ICE or Regional Bahn? CS 4641 Final Report

Jonathan Zhou[1] and Matthew Zhou[1]

[1]**College of Computing, Georgia Institute of Technology**

ABSTRACT

We provide a classifier which, given an image from either class, disambiguates between Deutsche Bahn InterCity Express and Regional Bahn trainsets.

Keywords: Image Classification, Transfer Learning, Deutsche Bahn

1 INTRODUCTION

German rail service is primarily served by the state-owned private company Deutsche Bahn (DB). DB operates numerous passenger rail services which can generally be two tiers of rail service: (1) regional rail service (operated by DB Regio) and long distance service (operated by DB Fernverkehr). Regional rail service includes S-Bahn (hybrid urban-suburban rail systems), Regional Trains, and Regional Express services(1).

Different kinds of rolling stock are operated by these two branches. In the flagship service of DB Fernverkehr, the InterCity Express (ICE), special aerodynamic electric-multiple-units from Siemens Mobility are used, and are painted in a white livery with a red stripe. On the other hand, regional train-sets, typically manufactured by Bombardier or Siemens Mobility, can vary by region and by service, can be both locomotive pulled or electric multiple units. Although they are more diverse in appearance, regional trains are most often painted in a red livery, and we limit our analysis to this category for the sake of simplicity.

We propose a model which, given an image of an ICE or Regional trainset, gives a classification between the two. Our model is based on top of Deep Residual Learning for Image Recognition (ResNET) (3). We then perform transfer learning starting from pre-trained ImageNet weights and using with our ICE and Regional Bahn image samples(5)(2). We also use image augmentation to increase the robustness of our training process (6). We implement our model using the PyTorch software library (4).

Figure 1. Representative images for (Left) Regional Bahn trainset (Right) InterCity Express trainset

Dataset	n	# Batches
Training	1151	72
Validation	143	9
Testing	143	9

Table 1. Dataset Overview

2 DATA

In total we sourced ($n = 1437$) images of trainsets, consisting of ($n = 815$) images of ICE trainsets and ($n = 622$) images of Regional trainsets respectively, operating across Germany and neighboring countries in `.jpg` format[1]. We divide the images into two sub-directories: `/ICE`, and `/Regio` which represent the two categories of train. The images are all of differing resolution and aspect ratio. Representative images are provided in Figure 1. We limited our samples to those depicting a singular trainset in primary focus (may be other objects/trainsets in the background, but not in the focus of the image) and where the trainset is clearly visible in the image. While we allowed all variations of ICE trainset, we limited our analysis to only Regional Trains painted in the most common red livery.

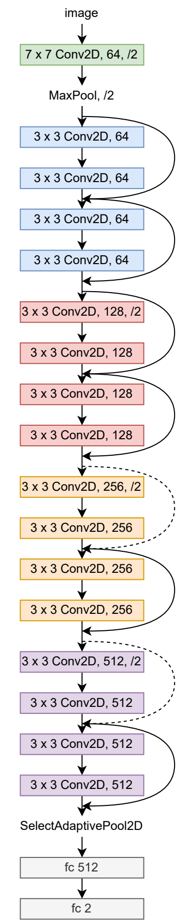

Figure 2. ResNet-18 + Transfer Learning Model Architecture [KernelSize, LayerType, OutputDim, Stride]

2.1 Data Acquisition and Preprossessing

To avoid watermarks and image rights issues, we pulled all of our images from Wikimedia Commons, where all images were licensed under a Creative Commons License. We used a shell script to pull images off of Wikimedia Commons, using different queries for the two data classes.[2]. The shell script allows us to pull images in a given *category* and pull all images related to a given *search query* matching certain tags and keywords using the Media Wiki search engine.

InterCity Express For the ICE we pulled images from select predetermined and precurated subcategories (particularly those aggregating images of certain trainset models) of the ICE main topic page on Wikimedia Commons with large amounts of exterior images of ICE trainsets.[3]

Regional Bahn We downloaded all images using the `"S-Bahn"` and `"DB Regio"` in bulk using queries to the Wikimedia search engine.

Finally, after making these queries and pulling the data, we then manually rejected all images which did not contain the trainsets as the primary subject of the image (including images of train interiors, supporting infrastructure, or signage). We also rejected trains painted in unusual livery such as non-red Regional Trains and special edition InterCity and EuroCity trains (which are technically a different service) or trains visibly involved in maintenance or disasters.

2.2 Data Preprocessing

Data Augmentation To increase robustness of our training process we introduce image augmentation into our training process. For the training set, we first resized the image to 281×281 (110% of the original size). Then, we randomly cropped the images to be 256×256. We then applied random rotations of up to $\pm 30°$ and randomly flipped the image horizontally. We finally normalized the image based on the mean and standard deviation of ImageNet samples. For the images in the testing set we correspondingly scaled images to 281×281, then performed a center crop and normalized according to ImageNet statistics (2).

Test Train Split Aiming for a 8-1-1 ratio of training to testing to validation samples, out of the 1437 images, we reserved 143 images for the test set, 143 images for the validation set, and the remaining 1151 images for our training using a random split. See Table 1 for an overview.

3 METHODS

3.1 Neural Architecture

We extend the Neural Architecture of ResNet-18, we which found to be of sufficient complexity to accomplish our task through transfer learning, while not being computationally intractable to train (3). We start using pretrained weights from ImageNet classifcation task (2). We motivate the

[1] Available at: `https://www.dropbox.com/sh/lof27s03kpt1hjr/AAA8HAef6hYb41FVQnMvffBEa?dl=0`
[2] Available at: `https://sr.ht/~nytpu/commons-downloader/`
[3] Available at: `https://commons.wikimedia.org/wiki/Category:ICE`

use of ResNet based on the high dimensionality of the data and the use of convolutional neural networks as common practice in image classification tasks.

An architectural diagram is given in Figure 2. Briefly, the ResNet network is divided into sets of Residual blocks. In line with other deep convolutional neural networks for computer vision, these blocks are primarily of composed of convolutional layers. The key insight of this model architecture is an *identity shortcut connection* wherein the network can skip one or more layers. This is primarily to address the vanishing gradient problem which affects deep neural networks in general, since unnecessary layers can be mitigated by going through identity mappings instead. Following the convolutional layers, fully connected layer(s) will transform the latent representation from the network into class probabilities (3).

3.2 Transfer Learning

Roughly speaking, *transfer learning* is the creation of new performant learners in a new domain using knowledge embedded in existing learners trained on data from different domains. Formally, given observations on $m^S \in \mathbb{N}^+$ source domains and tasks ($\{\mathscr{D}_{S_i}, \mathscr{T}_{S_i} | i = 1, \ldots, m^S\}$), and observations about $m^T \in \mathbb{N}^+$ target domains and tasks ($\{(\mathscr{D}_{T_j}, \mathscr{T}_{T_j} | j = 1, \ldots, m^T)\}$), transfer learning uses knowledge gleamed from m^S source domains to improve the performance of learner functions $f^{T_j}(j = 1, \ldots, m^T)$ on the target domains (8). Wherein m denotes the number of domains, S and T denote source and target domains respectively, and $\mathscr{D}$ and $\mathscr{T}$ domains and tasks respectively.

In our case the existing domain is general object classification as outlined by the ImageNet classification task, and the new domain encompasses specific knowledge about German ICE and local trains. Our objective is thus to synthesize existing real-world general image classification knowledge with some additional domain-specific classification insights to accomplish our task.

3.2.1 Tranfer Learning Implementation

To accomplish the transfer learning task we first freeze the parameters of the ResNet and cut off the original final linear layer off after the final Residual Block of the pretrained ResNet (after the average pool), and append our own fully connected linear model which first takes the latent representation to a hidden layer of size 512, before applying ReLU, and before transforming to a final linear output of dimension 2, with output corresponding to the class probability of ICE or Regional Bahn. We also implement a dropout of 0.2 on the latent linear layer before the final output to avoid overfitting.

3.3 Model Training

We train our model for 20 epochs using an initial learning rate of 0.0001 using the Adam optimizer and a batch size of 16. We use categorical cross entropy as our loss function. See Table 1 for an overview of the test-train split and batch information. Our final model contains $11,308,354$ parameters of which $131,842$ are trainable. In practice, each epoch took a little more three minutes of training time, a good trade-off of model complexity for computational cost, motivating our choice of epoch number. We note that training with further epochs would not have resulted in significant performance gains and there were diminishing returns in performance. We elected to pick a smaller batch size similarly based off of computational constraints, while we selected the learning rate empirically based on the rate of convergence (after running the first few epochs). Finally, at the end of training, we save the weights of our model. See Figure 3 for training and validations metrics across each epoch. We initialized our model using pretrained weights from the pytorch-image-models repository[4].

4 FINDINGS

4.1 Test Set Evaluation

We find that, after twenty training epochs, our classifier is able to effectively disambiguate between ICE and RB samples with high accuracy in all data splits. We report specific accuracy, loss, precision, recall, and F1 metrics in Table 2. This is because our data was not evenly split, as there were more ICE samples than Regional train samples. So, we needed to report precision, recall, and F1 in order to account for the imbalance in our dataset. We note that we achieve a higher test and validation set accuracy compared to the training set, indicating that our model has effectively generalized and acquired true domain knowledge about train differences, although we note that we have a lot fewer samples in the

[4]`https://github.com/rwightman/pytorch-image-models`

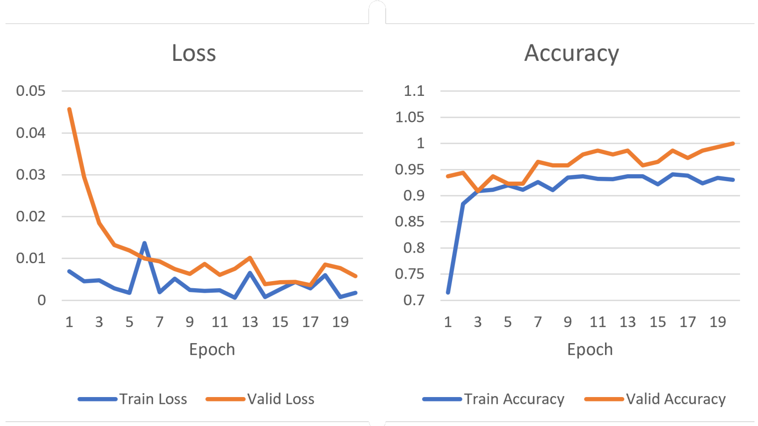

Figure 3. Training epoch to training loss, training accuracy, validation loss, and validation accuracy

Dataset	Accuracy	Loss	Precision	Recall	F1 Score
Training	0.930495	0.001757	0.931611	0.945988	0.938744
Validation	1.000000	0.005784	1.000000	1.000000	1.000000
Testing	0.979021	0.005039	1.000000	0.965909	0.982659

Table 2. Final Metrics Report

testing and validation sets with only around 100 samples each. We also note that the training loss is still lower than the validation and testing loss, which matches our expectations of the model performing better on the training set.

We report the confusion matrix of the testing set in 3. We note an effective discrimination of that three ICE trains were misidentified as Regional Bahn, which we hypothesize to be due to environmental conditions which make it difficult to identify the train.

In general, our findings match our expectations as we expected very good performance on our task. The original model, ResNet-18, had a 87.4% top-1 accuracy on ImageNet and a 96.3% top-5 accuracy on ImageNet (7). ImageNet has over 20,000 categories, meaning that it is a much harder task than our binary classification task. It also has a similar number of samples per category as our dataset. Thus, we expected to do better than ResNet did on ImageNet. Our accuracy of 97.9% thus matches our expectations. We also note that our precision, recall, and F1 scores were also very high, which also matched our expecation of the model being able to categorize ICEs and Regional trains equally well despite the imbalance in our dataset.

Overall, it took around a little more than an hour to train our model. This was only possible using transfer learning,leveraging existing weights and domain knowledge, which allowed us to reduce the amount of trainable parameters from 11 million to 131 thousand, around a 100x decrease in parameter count. We first note that the initial training classification accuracy was already quite high after the first few epochs and the model convergence was relatively smooth, indicating relatively fast adaption of the existing model to the new task.

We provide example classifications from the test set in Figure 4. We note that while the ICE sample is clearly disambiguated, there is a very small amount of belief in the ICE classification class for the RB, which we hypothesize may be attributed to slight streamlining in the shape of the trainset.

5 FUTURE WORK

In our work, we have presented a model which effectively disambiguate between images of ICE and RB trainsets. We note however, that we did not test our model with out of domain images such as non-trainset images, or images of non RB and ICE trainsets. Additionally our scope is limited to RB trainsets with the most common red livery and which does not represent the full scope of local trains within the Federal Republic of Germany. It is a strong assumption that we do not input any adversarial samples nor the presence of out domain samples. Another limitation is the collection of

		Predicted Class		
		ICE	Regio	Total
True Class	ICE	85	3	88
	Regio	0	55	55
	Total	85	58	143

Table 3. Test-Set Confusion Matrix

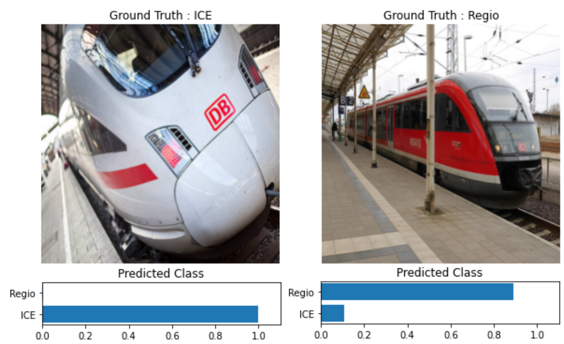

Figure 4. Sample classification probabilities of ICE and RB images from the test set

our dataset in that we only included $n = 1437$ samples due to time constraints in data collection, although our strong results on the test set indicate that the model was able to effectively generalize just based on the available data-points.

In the future we may elect to create classifiers which are robust to out of domain images and include a greater variety of train types, such as specific models of InterCity express and more variation of regional trainsets. Additionally the focus on livery rather than trainset for local trains is insufficient, since these two aspects of train appearance are independent of each other.

This multi-class extension of the project to a greater amount of trainset types could be used by railfans to help them identify specific trainsets. Coupled with data crowd-sourcing, this would allow for us to create heatmaps and gain insights on what types of trainsets are being used in what regions.

A particular interesting application of our model in its current form is that regional rail tickets (e.g. the Nine Euro Ticket) are not valid for long distance rail transport, so this model can be used to determine the validity of your ticket for travel on a given trainset/service. This project could also be used to identify which trains are currently running on the tracks and verify the location of their train. This would help prevent accidents and serve as an additional level of verification for each train's location.

REFERENCES

[1] Deutsche bahn's train fleet: From ice to s-bahn, 2022.

[2] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database.

[3] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

[4] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[5] Raghu, M., Zhang, C., Kleinberg, J., and Bengio, S. Transfusion: Understanding transfer learning for medical imaging, 2019.

[6] SHORTEN, C., AND KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data 6*, 1 (Jul 2019), 60.
[7] WIGHTMAN, R. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.
[8] ZHUANG, F., QI, Z., DUAN, K., XI, D., ZHU, Y., ZHU, H., XIONG, H., AND HE, Q. A comprehensive survey on transfer learning, 2019.