The MvMF Loss for Predicting Locations on the Earth's Surface

Mike Izbicki, Evangelos E. Papalexakis, and Vassilis J. Tsotras

UC Riverside, Riverside, CA 92507, USA
mike@izbicki.me, epapalex@cs.ucr.edu, tsotras@cs.ucr.edu

Abstract. Existing methods for predicting locations on the earth's surface use standard classification or nearest neighbor techniques. These methods have poor theoretical properties because they do not take advantage of the earth's spherical geometry. In some cases, they require training data sets that grow exponentially with the number of feature dimensions. This paper introduces the *Mixture of von-Mises Fisher* (MvMF) loss function, which is specifically designed to exploit the earth's spherical geometry. Theoretical analysis shows that the MvMF loss requires only a dataset of size linear in the number of feature dimensions, and empirical results show that it outperforms previous methods with orders of magnitude less training data and computation. As a motivating example, we focus on the problem of geolocating ground level images, but we emphasize that the MvMF loss is equally suitable for working with satellite image sources. This workshop paper is a short version of [1].

Keywords: Geolocation; Flickr; Deep Learning; von Mises-Fisher

1 Introduction

Consider the two images below:

Most people recognize that the left image is of the Eiffel Tower, located in Paris, France. A trained expert can further recognize that the right image is a replica of the Eiffel Tower. The expert uses clues in the image's background (e.g. replicas of other famous landmarks, tall cement skyscrapers) to determine that this image

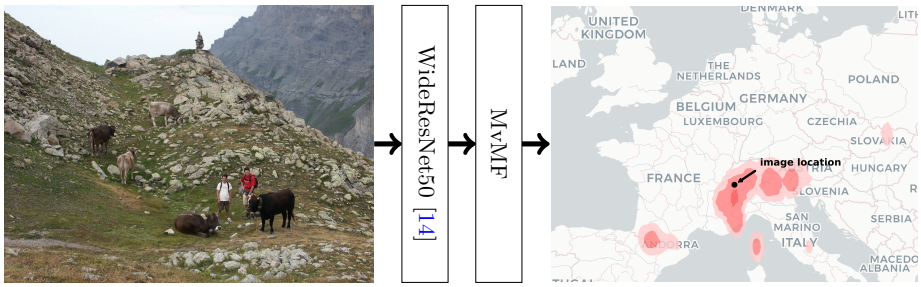

Fig. 1. To geolocate an image, we first generate features using the WideResNet50 [14], then pass these features to our novel *mixture of von Mises-Fisher* (MvMF) output layer. The MvMF outputs a probability distribution over the earth's surface, and is particularly well-suited for visualizing the output of hard-to-geolocate images.

was taken in Shenzhen, China. We call these images *strongly localizable* because the images contain all the information needed to exactly geolocate the images. Existing geolocation algorithms work well on strongly localizable images. These algorithms use deep neural networks to extract features, and can therefore detect the subtle clues needed to differentiate these images.

Most images, however, are only *weakly localizable* because the image does not contain enough information to exactly geolocate it. Consider the image in Figure 1 of two men hiking. An expert can use clues like the geology of the mountains, the breed of cattle, and the people's appearance to determine that this image was taken in the Alps. But the Alps are a large mountain range, and there is simply not enough information in the image to pinpoint exactly where in the Alps the image was taken. Existing geolocation algorithms are overconfident when predicting locations for these images. These algorithms use either nearest neighbor or classification methods to perform geolocation, and these procedures do not take advantage of the earth's spherical geometry. They therefore cannot properly represent the ambiguity of these weakly localizable images.

In this paper, we introduce the MvMF output layer for predicting GPS coordinates with deep neural networks. The MvMF has three advantages compared to previous methods:

1. The MvMF takes advantage of the earth's spherical geometry and so works with both strongly and weakly localizable images.
2. The MvMF has theoretical guarantees, whereas no previous method has a theoretical analysis.
3. The MvMF interpolates between the nearest neighbor and classification approaches to geolocation, retaining the benefits of both with the drawbacks of neither.

In our experiments, we use the WideResNet50 [14] convolutional neural network to generate features from images, but we emphasize that any deep neural network serve as input to an MvMF layer. An extended version of this paper [1] provides a more detailed comparison to prior work and details of the theoretical analysis.

2 Geolocation via the MvMF

The MvMF is the first neural network loss function designed for predicting GPS locations on the earth's surface. In this section, we first introduce the MvMF as a probabilistic model, then describe two alternative interpretations of the MvMF as a classification model with a non-standard loss or as a nearest neighbor model using non-standard features. A powerful property of the MvMF model is that it can interpolate between the classification and nearest neighbor approaches to geolocation, getting the best of both techniques while avoiding the limitations of both.

2.1 The probabilistic interpretation

This subsection formally introduces the MvMF output layer as a mixture of von Mises-Fisher distributions. Then we describe the training and inference procedures.

The *von Mises-Fisher* (vMF) distribution is one of the standard distributions in the field of directional statistics, which is the study of distributions on spheres. The vMF can be considered the spherical analogue of the Gaussian distribution [e.g. 10] and enjoys many of the Gaussian's nice properties. Thus, the *mixture of vMF* (MvMF) distribution can be seen as the spherical analogue of the commonly used Gaussian mixture model (GMM). While the MvMF model has previously been combined with deep learning for clustering [4] and facial recognition [5], we are the first to combine the MvMF and deep learning to predict GPS coordinates.

Formally, the vMF distribution is parameterized by the *mean direction* $\mu \in \mathbb{S}^2$, and the *concentration parameter* $\kappa \in \mathbb{R}^+$. The density is defined for all points $\mathbf{y} \in \mathbb{S}^2$ as

$$\text{vMF}(\mathbf{y}; \mu, \kappa) = \frac{\kappa}{\sinh \kappa} \exp(\kappa_i \mu^\top \mathbf{y}). \tag{1}$$

An important property of the vMF distribution is that it is symmetric about μ for all $\mu \in \mathbb{S}^2$. As shown in Figure 2, a gaussian distribution over GPS coordinates does not account for the earth's spherical geometry, and is therefore not symmetrical when projected onto the sphere.

The *mixture of vMF* (MvMF) distribution is a convex combination of vMF distributions. If the mixture contains c component vMF distributions, then it is parameterized by a collection of mean directions $M = (\mu_1, ..., \mu_c)$, a collection of concentration parameters $K = (\kappa_1, ..., \kappa_c)$, and a vector of mixing weights $\Gamma \in \mathbb{R}^c$ satisfying $\sum_{i=1}^{c} \Gamma_i = 1$. Notice that we use capital Greek letters for the parameters of the mixture distribution and lowercase Greek letters for the parameters of the corresponding component distributions. The density is given by

$$\text{MvMF}(\mathbf{y}; M, K, \Gamma) = \sum_{i=1}^{c} \Gamma_i \, \text{vMF}(\mathbf{y}, M_i, K_i). \tag{2}$$

To construct the MvMF loss function from this density, we assume that the mean direction and concentration parameters do not depend on the input features. The mixing weights are parameterized using the standard softmax function

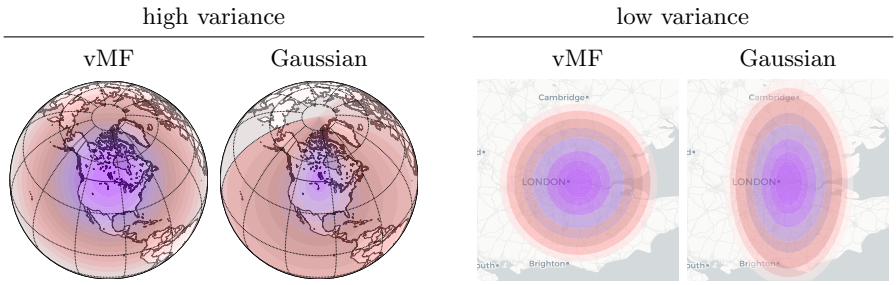

Fig. 2. The vMF distribution takes into account the curvature of the earth's surface, and so contour lines are equidistant from the center at all scales and locations. The Gaussian distribution over GPS coordinates, in contrast, becomes elongated far from the equator, and has discontinuities at the poles and at longitude $\pm 180°$.

as

$$\Gamma_i(\mathbf{x}; W) = \frac{\exp(-\mathbf{x}^\top \mathbf{w}_i)}{\sum_{j=1}^{c} \exp(-\mathbf{x}^\top \mathbf{w}_j)}. \tag{3}$$

where $W = (\mathbf{w}_1, ..., \mathbf{w}_c)$ and each $\mathbf{w}_i \in \mathbb{R}^d$. Taking the negative log of Equation (2) and substituting Γ_i gives us the final MvMF loss:

$$\ell_{\text{MvMF}}(\mathbf{x}, \mathbf{y}; M, K, W) = -\log \sum_{i=1}^{c} \Big(\Gamma_i(\mathbf{x}; W)\, \text{vMF}(\mathbf{y}, M_i, K_i) \Big). \tag{4}$$

When training a model with the MvMF loss, our goal is to find the best values for M, K, and W for a given dataset. Given a training dataset $(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)$ the training procedure solves the optimization

$$\hat{M}, \hat{K}, \hat{W} = \underset{M,K,W}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \ell_{\text{MvMF}}(\mathbf{x}_i, \mathbf{y}_i; M, K, W). \tag{5}$$

Training mixture models is difficult due to their non-convex loss functions, and good initial conditions are required to ensure convergence to a good local minimum. We use the following initialization in our experiments: initialize the W randomly; initialize μ_i to the center of the ith class used by the PlaNet method; and initialize all κ_i to the same initial value κ^0. We suggest using $\kappa^0 = \exp(16)$ based on experiments in Section 3.

The estimated GPS coordinate $\hat{\mathbf{y}}$ of a feature vector $\mathbf{x}$ is the coordinate with minimum loss. That is,

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathbb{S}^2}{\arg\min}\, \ell_{\text{MvMF}}(\mathbf{x}, \mathbf{y}; M, K, W). \tag{6}$$

Notice that this optimization is distinct from (5). This optimization does not get evaluated during model training, but only during inference. This optimization

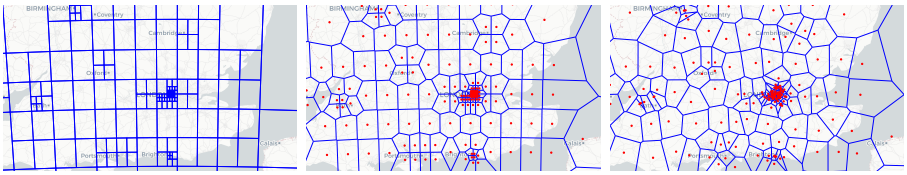

Fig. 3. (*Left*) Classes near London created using the PlaNet method. (*Middle*) Classes of the MvMF method with the μ_i initialized from the centers of the PlaNet classes. (*Right*) After training with the MvMF loss, the μ_i have shifted slightly to better fit the data, resulting in a new class partition.

is non-convex, and may have up to c distinct local minima. Algorithms exist for finding the minima of mixture models [3], but these algorithms require significant computation. Calculating $\hat{\mathbf{y}}$ for a single image may be feasible, but calculating $\hat{\mathbf{y}}$ for an entire test set is prohibitive. The classification interpretation of the MvMF loss presents an easy to interpret, computationally more efficient method for inference.

2.2 Interpretation as a classifier

The MvMF model can be interpreted as a classification model where each component represents a class. The mixture weights $\Gamma_i(\mathbf{x}, W)$ then become the probability associated with each class. The estimated location $\tilde{\mathbf{y}}$ is then the mean direction of the class with largest weight. Formally,

$$\tilde{\mathbf{y}} = \mu_{\tilde{i}}, \quad \text{where} \quad \tilde{i} = \underset{i \in \{1,\ldots,c\}}{\arg\max}\, \Gamma_i(\mathbf{x}, W). \tag{7}$$

Because this optimization is over a discrete space, it is extremely fast. When the mean directions M are initialized using the centers of the PlaNet classes, then there is a one-to-one correspondence between the MvMF classes and the PlaNet classes, albeit with the class shapes differing slightly (see Figure 3). In our experiments in Section 3, we use $\tilde{\mathbf{y}}$ as the estimated position.

Another advantage of the MvMF classes over the PlaNet classes is that the MvMF classes are fully parameterized by M. This means by jointly optimizing both W and M, we can learn not only which classes go with which images, but where on the earth the classes should be located.

2.3 Interpretation as a nearest neighbor method

We now describe how the MvMF model interpolates between classification models and nearest neighbor models. Recall that

$$\Gamma_i(\mathbf{x}, W) = \frac{\exp(-\mathbf{x}^\top \mathbf{w}_i)}{\sum_{j=1}^{c} \exp(-\mathbf{x}^\top \mathbf{w}_j)} \propto \exp(-\mathbf{x}^\top \mathbf{w}_i). \tag{8}$$

Solving for the $\tilde{i}$ in Equation (7) that maximizes Γ_i is therefore equivalent to finding the $\mathbf{w}_i$ that minimizes the inner product with $\mathbf{x}$. Minimum inner product search is a well studied problem, and in particular, it can be reduced to nearest neighbor search [2]. Therefore, when the number of classes equals the number of data points (i.e. $c = n$), and for each i we have $\mu_i = \mathbf{y}_i$ and $\mathbf{w}_i = \mathbf{x}_i$, then solving Equation (7) to find the output class is equivalent to solving a nearest neighbor problem.

2.4 Analysis

The MvMF's estimation error converges to zero at a rate of $O(\sqrt{d/n})$, where d is the number of feature dimensions and n the number of data points. This is in contrast to nearest neighbor methods (which converge at the exponential rate $\Omega(dn^{1/d})$ [Theorem 19.4 of 12]), and the cross entropy loss (which converges at a rate of $\Omega(\sqrt{cd/n})$ [Theorem 4 of 7]). Because c and d are both large in the geolocation setting, the MvMF loss requires significantly less training data to converge. Formal statements and proofs of these results can be found in [1].

3 Experiments

We evaluate the MvMF loss on the challenging task of image geolocation. In particular, we compare our MvMF method to the PlaNet method [13], since PlaNet is representative of cross entropy-based methods for geolocation. We show that the cross entropy-based methods require careful tuning of the hyperparameter c, but that our MvMF's performance always improves when increasing the number of classes c (as our theory predicts). This leads to significantly better performance of the MvMF method.

3.1 Procedure

Training Data. We use a previously existing publicly available dataset of geotagged images from Mousselly et. al. [11]. This dataset contains about 6 million images crawled from Flickr,[1] and the crawl was designed to be as representative as possible of Flickr's image database. The only filtering the dataset performed was to remove low resolution images. This dataset therefore comes from a distribution more similar to the PlaNet dataset than the other datasets.

Features. We use the WideResnet50 model [14] to generate a standard set of features in our experiments. WideResnet50 was originally trained on the ImageNet dataset for image classification, so we "fine-tune" the model's parameters to the geolocation problem. We chose the WideResnet50 model because empirical results show that fine-tuning works particularly well on resnet models [9], and the WideResnet50 is the best performing resnet model.

[1] The dataset originally contained about 14 million images, but many of them have since been deleted from Flickr and so were unavailable to us.

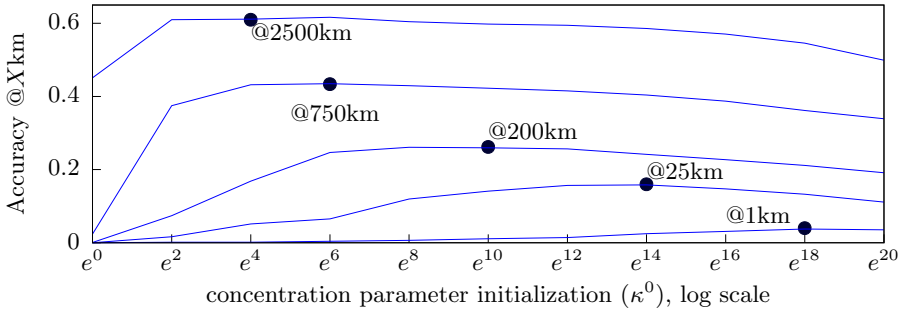

Fig. 4. Higher values of κ^0 result in better performance at fine grained prediction, and lower values of κ^0 result in better performance for course-grained prediction.

Fine-tuning a model is computationally cheaper than training from scratch, but it is still expensive. We therefore fine-tune the model only once, and use the resulting features in all experiments. To ensure that our fine-tuned features do not favor the MvMF method, we create a simple classification problem to fine-tune the features on. We associate each image with the country the image was taken in or "no country" for images from Antarctica or international waters. In total, this gives us a classification problem with 194 classes. We then fine-tune the WideResnet50 model for 20 epochs using the cross entropy loss, WideResnet50's standard feature augmentation, and the Adam [8] variant of SGD with a learning rate of $1 \times 10^{-}5$. This took about 2 months on a 4 CPU system with a Tesla K80 GPU and 64GB of memory. Because this fine-tuning procedure uses a cross entropy loss, the resulting features should perform especially well with cross entropy geolocation methods. Nonetheless, we shall see that the MvMF loss still outperforms cross entropy methods.

3.2 Results

Tuning the MvMF's hyperparameters. In this experiment, we set $c = 2^{15}$ and train MvMF models with $\kappa^0 = 0...20$. The results are shown in Figure 4. Accuracy @Xkm is a standard method for evaluating the performance of a geolocation system, and is equal to the fraction of data points whose estimated location is within Xkm of the true location. (Higher values are better.) For small X, Accuracy @Xkm measures the ability to geolocate strongly localizable images, and for large X, Accuracy @Xkm measures the ability to geolocate weakly localizable images.

We see that large values of κ^0 cause better geolocation for strongly localizable images, and small values of κ^0 cause better geolocation for weakly localizable images. This behavior has an intuitive explanation. When κ^0 is small, the variance of each component vMF distribution is large. So on each SGD step, weights from vMF components that are far away from the training data point will be updated. If the image is weakly localizable, then there are many locations where it might be placed, so many component weights should be updated. Conversely,

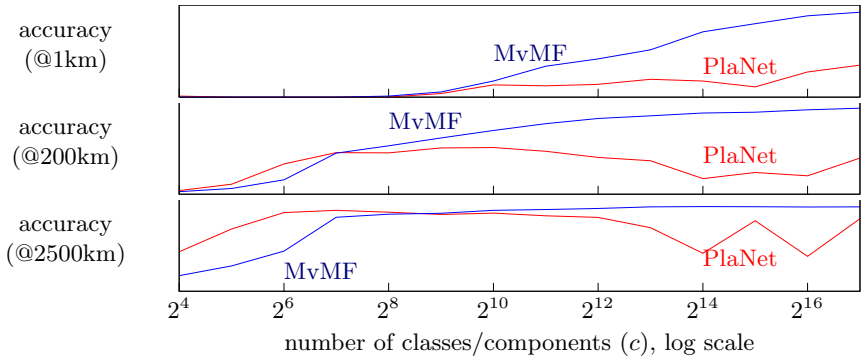

Fig. 5. The performance of the MvMF output layer increases monotonically as we increase the number of mixture components c, whereas the perfomance of PlaNet depends unpredictably on c.

when κ^0 is large, the component variances are small, and so only a small number of components get updated with each SGD step. Strongly localizable images can be exactly located to a small number of components, and so only a few components should be updated. We suggest using a value of $\kappa^0 = 16$ as a good balance, and use this value in all other experiments.

Tuning the number of classes *c*. This experiment demonstrates that c must be carefully tuned in the PlaNet method, but that increasing c always increases performance of the MvMF method. We emphasize that the original PlaNet paper [13] does not report results on the tuning of c, and so observing these limitations of the PlaNet method is one of the contributions of our work.

We train a series of models using the MvMF loss and PlaNet loss, varying c from 2^4 to 2^{17}. Theoretically, both methods support class sizes larger than $c = 2^{17}$, but our GPU hardware only had enough memory for 2^{17} classes. Figure 5 shows the results. For all X, we observe that PlaNet's performance is highly unpredictable as c varies, but the MvMF method always has improved accuracy as c increases. Figure 6 shows qualitatively why the PlaNet method is more sensitive to c than the MvMF.

Fine-tuned performance. In this experiment, we select several cross entropy and MvMF models and perform a second round of fine-tuning, this time with their true loss functions. We fine-tune with the Adam optimizer running for 5 epochs with learning rate 1×10^{-5}, which takes approximately 2 weeks per model on a single GPU. We evaluate the resulting model against the standard Im2GPS test set introduced by [6]. The results are shown in Table 1. When using the standardized training data and features, the MvMF loss significantly outperforms the cross entropy loss.

In Table 1, we also include results reported in the original PlaNet paper [13]. These results use a training data set that is 2 orders of magnitude larger than

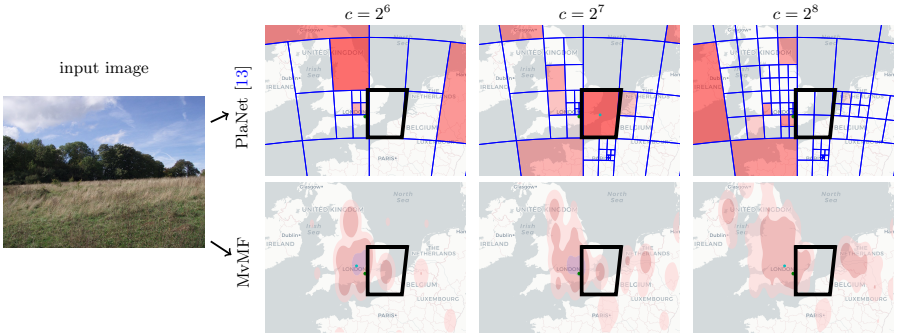

Fig. 6. The PlaNet [13] method's performance is highly sensative to the number of classes c. Consider the highlighted region. When $c = 2^6$, PlaNet assigns no probability to the region. (Brighter red indicates classes with higher probability.) When $c = 2^7$, PlaNet has split many other cells, causing the probability of the highlighted region to increase. When $c = 2^8$, PlaNet splits the highlighted region, causing the probability to drop again. This effect is exaggerated for weakly localizable images because many classes should be assigned high probability. In comparison, when the number of classes increases for the MvMF loss, the output smoothly takes on the shape of the underlying geography, which is the desired output for a weakly localizable image of grass.

the standardized training set, and so have significantly better performance than the cross entropy loss on the standard training set. This illustrates that the training data has a huge impact on the final model's performance. Surprisingly, the MvMF loss trained on standardized training set with only 6 million data points outperforms the PlaNet method trained on 126 million images.

4 Conclusion

The MvMF is the first neural network loss function designed for geolocating objects on the surface of the earth. The MvMF has better theoretical guarantees than previous nearest neighbor and classification methods, and these guarantees translate into better real world performance. We emphasize that the MvMF layer can be applied to any geolocation problem, not just image geolocation.

References

1. Exploiting the earth's spherical geometry to geolocate images. *ECMLPKDD*, 2019.
2. Yoram Bachrach, Yehuda Finkelstein, Ran Gilad-Bachrach, Liran Katzir, Noam Koenigstein, Nir Nice, and Ulrich Paquet. Speeding up the xbox recommender system using a euclidean transformation for inner-product spaces. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 257–264. ACM, 2014.
3. Miguel A. Carreira-Perpinan. Mode-finding for mixtures of gaussian distributions. *TPAMI*, 22(11):1318–1323, 2000.

			accuracy @				
loss	data/features	c	1km	25km	200km	750km	2500km
cross entropy	PlaNet [13]	$\approx 2^{15}$	**8.4**	24.5	37.6	53.6	71.3
cross entropy	standardized	2^{13}	1.0	4.1	10.1	24.8	44.8
cross entropy	standardized	2^{15}	0.6	2.0	7.3	26.1	49.9
cross entropy	standardized	2^{17}	1.8	6.0	11.8	27.9	51.3
MvMF	standardized	2^{13}	4.6	28.0	35.4	50.5	73.4
MvMF	standardized	2^{15}	6.0	31.2	41.1	58.0	75.7
MvMF	standardized	2^{17}	**8.4**	**32.6**	**39.4**	**57.2**	**80.2**

Table 1. Results on the Im2GPS test set [6]. The MvMF loss significantly outperforms the cross entropy loss at all distances when using standardized data and features. The MvMF loss trained on the standardized features even outperforms the PlaNet method, which was trained on a much larger dataset and required significantly more computation.

4. Siddharth Gopal and Yiming Yang. Von Mises-Fisher clustering models. In *ICML*, pages 154–162, 2014.
5. Md Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, Liming Chen, et al. von mises-fisher mixture model-based deep learning: Application to face verification. *arXiv preprint arXiv:1706.04264*, 2017.
6. James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *CVPR*. IEEE, 2008.
7. Elad Hazan, Tomer Koren, and Kfir Y Levy. Logistic regression: Tight bounds for stochastic and online optimization. In *COLT*, 2014.
8. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
9. Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? *arXiv preprint arXiv:1805.08974*, 2018.
10. K.V. Mardia and P.E. Jupp. *Directional Statistics*. 2009.
11. Hatem Mousselly-Sergieh, Daniel Watzinger, Bastian Huber, Mario Döller, Elöd Egyed-Zsigmond, and Harald Kosch. World-wide scale geotagged image dataset for automatic image annotation and reverse geotagging. In *MMSys*, 2014.
12. Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
13. Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *ECCV*, pages 37–55. Springer, 2016.
14. Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.