Journal of Machine Learning Research 15 (2014) 3847-3870 Submitted 12/12; Revised 7/14; Published 11/14

Transfer Learning Decision Forests for Gesture Recognition

Norberto A. Goussies NGOUSSIE@DC.UBA.AR
Sebastián Ubalde SEUBALDE@DC.UBA.AR
Marta Mejail MARTA@DC.UBA.AR
Departamento de Computación, Pabellón I
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Autónoma de Buenos Aires, C1428EGA
Argentina

Editor: Isabelle Guyon, Vassilis Athitsos, and Sergio Escalera

Abstract

Decision forests are an increasingly popular tool in computer vision problems. Their advantages include high computational efficiency, state-of-the-art accuracy and multi-class support. In this paper, we present a novel method for transfer learning which uses decision forests, and we apply it to recognize gestures and characters. We introduce two mechanisms into the decision forest framework in order to transfer knowledge from the source tasks to a given target task. The first one is mixed information gain, which is a data-based regularizer. The second one is label propagation, which infers the manifold structure of the feature space. We show that both of them are important to achieve higher accuracy. Our experiments demonstrate improvements over traditional decision forests in the ChaLearn Gesture Challenge and MNIST data set. They also compare favorably against other state-of-the-art classifiers.

Keywords: decision forests, transfer learning, gesture recognition

1. Introduction

Machine learning tools have achieved significant success in many computer vision tasks, including face detection (Viola and Jones, 2004), object recognition (Felzenszwalb et al., 2010), character recognition (LeCun et al., 1998) and gesture recognition (Guyon et al., 2013). Those tasks are often posed as a classification problem, namely identifying to which of a set of categories a new observation belongs. Such classifiers are usually learned from scratch using a training data set collected for the task. A major advantage of using machine learning tools is that they tend to deal robustly with the complexities found in real data.

However, in many cases it is difficult to create new training data sets for each new computer vision task. Although the problem remains unsolved, some progress has already been made in certain computer vision tasks, such as object recognition (Fei-Fei et al., 2006) and action recognition (Seo and Milanfar, 2011). The key insight is to try to replicate the ability of the human brain, which is capable of learning new concepts applying previously acquired knowledge.

Transfer learning aims at extracting the knowledge from one or more source tasks, and applying that knowledge to a target task. As opposed to multi-task learning, rather than

simultaneously learning the source and target tasks, transfer learning focus more on learning the target task. The roles of the source and target tasks are not symmetric (Pan and Yang, 2010). The goal is to exploit the knowledge extracted from the source tasks so as to improve the generalization of the classifier in the target task.

Many examples can be found in computer vision where transfer learning can be truly beneficial. One example is optical character recognition, which seeks to classify a given image into one of the characters of a given alphabet. Most methods have focused on recognizing characters from the English alphabet (LeCun et al., 1998). The recognition of characters from other alphabets, such as French, implies collecting a new training data set (Grosicki and Abed, 2011). In that case, it would be helpful to transfer the classification knowledge into the new domain.

The need for transfer learning also arises in gesture recognition (Guyon et al., 2013), which aims at recognizing a gesture instance drawn from a gesture vocabulary. For example, a gesture vocabulary may consist of Italian gestures or referee signals. In this case, the classifier needs to predict the gesture of the vocabulary that corresponds to a given video. Again, it would be interesting to improve the performance of a system by exploiting the knowledge acquired from similar vocabularies.

In this paper, we present a novel method for transfer learning which extends the decision forests framework (Breiman, 2001; Criminisi et al., 2012), and we apply it to transfer knowledge from multiple source tasks to a given target task. We introduce two mechanisms in order to transfer knowledge from the source tasks to the target task. The first one is mixed information gain, which is a data-based regularizer. The second one is label propagation, which infers the manifold structure of the feature space.

Decision forests have certain properties that make them particularly interesting for computer vision problems. First, decision forests are multi-class classifiers; therefore it is not necessary to train several binary classifiers for a multi-class problem. Second, they are fast both to train and test. Finally, they can be parallelized, which makes them ideal for GPU (Sharp, 2008) and multi-core implementations.

The first key contribution is to revise the criterion for finding the parameters of each internal node of the decision forests in the transfer learning setting. The novel criterion exploits the knowledge from the source tasks and the target task to find the parameters for each internal node of the decision forests. The additional information penalizes split functions with a high information gain in the target task and a low information gain in the source tasks. We prove that the novel criterion is beneficial.

The second key contribution is to propagate labels through leaves in order to infer the manifold structure of the feature space. The aim of this step is to assign a predictive model to the leaves without training samples of the target task after the trees of the decision forest are grown. We create a fully connected graph, for each tree in the forest, where the nodes are the leaves of the tree and the weight of each edge takes into account the training data reaching the leaves. An implicit assumption of this step is that nearby leaves should have similar predictive models.

We extensively validate our approach in two challenging data sets. First, our experiments in the ChaLearn gesture challenge data set (Guyon et al., 2012) show that our method does not have a uniform margin of improvement over all the tasks. However, we demonstrate that when there are source tasks related to the target task, we obtain greater

improvements. Second, our experiments in the MNIST data set (LeCun et al., 1998) show that greater improvements are obtained, compared to classification decision forests, when there are only a few training samples.

This paper is organized as follows. We summarize previous work on transfer learning in Section 2. Section 3 describes the novel transfer learning decision forest in, illustrates its performance on some artificial data sets, and proves some properties of the mixed information gain. In Section 4 we show how the transfer learning decision forests can be used to recognize gestures when there is only one training sample. We present our experiments on the ChaLearn data set and the MNIST data set in Section 5. Finally, Section 6 details our conclusions.

2. Related Work

In the following we will review transfer learning techniques which have been applied to computer vision problems. A recent survey (Pan and Yang, 2010) provides a comprehensive overview of the developments for classification, regression and clustering. In recent years, the computer vision community has become increasingly interested in using transfer learning techniques, especially for object recognition (Levi et al., 2004; Sudderth et al., 2005; Fei-Fei et al., 2006; Bart and Ullman, 2005; Torralba et al., 2007; Quattoni et al., 2008; Bergamo and Torresani, 2010; Gopalan et al., 2011; Saenko et al., 2010; Tommasi et al., 2014).

A variety of methods have been proposed in the generative probabilistic setting (Fei-Fei et al., 2006; Sudderth et al., 2005). These models consider the relationships between different object parts during the training process. The key idea is to share some parameters or prior distributions between object categories, using the knowledge from known classes as a generic reference for newly learned models. The association of objects with distributions over parts can scale linearly (Sudderth et al., 2005), or exponentially (Fei-Fei et al., 2006).

Moreover, discriminative models have been extended to the transfer learning setting (Dai et al., 2007; Yao and Doretto, 2010; Aytar and Zisserman, 2011; Tommasi et al., 2014; Lim et al., 2011; Torralba et al., 2007). Transfer learning has been applied to the SVM framework, during the training process of the target detector the previously learned template is introduced as a regularizer into the cost function (Tommasi et al., 2014; Aytar and Zisserman, 2011). Based on boosting (Freund and Schapire, 1997) a framework that allows users to utilize a small amount of newly labeled data has been developed (Dai et al., 2007). Later, the framework has been extended for handling multiple sources (Yao and Doretto, 2010) .

More similar to our method, instance transfer approaches (Pan and Yang, 2010) consider source and target data together during the training process. A loss function for borrowing examples from other classes in order to augment the training data of each class has been proposed by Lim et al. (2011). A method for learning new visual categories is described by Quattoni et al. (2008), using only a small subset of reference prototypes for a given set of tasks. As mentioned earlier, a boosting-based algorithm that allows knowledge to be effectively transferred from old to new data has been proposed by Dai et al. (2007) and extended later by Yao and Doretto (2010). The effectiveness of the novel algorithm is analyzed both theoretically and empirically. In this paper, we develop an instance transfer

approach that exploits source and target data to find the parameters of each internal node of the decision forest.

Few researchers have addressed the problem of transfer learning using decision forests or trees. Leistner et al. (2009) extends random forests to semi-supervised learning. In order to incorporate unlabeled data a maximum margin approach is proposed, which is optimized using a deterministic annealing-style technique. Wang et al. (2008) proposed to treat each input attribute as extra task to bias each component decision tree in the ensemble. Pei et al. (2013) proposed a novel criterion for node splitting to avoid the rank deficiency in learning density forests for lipreading. The method proposed by won Lee and Giraud-Carrier (2007) learns a new task by traversing and transforming a decision tree previously learned for a related task. The transfer learning decision tree learns the target task from a partial decision tree model induced by ID3 (Quinlan, 1986). In this paper, we follow a different approach, first we consider the source and target data when we build each tree of the decision forest. Second, decision forests reduce the variance of the classifier aggregating the results of multiple random decision trees.

Our approach shares some features with the work by Faddoul et al. (2012), who propose to transfer learning with boosted C4.5 decision trees. The main difference is that their method reduces the variance of the decision trees by means of boosting, which has been shown to be less robust against label noise when compared with decision forests (Breiman, 2001; Leistner et al., 2009). In addition, we use label propagation to learn the manifold structure of the feature space, and assign predictive models only to the leaves of the trees.

There has been a growing interest in applying transfer learning techniques to gesture recognition. A method for transfer learning in the context of sign language is described by Farhadi et al. (2007). A set of labeled words in the source and target data is shared so as to build a word classifier for a new signer on a set of unlabeled target words. A transfer learning method for conditional random fields is implemented to exploit information in both labeled and unlabeled data to learn high-level features for gesture recognition by Liu et al. (2010). More recently, the ChaLearn Gesture Competition (Guyon et al., 2013) provided a benchmark of methods that apply transfer learning to gesture recognition. Several approaches submitted to the competition have been published (Malgireddy et al., 2013; Lui, 2012; Wan et al., 2013).

3. Transfer Learning Decision Forests

We consider $N+1$ classification tasks $T_0, \ldots, T_N$ over the instance space $\mathbb{R}^d$ and label sets $\mathcal{Y}_0, \ldots, \mathcal{Y}_N$. We are interested in solving the classification task T_0 using the knowledge of the other tasks in order to improve classification accuracy. Our transfer learning algorithm will take as input the training set $S = \{(\mathbf{x}_i, \mathbf{y}_i, j) | \mathbf{x}_i \in \mathbb{R}^d,\ \mathbf{y}_i \in \mathcal{Y}_j,\ j \in \{0, \ldots, N\},\ 1 \leq i \leq M\}$. The projected sets $T_j S = \{(\mathbf{x}_i, \mathbf{y}_i) | \mathbf{x}_i \in \mathbb{R}^d,\ \mathbf{y}_i \in \mathcal{Y}_j, (\mathbf{x}_i, \mathbf{y}_i, j) \in S\}$ are the training sets for each task T_j. The empirical histogram for a training set S of a task T is defined as $\hat{p}_{TS}(\mathbf{y}) = \frac{1}{|TS|} \sum_{(\mathbf{x}', \mathbf{y}') \in TS} \delta_{\mathbf{y}'}(\mathbf{y})$ where $\delta_{\mathbf{y}'}(\mathbf{y})$ is the Kronecker delta and the empirical entropy is defined as $\mathcal{H}(TS) = -\sum_{\mathbf{y} \in \mathcal{Y}} \hat{p}_{TS}(\mathbf{y}) \log(\hat{p}_{TS}(\mathbf{y}))$, we will note $\hat{p}_S(\mathbf{y})$ or $\mathcal{H}(S)$ to make the notation simpler when it is convenient and unambiguous.

The goal is to find a decision forest $\mathcal{F} = \{F_1, \ldots, F_T\}$, defined as an ensemble of T decision trees F, which minimizes the classification error. A decision tree F is a strictly

binary tree in which each node k represents a subset R_k in the instance space $\mathbb{R}^d$ and all the leaves ∂F form a partition $\mathcal{P}$ of $\mathbb{R}^d$. In addition, each leaf $k \in \partial F$ of a decision tree F has a predictive model associated with it: $p_F(\mathbf{y}|\mathbf{x} \in R_k)$. The internal nodes $k \in F^\circ$ of a decision tree have a linear split function: $h(\mathbf{x}, \boldsymbol{\theta}_k) = \mathbf{x} \cdot \boldsymbol{\theta}_k$, where $\boldsymbol{\theta}_k$ are the parameters of node k. The subset represented by the left child k_L of node k is defined as $R_{k_L} = R_k^L = \{\mathbf{x} \in \mathbb{R}^d | \mathbf{x} \in R_k \wedge h(\mathbf{x}, \boldsymbol{\theta}_k) < 0\}$ and, similarly, we define $R_{k_R} = R_k^R = \{\mathbf{x} \in \mathbb{R}^d | \mathbf{x} \in R_k \wedge h(\mathbf{x}, \boldsymbol{\theta}_k) \geq 0\}$ as the subset represented by the right child k_R. The training set reaching node k is defined as $S_k = \{(\mathbf{x}, \mathbf{y}, j) \in S | \mathbf{x} \in R_k\}$.

3.1 Training

The training algorithm of a decision forest $\mathcal{F}$ consists in training each of the trees $F \in \mathcal{F}$ independently, introducing a certain level of randomness in the training process in order to de-correlate individual tree predictions and improve generalization.

We grow each tree using an extension of the classical training algorithm (Criminisi et al., 2012). The algorithm follows a top-down approach, optimizing the parameters $\boldsymbol{\theta}$ of the root node in the beginning and recursively processing the child nodes. The recursion is stopped when all the items in the training set have the same labels, or the maximum depth D is reached, or the number of points reaching the node is below the minimum number of points allowed κ.

In this paper, we adapt the procedure for optimizing the parameters $\boldsymbol{\theta}_k$ for each node $k \in F^\circ$ to the transfer learning setting (Pan and Yang, 2010). The difference between the classification decision forest (Criminisi et al., 2012) and the transfer learning decision forest is the objective function. In the former, the information gain is used to find the best parameters, taking into account only one task. By contrast, in this paper we use the mixed information gain function as described in Section 3.1.1.

The partition $\mathcal{P}$ defined by the leaves ∂F after making a tree F grow might contain regions R with no training samples of the target task T_0. Therefore, we cannot define a predictive model for those regions. In order to overcome this issue we infer the labels from the regions that have training samples of task T_0, as described in Section 3.1.2.

3.1.1 Mixed Information Gain

We believe that valuable knowledge can be transferred from the source tasks $T_1, \ldots, T_N$ to the target task T_0, as it happens with humans. For example, it is simpler to learn a new sign language if another sign language has already been learned. In other words, there is latent information that can be understood as common sense.

In our formulation, this common sense information is included in the process of making each tree $F \in \mathcal{F}$ in the forest grow. The main idea is, therefore, to find parameters $\boldsymbol{\theta}_k$ for each $k \in F^\circ$ in order to obtain a partition $\mathcal{P}$ of the feature space $\mathbb{R}^d$ such that, in each region $R \in \mathcal{P}$, the training samples of each task T have the same label. This aims at improving the generalization capabilities of each tree independently, since each region $R \in \mathcal{P}$ is found using more training samples, and is more general because it is encouraged to split the training samples of several tasks simultaneously.

Unfortunately, this is a very difficult problem. For this reason, we use a greedy heuristic which consist in recursively choosing for each internal node $k \in F^\circ$ the parameters $\boldsymbol{\theta}_k$ of the

split function $h(\mathbf{x}, \boldsymbol{\theta}_k)$, which makes the training samples reaching the child nodes as "pure" as possible. The information gain achieved by splitting the training set TS_k reaching the internal node $k \in F^\circ$ of a task T using parameter $\boldsymbol{\theta}_k$ is computed using the information gain function

$$\mathcal{I}(TS_k, \boldsymbol{\theta}_k) = \mathcal{H}(TS_k) - \sum_{i \in \{L,R\}} \frac{|TS_k^i|}{|TS_k|} \mathcal{H}(TS_k^i)$$

where $TS_k^L = \{(\mathbf{x}, \mathbf{y}) | (\mathbf{x}, \mathbf{y}) \in TS_k \wedge h(\mathbf{x}, \boldsymbol{\theta}_k) < 0\}$ and $TS_k^R = \{(\mathbf{x}, \mathbf{y}) | (\mathbf{x}, \mathbf{y}) \in TS_k \wedge h(\mathbf{x}, \boldsymbol{\theta}_k) \geq 0\}$. In this paper, the parameters $\boldsymbol{\theta}_k$ of each internal node $k \in F^\circ$ are found maximizing the information gain of all the tasks $T_0, \ldots, T_N$ simultaneously

$$\boldsymbol{\theta}_k^* = \arg \max_{\boldsymbol{\theta}_k \in \mathcal{T}_k} (1 - \gamma) \mathcal{I}(T_0 S_k, \boldsymbol{\theta}_k) + \gamma \sum_{n=1}^{N} p_{n,k} \mathcal{I}(T_n S_k, \boldsymbol{\theta}_k) \tag{1}$$

where γ is a scalar parameter that weights the two terms, $\mathcal{T}_k \subset \mathbb{R}^d$ is a small subset of the instance space available when training the internal node $k \in F^\circ$, and $p_{n,k}$ is the fraction of samples of the source task T_n in the samples reaching the node k, $p_{n,k} = \frac{|T_n S_k|}{\sum_{j=1}^{N} |T_j S_k|}$.

The maximization of (1) is achieved using randomized node optimization (Criminisi et al., 2012). We perform an exhaustive search over subset $\mathcal{T}_k$ of the feature space parameters $\mathbb{R}^d$. The size of the subset is a training parameter noted as $\rho = |\mathcal{T}_k|$. The randomized node optimization is a key aspect of the decision forest model, since it helps to de-correlate individual tree predictions and to improve generalization.

The first term of the objective function in (1) is the information gain associated with the training samples reaching node k for the target task T_0. This term encourages the parameters $\boldsymbol{\theta}_k$ to find a split function $h(\mathbf{x}, \boldsymbol{\theta}_k)$ that decreases the entropy of the training set of the target task T_0 reaching the children nodes of k.

Additional information is introduced into the second term of the objective function in (1) for the purposes of increasing the generalization performance. This information encourages the parameters $\boldsymbol{\theta}_k$ to make the training samples of source tasks reaching the descendant nodes of k as pure as possible. The key idea is that this term penalizes split functions $h(\mathbf{x}, \boldsymbol{\theta}_k)$ with a high information gain in the target task T_0 and a low information gain in the source tasks $T_1, \ldots, T_N$. Those splits might have a high information gain in the target task T_0 only because the training set for task T_0 is limited, and if we choose them the generalization performance will decrease.

A key insight of our work is an alternative representation of the second term in (1). It is possible to consider all the source tasks $T_1, \ldots, T_N$ together concatenating the label sets $\mathcal{Y}_1, \ldots, \mathcal{Y}_N$, denoted by $\mathcal{Y}_{1\ldots N} = \oplus_{n=1}^{N} \mathcal{Y}_n$. The new task is noted as $T_{1\ldots N}$ and the training sample is noted as $T_{1\ldots N} S = \{(\mathbf{x}, \mathbf{y}) | (\mathbf{x}, \mathbf{y}, j) \in S, j \in \{1, \ldots, N\}, \mathbf{y} \in \oplus_{n=1}^{N} \mathcal{Y}_n\}$. Using the generalized grouping rule of the entropy (Cover and Thomas, 2006) an alternative expression for the second term in (1) is found

$$\mathcal{I}(T_{1\ldots N} S_k, \boldsymbol{\theta}_k) = \sum_{n=1}^{N} p_{n,k} \mathcal{I}(T_n S_k, \boldsymbol{\theta}_k).$$

This equation relates the information gain of several source tasks $T_1, \ldots, T_N$ to the information gain of another source task $T_{1\ldots N}$. An important consequence of this equation

is that we can combine the training set of the simpler tasks $T_1, \ldots, T_N$ to obtain a larger training set for another source task $T_{1\ldots N}$. Therefore, increasing the number of training samples per source task or the number of source tasks has a similar effect.

This observation has previously been made in the multi-task learning literature (Faddoul et al., 2012). However, Faddoul et al. (2012) avoids the high variance of the decision trees by using the boosting framework, whereas we use a different approach, based on decision forest, for the same purpose.

We explain in more detail how the combination of the information gain of tasks $T_0, \ldots, T_N$ for finding the optimal parameters $\boldsymbol{\theta}_k$ improves the generalization properties of the decision forests. The parameters $\boldsymbol{\theta}_k$ are found using an empirical estimation of the entropy $\mathcal{H}(S_k)$ of the training samples S_k reaching node k and its children. Consequently, errors in estimating entropy can result in very different trees. Tighter bounds for the expected entropy are found by increasing the number of training samples, as explained in Theorem 1.

Theorem 1 *Let P be a probability distribution on $\mathbb{R}^d \times \mathcal{Y}$ such that the marginal distribution over $\mathcal{Y}$ is a categorical distribution with parameters $p_1, \ldots, p_{|\mathcal{Y}|}$, and suppose $S_K = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_K, \mathbf{y}_K)\}$ is the set generated by sampling K times from $\mathbb{R}^d \times \mathcal{Y}$ according to P. Let $\mathcal{H}(P) = -\sum_{y=1}^{|\mathcal{Y}|} p_y \log(p_y)$ be the entropy of distribution P. Then $\mathbb{E}(\mathcal{H}(S_K)) + \sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log\left(1 + \frac{1-p_{\mathbf{y}}}{Kp_{\mathbf{y}}}\right) \leq \mathcal{H}(P) \leq \mathbb{E}(\mathcal{H}(S_K))$.*

This theorem is proved in the Appendix A.

Theorem 1 shows that the empirical entropy $\mathcal{H}(S_K)$ is closer to the entropy of the distribution P when the training set is larger, since when $K \to \infty$, $\log\left(1 + \frac{1-p_{\mathbf{y}}}{Kp_{\mathbf{y}}}\right) \to 0$. Therefore, if we assume that the source tasks are related to the target task i.e., both have a similar distribution P, using Theorem 1 we can conclude that the mixed information gain (1) finds parameters $\boldsymbol{\theta}_k$ that achieve lower generalization errors than the traditional information gain $\mathcal{I}(T_0 S_k, \boldsymbol{\theta}_k)$.

To gain some insight into how the mixed information gain works, Figure 1 considers a toy problem with two tasks, each with two labels. It is intuitively clear that the problem of estimating the information gain of a split with only a few training samples of the target task is that there are a lot of possible splits with the same empirical information gain but different generalization capabilities. Our goal is to discover which split to use, and we intend to choose the one with the best generalization capability. In Figure 1 all the splits have the same information gain but different mixed information gain. When, in our formulation, we use the additional training samples from the source tasks to compute the information gain of a split, some of the splits are penalized for having a low information gain in the source task and, thus, this allows us to find a split with increased generalization.

One of the major problems with decision trees is their high variance. A small change in the training data can often result in a very different series of splits. The major reason for this instability is the hierarchical nature of the process: the effect of an error in the top split is propagated down to all the splits below it (Hastie et al., 2003). Decision forests (Breiman, 2001) build a large collection of de-correlated decision trees, and hence reduce the variance averaging the prediction of each of them. The mixed information gain is a complementary approach for reducing their variance which increases the generalization of each tree independently. It is important to note that the mixed information preserves the

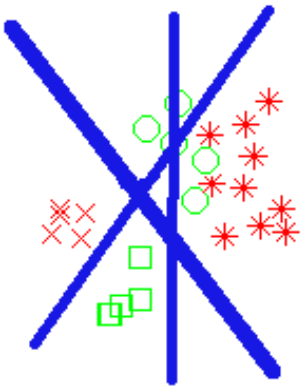


Figure 1: Illustration of mixed information gain on a toy problem in which there are two tasks, each with two labels. The thickness of the blue lines indicates the mixed information gain of the split (all the splits have the same information gain). Task T_0 has two green labels ($\mathcal{Y}_0 = \{\times, *\}$) and task T_1 has two red labels ($\mathcal{Y}_1 = \{\bigcirc, \square\}$).

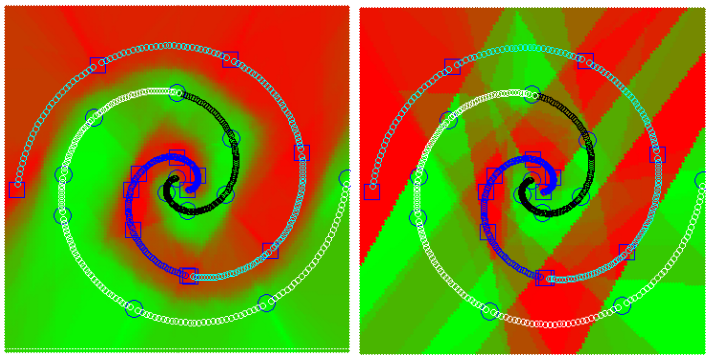
diversity of the forests, which is essential to improve the generalization error. The random nature of the random node optimization (Criminisi et al., 2012) used to optimize (1) allows us to keep a high diversity among the trees.

Figures 2a and 2b compare the output classification on all the points in a rectangular section of the feature space for a decision forest classifier and for our transfer learning decision forest classifier. Both decision forests were trained with the same maximum depth $D = 8$, and have the same number of trees $|\mathcal{F}| = 100$. The data set for the target and source task is organized in the shape of a two-arm spiral. We can see that the classification decision forests have serious generalization problems since, even when all the training data of the target task is correctly classified, the spiral structure is not predicted accurately. In contrast, the spiral structure is predicted by the transfer learning decision forests as shown in Figure 2a.

3.1.2 Label Propagation

For each leaf $k \in \partial F$ of each tree $F \in \mathcal{F}$, we must have a predictive model $p_F(\mathbf{y}|\mathbf{x} \in R_k)$ that estimates the probability of label $\mathbf{y} \in \mathcal{Y}_0$ given a previously unseen test input $\mathbf{x} \in R_k \subseteq \mathbb{R}^d$. This poses a problem when we make each tree grow using the mixed information gain because we may end up with leaves $k \in \partial F$ that have no training samples of the target task T_0 to estimate the predictive model $p_F(\mathbf{y}|\mathbf{x} \in R_k)$. In this paper we use label propagation to assign a predictive model $p_F(\mathbf{y}|\mathbf{x} \in R_k)$ to those leaves.

We are given a set of leaves $\mathcal{U} \subseteq \partial F$ without training samples of the target task T_0 and a set of leaves $\mathcal{L} \subseteq \partial F$ with training samples of the target task T_0. The goal is to obtain a predictive model $p_F(\mathbf{y}|\mathbf{x} \in R_k)$ for the leaves $k \in \mathcal{U}$ avoiding the propagation of labels through low density regions but, at the same time, propagating labels between nearby



(a) (b)

Figure 2: Left: Output classification of a transfer learning decision forest, tested on all points in a rectangular section of the feature space. The color associated with each test point is a linear combination of the colors (red and green) corresponding to the two labels ($\square, \bigcirc$) in the target task. The training data for the target task is indicated with big markers and the training data for the source task is indicated with small markers. Right: Output classification of a decision forest tested in the same feature space section as before but trained using only data for the target task.

leaves. We construct a complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \partial F$ is the vertex set and $\mathcal{E}$ is the edge set with each edge $\mathbf{e}_{ij} \in \mathcal{E}$ representing the relationship between nodes $i, j \in \partial F$.

Edge $\mathbf{e}_{ij} \in \mathcal{E}$ is weighted taking into account the training samples of tasks $T_0, \ldots, T_N$. For each leaf $k \in \partial F$ we define the estimated mean $\boldsymbol{\mu}_k$ and estimated covariance Σ_k using the training samples reaching the node

$$\begin{aligned} \boldsymbol{\mu}_k &= \frac{1}{|S_k|} \sum_{(\mathbf{x},\mathbf{y},j) \in S_k} \mathbf{x} \\ \Sigma_k &= \sum_{(\mathbf{x},\mathbf{y},j) \in S_k} \sum_{(\mathbf{x}',\mathbf{y}',j') \in S_k} (\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x}' - \boldsymbol{\mu}_k)^T. \end{aligned}$$

We use the estimated mean $\boldsymbol{\mu}_k$ and estimated covariance Σ_k to define the weight between two nodes $\mathbf{e}_{ij} \in \mathcal{E}$

$$\mathbf{e}_{ij} = \frac{1}{2} \left(d_{ij}^T \Sigma_i d_{ij} + d_{ij}^T \Sigma_j d_{ij} \right)$$

where $d_{ij} = \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$ is the difference between the estimated mean of the leaves $i, j \in \partial F$. Weight $\mathbf{e}_{ij} \in \mathcal{E}$ is the symmetric Mahalanobis distance. We use it to discourage the propagation of labels through low density regions. For each node $k \in \mathcal{U}$ we find the shortest path in graph $\mathcal{G}$ to all the nodes in $\mathcal{L}$. Let $s_k^* \in \mathcal{L}$ be the node with the shortest path to node k. We assign the predictive model $p_F(\mathbf{y}|\mathbf{x} \in R_{s_k^*})$ to $p_F(\mathbf{y}|\mathbf{x} \in R_k)$.

Label propagation methods are usually at least quadratic $\mathcal{O}(n^2)$ in terms of the number of training samples, making them slow when a large number of training samples is avail-

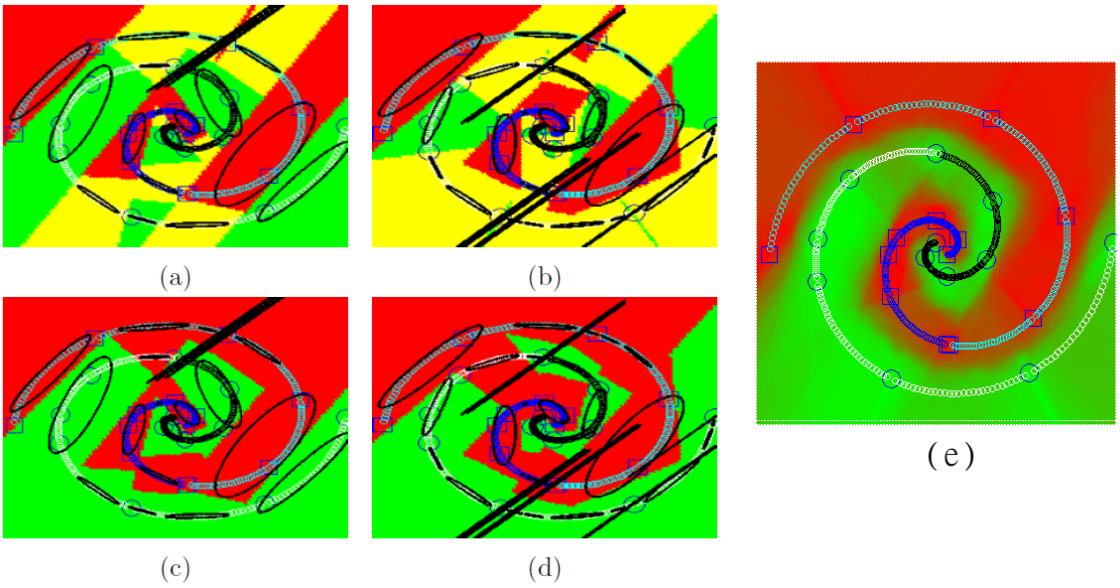


Figure 3: Illustration of the label propagation procedure between regions, as before the training data for the target task is indicated with big markers and the training data for the source task is indicated with small markers. The ellipses in black are the isocontours of a Gaussian distribution learned by maximum likelihood for each region using the training samples in the region. (a, b) show the predictive model for two different trees $F \in \mathcal{F}$ before propagating labels. The color associated with each region is a linear combination of the colors (red and green) corresponding to the two labels ($\square, \bigcirc$) in the target task. The regions in yellow are the ones without training data of the target task. (c, d) show the predictive model after the label propagation. (e) Output classification of the final transfer learning decision forest.

able. We avoid this problem by propagating the predictive model of the leaves, instead of propagating the labels of the training samples.

We illustrate the behavior of label propagation in Figure 3 using a 2D toy example. We consider the same two-arm spiral problem of Figure 2 which has data that follow a complex structure. We show the predictive models for the regions of two randomly grown trees before and after propagating labels. We observe that the predictive models are propagated following the intrinsic structure of the data, as a consequence of taking into account the training data of each region.

3.2 Testing

The predictive model of all the trees $F \in \mathcal{F}$ is combined to produce the final prediction of the forest

$$P_{\mathcal{F}}(\mathbf{y} = y|\mathbf{x}) = \frac{1}{|\mathcal{F}|} \sum_{F \in \mathcal{F}} P_F(\mathbf{y} = y|\mathbf{x}).$$

Let $l_F : \mathbb{R}^d \to \partial F$ be the function that, given a sample $\mathbf{x} \in \mathbb{R}^d$, returns the leaf such that $\mathbf{x} \in R_{l_F(\mathbf{x})}$. The prediction for a tree F is:

$$P_F(\mathbf{y} = y|\mathbf{x}) = P_F\left(\mathbf{y} = y|\mathbf{x} \in R_{l_F(\mathbf{x})}\right).$$

Finally, let $k \in \partial F$ be the leaf that is reached by sample $\mathbf{x} \in \mathbb{R}^d$. The class distribution for that leaf is:

$$P_F(\mathbf{y} = y|\mathbf{x} \in R_k) = \begin{cases} \hat{p}_{T_0 S_k}(y) & \textbf{if } T_0 S_k \neq \emptyset \\ \hat{p}_{T_0 S_{s_k^*}}(y) & \textbf{otherwise} \end{cases}.$$

Thus, $P_F(\mathbf{y} = y|\mathbf{x})$ is the empirical histogram of the training samples of the target task T_0 reaching node $l_F(\mathbf{x})$ if any. Otherwise, $P_F(\mathbf{y} = y|\mathbf{x})$ is the empirical histogram associated with the node that has the shortest path to $l_F(\mathbf{x})$.

4. Gesture Recognition

Gesture recognition is one of the open challenges in computer vision. There is a big number of potential applications for this problem, including surveillance, smart-homes, rehabilitation, entertainment, animation and human–robot interaction and sign language recognition just to mention a few. The task of gesture recognition is to determine the gesture label that best describes a gesture instance, even when performed by different people, from various viewpoints and in spite of large differences in manner and speed.

To reach that goal, many approaches combine vision and machine learning tools. Computer vision tools are employed to extract features that provide robustness to distracting cues and that, at the same time, are discriminative. Machine learning is used to learn a statistical model from those features, and to classify new examples using the models learned. This poses a problem in gesture recognition since it is difficult to collect big data sets to learn statistical models. Therefore, in this paper we perform experiments aimed at showing that our transfer learning decision forests are useful to mitigate this problem.

Recently, the ChaLearn competition (Guyon et al., 2012) provided a challenging data set to evaluate whether transfer learning algorithms can improve their classification performance using similar gesture vocabularies. The data set is organized into batches, with only one training example of each gesture in each batch. The goal is to automatically predict the gesture labels for the remaining gesture sequences (test examples). The gestures of each batch are drawn from a small vocabulary of 8 to 12 unique gestures, when we train a classifier to predict the labels of a target batch (or task) T_0 we use the training samples of T_0 and of the other batches $T_1, \ldots, T_N$.

Each batch of the ChaLearn competition includes 100 recorded gestures grouped in sequences of 1 to 5 gestures performed by the same user (Guyon et al., 2012). There is only one gesture in the training sequences, but there might be more than one gesture in the testing sequences. Therefore in order to use the method described in this section we need to temporally segment the testing sequences. To this end, we use the dynamic time warping (DTW) implementation given by the organizers.

In this section, we describe the features and the classifiers used to validate our approach, as well as their application to the ChaLearn competition (Guyon et al., 2012). First, Section 4.1 describes the features, and then, Section 4.2 describes the classifier.

$\tau \setminus \xi$	16	24	32	40
1	32.61 ± 0.14 %	32.61 ± 0.24 %	30.35 ± 0.22 %	29.26 ± 0.26 %
4	**30.43 ± 0.17 %**	31.52 ± 0.15 %	29.26 ± 0.15 %	28.17 ± 0.19 %
8	**30.43 ± 0.13 %**	**27.35 ± 0.16 %**	**28.09 ± 0.14 %**	**27.06 ± 0.18 %**
12	32.12 ± 0.23 %	32.61 ± 0.29 %	34.78 ± 0.31 %	29.35 ± 0.33 %
16	33.72 ± 0.28 %	32.61 ± 0.29 %	34.78 ± 0.25 %	30.43 ± 0.31 %

Table 1: Classification error in the test set of the *devel*11 batch for different combination of MHI parameters. In all the experiments we leave the the spatial resolution of each frame fixed to $\omega_1 \times \omega_2 = 16 \times 12$.

4.1 Motion History Images

Given a depth video V where $V(x, y, t)$ is the depth of the pixel with coordinates (x, y) at the tth frame. We compute the motion history image (MHI) (Bobick and Davis, 1996, 2001; Ahad et al., 2012) for each frame using the following function:

$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if } |V(x, y, t) - V(x, y, t-1)| \geq \xi \\ \max(0, H_\tau(x, y, t-1) - 1) & \text{otherwise} \end{cases}$$

where τ defines the temporal extent of the MHI, and ξ is a threshold employed to perform the foreground/background segmentation at frame t. The result is a scalar-valued image for each frame of the original video V where pixels that have moved more recently are brighter. MHI H_τ represents the motion in an image sequence in a compact manner, the pixel intensity is a function of the temporal history of motion at that point. A common problem when computing MHI H_τ using the color channel is the presence of textured objects in the image sequence; here we use the depth video V to overcome this issue. This is a relevant problem in gesture recognition, because, as a result of the clothes texture, the MHI is noisy (Ahad et al., 2012).

An interesting property of the MHI is that it is sensitive to the direction of motion; hence it is well suited for discriminating between gestures with an opposite direction. An advantage of the MHI representation is that a range of times may be encoded in a single frame, and thus, the MHI spans the time scale of human gestures. After computing MHI H_τ we reduce the spatial resolution of each frame to $\omega_1 \times \omega_2$ pixels. Then, we flatten the MHI for each frame and obtain a feature $\mathbf{x}_m \in \mathbb{R}^{\omega_1 \omega_2}$.

Figure 4 contrasts the result of computing the MHI using the RGB channel with the one obtained using the depth channel. In the first row, we see that the clothes texture generates noise in the MHI computed using the RGB channel. In the second row, the MHI of the RGB channel is noisy because of the shadow from the moving arm. Both problems are avoided using the depth channel for computing the MHI. The parameters to compute the MHI in all the cases were $\tau = 15$, and $\xi = 30$. Table 1 shows the classification error in the test set of the *devel*11 batch of the ChaLearn competition, after training a decision forest with the following parameters $D = 8, T = 50$.

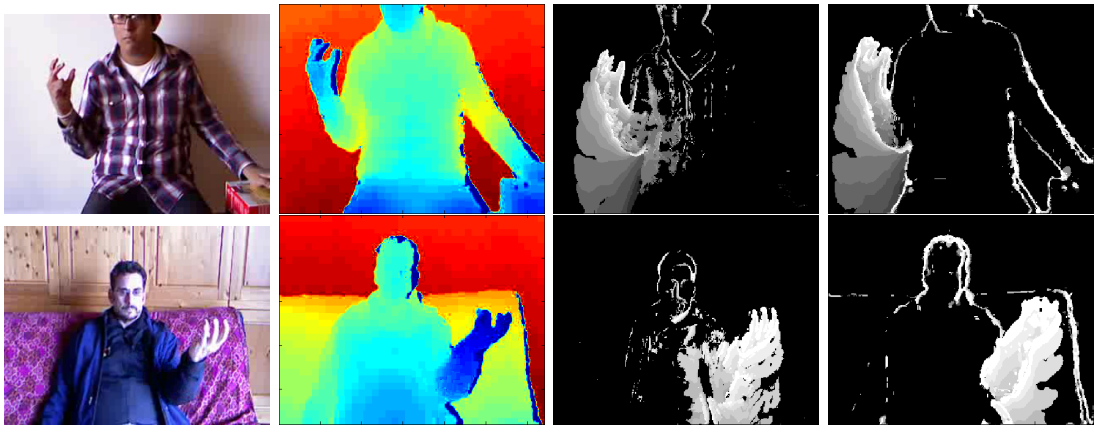


Figure 4: Comparison of the MHI computed using the depth channel or the RGB channel for two different training videos of the ChaLearn competition. The first two columns show the RGB channel and the depth channel, whereas the third and fourth columns show the MHI computed using the RGB channel and the MHI computed using the depth channel, respectively.

4.2 Naive Bayes

A main research trend in gesture recognition is to train hidden Markov models (HMMs) and theirs variants (Bowden et al., 2004; Kurakin et al., 2012), in order to exploit the temporal relation of a gesture. A drawback of this approach is that many training samples are required to train the large number of parameters of an HMM. Additionally, recognition rates might not improve significantly (Li et al., 2008). This limitation has been recognized by Bowden et al. (2004) and a two-stage classifier was proposed to obtain one-shot learning.

Since in the ChaLearn competition (Guyon et al., 2012) there is only one labeled training sample of each gesture, we use the naive Bayes model which has a smaller number of parameters than HMM. We use transfer learning decision forests to predict the probability that each frame will be part of a given gesture. We combine the predictions of the transfer learning decision forests for each frame using the naive Bayes model. An advantage of the naive Bayes assumption is that it is not sensitive to irrelevant frames (the probabilities for all the labels will be quite similar).

Given a video V of an isolated gesture, we want to find its label $\mathbf{y} \in \mathcal{Y}_0$. Assuming that the class prior $p(\mathbf{y})$ is uniform we have:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}_0} p(\mathbf{y}|V) = \arg\max_{\mathbf{y}\in\mathcal{Y}_0} p(V|\mathbf{y})$$

Let $\mathbf{x}_1, \ldots, \mathbf{x}_M$ denote the MHI for each frame of a video V with M frames. We assume the naive Bayes model i.e., that the features $\mathbf{x}_1, \ldots, \mathbf{x}_M$ are i.i.d. given the label $\mathbf{y}$, namely:

$$p(V|\mathbf{y}) = p(\mathbf{x}_1, \ldots, \mathbf{x}_M|\mathbf{y}) = \prod_{m=1}^{M} p(\mathbf{x}_m|\mathbf{y}) = \prod_{m=1}^{M} p(\mathbf{y}|\mathbf{x}_m)\frac{p(\mathbf{x}_m)}{p(\mathbf{y})} \tag{2}$$

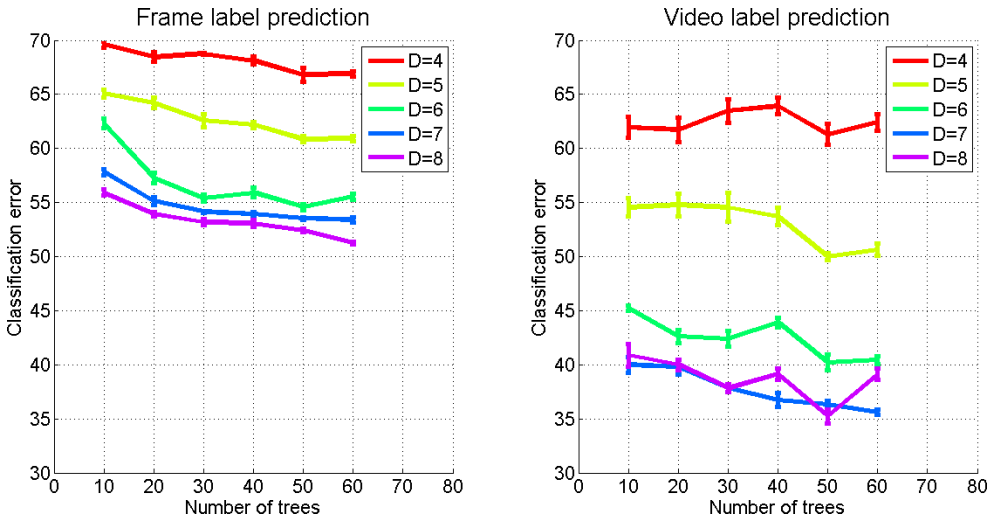


Figure 5: Effect of the training parameters for the frame label classification error $p(\mathbf{y}|\mathbf{x})$ (left) and video label classification error $p(\mathbf{y}|V)$ (right) in the devel11 batch using the transfer learning decision forests.

We compute the probability $p(\mathbf{y}|\mathbf{x}_m)$ using our proposed transfer learning decision forest $\mathcal{F}$. The data set for training the forest $\mathcal{F}$ consists of all the frames in each training video in the target task T_0 and source tasks $T_1, \ldots, T_N$. We propose to use the frames of the training videos in the source tasks to obtain a better classifier for each frame.

Taking the logarithm in (2) and ignoring the constant terms we obtain the following decision rule:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}_0} p(\mathbf{y}|V) = \arg\max_{\mathbf{y}\in\mathcal{Y}_0} \sum_{m=1}^{M} \log\left(p_{\mathcal{F}}(\mathbf{y}|\mathbf{x}_m)\right)$$

Note that we use the same forest $\mathcal{F}$ for computing the label distribution of all the frames in video V. For this reason, given a frame $\mathbf{x}$, we expect distribution $p_{\mathcal{F}}(\mathbf{y}|\mathbf{x})$ to be multimodal, which is an issue for several statistical methods. However, since the transfer learning decision forest has a predictive model for each leaf of its tree, it can deal with this type of distribution without major problems

Figure 5 compares the classification error when predicting the label of a frame $p(\mathbf{y}|\mathbf{x})$ with the classification error when predicting the label of a video $p(\mathbf{y}|V)$, for different combinations of training parameters in the devel11 batch. We observe that the maximum depth D has a larger impact to predict the label of a video than the number of trees $|\mathcal{F}|$. Moreover, the classification error when predicting the label of a frame is greater than the classification error when predicting the label of a video. This means, as expected, that some frames are more discriminative than others, and that the misclassification of some frames is not a decisive factor for classifying a video correctly.

5. Experiments

In this section we present a series of experiments on the ChaLearn Gesture Challenge (Guyon et al., 2012) and MNIST data set (LeCun et al., 1998) to assess the performance of our proposed algorithm.

5.1 ChaLearn Gesture Challenge

Here, we evaluate the transfer learning decision forests on the ChaLearn Gesture Challenge. First, we compare the results obtained for different parameters of the transfer learning decision forests, and then we compare these results with the ones reported in related works. For the MHI computation in this section, we set the temporal extent $\tau = 8$, the threshold $\xi = 25$, and reduce the spatial resolution of each frame to $\omega_1 \times \omega_2 = 16 \times 12$ pixels.

5.1.1 Transfer Decision Learning Parameters

To obtain a general idea of the effect of the training parameters, Figure 6 evaluates the classification error for different combinations of training parameters. We report the average classification error obtained in the *devel* batches. We use the temporal segmentation of the videos provided by the ChaLearn competition organizers. The experiments show that when the mixing coefficient γ is between 25% and 50%, the classification error is the smallest. This means that we obtain improvements when transferring knowledge from related tasks but, nevertheless, we still need to make the decision trees grow using information of the target task.

It is important to remark that when $\gamma = 0$ we are not using the training data of the source tasks and our mixed information gain simplifies to the usual information gain, thus, only the label propagation extension is being used. The classification error for the case $\gamma = 0$ indicates that we achieve an improvement using the label propagation alone. We obtain an additional improvement when γ is between 25% and 75%, therefore we can conclude that both extensions are important to reduce the classification error.

The maximum depth of the trees is a highly relevant parameter for the transfer learning decision forests, and has some influence for the classification decision forests. As expected, the greater the maximum depth, the smaller the classification error. It is interesting to observe that the difference in the classification error between different values of the mixing coefficients γ is reduced when the maximum depth is increased.

Figure 7 shows the confusion matrices for the classifiers of the transfer learning decision forests (TLDFs) and the decision forests (DFs) in the batches *devel*06, *devel*11 and *devel*14. To train the TLDFs, we set the number of trees $T = 50$, the maximum depth $D = 8$, the mixing coefficient $\gamma = 25\%$, and the size of the subset $|\mathcal{T}| = 50$. In these batches the TLDFs classifier shows improvements over the DFs classifier. The improvement is not uniform for all the gestures of the batches, but only for some of them. This is because not all the gestures can benefit from the training data of the source tasks. Only the gestures that have, at least, one similar gesture in a source task show improvements.

The confusion matrix for the *devel*06 batch in Figure 7 shows significant improvements in the classification of the last gesture. Figure 8 shows a representative image of that gesture and similar gestures in the *devel*13 and *devel*15 batches. The person in front of the camera

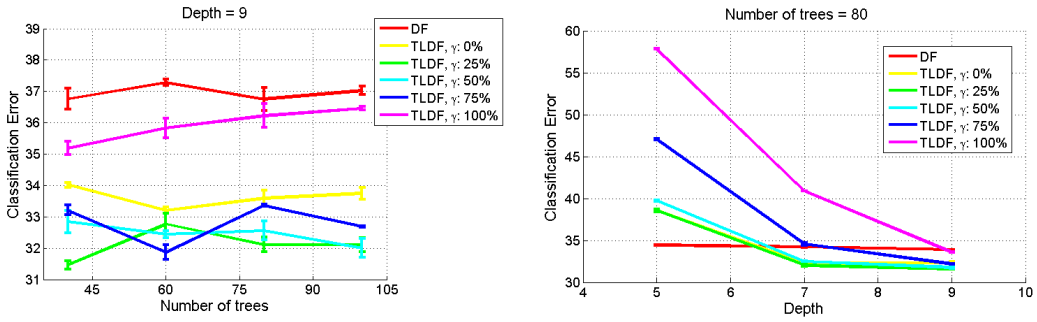


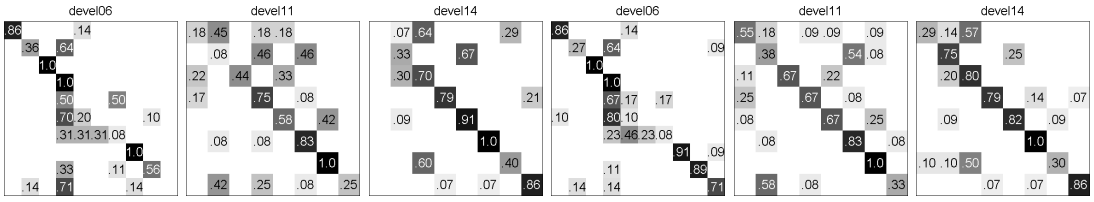
Figure 6: Comparison of the classification error using different combination of training parameters.

moves the left hand to a fixed position and then shows a similar pattern of the fingers, for all these gestures. The frames of these gestures are usually found in the same leaf after training the decision forest.

5.1.2 Devel and Final Data

Table 2 compares our results for the development batches of the ChaLearn Challenge with the ones previously reported by Lui (2012) and Malgireddy et al. (2013), using the evaluation procedure of the ChaLearn competition (Guyon et al., 2012). To train the TLDFs, we set the number of trees $T = 50$, the maximum depth $D = 8$, the mixing coefficient $\gamma = 25\%$, and the size of the search space $|\mathcal{T}| = 50$. As shown in Table 2, for most batches, our transfer learning decision forests obtain improvements over the DFs, and for some batches, they obtain the smallest errors.

Table 3 compares our results for the final evaluation data with the final results of the ChaLearn competition (Guyon et al., 2013). The Joewan team proposed a novel feature which fuses RGB-D data and is invariant to scale and rotation (Wan et al., 2013). Most of the other teams have not described their approach in a publication.



(a) (b) (c) (d) (e) (f)

Figure 7: Comparison of the confusion matrices obtained using the DF (a),(b),(c) and TLDF (d),(e),(f) classifiers on the *devel*06, *devel*11 and *devel*14 batches.

Figure 8: Similar gestures in different batches. The first, second and third rows show a gesture in the *devel*06, *devel*13 and *devel*15 batches respectively. The first column shows the RGB image for a representative frame of the video, the second column shows the corresponding depth image and the last column shows the MHI.

	devel01	devel02	devel03	devel04	devel05	devel06	devel07	devel08	devel09	devel10
Principal motion	6.67%	33.33%	71.74%	24.44%	**2.17%**	43.33%	23.08%	10.11%	19.78%	56.04%
Lui (2012)	–	–	–	–	–	–	–	–	–	–
Malgireddy et al. (2013)	13.33%	35.56%	71.74%	**10.00%**	9.78%	**37.78%**	18.68%	**8.99%**	**13.19%**	**50.55%**
DF	4.44%	28.89%	65.22%	25.56%	3.26%	48.89%	19.78%	17.98%	19.78%	59.34%
TLDF	**3.89%**	**25.00%**	**62.50%**	13.89%	4.89%	45.00%	**14.29%**	10.11%	15.38%	60.99%

	devel11	devel12	devel13	devel14	devel15	devel16	devel17	devel18	devel19	devel20	Avg.
Principal motion	**29.35%**	21.35%	12.50%	39.13%	40.22%	34.48%	48.91%	44.44%	60.44%	39.56%	33.15%
Lui (2012)	–	–	–	–	–	–	–	–	–	–	**24.09%**
Malgireddy et al. (2013)	35.87%	22.47%	**9.09%**	28.26%	**21.74%**	31.03%	**30.43%**	40.00%	**49.45%**	**35.16%**	28.73%
DF	42.39%	23.60%	19.32%	45.65%	26.09%	31.03%	53.26%	40.00%	60.44%	46.15%	34.14%
TLDF	39.13%	**19.10%**	25.00%	**27.71%**	31.52%	**27.01%**	45.11%	**38.33%**	54.95%	67.22%	31.55%

Table 2: Comparison of reported results using the Levenshtein distance.

Team	Private score set on final set #1	For comparison score on final set #2
alfnie	0.0734	0.0710
Joewan	0.1680	0.1448
Turtle Tamers	0.1702	0.1098
Wayne Zhang	0.2303	0.1846
Manavender	0.2163	0.1608
HIT CS	0.2825	0.2008
Vigilant	0.2809	0.2235
Our Method	0.2834	0.2475
Baseline method 2	0.2997	0.3172

Table 3: ChaLearn results of round 2.

	1/-1	2/-2	3/-3	4/-4	5/-5	6/-6
Adaboost (Faddoul et al., 2012)	91.77±1.89%	83.14±2.35%	82.96±1.24%	83.98±1.41%	78.42±0.69%	88.95±1.60%
MTL (Quadrianto et al., 2010)	96.80±1.91%	69.95±2.68%	74.18±5.54%	71.76±5.47%	57.26±2.72%	80.54±4.53%
MT-Adaboost (Faddoul et al., 2012)	96.80±0.56%	86.87±0.68%	87.68±1.04%	**90.38±0.71%**	84.25±0.73%	92.88±0.90%
Our approach	**97.23±0.44%**	**96.74±0.31%**	**93.29±0.96%**	90.10±1.23%	**92.79±1.62%**	**97.35±0.45%**

	7/-7	8/-8	9/-9	0/-0	Avg.
Adaboost (Faddoul et al., 2012)	87.11±0.90%	77.51±1.90%	81.84±1.85%	93.66±1.29%	84.93%
MTL (Quadrianto et al., 2010)	77.18±9.43%	65.85±2.50%	65.38±6.09%	97.81±1.01%	75.67%
MT-Adaboost (Faddoul et al., 2012)	92.81±0.57%	85.28±1.73%	**86.90±1.26%**	97.14±0.42%	90.10%
Our approach	**95.55±1.39%**	**91.99±1.30%**	84.76±1.67%	**98.05±0.28%**	**93.78%**

Table 4: Comparison of the accuracies on the MNIST data set.

5.2 MNIST

The MNIST (LeCun et al., 1998) data set has been used to compare transfer learning results (Quadrianto et al., 2010; Faddoul et al., 2012). A small sample of the training set is used to simulate the situation when only a limited number of labeled examples is available. For each digit $0 \dots 9$, we consider a binary task where label $+1$ means that the example belongs to the digit associated with the respective task, and label -1 means the opposite. We randomly choose 100 training samples for each task and test them on the $10,000$ testing samples. The experiments are repeated ten times and the results are summarized in Table 4. We train the TLDFs with $D = 6, T = 40, \gamma = 50\%$, and we do not apply any preprocessing to the sample images. The experiments show that our approach achieves better results than state-of-the-art methods in terms of transfer learning.

To analyze the influence of the number of training samples, we compare the classification error of the TLDFs with the classification error of the DFs. Figure 9 plots the classification error as a function of the number of training samples for each classifier. As we did previously, we compute the classification error using the 10000 test samples of the MNIST data set. We see that the classification error of the TLDF is smaller than that of the DF. In addition, it is interesting to note that the gap between both classifiers is larger when the number of training samples is smaller, thus suggesting that the TLDF is more suitable than DF for small training samples.

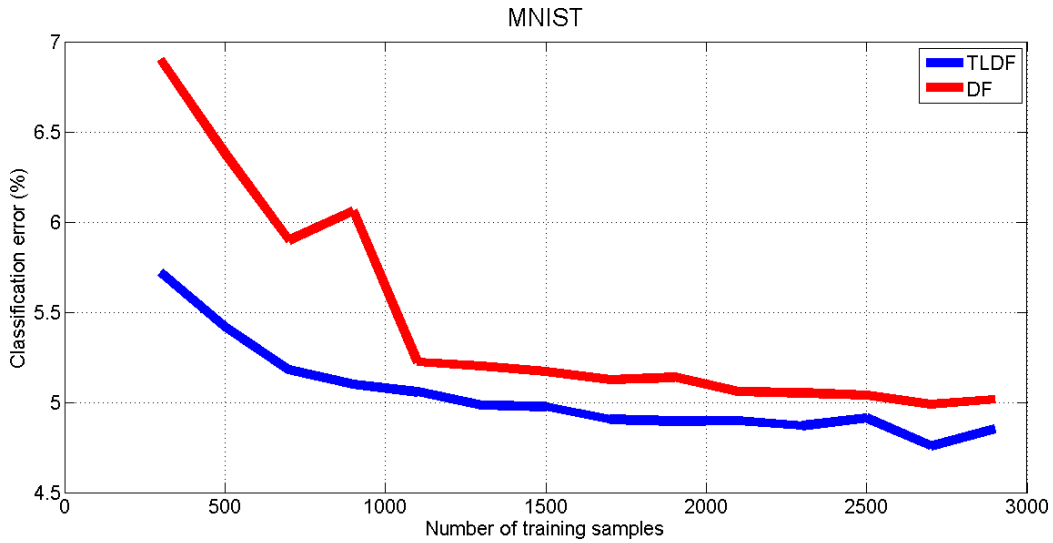


Figure 9: This figure evaluates the classification error as a function of the number of training samples.

6. Conclusions

In this paper we have introduced a novel algorithm to transfer knowledge from multiple source tasks to a given target task. The result is a classifier that can exploit the knowledge from similar tasks to improve the predictive performance on the target task. Two extensions were made to the decision forest framework in order to extract knowledge from the source tasks. We showed that both extensions are important in order to obtain smaller classification errors. The major improvements are obtained when there are only a few training samples.

We have applied the algorithm to two important computer vision problems and the results show that the proposed algorithm outperforms decision forests (which are a state-of-the-art method). We believe that transfer learning algorithms will be an essential component of many computer vision problems.

Acknowledgements

We would like to thank Zicheng Liu and Julio Jacobo-Berlles for their feedback and assistance.

Appendix A

We prove Theorem 1. First, we prove $\mathbb{E}(\mathcal{H}(S_K)) + \sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log\left(1 + \frac{1-p_{\mathbf{y}}}{K p_{\mathbf{y}}}\right) \leq \mathcal{H}(P)$

By definition of the empirical entropy and linearity of the expectation, we have:

$$\mathbb{E}(\mathcal{H}(S_K)) = -\mathbb{E}\left[\sum_{\mathbf{y}\in\mathcal{Y}} \hat{p}_{S_K}(\mathbf{y}) \log(\hat{p}_{S_K}(\mathbf{y}))\right] = -\sum_{\mathbf{y}\in\mathcal{Y}} \mathbb{E}\left[\hat{p}_{S_K}(\mathbf{y}) \log(\hat{p}_{S_K}(\mathbf{y}))\right]$$

Using the definitions of the empirical histogram $\hat{p}_{S_K}(\mathbf{y})$ and the expectation:

$$-\sum_{\mathbf{y}\in\mathcal{Y}} \mathbb{E}\left[\hat{p}_{S_K}(\mathbf{y}) \log(\hat{p}_{S_K}(\mathbf{y}))\right] = -\sum_{\mathbf{y}\in\mathcal{Y}} \sum_{j=0}^{K} P\left(\hat{p}_{S_K}(\mathbf{y}) = \frac{j}{K}\right) \frac{j}{K} \log \frac{j}{K}$$

Assuming that the samples are iid, then:

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} \sum_{j=0}^{K} \binom{K}{j} p_{\mathbf{y}}^{j} (1-p_{\mathbf{y}})^{K-j} \frac{j}{K} \log \frac{j}{K}$$

Note that, in this equation, $p_{\mathbf{y}}$ is the true probability of distribution P. After some algebraic manipulations, we obtain the following:

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \sum_{j=0}^{K-1} \binom{K-1}{j} p_{\mathbf{y}}^{j} (1-p_{\mathbf{y}})^{K-1-j} \log \frac{j+1}{K}$$

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \sum_{j=0}^{K-1} P\left(\hat{p}_{S_K}(\mathbf{y}) = \frac{j}{K}\right) \log \frac{j+1}{K}$$

Applying Jensen's inequality for the convex function $-\log(x)$, we obtain:

$$\geq -\sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log \left(\sum_{j=0}^{K-1} P\left(\hat{p}_{S_K}(\mathbf{y}) = \frac{j}{K}\right) \frac{j+1}{K} \right)$$

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log \frac{(K-1)p_{\mathbf{y}} + 1}{K}$$

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log \left(p_{\mathbf{y}} + \frac{1-p_{\mathbf{y}}}{K} \right)$$

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log \left(p_{\mathbf{y}} \left(1 + \frac{1-p_{\mathbf{y}}}{K p_{\mathbf{y}}} \right) \right)$$

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log p_{\mathbf{y}} - \sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log \left(1 + \frac{1-p_{\mathbf{y}}}{K p_{\mathbf{y}}} \right)$$

$$= \mathcal{H}(P) - \sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log \left(1 + \frac{1-p_{\mathbf{y}}}{K p_{\mathbf{y}}} \right)$$

Now we prove $\mathcal{H}(P) \leq \mathbb{E}(\mathcal{H}(S_K))$.

By definition of the empirical entropy and linearity of the expectation, we have:

$$\mathbb{E}(\mathcal{H}(S_K)) = -\mathbb{E}\left[\sum_{\mathbf{y}\in\mathcal{Y}} \hat{p}_{S_K}(\mathbf{y}) \log(\hat{p}_{S_K}(\mathbf{y})) \right] = -\sum_{\mathbf{y}\in\mathcal{Y}} \mathbb{E}\left[\hat{p}_{S_K}(\mathbf{y}) \log(\hat{p}_{S_K}(\mathbf{y}))\right]$$

Applying Jensen's inequality for the convex function $x \log x$, we obtain the following:

$$\leq -\sum_{\mathbf{y} \in \mathcal{Y}} \mathbb{E}\left[\hat{p}_{S_K}(\mathbf{y})\right] \log(\mathbb{E}\left[\hat{p}_{S_K}(\mathbf{y})\right])$$

Since $\mathbb{E}\left[\hat{p}_{S_K}(\mathbf{y})\right] = p_{\mathbf{y}}$, we have:

$$= -\sum_{\mathbf{y} \in \mathcal{Y}} p_{\mathbf{y}} \log(p_{\mathbf{y}}) = \mathcal{H}(P)$$

References

M. A. R. Ahad, J. K. Tan, H. Kim, and S. Ishikawa. Motion history image: its variants and applications. *Machine Vision and Applications*, 23(2):255–281, 2012.

Y. Aytar and A. Zisserman. Tabula rasa: model transfer for object category detection. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2011.

E. Bart and S. Ullman. Cross-generalization: learning novel classes from a single example by feature replacement. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2005.

A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 181–189, 2010.

A. Bobick and J. Davis. An appearance-based representation of action. In *Proceedings of the International Conference on Pattern Recognition*, pages 307–312, 1996.

A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine*, 2001.

R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and J. M. Brady. A linguistic feature vector for the visual interpretation of sign language. In *Proceedings of the European Conference on Computer Vision*, 2004.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006.

A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3):81–227, 2012.

W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of the International Conference on Machine Learning*, pages 193–200, New York, NY, USA, 2007.

J. B. Faddoul, B. Chidlovskii, R. Gilleron, and F. Torre. Learning multiple tasks with boosted decision trees. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, 2012.

A. Farhadi, D. Forsyth, and R. White. Transfer learning in sign language. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 1–8, 2007.

L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine*, 28(4):594–611, 2006.

P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine*, 32(9):1627–1645, 2010.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: an unsupervised approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 999–1006, 2011.

E. Grosicki and H. E. Abed. ICDAR 2011 - French handwriting recognition competition. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 1459–1463, 2011.

I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H. J. Escalante. ChaLearn gesture challenge: design and first results. In *Workshop on Gesture Recognition and Kinect Demonstration Competition*, 2012.

I. Guyon, V. Athitsos, P. Jangyodsuk, H. J. Escalante, and B. Hamner. Results and analysis of the chalearn gesture challenge 2012. In *Advances in Depth Image Analysis and Applications*, volume 7854 of *Lecture Notes in Computer Science*, pages 186–204. Springer, 2013.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2003.

A. Kurakin, Z. Zhang, and Z. Liu. A real-time system for dynamic hand gesture recognition with a depth sensor. In *Proceedings of the European Signal Processing Conference*, pages 1980–1984, 2012.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

C. Leistner, A. Saffari, J. Santner, and H. Bischof. Semi-supervised random forests. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 506–513, 2009.

K. Levi, M. Fink, and Y. Weiss. Learning from a small number of training examples by exploiting object categories. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshop*, pages 96–102, 2004.

W. Li, Z. Zhang, and Z. Liu. Graphical modeling and decoding of human actions. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing*, pages 175–180, 2008.

J. J. Lim, R. Salakhutdinov, and A. Torralba. Transfer learning by borrowing examples for multiclass object detection. In *Proceedings of the Advances in Neural Information Processing Systems*, 2011.

J. Liu, K. Yu, Y. Zhang, and Y. Huang. Training conditional random fields using transfer learning for gesture recognition. In *Proceedings of the IEEE International Conference on Data Mining*, pages 314 – 323, 2010.

Y. M. Lui. Human gesture recognition on product manifolds. *Journal of Machine Learning Research*, 13:3297–3321, Nov 2012.

M. R. Malgireddy, I. Nwogu, and V. Govindaraju. Language-motivated approaches to action recognition. *Journal of Machine Learning Research*, 14:2189–2212, 2013.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010. ISSN 1041-4347.

Y. Pei, T.-K. Kim, and H. Zha. Unsupervised random forest manifold alignment for lipreading. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

N. Quadrianto, A. J. Smola, T. Caetano, S. Vishwanathanand, and J. Petterson. Multitask learning without label correspondences. In *Proceedings of the Advances in Neural Information Processing Systems*, 2010.

A. Quattoni, M. Collins, and T. Darrell. Transfer learning for image classification with sparse prototype representations. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 1 – 8, 2008.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proceedings of the European Conference on Computer Vision*, pages 213–226, 2010.

H. J. Seo and P. Milanfar. Action recognition from one example. *IEEE Transactions on Pattern Analysis and Machine*, 33(5):867–882, 2011.

T. Sharp. Implementing decision trees and forests on a GPU. In *Proceedings of the European Conference on Computer Vision*, pages 595–608, 2008.

E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1331–1338, 2005.

T. Tommasi, F. Orabona, and B. Caputo. Learning categories from few examples with multi model knowledge transfer. *IEEE Transactions on Pattern Analysis and Machine*, 36(5):928–941, 2014.

A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine*, 29(5): 854–869, 2007.

P. A. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

J. Wan, Q. Ruan, W. Li, and S. Deng. One-shot learning gesture recognition from RGB-D data using bag of features. *Journal of Machine Learning Research*, 14:2549–2582, 2013.

Q. Wang, L. Zhang, M. Chi, and J. Guo. MTForest: ensemble decision trees based on multi-task learning. In *Proceedings of the European Conference on Artificial Intelligence*, pages 122–126, 2008.

J. won Lee and C. Giraud-Carrier. Transfer learning in decision trees. In *Proceedings of the International Joint Conference on Neural Networks*, 2007.

Y. Yao and G. Doretto. Boosting for transfer learning with multiple sources. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 1855 – 1862, 2010.