Adapting Classification Cascades to New Domains

Vidit Jain
Yahoo! Labs Bangalore
viditj@yahoo-inc.com

Sachin Sudhakar Farfade
Yahoo! Labs Bangalore
fsachin@yahoo-inc.com

Abstract

Classification cascades have been very effective for object detection. Such a cascade fails to perform well in data domains with variations in appearances that may not be captured in the training examples. This limited generalization severely restricts the domains for which they can be used effectively. A common approach to address this limitation is to train a new cascade of classifiers from scratch for each of the new domains. Building separate detectors for each of the different domains requires huge annotation and computational effort, making it not scalable to a large number of data domains. Here we present an algorithm for quickly adapting a pre-trained cascade of classifiers – using a small number of labeled positive instances from a different yet similar data domain. In our experiments with images of human babies and human-like characters from movies, we demonstrate that the adapted cascade significantly outperforms both of the original cascade and the one trained from scratch using the given training examples.

1. Introduction

Object detection is a problem of rare-event classification, where the positive examples are overwhelmed by a deluge of negative examples. Many of the negative examples are easy to separate from the positive examples, whereas the rest require a detailed analysis to do so. Therefore, instead of learning a single complex classifier, a cascade of classifiers with increasing complexity is often used. This cascade may employ several simple binary classifiers and accept a candidate image region as detection if and only if all of these binary classifiers accept it. For object classes with complex appearance models, e.g., faces, such cascades have been found to be very effective in reducing both the complexity of the detector and the processing time.

Once trained, a cascade classifier is often used in different, unconstrained data domains (or acquisition settings) with variations in appearances that may not be captured in the training examples. This classifier often fails to perform well in domains with even minor variations from the training examples. For instance, a cascade trained on the images of human faces only from a particular age group (e.g., adults) fails to detect faces from another age group (e.g., babies) (see Figure 1). This limited generalization of the cascade classifiers severely restrict the data domains for which they can be used effectively. A common approach to address this limitation is to train a new cascade of classifiers from scratch for each of the new domains. Training these multiple cascades is a formidable task. Not only do they require a long training time, but they also need a large collection of labeled – both positive and negative – instances. As we encounter more number of different domains, this approach becomes infeasible. Instead, we need an approach that can quickly adapt a pre-trained cascade to perform well on a new domain.

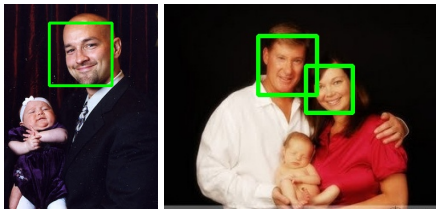


Figure 1. *Narrow range of detections*. In both of these images, a standard face detector correctly identified the adult faces but failed to detect the faces of the babies.

We consider the problem of domain adaptation for cascade classifiers when the positive examples available from the target class are not sufficient to train the cascade from scratch. Furthermore, we assume that only the pre-trained cascade is available, and *not* the data used for training it.[1] This setting of limited availability of training data in a new domain arises not only for object detection but also for several other rare-event classification problems such as medical diagnosis and intrusion detection. For some of these problems, domain adaptation and transductive learning of general classifiers have been explored, but adaptation techniques specific to cascade classifiers have not been studied.

[1] While it is common to make a pre-trained classification cascade available, it is sometimes not feasible to retain the examples used for training it due to operational and copyright issues.

We observe that the lack of robustness in the cascade classifiers is primarily due to their over-fitting to training examples. To address this issue, we split the trained cascade into three functional components, and devise appropriate adaptation techniques for these components. There are two main contributions in this paper: (a) a mathematical model that systematically identifies and removes the classifiers in a cascade that contribute little to detection in the new domain; and (b) an efficient generative model for an in-domain verification of the detected regions. These two models are used to adapt a pre-trained (base) classification cascade to a new domain with a few training examples.

In our experiments, we consider cascade adaptation for the problem of face detection. Here the different data domains arise from the appearances diversity across age groups, race, acquisition settings, and human-like characters in virtual environments and sci-fi or fantasy movies. Considering the abundance of such images in personal collections and online social networks, these variations present important, practical settings for face detection. For these settings, the detected faces are often used to improve the performance of subsequent tasks including multimedia search, video annotation and tracking, and moderation of offensive content in images and videos. It is challenging to collect representative training examples from all of these domains. Also, some of these domains gain significance too rapidly (e.g., after the release of a new sci-fi movie or a popular video game) to allow for a cascade to be trained from scratch. A system that can quickly build a face detector for a new target domain from a pre-trained face detector is useful for these new domains. In this work, we consider the set up where it is feasible to obtain only a few (hundred) positive examples of the target class, which are not sufficient to train an effective cascade classifier from scratch.

Sections 3 and 4 describe the details of our approach for adapting a pre-trained cascade. The image collections comprising faces of human babies and human-like characters from movies are presented in Section 5; the related improvement in detection performance are shown in Section 6.

2. Related Work

We first distinguish our work from the relevant work from the domain adaptation and transfer learning literature. Then we discuss some of the key research related to cascade classifiers and face detection.

Domain Adaptation. The problem formulation used in this paper is similar to the work in *domain adaptation*. In domain adaptation, labeled data from one or multiple "source" domains is used to train models to perform well on a different yet related "target" domain. Daumé and Marcu [6] approach this problem by modeling the data distribution for each of these domains as a mixture of a global and a domain-specific component. This global component is inferred from the data of the source domain(s) and applied to the data of the target domain. Another approach to the domain adaptation problem employs models trained on the data from the source domain to label a subset of the unlabeled data from the unlabeled target domain, and re-trains the classifier on the combined labeled data set [4]. Most of the work in domain adaptation (including the above two) suggests minimizing a convex combination of source and target empirical risk [10]. Thus the classifier needs to be re-trained (repeatedly) from scratch for every new domain. Similarly, most of the semi-supervised approaches [21] also require access to the original training data to adapt the learned model to a new data domain. In this paper, we consider the problem of domain adaptation without an access to the original training data.

Transfer learning. Another problem related to ours is transfer learning, where the general goal is to share knowledge across different learning problems and different data domains. Most of the work in this area can be broadly categorized into three types of transfer learning: inductive [16], transductive [1], and unsupervised [5]. The first category assumes that knowledge transfer is done between two different learning problems that share the same data domain. Whereas, the second category addresses a single learning problem for different source and target data domains. It further assumes that the data for the two domains is available during training (hence the term transductive). The third category considers the most general scenario where both the data domains and the learning problems can be different. Most of the work belonging to this category, however, only addresses unsupervised learning problems such as clustering and density estimation. Our setup is similar to the third category, however the learning problems are supervised classification problems.

Cascade classifiers. Cascade classifiers are commonly used for anomaly detection [8] and one-class classification [18]. The cascade classifier by Viola and Jones [18] is arguably the most popular solution for face detection. This detector not only achieves a high detection accuracy on standard face detection data sets, but is also known for its fast processing speed that makes it useful in practical applications. This classifier has been shown [2] to exhibit overfitting to the training examples. Bourdev et al.'s soft cascade [3] reduces the over-fitting issue by allowing the cascade to make decisions based on cumulative performance. These lazy decisions however compete with the computational efficiency of classification cascades. Saberian et al. [17] presented a formal framework to capture the tradeoff between speed and accuracy. Similar to previous cascade classifiers, their model also does not consider the generalizability of the trained classifier to other domains. Jain et al. [12] suggested the adaptation of a pre-trained classifier to a single image, and reported significant improvement

in face detection performance on the FDDB data set [11]. Their algorithm adapts a cascade classifier to a new data domain, but considers the same classification task, i.e., detection of human adult faces. There have been other similar studies that address different aspects of cascade classifiers (e.g., see Zhang's survey [20]). To our knowledge, none of them focuses on our set up of adapting a cascade classifier to a different but related classification problem.

3. Cascade adaptation

A cascade of classifiers $\mathcal{F}$ is a classifier that is composed of m *stage* classifiers $\{f_1, \ldots, f_m\}$ that are applied in a sequential manner. For computational efficiency, a rejection cascade is typically employed in rare-class classification tasks where the input is instantaneously rejected if it is rejected by any of these m classifiers. In face detection, we are given a candidate image patch x and it is classified as a face region if and only if it is accepted by all of these stage classifiers in the cascade.

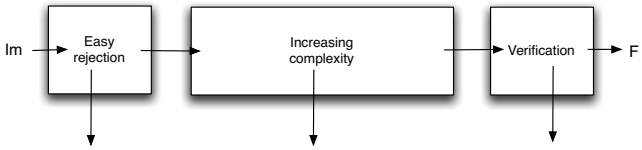


Figure 2. Phase-wise split of a cascade of classifiers.

Functionally, this cascade can usually be split into two phases: rejection of false positives and validation of true positives. The first phase corresponds to the early stages of the cascade that are designed to perform easy rejection and the subsequent stages of increasing complexity. The first step quickly discards the easy-to-reject examples, maintaining a high recall rate for positive examples. Because of this easy rejection, most of the computation is focused towards only a few candidates and therefore keeps the computation under control. In the second phase, the stage classifiers are very detailed and typically use several hundred features. These classifiers capture most of the structure in a face and can be considered similar to a descriptive model of face appearances. This interpretation of cascade classifiers is illustrated in Figure 2.

Below we present methods for adapting the two steps in the first phase of a cascade. A generative model for the second phase will be discussed in Section 4.

3.1. Training new stage classifiers

Compared to the later stages of the cascade, the first few stages $\{f_1, \ldots, f_h\}$ usually lack robustness to minor variations across similar classes. As a result, a large number of positive examples from a similar class are often rejected by these early stages. Since these early stages are expected to eliminate only the easy-to-reject instances, they can be trained effectively from scratch even with a few training examples. To this end, we train a short cascade with very few stages using the positive examples from the target class. The stage classifiers from this new cascade will become candidate replacements to the stage classifiers in an existing (generic) cascade classifier.

To maintain the computational efficiency of the original cascade, we consider the same family of classification functions to learn stage classifiers for the new cascade. Similar to Viola and Jones [18], a variant of AdaBoost learning algorithm is employed to train the individual stage classifiers from the few training examples from the target domain. This algorithm simultaneously selects from a collection of weak classification functions and combines them to form a stronger classifier. Here we use Haar-like rectangular features to form the pool of weak classifiers and use the desired rates for hit rate and false alarm as the stopping criteria for the learning algorithm.

3.2. Selecting from existing stage classifiers

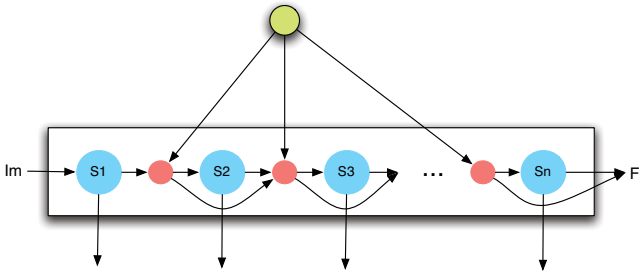


Figure 3. *Classification cascade with stage selection*. Each of the stage classifiers (blue) has a binary selection variable (red) associated with it. These selection variables share the relaxation parameters (green).

The second step of the rejection phase is composed of an ordered set of stage classifiers $\{f_{h+1}, \ldots, f_t\}$ of increasing complexity. If any of these stage classifiers rejects a given image patch, the patch is immediately discarded, otherwise it is evaluated by the next stage classifier. The increase in complexity of the subsequent classifiers is because the acceptable false-alarm and hit-rate for the trained classifier becomes stricter for subsequent stages.

As discussed earlier in Section 1, this step captures different structures of medium complexity in the appearance of the source domain. Since we assume that the target domain is similar to the source domain, we expect many of these structures to be shared across the two domains. By selecting only the stage classifiers that capture these shared structures, we can construct a new classification cascade for the target domain. To this end, we modify the pre-trained cascade (for the source domain) as follows. For each stage

in this cascade, we introduce a binary selection variable θ that specifies if the evaluation of this stage is useful for the target domain. This adapted cascade is illustrated in Figure 3. Note that since we are removing some intermediate stages from the given cascade, it is possible that the subsequent, expensive stages are evaluated for more candidate windows, thereby leading to a decrease in the processing time. On the contrary, as reported later in Section 6, we observed an increase in the processing speed in our experiments. Our observations validate the existence of stage classifiers in the pre-trained cascade that are ineffectual for the target domain. Now we present the details of the parameter estimation for the proposed selection of useful stages in a given cascade.

3.2.1 Preliminaries

Let $-r$ denote the complement of a binary random variable r. We denote the combined output of the set of functions $\{f_i, \ldots, f_j\}$ applied in order in a cascade as $\mathcal{F}_{i,j}$. Using $u(\cdot)$ to represent a step function and $u(f)$ to represent $u(f(\cdot))$, we have

$$\mathcal{F}_{1,2} = f_1 u(-f_1) + u(f_1) f_2, \quad (1)$$

$$\mathcal{F}_{1,m} = \sum_{i=1}^{m} f_i u(-f_i) \prod_{j=1}^{i-1} u(f_j) + \mathcal{F}_{k+1,m} \prod_{j=1}^{k} u(f_j) \quad (2)$$

$\forall k, 1 \leq k < m$. For clarity, let us denote the two summation and product terms in the above equation by A_{k+1} and B_{k+1} respectively. Thus, we have

$$\begin{aligned} \mathcal{F}_{1,m} &= A_{k+1} + \mathcal{F}_{k+1,m} B_{k+1} \quad (3) \\ &= A_k + f_k u(-f_k) B_k + u(f_k) B_k \mathcal{F}_{k+1,m}. \end{aligned}$$

Note that A_k and B_k do not depend on the function f_k.

Our modified cascade use binary variables $\{\theta_1, \ldots, \theta_m\}$. We represent these variables as individual step functions over respective continuous random variables $\{\alpha_i, \ldots, \alpha_m\}$. Formally, this modification implies

$$\begin{aligned} \theta_i &= u(\alpha_i), \quad (4) \\ \tilde{f}_i &= 1 + u(\alpha_i)[f_i - 1], \quad (5) \end{aligned}$$

where $\tilde{f}_i$ corresponds to the i^{th} stage classifier in the new cascade. We further denote the sequential application of functions $\{\tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_m\}$ in the new cascade as $\tilde{\mathcal{F}}_{1,m}$.

3.2.2 Loss function

To learn the parameters $\boldsymbol{\alpha}$ of the above model, we want to minimize the empirical exponential-loss with appropriate $L1$ regularization for the parameters

$$\mathcal{L}_{\tilde{\mathcal{F}}}(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \exp[-y_i \tilde{\mathcal{F}}(x_i, \boldsymbol{\alpha})] + \lambda \sum_{j=1}^{m} \alpha_j. \quad (6)$$

Assuming the set $\{f_i\}$ is known and fixed, we are interested in estimating parameters we are that minimize the above loss-function

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \mathcal{L}_{\tilde{\mathcal{F}}}(\boldsymbol{\alpha}). \quad (7)$$

To avoid undesirable minima with arbitrarily large magnitude of α, we constraint the solution space to $\boldsymbol{\alpha} \in [-c, c]^m$. Solving this minimization is NP-hard because the expression for the loss function includes several instances of the step function $u(\cdot)$. Therefore we employ a differential approximation for the step function. The details of the modified minimization function are presented below.

3.2.3 Differentiable approximation

We approximate the step functions $u(\cdot)$ as

$$u(\tilde{f}) \approx \frac{1}{2}[\tau(\sigma \tilde{f}) + 1] \quad \text{and} \quad u(\alpha) \approx \frac{1}{2}[\tau(\eta\alpha) + 1], \quad (8)$$

where σ and η are a relaxation parameters and $\tau(\cdot) = tanh(\cdot)$. The partial derivative of the cascade function is given by

$$\frac{\partial \tilde{\mathcal{F}}}{\partial \alpha_k} = \frac{1}{2} B_k \{[1 - \tau(\sigma \tilde{f}_k)] + \sigma[\tilde{\mathcal{F}}_{k+1} - \tilde{f}_k][1 - \tau^2(\sigma \tilde{f}_k)]\} \frac{\partial \tilde{f}}{\partial \alpha_k}, \quad (9)$$

where $\frac{\partial \tilde{f}}{\partial \alpha_k} = f_k \eta [1 - \tau^2(\eta \alpha_k)]$. The derivative of the optimization criterion is given by

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = -\sum_{i=1}^{N} \exp[-y_i \tilde{\mathcal{F}}(x_i, \boldsymbol{\alpha})] y_i \frac{\partial \tilde{\mathcal{F}}(x_i, \boldsymbol{\alpha})}{\partial \alpha_k} + \alpha_k. \quad (10)$$

We apply a gradient descent algorithm to obtain a solution $\boldsymbol{\alpha}^*$ for our optimization. Applying individual step functions on different components of this solution, we obtain the model parameters $\boldsymbol{\Theta}^*$. In our experiments, we use cross-validation to determine λ and the relaxation parameters σ and η.

4. Generative validation

Now we discuss our proposal for adapting the second phase of the cascade i.e., $\{f_{t+1}, \ldots, f_m\}$. We postulate that this phase is functionally performing a validation of the hypothesized detection through a detailed matching of the detailed structures present in the instances of the given object class. This functionality can alternatively be obtained using a generative modeling of the appearances of the given class instances. Although the selected generative model should require similar computational effort as the last phase of the original cascade. Furthermore, this generative model should learn effective parameters from only a few examples.

In this work, we propose the use of a nearest-neighbor based probabilistic model in a projection space as the generative model for validating the hypothesized detections. In brief, we first compute a projection space suitable for a robust representation of the instances for the given object class. Then we use a kernel density estimator over the k nearest neighbors for each of the projected training instances. Finally, for a new test instance, we use this probability density estimator to determine the likelihood of the test instance to belong to the object class. The details of each of these three steps are given below.

We learn the projection space as the basis obtained from the non-negative matrix factorization (NMF) [13] of the training images X from the target domain. The standard NMF model factorizes X as

$$X = WH + \epsilon, \tag{11}$$

where H is the latent projection space, W gives the projection coefficients, and ϵ denotes the noise. Compared to the principal component analysis based projection, an NMF based is more interpretable and more sparse due to its non-negative additive nature. Sparsity is key to our work as we want the projection to be computationally efficient. Several other variants of the NMF models (e.g., sparse-NMF [14] and NMF with explicit sparseness constraints [7]) have been proposed. These models were found to be rather unstable in our experiments.

In this learned subspace, we employ a non-parametric density estimator using the k-nearest neighbors for each of the training examples.

$$\zeta_i = \arg\max_{\zeta} \prod_{z \in \mathcal{N}_k(x_i)} p(d(z, x_i)|\zeta), \tag{12}$$

where $d(z, x_i)$ is the distance between z and x_i, and $\mathcal{N}_k(x_i)$ denotes the set of k nearest neighbors of x_i.

For a test instance x^*, the validation probability is compute as

$$p(x^*|X_{train}) = \frac{1}{norm} \sum_{i=1}^{nTrain} p(x^*|\zeta_i). \tag{13}$$

5. Data sets

We consider two key application areas from the web domain: offensive content analysis (OCA) and image search. OCA systems are critical for social networking and photo sharing websites that allow users to upload photographs. These automated systems are expected to detect and filter the pornographic content. Most of these systems employ the presence of exposed skin outside face regions as features. A significant failure case for such classifiers is that of the photos of babies. Not only do they contain a large number of skin pixels, but also the typical detectors fail to detect their faces. The abundance of baby photos shared on these websites adds significance to this data domain. Since the configuration of facial features for babies is different from normal adults [15], the problem of detecting baby face images is an example of adaptation to a similar class. A collection of 764 images of babies is annotated with face regions, which is referred to as *BabyFaces* data set. Figure 4 shows a few example images from this collection.

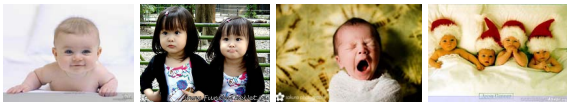


Figure 4. *BabyFaces data set.* Each image in this collection is annotated with the position and size of the faces of babies (and infants) appearing in them. Face regions of human adults are not included in the annotations.

Another relevant application domain is image search. When a new movie or a video game is released, there is a rapid increase in the queries for its characters, scenes, and wallpapers on images search. For instance, when the *Avengers* movie was released, at least $66K$ queries for characters from this movie (e.g., Iron Man, and the Incredible Hulk) were issued in the month of April 2012. Similar was the case with the Na'vi and Star Wars characters for the *Avatar* movie and the *Star Wars 1313* video game, respectively. Many image search engines employ the presence of faces to re-rank the retrieved images to better serve such queries. These systems need to be generic enough to be able to detect faces across huge variations – some of which may be non-human[2]. On the one hand, it is challenging to build a single system that can detect these different human-like faces with such diverse appearances. On the other hand, it is infeasible to employ separate detectors for individual characters due to the large number of labeled examples and the large computation time required to train these detectors.

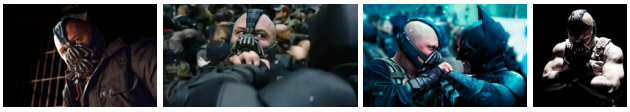
To assess the applicability of our algorithm to image search, we collected images for four different movie characters that are "human like" (see Figure 5). These collections are built by retrieving the top 1000 results for appropriate queries to the Bing image search engine. After filtering the broken links, we obtained 947, 959, 955, and 935 images in the four collections, respectively. We refer to the combined set of these four collections as *HumanLike* data set.

In each of the images in these collections, we annotated the regions corresponding to the respective faces. We also split each of these collections separately into five folds for cross-validation. As a result, we have less than 800 positive examples to train a pose-invariant face detector in each of our experiments.[3] A careful selection of 800 examples to

[2] In fact, the main characters in six of the top-10 highest-grossing hollywood movies of the year 2012 are non-human characters that have appearances with strong similarity to humans.

[3] Typical approaches for face detection employ up to $10-20K$ positive

(a) Na'vi people (Avatar)



(b) Bane (The Dark Knight Rises)

(c) Davy Jones (Pirates of the Carribean)

(d) Yoda (Star Wars)

Figure 5. *HumanLike data set.* This data set contains four sets, example images from each of which are shown in the four rows. The caption shows the name of the character followed by the name of the movie within the brackets. While the images may also contain human characters, here we are only interested in detecting the given character and hence, will treat any detections of human faces as false alarms.

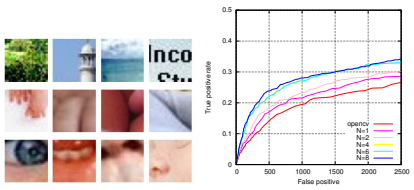
represent the variations for a simple face class (e.g., babies) may be feasible. However, selecting 800 clean and useful example images to represent the complex variations in pose, occlusions, extensions, resolution, and texture observed in images on the Internet is an painstaking task. These challenges are evidently present for the non-human characters. Clearly, this approach of relying only on careful selection is not scalable when we wish to generalize face detection to several of these new domains. That said, we use our best effort (e.g., bootstrapping, selection of negative examples) to achieve a trained cascade from scratch that comprises the first few stages of the final cascade.

6. Experiments

Below we present the detailed observations for the BabyFaces data set. The described methodology was followed for the experiments on the HumanLike data set as well; the general conclusions were the same for the two data sets.

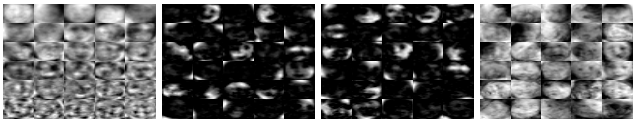
Our approach for cascade adaptation is a supervised approach – i.e., it requires both positive and negative examples from the target data domain. We take positive examples from the in-domain training set, whereas the negative examples are collected from three types of image regions: (a) general image regions, (b) in-face, in-domain negatives, and (c) out-of-face, in-domain negatives. Some of the negative examples are shown in Figure 6(a). Using these labeled examples, we trained new in-domain stage classifiers as candidate replacements for the start of the frontal face detection cascade (hereafter referred to as the original cascade) available with the OpenCV distribution. Figure 6(b) shows the improvement in performance for the different number of stage classifiers replaced in the original cascade. Based on these observations, we chose to replace the first eight stages of the original cascade with the in-domain stage classifiers.

examples per pose [20].



(a) (b)

Figure 6. *Training new stage classifiers.* (a) Three types of non-face image regions are selected as negative examples for training new stage classifiers: regions from non-face images, regions from outside the face, and small regions inside the face. Some examples are shown in the three rows, respectively. (b) The performance initially improves by introducing multiple in-domain, stage classifiers. But for the later stages, the newly trained classifiers are susceptible to over-fitting.



(a) PCA (b) NMF (c) SNMF (d) NMFsc

Figure 7. *Learned bases from different approaches*. PCA provides more dense basis vectors than the NMF variants. Also, the basis vectors in NMFsc were found to converge to individual examples in the training set. The basis vectors obtained from both of the NMF and SNMF models highlighted different parts of the face.

For stage selection, we found the useful values of the relaxation parameters σ and η to be close to 0.01 and 0.001, respectively. The solution space is constrained by ensuring that the value of any of the α variables $\in [-10, 10]$. Expanding this range had little effect on the eventual selection of stages, but it increased the number of iterations needed for convergence. The stage selection algorithm converged to recommend the rejection of 13^{th}, 14^{th} and 17^{th} stage of the cascade. Although the removal of these stages had little effect on the resulting performance curve, we observed an average reduction in $7.2 \pm 0.4\%$ in computation time.

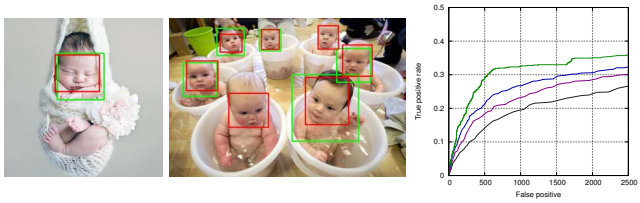
To compute an compact representation for generative validation, we consider the basis obtained using principal

component analysis (PCA), and three variants of NMF. Figure 7 shows the learned bases for the babies data set. As expected, PCA provides more dense basis vectors than the NMF variants. Also, the basis vectors in NMFsc were found to converge to individual examples in the training set. The basis vectors obtained from both of the NMF and SNMF models highlighted different parts of the face. Considering the computational advantage of the NMF model, we chose this basis for the rest of the experiments. We represent a given image region as its projection in the space spanned by these basis vectors. A kernel density estimator is then learned using a Gaussian kernel over the k-nearest neighbors for each of the training examples. In our experiments, we set the value of k to 10. To obtain similar computation cost, we drop the last few stages of the cascade that are computationally equivalent to the trained kernel density estimator. For a test image region, the final output of the adapted cascade is computed as a linear combination[4] of the score from the cascade and the validation score from the generative model.

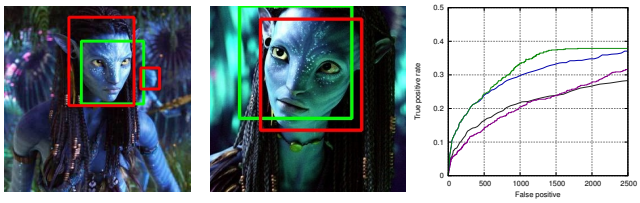
Figure 8 shows the results for the above data sets. In all of the image collections, the original cascade is significantly outperformed by the adapted cascade. A cascade trained from scratch using the training examples correspond to the first few stages of the cascade (Section 3.1). Since these stages only serve the purpose of easy-rejection, we observe a large number (tens of thousands) of false positives for this cascade; the true positive rates are close to zero for $< 5K$ false positives. Also, even though the detection rates are low (on an absolute scale) for Yoda and Bane, they make a significant impact for image search (where the user may only look at the top few images from the retrieved set of thousands of images). Also, in videos, these detection rates can be significantly improved by tracking detections across frames [19]. As shown in Figure 9, the detections from the adapted cascade are expected to improve more than those from the original cascade. Table 1 shows the comparison of the average computation time for different approaches.

Animation and masked faces. We also experimented with characters from animation movies such as Anton Ego from the movie Ratatouille (see Figure 10). The faces of such characters have very few edge and gradient features. So the Haar-like features employed in our cascade are not very effective for detecting these face regions. The example of the masked Spiderman is the opposite case, where the edges are uniformly distributed in all orientations and across the face regions, and the facial features are occluded. For these characters, cascade adaptation showed no improvement over the original face detector. We believe our algorithm will also be useful for detectors that employ region-based features. Generative validation also showed little improvement in the performance for these characters;

[4]learned through cross-validation



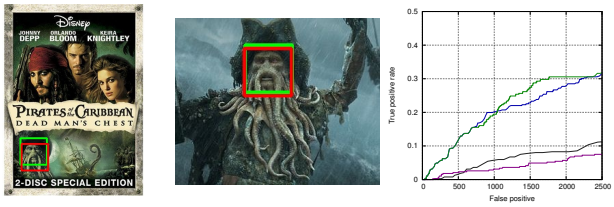
(a) Baby photos



(b) Avatar Na'vi

(c) Yoda (Star wars)

(d) Bane (The Dark Knight Rises)



(e) Davy Jones (Pirates of the Carribean)

Figure 8. (Cols 1-2) Ground truth (red) and the output of our detector (green) are shown. Performing at the same false positive rate, the original cascade did not detect any of these face regions. (Col 3) Performance curves using the FDDB discrete matching score [11]: original cascade (**black**), original + validation (**magenta**), original + adaptation (**blue**), and original + adaptation + validation (**green**). The standard techniques for post-processing will improve the performance curves further uniformly for all of these approaches. However, since our main contribution is an approach for adapting the detector to work better in a new domain, we focus only on the algorithmic improvement in performance.

	Original+Adapt	Original+Validate	Original+Adapt+Validate
BabyFaces	-8.4 ± 0.6	0.2 ± 1.1	-4.5 ± 1.4
Na'vi people	-2.8 ± 1.3	4.2 ± 0.9	3.0 ± 0.5
Yoda	0.3 ± 0.2	8.2 ± 0.8	10.0 ± 0.4
Bane	-4.0 ± 0.9	9.3 ± 2.0	7.1 ± 0.4
Davy Jones	-7.7 ± 0.1	8.7 ± 0.6	-1.6 ± 2.0

Table 1. *Computational cost*. Each number is the percentage difference between the average detection CPU time of the approach relative to that of the original cascade. (Lower values are better.) The proposed approach is never slower by more than 11% compared to the original cascade, while obtaining performance gains of more than 100% in some cases.

Figure 9. *Video frame sequence*. Adaptive cascade (bottom row) obtains more robust detections than the original cascade (top row). This robustness would further allow a tracking algorithm to discard spurious false positives (e.g., in the third frame). The complete video examples are included in the supplementary material.

Figure 10. *Failure cases.* The proposed approach showed little improvement in performance for animation and masked faces.

the true positive rate at 2500 false positives increased by only 0.5% for the Spiderman collection.

7. Discussion

We presented an approach for adapting a cascade of classifiers to perform classification in a similar domain for which only a few positive examples are available. Using this approach, we demonstrated huge gains in performance in detecting faces of human babies and human-like characters from movies. Also we maintain the computational efficiency of the original classification cascade. Given a few labeled examples of a target domain, this approach constructed an effective detector for this domain within a day. Training similar detector from scratch would typically require several days of annotation and computational efforts for training these classification cascades. It would be interesting to extend our approach to handle noisy training data that can be obtained in a semi-supervised or unsupervised manner. Another interesting future direction would be to explore the use of unsupervised alignment techniques [9] for improving our approach for adaptation.

References

[1] A. Arnold et al. A comparative study of methods for transductive transfer learning. In *ICDMW*, 2007.

[2] B. Babenko, P. Dollár, and S. Belongie. Task specific local region matching. In *ICCV*, 2007.

[3] L. Bourdev and J. Brandt. Robust object detection via soft cascade. In *CVPR*, 2005.

[4] D. Wu et al. Domain adaptive bootstrapping for named entity recognition. In *EMNLP*, 2009.

[5] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Self-taught clustering. In *ICML*, 2008.

[6] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *JAIR*, 2006.

[7] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *JMLR*, 2004.

[8] W. Hu, W. Hu, and S. J. Maybank. Adaboost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics*, 2008.

[9] G. B. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *ICCV*, 2007.

[10] J. Blitzer et al. Learning bounds for domain adaptation. In *NIPS*, 2008.

[11] V. Jain and E. Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical report, University of Massachusetts Amherst, 2010.

[12] V. Jain and E. Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *CVPR*, 2011.

[13] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2000.

[14] W. Liu, N. Zheng, and X. Lu. Non-negative matrix factorization for visual coding. In *ICASSP*, 2003.

[15] L.S. Mark et al. Wrinkling and head shape as coordinated sources of age level information. *J. Per. and Psy.*, 1980.

[16] R. Raina et al. Self-taught learning: Transfer learning from unlabeled data. In *ICML*, 2007.

[17] M. Saberian and N. Vasconcelos. Boosting classifier cascades. In *NIPS*, 2010.

[18] P. A. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 2004.

[19] X. Ren. Finding people in archive films through tracking. In *CVPR*, 2008.

[20] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical report, Microsoft Research, 2010.

[21] X. Zhu. *Semi-Supervised Learning with Graphs*. PhD thesis, CMU, 2005.