Landmark Recognition Final Report

Matthew Tran
University of Minnesota
Minneapolis, MN 55455
tran0923@umn.edu

Navya Ganta
ganta016@umn.edu

Alireza Khataei
khata014@umn.edu

Sai Pratyusha Attanti
attan005@umn.edu

1. Introduction

For many years, people have strived to use computers to find and recognize objects inside an image. Object recognition is a core area in computer vision that provides some techniques to find and identify things in a digital image or video. Have you ever visited a place or seen a photo and wondered where that place is? Landmark recognition is to help us get such information. As the name implies, this is an application of object recognition that specifically deals with landmarks. In other words, it predicts the label of such photos based on some feature extracted from the pixels of the image.

In this project we aim to use parts of the Google landmarks dataset v2 [9, 14] which contains more than 81K different classes to fine-tune and develop a pre-trained neural network (NN) to classify landmarks. This places our problem as an instance level recognition problem, where we care about which object we're looking at and not what. However, the process is not that straightforward and there are some challenges that we need to overcome. One of the most challenging issues is the limited number of images per class at the low end of the distribution which makes the direct learning difficult. On the other hand, some existing pre-trained NNs cannot be tuned to classify images with that number of classes. A promising solution to address these issues is data pre-processing and cleaning techniques which manipulate the data before it is used in the NN to increase the efficiency and performance of the system [11, 17]. These methods make the learning system capable of dealing with massive big data collected from different sources which may significantly impact the quality of results. Additionally, transfer learning is a promising technique we we will attempt to employ in order to circumvent these issues.

2. Related Work

Substantial work has been completed in order to grapple with the problem of landmark recognition varying from applications of classic data mining techniques to modern approaches leveraging the power of deep learning. Across these approaches common themes emerge

2.1. Data Mining and Classical AI

Before the deep learning revolution, many heuristic algorithms were applied in order to extract relevant information through the application of data mining techniques. In 2008, Zheng et. al. implemented a web scale landmark recognition system that leveraged classic techniques and representations [19]. Their group started by scraping image sharing services like picasa.google.com, which is now defunct, in order to gather a set of images descriptive of landmarks. This initial set was then expanded and cross referenced by scraping tour guide websites. After initial data collection, they extracted feature descriptors similar to SIFT based on Laplacian of Gaussian filters to describe points of interest in each image, applied PCA to reduce the dimensionality, and then performed point matching to characterize the similarity between images. With these matching scores, they performed agglomerative clustering with single link inter cluster distance to form image clusters that were characteristic of each landmark. To prune their model, they filtered for non-images (since many maps were accidentally mined) and images with too many faces which negatively impacted their clustering. The final model they generated contained 5312 landmarks densely populated in north America and Europe, reflecting the language bias of their data mining approach. Their model correctly identified 80.8% of landmarks detected, recognized that 46.3% of positive instances had landmarks to begin with, and had a false positive rate of 1.1%.

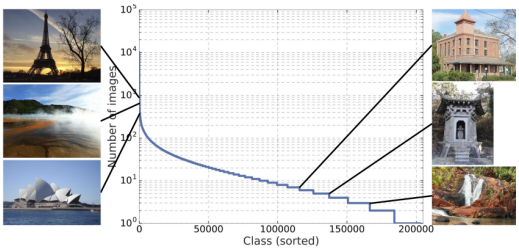
Their method generally struggled due to shared structures between non-landmarks and landmarks or regions in the landmark images that were not representative of the landmark and too general.

2.2. Google Landmarks Data Set Version 2

Much work has been completed in the field of Computer Vision since 2008. Earlier approaches including Zheng et. al. often had proprietary and unstable data sets which impeded the development and comparison of new methods.

Since the creation of shared data sets such as ImageNet, many shared and stable data sets have been developed [4]. For landmark recognition, the cutting edge data set is the Google Landmarks Data Set Version 2 (GLDv2), which easily exceeds the scale and difficulty of previous data sets [15]. GLDv2 has over 5 million total images, has an extremely long tailed class distribution, 200 thousand individual landmark classes, and high intra class variability with diverse perspectives and indirect relevance.

Figure 1. Class imbalance for GLDv2 [15]



Another significant design choice is the large imbalance in the test set between out of domain images (90%) and out of domain images(10%) which is designed to force models to have low false positive detections in order to perform well. With these general features, GLDv2 is specifically designed to promote the development of instance recognition and retrieval, of which we'll be focusing on recognition. There are also subsets of the data set provided which include the removal of the tail and a clean version that ensures visual coherence in the training set. Alongside the data, there is also a common implemented metric (μAP) which we will utilize to compare our approaches to other solutions in the field. While world spanning, the data set is still limited due to the user bias of Wikimedia commons resulting in under representation in countries beyond Europe and North America.

Figure 2. GLDv2 available training data sets [15]

Training set	# Images	# Labels
GLDv1-train [40]	1,225,029	14,951
GLDv2-train	4,132,914	203,094
GLDv2-train-clean	1,580,470	81,313
GLDv2-train-no-tail	1,223,195	27,756

2.3. Student Projects

At least two other student based course projects have been completed investigating landmark recognition on the earlier data set GLDv1, which is the smaller and unstable predecessor to GLDv2 [13]. In each approach, a smaller subset of the data set is utilized in order to simplify the overall problem for their course projects.

2.3.1 Google Landmark Recognition using Transfer Learning

Catherine McNabb et. al. implemented transfer learning using VGG16 combined with DELF to limit false positive detections [10]. They discuss data pre-processing, image augmentation (including shift, zoom, brightness, rotation, and sheer), the training and general approach to transfer learning, as well as DELF. A core motivating requirement of GLDv2 is that models have low false positive rates. To solve this, McNabb et. al. utilized DELF (DEep Local Features) to extract local features and match them to images with landmarks in them. In this way, they effectively trained a landmark classifier to find the specific instance and employed a landmark discriminator (DELF) to ensure detections were of actual landmarks. With their approach they achieved an 83% test accuracy on a specific subset they had withheld themselves for testing. It is unfortunately unclear if they tested their final model on the actual test set of GLDv1 to achieve this result since it seems their test data was only a subset of the less adversarial test set.

2.3.2 Google-Landmark Recognition with Deep Learning

Chien-Yi Chang utilized transfer learning with Inception v3 [3]. Their approach utilizes data augmentation with random shifts, rotations, flips, shears, and zooms. Another novel augmentation approach they employed was utilizing GANs to generate useful images for the under represented classes in the training data. Utilizing this augmented data set yielded an increase in top 1 accuracy of their baseline model from 0.03588 to 0.12874, demonstrating in principle the efficacy of their approach. Their final model achieved a top 1 accuracy of 68.75% on the subset of GLDv1 with at least image instances per class.

Our approach will be distinguished from these previous efforts since we'll be working with the second newer data set GLDv2 and employing more modern approaches. Beyond this, three important lessons can be gleaned from their previous work:

- Data augmentation can substantially improve performance.
- A discriminator is required in order to have reasonable performance on GLDv2's test set.
- Working with a subset of GLDv2 is substantially helpful in scaling back the difficulty of the problem.

These considerations and student group's approaches have largely guided our approach in our project. In terms of our project, our goal was to apply the results of past student groups on GLDv1 to the more modern GLDv2 data

set, thus exploring a "new" data set with previously tested approaches.

2.4. Baselines and the Cutting Edge

With the publication of GLDv2, many baseline (3) and cutting edge methods developed for the 2021 Kaggle competition (4) are discussed [15]. The baseline methods discussed were

- ResNet101+ArcFace [5]
- DELF (DEep Local Feature) [13]
- DELG (DEep Local and Global Features) [2]

The ResNet101 is employed directly while DELF and DELG are used to extract feature descriptions which are then used for clustering. The additional SP suffix refers to spatial verification which utilizes inlier counting [15]. DELG is the successor to DELF with superior performance achieved through a leveraging local and global information.

Figure 3. Results of baseline methods on GLDv2 [15]

Technique	Training Dataset	Testing	Validation
ResNet101+ArcFace	Landmarks-full [22]	23.20	20.07
	Landmarks-clean [22]	22.23	20.48
	GLDv1-train [40]	33.25	33.21
	GLDv2-train-clean	27.34	26.40
DELF-KD-tree [40]		44.84	41.07
DELG global-only [10]	GLDv1-train [40]	32.37	32.02
DELG global+SP [10]		**56.35**	**55.01**

Table 4: Baseline results (% μAP) for the GLDv2 recognition task.

The top three solutions to the 2021 Kaggle competition on GLDv2 are also discussed [6, 15, 16, 18]. Between these methods, prediction re ranking, ensembles, and general global features are utilized to achieve peer performance with DELG global+SP at the cost of added complexity

Figure 4. Results of Kaggle competition methods on GLDv2 [15]

Team Name	Technique	Testing	Validation	Before re-annotation Testing	Before re-annotation Validation
smlyaka [64]	GF ensemble → LF → category filter	69.39	65.85	35.54	30.96
JL [24]	GF ensemble → LF → non-landmark filter	66.53	61.86	37.61	32.10
GLRunner [15]	GF → non-landmark detector → GF+classifier	53.08	52.07	35.99	37.14

3. Data Augmentation

In Google Landmarks Dataset V2, there are some challenges that we need to address. Firstly, there is a inconsistency in the number of samples per class. For instance, some classes have 3 samples, whereas some other have 20 samples. On the other hand, the image dimensions vary from one sample to another.

To tackle these problems, we develop a Python class to generate more samples for the under-represented classes using data augmentation techniques. This class requires parameters *min_thr* and *max_thr*, which correspond to the minimum and maximum number of samples in each class that we need to train the neural network, respectively. In case the number of samples in a class is greater than *max_thr*, it randomly pick some of them to satisfy the *max_thr* limitation. If the number of samples in a class is less than *min_thr*, it will generate more samples using data augmentation methods as follows:

1. Changing brightness
2. Rotating by a random value
3. Flipping

After tuning the number of samples per class, we resize all the images to the same size using the zero-padding. Algorithm 1 shows the procedure of the data pre-processing and augmentation step by step. Figure 5 shows a sample image and its corresponding augmented images.

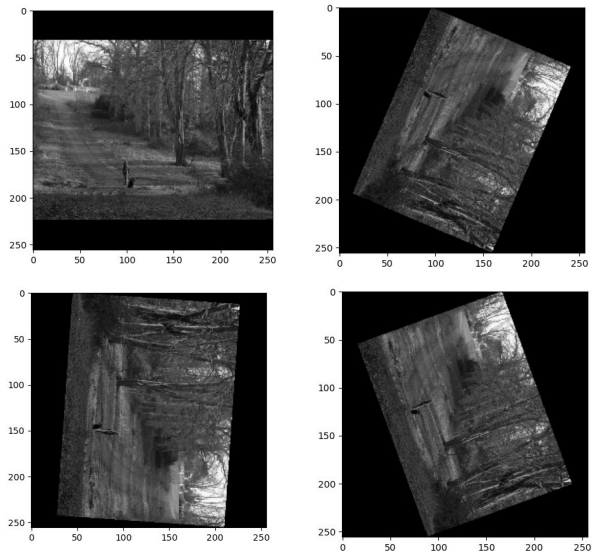
Algorithm 1: Training Data Augmentation and Preprocessing

Parameters: min_thr, max_thr
Input: $in\_samples$
Output: $out\_samples$
$in\_samples \leftarrow in\_samples/255$
if $len(in\_samples) > max\_thr$ **then**
  $idx \leftarrow arange(0, max\_thr)$
  $idx \leftarrow shuffle(idx)$
  $out\_samples \leftarrow in_samples[idx]$
else
  $out_samples \leftarrow in_samples$
  $angle \leftarrow randint(0, 360)$
  $dir \leftarrow randint(-1, 1)$
  $brightness \leftarrow random(0, 1)$
  $i \leftarrow 0$
  while $len(out_samples)! = min\_thr$ **do**
    $sample \leftarrow rotate(in\_samples[i], angle)$
    $sample \leftarrow flip(sample, dir)$
    $sample \leftarrow sample \times brightness$
    $out\_samples.append(sample)$
    $idx \leftarrow idx + 1$
  end
end
return $out\_samples$

4. Baseline Method (Transfer Learning)

As our baseline approach, we used transfer learning [20] for classifying landmarks. The main challenge with Google

Figure 5. Original and augmented images



Landmarks Dataset V2 is a severely uneven class distribution. So building a CNN model from scratch can lead to overtrained models. So as our starting point, we used the DenseNet-201 [7] network pre-trained on the Imagenet [4] dataset for transfer learning and carefully fine-tuned the last few layers of the model by freezing the initial layers to get better accuracies. This model is pre-trained on the imagenet dataset which has similar characteristics to our dataset. Also, because of its dense network, it reduces the vanishing-gradient problem and strengthens feature propagation as every layer is connected to every other layer in a feed-forward manner. So considering the above advantages it is ideal to use DenseNet-201 for transfer learning. We also explored other state of art literature classifiers like ResNet-50, Inception V3 [8, 12] etc. but these models resulted in less accurate results.

Our baseline classifier DenseNet-201 has 5 dense blocks, where each block is connected to other blocks in the network. Each dense block has convolutional layers, the transition layer has convolutional and pooling layers, and a Global Average Pooling at the end of the last dense block. The output is fed to the classifier that classifies the images into different classifiers. In our approach, we freeze the first 101 layers of the network and fine-tuned the remaining 100 layers to make the network more adaptable to our dataset. For our starter code, we used the code provided by the Keras library [1] and modified it to suit our dataset. And finally, the softmax classifier is added at the end of the fully connected layer to classifier the landmarks.

We performed a 80-20 split to divide a subset of the training data into training and validation datasets, which consisted of around 116 landmark classes and ∼15,000 instances. The reasoning behind utilizing this substantially smaller subset of the data set is discussed in the challenges section of our report 7. All the images are resized to 128*128 before they are fed to the network to reduce the computational complexity of the model. Information about the hyperparameters used for the training is shown in Table 1. The results of this method are represented in the Results section

Table 1. Hyperparameters used for training

Parameters	
Optimizer	ADAM
Loss Function	Categorical Cross Entropy
Batch Size	32
Learning Rate	0.001
Dropout	0.2
Epochs	100

5. Proposed Method

Due to the various challenges 7 our group faced with computational resources and wrangling the data set, we were largely unsuccessful in completing the proposed extension for our project. Our original planned extension was to utilize the DELG feature extractor in order to extract effective descriptors for each image in the test set, which would function similarly to the SIFT based approach we have previously pursued in class [2]. From there, we could utilize these features with the KNN algorithm in order to predict which landmark the test image was, or, if the test image was sufficiently far away based on a threshold it would be categorized as a non-landmark. Alternatively, we could have utilized the pre-made retrieval system provided on the DELG github repository for this purpose. The greatest difficulty our group encountered with working with DELG was installation and usage. While instructions were provided by the original authors, due to the rapid pace of development in computer vision many of the required libraries had been substantially changed, breaking the installation. Additionally, specific instructions for recovering the author's results were only available for the retrieval task since the link for recovering the instance recognition results led to a 404 page, further stymieing our efforts. While not substantial towards our original goal, our group was at least able to recover the results of the GLDv2 authors on the revisited Oxford data set for the retrieval task, minimally demonstrating the functionality of their model 8.

6. Results

6.1. Quantitative Results

Our group applied our model to unseen images within the training set comprising of 3689 "testing" images across 116 different classes. This test set was comprised of a subset of the unseen training set, which, for our work, would function as a testing set. On this effective test set, our model achieved an accuracy of 47.6% ($\frac{correct}{correct+incorrect}$). Since our test set wasn't perfectly uniform, we also investigated other metrics to evaluate its performance. Micro average precision, which was presented in the GLDv2 paper, was also utilized with $\mu AP = \frac{1}{M}\sum_{i=1}^{N} P(i)rel(i)$ where M is the number of test images with landmarks, $P(i)$ is the precision at rank i, N are all of the predictions our model made, and $rel(i)$ is a binary indicator specifying whether our model made a relevant prediction (i.e. was it correct). This metric is calculated after sorting all predictions by their confidence with higher values representing stronger results. With all correct predictions we could expect μAP of 1 and with all incorrect predictions we would achieve μAP of 0.

In the end, our model achieved 11.5 μAP on our reduced test set. This low μAP is partially caused by our accuracy of around 47.6%, causing around half of the $rel(i)$ to be zero. However, this metric also highlights the weaker ranked precision of our model since if we had achieved perfect precision at all ranks, our μAP would be 47.6. This implies that, while our final model has alright accuracy, it's ranked precision in terms of confidence is weaker.

In terms of comparing our model's performance, due to our simplification of the training and test set, we cannot fairly compare our model against the existing models in the literature. Based on our μAP, it appears that our model is weaker than the comparable baseline in the original GLDv2 paper, which was a ResNet101 model retrained with ArcFace loss which achieved 23.20 μAP on GLDv2's actual test set [14]. This comparison, however, is tenuous at best. A fairer comparison is against a theoretically random predictor. With an even test class distribution, we would expect around $\frac{100}{116} = 0.862\%$ accuracy. For our unbalanced case, the average accuracy of a random classifier, which randomly selects the correct class for each instance, has an accuracy of 0.8612% as expected. In this way, the classifier we've trained on our reduced data set is substantially stronger than random, at least in aggregate.

However, referencing the accuracy matrix 6 yielded by our model on the test set, it's clear that our model performs strongly on only a handful of classes but poorly on others. A summary of the accuracy distribution 7 across 10 bins, which suggests around 70% of classes have an accuracy worse than 60%. Overall, these results from the baseline model demonstrate the efficacy of utilizing transfer learning while highlighting that further tuning and training would likely yield better results.

Figure 6. Accuracy Matrix for All of the Classes in the Testing Set

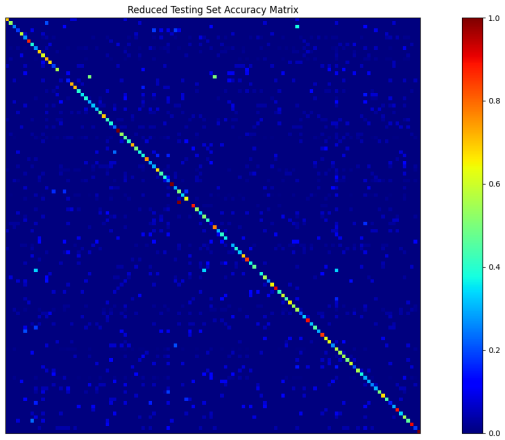
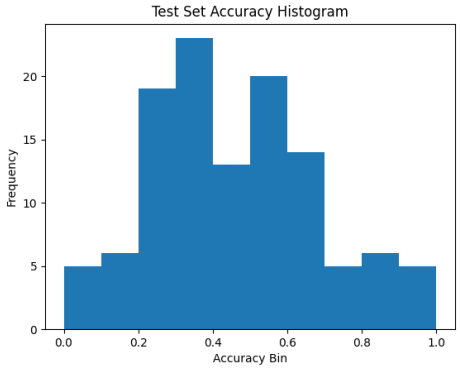


Figure 7. Accuracy Histogram for All Classes



In terms of the partial results for our proposed method, the best we could achieve was recovering the results produced by the DELG retrieval method on the revisited Oxford data set 8 [14]. The results of the original authors were 76.2 μAP on the medium difficulty data set and 55.6 μAP on the hard data set, in line with our reconstructed results.

Figure 8. Recovered Results on the Revisited Oxford Data Set

```
hard
  mAP=55.59
  mP@k[ 1  5 10] [88.57 80.86 70.29]
  mR@k[ 1  5 10] [19.46 33.8  42.62]
medium
  mAP=76.25
  mP@k[ 1  5 10] [95.71 92.86 90.29]
  mR@k[ 1  5 10] [10.17 25.96 35.18]
```

6.2. Qualitative Results

Our model was best able to identify land marks with visually distinctive features as seen in figure 9 while it struggled with landmarks with indistinct or extremely varied views 10. The qualitative difficulty our model encountered was likely due to the incoherence of some images which could be addressed in the future by utilizing the clean split of the training data set provided with GLDv2, which has been vetted for visual coherence [14].

Figure 9. Best Classes Our Model Predicted

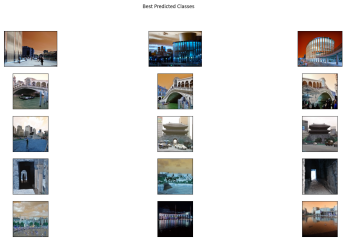
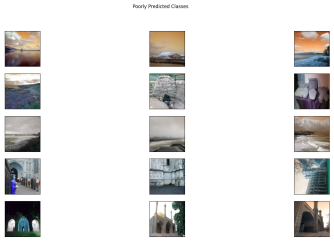


Figure 10. Worst Classes Our Model Predicted



7. Challenges

The most challenging aspect of our project was the scale of the data set. With around 4 million images, the training set alone for GLDv2 is around 2TB. While having more data is generally good, in our project we were severely limited by the hardware we had available to us.

Since GLDv2 is so large, we were unable to download the entirety of the data set to any of our local machines. With Google Colab, even with premium we were limited to only 200 GB of storage which was less than 5% of the data. Additionally, since Colab is a VM based environment with temporary storage if we had reached the end of our session, all of the data we were working with would be deleted.

One solution we attempted was downloading data via Google Colab into Google drive, since, Google drive is a static relative to Colab. However, within Colab, mounted Gdrives are also virtual - they're only available as virtual drives within the Colab disk space. As a consequence, whenever data was downloaded via Colab into Google drive, we didn't actually have access to further storage, only really access to a read only space. As further insult to injury, it was unclear when or what data would actually be written out to Google drive when written to the mounted drive which led to around 8 hours of repeated attempts to download a subset of the data being wasted after VM exit.

In the end, we decided to manually download a subset of the data set and then *reupload* it to Google drive which was slow due to the sheer number of small files being uploaded.

As a further consequence of the scale of our data set and resolution of the images, training time was also extensive. Even with the reduction in scale, training was still substantially slow. To solve this, our group decided to sacrifice performance with smaller image sizes in exchange for faster training times.

8. Conclusion

Through our efforts, our group was able to apply transfer learning in order to retrain a classifier on a reduced version of the GLDv2 dataset, (115 classes, $\sim$15,000 images) with an accuracy of 47.6%. This model was substantially stronger than a random classifier on our reduced training and testing set, minimally demonstrating the efficacy of the approach. Beyond this, comparison to other solutions is not feasible due to the simplifications employed to improve the tractability of our problem.

In the future, with better computational and storage resources our group would be able to approach the true challenge of this data set with models trained and tested on the entire data set instead of a subset. This includes the challenge imposed upon models per the design of GLDv2's test set consisting of a 10% in domain and 90% out of domain split. With our current approach and similar transfer learning approaches, our models would likely struggle on the true or even a subset of the true test set due to this challenge since our model would not have been trained to account for non-landmarks. To address this, a complete implementation of a DELG based retrieval system on our data set would solve this issue as has been demonstrated in the literature.

While we weren't fully successful in our project, our work demonstrates the value of transfer learning in approaching challenging problems. A major lesson we learned is that the scale of the problem you're able to solve is largely determined by the power and capability of the hardware you have available. Unfortunately in our case, the problem we set out to solve was around an order of magnitude larger than we were capable of approaching, harming our efforts to develop a comparable and truly effective solution. With the lessons and skills we've learned through our project, our group will be better capable of approaching more feasible problems in the future.

References

[1] Transfer learning using keras library. 4

[2] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for efficient image search. *CoRR*, abs/2001.05027, 2020. 3, 4

[3] Chien-Yi Chang. Google-landmark recognition with deep learning. *Student Projects*, 2021. 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 4

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 3

[6] Christof Henkel. Efficient large-scale image retrieval with deep feature orthogonality and hybrid-swin-transformers. *CoRR*, abs/2110.03786, 2021. 3

[7] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 4

[8] Mahbub Hussain, Jordan J. Bird, and Diego R. Faria. A study on cnn transfer learning for image classification. In Ahmad Lotfi, Hamid Bouchachia, Alexander Gegov, Caroline Langensiepen, and Martin McGinnity, editors, *Advances in Computational Intelligence Systems*, pages 191–202, Cham, 2019. Springer International Publishing. 4

[9] Zu Kim, André Araujo, Bingyi Cao, Cam Askew, Jack Sim, Mike Green, N Yilla, and Tobias Weyand. Towards a fairer landmark recognition dataset. *arXiv preprint arXiv:2108.08874*, 2021. 1

[10] Catherine McNabb, Anuraag Mohile, Avani Sharma, Evan David, and Anisha Garg. Google landmark recognition using transfer learning, Dec 2018. 2

[11] Junfei Qiu, Qihui Wu, Guoru Ding, Yuhua Xu, and Shuo Feng. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016(1):1–16, 2016. 1

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. 4

[13] Giorgos Tolias, Tomas Jenicek, and Ondřej Chum. Learning and aggregating deep local descriptors for instance-level recognition. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 460–477, Cham, 2020. Springer International Publishing. 2, 3

[14] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2575–2584, 2020. 1, 5, 6

[15] Tobias Weyand, André Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2 – a large-scale benchmark for instance-level recognition and retrieval. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2572–2581, 2020. 2, 3

[16] Cheng Xu, Weimin Wang, Shuai Liu, Yong Wang, Yuxiang Tang, Tianling Bian, Yanyu Yan, Qi She, and Cheng Yang. 3rd place solution to google landmark recognition competition 2021, 2021. 3

[17] Natalia Yerashenia, Alexander Bolotov, David Chan, and Gabriele Pierantoni. Semantic data pre-processing for machine learning based bankruptcy prediction computational model. In *2020 IEEE 22nd Conference on Business Informatics (CBI)*, volume 1, pages 66–75, 2020. 1

[18] Yuqi Zhang, Xianzhe Xu, Weihua Chen, Yaohua Wang, Fangyi Zhang, Fan Wang, and Hao Li. 2nd place solution to google landmark retrieval 2021. *CoRR*, abs/2110.04294, 2021. 3

[19] Yan-Tao Zheng, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua, and Hartmut Neven. Tour the world: Building a web-scale landmark recognition engine. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1085–1092, 2009. 1

[20] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *CoRR*, abs/1911.02685, 2019. 3