Have Fun Storming the Castle(s)!

Connor Anderson Adam Teuscher Elizabeth Anderson
Alysia Larsen Josh Shirley Ryan Farrell
Brigham Young University
connnor.anderson@byu.edu farrell@cs.byu.edu
{adam.m.teuscher,e13cook,alarsen93,joshshirley9}@gmail.com

Figure 1: **Castles Dataset.** The proposed Castles dataset consists of over 770k images from more than 2,400 castle instances. Bodiam Castle is shown here as a mosaic of images from the dataset.

Abstract

In recent years, large-scale datasets, each typically tailored to a particular problem, have become a critical factor towards fueling rapid progress in the field of computer vision. This paper describes a valuable new dataset that should accelerate research efforts on problems such as fine-grained classification, instance recognition and retrieval, and geolocalization. The dataset, comprised of more than 2400 individual castles, palaces and fortresses from more than 90 countries, contains more than 770K images in total. This paper details the dataset's construction process, the characteristics including annotations such as location (geotagged latlong and country label), construction date, Google Maps link and estimated per-class and per-image difficulty. An experimental section provides baseline experiments for important vision tasks including classification, instance retrieval and geolocalization (estimating global location from an image's visual appearance). The dataset is publicly available at vision.cs.byu.edu/castles.

1. Introduction

> *"The castle, and all it represents, will always be with us. Once it was born, once the stone was made living, the repository of power made real, the idea could never be unmade. Even if all the castles of all the world were destroyed, in the minds of men they would be built anew; the wizard called imagination would raise high walls and towers out of ruins."*
>
> *- David Day*

The ImageNet dataset [4] became a catalyst in 2012 [19] for the success of deep learning methods which rapidly transformed the field of computer vision. In the ensuing years, dozens of increasingly large-scale datasets, each typically tailored to a particular problem, have fueled continued progress and innovation. Future progress will require both novel theoretical/algorithmic advances and increasingly challenging datasets to help us see and identify

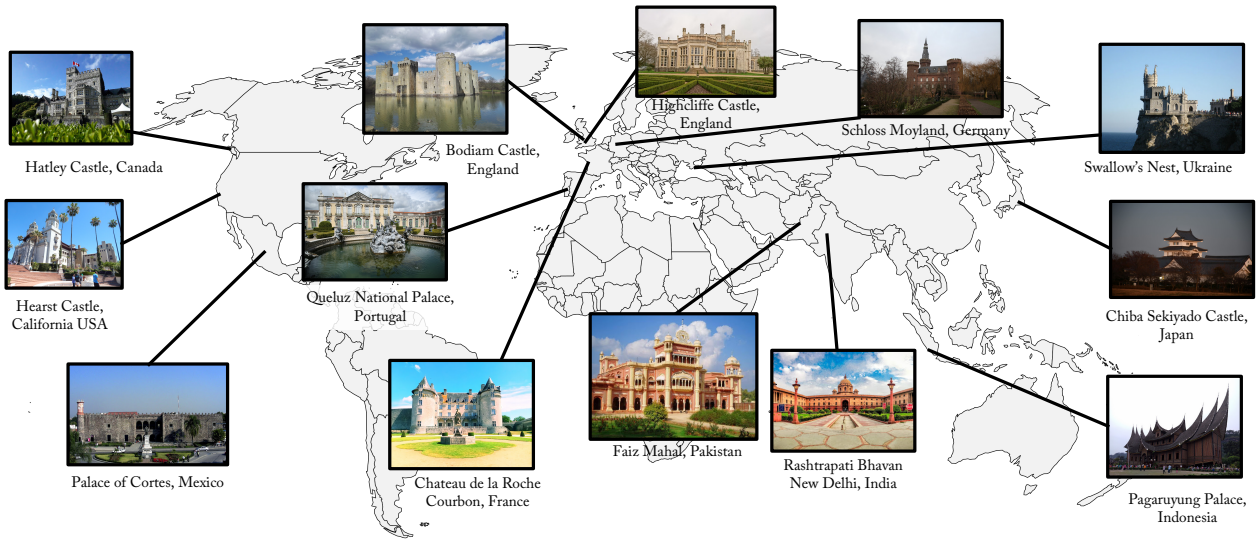


Figure 2: **Dataset Examples**. The images in our dataset represent castles and palaces from over 90 countries. This figure exhibits a sample of images from our database from varying countries.

the shortcomings of prior methodologies.

With this in mind, this work proposes a new dataset aimed at tasks such as fine-grained instance recognition, retrieval and geolocalization. While there are certainly existing landmark/place/scene datasets (see Section 2), the proposed dataset intentionally deviates from the pattern of highly-varied large-scale landmark/place recognition datasets, instead acutely focusing on a single fine-grained domain, Castles, as the subject domain of the dataset. Recognition in fine-grained domains cannot simply be solved with the same approaches as large-scale recognition – there are unique and significant challenges imposed by high intra-class variation and very low inter-class variation.

Why Castles? Among the oldest structures on our planet, castles constitute a fascinating part of our history. They inspire the imagination with their strong walls and high towers, often perched on hills or cliff faces. They also provide an interesting challenge from a computer vision perspective: they are large structures which can be viewed from many angles; they are subject to many kinds of occlusion; and they have recognizable features at multiple scales.

We should note that we use a loose definition of "castle" in this work, including some buildings that would more accurately be defined as palaces, mansions or manors, and some others that are actually forts or compounds. Our focus is less on specific function (such as defense) and more on "castle-like" and palatial architecture. We exclude many ruined castles. We also make no attempt to be comprehensive. Our proposed dataset does include over 2400 castles, many well known and others less so.

This paper makes two significant **contributions**:

- First, a fine-grained/instance-level dataset with an order of magnitude more classes and nearly two orders of magnitude more images than existing fine-grained datasets. The dataset also provides a geolocations, country label, approximate construction date (on 85% of the classes) and a Google Maps link for each class in the dataset.

- Second, baseline experiments are provided for several important vision tasks including fine-grained instance recognition, superclass (country) prediction, fine-grained image retrieval and geo-localization of held out query images.

2. Background and Related Work

There are several areas of computer vision that are relevant to this work, which we discuss here.

Fine-grained and Instance Recognition Fine-grained recognition is the task of distinguishing between subtle subcategories of objects, such as different species of birds [36, 32], dog breeds [16, 30], models of cars [18, 43] or aircraft [21], or types of food [2, 15]. Recent efforts have been made to create challenging datasets that are both fine-grained and large-scale, such as iNaturalist [33], Products 10k [1], iMaterialist Fashion [7], Herbarium [31], or iMet [44].

Instance recognition [11] is the extreme end of fine-grained, where each "class" consists of a single entity, such as in face recognition or landmark recognition. When instance recognition focuses on biological markers (face, fingerprint, stripe-patterns [3]), is is typically called biometrics.

Place Landmark Datasets A number of datasets for landmark and location recognition have been proposed over the years. The original Google Landmarks [45] dataset is an early example. The Google Landmarks v2 [39] dataset is a recent large-scale landmark dataset built specifically for benchmarking progress in instance recognition and retrieval. It is very large and very diverse, with over 5 million images of more than 200k landmarks. One particular challenge with Landmarks v2, which might be considered a downside in some settings, is that images for each landmark can contain visual entities that wouldn't be considered the landmark itself: for instance, pictures of a band playing inside a bar, which might not contain any visual information about the bar (the landmark) itself.

Hotels-50K [29] is a large dataset of hotel-room images, developed to help fight human trafficking by identifying hotels where human trafficking pictures have been taken. Hotels-50k has over a million images of 50,000 different hotel interiors.

Places2 [46] is a very large scene recognition dataset with 10 million images from 434 scene categories. Places2 builds on the ideas of the SUN Database [41, 42], a comparatively smaller scene recognition dataset. Other specialized datasets include KITTI [6], a dataset consisting of video and sensory data from cars, or the TorontoCity dataset [37], which represents the culmination of an enormous effort to digitize much of the city of Toronto.

Image Geolocalization Im2GPS [8, 9, 14] was the first attempt to do image geolocalization on a global scale. Later efforts incorporated SIFT-based features and "lazy learning" [9], and eventually CNN feature extractors [35]. PlaNet [40] dispensed with the reference database by framing geolocalization as a classification problem, dividing the surface of the earth into cells that correspond to target classes. A classifier outputs a probability distribution over the cells that expresses a level of uncertainty about where the image was taken. Subsequent work [26] improved the cell partitioning technique. Two additional related applications include interesting work on Webcam Localization [13] and Photo-tourism [27, 28]. More recent work explores using hierarchical information [22] and novel loss functions [20, 12].

3. The Castles Dataset

"Time builds castles, and time destroys them."
Serbian Proverb

3.1. Building the Dataset

In order to construct a large dataset of castle instances, we first needed to compile a large collection of castle names and locations. There are many small and partial lists of castles on the web, scattered across travel sites, local government sites, blogs, and wikis; however, these lists tend to be limited, repetitive, and not convenient for large-scale search. Instead of relying on these partial lists, we used Google Maps to search metadata about castles, including names and locations.

Compiling a Castle List We began by searching for potential castles on Google maps. We used search terms such as "castles in [country name]" for over forty different countries, and compiled a list of the returned results. This led to an initial list of around 5000 potential castle locations. This initial list contained many castles from the countries we searched in, and castles from other nearby countries as well. It also contained many locations that were not castles, such as businesses, hotels, or national parks — even a few bouncy castle companies. In order to sort out castles from other locations, we developed a web interface to quickly scan through locations and provide an initial accept or reject decision. The interface showed the name and any summary information collected from Google Maps, along with the first few images returned from a Google image search using the location name as the search term. Using this interface, we were able to fairly quickly sort through the initial 5000 and reject 30-40% that were either not castles, or else too ruined for our purposes.

Another issue we encountered was that there were duplicates in our initial list. Often these duplicates referred to the same location but used different names, sometimes in different languages. To deduplicate, we first obtained the Google Maps link corresponding to each location. From the Maps link, we retrieved the latitude-longitude location for each castle. We then calculated the pairwise distances between all locations, and manually inspected any that were close together. We were able to detect and remove a number of duplicates this way. We used the Vincenty distance for accurate geospatial distances, and a Ball-tree data structure for efficient nearest neighbor searches. After removing duplicate locations, we were left with a list of around 2500 castles.

Obtaining images Using our list of castles, we obtained images from user-uploaded photos associated with each

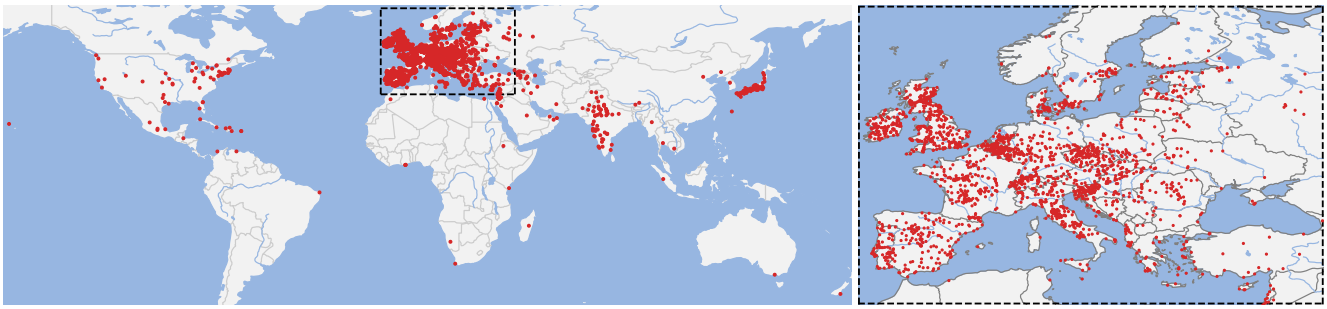


Figure 3: **Distribution of Castle Locations**. The dataset contains castles from around the world. The highest concentration is in Europe.

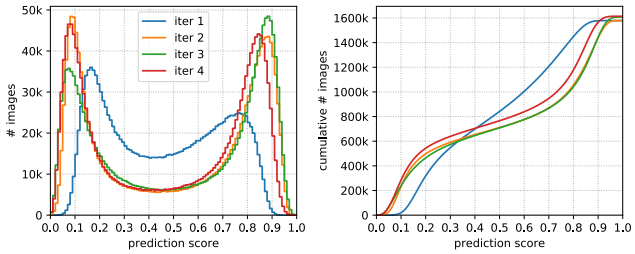


Figure 4: **Binary Castle-image Scores**. Images were included in the dataset based on a binary classifier trained on manually labeled images. The labeling process included several iterations of manual labeling and classifier retraining. The distribution of classifier scores is shown for four iterations. The classifier was trained with soft labels — 0.1 for not-castle, 0.9 for castle. The final threshold for inclusion was 0.6.

castle location on Google Maps. Some locations had few-to-no images available, and we dropped those castles from the our final set. We downloaded up to 1000 images per castle, or as many as were available. In total, we collected over 1.6 million images.

Pruning images The primary purpose of this dataset is to visually capture the exterior architecture of castles at the macro scale; we did not want to include images of the interior, close-up shots of small features (such as a single window, statue, etc.), adjacent or nearby buildings, or views of the grounds and surrounding landscape which do not include the castle itself. However, many of the images we downloaded fell into one of those categories (see Figure 5).

With such a large set of images, it would be prohibitively expensive to manually sort out the ones we wanted to exclude. Instead, we relied on a semi-automated procedure. We started by labeling several hundred images by hand. For our purpose, we used a binary label indicating whether the image contained the kind of view of a castle that we were looking for or not. We then trained a binary classifier on the manually labeled images, and used it to generate predictions for the rest of the images. We used a sigmoid function

Images	Categories	Countries
772,927	2412	93

Table 1: **Dataset Statistics.**

on the raw classifier outputs to get a pseudo-probability for each image, and then sorted them based on their predictions, helping us to focus on the areas of confusion as we labeled more samples. We repeated this process several times, eventually annotating roughly 16,000 images. After training the final classifier, we chose a conservative prediction threshold (we used 0.6) and kept only the images with higher prediction scores. Figure 4 shows the distribution of prediction scores for several iterations of labeling and retraining. After thresholding, roughly 773,000 images remained. In order to facilitate multiple train/val/test splits (see Section 4), we further removed any instances with less than 5 images. The final image statistics are shown in Table 1.

3.2. Relation to Fine-grained/Large-scale Datasets

As outlined above in Section 2, there are a number of fine-grained and/or large-scale datsets available to and frequently used by the vision community. The two most relevant types to our discussion are: fine-grained datasets like CUB Birds [36], Stanford Cars [18], FGVC Aircraft [21] or iNaturalist [34]; and large-scale landmark/place/scene recognition datasets such as Google Landmarks [34], SUN Database [41], and MIT Places [46].

Fine-grained recognition is a very challenging domain due to the high degree of perceived visual ambiguity between very similar classes; the differences can be extremely subtle—*inter-class* variation can be extremely low. Conversely, the *intra-class* variation (differences between images of the same class) can be very high due to changes in pose/articulation, viewpoint and illumination. Fine-grained datasets are tremendously difficult to curate, often requiring substantial domain expertise to generate datasets of 100-200 closely-related classes (all from a shared domain) and typically 50-100 images per class.

Figure 5: **Examples of Rejected Images**. Images deemed irrelevant to the exterior of castles were rejected. Examples shown here include non-target subjects, paintings or models of castles, castles that are severely ruined, reliefs/details, shots from castles, and images from the interior.

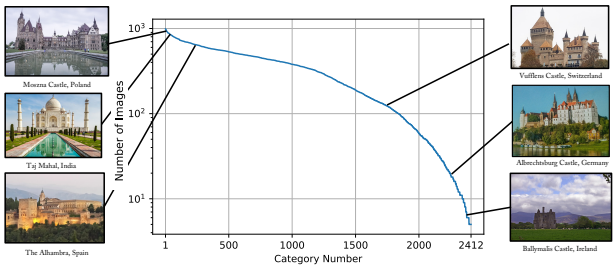


Figure 6: **Distribution of Images**. Each image in the dataset was assigned a category number and the total number of images in the dataset was recorded and graphed. Images shown in the figure are examples of images at various locations along the distribution. Images of large or famous castles tend to have more images than unknown or obscure castles. More than 75% of the images in the dataset have more than 100 images and 55.6% have 250 or more.

Large-scale place/scene datasets, on the other hand, can have a similar (though larger) number of classes – SUN and Places have 397 and 365 classes, respectively – but often have far more images per class. The Places-2 dataset includes up to 40,000 images per class. The Google Landmarks dataset couples a huge number of classes (three orders of magnitude large, over 200K) with a large number of images (more than 4 million in total).

Our proposed castle dataset focuses on combining the challenges from both the large-scale landmark/place and the fine-grained dataset regimes. We couple the scale of thousands of classes across a class-imbalanced distribution with the intrinsic challenges inherent to closely-related categories that exhibit high visual similarity.

3.3. Ranking by Difficulty

An important facet that we have endeavored to bring focus to is the notion of inherent difficulty. In fine-grained and large-scale datasets, highly similar categories/images are generally treated just like highly dissimilar categories/images: they're all just *different*. Viewing things in this fashion, however, hides a vitally important and interesting dimension to the recognition problem – there is a spectrum of *different* which could appropriately be viewed as gradations of difficulty. Recognition of highly similar categories/images is fundamentally more challenging than that of highly distinctive/dissimilar categories.

Difficulty exists and can be assessed or estimated at the image or at the category level. Some images are inherently difficult while others are easy. Similarly, there are varying degrees of underlying visual/structural similarity between whole categories, not just images. We draw attention here to this interesting dimension of the problem, but further describe our efforts to quantify difficulty in Section 4.5.

4. Experiments and Analysis

We run several experiments to provide baseline performance on a few different recognition tasks using the Castles dataset. We train convolutional neural networks for two related classification tasks: castle instance recognition, where the goal is to predict one of N possible classes for each image; and country-location prediction, where the goal is to predict the country where a castle is located for castles not seen during training. We also train models for the task of image retrieval where the goal is to rank images in a gallery based on semantic (class) similarity to the query images. Using our image retrieval models, we also provide a naive baseline for geo-location regression.

Dataset Splits In all of our experiments, we train models on five different train/val/test splits and report aggregated results. We obtain these splits by dividing the data into five folds and taking different combinations of three for training/validation, with the remaining two for test. We use one third of the train-val data for validation: to monitor model performance during training and to select the best model for measuring test performance. After training, we report performance on the held-out test subset.

For instance recognition, we divide the data into five folds such that the images for any given class are equally divided, preserving the overall proportion of class labels in each fold. For country prediction, we first remove any castles belonging to countries with less than five castles represented in the dataset; then we evenly distribute the castles from each country among the five folds, preserving the overall ratio of castles in a country within each fold. For retrieval, we evenly distribute the number of castles across the folds, so that each fold contains a unique one-fifth of the castle instances.

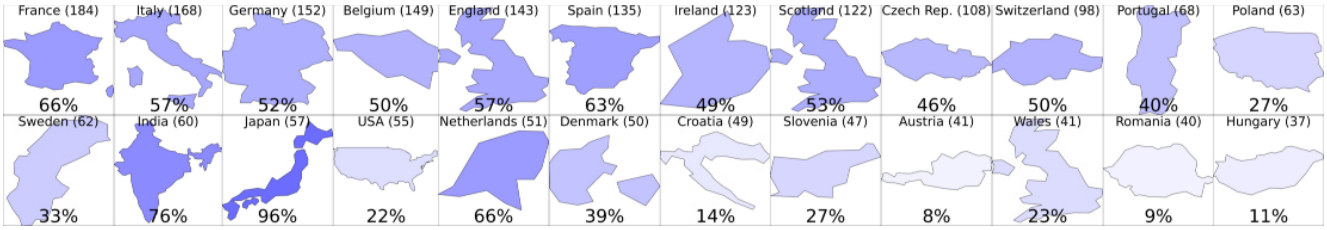


Figure 7: **Country Prediction**. The percent of castles that were correctly predicted for each country (top 24 by number of castles). The country color saturation correlates to the percentage and the number of castles per country is listed next to the country name.

Layer	C-in	C-out	P
BN-DO-FC-ReLU	2048	1024	0.25
BN-DO-FC	1024	num-class	0.5

Table 2: **Classification Head**. Replaces the single linear layer of Resnet during recognition experiments. **Key**: *BN* is batch normalization; *DO* is dropout; *FC* is a fully-connected linear layer; *C-in* and *C-out* are the number of feature channel inputs and output; *P* is the dropout probability.

General Training Details All of our experiments are run using Pytorch and Pytorch Lightning [25, 5] on a machine with 8 GTX 1080 Ti GPUs. Unless specifically noted otherwise, we use the following settings in each experiment:

- Resnet-50 [10] as the backbone network, initialized from ImageNet-pretrained weights
- Adam [17] as the optimizer
- Train for 30 epochs, with a learning rate schedule using a linear warmup from 0 to 10^{-3} over 9 epochs, and a cosine decay back to 0 over the following 21 epochs
- Training data augmentation: random resized crops with resolution 224x224, color jitter, and random grayscale conversion. For validation/test: resize images to resolution 256 in the smaller dimension and take a square 256x256 center crop

4.1. Instance Recognition

For instance recognition, we perform standard N-way classification using five different splits and report the aggregate performance. Results are shown in Figure 8a. We report predictive accuracy on the test set, both globally (for all images together) as well as within each class individually. Results are averaged across five splits, and min/max values are shown for the per-class distribution.

At just below 90%, the recognition performance is quite good. This is likely due, at least partially, to a number of distinctive castles that have many images which are easy to recognize. Indeed, when we normalize for the number of images per-class we see a drop in accuracy from 89.8% to 82.5%, and we can see from Figure 8a that there are a number of classes for which performance tends to be low. Figure 8a also shows performance by number of images in the class, and we can see that classes with more images tend to perform better on average than those with fewer images.

Mean accuracy	Std. dev.
44.1%	±1.1%

Table 3: **Country Prediction**. Overall accuracy for country prediction on held out set of castle instances. The accuracy and standard deviation are reported over the five different test splits.

Details We replace the final linear layer of Resnet-50 [10] with an extended classification head (see Table 2). Models were trained using distributed-data-parallel training on 8 GPUs, with a batch size of 96 per GPU.

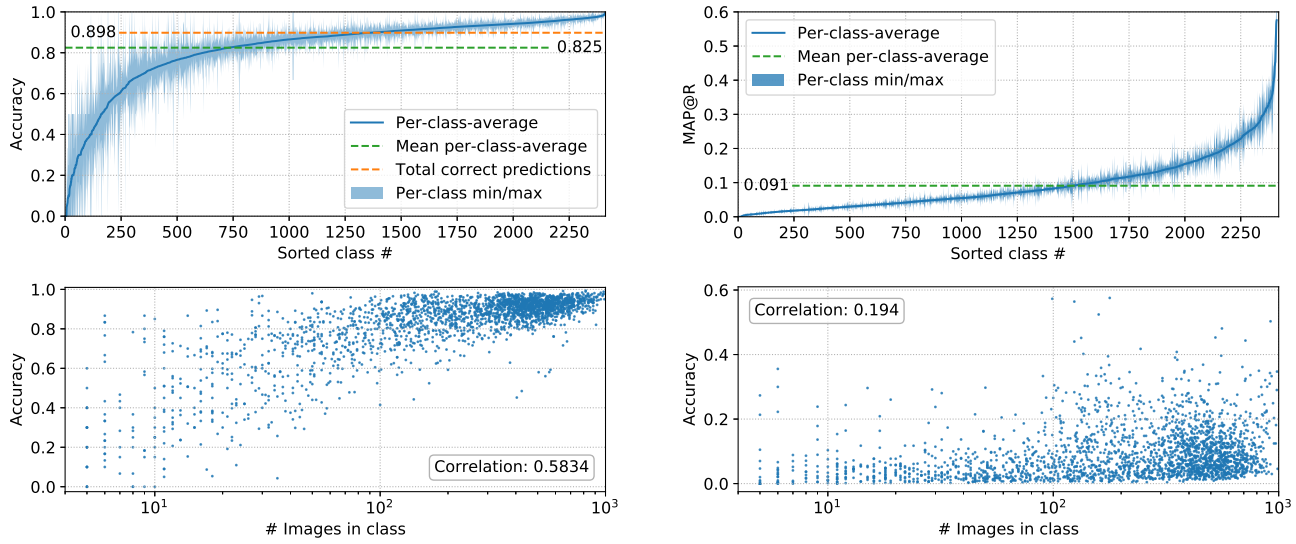
4.2. Country Prediction

We treat country prediction as another N-way classification problem, where N in this case is the number of countries represented by the various castle instances. All images belonging to any castle located in a given country are assigned the same country label. For testing, we hold out all the images belonging to a subset of the castles from each country; predictions are made by images from castles that were not seen during training, about the country where that castle is located.

Results for country prediction for the top 24 countries (by number of castles) are shown in Figure 7. In some cases, the results agree well with intuition. Japan, for instance, has extremely high accuracy; this makes sense, since Japanese castles have a style that is distinctive from any of the other countries. Some of the other results are interesting, such as the fact that the Netherlands is fairly high, but Austria is extremely low. Table 3 reports the mean overall accuracy and standard deviation across the five splits.

4.3. Image Retrieval

The goal of image retrieval is to match a query image to gallery images of the same class. We use a held-out set of castle instances as the test set, which serves as both the



(a) **Instance Recognition** (b) **Image Retrieval**

Figure 8: **Instance Recognition and Image Retrieval Results**. (a) The overall accuracy (proportion of correct image predictions), along with the average per-class accuracy and the distribution of per-class accuracies (sorted by average performance) averaged over five splits (top). Class accuracy vs. image count (bottom), showing strong correlation between number of images in a class and class accuracy. (b) Distribution of average per-class MAP@R (top): each class appears in two of the five splits, so the average involves two scores (the min and max are also shown); the average over all the per-class scores is also shown. Class-average retrieval scores vs. image counts (bottom): most classes score low, but the ones that are higher tend to have more images. Correlation values are Pearson coefficients; in both cases, the p-value is $< 10^{-22}$.

query set and the gallery. The model is trained using the triplet loss to produce an embedding space where images of the same class (in the training set) are closer together than images of different classes. At test time, each image in the test set is treated as a query and the rest of the test set is used as the gallery and ranked according to euclidean distance between the embeddings.

As with the recognition experiments, we train one model each on five different splits of the data. In this setting, each of the castle instances end up in the test set for two of the five splits. When reporting results, we report the average performance for a given instance based on those two test splits.

Retrieval results are shown in Figure 8b. We report Mean Average Precision at R (MAP@R), as suggested in [23]. MAP@R computes the Mean Average Precision over the set of R nearest neighbors, where R is the number of true matches in the gallery for a given query image. We report the average MAP@R for each class individually, as well as the average over all the classes.

In our test sets, each class has between 2 to 400 images, and the total size of the set is over 300,000, making for a large gallery and a challenging retrieval problem. This is reflected in the results shown in Figure 8b, where the reported average per-class MAP@R is just under 0.1. The distribution shows us that there are some classes which on average have MAP@R over 0.5, but more than 60% of the classes have MAP@R$<$ 0.1, leaving a great deal of room for improvement.

Details We use a linear layer directly after global average pooling to reduce the image embedding dimension from 2048 to 128. Models were trained using data-parallel training across 8 GPUs, with a total batch size of $B = 768$. For each batch, we sample 4 images from $B/4$ unique classes. We train using the standard triplet loss and the online mining strategy of [38], as implemented in the `pytorch metric learning` library [24].

4.4. Blind Castle Localization

An important and interesting task, given the known geographic locations of the castles, is trying to geolocate a castle, simply by visual similarity to a known set of castles.

In the original im2gps paper [8], the process used to localize a held-out query image relative to a large database of geotagged training images involved finding the $k = 120$ most visually similar database images. From these k-NN (k nearest neighbors), mean-shift clustering was performed (using a bandwidth of 500km) on the corresponding 120

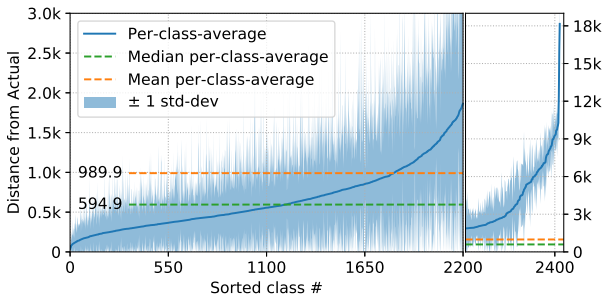


Figure 9: **Blind Castle Localization (Geolocation)**. The location of held-out query images is estimated based on visual similarity relative to the geotagged set of training images. The median localization error across all of the castles is only 600.3km.

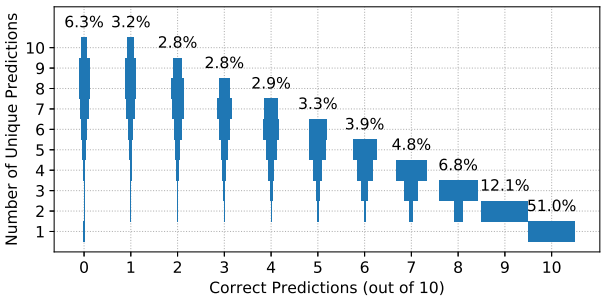


Figure 10: **Image Difficulty**. The number of correct and unique predictions for each image, obtained via cross-validation, are used as a proxy for its "difficulty". Over half of all images are consistently classified correctly, but a significant number have few or no correct predictions (perhaps because they are irrelevant to castle exteriors).

geolocations, retaining the top 6-12 location clusters (dropping any cluster with less than 4 locations). The mode of the maximum-cardinality cluster was used as the estimated location.

Our locations, in turn, are discrete castles. If multiple of the visual nearest neighbors are images of the same castle, then we will have the same location. Because of this, instead of mean-shift clustering on a smaller number of unique locations, we rank the castles in the nearest neighbor set in decreasing frequency and retain the smallest set of castles that has an aggregate frequency of 20 or more. We then calculate the mean location weighted by the respective frequency of each nearest neighbor castle (akin to the center of mass). Using Vincenty's formula for geodesic distance, we calculate the distance from a castle's ground-truth location to the estimated location computed using the weighted mean described above.

For each castle, the mean across the estimated locations is computed. Figure 9 plots this mean distance (measured in kilometers) for each castle and shows them sorted by increasing distance (decreasing precision of the estimated location). Also shown are +/- one standard deviation in the estimated distance for images from each respective castle. The average per-castle error is only 989.9km. A substantial majority of castles are below this, however a small number of castles have significant errors; the median error is only 594.9km, indicated with a green dashed line in the figure.

4.5. Analysis of Difficult Images

We use cross-validation to estimate the difficulty of individual images. We divide the data into six folds, and train on each subset of three folds to get predictions on the other three. This requires us to train $\binom{6}{3} = 20$ models, and we get $\binom{5}{2} = 10$ predictions for each image. We train a Resnet-34 with an extended classifier (Table 2) for ten epochs on the three training folds and then extract predictions for each image in the remaining folds.

We assess image difficulty by the number of incorrect validation predictions made across fold-groups for each image: an "easy" image might have all ten predictions for the correct class, while a more "difficult" image would have additional unique predictions for other (incorrect) classes. However, we note that there is a distinction between images with few correct predictions *and* many unique predictions, and those that have few correct predictions but also few unique predictions. The former set may include images which are not relevant to the exterior of any castle (*e.g.* interior images, shots from castles) that slipped past our rejection threshold, while the latter set may correspond to images that are similar to those of an incorrect class because of angle, lighting, or semantic similarity. In Figure 10 we show the marginal distribution of unique predictions for each image, grouped by number of correct predictions.

5. Conclusion

We have presented the Castles dataset, a large-scale fine-grained instance recognition dataset consisting of over 770k images of more than 2,400 castles. Our experiments and analysis focused on classification, image retrieval, and geolocalization; however, we anticipate that the data can be used in a diverse set of applications, including local descriptor learning, correspondence and matching, multi-view reconstruction, and generative modeling.

Acknowledgments

This work was supported by the National Science Foundation under Grant No. IIS1651832. We gratefully acknowledge the support of NVIDIA Corporation for their donation of multiple GPUs that were used in this research.

References

[1] Y. Bai, Y. Chen, W. Yu, L. Wang, and W. Zhang. Products-10k: A large-scale product recognition dataset. *arXiv preprint arXiv:2008.10545*, 2020.

[2] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

[3] J. P. Crall, C. V. Stewart, T. Y. Berger-Wolf, D. I. Rubenstein, and S. R. Sundaresan. HotSpotter - Patterned species instance recognition. In *WACV*, 2013.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[5] W. Falcon. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3, 2019.

[6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012.

[7] S. Guo, W. Huang, X. Zhang, P. Srikhanta, Y. Cui, Y. Li, H. Adam, M. R. Scott, and S. Belongie. The imaterialist fashion attribute dataset. In *ICCV Workshops*, 2019.

[8] J. Hays and A. A. Efros. IM2GPS: estimating geographic information from a single image. In *CVPR*. 2008.

[9] J. Hays and A. A. Efros. Large-Scale Image Geolocalization. In J. Choi and G. Friedland, editors, *Multimodal Location Estimation of Videos and Images*, pages 41–62. Springer International Publishing, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[11] D. Held, S. Thrun, and S. Savarese. Deep Learning for Single-View Instance Recognition. In *ICRA*, 2016.

[12] M. Izbicki, E. E. Papalexakis, and V. J. Tsotras. Exploiting the Earth's Spherical Geometry to Geolocate Images. In *ECML PKDD*, 2019.

[13] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless. Geolocating Static Cameras. In *ICCV*, 2007.

[14] E. Kalogerakis, O. Vesselova, J. Hays, A. A. Efros, and A. Hertzmann. Image Sequence Geolocation with Human Travel Priors. In *ICCV*, 2009.

[15] P. Kaur, , K. Sikka, W. Wang, s. Belongie, and A. Divakaran. Foodx-251: A dataset for fine-grained food classification. *arXiv preprint arXiv:1907.06167*, 2019.

[16] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for Fine-Grained Image Categorization. In *CVPR Workshops (FGVC)*, 2011.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *ICCV Workshops (3DRR)*, 2013.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

[20] L. Liu, H. Li, and Y. Dai. Stochastic Attraction-Repulsion Embedding for Large Scale Image Localization. In *ICCV*, 2019.

[21] S. Maji, E. Rahtu, J. Kannala, M. B. Blaschko, and A. Vedaldi. Fine-Grained Visual Classification of Aircraft. *arXiv.org*, 2013.

[22] E. Müller-Budack, K. Pustu-Iren, and R. Ewerth. Geolocation Estimation of Photos using a Hierarchical Model and Scene Classification. In *ECCV*, 2018.

[23] K. Musgrave, S. Belongie, and S.-N. Lim. A metric learning reality check. *arXiv preprint arXiv:2003.08505*, 2020.

[24] K. Musgrave, S. Belongie, and S.-N. Lim. PyTorch Metric Learning. *arXiv preprint arXiv:2008.09164*, 2020.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*. 2019.

[26] P. H. Seo, T. Weyand, J. Sim, and B. Han. CPlaNet: Enhancing Image Geolocalization by Combinatorial Partitioning of Maps. In *ECCV*, 2018.

[27] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *SIGGRAPH*, 2006.

[28] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from Internet photo collections. *IJCV*, 80(2):189–210, 11 2008.

[29] A. Stylianou, H. Xuan, M. Shende, J. Brandt, R. Souvenir, and R. Pless. Hotels-50K: A Global Hotel Recognition Dataset. In *AAAI*, 2019.

[30] M. Sun, Y. Yuan, F. Zhou, and E. Ding. Multi-Attention Multi-Class Constraint for Fine-grained Image Recognition. In *ECCV*, 2018.

[31] K. C. Tan, Y. Liu, B. Ambrose, M. Tulig, and S. Belongie. The herbarium challenge 2019 dataset. *arXiv preprint arXiv:1906.05372*, 2019.

[32] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a Bird Recognition App and Large Scale Dataset With Citizen Scientists: The Fine Print in Fine-Grained Dataset Collection. In *CVPR*, 2015.

[33] G. Van Horn, S. Branson, S. Loarie, S. Belongie, and P. Perona. Lean Multiclass Crowdsourcing. In *CVPR*, 2018.

[34] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. The INaturalist Species Classification and Detection Dataset. In *CVPR*, 2018.

[35] N. Vo, N. Jacobs, and J. Hays. Revisiting IM2GPS in the Deep Learning Era. In *ICCV*, 2017.

[36] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011.

[37] S. Wang, M. Bai, G. Mattyus, H. Chu, W. Luo, B. Yang, J. Liang, J. Cheverie, S. Fidler, and R. Urtasun. TorontoCity: Seeing the World with a Million Eyes. In *ICCV*, 2017.

[38] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, 2019.

[39] T. Weyand, A. Araujo, B. Cao, and J. Sim. Google Landmarks Dataset v2-A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In *CVPR*, 2020.

[40] T. Weyand, I. Kostrikov, and J. Philbin. PlaNet - Photo Geolocation with Convolutional Neural Networks. In *ECCV*, 2016.

[41] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva. SUN Database: Exploring a Large Collection of Scene Categories. *IJCV*, 2014.

[42] J. Xiao, J. Hays, K. A. E. Aude, and O. A. Torralba. SUN Database: Large-scale Scene Recognition from Abbey to Zoo. In *CVPR*, 2010.

[43] L. Yang, P. Luo, C. C. Loy, and X. Tang. A Large-Scale Car Dataset for Fine-Grained Categorization and Verification. In *CVPR*, 2015.

[44] C. Zhang, C. Kaeser-Chen, G. Vesom, J. Choi, M. Kessler, and S. Belongie. The imet collection 2019 challenge dataset. *arXiv preprint arXiv:1906.00901*, 2019.

[45] Y.-T. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven. Tour the world: building a web-scale landmark recognition engine. In *CVPR*, pages 1085–1092. IEEE, 2009.

[46] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million Image Database for Scene Recognition. *PAMI*, 2017.