Transformation-Invariant Convolutional Jungles

Dmitry Laptev, Joachim M. Buhmann
Department of Computer Science, ETH Zurich, Switzerland

Many Computer Vision problems arise from information processing of data sources with nuisance variances like scale, orientation, contrast, perspective foreshortening or – in medical imaging – staining and local warping. In most cases these variances can be stated a priori and can be used to improve the generalization of recognition algorithms. We propose a novel supervised feature learning approach, which efficiently extracts information from these constraints to produce interpretable, transformation-invariant features. The proposed method can incorporate a large class of transformations, e.g., shifts, rotations, change of scale, morphological operations, non-linear distortions, photometric transformations, etc. These features boost the discrimination power of a novel image classification and segmentation method, which we call Transformation-Invariant Convolutional Jungles (TICJ).

The feature of an image X with associated discrete label y is defined through a set of a priori known transformations $\Phi = \{\phi_1, \dots, \phi_T\}$, where ϕ_t denotes a transformation function and T specifies the number of transformations considered. The results of different simple transformations $\phi(X)$ are shown in figure 1, however Φ can also contain any combination of these transformations. We parametrize a feature with a convolutional kernel θ. The value of the feature for an image X is given by:

$$f_\theta(x) = \max_{\phi \in \Phi} \theta^T \phi(x) \qquad (1)$$

Because of the maximum, inspired by max-pooling operation in Neural Networks [2], this equation in most cases gives exactly the same result $f_\theta(x)$ for the image X itself, and for the transformations of this image $\phi(X)$. Lemma 1 formulates the conditions on the set Φ for which this holds true.

Lemma 1. *The feature of the image X defined in equation 1 is transformation-invariant if the set Φ of all possible transformations forms a group, i.e. satisfies the axioms of closure, associativity, invertibility and identity.*

To learn the feature parameter vector θ, we select two classes c_1, c_2 and solve the following optimization problem:

$$\theta = \arg\min_{\theta} \sum_{i:\ y_i = c_1 \text{ or } y_i = c_2} \left(f_\theta(X_i) + [y_i = c_1] - [y_i = c_2]\right)^2 + \lambda ||\Gamma\theta||_2^2 \qquad (2)$$

Here $[\cdot]$ refers to Iverson brackets, that are equal to 1 if $\cdot$ is true and zero otherwise. Matrix Γ is a matrix of a 2D differentiation operator in a vectorized space, that is a Tikhonov regularization matrix. Penalizing the gradient of the kernel enforces the kernel to be smooth and ensures interpretability of the inferred kernels (see figure 2). λ is a regularization parameter that controls the trade-off between the goodness of separation and the smoothness of the kernel learned.

Parameter vector θ learned in this manner results in a transformation-invariant feature that splits the dataset into two subsets: one subset consists of the images $X_i : f_\theta(X_i) > 0$, another of images $X_i : f_\theta(X_i) \leq 0$.

That means that a feature defines a split predicate on the space of images, and therefore can be used in algorithms such as decision trees: recursively learning new features, splitting the dataset in two parts and dividing the space until required granularity is achieved. We call this algorithm TICT (Transformation-Invariant Decision Trees). Because the split takes the linear combination of all the pixels into account, the proposed algorithm is similar to convolutional decision trees [1]. However, the features in our case are non-linear, and therefore TICT does not belong to this category.

The major problem with TICT is that the tree size grows exponentially with its depth, resulting in overfitting. Therefore, as the final algorithm we use a modification of it inspired by Decision Jungles [3].

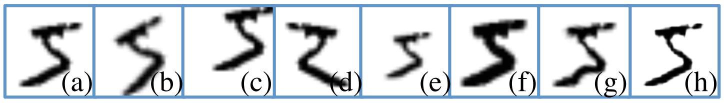

Figure 1: Example of transformations $\phi(X)$: identity transformation (a), rotation (b), translation (c), reflection (d), scaling (e), morphological operations (f), non-linear distortions (g), brightness and contrast change (h).

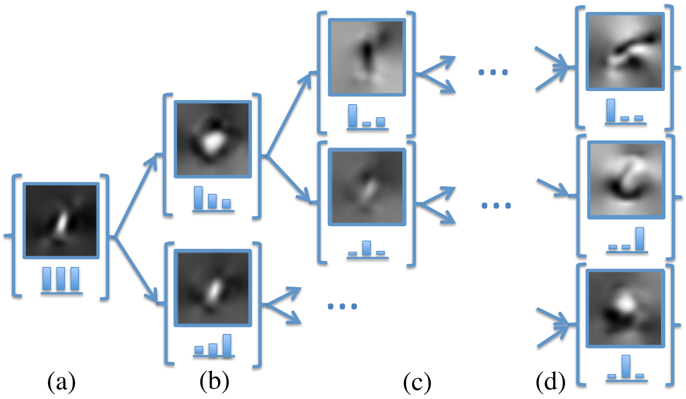

Figure 2: A visualization of TICJ training process. Each node is represented with feature parameters θ and a histogram h of input object classes (for simplicity we consider three classes here). (a) shows the root node, for which the whole dataset is an input. Using the learned feature f_θ – the dataset is split in two subsets to serve as input for two other nodes (b). The algorithm proceeds by splitting the dataset until the maximum width is achieved (c). Then some of the data subsets can be joined together with a histogram clustering technique (d). One can say that the resulting feature parameters θ are interpretable, e.g. looking like edge and curvature detectors.

TICJ (Transformation-Invariant Decision Jungles) overcome the issues of TICT by limiting the tree width and therefore space granularity. The idea of TICJ is very simple: after adding one layer, we perform the clustering of leaves and join similar leaves together where the similarity of leaves is measured as the similarity of the histograms of the classes present in a leaf (see figure 2).

We test the proposed approach on two very different datasets: (i) Yale face recognition dataset, that is very small (15 classes, 5 images per class for training), and (ii) Neuronal structures segmentation dataset (contains tens of thousands of samples for each of two classes). In both datasets we achieve state of the art results. On the Yale dataset we outperform the competitors by at least 0.3%, if we consider the algorithms that do not use additional training data. For the Neuronal membrane segmentation dataset we achieve the same F-score as Convolutional Neural Networks approach, but we train TICJ within 3 hours in a single CPU, comparing to about one week CNN training on a GPU cluster.

[1] Dmitry Laptev and Joachim M Buhmann. Convolutional decision trees for feature learning and segmentation. In *Pattern Recognition*. 2014.

[2] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 1995.

[3] Jamie Shotton, Toby Sharp, Pushmeet Kohli, Sebastian Nowozin, John Winn, and Antonio Criminisi. Decision jungles: Compact and rich models for classification. In *Advances in Neural Information Processing Systems 26*. 2013.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.