

Lightweight Automatic Face Annotation in Media Pages

Dmitri Perelman*
Technion
Haifa, Israel
dima39@techunix.technion.ac.il

Edward Bortnikov
Yahoo! Labs
Haifa, Israel
ebortnik@yahoo-inc.com

Ronny Lempel
Yahoo! Labs
Haifa, Israel
rlempel@yahoo-inc.com

Roman Sandler
Yahoo! Research
Haifa, Israel
romats@yahoo-inc.com

ABSTRACT

Labeling human faces in images contained in Web media stories enables enriching the user experience offered by media sites. We propose a lightweight framework for automatic image annotation that exploits named entities mentioned in the article to significantly boost the accuracy of face recognition. While previous works in the area labor to train comprehensive offline visual models for a pre-defined universe of candidates, our approach models the people mentioned in a given story on the fly, using a standard Web image search engine as an image sampling mechanism. We overcome multiple sources of noise introduced by this ad-hoc process, to build a fast and robust end-to-end system from off-the-shelf error-prone text analysis and machine vision components. In experiments conducted on approximately 900 faces depicted in 500 stories from a major celebrity news website, we were able to correctly label 81.5% of the faces while mislabeling 14.8% of them.

Categories and Subject Descriptors

I.4.9 [Computing Methodologies]: Image Processing and Computer Vision – *applications*

General Terms

Algorithms, Experimentation

Keywords

Face recognition, Text analysis, Web search, Machine learning

1. INTRODUCTION

The Web is a heaven of (and for) media outlets. Media sites of all shapes and forms, topics and languages, independent and part of a network, abound. Still, despite this variety, the prototypical online media story page revolves around an article that is typically accompanied by a centerpiece image (see Figure 1). True to the saying “a picture is worth a thousand words”, the centerpiece image is among

*Work done while interning at Yahoo! Labs, Haifa

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.

WWW 2012, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1229-5/12/04.



Figure 1: An example media page, with highlighted entity names and annotated faces.

the dominant attention hubs of media story readers, and media editors carefully select and position that image for each story. Another typical attribute of media stories, especially in large and diverse sites and portals, is that named entities mentioned in stories – in particular celebrities – are often linked to a “topic page” dedicated to that entity on the site.

Images embedded in HTML pages, and in particular in media stories, are typically clickable as a single unit – a click on any pixel of the image transfers the user to the same destination URL. However, nowadays it is relatively easy to associate different target URLs with different portions of an image, i.e. to designate different areas of the image as leading to different URLs.

We tap the capability above in the context of media stories that focus on people. The centerpiece image in such stories naturally contains faces, and our aim is to identify which person is depicted by which face, i.e. to associate names to the depicted faces. An immediate application of this capability is then to link each face in the image to a destination specific to the depicted person, just as is typically done for mentions of that person in text, without editorial intervention. To achieve this goal, we present FRUIT – a system for

Face Recognition Using Information reTreival. FRUIT combines face recognition techniques with entity extraction and ranking schemes in a maximum likelihood framework.

Unlike mainstream applications of face recognition technology, FRUIT does not build nor maintain a large database of celebrities' images. Rather, given a story and its center-piece image, it employs a lightweight and ad-hoc processing pipeline for associating faces with names, built mostly from off-the-shelf building blocks. At a high level, the process is as follows. We begin by detecting faces in the image, and by extracting named entities from the story's text. Each name is further scored by the prominence of the name in the story. Next, we send the extracted names as queries to an image search engine, retrieving a few dozen results per query. We treat those results as positive examples for the queried name, and using standard face similarity functions associate a probability between each name and each face. After further adjustment of these probabilities based on the score of each name, we solve a bipartite matching instance to arrive at the final, maximum likelihood assignment of names to faces. Furthermore, a post-processing step assigns a confidence score to each name-face assignment, allowing us in low confidence cases to leave a face unnamed rather than to wrongly associate it with some name.

Each step in the processing pipeline above may introduce noise that impacts our success rates. Face detection in images suffers from false positives, false negatives and misalignment (bounding boxes that do not correctly capture the face). Entity extraction is imperfect, and even if it were, not every face in the image is necessarily mentioned in the text of the story. Image search engines are not very precise, and even images that are of the queried name and thus considered as good results for human consumption, might be ill-suited for face recognition tasks. In particular, the results of image search for people with a small Web footprint are fairly random.

Despite the above obstacles, our experiments – over a corpus of approximately 500 stories from Yahoo! OMG¹, a major site of entertainment news – show that FRUIT exhibits a robust behavior. For example, in one operating point it correctly associates names to 81.5% of faces, while associating false names with 14.8% of faces. The error rate can be significantly decreased, at the expense of the cumulative fraction of labeled images. We analyze our results along several axes by performing sensitivity analysis to multiple design parameters of our system, and isolating the noise introduced by its individual components.

Performance measurements demonstrate that an optimized version of our algorithm can run in less than 10 sec per media page. This low overhead renders FRUIT image annotation practical for use in media publishing pipelines.

The contributions of this work are the following:

- We present FRUIT – an accurate, light-weight, ad-hoc process for associating names with faces appearing in images of media stories.
- Unlike traditional face recognition approaches in machine vision, FRUIT does not require large and well-maintained databases of carefully chosen, high quality pictures for each encountered person. Rather, we show that accurate recognition can be achieved by leverag-

ing a small number of results returned by image search engines. In a sense, the Web – as mediated by image search engines and accessed through public APIs – is our training set.

- FRUIT is built from careful integration of standard, off-the-shelf tools for text and image analysis. Our innovation lies in how we put those tools to work together to overcome the noise each of them introduces.

The rest of this paper is organized as follows. Section 2 surveys the related work. Section 3 overviews the system design and the details of its components. Section 4 exemplifies the algorithm's behavior on a complex real-world page. The algorithm's accuracy is extensively evaluated in Section 6 on the dataset described in Section 5. We discuss the system's performance bottlenecks and the ways to eliminate them in Section 7, and conclude in Section 8.

2. RELATED WORK

Labeling faces on the Web is profoundly different from traditional face recognition, which originally was limited to controlled environments. In that context, both the training and test sets used to be part of carefully selected and manually labeled datasets (e.g., Yale [3] and FERET [15]). These data collections featured high-resolution images taken in studio conditions (in a variety of handpicked poses, illumination conditions, etc.). The classical face recognition algorithms (*Eigenfaces* [19], *Bunch Graph* [22], etc.) have been designed to work under these conditions.

In contrast, classifying faces on the Web exposes the process to extremely diverse image qualities. Moreover, ad-hoc learning from faces on the Web, alongside its obvious advantages, introduces even more noise. For example, image search by name provides only a weak guarantee that the results are relevant for the query, as well as a basic promise of image quality. It fetches a very small sample, which is susceptible to name ambiguity and person popularity. Therefore, the probability of face models being contaminated by wrongly labeled images is high.

A number of research efforts try to bridge the gap between the two extremes described above. The Labeled Faces in the Wild (LFW) project [10], which is constrained to a closed dataset like [3, 15], deals with variable-quality (albeit still manually labeled) images. In this work, we use two techniques – face alignment [9] and Three-Patch Linear Binary Patches (TP-LBP) [23] – that have been developed to tackle the image quality challenges. We deliberately avoid over-optimizing the image processing part, since our goal is demonstrating the feasibility of achieving acceptable robustness with standard building blocks. For example, we do not present our experiments with more advanced face recognition methods (e.g., [16]) that achieve better precision but are prohibitive for performance reasons.

Some works addressed the challenge of automatically labeling the training examples in images [4, 14] and video frames [6], using the related text. They analyze a large corpus of text and multimedia, and associate entities with visual information by clustering their co-occurrences in media pages. While this approach works well for popular entities, labeling the infrequent ones is more challenging [12]. We adopt a complementary path, by outsourcing labeled image sampling (ad-hoc training) to Web search.

¹<http://omg.yahoo.com>

Image search was previously employed for sampling face data, in the context of video annotation [24, 13]. Both works focus on the single-person recognition use case, whereas our solution labels multiple candidates simultaneously. Their application domain is also different from ours (video analysis versus single-shot recognition), which offers a lot of redundancy and simplifies the classification.

Our work capitalizes on exploiting auxiliary signals to enhance traditional face recognition. Namely, we demonstrate a boost in recognition rates stemming from a simple ranking of named entities extracted from the text. This result is in concert with earlier work by Gallagher et. al. [8] that highlighted the possibility of using image-related structured information, e.g., the age and gender of the persons in the image, for classification without any visual training data.

The literature on topic detection and tracking in text is abundant, including textbooks [1]. Our entity scoring function is deliberately unsophisticated, since our goal is a simple proof of concept. For example, we use popular signals like term frequency and order of occurrence, but do not learn advanced text segmentation models that have been shown useful to improve the quality of many IR tasks [18, 20].

Face recognition capabilities (also called automatic album tagging) have been recently added to popular photo sharing sites, e.g., Google Picasa² and Facebook³. These applications require manual tagging of example images, and apparently restrict themselves to recognizing people within a small reach in the social network. In this sense, our application is more generic.

3. SYSTEM OVERVIEW

Figure 2 illustrates FRUIT – a framework for automated face annotation in media pages.

The processing pipeline works as follows. At the first step, two actions happen independently: (1) named person entities are extracted from the article’s text, and (2) human faces are detected in the associated image. The extracted named entities become *candidates* (further denoted by C) for matching to the discovered faces (denoted by F).

Following this, the process diverges into two tracks: (1) *static scoring* of candidates, which estimates the prior probabilities for each candidate to match some face in the image, and (2) pairwise *similarity scoring*, which computes the match probabilities for each face-candidate pair. The first path is based entirely on text analysis, and is relatively straightforward. The second one captures visual relatedness. The idea behind it is (1) sampling a set of representative images for each candidate through web search, and (2) comparing these ad-hoc visual models with the detected faces through a machine vision algorithm.

Finally, we feed the static scores and the similarity scores into a matching component, which associates each detected face with at most one entity mined from the text. This process is cast as maximum-likelihood assignment, which translates to matching in the bipartite face-entity graph.

FRUIT receives many noisy inputs, therefore it is essential to introduce some internal resilience mechanisms. Our evaluation (Section 6) shows that even naïve static scoring is es-

sential for focusing the context, and thus reducing the noise introduced by the face recognition component. In order to further reduce the probability of false matches, we install a post-matching filter that drops annotations from faces that are more similar to some false candidate randomly sampled from the Web than to the original assignment.

Figure 2 depicts this flow, using an example of a news story about Bart Simpson catching a three-eyed mutated fish. The entities mentioned in the text are Homer, Bart, and Lisa Simpson, whereas the detected faces belong to Bart, Lisa, and the fish (false detection), respectively. The image sampling process might fetch some wrong images – e.g., one Marge’s image is returned for the query “Homer”. The static scores and the similarity scores are used to match faces to entities, which correctly labels Bart and Lisa, but Homer is misclassified. This false label is eventually dropped due to low confidence.

In what follows, we extend on the technical details of each of the building blocks – face detection (Section 3.1), entity extraction and (static) scoring (Section 3.2), face-entity similarity scoring (Section 3.3), and eventually the matching (Section 3.4) and post-match filtering (Section 3.5) phases.

3.1 Face Detection

The face detection procedure is used by multiple parts of our system. We employ an OpenCV⁴ implementation of the well-known Viola-Jones face detection algorithm [21], without any modification or parameter tuning. The minimum size of a detectable face in a media image is set to 50×50 pixels – a small value, stemming from the fact that the images themselves can be of variable size and quality.

3.2 Entity Extraction and Scoring

FRUIT uses a simple standard entity extraction procedure. The entities are extracted through (1) a shallow parsing of text and HTML markup, and (2) looking up the parsed terms in a dictionary or taxonomy.

We project that there is a close correlation between an entity’s relevance in the text and the probability of appearing in the associated image. Similarly to previous studies on primary topic detection [1], we rank the entities mentioned in the text by (1) their frequencies (denoted T) and (2) the relative order of their occurrences (denoted R). We assume that these two random variables are independent, and model the probability of candidate c appearing in the image as (denoted $P(c)$) as a product of the two conditional probabilities $P(c|R = r)$ and $P(c|T = t)$. The latter are modeled as sigmoid distributions:

$$P(c|R = r) = \frac{1}{1 + e^{-2(r-1.5|F|)}},$$

and

$$P(c|T = t) = \frac{1}{1 + e^{-2(t-0.5\tilde{T})}},$$

where \tilde{T} is the average frequency of named entity occurrences in the text.

In other words, the favorably scored entities are (1) among the topmost $1.5|F|$ (have a chance to fit in the image, with a small slack), and (2) appear no less than $0.5\tilde{T}$ (not too infrequent). For example, if two faces are detected in the

²<http://picasa.google.com/support/bin/answer.py?answer=156272>

³<http://www.facebook.com/blog.php?post=467145887130>

⁴<http://opencv.willowgarage.com>

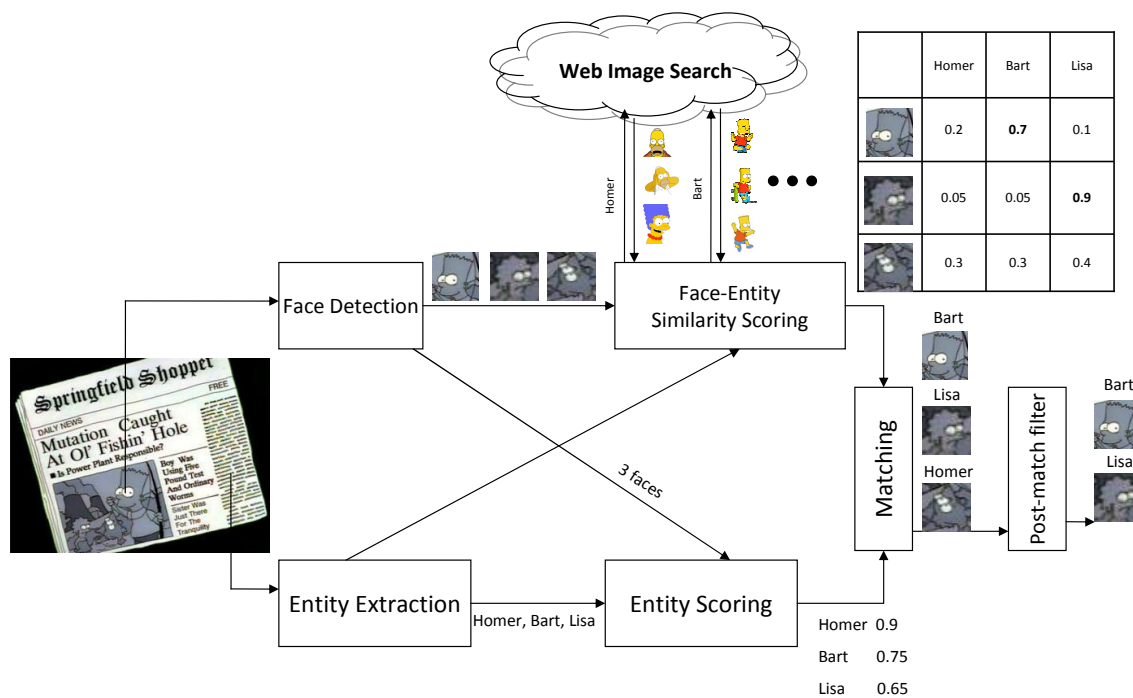


Figure 2: The system architecture and the processing flow for a particular example. Note that the system overcomes the errors of its individual components, e.g., a face mis-detection and a mis-assignment of a face to a candidate.

image, and all candidate names appear exactly once in the text, then the first three will de-facto compete for the match.

The distributions’ parameters have been selected heuristically, without any prior training, to exemplify the applicability of our framework to arbitrary datasets. We don’t claim the optimality of this simple scoring function in any respect – in fact, a more careful fine-tuning might positively impact the ultimate annotation goal.

3.3 Face-Entity Similarity Scoring

The pairwise similarity scoring procedure receives a set of candidate entities extracted from the text, and a set of face images detected in the associated picture. It returns a value between 0 and 1 for each face-candidate pair, which captures the probability of the face belonging to the candidate. The procedure can be roughly partitioned into the following steps:

Similarity Scoring:

1. Sample a small set of representative images for each candidate from a Web search engine.
2. Map the faces from the media page and from the sampled images to a low-dimension feature space via a face recognition algorithm.
3. For all face-candidate pairs, compute similarity scores – a distance metric which reflects proximity between a point (face) and a set of points (samples) in the feature space.

We now describe each of these steps in more detail.

3.3.1 Sampling through Image Search

We use a public image search engine API to retrieve a set of images by entity name. Large and wallpaper-size images are preferred, to exclude poor-quality pictures. No additional facets (e.g., filter by color, being a photo, containing faces, etc.) are used. This set is further pruned to a subset of “portrait-quality” images that are fit for modeling a human face. An image is defined portrait-quality if it contains a single face no smaller than 100×100 pixels⁵. We use the Viola-Jones algorithm [21] (Section 3.1) for face detection. The search result set contains up to 35 URL’s, which we retrieve top-down until δ portrait-quality images are fetched.

Note that the sampling phase is inherently noisy. In the first place, it suffers from entity name ambiguities. Beyond that, search engines often return related person images alongside the real candidate’s ones, especially for candidates with a small footprint on the Web. The further stages of our algorithm therefore need to compensate for this noise.

3.3.2 Face Recognition

Both the analyzed and the sampled face images undergo a pre-processing stage, which includes careful bounding box alignment [9], scaling to the uniform size of 64×64 pixels, transformation to grayscale, and intensity histogram equalization [7].

Following this, a machine vision algorithm is employed to map all faces to points in a moderate-dimensionality feature space (tens to thousands of dimensions), in which the proximity relationships between the original images are pre-

⁵In contrast with images as small as 50×50 pixels that we allow on media pages, Section 3.1.

served. The centerpiece requirement for the face recognition technique is computational efficiency. In this context, we experiment with three off-the-shelf tools. We used the modern Three Patch - Local Binary Pattern (TP-LBP) algorithm [23] as the main solution. For baseline we used features created with the classical **Eigenfaces** technique [19], and a fast approximation of the Earth Moving Distance (EMD) algorithm [17]. In minimum detail, they work as follows.

Eigenfaces maps each image to a space induced by all the sampled images. This space is concisely captured by its d first eigenvectors. An arbitrary image is therefore expressed as a linear combination thereof – i.e., a d -coordinate point. We set $d = \min(4|C|, 30)$ – typically, the number of dimensions does not exceed 15.

The earth moving distance between two images is the minimum cost of turning one image into the other, where the cost is defined to be the amount of bright pixels to be moved times the geometric distance by which they are moved. EMD is expected to be less sensitive to image misalignments than **Eigenfaces**. We map each image to a vector of EMD distances to the sampled images. The resulting space’s dimensionality is typically below 100 dimensions.

TP-LBP employs local texture descriptors for representing an image. The features are histograms of local texture descriptor values calculated in rectangular regions tessellating the face image. The concatenation of these histograms is the image’s footprint (a 1024-dimension vector in our case). In contrast with **Eigenfaces**, each image can be processed individually to create the representation.

3.3.3 Similarity Computation

We interpret the *similarity* between face f and candidate $c \in C$, denoted $S(f, c)$, as the probability of a multi-label classifier associating the face with the candidate’s name. We explore two such classifiers – k -nearest neighbor (kNN) and multi-class support vector machine (MC-SVM).

The kNN classifier defines the face-candidate similarity as $S(f, c) \sim e^{-D(f, c)}$ (the sum is normalized to 1), where $D(f, c)$ is a distance function between the face and the candidate sample image set. $D(f, c)$ is defined as the average L_2 distance in the feature space between f and c ’s k closest samples. We set $k = 3$.

MC-SVM is a popular multi-label classifier [5]. Given a face representation f , MC-SVM simultaneously computes $S(f, c)$ for all c , i.e., the vector of probabilities of f belonging to each of the candidate clusters. The algorithm defines its L_2 distance function internally. We employ the implementation from `libsvm`⁶, parameterized with a Gaussian kernel.

3.4 Matching

FRUIT seeks an injective mapping $\sigma : F \rightarrow C$ that matches each face to a single candidate⁷. We approach the computation of this mapping as a maximum-likelihood estimation problem, which maximizes the joint probability of $\sigma(f)$ being a correct label for all $f \in F$.

Section 3.2 defined static entity scores $P(\cdot)$ as an estimate for the distribution of a-priori probabilities of entities appearing in the media page image. Therefore, for face f and candidate c , the product $S(f, c)P(c)$ reflects the likelihood

of matching f to c . Assuming that probability distributions $S(f, \cdot)$ are all independent for all f ⁸, the likelihood of a particular mapping σ is

$$\mathcal{L}(\sigma) = \prod_{f \in F, c = \sigma(f)} P(c)S(f, c).$$

The corresponding log-likelihood is therefore

$$\log \mathcal{L}(\sigma) = \sum_{f \in F, c = \sigma(f)} (\log P(c) + \log S(f, c)).$$

This formulation natively translates to min-cost matching in a complete bipartite graph (F, C) , in which the cost of edge (f, c) is $-(\log P(c) + \log S(f, c))$. This classical problem can be solved, e.g., by the Hungarian algorithm [11], which runs in negligible time on small graph instances.

3.5 Post-Filtering

The matching phase might falsely assign faces to candidates. The reasons can be numerous – e.g., wrong samples might be fetched from the Web, the person on the image might not be either of the candidates, a non-face patch of the image might be wrongly identified as a face, or a detected face might be challenging to recognize for the vision algorithm. The post-filtering phase eliminates these wrong assignments, as follows.

We validate each matched pair $(f, \sigma(f))$ through a procedure similar to police lineup. Namely, f is compared to the images of 5 random *false candidates*. If f is significantly closer to one of these candidates than to $\sigma(f)$ – the $(f, \sigma(f))$ mapping is canceled.

Technically, the false candidates are randomly chosen from the domain taxonomy, and pre-processed offline, identically to the flow described in Section 3.3 For each $\bar{c} \in \bar{C}$, f undergoes a *binary* classification which discriminates between two classes – $\sigma(f)$ and \bar{c} (we use a standard binary SVM implementation in `libsvm`). The classifier outputs the likelihood of f belonging to $\sigma(f)$ in this context, denoted $\mathcal{L}(\sigma(f)|\bar{c})$. We define the *confidence* of $\sigma(f)$ as

$$\Theta(\sigma(f)) = \min_{\bar{c} \in \bar{C}} \mathcal{L}(\sigma(f)|\bar{c}).$$

Face f is de-labeled if $\Theta(\sigma(f))$ is below some user-defined threshold θ .

4. ILLUSTRATIVE EXAMPLE

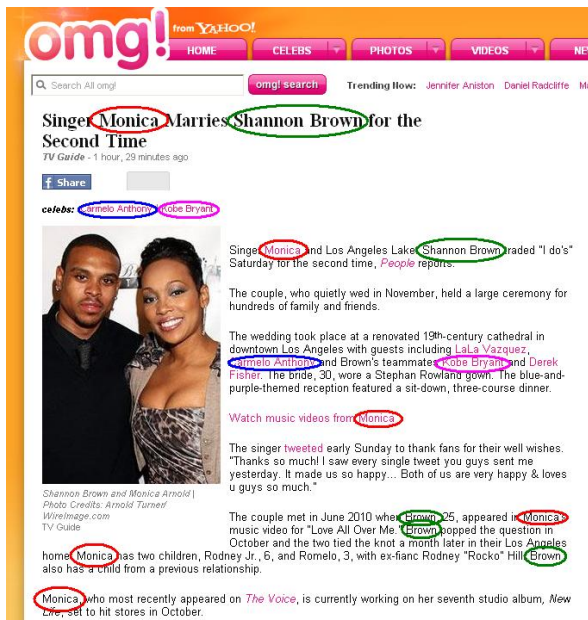
This section is a guided walkthrough of FRUIT’s operation on a real media page from the Yahoo! OMG celebrity gossip website (Figure 3(a)), which is correctly processed by our framework. The story references several persons – Monica (Monica Arnold’s stage name), Shannon Brown, Kobe Bryant, and Carmelo Anthony. All characters are mentioned more than once. The centerpiece image pictures two of the above candidates.

Computing the similarity scores is challenged by multiple obstacles. In the first place, the “Monica” named entity (derived from the text) is an ambiguous query. Most of the images retrieved for this name from image search belong to a different person (Monica Bellucci, see Figure 3(d)). Moreover, the images of the correct Monica are taken from a bad angle. Hence, the data points of the “Monica” class are

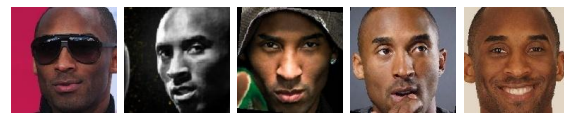
⁶<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.

⁷The injection property is satisfied by the vast majority of media images, in which no person appears twice. The exceptions are rare – e.g., artistic collages.

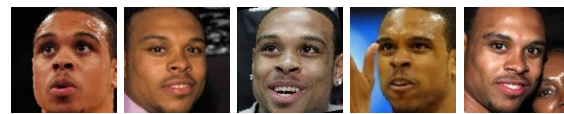
⁸E.g., the pairwise distance function in Section 3.3.3 satisfies this assumption.



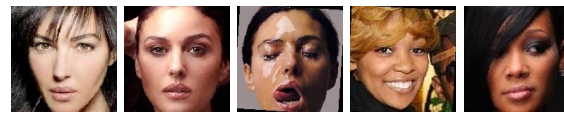
(a) The page with highlighted named entities



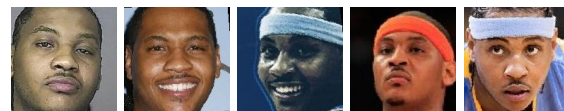
(b) Kobe Bryant



(c) Shannon Brown



(d) Monica



(e) Carmelo Anthony

Figure 3: A media page and the sampled candidate images.

highly dispersed in the feature space. (Note that trying to build a “consistent model” from a cluster of the dominating images, similarly to [24], would have resulted in complete filtering out of the correct person).

Another true character in the image, Shannon Brown, raises a different problem. He has a large footprint on the Web, however the top images returned by the search engine are replicas of the same few shots. Therefore, the classification ends up scoring Shannon less similar to himself than to Monica (probably due to high diversity of her sample images) as well as to Kobe Bryant (Table 1).









	Static Score	Similarity Score		Final	
					
	0.048	0.189	0.064	0.009	0.003
	0.982	0.167	0.164	0.164	0.162
	0.998	0.509	0.700	0.507	0.698
	0.084	0.136	0.072	0.011	0.006

Table 1: Static and Similarity Scores Computed by FRUIT.

These problems are fixed by the static scoring component. Monica and Shannon are referenced first in the text, and appear more than the other candidates. Hence, their static scores are significantly higher than Kobe’s and Cameron’s (see Table 1), therefore compensating for the image processing algorithm’s error. All in all, “Monica” is the most likely label for both faces in the image. However, since the solu-

tion is an injective matching, the most similar (right) face is labeled “Monica” while the other (left) face is labeled with the second best option – “Shannon Brown” – which is the correct labeling.

5. EVALUATION DATASET

Below we describe the dataset we use for evaluating FRUIT. Our experiments are conducted on a set of 500 pages randomly sampled from a few weeks’ feed of Yahoo! OMG – a major entertainment website⁹. From this ground set, we focus on 487 pages relevant for FRUIT – i.e., the stories with a centerpiece image that contains at least one detectable face of size 50 × 50 pixels or larger. 185 pages in the set contain one face, 266 pages contain two faces, and 36 pages are with three or more faces. Our experiment does not exploit the text captions associated with images, since they are not present in all media sites.

The taxonomy we use for entity extraction is assembled from multiple information sources, e.g., WordNet¹⁰, DBpedia¹¹, IMDB¹², etc.

FRUIT can correctly classify faces in the story image only if the corresponding people names appear in the article text. In our dataset, 94.3% of the people appearing in the pictures are referenced in the text. In this context, we count only direct references to person names – i.e., named entities are not inferred from composite semantic objects like movie casts, music band names, etc.

Figure 4 depicts the distribution of the number of named entities on a page, and its correlation with the number of faces in the image. The number of candidates varies from 2–3 (typical for the articles dedicated to specific persons) to

⁹The set’s size is limited only by our labeling capabilities.

¹⁰<http://wordnet.princeton.edu>

¹¹<http://dbpedia.org>

¹²<http://www.imdb.com>

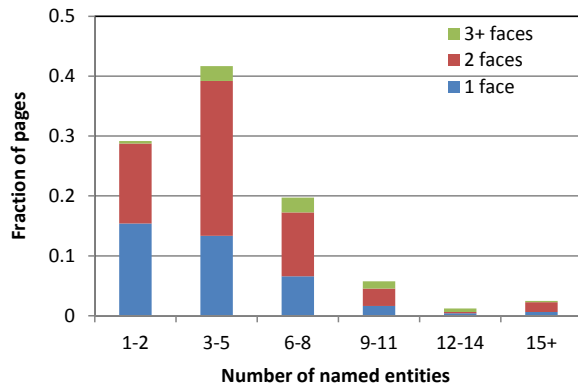


Figure 4: The distribution of the number of named entities per page. Most pages reference 3 to 5 named entities, however the distribution is heavy-tailed.

dozens (in stories about social events). In pages with moderate numbers of candidates, images with two faces prevail.

In Section 6, we separately study the algorithm’s behavior on pages varying by the number of candidates and faces, and explain the impact of its multiple design parameters.

6. EVALUATION

In this section we demonstrate and analyze the results achieved by FRUIT. We measure the annotation quality in terms of tradeoff between the correctly and wrongly labeled faces. The fractions thereof are denoted ρ and $\bar{\rho}$, respectively. (The ratio of unlabeled faces is therefore $1 - \rho - \bar{\rho}$).

In most of our experiments, the operating point is determined by the post-filtering confidence threshold θ , which determines when a previously matched face must be de-labeled (Section 3.5). We consider running θ ’s in the range $0 \dots 0.7$ (a higher θ corresponds to a more aggressive post-filtering).

The zoom is on applications that require high coverage (70% and more of totally labeled faces) to minimize editorial work. A wider range of considered thresholds could be useful for applications that prefer precision – e.g., a negligible error rate $\bar{\rho}$ at the expense of dropping the coverage to 40%. However, the phenomena highlighted by our study are qualitatively the same in that range as well.

6.1 Similarity Scoring

The similarity scoring component employs a computer vision algorithm for mapping the faces to a low-dimensional space, and a multi-class classification algorithm for computing the likelihood probabilities (Section 3.3). This section motivates our choices for these algorithms, which we continue to exploit in the following sections.

Figure 5 depicts the comparison between the *Eigenfaces*, TP-LBP and EMD image processing method, in conjunction with the MC-SVM classifier. TP-LBP clearly outperforms the competition, which is consistent with the result reported for the LFW dataset by machine vision researchers [23].

We now turn to comparing the MC-SVM and kNN similarity classifiers (Figure 6), in conjunction with TP – LBP. MC-SVM supercedes kNN in most cases. Note a particular working point for $\theta = 0.2$ (Figure 6(a)), in which FRUIT labels correctly $\rho = 81.5\%$ faces with an error rate $\bar{\rho} = 14.8\%$ (3.7%

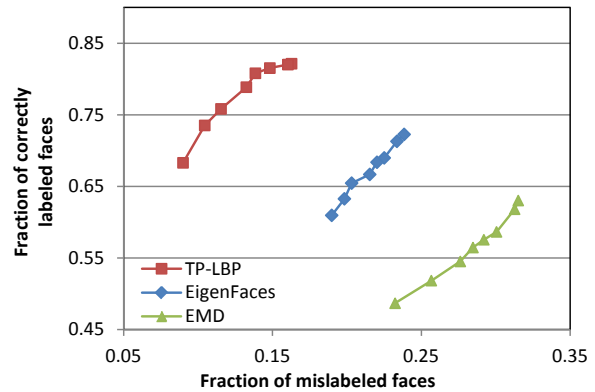


Figure 5: Comparison between the TP-LBP, Eigenfaces and EMD vision algorithms in conjunction with the MC-SVM classifier. The θ confidence threshold values run from 0 to 0.7. TP-LBP clearly outperforms the other competitors.

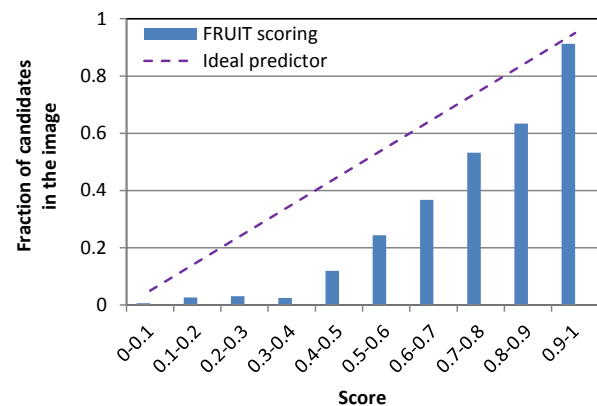


Figure 7: The quality of the FRUIT’s entity scoring. FRUIT’s scores weakly correlate with the ideal predictor of the entities’ probabilities of being part of the picture.

faces remain unlabeled). MC-SVM’s advantage is mostly pronounced in pictures with 3 and more faces, in which it outperforms kNN by 9% (Figure 6(b)). We revisit this phenomenon in Section 6.2.

FRUIT configured with TP-LBP and MC-SVM also improves over the previous works on face labeling in media stories [4, 14]. Similarly to us, these works exploit the article’s text in conjunction with the image, to enhance the recognition. They report lower precision (60% to 70%, in contrast with 66% to 84% for FRUIT), as well as lower recall, or the total fraction of labeled faces (below 70%, compared to 76% to almost 100% for FRUIT).

All the subsequent experiments use TP-LBP and MC-SVM as building blocks.

6.2 Entity Scoring

This section evaluates the impact of entity ranking (Section 3.2) on FRUIT’s results. We start from validating the basic assumption that entity scores predict the probabilities of respective entities appearing in the image. Namely, we mea-

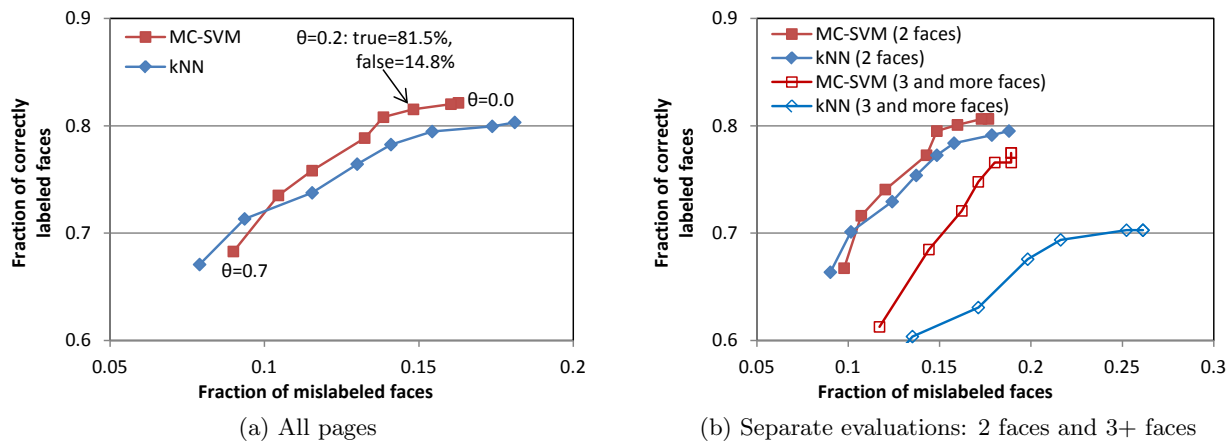


Figure 6: Comparison between the MC-SVM and kNN classifiers in conjunction with TP-LBP. The θ confidence threshold values run from 0 to 0.7. MC-SVM is more accurate than kNN, especially in pages with three and more faces.

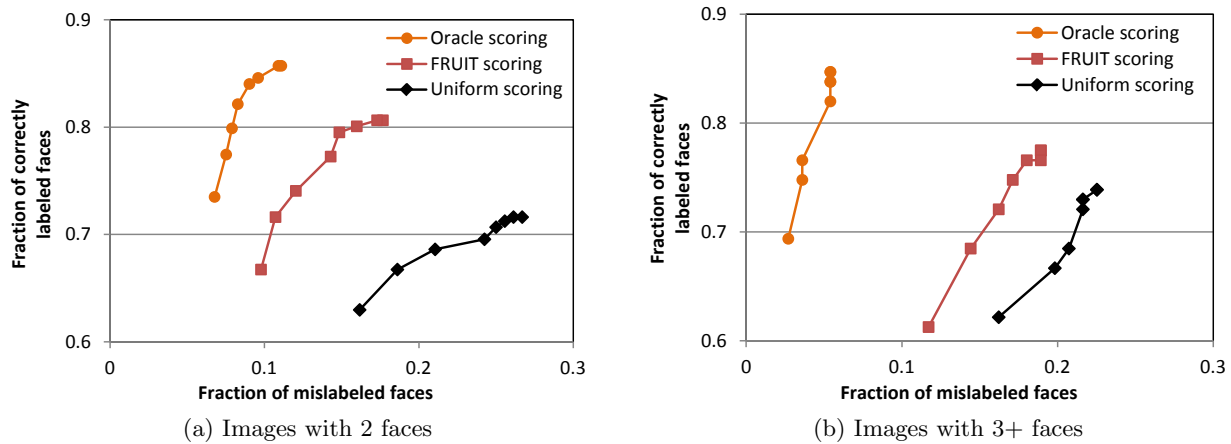


Figure 8: The impact of entity scoring on the system’s accuracy (all configurations run with TP-LBP and MC-SVM). The θ confidence threshold values run from 0 to 0.7. FRUIT improves the fraction of correctly labeled faces by more than 10% compared to a system that does not leverage entity relevance in the text (uniform scoring). However, there is a gap of approximately 7% between FRUIT and synthetic (oracle) scoring that perfectly predicts the identities of people in the image.

sure the fraction of candidates appearing in the images for different score ranges. For an ideal predictor, this fraction should be equal to the corresponding score (e.g., 70% of the candidates with score 0.7 appear in the pictures). Figure 7 shows the correlation between our method and the ideal predictor. FRUIT’s scoring is clearly imperfect, although it is close to the ideal in the higher bins. Note that the FRUIT scores are consistently below the ideal predictor’s dashed line. This is due to the use of sigmoid ranking functions with fairly generous slacks (see Section 3.2) that protect low-ranking candidates from being outruled. Using more advanced scoring methods (e.g., [1]) is likely to improve the predictor’s accuracy, as well as the overall results.

We are now ready to assess the bottom-line impact of FRUIT’s entity scoring. For this purpose, we compare it with two other methods, which provide the upper and lower bounds for the power of prediction. The upper bound is embodied by *oracle scoring* – a synthetic imaginary ranking in

which the named entities that appear in the image receive score 1, whereas the rest receive score 0. Note that this essentially reduces the number of candidates to the number of faces, hence the original matching problem transforms to finding the correct permutation of the candidates in the picture (a simpler de-noised variation). The lower bound is captured by *uniform scoring* that gives equal scores (e.g., 1) to all entities in the text. This is a degenerated scoring method that ignores entity relevance.

Figure 8 depicts the gaps in impact between FRUIT’s scoring versus the upper and lower bounds. For example, our heuristic beats the uniform scoring by 10%. We study the effect of scoring separately for images with two faces (Figure 8(a)) and three and more faces (Figure 8(b)). Note that the positive effect of FRUIT’s entity ranking is larger for the pages with less faces in the image. This happens because the number of highly scored candidates in FRUIT is proportional to the number of detected faces (Section 3.2). Therefore, in

this setting the classification algorithm must handle more irrelevant high-scoring candidates, thereby approaching the uniform scoring scenario.

Note that the accuracy of the system with both oracle and uniform rankings is surprisingly better for images with more faces. This happens because in an image with two faces, assigning a candidate whose face appears in the image to a wrong face automatically implies that the second assignment is incorrect as well. In contrast, in images with three faces, a wrong swap of two face-candidate matches still leaves room for one correct assignment.

6.3 FRUIT vs Bare Metal Face Recognition

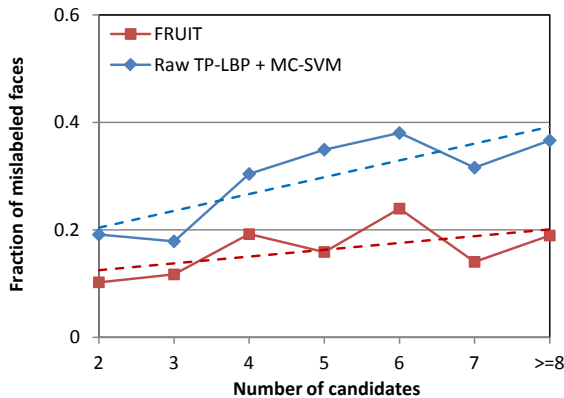


Figure 9: Comparison between FRUIT ($\theta = 0.2$) and raw face recognition (TP-LBP in conjunction with MC-SVM), for varying numbers of candidates in the text. We measure the fraction of mislabeled faces (lower is better). FRUIT is less sensitive to noise (the trendline is almost flat), and consistently prevails over the counterpart.

We conclude with underscoring our framework’s superiority to the bare-metal face recognition system with the same image processing part. Namely, we compare FRUIT to a system that runs TP-LBP in conjunction with MC-SVM, and assigns each face to the most similar candidate. Note that the latter algorithm does not necessarily produce an injective mapping, since two faces might be most similar to the same person (e.g., see Section 4). FRUIT is parameterized with $\theta = 0.2$ (less than 5% of faces remain unlabeled). We contrast the two alternatives by the ratio of false labelings, for the varying number of candidates in the text.

The results in Figure 9 show that FRUIT is more robust than the machine vision algorithm. In particular, it is less sensitive to noise (the number of entities in text). The gap grows with the number of candidates, reaching 16% for pages with more than 8 candidates.

7. SYSTEM PERFORMANCE

We evaluate our system on an 8-core Intel Xeon CPU running at 2.50 Ghz clock speed, with 32 GB of RAM. In what follows, we describe the breakdown of execution times of different stages of FRUIT’s pipeline, and suggest potential performance optimizations.

7.1 CPU Bottlenecks

The processing latency is heavily dominated by the alignment of the bounding box around faces detected in the sampled images [9]. A fine-grained alignment takes 3.2 sec per sample on average. Our experiments show, however, that this phase is most significant for images with 3 and more faces (7% of the dataset), for which it boosts the precision by 8%. For other images, its impact is negligible, hence the system designer might choose to skip this phase altogether.

The second-largest CPU consumer is face detection [21], which requires 150 to 200 msec per sample. The rest of the local processing overhead is insignificant.

7.2 I/O Overhead

The I/O latency is prevailed by download times, which range from 100 msec to 1.5 sec per image, depending on image size and hitting the Web caching services. It can be addressed through standard Web crawler optimizations, e.g., I/O parallelism [2].

Reducing the number of sampled images further decreases the I/O overhead. In this context, the crucial issue is deciding how many samples are enough to retain the annotation quality. Recall that FRUIT downloads up to 35 samples per candidate from image search, keeping only the images with one face detected (Section 3.3). Figure 10 studies FRUIT’s sensitivity to the number of candidate samples, δ . We consider $\delta = 5$, $\delta = 10$, and $\delta = 35$. Surprisingly, some pages can be successfully annotated even with 5 samples per candidate. The success rate for $\delta = 10$ is only 3% below the one for $\delta = 35$. This implies that the first ten faces retrieved from image search usually capture the data required for successful face recognition. The difference between images with two faces and images with three and more faces is immaterial in this context.

7.3 Summary

All in all, the average processing time of a media page on an 8-core CPU is within 30 sec, and can be reduced to less than 10 sec with judicious optimizations.

We consider face annotation within a given media page as a one-shot problem. Clearly, in a real setting the system might exploit the dataset locality, and cache the heavyweight phases’ intermediate results among multiple executions, approaching subsecond page processing times. However, this optimization edges our solution closer to the offline database approaches, e.g., [10].

8. CONCLUSIONS

This work demonstrates that accurate face labeling in Web media page images can be done in ad-hoc manner with real-time speeds. We show that the intuitions that guide humans for this task – searching online for several images of the most likely candidates – are applicable for a fully automatic technique as well. In particular, most named entities require no more than 10 image examples for being robustly recognized in media images.

Our face labeling framework, FRUIT, leverages web search for ad-hoc image sampling. It is constructed from simple off-the-shelf building blocks from the worlds of face recognition and information retrieval. We demonstrate that when carefully applied together, these components overcome significant volumes and miscellaneous types of noise that are inherent to the individual parts.

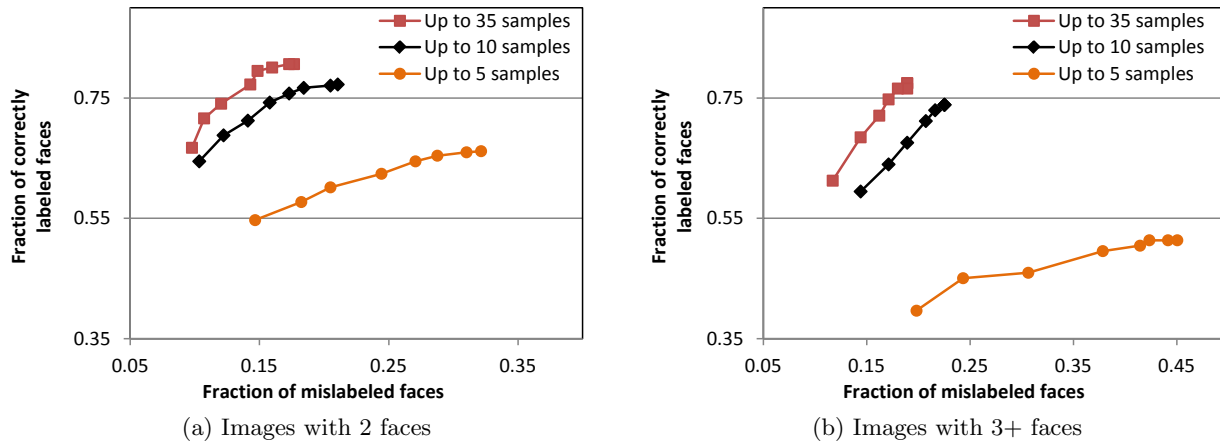


Figure 10: Sensitivity of FRUIT's accuracy to the number of samples per candidate, δ . The θ confidence threshold values run from 0 to 0.7. The success rate for $\delta = 10$ is only 3% below that for $\delta = 35$.

9. REFERENCES

- [1] J. Allan. *Introduction to topic detection and tracking*. Kluwer Academic Publishers Norwell, 2002.
- [2] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison Wesley, New York, NY, 1999.
- [3] P. N. Bellhumer, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *PAMI*, 17(7):711–720, 1997.
- [4] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y. W. Teh, E. Learned-Miller, and D. A. Forsyth. Names and faces in the news. In *CVPR*, June 2004.
- [5] K. Crammer and Y. Singer. On the Algorithmic Implementation of Multi-class SVMs. *JMLR*, 2001.
- [6] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy - automatic naming of characters in tv video. In *BMVC*, 2006.
- [7] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.
- [8] A. Gallagher and T. Chen. Estimating age, gender, and identity using first name priors. In *CVPR*, 2008.
- [9] G. B. Huang and V. Jain. Unsupervised joint alignment of complex images. In *ICCV*, 2007.
- [10] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.
- [11] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.
- [12] D. Le and S. Satoh. Unsupervised face annotation by mining the web. In *IEEE ICDM*, pages 383–392, 2008.
- [13] C. Liu, S. Jiang, and Q. Huang. Naming faces in broadcast news video by image google. In *ACM Multimedia*, pages 717–720, 2008.
- [14] D. Ozkan and P. Duygulu. A graph based approach for naming faces in news photos. In *CVPR*, 2006.
- [15] P. Phillips, H. Moon, and S. R. P. Rauss. The feret evaluation methodology for face recognition algorithms. *PAMI*, 22:1090–1104, 2000.
- [16] R. Sandler and M. Lindenbaum. Nonnegative Matrix Factorization with Earth Mover's Distance Metric for Image Analysis. *PAMI*, 33:1590–1602, 2011.
- [17] S. Shirdhonkar and D. Jacobs. Approximate earth mover's distance in linear time. In *CVPR*, 2008.
- [18] R. Song, H. Liu, J. Wen, and W. Ma. Learning block importance models for web pages. In *WWW*, 2004.
- [19] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3 (1):71–86, 1991.
- [20] S. Vadrevu and E. Velipasaoglu. Identifying primary content from web pages and its application to web search ranking. In *WWW*, 2011.
- [21] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.
- [22] L. Wiskott, J.-M. Fellous, N. Krüger, and C. V. D. Malsburg. Face recognition by elastic bunch graph matching. *PAMI*, 19:775–779, 1997.
- [23] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop in ECCV*, 2008.
- [24] M. Zhao, J. Yagnik, H. Adam, and D. Bau. Large scale learning and recognition of faces in web videos. In *CVPR*, 2008.