Mining self-similarity: Label super-resolution with epitomic representations

Nikolay Malkin[1], Anthony Ortiz[2], and Nebojsa Jojic[3]

[1] Yale University, New Haven, CT 06520, USA
kolya.malkin@yale.edu
[2] Microsoft AI for Good Research Lab, Redmond, WA 98052, USA
[3] Microsoft Research, Redmond, WA 98052, USA
{anthony.ortiz,jojic}@microsoft.com

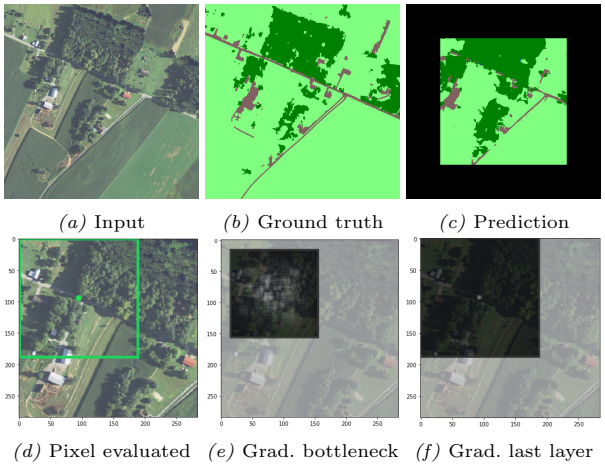
Abstract. We show that simple patch-based models, such as epitomes (Jojic et al., 2003), can have superior performance to the current state of the art in semantic segmentation and label super-resolution, which uses deep convolutional neural networks. We derive a new training algorithm for epitomes which allows, for the first time, learning from very large data sets and derive a label super-resolution algorithm as a statistical inference over epitomic representations. We illustrate our methods on land cover mapping and medical image analysis tasks.

Keywords: Label super-resolution, Semantic segmentation, Self-similarity

1 Introduction

Deep convolutional neural networks (CNNs) have become a tool of choice in computer vision. They typically outperform other approaches in core tasks such as object recognition and segmentation, but suffer from several drawbacks. First, CNNs are hard to interpret, which makes them difficult to improve by adding common-sense priors or invariances into the architecture. Second, they are usually trained in a supervised fashion on large amounts of labeled data, yet in most applications labels are sparse, leading to various domain adaptation challenges. Third, there is evidence of failure of the architecture choices that were meant to promote CNNs' reasoning over large distances in images. The *effective* receptive field [17] of CNNs – the distance at which faraway pixels stop contributing to the activity of deeper neurons – is often a small fraction of the theoretical one.

With the third point in mind, we ask a simple question, the answer to which can inform an agenda in building models which are interpretable, can be pretrained in an unsupervised manner, adopt priors with ease, and are amenable to well-understood statistical inference techniques: *If deep CNNs effectively use only small image patches for vision tasks, and learn from billions of pixels, then how would simple exemplar-like approaches perform, and can they be made practical computationally?* We show that models based on epitomic representations [14], illustrated in Fig. 2, match and surpass deep CNNs on several weakly supervised segmentation and domain transfer tasks.



(a) Input *(b)* Ground truth *(c)* Prediction

(d) Pixel evaluated *(e)* Grad. bottleneck *(f)* Grad. last layer

Fig. 1. Gradient-based effective receptive field estimation: We use the gradients from selected intermediate layers to the input image to estimate the size of the effective receptive field. In *(e)*, we visualize the normalized gradient map (at a single coordinate shown on green in *(d)*) of the U-Net's bottleneck (highest downsampling) layer with respect to the input image; *(f)* shows gradients of the *final* layer for the same pixel. The dark squares show the theoretical receptive field of the layers in question (139×139 for the bottleneck and 183×183 for the final layer). However, the gradient map *(f)* suggests that the effective receptive field is only about 13×13 pixels on average

For example, in Fig. 1 we show a patch of aerial imagery and the output of a U-Net [29] trained to predict land cover. The network misclassifies as vegetation the road pixels that appear in tree shadows. The model was trained on a large land cover map [3,27] that presents many opportunities to learn that roads are long and uninterrupted. The land cover data contains many more patterns that would help see rivers through a forest, recognize houses based on their proximity to roads, etc., but the U-Nets do not seem to learn such long-range patterns. This myopic behavior has been observed in other architectures as well [17,2,9].

In contrast, our algorithms directly model small image patches, forgoing long-range relationships. As generative models of images, epitomes are highly interpretable: they look like the images they were trained on (Fig. 2). Our generative formulation of image segmentation allows the inference of labels in the latent variable space, with or without high-resolution supervision (Fig. 4). They achieve comparable performance to the state-of-the-art CNNs on semantic segmentation tasks, and surpass the CNNs' performance in domain transfer and weakly supervised (label super-resolution) settings.

In summary, our contributions are as follows:

(1) As previous training algorithms fail to fit large epitomes well, we develop new algorithms that are suitable for mining self-similarity in very large datasets.

(2) We develop a new label super-resolution formulation that mines image self-similarity using epitomes or directly in a single (small) image.

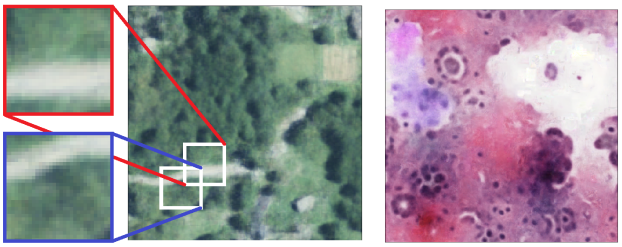


Fig. 2. A quarter of an epitome ($\boldsymbol{\mu}$ parameters shown) trained on aerial imagery (*left*) and an epitome trained on pathology slides (*right*). Any 31×31 training data patch is generated by, and likely similar to, some 31×31 window in the epitome. Note the two overlapping windows: the patches are distant in color space, but their corresponding mixture components share parameters on the intersection. The epitomes are $200\times$ and $30000\times$ smaller, respectively, than their total training data

(3) We show how these models surpass the recent (neural network) state of the art in aerial and pathology image analysis.

(4) We illustrate that our approaches allow and even benefit from unsupervised pre-training (separation of feature learning from label embedding).

(5) We show that our models deal with data size gracefully: We can train an epitome on a large fully labeled aerial imagery / land cover map and obtain better transfer in a new geography than CNNs [18,27], but we get even better results by analyzing one 512×512 tile at a time, with only low-resolution labels.

2 Epitomes as segmentation models

Epitomes [14] are an upgraded version of a Gaussian mixture model of image patches. In this section we present, for completeness, the definition of these models. We then explain how they can be turned into segmentation models.

Consider a training set consisting of image patches x^t unwrapped as vectors $x^t = \{x^t_{i,j,k}\}$, where i, j are coordinates in the patch and k is the spectral channel (R,G,B,...), and the corresponding vector of one-hot label embeddings $y_t = \{y^t_{i,j,\ell}\}$, $\ell \in \{1, \ldots, L\}$. In a mixture model, the distribution over the (image, label) data is represented with the aid of a latent variable $s \in \{1, \ldots, S\}$ as

$$p(x^t, y^t) = \sum_{s=1}^{S} p(x^t|s)p(y^t|s)p(s), \tag{1}$$

where $p(s)$ is the frequency of a mixture component s, while the conditional probability $p(x^t|s)$ describes the allowed variation in the image patch that s generates and $p(y^t|s)$ describes the likely labels for it. Under this model, the estimate for $\hat{y}$, the expected segmentation of a new image x, is

$$p(y|x) = \sum_{s} p(s|x)p(y|s). \tag{2}$$

A natural choice for $p(x|s)$ is a diagonal Gaussian distribution,

$$p(x|s) = \prod_{i,j,k} \frac{\exp\left(-\frac{1}{2}(x_{i,j,k} - \mu_{s,i,j,k})^2/\sigma^2_{s,i,j,k}\right)}{(2\pi\sigma^2_{s,i,j,k})^{\frac{1}{2}}} \tag{3}$$

and for $p(y|s)$ a product of categorical distributions over labels at each pixel position. The mean of the mixture component s contains pixel values $\mu_{s,i,j,k}$, while the covariance matrix is expressed in terms of its diagonal elements $\sigma^2_{s,i,j,k}$, the variances of different color channels k for individual pixels i, j.

Epitomic representations [14] compress this parametrization by recognizing that patches of interest come from *overlapping* regions and that different components s should share parameters. The component index $s = (s_1, s_2)$ lives on a $N \times N$ grid, so $0 \le s_1, s_2 \le N - 1$, and the parameters are shared:

$$\mu_{s,i,j,k} = \boldsymbol{\mu}_{s_1+i,s_2+j,k} \qquad \sigma^2_{s,i,j,k} = \boldsymbol{\sigma}^2_{s_1+i,s_2+j,k} \tag{4}$$

(Indices are to be interpreted modulo N, i.e., with toroidal wrap-around.) Thus, the epitome is a large grid of parameters $\boldsymbol{\mu}_{m,n,k}, \boldsymbol{\sigma}_{m,n,k}$, so that the parameters for the mixture component $s = (s_1, s_2)$ start at position s_1, s_2 and extend to the left and down by the size of the patch, as shown in Fig. 2. Modeling $K \times K$ patches will take K^2 times fewer parameters for the similar expressiveness as a regular mixture model trained on $K \times K$ patches. The posterior $p(s|x) \propto p(x|s)p(s)$ is efficiently computed using convolutions/correlations, e.g.,

$$p(s_1, s_2|x) \propto \exp \sum_{i,j,k} \frac{-1}{2}\left(\frac{x^2_{i,j,k}}{\boldsymbol{\sigma}^2_{s_1+i,s_2+j,k}} - \frac{2x_{i,j,k}\boldsymbol{\mu}_{s_1+i,s_2+j,k}}{\boldsymbol{\sigma}^2_{s_1+i,s_2+j,k}} + \frac{\boldsymbol{\mu}^2_{s_1+i,s_2+j,k}}{\boldsymbol{\sigma}^2_{s_1+i,s_2+j,k}} + \log \boldsymbol{\sigma}^2_{s_1+i,s_2+j,k}\right) \cdot p(s_1, s_2). \tag{5}$$

Epitomes are a summary of self-similarity in the images on which they are trained. They should thus contain a much smaller number of pixels than the training imagery, but be much larger than the patches with which they are trained. Each pixel in the epitome is contained in K^2 patches of size $K \times K$ and can be tracked back to many different positions in many images.

Conversely, this mapping of images enables embedding of *labels* into the epitome after the epitome of the images x has been trained. Every location in the epitome m, n will have (soft) label indicators $z_{m,n,\ell}$, computed as

$$p(\ell|m, n) \propto z_{m,n,\ell} = \sum_t \sum_{s_1,s_2:(m,n)\in W_{s_1,s_2}} p(s_1, s_2|x^t) y^t_{m-s_1,n-s_2,\ell}, \tag{6}$$

where W_{s_1,s_2} is the epitome window starting at (s_1, s_2), i.e. the set of K^2 coordinates (m, n) in the epitome that belong to the mixture component (s_1, s_2). The posterior tells us the strength of the mapping of the patch x^t to each component s that overlaps the position (m, n). The corresponding location in the patch of labels y^t is $(m - s_1, n - s_2)$, so $y^t_{m-s_1,n-s_2,\ell}$ is added to the count $z_{m,n,l}$ of label ℓ at location (m, n). Finally, we declare $p(y_{i,j,\ell}|s_1, s_2) \propto z_{s_1+i,s_2+j,\ell}$, allowing inference of ℓ for a new image patch by (2).

Fig. 3. Numerical near-fixed points of naïve epitome training by SGD without location promotion, caused by vanishing posteriors, and a 399×399 epitome trained with location promotion (*left*); non-diversifying and self-diversifying 499×499 epitomes trained on imagery of forests (*right*)

3 A large-scale epitome training algorithm

Epitomes have been used in recognition and segmentation tasks, e.g. [30,21,1,20,23,34,24,35]. However, the standard EM training algorithm [14] that maximizes the data log-likelihood $\sum_t \log \sum_s p(x^t|s)p(s)$ is not suitable to building *large* epitomes of *large* data sets due to the problem of "vanishing posterior". As training advances, the dynamic range of the posterior $p(s|x^t)$ becomes too big for machine precision, and the small probabilities are set to zero. Further parameter updates discourage mapping to these unlikely positions, leading to a die-off of chunks of "real estate" in the epitome. The problem is exacerbated by the size of the data (and of the epitome). Due to stability issues or computational cost, previous solutions to this [15] do not allow the models to be trained on the scale on which neural networks are trained. The analogous problem exists in estimating the prior $p(s)$ over epitome positions, which also needs to have a large dynamic range. If the range is flatter (e.g., if we use a uniform prior) then maximization of likelihood requires that the epitome learn only the most frequent patterns in the data, replicating slight variations of them everywhere. As imagery is mostly uniform and smooth, this creates blurry epitomes devoid of rarer features with higher variances, like various edges and corners.

Instead of EM, we develop a large-scale epitome learning algorithm combining three important ingredients: stochastic gradient descent, location promotion techniques, and the diversity-promoting optimization criterion:

Stochastic gradient descent. Instead of changing the parameters of the model based on all data at once, we update them incrementally in the direction of the gradient of the log-likelihood of a batch of individual data points $\frac{d}{d\theta} \log \sum_t \sum_s p(x^t|s)p(s)$, where $\theta = \{\boldsymbol{\mu}_{m,n,k}, \boldsymbol{\sigma}^2_{m,n,k}, p(s_1, s_2)\}$. Note that gradient descent alone does not solve the vanishing posterior problem, as the posterior also factors into the expression for the gradient (see the SI). In fact, SGD makes the situation worse (Fig. 3): the model parameters evolve before all of the data is seen, thus speeding up the extinction of the epitome's "real estate".

Location promotion. To maintain the relatively uniform evolution of all parts of the epitome, we directly constrain the learning procedure to hit all areas of the epitome through a form of posterior regularization [8]. Within an SGD framework, this can be accomplished simply by keeping counters R_{s_1,s_2} at each

position s_1, s_2 and incrementing them by the posterior $p(s_1, s_2|x^t)$ upon every sample x^t, then disallowing mapping to the windows s_1, s_2 which contain the *most frequently* mapped pixels. In particular, we compute a mask $M = \{R_{s_1,s_2} < c/N^2\}$, where $N \times N$ is the size of the epitome, for some small constant $c < 1$, and optimize only $\log \sum_{(s_1,s_2)\in M} p(x^t|s_1, s_2)p(s_1, s_2)$ at each gradient descent step. When $|M| > (1 - \delta)|N|^2$ for some small δ, all counters are reset to 0.

Diversification training. As illustrated in Fig. 3 (right), standard SGD tends to learn uniform patterns, especially when trained on large datasets. Just like EM, it has to rely on the prior $p(s)$ to avoid learning blurry epitomes, but the dynamic range needed to control this is too high. Additionally, through location promotion, we in fact encourage more uniform coverage of locations. Thus, we change the optimization criterion from log-likelihood of *all* data to log-likelihood of the worst modeled subset of each batch, $\sum_{t\in L_p} \sum_s p(x^t|s)$, were L_p is the set of data in the worst-modeled quantile p (the lowest quarter, in our experiments) in terms of data likelihood, either under a previously trained model or under the model being trained (*self-diversification*). This version of a max-min criterion avoids focusing on outliers while ensuring that the data is uniformly well modeled. The resulting epitomes capture a greater variety of features, as seen in the right panel of Fig. 3. The diversification criterion also helps the model generalize better on the test set, as we show in the experiments.

In the SI, we provide the details of the training parameters and analysis of execution time. The simple and runnable example training code[4] illustrates all three features of the algorithm.

4 Label super-resolution by self-similarity

Labeling images at a pixel level is costly and time-consuming, so a number of semi-supervised approaches to segmentation have been studied, e.g., [22,5,12,25]. Recently, [18] proposed a "label super-resolution" (LSR) technique which uses statistics of occurrence of high-resolution labels within coarse blocks of pixels labeled with a different set of low-resolution classes. (For clarity, we refer to low-res information as *classes* and high-res information as *labels*.) Each class, indexed by c, has a different composition of high-resolution labels, indexed by ℓ.

The label super-resolution technique in [18] assumes prior knowledge of the compositions $p(\ell|c)$ of high-res labels in low-res classes and uses them to define an alternative optimization cost at the top of a core segmentation network that predicts the high-res labels. Training the network end-to-end with coarse classes results in a model capable of directly predicting the high-res labels of the individual pixels. Backpropagation through such alternative cost criteria is prone to collapse, and [18] reports best results when the data with high-res labels (HR) is mixed with data with low-res labels (LR). Furthermore, the problem is inherently ill-posed: given an expressive enough model and a perfect learning algorithm, many solutions are possible. For example, the model could learn

[4] `https://github.com/anthonymlortiz/epitomes_lsr`

to recognize an individual low-res block and then choose an arbitrary pattern of high-res labels within it that satisfies the counts $p(\ell|c)$. Thus the technique depends on the inductive biases of the learning algorithm and the network architecture to lead to the desirable solutions.

On the other hand, following statistical models we discuss here, we can develop a statistical LSR inference technique from first principles. The data x is modeled by a mixture indexed by the latent index s. Using this index to also model the structure in the joint distribution over labels ℓ inside the patches generated by component s and classes c to which the patches belong, the known distribution of labels given the classes should satisfy $p(\ell|c) = \sum_s p(\ell|s)p(s|c)$. Thus, we find the label embedding $p(\ell|s)$ by minimizing the KL distance between the known $p(\ell|c)$ and the model's prediction $\sum_s p(\ell|s)p(s|c)$, i.e, by solving

$$p(\ell|s) = \arg\max_{p(\ell|s)} \sum_c p(c) \sum_\ell p(\ell|c) \log \sum_s p(\ell|s)p(s|c), \tag{7}$$

where $p(c)$ are the observed proportions of low-res classes in the data and $p(s|c)$ is obtained as the posterior over s for data of label c, as we will discuss in a moment. First, we derive an EM algorithm for solving the problem in Eq. 7 using auxiliary distributions $q_{\ell,c}(s)$ to repeatedly bound $\log \sum_s p(\ell|s)p(s|c)$ and reestimate $p(\ell|s)$. To derive the E step, we observe that

$$\log \sum_s p(\ell|s)p(s|c) = \log \sum_s q_{\ell,c}(s) \frac{p(\ell|s)p(s|c)}{q_{\ell,c}(s)} \geq \sum_s q_{\ell,c}(s) \log \frac{p(\ell|s)p(s|c)}{q_{\ell,c}(s)}.$$

The bound holds for all distributions $q_{\ell,c}$ and is made tight for

$$q_{\ell,c}(s) \propto p(\ell|s)p(s|c). \tag{8}$$

Optimizing for $p(\ell|s)$, we get

$$p(\ell|s) \propto \sum_c p(c)p(\ell|c)q_{\ell,c}(s). \tag{9}$$

Coordinate ascent on the $q_{l,c}(s)$ and $p(\ell|s)$ by iterating (8) and (9) converges to a local maximum of the optimization criterion.

Therefore, all that is needed for label super-resolution are the distributions $p(s|c)$ that tell us how often each mixture component is seen within the class c. Given low-res labeled data, i.e., pairs (x^t, c^t) and a trained mixture model for image patches x^t, the answer is

$$p(s|c) \propto \sum_{t:c^t=c} p(s|x^t). \tag{10}$$

In other words, we go through all patches, look at the posterior of their assignment to prototypes s, and count how many times each prototype was associated with each of the classes.

The epitomic representation with its parameter sharing has an additional advantage here. With standard Gaussian mixtures of patches, the level of the

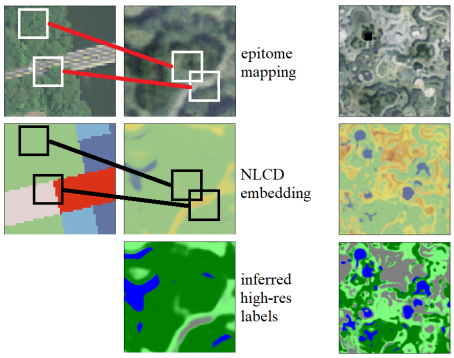


Fig. 4. Two image patches are shown mapped to a piece of an epitome (*left*). Below the source image, we show class labels for 30×30m blocks. Below the epitome we show a piece of the class embedding $p(m, n|c)$ at a pixel level (Eqn. 11) using the same color scheme, with colors weighted by the inferred class probabilities. Below the class embedding we show the piece of the output of the label super-resolution algorithm in Section 4. We also show the full epitome and its embeddings (*right*)

super-resolution we can accomplish is defined by the size of the patch x we use in the analysis, because all of the reasoning is performed on the level of the patch index s, not at individual pixels. Thus, to get super-resolution at the level of a single pixel, our mixture model would have to be over individual pixels, i.e., a simple color clustering model (see the SI for examples). With epitomes, however, instead of using whole patch statistics, we can assign statistics $p(m, n|c)$ to individual positions in the epitome,

$$p(m, n|c) \propto \sum_t \sum_{i,j} p((s_1, s_2) = (m - i, n - j)|x^t)[c^t = c], \tag{11}$$

where $p(\cdot, \cdot|x^t)$ is the posterior over positions. This equation represents counting how many times each *pixel* in the epitome was mapped to by a patch that was inside a block of class c, as illustrated in Fig. 4: While the two patches map close to each other into the epitome, the all-forest patch is unlikely to cover any piece of the road. Considering all patches in a larger spatial context, the individual pixels in the epitome can get statistics that differ from their neighbors'. This allows the inference of high-res labels ℓ for the entire epitome, shown with its embedding of low-res classes c and super-resolved high-res labels ℓ on the right.

In summary, our LSR algorithm first uses the epitome model of $K \times K$ patches to embed class labels on an individual pixel level using Eq. 11. This then allows us to run the EM algorithm that iterates Eqs. 8 and 9 on positions m, n associated with the shared parameters in the epitome instead of mixture components s, using $p(m, n|c)$ in Eq. 11 in place of $p(s|c)$. Once the estimate of the high-res labels $p(\ell|m, n)$ is computed for each position in the epitome, we can predict labels in imagery using Eq. 2. This procedure performs probabilistic

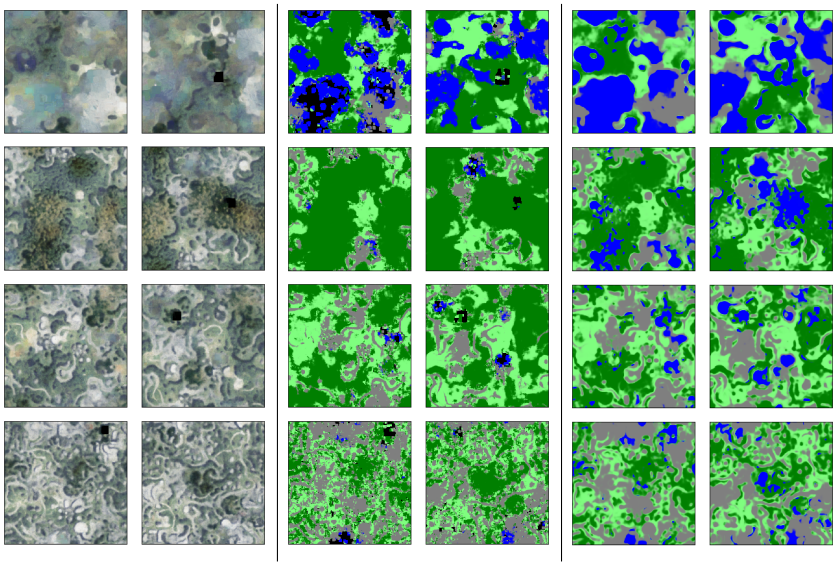


Fig. 5. Epitomes (total area $2 \cdot 10^6$ pixels) trained on $5 \cdot 10^9$ pixels of **South** imagery (*left*); land cover embeddings (argmax label shown) derived from high-resolution **South** ground truth (*middle*), land cover embeddings derived by epitomic LSR from **North** 30m-resolution NLCD data (*right*)

reasoning over the frequencies of repeating patterns in imagery labeled with low-resolution classes to reason over individual pixels in these patterns.

5 Experiments

5.1 Land cover segmentation and super-resolution

Our first example is the problem of land cover segmentation from aerial imagery. We work with the data studied by [27], available for 160,000km^2 of land in the Chesapeake Bay watershed (Northeast US):

(1) 1m-resolution 4-band aerial imagery (NAIP) taken in the years 2013-4;

(2) High-resolution (1m) land cover segmentation in four classes (water, forest, field/low vegetation, built/impervious) produced by [3];

(3) Low-resolution (30m) land cover labels from the National Land Cover Database (NLCD) [11].

As in [27], the data is split into **South** and **North** regions, comprising the states of MD, VA, WV, DE (S) and NY and PA (N). Our task is to produce 1m-resolution land cover maps of the **North** region, using only the imagery, possibly the low-res classes, and possibly the high-res labels from just the **South** region. The predictions are evaluated against high-res ground truth in the **North** region.

Despite the massive scale of the data, differences such as imaging conditions and frequency of occurrence of vegetation patterns make it difficult for neural

networks trained to predict high-res labels from imagery in the **South** region to transfer to **North**. However, in their study of this problem using data fusion methods, [27] obtained a large improvement in **North** performance by multi-task training: the networks were trained to predict high-res labels with the objectives of (1) cross-entropy loss against high-res labels in **South** and (2) super-resolution loss [18] against the distributions determined by low-res NLCD labels in **North** (see the first and third rows of Table 1).

Epitome training. We fit eight 499×499 epitomes to all available **South** imagery. To encourage a diversity of represented land types, for each of the four high-res labels ℓ (water, forest, field, built), we trained a self-diversifying epitome $E_0^{(\ell)}$ on patches of size 11×11 to 31×31 containing at least one pixel labeled with label ℓ. We then trained a model $E_1^{(\ell)}$ on the quarter of such patches with lowest likelihood under $E_0^{(\ell)}$ and a model $E_2^{(\ell)}$ on the quarter with lowest likelihood under $E_1^{(\ell)}$. The first epitome $E_0^{(\ell)}$ was then discarded.[5] The final model is a uniform mixture of the $E_i^{(\ell)}$ $(i = 1, 2)$. The $\boldsymbol{\mu}_{m,n}$ parameters of its components can be seen in the left column of Fig. 5. (Notice that while the epitomes in each row were trained on patches containing pixels of a given label ℓ, other label appear in them as well. For example, we see roads in the forest epitome (second row), since roads are sometimes found next to trees, and indeed are poorly modeled by a model of only trees, cf. Fig. 3.)

High-resolution label embedding. We derive high-resolution soft label embeddings $p(\ell|m, n)$ from high-res **South** labels by the following procedure: for 10 million iterations, we uniformly sample a 31×31 patch of **South** imagery x^t and associated high-res labels y^t and evaluate the posterior over positions $p(s_1, s_2|x^t)$, then embed the center 11×11 patch of labels y^t weighted by the posterior (sped up by sampling; see the SI for details). The label embeddings $p(\ell|m, n) \propto z_{m,n,\ell}$ are proportional to the sum of these embeddings over all patches; these quantities estimate the probability that a patch generated by an epitome window with center near (s_1, s_2) would generate label ℓ at the corresponding position. These embeddings are shown in the middle column of Fig. 5.

Low-resolution NLCD embedding. Using the same set of epitomes trained on **South**, we derive the posteriors $p(m, n|c)$ given a low-resolution class c: we sample 11×11 patches x^t from **North** with center pixel labeled with low-res class c^t and embed the label c^t weighted by the posterior $p(s_1, s_2|x^t)$. By (11), $p(m, n|c)$ is then proportional to the sum of these embeddings. An example of the embeddings in one epitome component is shown in Fig. 4.

Epitomic label super-resolution. The joint distribution of high-res and low-res classes, $p(\ell|c)$, can be estimated on a small subset of jointly labeled data; we use the statistics reported by [18]. We apply our LSR algorithm to the low-res embeddings $p(m, n|c)$, the joint $p(c, \ell)$, and the known distribution $p(c)$ to arrive at high-res label probabilities at each epitome position, $p(\ell|m, n)$. They are shown in the right column of Fig. 5.

[5] $E_2^{(\ell)}$ is trained to model the patches poorly modeled by the self-diversifying $E_1^{(\ell)}$. Hence, $E_2^{(\ell)}$ simply has much higher posteriors and more diversity of texture.

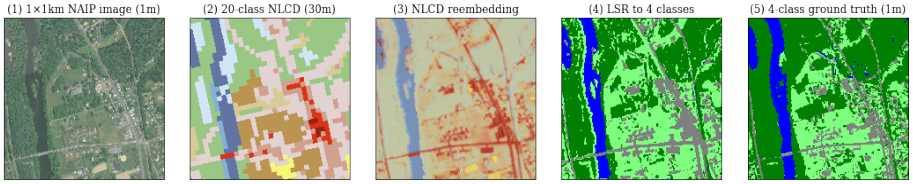


Fig. 6. Self-epitomic LSR on a 1024×1024 patch of land (1). The low-res classes (2) are embedded at locations similar in appearance, yielding (3). The inference procedure described in Sec. 4 produces (4), which closely resembles the ground truth (5)

Table 1. Performance of various methods on land cover segmentation in the **North** region. We report overall accuracy and mean intersection/union (Jaccard) index

Model	Label training set	Acc.	IoU
U-Net [27]	HR (S)	59.4%	40.5%
Epitome (S imagery)	HR (S)	79.5	59.3
U-Net neural LSR [18,27]	HR (S), LR (N)	86.9	62.5
U-Net neural LSR [18]	LR (N)	80.1	41.3
256^2 self-epitomic LSR	LR (N)	85.9	63.3
512^2 self-epitomic LSR	LR (N)	87.0	65.3
1024^2 self-epitomic LSR	LR (N)	87.8	66.9
2048^2 self-epitomic LSR	LR (N)	88.0	67.8
All-tile epitomic LSR	LR (N)	83.9	58.5

We evaluate the two epitome embeddings $p(\ell|m,n)$, derived from high-res labels in **South** or from low-res classes in **North**, on a sample of 1600km^2 of imagery in the **North** region in the following fashion: we select 31×31 patches x^t and reconstruct the labels in the center 11×11 blocks as the posterior-weighted mean of the $p(\ell|m,n)$. At the large scale of data, this requires an approximation by sampling, see the SI for details. The results are shown in the second and last rows of Table 1.

When the area to be super-resolved is small, we can perform epitomic LSR *using the imagery itself as an epitome*. We experiment with small tiles from **North** (256×256 up to 2048×2048 pixels). For a given tile, we initialize an epitome with the same size as the tile, with uniform prior, mean equal to the true pixel intensities, and fixed variance $\sigma^2 = 0.01$. We then embed low-res NLCD labels from the tile into this epitome just as described above and run the LSR inference algorithm. The probabilities $p(\ell|m,n)$ are then the predicted land cover labels[6]. An example appears in Fig. 6, and more in the SI. The results of this *self-epitomic* LSR, performed on a large evaluation set dissected into tiles of different sizes, can be seen in Table 1.

Results. From Table 1, we draw the following conclusions:

[6] We found it helpful to work with $2\times$ downsampled images and use 7×7 patches for embedding, with approximately $0.05|W|^2$ patches sampled for tiles of size $W \times W$.

*Epitomes trained only on imagery and high-res labels in **South** transfer better to **North** than U-Nets that use the same data.* The U-Nets trained only on imagery and high-res labels in the **South** region transfer poorly to **North**: patterns associated, for example, with forests in the **North** are more frequently associated with fields in **South**, and the discriminatively trained models couple the high-frequency patterns in **South** with their associated land cover labels. Most surprisingly, even the U-Nets trained on the LR **North** imagery perform worse than any of the epitome models trained on the same data.[7]

There is evidence that the far better transfer performance of the epitomes is due to generative training. First, it is nearly unsupervised: no labels are seen in training, except to weakly guide the sampling of patches. Second, diversification training ensures, for example, that forests resembling those found in **North**, while rare, still appear in the epitomes trained on **South** imagery and receive somewhat accurate label embeddings. The posterior on those areas of the epitomes is then much higher in the **North**. (In the SI we show the mean posteriors over epitome positions illustrating this point.)

The self-similarity in images that defines the repetition of patterns in certain classes is highly local. If we were to study self-similarity in a large region, we would be bound to find that some imagery patterns that are associated with a particular high-res label in one area are less so in another. Therefore, the size of the area on which to perform LSR reasoning is an important design parameter. If the area is too small, then we may not get enough observations of coarse classes to unambiguously assign high-res patterns to them: indeed, self-epitomic LSR accuracy increases with the size of the tile. It is remarkable that we can get better high-res segmentation results than the state of the art by studying one 512×512 patch at a time, together with low-res classes for 30×30 blocks, and no other training data or high-res labels.

On the other hand, when the area is too large, then the pattern diversity increases and ambiguity may reduce the effectiveness of the method. Furthermore, when the area is too large, self-epitomic LSR is not computationally practicable – the imagery must be compressed in an epitome to mine self-similarity. All-tile epitomic LSR improves over the baseline models although *no high-res labels are seen*, while the best-performing U-Nets required high-res labels in **South**, low-res classes in **North**, and imagery from both **South** and **North** in training.

5.2 Lymphocyte segmentation in pathology images

Our second example is the task of identifying tumor-infiltrating lymphocytes (TILs) in pathology imagery. We work with a set of 50000 240×240 crops of 0.5μm-resolution H&E-stained tumor imagery [31]. There is no high-res *segmentation* data available for this task. However, [13] produced a set of 1786 images centered on single cells, labeled with whether the center cell is a TIL, on which our methods can be evaluated.

[7] We used training settings identical to those of [18]. The training collapsed to a minimum in which the "water" class was not predicted, but the accuracy would be lower than that of all-tile epitomic LSR even if all water were predicted correctly.

Table 2. Performance of various methods on the TIL segmentation task. We report the area under the ROC curve

Model	Label training set	AUC
Manual features SVM [36,13]	HR	0.713
CNN [13]	HR	0.494
CNN with pretraining [13]	HR	0.786
U-Net neural LSR [18]	LR + color masks	0.783
Non-div. epitomic LSR	LR	0.794
Div. epitomic LSR	LR	0.801

The best results for this task that used high-resolution supervision required either a manually tuned feature extraction pipeline and SVM classifier [36,13] or, in the case of CNNs, a sparse autoencoder pretraining mechanism [13]. More recently, [18] nearly matched the supervised CNN results using the neural label super-resolution technique: the only guidance available to the segmentation model in training was low-resolution estimates of the probability of TIL infiltration in 100×100 regions for the entire dataset derived by [31], as well as weak pixel-level rules (masking regions below certain thresholds of hematoxylin level).

We address the same problem as [18], using the low-res probability maps as the only supervision in epitomic LSR:

Epitome training. We train 299×299 epitomes on patches of size 11×11 to 31×31 intersecting the center pixels of the images to be segmented. The resulting models trained with and without self-diversification are shown in Fig. 7.

Low-resolution embedding. Following [18], we define 10 classes c, for each range of density estimates $[0.1 \cdot n, 0.1 \cdot (n+1)]$. We find the posteriors $p(m, n|c)$ by embedding 1 million 11×11 patches from the entire dataset.

Epitomic label super-resolution. We estimate the mean TIL densities in each probability range, $p(\ell|c)$ and set a uniform prior $p(c)$. We then produce the probabilities of TIL presence per position $p(\ell|m, n)$ by the LSR algorithm.

We then evaluate our models on the data for which high-res labels exist by sampling 11×11 patches x containing the center pixel – 100 for each test image – and computing the mean probability of TIL presence $\sum_s p(\ell|s)p(s|x)$ as the final prediction score. We obtained better results when we instead averaged the probability of TIL presence *anywhere* in an embedded patch in the epitome, that is, convolved $p(\ell|s)$ with a 11×11 uniform filter before computing this sum.

Results. As summarized in Table 2, our epitomic LSR outperforms all previous methods, including both the supervised models and the neural LSR, with self-diversifying epitomes providing the greatest improvement. The results suggest that TIL identification is a highly local problem. Deep CNNs, with their large receptive fields, require hand-engineered features or unsupervised pretraining to reach even comparable performance. In addition, epitomes are entirely unsupervised and thus amenable to adaptation to new tasks, such as classifying other types of cells: given coarse label data, we may simply embed it into the pretrained epitomes and perform LSR.

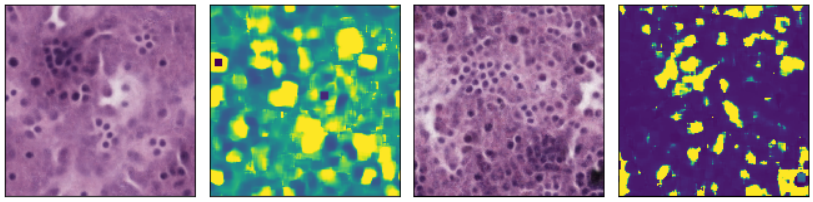


Fig. 7. Epitomes trained on tumor imagery and the embedding of the tumor-infiltrating lymphocyte label. The model on the right was trained with self-diversification

6 Conclusion

Motivated by the observation that deep convolutional networks usually have a small effective receptive field, we revisit simple patch mixture models, in particular, epitomes. As generative models that allow addition of latent variables, these approaches have several advantages. They are interpretable: an epitome looks like the imagery on which it was trained (Fig. 2), and examining the posteriors over epitome positions is akin to understanding weights for many neurons at once. The desired invariances can be directly modeled with additional hidden variables, just as [7] modeled illumination. They can be combined with other statistical procedures, as we show with our novel label super-resolution formulation (Sec. 4). They can be pretrained on a large amount of unlabeled data so that a small number of labeled points are needed to train prediction models, and they can be a base of hierarchical or pyramidal models that reason over long ranges, e.g., [4,26,33,6,16]. Using epitome-derived features in tasks that require long-range reasoning, such as common benchmarks for segmentation or classification of large images, is an interesting subject for future work.

Just as deep neural networks suffered from the vanishing gradient problem for years, before such innovations as stagewise pretraining [10], dropout [32], and the recognition of the numerical advantages of ReLU units [19], epitomic representations had suffered from their own numerical problems stemming from the large dynamic range of the posterior distributions. As a remedy, we designed a new large-scale learning algorithm that allowed us to run experiments on hundreds of billions of pixels. We showed that simply through mining patch self-similarity, epitomic representations outperform the neural state of the art in domain transfer and label super-resolution in two important application domains.

We direct the reader to the SI for more examples, code, results on another competition dataset (in which epitomes were the basis for the winning method [28]), and discussion on future research.

Acknowledgments: The authors thank Caleb Robinson for valuable help with experiments [28] and the reviewers for comments on earlier versions of the paper.

References

1. Bazzani, L., Cristani, M., Perina, A., Murino, V.: Multiple-shot person re-identification by chromatic and epitomic analyses. Pattern Recognition Letters **33**(7), 898–903 (2012)
2. Brendel, W., Bethge, M.: Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In: International Conference on Learning Representations (ICLR) (2019)
3. Chesapeake Conservancy: Land cover data project (January 2017), `https://chesapeakeconservancy.org/wp-content/uploads/2017/01/LandCover101Guide.pdf`, [Online]
4. Cheung, V., Jojic, N., Samaras, D.: Capturing long-range correlations with patch models. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2007)
5. Dai, J., He, K., Sun, J.: Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1635–1643 (2015)
6. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. vol. 2, pp. 1816–1823. IEEE (2005)
7. Frey, B.J., Jojic, N.: Transformed component analysis: Joint estimation of spatial transformations and image components. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. vol. 2, pp. 1190–1196. IEEE (1999)
8. Ganchev, K., Gillenwater, J., Taskar, B., et al.: Posterior regularization for structured latent variable models. Journal of Machine Learning Research **11**(Jul), 2001–2049 (2010)
9. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations (ICLR) (2019)
10. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. science **313**(5786), 504–507 (2006)
11. Homer, C., Dewitz, J., Yang, L., Jin, S., Danielson, P., Xian, G., Coulston, J., Herold, N., Wickham, J., Megown, K.: Completion of the 2011 national land cover database for the conterminous united states–representing a decade of land cover change information. Photogrammetric Engineering & Remote Sensing **81**(5), 345–354 (2015)
12. Hong, S., Noh, H., Han, B.: Decoupled deep neural network for semi-supervised semantic segmentation. In: Advances in neural information processing systems. pp. 1495–1503 (2015)
13. Hou, L., Nguyen, V., Kanevsky, A.B., Samaras, D., Kurc, T.M., Zhao, T., Gupta, R.R., Gao, Y., Chen, W., Foran, D., et al.: Sparse autoencoder for unsupervised nucleus detection and representation in histopathology images. Pattern Recognition (2018)
14. Jojic, N., Frey, B.J., Kannan, A.: Epitomic analysis of appearance and shape. In: ICCV. vol. 3, p. 34 (2003)
15. Jojic, N., Perina, A., Murino, V.: Structural epitome: a way to summarize one's visual experience. In: Advances in neural information processing systems. pp. 1027–1035 (2010)

16. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). vol. 2, pp. 2169–2178. IEEE (2006)
17. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 4898–4906. Curran Associates, Inc. (2016), http://papers.nips.cc/paper/6203-understanding-the-effective-receptive-field-in-deep-convolutional-neural-networks.pdf
18. Malkin, K., Robinson, C., Hou, L., Soobitsky, R., Czawlytko, J., Samaras, D., Saltz, J., Joppa, L., Jojic, N.: Label super-resolution networks. International Conference on Learning Representations (2019)
19. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814 (2010)
20. Ni, K., Kannan, A., Criminisi, A., Winn, J.: Epitomic location recognition. IEEE transactions on pattern analysis and machine intelligence **31**(12), 2158–2167 (2009)
21. Nilsback, M.E., Zisserman, A.: A visual vocabulary for flower classification. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). vol. 2, pp. 1447–1454. IEEE (2006)
22. Papandreou, G., Chen, L.C., Murphy, K.P., Yuille, A.L.: Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: Proceedings of the IEEE international conference on computer vision. pp. 1742–1750 (2015)
23. Papandreou, G., Chen, L.C., Yuille, A.L.: Modeling image patches with a generic dictionary of mini-epitomes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2051–2058 (2014)
24. Papandreou, G., Kokkinos, I., Savalle, P.A.: Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 390–399 (2015)
25. Pathak, D., Krahenbuhl, P., Darrell, T.: Constrained convolutional neural networks for weakly supervised segmentation. In: Proceedings of the IEEE international conference on computer vision. pp. 1796–1804 (2015)
26. Perina, A., Jojic, N.: Spring lattice counting grids: Scene recognition using deformable positional constraints. In: European Conference on Computer Vision. pp. 837–851. Springer (2012)
27. Robinson, C., Hou, L., Malkin, K., Soobitsky, R., Czawlytko, J., Dilkina, B., Jojic, N.: Large scale high-resolution land cover mapping with multi-resolution data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12726–12735 (2019)
28. Robinson, C., Malkin, K., Hu, L., Dilkina, B., Jojic, N.: Weakly supervised semantic segmentation in the 2020 IEEE GRSS Data Fusion Contest. Proceedings of the International Geoscience and Remote Sensing Symposium (2020)
29. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)

30. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
31. Saltz, J., Gupta, R., Hou, L., Kurc, T., Singh, P., Nguyen, V., Samaras, D., Shroyer, K.R., Zhao, T., Batiste, R., et al.: Spatial organization and molecular correlation of tumor-infiltrating lymphocytes using deep learning on pathology images. Cell reports **23**(1), 181 (2018)
32. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
33. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. In: European conference on computer vision. pp. 18–32. Springer (2000)
34. Yeung, S., Kannan, A., Dauphin, Y., Fei-Fei, L.: Epitomic variational autoencoders (2016)
35. Zhang, H., Fritts, J.E., Goldman, S.A.: Image segmentation evaluation: A survey of unsupervised methods. computer vision and image understanding **110**(2), 260–280 (2008)
36. Zhou, N., Yu, X., Zhao, T., Wen, S., Wang, F., Zhu, W., Kurc, T., Tannenbaum, A., Saltz, J., Gao, Y.: Evaluation of nucleus segmentation in digital pathology images through large scale image synthesis. In: Medical Imaging 2017: Digital Pathology. vol. 10140, p. 101400K. International Society for Optics and Photonics (2017)