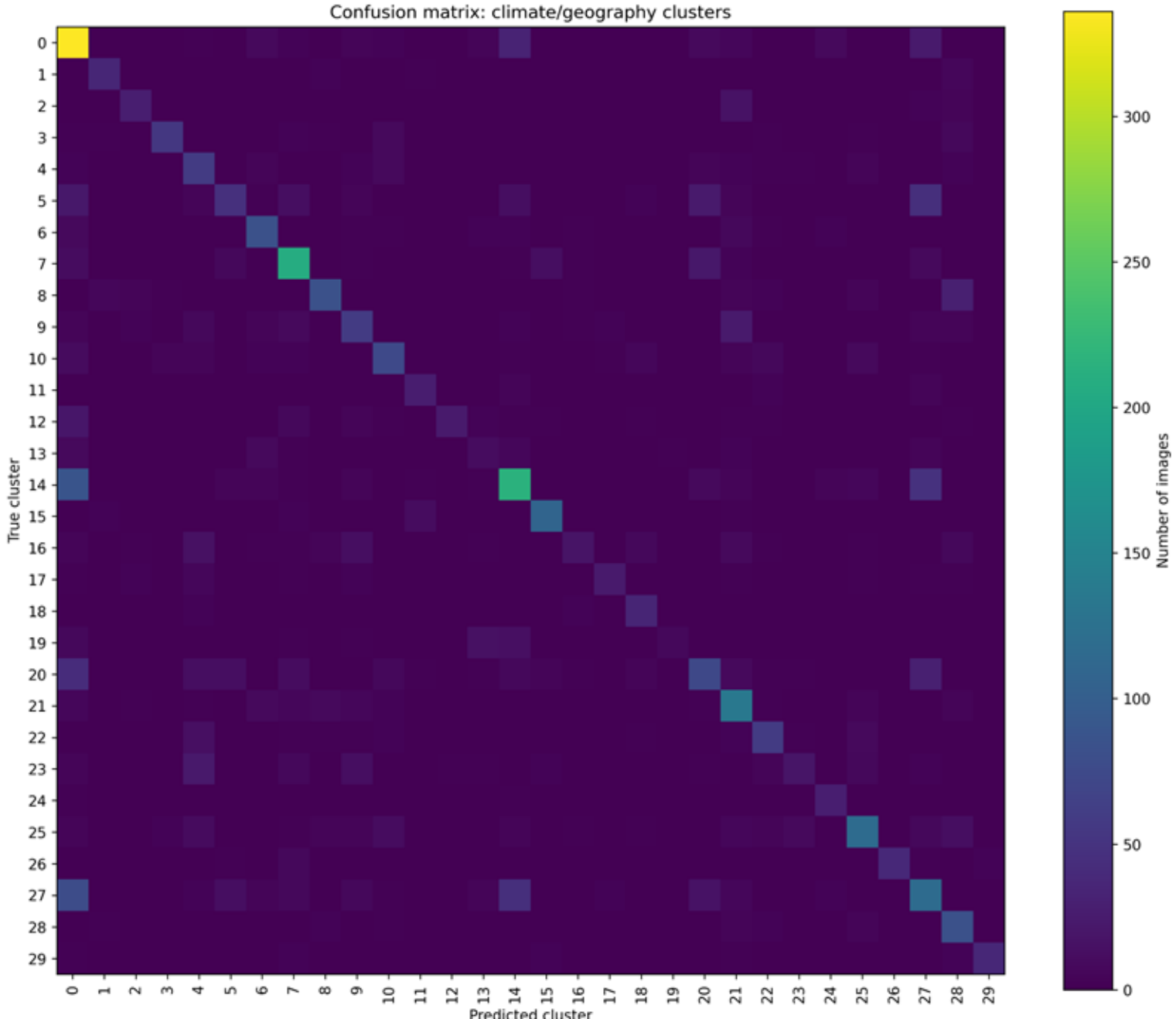


Confusion matrix for climate/geography

For the K=30 model

Most predictions lie on the diagonal, showing that the model generally predicts the correct cluster.
The off-diagonal cells indicate remaining confusion between some clusters.

Takeaway: The cluster model works overall, but not all clusters are equally separable

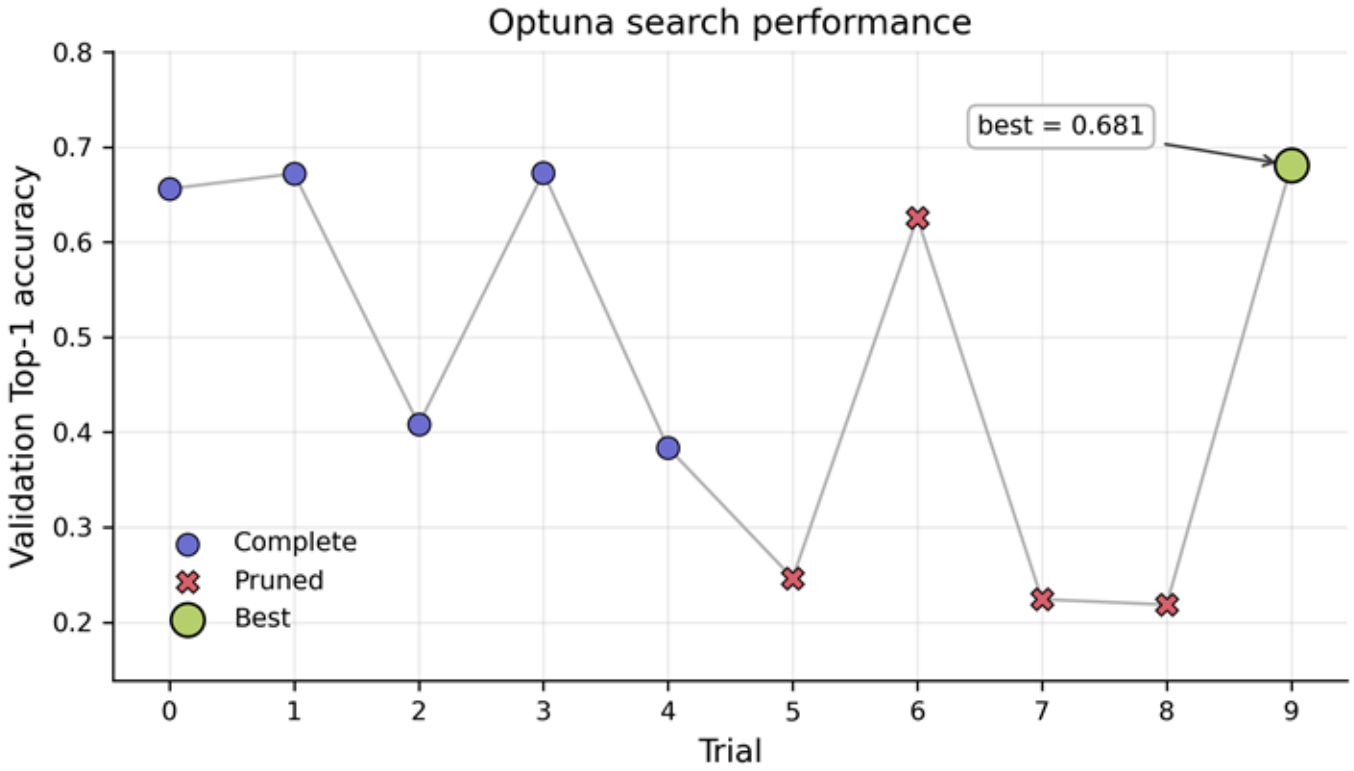


Hyperparameter optimization with Optuna

Optuna was used to tune the ResNet34 climate/geography (K=15) model. The objective was to maximize validation Top-1 accuracy. The search tested combination of learning rate, dropout, optimizer, weight decay and whether to freeze the backbone. The best complete trial reached validation **Top-1 accuracy (=0.681)**. This optimized model was used for the final cluster based evaluation.

Best Optuna hyperparameters
ResNet34 climate/geography K=15

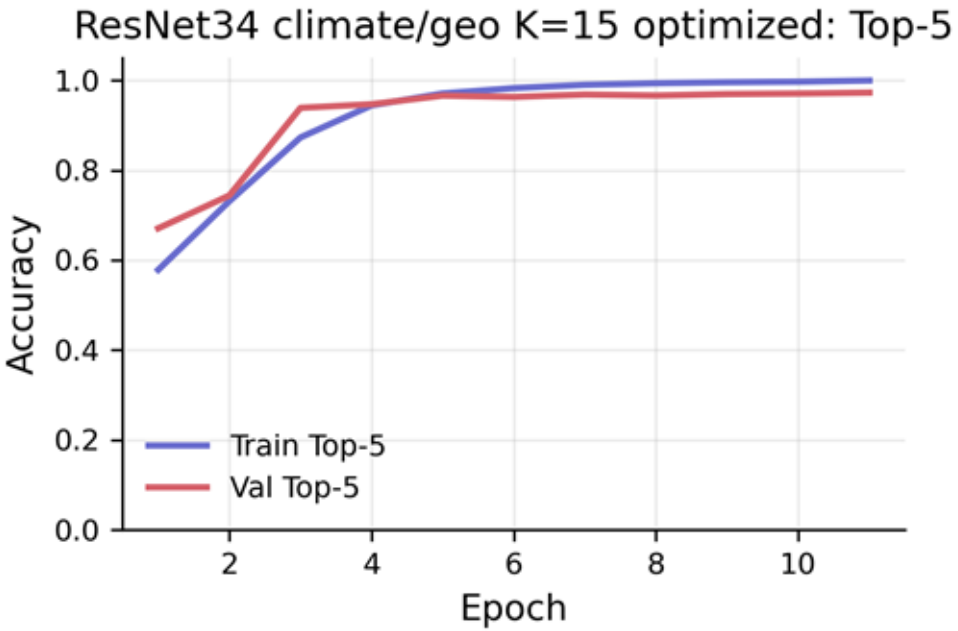
Quantity	Value
Best trial type	**Best complete trial**
Trial number	**9**
Validation Top-1	**0.6810**
State	**COMPLETE**
batch size	**32**
dropout	**0.4951**
freeze backbone	**False**
lr	**3.39e-05**
optimizer	**AdamW**
weight decay	**1.31e-04**

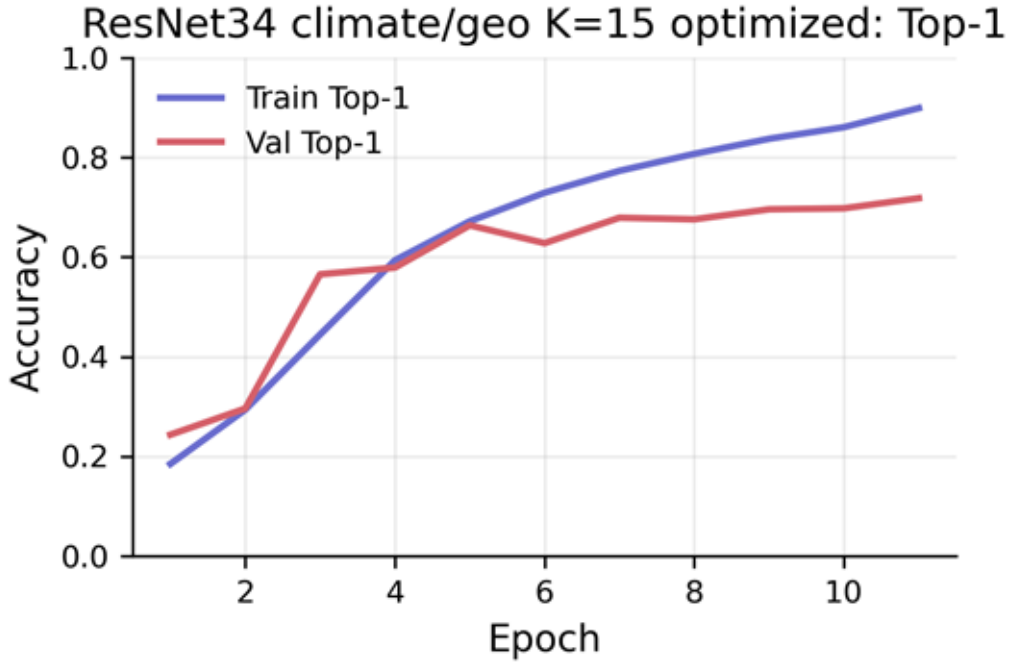


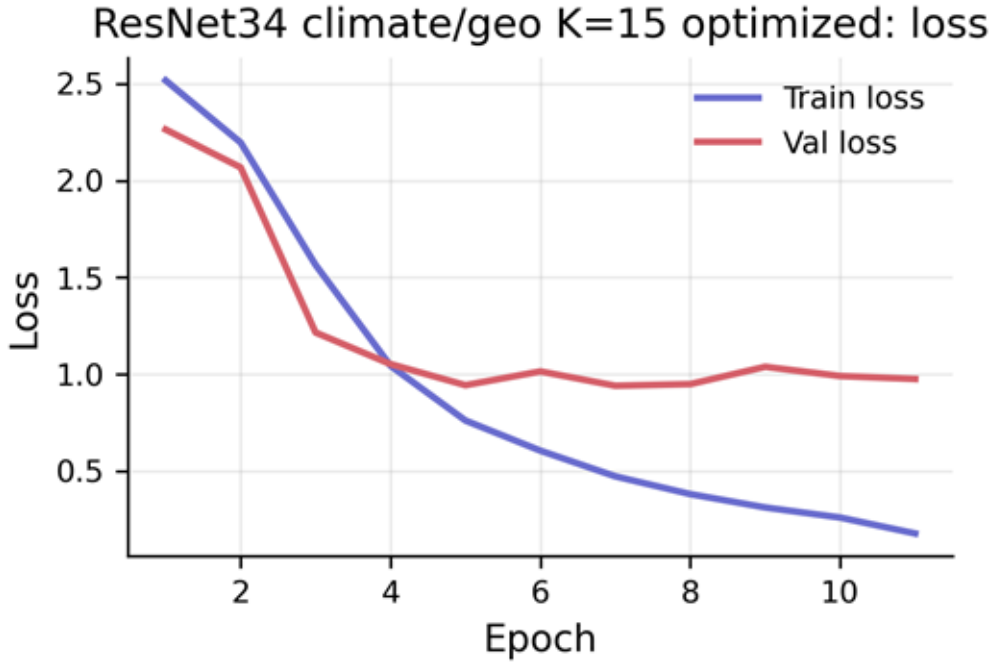
Training curves for optimized cluster model

The optimized ResNet34 climate/geography (K=15) model improves during training. Top-1 validation accuracy increases and then kind of stabilizes, while Top-5 accuracy becomes very high.

The training loss continues to decrease, while validation loss levels off, indicating some overfitting but stable validation performance.

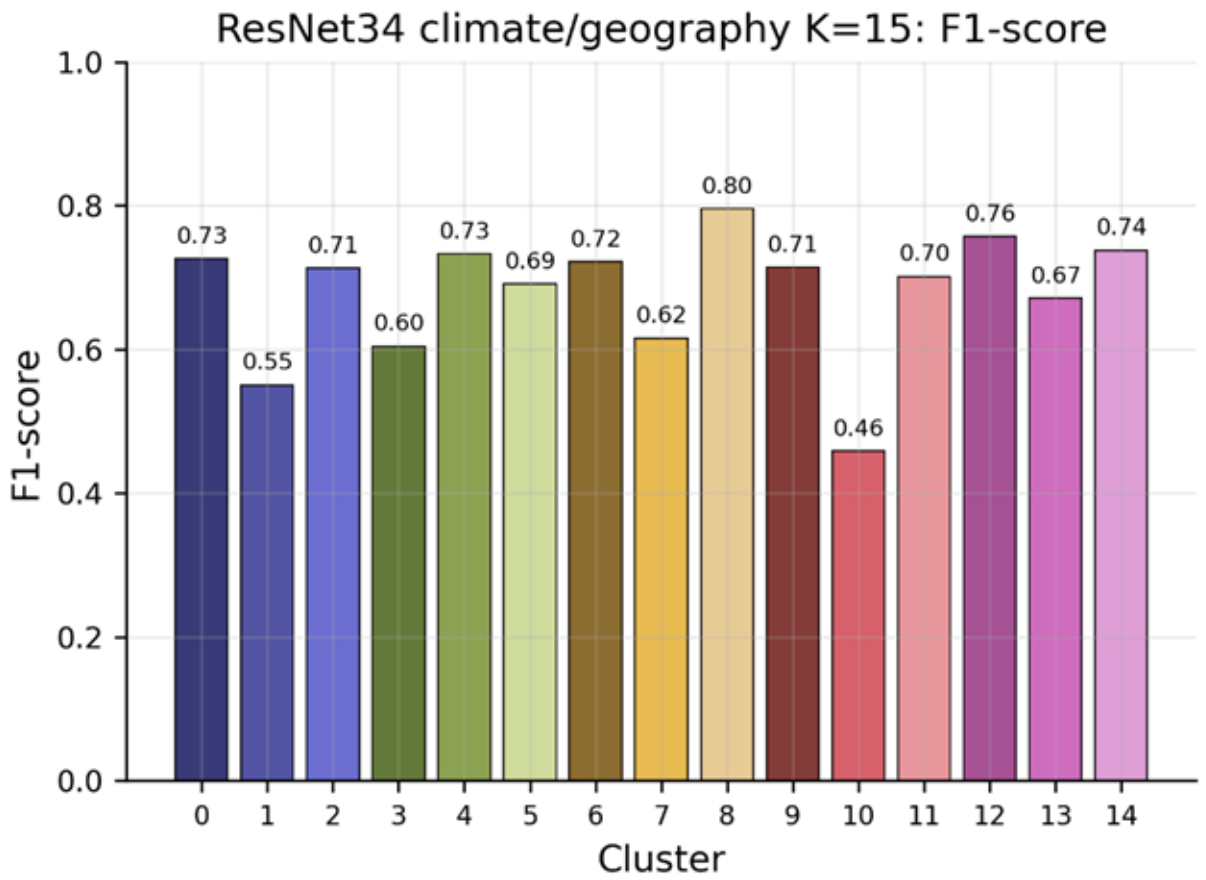


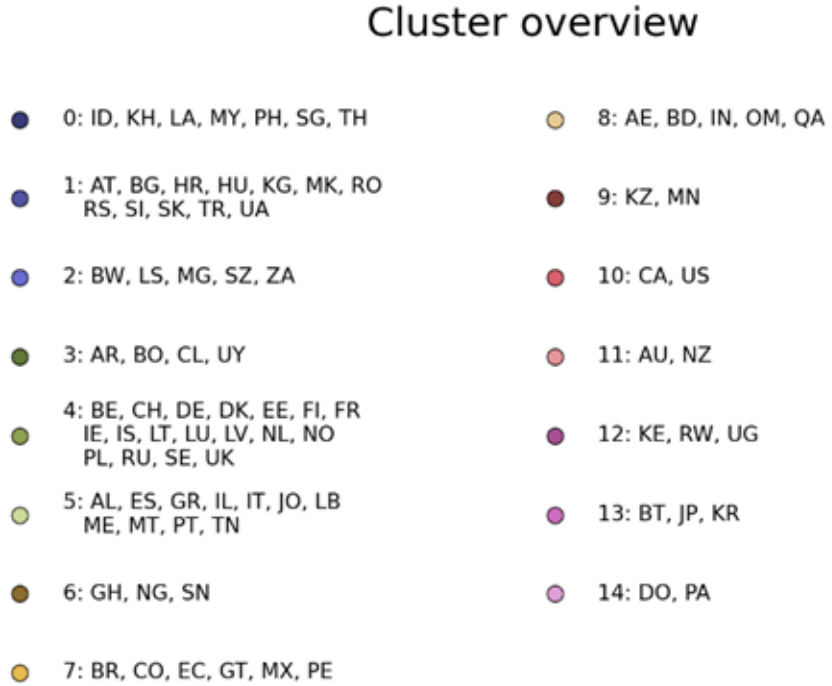




F1-score per cluster

The optimized climate/geogrpagy K=15 model does not perform equally well across all clusters. Most clusters have relatively high F1-sores, but a few clusters are noticeably lower. This suggest that some clusters are more visually coherent, while others constrain countries that are harder to separate. Ex. USA/Canda cluster may score lower because it's visually very diverse.

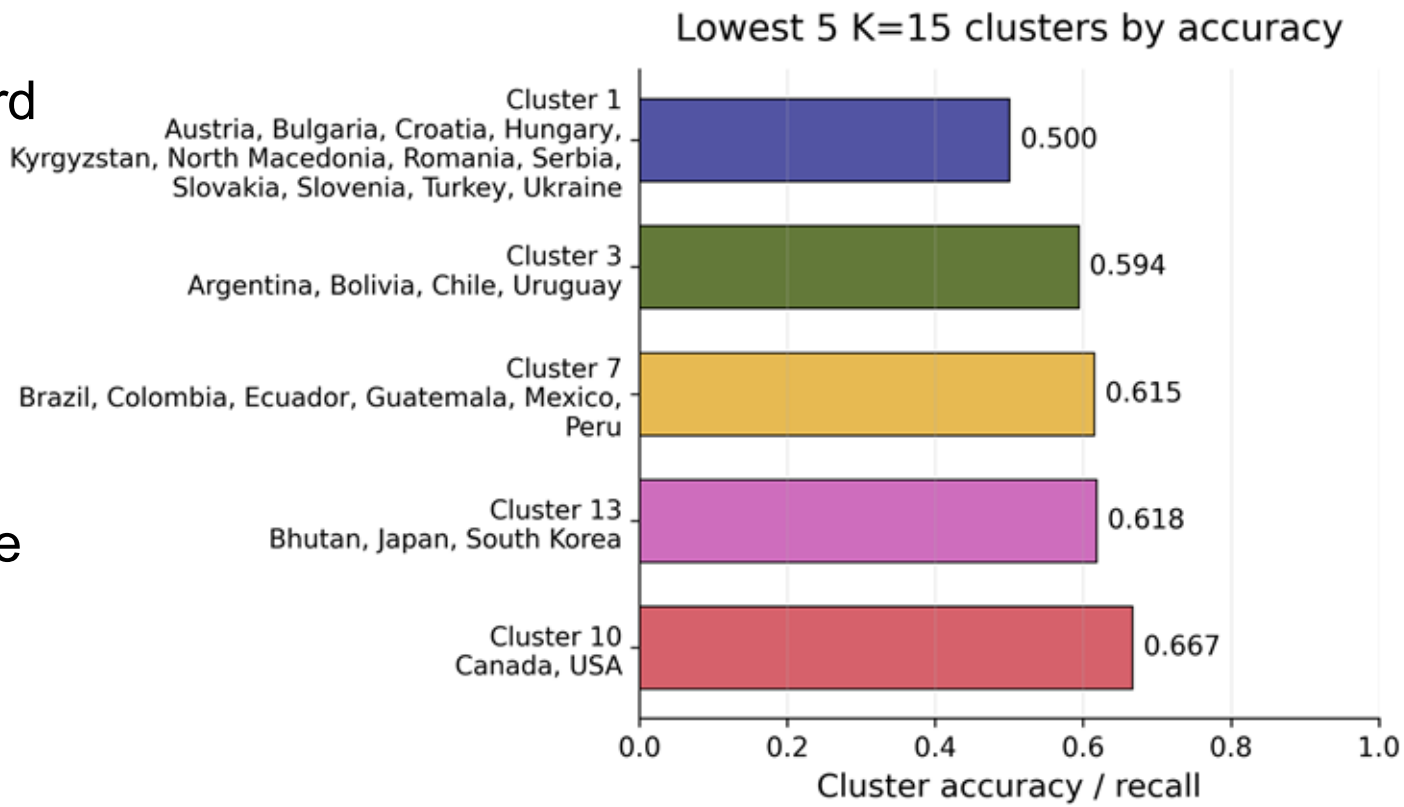




Cluster

Lowest 5 clusters by accuracy

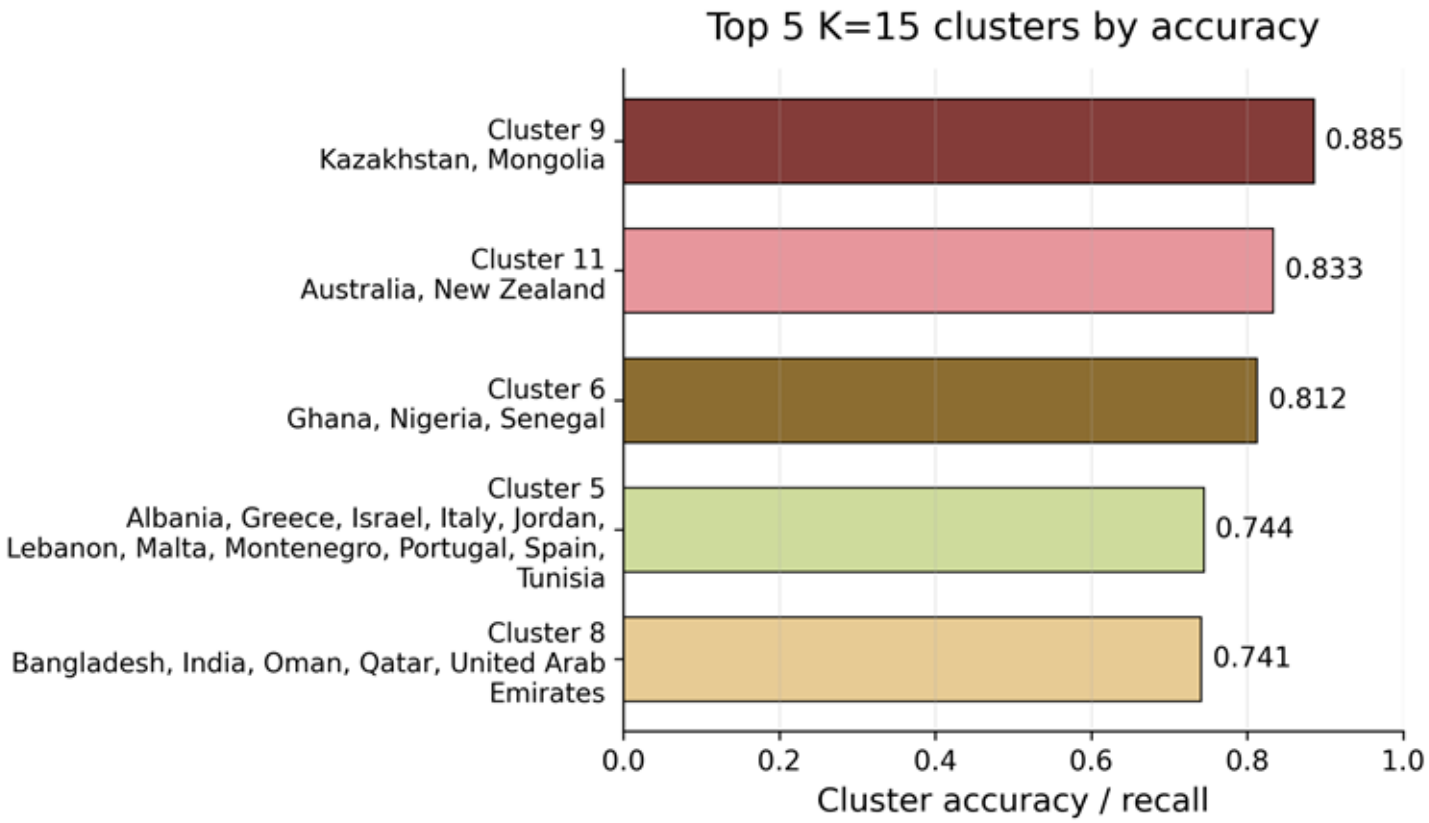
- Several low performing clusters contain countries that were also hard individually
- E.g. Croatia, Slovakia, Brazil, and Mexico
- Larger or visually diverse lusters are harder for the CNN to learn
- Some country-level difficulties disappear when countries are grouped into broader regions



Cluster

Best 5 performing clusters

- Several high performing clusters contain countries that were also easier in the country level model
- Eg. Mongolia, Ghana, Senegal, Bangladesh, India and Qatar
- Broader clusters labels can be easier than exact country prediction



Cluster - Conclusion

Overall, the clustering approach reduces the complexity of the classification task by grouping countries according to climate and geographical similarity. The temperature-only clusters are slightly more compact in metadata sace according to the silhouette score, while the climate/geography clusters provide more regional structure.
The final comparison is therefore based on mainly on CNN performance, especially validation accuracy, Top-1 accuracy, F1-score and confusion matrices. Since the cluster based model performs better than the original country level model, this suggest that climate/geography based labels provide a useful structure for image classification.

However, the clustering is still based on simplified country level averages. For example, large countries such as the USA can contain very different landscapes, climates and visual environments, which may not be well represented by a single average temperature or coordinate value

Cluster - Future work

A possible improvement would be to include more detailed country metadata, such as precipitation, vegetation, elevation and regional climate variation. This could make the clusters more visually meaningful and potentially improve classification performance.

Dataset pollution: Car artefacts Rwanda








Patch importance

Which patches are most important

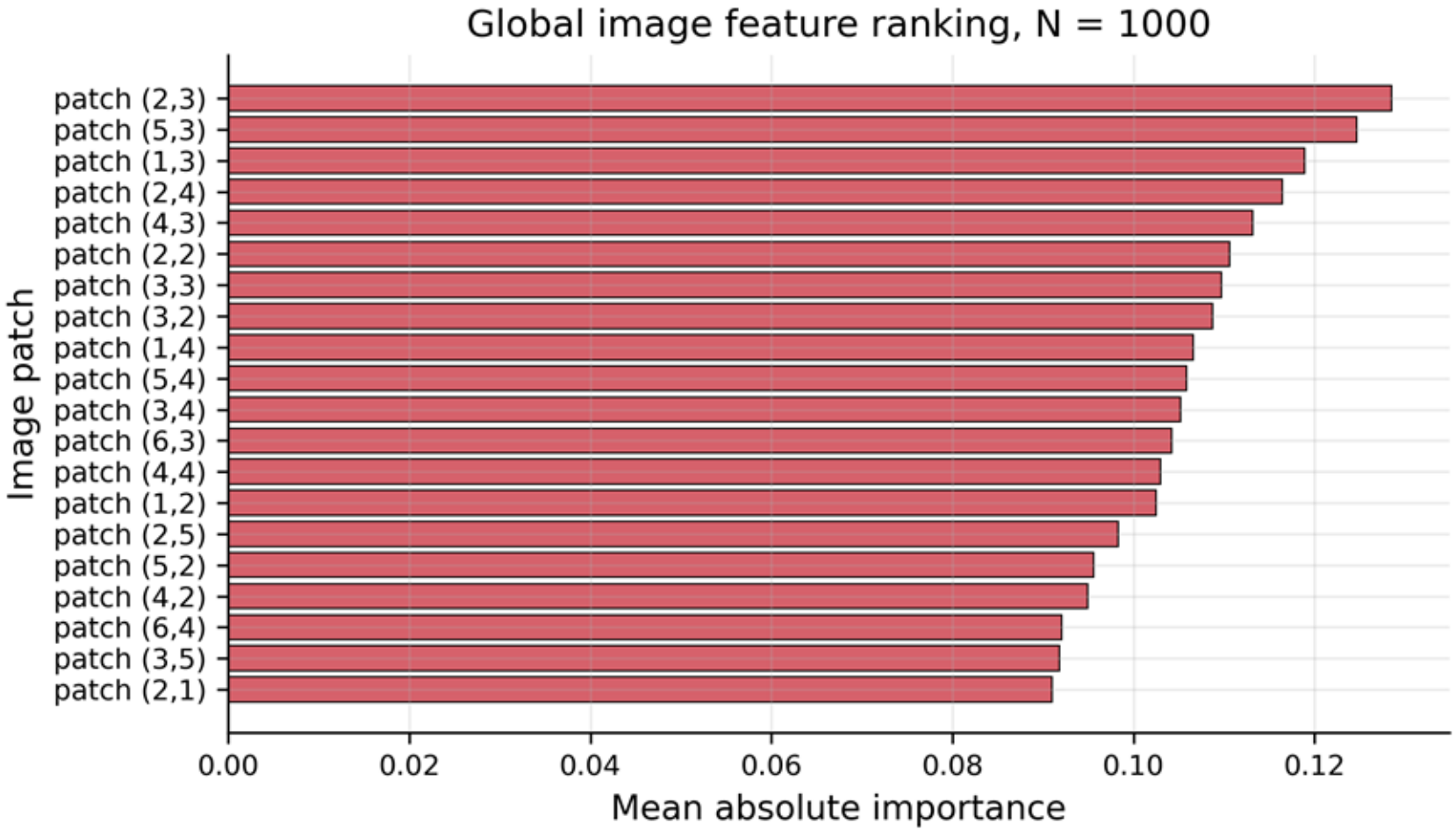
- Each validation image was divided into smaller patches
- One patch was removed/occluded at a time
- The model prediction was computed again
- Patch importance was defined as the change in confidence for the predicted class
- A large drop in confidence means that the patch contained information the model relied on

Important: Patch importance does not prove that the model “understands” the object or scene. It only shows which image regions affect the model’s confidence.

Patch importance

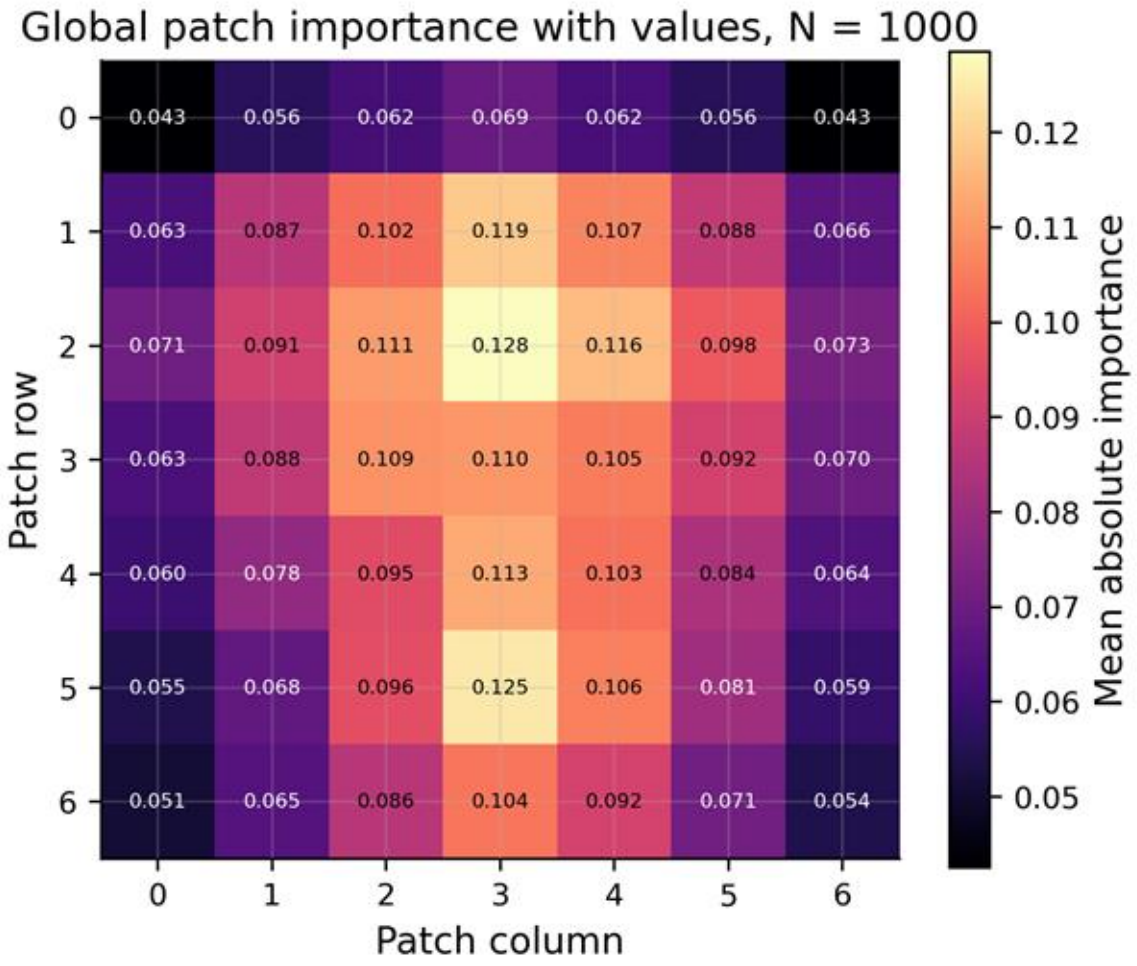
Which patches are most important

- Computed across (N=1000) validation images
- Each image was divided into patches
- One acth was removed at a time
- Importance = how much the rendition confidence changed
- The most important patches are mainly in the central part of the image



Patch importance

- We see the most important patches are mainly located near the center of the image
- This may reflect useful visual information:
  - Road type
  - Lane markings
  - Vegetation
  - Building signs
- But it may also reflect dataset/image composition bias
- GeoGuessr images often place the roand and horizon near the center

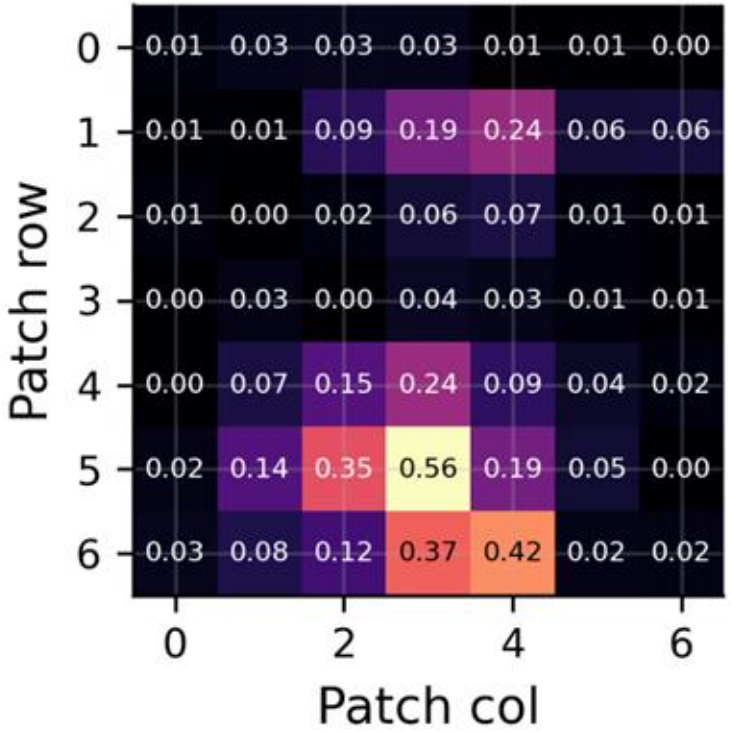


Example of patch analysis

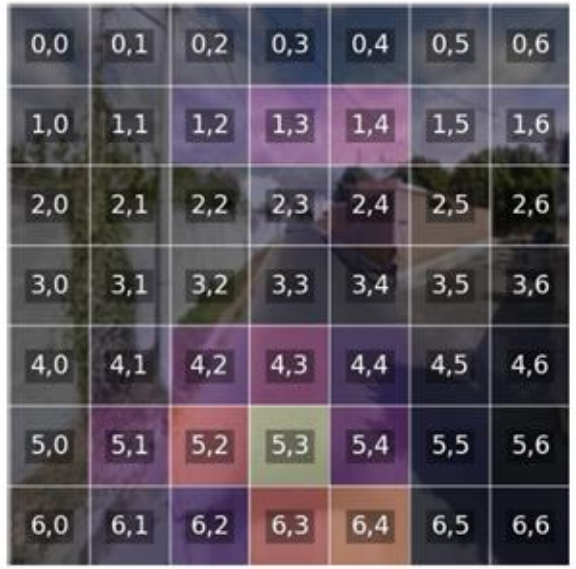
Mexico | true cluster 8 | pred cluster 8

Original

Patch importance

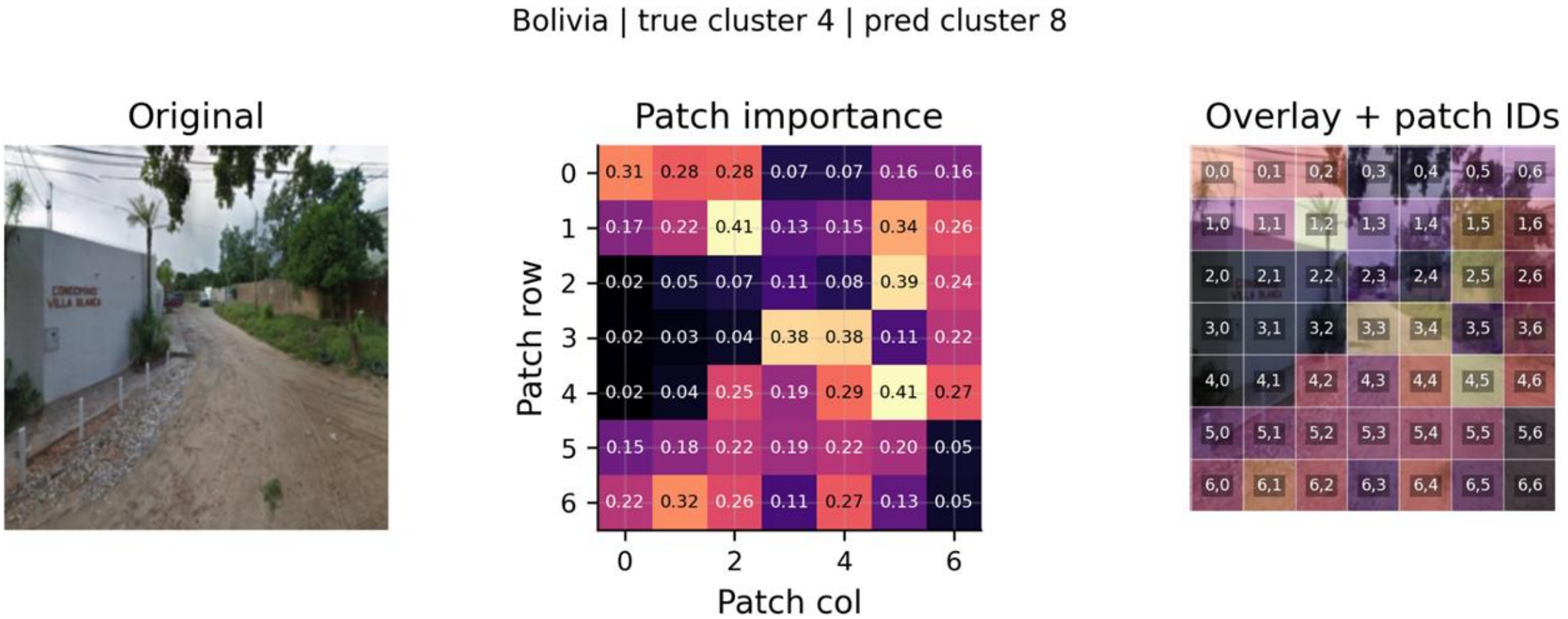


Overlay + patch IDs



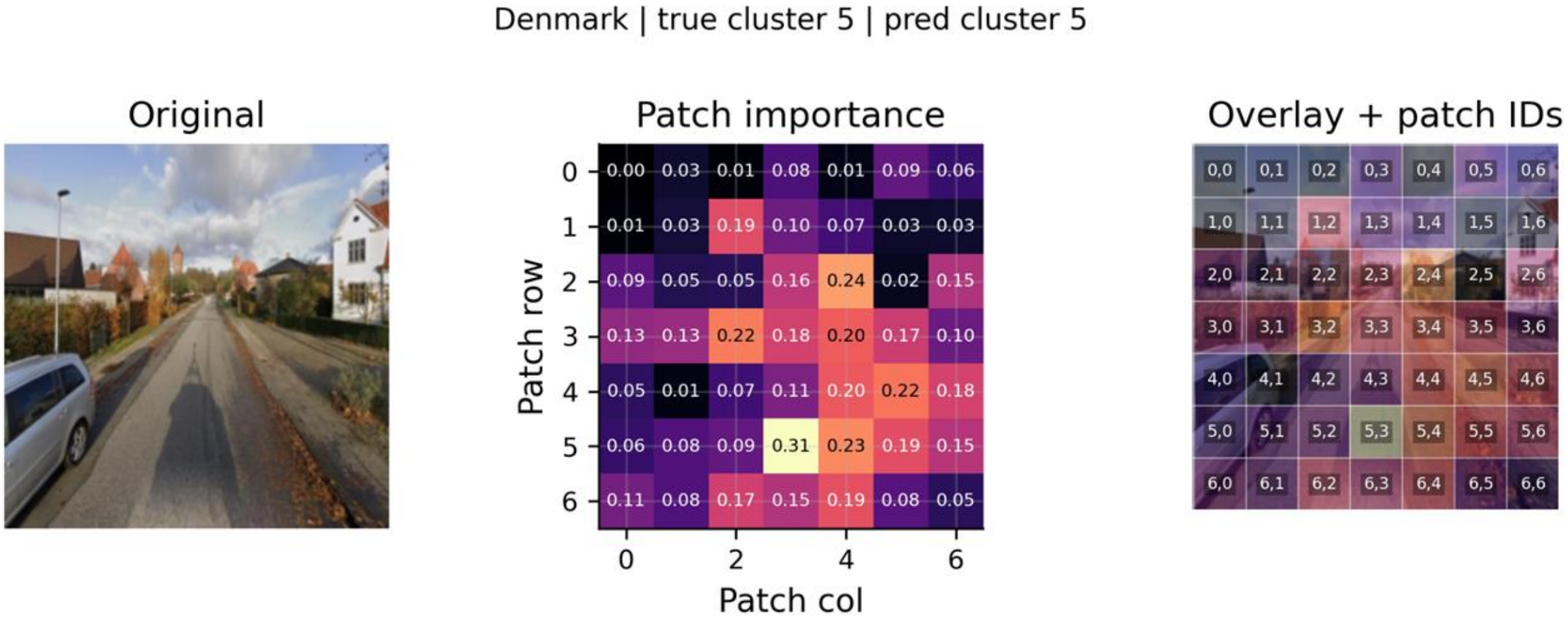
For correct predictions, important patches may correspond to meaningful location cues..

Example of patch analysis



For incorrect preditcions, the model may rely on misleading or generaric visual features

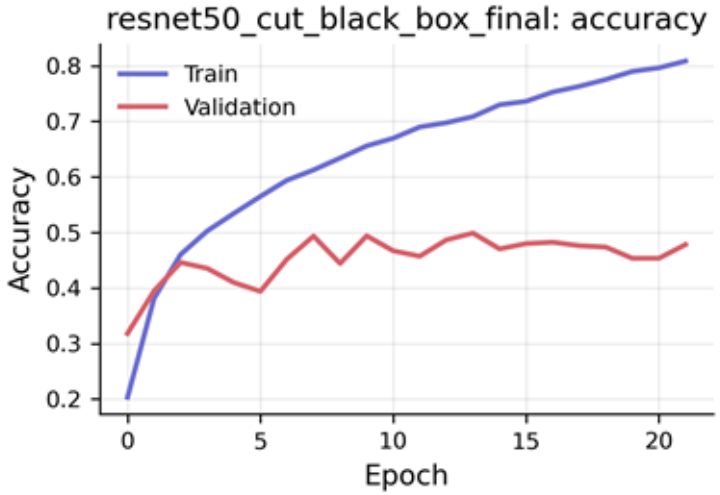
Example of patch analysis



Patches future work

It might be interesting to compare patch maps for correct and incorrect predictions. If an incorrect prediction relies on sky or generic road texture, it may indicate that model is using weak or misleading cues

Additional experiment with black box



ResNet50 Test Accuracy	
Top 1	Top 5
0.502	0.813

The HP's was retuned from the last drop and crop results