Learning to Select Pre-trained Deep Representations with Bayesian Evidence Framework

Yong-Deok Kim[*1]   Taewoong Jang[*2]   Bohyung Han[3]   Seungjin Choi[3]

[1]Software R&D Center, Device Solutions, Samsung Electronics, Korea
[2]Stradvision Inc., Korea
[3]Department of Computer Science and Engineering, POSTECH, Korea

yd.mlg.kim@samsung.com taewoong.jang@stradvision.com {bhhan,seungjin}@postech.ac.kr

Abstract

We propose a Bayesian evidence framework to facilitate transfer learning from pre-trained deep convolutional neural networks (CNNs). Our framework is formulated on top of a least squares SVM (LS-SVM) classifier, which is simple and fast in both training and testing, and achieves competitive performance in practice. The regularization parameters in LS-SVM is estimated automatically without grid search and cross-validation by maximizing evidence, which is a useful measure to select the best performing CNN out of multiple candidates for transfer learning; the evidence is optimized efficiently by employing Aitken's delta-squared process, which accelerates convergence of fixed point update. The proposed Bayesian evidence framework also provides a good solution to identify the best ensemble of heterogeneous CNNs through a greedy algorithm. Our Bayesian evidence framework for transfer learning is tested on 12 visual recognition datasets and illustrates the state-of-the-art performance consistently in terms of prediction accuracy and modeling efficiency.

1. Introduction

Image representations from deep CNN models trained for specific image classification tasks turn out to be powerful even for general purposes [2, 6, 7, 21, 23] and useful for transfer learning or domain adaptation. Therefore, CNNs trained on specific problems or datasets are often fine-tuned to facilitate training for new tasks or domains [2, 6, 13, 17, 19, 36], and an even simpler approach— application of off-the-shelf classification algorithms such as SVM to the representations from deep CNNs [7]—is getting more attractive in many computer vision problems. However, fine-tuning of an entire deep network still requires a lot of efforts and resources, and SVM-based methods also
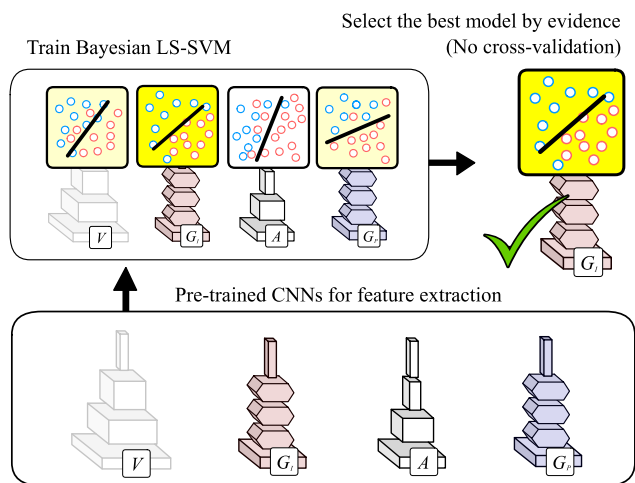

Figure 1. We address a problem to select the best CNN out of multiple candidates as shown in this figure. Additionally, our algorithm is capable of identifying the best ensemble of multiple CNNs to further improve performance.

involve time consuming grid search and cross validation to identify good regularization parameters. In addition, when multiple pre-trained deep CNN models are available, it is unclear which pre-trained models are appropriate for target tasks and which classifiers would maximize accuracy and efficiency. Unfortunately, most existing techniques for transfer learning or domain adaptation are limited to empirical analysis or ad-hoc application specific approaches.

We propose a simple but effective algorithm for transfer learning from pre-trained deep CNNs based on Bayesian least squares SVM (LS-SVM), which is formulated with Bayesian evidence framework [16, 29] and LS-SVM [26]. This approach automatically determines regularization parameters in a principled way, and shows comparable performance to the standard SVMs based on hinge loss or squared hinge loss. More importantly, Bayesian LS-SVM provides an effective solution to select the best CNN out of multiple

---

[*]This work was done when Y. Kim and T. Jang were with POSTECH.

candidates and identify a good ensemble of heterogeneous CNNs for performance improvement. Figure 1 illustrates our approach. We also propose a fast Bayesian LS-SVM, which maximizes the evidence more efficiently based on Aitken's delta-squared process [1].

One may argue against the use of LS-SVM for classification because the least squares loss function in LS-SVM tends to penalize well-classified examples. However, least squares loss is often used for training multilayer perceptron [4] and shows comparable performance to SVMs [28, 37]. In addition, Bayesian LS-SVM provides a technically sound formulation with outstanding performance in terms of speed and accuracy for transfer learning with deep representations. We also propose a fast Bayesian LS-SVM, which maximizes the evidence more efficiently based on Aitkens delta-squared process [1]. Considering simplicity and accuracy, we claim that our fast Bayesian LS-SVM is a reasonable choice for transfer learning with deep learning representation in visual recognition problems. Based on this approach, we achieved promising results compared to the state-of-the-art techniques on 12 visual recognition tasks.

The rest of this paper is organized as follows. Section 2 describes examples of transfer learning or domain adaptation based on pre-trained CNNs for visual recognition problems. Then, we discuss Bayesian evidence framework applicable to the same problem in Section 3 and its acceleration technique using Aitken's delta-squared process in Section 4. The performance of our algorithm in various applications is demonstrated in Section 5.

2. Related Work

Since *AlexNet* [15] demonstrated impressive performance in the ImageNet large scale visual recognition challenge (LSVRC) 2012, a few deep CNNs with different architectures, *e.g.*, *VGG* [25] and *GoogLeNet* [27], have been proposed in the subsequent events. Instead of training deep CNNs from scratch, some people have attempted to refine pre-trained networks for new tasks or datasets by updating the weights of all neurons or have adopted the intermediate outputs of existing deep networks as generic visual feature descriptors. These strategies can be interpreted as transfer learning or domain adaptation.

Refining a pre-trained CNN is called fine-tuning, where the architecture of the network may be preserved while weights are updated based on new training data. Fine-tuning is generally useful to improve performance [2, 6, 13, 36] but requires careful implementation to avoid overfitting. The second approach regards the pre-trained CNNs as feature extraction machines and combines the deep representations with the off-the-shelf classifiers such as linear SVM [7, 34], logistic regression [7, 34], and multi-layer neural network [21]. The techniques in this category have been successful in many visual recognition tasks [2, 23, 24].

When combining a classification algorithm with image representations from pre-trained deep CNNs, we often face a critical issue. Although several deep CNN models trained on large scale image repositories are publicly available, there is no principled way to select a CNN out of multiple candidates and find the best ensemble of multiple CNNs for performance optimization. Existing algorithms typically rely on ad-hoc methods for model selection and fail to provide clear evidence for superior performance [2].

3. Bayesian LS-SVM for Model Selection

This section discusses a Bayesian evidence framework to select the best CNN model(s) in the presence of transferable multiple candidates and identify a reasonable regularization parameter for LS-SVM classifier automatically.

3.1. Problem Definition and Formulation

Suppose that we have a set of pre-trained deep CNN models denoted by $\{\mathsf{CNN}_m | m = 1 \dots M\}$. Our goal is to identify the best performing deep CNN model among the M networks for transfer learning. A naïve approach is to perform fine tuning of network for target task, which requires substantial efforts for training. Another option is to replace some of fully connected layers in a CNN with an off-the-shelf classifier such as SVM and check the performance of target task through parameter tuning for each network, which would also be computationally expensive.

We adopt a Bayesian evidence framework based on LS-SVM to achieve the goal in a principled way, where the evidence of each network is maximized iteratively and the maximum evidences are used to select a reasonable model. During the evidence maximization procedure, the regularization parameter of LS-SVM is identified automatically without time consuming grid search and cross-validation. In addition, the Bayesian evidence framework is also applied to the construction of an ensemble of multiple CNNs to accomplish further performance improvement.

3.2. LS-SVM

We deal with multi-label or multi-class classification problem, where the number of categories is K. Let $\mathcal{D} = \{(\boldsymbol{x}_n, y_n^{(k)}), k = 1 \dots K\}_{n=1 \dots N}$ be a training set, where $\boldsymbol{x}_n \in \mathbb{R}^D$ is a feature vector and $y_n^{(k)}$ is a binary variable that is set to 1 if label k is given to $\boldsymbol{x}_n$ and 0 otherwise. Then, for each class k, we minimize a least squares loss with L_2 regularization penalty as follows:

$$\min_{\boldsymbol{w}^{(k)} \in \mathbb{R}^D} \|\boldsymbol{y}^{(k)} - \boldsymbol{X}^\top \boldsymbol{w}^{(k)}\|^2 + \lambda^{(k)} \|\boldsymbol{w}^{(k)}\|^2, \quad (1)$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n] \in \mathbb{R}^{D \times N}$ and $\boldsymbol{y}^{(k)} = [y_1^{(k)}, \dots, y_N^{(k)}]^\top \in \mathbb{R}^N$. The optimal solution of the prob-

lem in (1) is given by

$$
\begin{aligned}
\boldsymbol{w}^{(k)} &= (\boldsymbol{X}\boldsymbol{X}^\top + \lambda^{(k)}\boldsymbol{I})^{-1}\boldsymbol{X}\boldsymbol{y}^{(k)}, \\
&= \boldsymbol{U}(\boldsymbol{S} + \lambda^{(k)}\boldsymbol{I})^{-1}\boldsymbol{U}^\top\boldsymbol{X}\boldsymbol{y}^{(k)}, \quad (2)
\end{aligned}
$$

where $\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ is the eigen-decomposition of $\boldsymbol{X}\boldsymbol{X}^\top$ and $\boldsymbol{I}$ is an identity matrix. This regularized least squares approach has clear benefit that it requires only one eigen-decomposition of $\boldsymbol{X}\boldsymbol{X}^\top$ to obtain the solution in (2) for all combinations of $\lambda^{(k)}$ and $\boldsymbol{y}^{(k)}$.

3.3. Bayesian Evidence Framework

The optimization of the regularized least squares formulation presented in (1) is equivalent to the maximization of the posterior with fixed hyperparamters α and β denoted by $p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}, \alpha, \beta)$, where $\lambda = \alpha/\beta$. The posterior can be decomposed into two terms by Bayesian theorem as

$$
p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}, \alpha, \beta) \propto p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\alpha), \quad (3)
$$

where $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \beta)$ corresponds to Gaussian observation noise model given by

$$
p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(y_n|\boldsymbol{x}_n^\top\boldsymbol{w}, \beta^{-1}) \quad (4)
$$

and $p(\boldsymbol{w}|\alpha)$ denotes a zero-mean isotropic Gaussian prior as

$$
p(\boldsymbol{w}|\alpha) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \alpha^{-1}\boldsymbol{I}). \quad (5)
$$

Note that we dropped superscript (k) for notational simplicity from the equations in this subsection.

In the Bayesian evidence framework [16, 29], the evidence, also known as marginal likelihood, is a function of hyperparameters α and β as

$$
p(\boldsymbol{y}|\boldsymbol{X}, \alpha, \beta) = \int p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\alpha)d\boldsymbol{w}. \quad (6)
$$

Under the probabilistic model assumptions corresponding to (4) and (5), the log evidence $\mathcal{L}(\alpha, \beta)$ is given by

$$
\begin{aligned}
\mathcal{L}(\alpha, \beta) &\equiv \log p(\boldsymbol{y}|\boldsymbol{X}, \alpha, \beta) \quad (7) \\
&= \frac{D}{2}\log\alpha + \frac{N}{2}\log\beta - \frac{1}{2}\log|\boldsymbol{A}| \\
&\quad - \frac{\beta}{2}\|\boldsymbol{y} - \boldsymbol{X}^\top\boldsymbol{m}\|^2 - \frac{\alpha}{2}\boldsymbol{m}^\top\boldsymbol{m} - \frac{N}{2}\log 2\pi,
\end{aligned}
$$

where the precision matrix and mean vector of the posterior $p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}, \alpha, \beta) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}, \boldsymbol{A}^{-1})$ are given respectively by

$$
\boldsymbol{A} = \alpha\boldsymbol{I} + \beta\boldsymbol{X}\boldsymbol{X}^\top \quad \text{and} \quad \boldsymbol{m} = \beta\boldsymbol{A}^{-1}\boldsymbol{X}\boldsymbol{y}.
$$

The log evidence $\mathcal{L}(\alpha, \beta)$ is maximized by repeatedly alternating the following fixed point update rules

$$
\alpha = \frac{\gamma}{\boldsymbol{m}^\top\boldsymbol{m}} \quad \text{and} \quad \beta = \frac{N - \gamma}{\|\boldsymbol{y} - \boldsymbol{X}^\top\boldsymbol{m}\|^2}, \quad (8)
$$

which involves the derivation of γ as

$$
\gamma = \sum_{d=1}^{D}\frac{\beta s_d}{\alpha + \beta s_d} = \sum_{d=1}^{D}\frac{s_d}{\lambda + s_d}, \quad (9)
$$

where $\{s_d\}_{d=1}^{D}$ are eigenvalues of $\boldsymbol{X}\boldsymbol{X}^\top$. Note that $\boldsymbol{m}$ and γ should be re-estimated after each update of α and β.

Another pair of update rules of α and β are derived by an expectation-maximization (EM) technique as

$$
\alpha = \frac{D}{\boldsymbol{m}^\top\boldsymbol{m} + \mathrm{Tr}(\boldsymbol{A}^{-1})} \quad \text{and} \quad (10)
$$

$$
\beta = \frac{N}{\|\boldsymbol{y} - \boldsymbol{X}^\top\boldsymbol{m}\|^2 + \mathrm{Tr}(\boldsymbol{A}^{-1}\boldsymbol{X}\boldsymbol{X}^\top)}, \quad (11)
$$

but these procedures are substantially slower than the fixed point update rules in (8).

Through the optimization procedures described above, we determine the regularization parameter $\lambda = \alpha/\beta$. Although the estimated parameters are not optimal, they may still be reasonable solutions since they are obtained by maximizing marginal likelihood in (6).

3.4. Model Selection using Evidence

The evidence computed in the previous subsection is for a single class, and the overall evidence for entire classes, denoted by $\mathcal{L}^*$, is obtained by the summation of the evidences from individual classes, which is given by

$$
\mathcal{L}^* = \sum_{k=1}^{K}\mathcal{L}(\alpha^{(k)}, \beta^{(k)}). \quad (12)
$$

We compute the overall evidence corresponding to each deep CNN model, and choose the model with the maximum evidence for transfer learning. We expect that the selected model performs best among all candidates, which will be verified in our experiment.

In addition, when an ensemble of deep CNNs needs to be constructed for a target task, our approach selects a subset of good pre-trained CNNs in a greedy manner. Specifically, we add a network with the largest evidence in each stage and test whether the augmented network improves the evidence or not. The network is accepted if the evidence increases, or rejected otherwise. After the last candidate is tested, we obtain the final network combination and its associated model learned with the concatenated feature descriptors from accepted networks.

4. Fast Bayesian LS-SVM

Bayesian evidence framework discussed in Section 3 is useful to identify a good CNN for transfer learning and a reasonable regularization parameter. To make this framework even more practical, we present a faster algorithm to accomplish the same goal and a new theory that guarantees the converges of the algorithm.

4.1. Reformulation of Evidence

We are going to reduce $\mathcal{L}(\alpha, \beta)$ to a function with only one parameter that directly corresponds to the regularization parameter $\lambda = \alpha/\beta$. To this end, we re-write $\mathcal{L}(\alpha, \beta)$ by using the eigen-decomposition $\boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ as

$$\mathcal{L}(\alpha, \beta) = \frac{D}{2}\log\alpha + \frac{N}{2}\log\beta - \frac{1}{2}\sum_{d=1}^{D}\log(\alpha + \beta s_d)$$

$$-\frac{\beta}{2}\boldsymbol{y}^\top\boldsymbol{y} + \frac{\beta^2}{2}\sum_{d=1}^{D}\frac{h_d^2}{\alpha + \beta s_d} - \frac{N}{2}\log 2\pi, \quad (13)$$

where s_d is the d-th diagonal element in $\boldsymbol{S}$ and h_d denotes the d-th element in $\boldsymbol{h} = \boldsymbol{U}^\top\boldsymbol{X}\boldsymbol{y}$. Then, we re-parameterize $\mathcal{L}(\alpha, \beta)$ into $\mathcal{F}(\lambda, \beta)$ as

$$\mathcal{F}(\lambda, \beta) = \frac{D}{2}\log\lambda + \frac{N}{2}\log\beta - \frac{1}{2}\sum_{d=1}^{D}\log(\lambda + s_d)$$

$$-\frac{\beta}{2}\left(\boldsymbol{y}^\top\boldsymbol{y} - \sum_{d=1}^{D}\frac{h_d^2}{\lambda + s_d}\right) - \frac{N}{2}\log 2\pi. \quad (14)$$

The derivative of $\mathcal{F}(\lambda, \beta)$ with respect to β is given by

$$\frac{\partial\mathcal{F}}{\partial\beta} = \frac{N}{2\beta} - \frac{1}{2}\left(\boldsymbol{y}^\top\boldsymbol{y} - \sum_{d=1}^{D}\frac{h_d^2}{\lambda + s_d}\right),$$

and we obtain the following equation by setting this derivative to zero,

$$\beta = \frac{N}{\boldsymbol{y}^\top\boldsymbol{y} - \sum_{d=1}^{D}\frac{h_d^2}{\lambda + s_d}}. \quad (15)$$

Finally, we obtain a one-dimensional function of the log evidence by plugging (15) into (14), which is given by

$$\mathcal{F}(\lambda) = \frac{1}{2}\sum_{d=1}^{D}\log\frac{\lambda}{\lambda + s_d} + \frac{N}{2}\log N - \frac{N}{2} - \frac{N}{2}\log 2\pi$$

$$-\frac{N}{2}\log\left(\boldsymbol{y}^\top\boldsymbol{y} - \sum_{d=1}^{D}\frac{h_d^2}{\lambda + s_d}\right). \quad (16)$$

Figure 2 illustrates the curvature of this log evidence function with respect to $\log\lambda$.

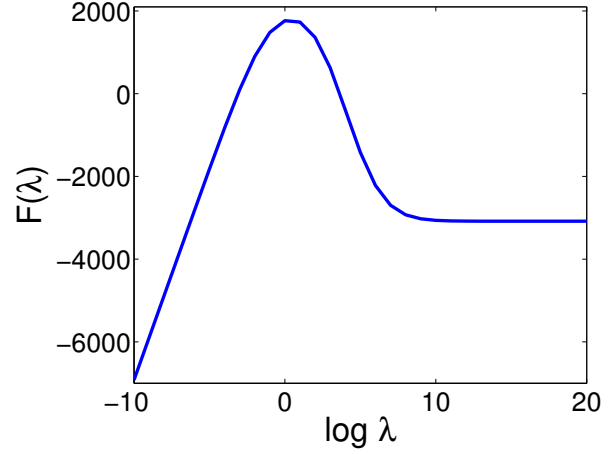Figure 2. Plot of the log evidence $\mathcal{F}(\lambda)$ with respect to $\log\lambda$. Note that $\mathcal{F}(\lambda)$ is neither convex nor concave.

4.2. New Fixed-point Update Rule

We now derive a new fixed point update rule and present the sufficient condition for the existence of a fixed point. The stationary points in (16) with respect to λ satisfy

$$\frac{1}{2}\sum_{d=1}^{D}\frac{s_d}{\lambda(\lambda + s_d)} - \frac{N}{2}\frac{\sum_{d=1}^{D}\frac{h_d^2}{(\lambda + s_d)^2}}{\boldsymbol{y}^\top\boldsymbol{y} - \sum_{d=1}^{D}\frac{h_d^2}{\lambda + s_d}} = 0, \quad (17)$$

and we update the fixed-point by maximizing (16) as

$$\lambda = \frac{\sum_{d=1}^{D}\frac{s_d}{\lambda + s_d}}{\left(\frac{N}{\boldsymbol{y}^\top\boldsymbol{y} - \sum_{d=1}^{D}h_d^2/(\lambda + s_d)}\right)\left(\sum_{d=1}^{D}\frac{h_d^2}{(\lambda + s_d)^2}\right)}. \quad (18)$$

As illustrated in Figure 2, $\mathcal{F}(\lambda)$ in (16) is neither convex nor concave as illustrated in the supplementary file. However, we can show the sufficient condition of the existence of the fixed point using the following theorem.

Theorem 1. *Denote the update rule in* (18) *by* $f(\lambda)$. *If* $\boldsymbol{y}$ *is a binary variable and* $\boldsymbol{x}_n$ *is an* L_2 *normalized nonnegative vector, then* $f(\lambda)$ *has a fixed point.*

Proof. We first show that $f(\lambda)$ is asymptotically linear in λ as

$$\lim_{\lambda\to\infty}\frac{f(\lambda)}{\lambda} = \lim_{\lambda\to\infty}\frac{\left(\boldsymbol{y}^\top\boldsymbol{y} - \sum_d\frac{h_d^2}{\lambda + s_d}\right)\sum_{d=1}^{D}\frac{s_d}{\lambda + s_d}}{\lambda N\sum_{d=1}^{D}\frac{h_d^2}{(\lambda + s_d)^2}}$$

$$= \frac{\boldsymbol{y}^\top\boldsymbol{y}\sum_{d=1}^{D}s_d}{N\sum_{d=1}^{D}h_d^2} = \frac{\|\boldsymbol{y}\|^2\|\boldsymbol{X}\|_F^2}{N\|\boldsymbol{X}\boldsymbol{y}\|^2}.$$

Since $\boldsymbol{y}$ is binary and $\boldsymbol{x}_n$ is L_2 normalized and nonnegative, we can derive the following two relations,

$$\|\boldsymbol{y}\|^2\|\boldsymbol{X}\|_F^2 = PN \text{ and} \quad (19)$$

$$\|\boldsymbol{X}\boldsymbol{y}\|^2 = \left(\sum_{n:y_n=1}x_n\right)^2 > \sum_{n:y_n=1}x_n^2 = P, \quad (20)$$
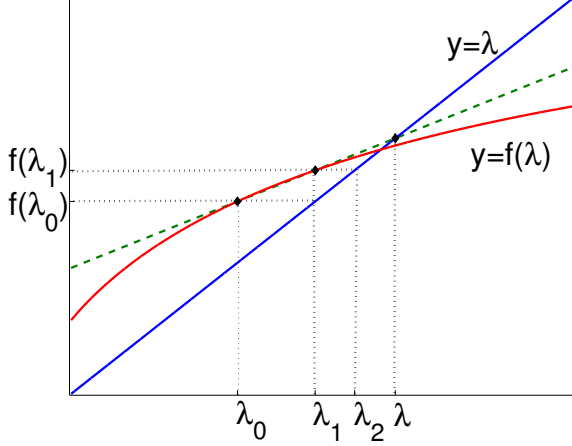
Figure 3. Aitken's delta-squared process. The fixed point update function $f(\lambda)$ is approximated by green dashed line and its intersection with $y = \lambda$ becomes the next update point.

Algorithm 1 Fast Bayesian Least Squares

Input: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ and $\boldsymbol{y} \in \mathbb{R}^{N}$.
Output: Optimal solutions $(\boldsymbol{w}, \lambda)$.
Initialize λ // e.g., $\lambda = 1$
$(\boldsymbol{U}, \boldsymbol{S}) \leftarrow$ eigen-decomposition$(\boldsymbol{X}\boldsymbol{X}^{\top})$
$\boldsymbol{s} \leftarrow \text{diag}(\boldsymbol{S}), \;\; \boldsymbol{h} \leftarrow \boldsymbol{U}^{\top}\boldsymbol{X}\boldsymbol{y}$
repeat
   $\lambda_0 \leftarrow \lambda$
   $\lambda_1 \leftarrow$ UPDATE $(\lambda_0, \boldsymbol{s}, \boldsymbol{h}, N, \boldsymbol{y}^{\top}\boldsymbol{y})$
   $\lambda_2 \leftarrow$ UPDATE $(\lambda_1, \boldsymbol{s}, \boldsymbol{h}, N, \boldsymbol{y}^{\top}\boldsymbol{y})$
   $\lambda \leftarrow \lambda_0 - \frac{(\lambda_1 - \lambda_0)^2}{(\lambda_2 - \lambda_1) - (\lambda_1 - \lambda_0)}$
   if $\lambda < 0$ or $\lambda = \pm\infty$ **then**
     $\lambda \leftarrow \lambda_2$
   end if
until $|\lambda - \lambda_0| < \epsilon$ // e.g., $\epsilon = 10^{-5}$
$\boldsymbol{w} \leftarrow \boldsymbol{U}(\boldsymbol{S} + \lambda\boldsymbol{I})^{-1}\boldsymbol{h}$

---

where $P = \sum_{n=1}^{N} y_n$. From (19) and (20), it is shown that $\|\boldsymbol{y}\|^2 \|\boldsymbol{X}\|_F^2 < N\|\boldsymbol{X}\boldsymbol{y}\|^2$.

Obviously, $f(0) > 0$ and there exists a λ^+ such that $f(\lambda^+) < \lambda^+$. The intermediate value theorem implies the existence of λ^* such that $f(\lambda^*) = \lambda^*$, where $0 < \lambda^* < \lambda^+$ as illustrated in Figure 3. $\square$

The fixed point is unique if $f(\lambda)$ is concave. Although it is always concave according to our observation, we have no proof yet and leave it as a future work

4.3. Speed Up Algorithm

We accelerate the fixed point update rule in (18) by using Aitken's delta-squared process [1]. Figure 3 illustrates the Aitken's delta-squared process. Let's focus on the two points $(\lambda_0, f(\lambda_0))$ and $(\lambda_1, f(\lambda_1))$, and line going through

---

Algorithm 2 $\lambda =$ UPDATE$(\lambda, \boldsymbol{s}, \boldsymbol{h}, N, \boldsymbol{y}^{\top}\boldsymbol{y})$

$\gamma \leftarrow \sum_{d=1}^{D} \frac{s_d}{\lambda + s_d}$
$\beta \leftarrow N / (\boldsymbol{y}^{\top}\boldsymbol{y} - \sum_{d=1}^{D} \frac{h_d^2}{\lambda + s_d})$
$\boldsymbol{m}^{\top}\boldsymbol{m} \leftarrow \sum_{d=1}^{D} \frac{h_d^2}{(\lambda + s_d)^2}$
$\lambda \leftarrow \frac{\gamma}{\beta \, \boldsymbol{m}^{\top}\boldsymbol{m}}$
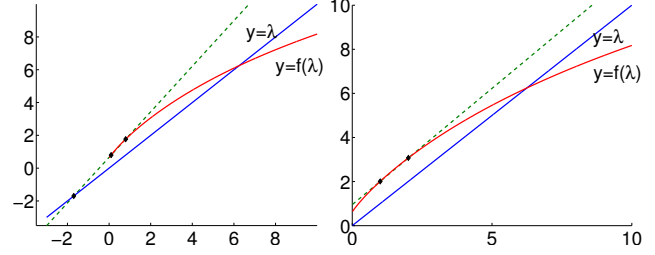**return** λ

Figure 4. Two failure cases of Aitken's delta-squared process. (left) The first case arises if initial λ_0 is far from the fixed point $\lambda^\star$, which results in $\lambda < 0$. (right) The second case occurs when approximating line (dashed green) is parallel to $y = \lambda$, where $\lambda = \pm\infty$.

these two points. The equation of this line is

$$y = \lambda_1 + (\lambda - \lambda_0)\frac{\lambda_2 - \lambda_1}{\lambda_1 - \lambda_0}, \tag{21}$$

where $f(\lambda_0)$ and $f(\lambda_1)$ are replaced by λ_1 and λ_2, respectively. The idea behind Aitken's method is to approximate fixed point λ^* using the intersection of the line in (21) with line $y = \lambda$, which is given by

$$\lambda = \lambda_0 - \frac{(\lambda_1 - \lambda_0)^2}{(\lambda_2 - \lambda_1) - (\lambda_1 - \lambda_0)}. \tag{22}$$

Our fast Bayesian learning algorithm for the regularized least squares problem in (1) is summarized in Algorithm 1. In our algorithm, we first compute the eigen-decomposition of $\boldsymbol{X}\boldsymbol{X}^{\top}$. This is the most time consuming part but needs to be performed only once since the result can be reused for every label in $\boldsymbol{y}$. After that, we obtain the regularization parameter λ through an iterative procedure.

When we apply the Aitken's delta-squared process, we have two potential failure cases as in Figure 4(a) and 4(b). The first case often arises if the initial λ_0 is far from the fixed point λ^*, and the second case occurs when the approximating line in (21) is parallel to $y = \lambda$. Fortunately, these failures rarely happen in practice and can be handled easily by skipping the procedure in (22) and updating λ with λ_2.

Figure 5 demonstrates the relative convergence rates of three different techniques—Aitken's delta-squared process in Algorithm 1, fixed point update rules in (8), and EM update method, where the Aitken's delta-squared process is significantly faster than others for convergence.
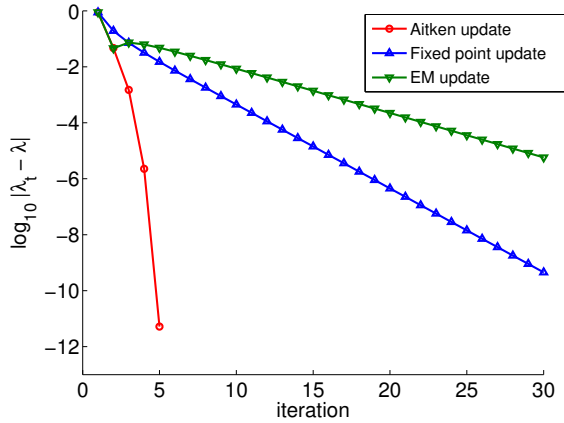
Figure 5. Comparison between Aitken's delta-squared process, fixed point update rules, and EM update rules on PASCAL VOC 2012 dataset (class = *aeroplane*). Aitken's delta-squared process significantly faster than other methods.

5. Experiments

We present the details of our experiment setting and the performance of our algorithm compared to the state-of-the-art techniques in 12 visual recognition benchmark datasets.

5.1. Datasets and Image Representation

The benchmark datasets involve various visual recognition tasks such as object recognition, photo annotation, scene recognition, fine grained recognition, visual attribute detection, and action recognition. Table 1 presents the characteristics of the datasets. In our experiment, we followed the given train and test split and evaluation measure of each dataset. For the datasets with bounding box annotations such as CUB200-2011, UIUC object attribute, Human attribute, and Stanford 40 actions, we enlarged the bounding boxes by 150% to consider neighborhood context as suggested in [23, 2].

For deep learning representations, we selected 4 pre-trained CNNs from the Caffe Model Zoo: *GoogLeNet* [31], *VGG19* [25], and *AlexNet* [7] trained on ImageNet, and *GoogLeNet* trained on Places [31]. As generic image representations, we used the 4096 dimensional activations of the first fully connected layer in VGG19 and AlexNet and the 1024 dimensional vector obtained from the global average pooling layer located right before the final softmax layer in GoogLeNet.

Our implementation is in Matlab2011a, and all experiments were conducted on a quad-core Intel(R) core(TM) i7-3820 @ 3.60GHz processor.

5.2. Bayesian LS-SVM vs. SVM

We first compare the performance of our Bayesian LS-SVM with the standard SVM when they are applied to deep

CNN features for visual recognition problems. We used only a single image scale 256×256 in this experiment. LI-BLINEAR [10] package is used for SVM training and the regularization parameters are selected by grid search with cross validations.

Table 2 presents the complete results of our experiment. Bayesian LS-SVM is competitive to SVM in terms of prediction accuracy even with significantly reduced training time. Training SVM is getting slower than Bayesian LS-SVM as the number of classes increases so it is particularly slow in Caltech 256 and SUN 397 datasets.

Another notable observation in Table 2 is that the order of prediction accuracy is highly correlated to the evidence. This means that the selected model by Bayesian LS-SVM produces reliable testing accuracy and a proper deep learning image representation is obtained without time consuming grid search and cross validation. Note that cross validations in LS-SVM and SVM play the same role, but are less reliable and slower than our Bayesian evidence framework. The capability to select the appropriate CNN model and the corresponding regularization parameter is one of the most important properties of our algorithm.

5.3. Comparison with Other Methods

We now show that our Bayesian LS-SVM identifies a combination of multiple CNNs to improve accuracy without grid search and cross validation. For each task, we select a subset of 4 pre-trained CNNs in a greedy manner; we add CNNs to our selection, one by one, until the evidence does not increase. Our algorithm is compared with DeCAF [7], Zeiler [34], INRIA [21], KTH-S [23], KTH-FT [2], VGG [25], Zhang [35, 36], and TUBFI [3]. In addition, our ensembles identified by greedy evidence maximization are compared with the oracle combinations—the ones with the highest accuracy in test set found by exhaustive search—and the best combinations found by exhaustive evidence maximization.

Table 3 presents that our ensembles approach achieves the best performance in most of the 12 tasks. The identified ensembles by the greedy approach are consistent with the selections by exhaustive evidence maximization and even oracle selections[1] made by testing accuracy maximization. Note that our network selections are natural and reasonable; GoogLeNet-ImageNet and VGG19 are selected frequently while GoogLeNet-Place is preferred to GoogLeNet-ImageNet in MIT Indoor and SUN-397 since the datasets are constructed for scene recognition. It turns out that the proposed algorithm tends to choose the networks with higher accuracies in the target task even though it makes selections based only on the evidence in a greedy manner. An interesting observation is that our result is less

---

[1]This option is practically impossible since it requires evaluation with test dataset using all available models for model selection.

Table 1. Characteristics of the 12 datasets. N_1: number of training data, N_2: number of test data, K: number of classes, L: average number of labels per image, AP: average precision, Acc.: accuracy, AUC: area under the ROC curve.

Dataset	Task	N_1	N_2	K	L	Box	Measure
PASCAL VOC 2007 [8]	object recognition	5011	4952	20	1.5		mean AP
PASCAL VOC 2012 [9]	object recognition	5717	5823	20	1.5		mean AP
Caltech 101 [12]	object recognition	3060	6086	102	1		mean Acc.
Caltech 256 [14]	object recognition	15420	15187	257	1		mean Acc.
ImageCLEF 2011 [20]	photo annotation	8000	10000	99	11.9		mean AP
MIT Indoor Scene [22]	scene recognition	5360	1340	67	1		mean Acc.
SUN 397 Scene [32]	scene recognition	19850	19850	397	1		mean Acc.
CUB 200-2011 [30]	fine-grained recognition	5994	5794	200	1	✓	mean Acc.
Oxford Flowers [18]	fine-grained recognition	2040	6149	200	1		mean Acc.
UIUC object attributes [11]	attribute detection	6340	8999	64	7.1	✓	mean AUC
Human attributes [5]	attribute detection	4013	4022	9	1.8	✓	mean AP
Stanford 40 actions [33]	action recognition	4000	5532	40	1	✓	mean AP

Table 2. Comparisons between Bayesian LS-SVM and SVM. Without time consuming cross validation procedure, Bayesian LS-SVM achieves prediction accuracy competitive to SVM. In addition, Bayesian LS-SVM selects the proper CNN for each task by using the evidence, which is denoted by bold-faced numbers. Best accuracy in LS-SVM and SVM denotes the maximum achievable accuracy in test dataset using all available learned models. Note that the selected model by Bayesian evidence framework or cross validation may not be the best one in testing. The following sets of regularization parameters are tested for cross validation in LS-SVM and SVM, respectively: $\{2^{-10}, 2^{-9}, \ldots, 1, \ldots, 2^9, 2^{10}\}$ and $\{0.01, 0.05, 0.1, 0.5, 1, 2, 5, 10\}$. (G$_\mathrm{I}$: *GoogLeNet-ImageNet*, G$_\mathrm{P}$: *GoogLeNet-Place*, V: *VGG19*, and A: *AlexNet*)

	LS-SVM				CV (5 folds)		SVM	CV (5 folds)	
CNN	Best	Acc.	Bayesian Evidence	Time	Acc.	Time	Best	Acc.	Time
PASCAL VOC 2007 [8]									
G$_\mathrm{I}$	85.3	85.2	46.9×10^3	1.1	85.2	8.4	85.0	84.7	122.4
G$_\mathrm{P}$	74.1	73.8	38.6×10^3	1.0	74.0	8.1	74.1	73.9	144.3
V	85.9	**85.8**	$\mathbf{48.0 \times 10^3}$	41.9	85.8	172.2	85.9	85.8	257.5
A	75.2	75.0	32.5×10^3	41.7	75.0	160.4	75.3	75.2	211.1
PASCAL VOC 2012 [9]									
G$_\mathrm{I}$	84.4	84.3	51.3×10^3	1.2	84.3	8.6	83.9	83.7	140.8
G$_\mathrm{P}$	73.2	72.9	40.6×10^3	1.1	73.1	8.4	73.2	73.1	170.7
V	85.2	**85.1**	$\mathbf{52.9 \times 10^3}$	42.7	85.2	161.5	85.6	85.4	295.9
A	74.1	73.9	34.3×10^3	42.7	74.0	161.8	74.4	74.3	160.7
Caltech 101 [12]									
G$_\mathrm{I}$	90.6	90.0	37.8×10^4	1.0	89.6	6.0	91.4	85.1	325.0
G$_\mathrm{P}$	57.0	54.3	30.6×10^4	0.9	55.1	5.9	57.2	41.8	390.3
V	92.2	**92.1**	$\mathbf{40.9 \times 10^4}$	31.5	88.8	142.7	92.2	86.8	729.4
A	89.3	89.2	37.3×10^4	32.0	83.4	146.9	90.0	83.5	595.3
Caltech 256 [14]									
G$_\mathrm{I}$	77.8	77.2	59.9×10^5	2.3	77.8	21.8	81.2	81.2	4060.4
G$_\mathrm{P}$	44.9	42.6	55.9×10^5	2.2	44.9	21.2	48.6	48.6	4991.8
V	82.0	**81.1**	$\mathbf{62.3 \times 10^5}$	52.5	81.7	339.7	82.7	82.7	9653.1
A	69.7	68.9	58.6×10^5	52.9	69.2	336.9	72.3	72.3	5348.6
ImageCLEF [20]									
G$_\mathrm{I}$	49.1	48.9	20.5×10^4	1.5	48.8	37.0	47.7	47.4	1218.6
G$_\mathrm{P}$	47.5	47.1	20.8×10^4	1.4	47.1	36.9	47.1	46.7	1410.5
V	50.7	**50.3**	$\mathbf{21.3 \times 10^4}$	45.9	50.4	248.5	50.4	50.1	2531.2
A	44.8	44.6	18.7×10^4	46.1	44.6	245.9	44.4	44.1	2140.0
MIT Indoor [22]									
G$_\mathrm{I}$	66.7	66.0	30.1×10^4	1.2	66.7	5.8	69.4	69.2	400.9
G$_\mathrm{P}$	80.0	**79.9**	$\mathbf{35.2 \times 10^4}$	1.1	80.0	5.8	81.1	80.4	402.5
V	73.2	73.1	31.1×10^4	42.6	73.2	186.8	74.7	74.7	895.5
A	62.0	61.1	28.6×10^4	42.2	60.5	187.4	63.1	63.1	460.9

	LS-SVM				CV (5 folds)		SVM	CV (5 folds)	
CNN	Best	Acc.	Bayesian Evidence	Time	Acc.	Time	Best	Acc.	Time
SUN-397 [32]									
G$_\mathrm{I}$	48.1	47.0	12.8×10^6	3.1	48.1	36.5	54.2	54.2	8739.6
G$_\mathrm{P}$	61.1	**60.1**	$\mathbf{13.2 \times 10^6}$	2.9	61.1	34.4	63.3	63.3	8589.4
V	55.0	53.7	12.9×10^6	57.4	54.9	419.8	57.1	57.1	20254.0
A	45.4	44.9	12.7×10^6	50.8	45.4	419.0	48.6	48.6	10781.8
CUB-200 [30]									
G$_\mathrm{I}$	65.2	64.3	15.6×10^5	1.3	64.1	11.0	67.6	56.5	1201.9
G$_\mathrm{P}$	16.4	13.6	14.9×10^5	1.5	15.0	11.1	16.8	11.1	1664.6
V	69.2	**68.6**	$\mathbf{15.8 \times 10^5}$	44.1	61.5	259.2	71.1	59.4	2776.2
A	59.0	58.5	15.5×10^5	45.3	46.6	257.9	61.4	51.6	1645.5
Oxford Flowers [18]									
G$_\mathrm{I}$	85.5	84.7	21.8×10^4	0.9	82.0	5.5	87.4	72.0	198.8
G$_\mathrm{P}$	55.6	51.7	19.4×10^4	0.9	51.8	5.5	57.1	32.8	234.7
V	87.5	87.1	22.5×10^4	26.9	82.1	142.2	87.6	73.4	520.9
A	87.6	**87.6**	$\mathbf{22.9 \times 10^4}$	27.3	81.8	146.7	88.3	77.1	271.3
UIUC Attributes [11]									
G$_\mathrm{I}$	91.5	90.3	13.5×10^4	1.4	90.9	8.0	91.3	90.6	605.5
G$_\mathrm{P}$	87.8	86.6	10.5×10^4	1.3	87.1	7.4	88.0	87.6	726.0
V	92.5	**91.1**	$\mathbf{14.4 \times 10^4}$	43.8	92.0	186.3	92.2	91.7	1285.4
A	91.4	89.9	12.9×10^4	44.1	91.0	191.2	90.8	90.5	683.7
Human Attributes [5]									
G$_\mathrm{I}$	76.0	**75.8**	$\mathbf{-74.8 \times 10^2}$	1.0	75.8	5.0	74.2	74.1	70.6
G$_\mathrm{P}$	58.7	58.4	-103.1×10^2	1.0	58.0	4.8	56.9	56.5	85.5
V	75.4	75.1	-76.0×10^2	40.3	75.2	124.2	73.1	72.8	131.9
A	71.9	71.3	-84.4×10^2	40.7	71.7	121.2	70.0	69.9	63.3
Stanford 40 Action [33]									
G$_\mathrm{I}$	70.2	69.8	100.4×10^3	1.0	69.6	11.6	69.8	69.6	211.7
G$_\mathrm{P}$	48.3	47.6	86.5×10^3	1.1	47.9	11.4	48.2	47.7	246.2
V	75.4	**75.2**	$\mathbf{109.3 \times 10^3}$	41.1	75.1	142.9	75.8	75.3	418.7
A	58.0	57.7	89.6×10^3	41.5	57.5	156.5	57.4	57.1	206.8

consistent with the selections by oracle and exhaustive evidence maximization in Stanford 40 Actions dataset, where GoogLeNet-Place seems to provide complementary infor-

mation even with its low accuracy and is helpful to improve recognition performance. It is probably because actions are frequently performed at typical places, *e.g.*, a fair portion of

Table 3. Comparison to existing methods in the 12 benchmark datasets. The best ensembles identified by maximizing evidence through exhaustive search mostly coincide with the oracle combinations—the ones with the highest accuracy in test set, which is also found by exhaustive search. The ensembles identified by our greedy search are very similar to the ones by these exhaustive search methods, and our algorithm consequently performs best in many tested datasets. We used three scales $\{256, 384, 512\}$ as done in [25], where we simply averaged the prediction scores from three scales.

Method	VOC07	VOC12	CAL101	CAL256	CLEF	MIT	SUN	Birds	Flowers	UIUC	Human	Action
DeCAF	-	-	86.9	-	-	-	38.0	65.0	-	-	-	-
Zeiler	-	79.0	86.5	74.2	-	-	-	-	-	-	-	-
INRIA	77.7	82.8	-	-	-	-	-	-	-	-	-	-
KTH-S	71.8	-	-	-	-	64.9	49.6	62.8	90.5	90.6	73.8	58.9
KTH-FT	80.7		-	–	-	71.3	56.0	67.1	91.3	91.5	74.6	66.4
VGG	89.7	89.3	92.7	**86.2**	-	-	-	-	-	-	-	-
Zhang	-	-	-	-	-	-	-	76.4	-	-	79.0	-
TUBFI	-	-	-	-	44.3	-	-	-	-	-	-	-
G_I	87.5	86.2	90.5	77.7	50.3	71.3	48.3	64.7	88.1	91.1	78.4	71.0
G_P	75.7	74.9	53.8	42.1	48.1	80.8	59.8	14.9	57.8	87.3	59.7	48.4
V	88.4	87.8	93.3	83.3	52.4	77.8	56.1	69.9	91.5	91.8	79.1	77.0
A	75.0	73.9	88.3	69.7	52.3	77.5	42.4	60.7	86.7	89.9	71.3	57.7
Oracle (exhaustive)	$G_I G_P V$ 90.0	$G_I G_P V$ 89.4	$G_I V A$ 95.3	$G_I G_P V A$ 86.1	$G_I G_P V A$ 55.7	$G_P V A$ 84.9	$G_I G_P V A$ 67.5	$G_I V A$ 77.3	$G_I G_P V A$ 94.7	$G_I V A$ 92.0	$G_I V A$ 80.8	$G_I G_P V$ 78.6
Max evid. (exhaustive)	$G_I G_P V$ 90.0	$G_I G_P V$ 89.4	$G_I V A$ 95.3	$G_I G_P V A$ 86.1	$G_I G_P V$ 55.5	$G_P V$ 84.7	$G_P V$ 67.5	$G_I V A$ 77.3	$G_I V A$ 94.5	$G_I G_P V A$ 92.0	$G_I V A$ 80.8	$G_I G_P V$ 78.6
Ours (greedy)	$G_I G_P V$ **90.0**	$G_I G_P V$ **89.4**	$G_I V A$ **95.3**	$G_I G_P V A$ 86.1	$G_I G_P V$ **55.5**	$G_P V$ **84.7**	$G_P V$ **67.5**	$G_I V A$ **77.3**	$G_I V A$ **94.5**	$G_I G_P V A$ **92.0**	$G_I V A$ **80.8**	$G_I V A$ **77.8**

images in *brushing teeth* class are taken from bathrooms.

6. Conclusion

We described a simple and efficient technique to transfer deep CNN models pre-trained on specific image classification tasks to another tasks. Our approach is based on Bayesian LS-SVM, which combines Bayesian evidence framework and SVM with a least squares loss. In addition, we presented a faster fixed point update rule for evidence maximization through Aitken's delta-squared process. Our fast Bayesian LS-SVM demonstrated competitive results compared to the standard SVM by selecting a deep CNN model in 12 popular visual recognition problems. We also achieved the state-of-the-art performance by identifying a good ensemble of the candidate models through our Bayesian LS-SVM framework.

Acknowledgements

References

[1] A. C. Aitken. On Bernoulli's numerical solution of algebraic equations. *Proceedings of the Royal Society of Edinburgh*, 46:289–305, 1927. 2, 5

[2] H. Azizpour, A. S. Razavian, J. Sulivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. In *CVPR Workshops*, 2015. 1, 2, 6

[3] A. Binder, W. Samek, M. Kloft, C. Müller, K.-R. Müller, and M. Kawanabe. The joint submission of the TU Berlin and Fraunhofer FIRST (TUBFI) to the ImageCLEF2011 photo annotation task, 2011. 6

[4] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon press Oxford, 1995. 2

[5] L. D. Bourdev, S. Maji, and J. Malik. Describing people: A poselet-based approach to attribute classification. In *ICCV*, 2011. 7

[6] K. Chatfield, K. Simonyan, A. Vedaldi, and a. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 1, 2

[7] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: a deep convolutional activation feature for generic visual recognition. In *ICML*, 2014. 1, 2, 6

[8] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) Results, 2007. 7

[9] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC 2012) Results, 2012. 7

[10] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classification. *JMLR*, 9:1871–1874, 2008. 6

[11] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 7

[12] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVIU*, 106(1):59–70, 2007. 7

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2

[14] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007. 7

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification wit deep convolutional neural networks. In *NIPS*, volume 25, 2012. 2

[16] D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992. 1, 3

[17] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016. 1

[18] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 7

[19] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 1

[20] S. Nowak, K. Nagel, and J. Liebetrau. The CLEF 2011 photo annotation and concept-based retrieval tasks. In *CLEF Workshop Notebook Paper*, 2011. 7

[21] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. 1, 2, 6

[22] A. Quattoni and A. Torrabla. Recognizing indoor scenes. In *CVPR*, 2009. 7

[23] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *CVPR Workshops*, 2014. 1, 2, 6

[24] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. 2

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2, 6, 8

[26] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999. 1

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, D. A. S. Reed, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2

[28] T. Van Gestel, J. A. K. S. B. Baesems, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, and J. Vandewalle. Benchmarking least squares support vector machines classifiers. *Machine Learning*, 54(1):5–32, 2004. 2

[29] T. Van Gestel, J. A. K. Suykens, G. Lanckrie, A. Lambrechts, B. D. Moor, and J. Vandewalle. Bayesian framework for least-squares support vector machine classifiers, gaussian processes, and kernel fisher discriminant analysis. *Neural Computation*, 14(5):1115–1147, 2002. 1, 3

[30] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical report, California Institute of Technology, 2011. 7

[31] Z. Wu, Y. Zhang, F. Yu, and J. Xiao. A GPU implementation of GoogLeNet. Technical report, Princeton University, 2014. 6

[32] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torrabla. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 7

[33] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. J. Guibas, and L. Fei-Fei. Action recognition by learning bases of action attributes and parts. In *ICCV*, 2011. 7

[34] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2, 6

[35] N. Zhang, , M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. PANDA: Pose aligned networks for deep attribute modeling. In *CVPR*, 2014. 6

[36] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based R-CNNs for fine-grained category detection. In *ECCV*, 2014. 1, 2, 6

[37] P. Zhang and J. Peng. SVM vs regularized least squares classification. In *ICPR*, 2004. 2