Google-Landmark Recognition with Deep Learning

Chien-Yi Chang
Stanford University

Abstract

Our problem is a 6,151 class landmark classification problem on a very large-scale dataset, Google-Landmarks. This dataset has been recently released to public by Google featuring millions of images on thousands of distinct landmarks captured at various locations and uploaded on-line by users over the world. This dataset, in spite of its extensive number of landmarks, is extremely unbalanced, posing a key problem in developing powerful models. In this paper we show how to use transfer learning along with data augmentation to obtain a model giving Top-5 accuracy of 82.03% on a derivative version of the original Google-Landmarks dataset, which contains images from 6,151 unique landmarks. Additionally, we explore how a Generative Adversarial Network can be used to alleviate the problems posed by an extremely unbalanced dataset.

1. Introduction

Object recognition is one of the fundamental problems widely studied in computer vision. The problem that we will be investigating falls into this very category: we want to recognize landmarks (e.g. White House, Great Wall of China etc.) present in an image directly from the image pixels. Landmark recognition is interesting because it can help people better understand and organize their digital photo collections. Despite being a straightforward objective, i.e. identifying landmark presented in a given image, the task itself is challenging especially as we increase the number of distinct landmarks.

Our proposed landmark recognition problem can be best described as an instance-level recognition problem. It differs from categorical recognition problem in that instead of recognizing general entities such as mountains and buildings, it requires fine-grained learning algorithms that can identify Mount Everest/Mount Whitney, or White House/Big Ben. Landmark recognition also differs from what we have seen in the ImageNet classification challenge. For example, since landmarks are generally rigid, immobilized, one-of-the-kind objects, the intra-class variation is very small (in other words, a landmark's appearance does not change that much across different images of it). As a result, variations only arise due to image capture conditions, such as foreground/background clutter, occlusions, different viewpoints, weather and illumination, making this distinct from other image recognition datasets where images of a particular class (such as a cat) can vary much more in shape and appearance. These characteristics are also shared with other instance-level recognition problems, such as artwork recognition — ideally with mild modification, our solution to landmark recognition problem can be applied to research for other image recognition problems as well. In this project, we applied transfer learning with Inception-v3 [10], [9] network to obtain 82.03% top-5 accuracy for a 6151 class landmark recognition problem. We also explored the application of Generative Adversarial Networks (GANs) for data augmentation to alleviate the class imbalance problem in our dataset.

2. Related Work

One of the major challenge in building a practical and universal landmark recognition system is to acquire a large labeled dataset with diverse landmarks. Previous research have been focused on relatively small number of distinct landmarks. Philbin *et al*. [8], for example, present a dataset containing 6,412 images on 12 distinct landmarks in Paris. Li *et al*. [6] study the landmark recognition problem using a dataset with one million labeled images on 500 landmarks. In contrast we work with a much larger Google-Landmarks dataset [7]. As mentioned before, our approach for this problem is to utilize transfer learning and use CNN based state-of-the-art neural network models. For this purpose, we first examined state of the art literature on image classifiers (e.g. Inception net [10, 9], ResNet [4] etc.), specifically geared towards tackling very large sets of classes. We decided to go with Inceptionv3 because of it's compact nature and it's performance (3.46% top-5 error rate) on the ImageNet [5] (has around 1000 classes) dataset.

3. Dataset

Data Exploration

Since the original Google-Landmarks (G-L) dataset is given as lists of URLs of each image, we first crawled the

Internet and managed to download 99.4% images from the G-L dataset (a small fraction of links were broken at the time of our acquisition). Images are in various dimensions. The total storage space needed for the original G-L dataset is approximately 500GB, which exceeds the space we have on our computing platform. We thus resized all downloaded images to 128x128 pixels with anti-aliasing.

Before moving to build a specific model, we first performed thorough data explorations on the labeled images to study the overall structure of G-L dataset. Despite its large size, the class distribution within G-L dataset is very unbalanced. The distribution and associated CDF of class size can be seen in Fig. 1 In G-L dataset, each class contains only 82 images on average, while the most frequent class contains more than 50,000 images and the least frequent classes (about 1.06% of all classes) contain only one image. The CDF also shows that 60% of classes contain less than 20 labeled images. Fig. 2 shows the top 20 most frequent landmarks presented in the dataset and eight randomly sampled images in the most frequent landmark class. These images, despite being labeled as to contain the same landmark, show wild variations including foreground/background clutter, occlusion, variations in viewpoint and illumination.

Data Manipulations

In order to make the dataset more tractable for our recognition model, we focused on the 6,151 classes that have at least 20 labeled images. We call this dataset Landmarks-230, which is the dataset we focus on, in the rest of this paper to distinguish it from the original G-L dataset. Since the 115k query images in Landmarks-230 are unlabeled, we cannot readily use these images for training or testing in the recognition task without acquiring more information, such as their corresponding GPS locations. As a result, we decided to prepare our train/val/test data using only the labeled images from Landmarks-230. The validation and test sets were made to have 2 images from each class (in total 12,302 images in each set), and the rest were given to the training set. Instead of working with the entire 1M image training dataset we decided to sample Landmarks-230 training data and get a smaller but perfectly balanced dataset, which we call Landmarks-S 1. Landmarks-S has 6,151 classes and 123,020 labeled images in total that includes, a training dataset where each class has exactly 16 labeled images for training (randomly picked from the available images in that class), and the original validation and test sets (each one with 12,302 images). We used this dataset for our hyperparameter search as discussed in the results section.

We face a problem when we try to have more than 16 training images per class (i.e., increase the training data) so as to have better predictive models. For some of the classes we have many more images available for training, than the 16 images we used in Landmarks-S, in the Landmarks-230 dataset, but for many classes there are none available. So, to increase the number of training images per class and to simultaneously keep the dataset balanced we decided to use data augmentation (more details in next section) for the classes not having sufficient training images. We ended up developing another dataset, namely, Landmarks-M, having 6,151 classes and 332,154 labeled images in total that includes, a training dataset where each class has exactly 50 labeled images for training (randomly picked from the available images in that class or if none available then using data augmentation on the available training examples in that class), and the original validation and test sets (each one with 12,302 images) from Landmarks-230. We used Landmarks-M for training our best performing model.

In this project, we also try to study how to deal with extreme imbalance presented in G-L. Due to limited time and budget, we decided to assemble a much smaller yet very unbalanced dataset for our experiments. We first randomly sample 1,000 classes from Landmarks-S, keep 16 labeled images as training data and 2 labeled images as test per class, and mark these classes as majority classes. Then we randomly sample (without replacement) another 1,000 classes from Landmarks-S, keep 2 labeled images as training and 2 as test per class, and mark these classes as minority classes. We call this dataset Landmarks-U, where U stands for unbalanced. Landmarks-U has 2,000 unique classes while each class contains only 9 images on average (excluding test images). Landmarks-U is unbalanced just like G-L as 50% of it classes contain less than an average number of images.

Dataset	Classes	Labeled Images	Balanced?
G-L	14,951	1,225,029	No
Landmarks-230	6,151	1,157,977	No
Landmarks-S	6,151	123,020	Yes
Landmarks-M	6,151	332,154	Yes
Landmarks-U	2,000	22,000	No

Table 1: Comparison of derivative datasets and the original Google-Landmarks (G-L).

4. Methods

Our main evaluation metric will be top-1 accuracy for this classification task, we subsequently use this metric's value on the validation set to guide our model selection.

Baseline

Our main approach is to use CNN based Deep Neural Nets for the "brains" of the classifier. As a starting point we re-implemented a CNN architecture which has been shown to achieve 72.6% top-1 accuracy for CIFAR-10 [2] classification. The baseline classifier has a total of 14 layers.

It consists of 3 convolutional blocks and 3 2x2 average pooling layers followed by 1 fully-connected layer with 0.5 dropout and finally 1 fully-connected layer with softmax activation. Each convolutional block has 2 convolutional layers with ReLU activation, and 1 BN layer.

Transfer Learning

We used transfer learning for our main model, as the core problem we are tackling is similar to the other many-class classification problems (e.g., ImageNet challenge) which people have extensively worked on, and transfer learning gives us a perfect way to leverage that effort. In addition the G-L dataset is extremely unbalanced, 60% of the classes have less than 20 images, thus training a model from scratch on such a dataset is extremely challenging and may not lead to good models. For these reasons, we went forward with using Transfer learning for our problem (the large gap in performance 2 we observed between the baseline and our main model adds weight to this assertion). As discussed before, for applying transfer learning we first examined state of the art literature on image classifiers (e.g. Inception net [10, 9], ResNet [4] etc.), specifically geared towards tackling very large sets of classes. We decided to go with Inceptionv3 because of it's compact nature and it's performance on the ImageNet dataset (3.46% top-5 error rate). For starter code we used the code provided in the Tensorflow tutorial [1] and modified it to suit our data processing pipeline and also modified the network architecture for better performance on our dataset.

Data Augmentation

We propose to use data augmentation to alleviate the extreme imbalance presented in the original G-L dataset and eventually improve our classifier performance in solving the recognition task.

Consider a classifying function ψ, where maps vectors in the input space $\mathcal{X}$ to their counterparts in label space $\mathcal{Y}$. Data augmentation utilizes prior knowledge about data and its label. In general if we know the class label will be invariant to some transformation T such that $y = \psi(T(x))$, then we can apply such transformation on existing data $x \in \mathcal{X}$ to obtain additional data $\tilde{x} = T(x)$ within that class. Some transformations come straight from image priors, as we know the label for given image would be invariant up to various transformations including rotations, horizontal/vertical translations, horizontal/vertical flips as well as zooming/resizing. We apply this technique to our classification model.

Besides standard data augmentation with pre-defined transformation, we argue that we can use a generative neural network (GAN) trained on majority classes (classes with ample samples) to learn the transformation T and thus generate more images for the minority classes. The idea is the variation within each class in the landmark dataset should bear similarity across different classes, and therefore the variation learned within majority class can be transferred to minority class. For example, class 'White House' contains many images of White House from various viewing angles with different levels of occlusion. We can use a generative adversarial network to capture this variation presented in the 'White House' class and then use it to generate more samples for other class, such as 'Parliament Building'.

Our GAN consists of two models: A discriminator D that estimates the probability of a given sample coming from the real dataset. It works as a critic and is optimized to tell the generated samples from the real ones. A generator G that produces synthetic samples given a condition (in our case, a sample from the real dataset) and a random variable z (Gaussian noise) which brings in potential diversity. It is trained to capture the real data distribution so that its generative samples can be as real as possible, or in other words, can trick the discriminator to offer a high probability.

The generator G has 8 [Conv-Leaky ReLu-BN]*4 blocks, where each block is followed by 2x2 max pooling operation with stride 2 for downsampling. The number of filters for each block alternates between 64 and 128 for every two blocks, i.e. the first two blocks have 64 filters, they are followed by two blocks with 128 filters. We apply dropout to the last convolution layer within each block to improve the overall performance. Similar to what have been used in the ResNet[4], we add skip connections between each block in the generator by using 1x1 convolution to pass gradients directly between different blocks without applying non-linearity.

The discriminator D consists of 5 [Conv-ReLu-LN]*3 blocks and 5 [Conv(1x1)-AVGPool(2x2)] pooling blocks. Each convolution layer has 16 filters. To further improve the information flow between layers, we a similar connectivity pattern as proposed in cite, which introduces direct connections from any layer to all subsequent layers. The use of 1x1 Convolution in pooling blocks is to reduce the number of input feature maps and together with average pooling to increase computational efficiency.

Consider a class $\mathcal{C}$ where $x_1, x_2, ...$ are samples labeled as class $\mathcal{C}$. The generator takes in concatenated random noise z and conditional real image x_i, and outputs a fake image $\tilde{x}_i$. Then the discriminator need to distinguish between the real distribution $f(x_i, x_j)$ from the fake distribution $f(x_i, \tilde{x}_i)$

$$\tilde{x}_i = G(z, x_i)$$
$$D(f(x_i, \tilde{x}_i), f(x_i, x_j))$$

To achieve stable training of and minimize effort in hyperparameter tuning, we adopt the Wasserstein loss and clipping weights as proposed in improved WGAN. [3]

5. Experiments/Results/Discussion

Baseline

We first train the baseline classifier on Landmarks-S dataset. We train the baseline model from scratch for 50 epochs, a set of learning rates ranging from 0.0001 to 0.001, and an Adam optmizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We use validation set to select best learning rate at 0.001. When evaluated on the validation set, our baseline model yields a top-1 accuracy at 0.0479% and top-5 accuracy at 0.193%. A random guesser on 6,151 classes would have top-1 accuracy at $\frac{1}{6151} = 0.0163\%$ and top-5 accuracy at $1 - (1 - \frac{1}{6151})^5 = 0.0813\%$. Our baseline model shows only slight improvement over random in recognition task.

As mentioned in earlier section, our baseline model has been shown to be a success on CIFAR-10 [2] classification task with 72.6% top-1 accuracy. However, it fails to provide satisfying performance on the Landmarks-S dataset despite considerable efforts in hyperparameter tuning. Landmarks-S contains 615x more classes than CIFAR-10. The number of sample images per class is only $\frac{16}{5000} = 0.0032$ of CIFAR-10. In order to achieve better performance, we need to increase the complexity of our model and also address the issue of data scarcity. We do this by leveraging transfer learning and data augmentation.

Standard Data Augmentation

We enrich Landmarks-S with standard data augmentation whenever necessary: random horizontal and vertical shifts within 20% of original size; random rotations of -20 degrees to 20 degrees; random horizontal flip; random shears with 0 to 20 degree angle in counter-clockwise direction; random zoom ranging from -20% to 20%. We use these data augmentation techniques to arrive at the Landmarks-M dataset as discussed in the previous section. Our best performing model was trained on this dataset. Additionally, we explore how using GANs for data augmentation can also bolster a model's performance.

Data Augmentation with GAN

Without loss of generality, we train the GAN on a very unbalanced dataset, Landmarks-U. 1 Just like G-L dataset, Landmarks-U is very unbalanced. We hope that by showing how data augmentation techniques can improve the classifier's performance on a smaller, unbalanced dataset, we can get some insights towards how to handle a very unbalanced, yet much larger dataset like G-L. Note that in the training of our GAN, we only use the 1,000 classes with 16 real images per class as training examples (The 1,000 majority classes are split into 90% and 10% train/val). The 1,000 minority classes remains unseen during the GAN training process.

We train the GAN using RMSProp optimizer for 50 epochs, a learning rate of 0.0001, a clipping parameter of 0.01, a batch size of 64, and 5 iterations of critic per generator iteration. Examples of the generated landmark images are shown in Fig. 3.

Qualitatively we can see that GANs are able to learn intra-class variations and perform non-trivial transformations on the input data from minority classes. By using GANs, we are able to obtain more "interesting" augmented images compared to images obtained by simply cropping or scaling the input image. As an example in Figure 3, row 2, people are removed from the foreground of the picture, more illustrations are given in the caption for Figure 3.

Quantitatively we evaluate the "goodness" of our generated images by testing how those generated images can help our baseline classifier perform better. We train our baseline classifier on Landmarks-U for 50 epochs, a learning rate of 0.0005, and an Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The top-1 accuracy on the test set is 0.03588. We then use the GAN that has been trained on 1,000 majority classes to generate images for 1,000 minority classes. We augment Landmarks-U with these GAN-generated images such that each class has exactly 16 images for training (no augmentation for majority classes). Again we train the baseline classifier on the GAN augmented Landmark-U dataset for 50 epochs, a learning rate of 0.001, and an Adam optimizer with default setting. The top-1 accuracy of the baseline model trained on augmented Landmarks-U is 0.12874, which gives us about 260% improvement over the model without data augmentation. This shows that data augmentation by using this approach can provide significant improvement in cases where the dataset is very unbalanced.

Model

Our best performing model was obtained by applying transfer learning, specifically using the InceptionV3 network by removing it's final fully-connected layers and adding our own fully connected layer followed by a softmax layer. In total our model has 2049*6151 = 12,603,399 learnable parameters. As we discuss in the results section, we experimented with adding more fully-connected layers and varying degrees of dropout before the softmax layer, but these changes were ineffective to reduce the validation loss.

Results

For our hyperparameter search we focused on the smaller Landmarks-S dataset. Here we describe the various hyperparameters we tuned to obtain the best possible accuracy on the validation set within 3 epochs (2400 update steps) of the Landmarks-S dataset. We first experimented with the Adam and Gradient descent optimizers, and we found that Adam significantly outperforms the GD optimizer for the same learning rate in terms of validation accuracy 4. We then experimented with various learning rates and found that 0.0007 works very well for our task, some of the learn-

ing rates we experimented with are shown in 5. Finally we experimented with various batch sizes to find a good tradeoff between good updates and time per update, and we found that a batch size of 512 works well 6 with quick updates compared to 1024 which we found to have very slow updates on our machine. With the chosen hyperparameters we trained our model for 6000 iterations and were able to obtain a validation top-1 accuracy of 61.71%, while we obtained near 100% accuracy on the training data 7.

From the accuracy/loss curves we observe that as the training accuracy increases the validation accuracy increases and as the training accuracy tapers off the validation accuracy does the same. This tells us that getting more training data should definitely help in bridging the gap, in retrospect this gap was inevitable as we are only using 16 images per class for training. So, to improve the model further we use the Landmarks-M dataset (details given before). Even with this larger dataset we observed that our model with only one fully-connected layer added to the Inception-v3 network fits the training data with near 100% accuracy, so we found no reason to try adding more fully-connected layers or using bigger models (i.e., increasing the number of learnable parameters of the model). To see whether we can reduce the gap between the training and validation accuracies and improve the final validation accuracy we added a dropout layer before the final fully-connected layer of our model, and experimented with different rates as shown in 8. From the results we conclude that no dropout gives the best validation accuracy within 3000 update steps, so we didn't include a dropout layer in our final model. With the chosen hyperparameters we trained our final model for 12000 iterations on the Landmarks-M dataset and were able to obtain a validation top-1 accuracy of 69.67%, while we were able to obtain near 100% accuracy on the training data 9. Table 2 compares the different models we have discussed so far, and Table 3 gives several useful metric values of our best performing model on the test set. We observe that we were able to get a Top-5 accuracy of 82.03% which is a decent result for a 6000 class classification problem.

Model	Top-1 Accuracy
Random guesser	0.0163%
CNN baseline	0.0479%
Inception-v3 with Landmarks-S	61.71%
Inception-v3 with Landmarks-M	69.67%

Table 2: Validation accuracy of different models.

Discussion

Now let's analyze and visualize the predictions and mispredictions made by our final model on the test set. First let's look at some cases where our model mis predicted the class label in Figure 10. Each subfigure in Figure 10 corresponds to a commonly seen cause of misprediction in our test set, the left image in the subfigure is a representative image for that class and on the right is the mispredicted test image from that class. We observed several causes for mispredictions such as Occlusion, deformation, bad illumination, viewpoints, and foreground clutter covering the landmark. We can reduce these mispredictions by adding more data having these characteristics in the training set, thus indicating that we can further improve our model.

Metric	Value
Cross-entropy loss	1.7086
Top-1 Accuracy	68.75%
Top-5 Accuracy	82.03%
Top-10 Accuracy	86.23%
Avg. Recall	68.75%
Avg. Precision	72.41%
Avg. F1 Score	67.49%

Table 3: Useful metric values on test set for our final model.

We also produce saliency maps for test images from some classes to see whether the network is able to "learn" and distinguish the landmarks from extraneous information. The saliency images along with the actual test images are given in Figure 11. In each subfigure we provide two test images, belonging to the same class, and their saliency maps obtained from our final model. We provide two test images of the same class to illustrate how the saliency image changes when the landmark moves around, for e.g. in Figure 11(a) we see that when the landmark is in the top-left corner of the image the saliency image is mostly highlighted in that region, it is interesting to note that the couple in the picture are mostly ignored in the saliency image.

Additionally we observed that our data augmentation with GAN improved the top-1 accuracy of baseline model by about 260%. Qualitatively we have found that GAN trained on majority classes is able to learn intra-class variations, such as image re-focusing, elements removal, elements addition, viewing-angle variation, and foreground/background mixing, and apply them to unseen minority classes, thus giving us a way to generate "richer" images for augmentation.

6. Conclusions

In this project, we tackled the problem of Landmark recognition using transfer learning and data augmentation. We obtained a model giving Top-5 accuracy of 82.03% on a 6,151-class recognition task. Additionally, we showed how a Generative Adversarial Network (GAN) can be used to improve a simple classifier's performance by 260% in an extremely unbalanced dataset setting. We hope that people can use similar GAN architectures to address unbalanced data setting in other domains.

References

[1] Retrain an Image Classifier for New Categories. https://www.tensorflow.org/tutorials/image_retraining.

[2] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[3] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Y. Li, D. J. Crandall, and D. P. Huttenlocher. Landmark classification in large-scale image collections. In *Computer vision, 2009 IEEE 12th international conference on*, pages 1957–1964. IEEE, 2009.

[7] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3456–3465, 2017.

[8] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[9] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

Appendices

Contribution

This is a single person project.

Figures

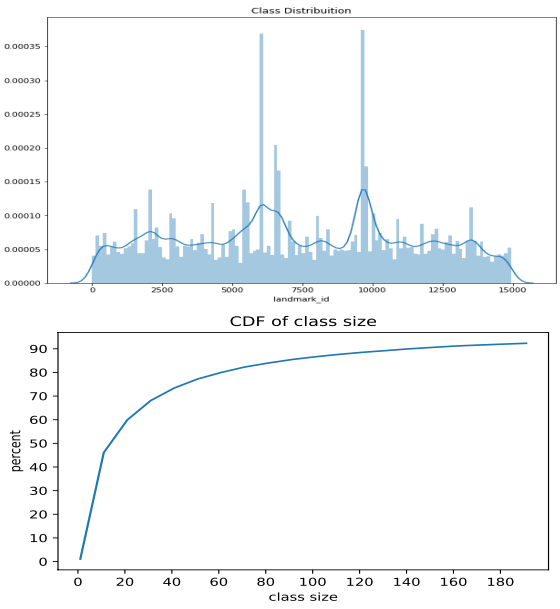


Figure 1: The class distribution and CDF of Google-Landmarks dataset. Mean: 82. Std: 707

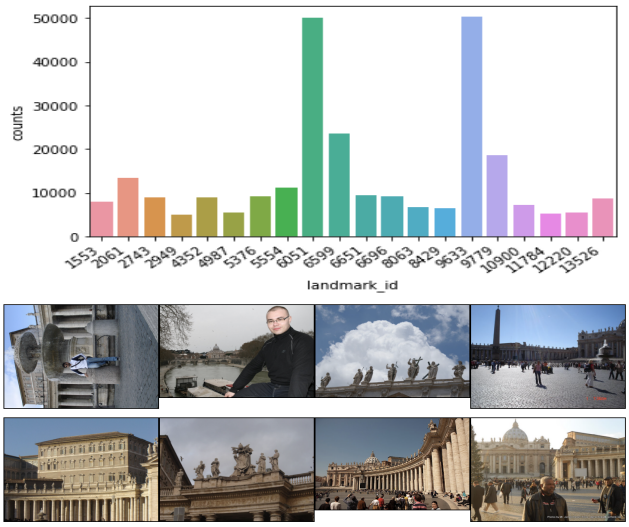


Figure 2: Top 20 most frequent landmarks presented in the Landmarks-230 dataset. We also show eight out of total 50,337 images from the most frequent landmark class, i.e. landmark-id 9633.

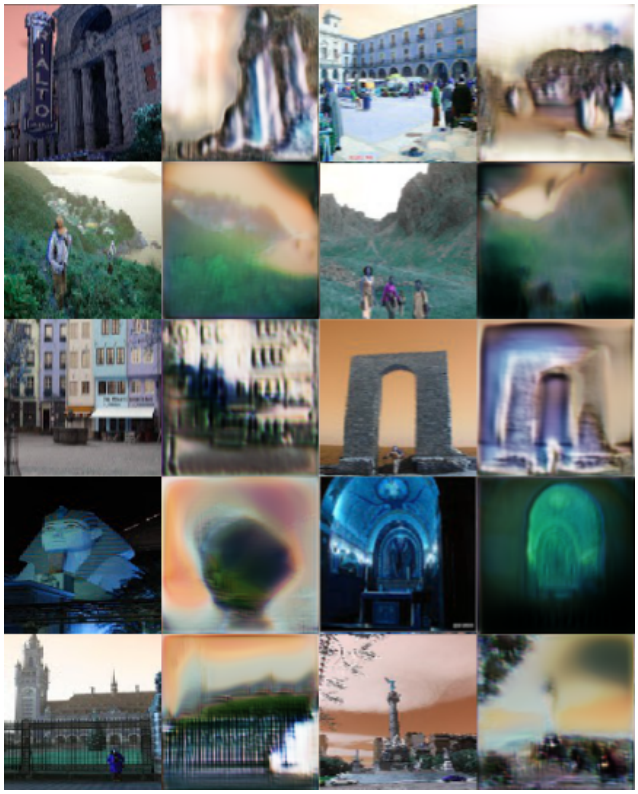


Figure 3: Examples of generated landmark images. Column 1 and 3 contains real images from 10 test classes, column 2 and 4 contain corresponding GAN generated images. Row 1: generated images are re-focused. Row 2: people in the foreground are removed. Row 3: shadow and duplicate arch are added into generated images. Row 4: left: generated image shows top viewing angle of Sphinx. right: flipped viewing angle. Row 5: left: foreground remains upright while background is sheared to the right. right: synthetic leaves are added in the foreground while the tower in the background is removed.

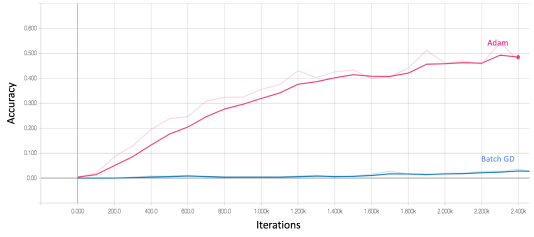


Figure 4: Adam vs Gradient Optimizer with 128 Batch updates

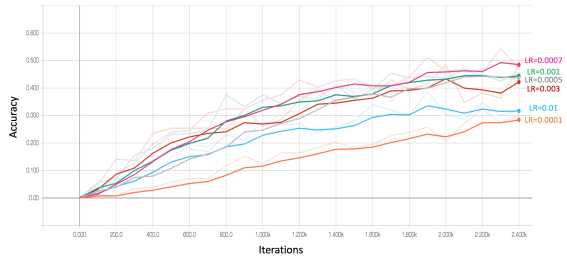


Figure 5: Learning rate tuning; Validation accuracy vs #updates

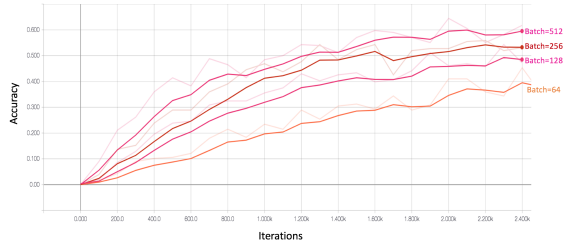


Figure 6: Batch size tuning; Validation accuracy vs #updates

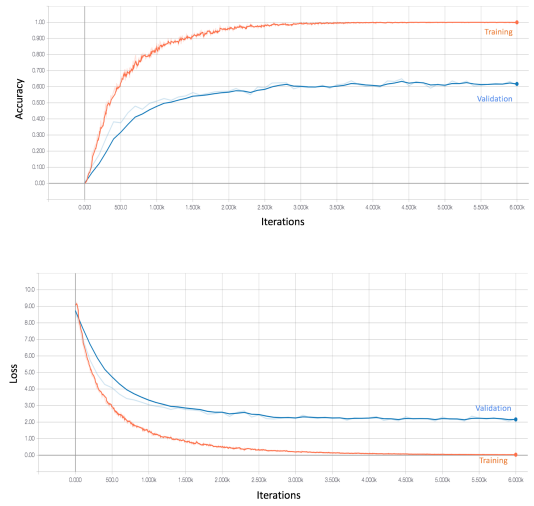


Figure 7: Accuracy and Loss vs #iterations on the Landmarks-S dataset using tuned hyperparameters

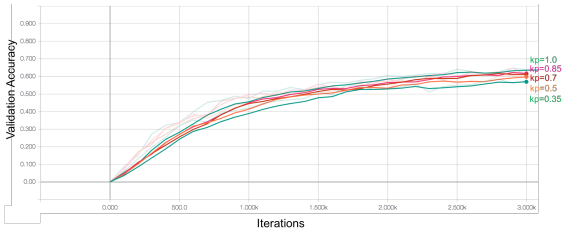


Figure 8: Dropout keep probability tuning; Validation accuracy vs #updates

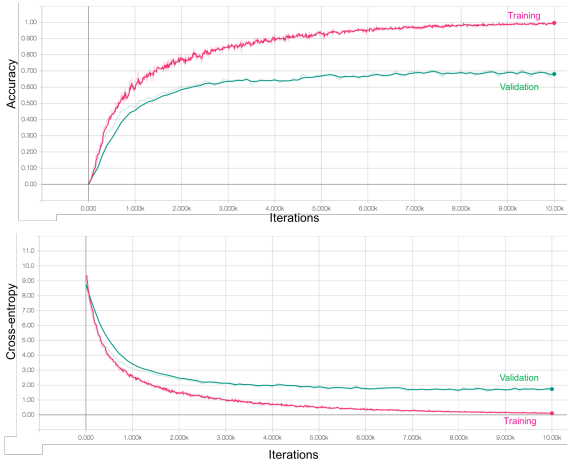


Figure 9: Accuracy and Loss vs #iterations on the Landmarks-M dataset using tuned hyperparameters

(a) Misprediction due to Occlusion

(b) Misprediction due to bad Illumination

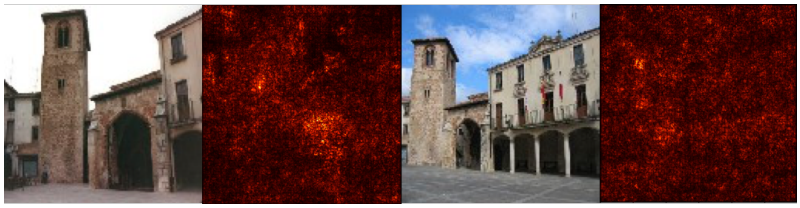
(c) Misprediction due to deformed test image

(d) Misprediction due to bad viewpoint

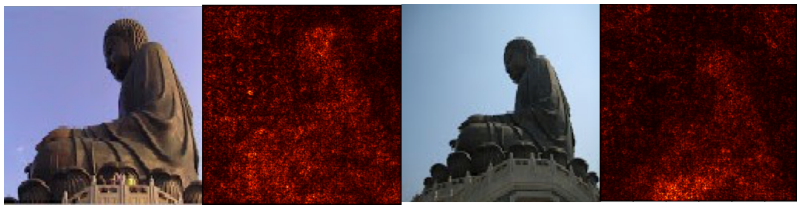
(e) Misprediction due to foreground clutter

Figure 10: Examples of test set mispredictions

(a) Class-10744



(b) Class-10100



(c) Class-5422

Figure 11: Saliency images for some test set examples