Computer Vision and Image Understanding xxx (2013) xxx–xxx

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Face recognition for web-scale datasets ☆

Enrique G. Ortiz [a,*], Brian C. Becker [b]

[a] Department of Electrical and Computer Engineering, University of Central Florida, 4000 Central Florida Blvd., Orlando, FL 32816, United States
[b] Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., NSH 4200, Robotics, Pittsburgh, PA 15213, United States

ARTICLE INFO

Article history:
Received 11 October 2012
Accepted 15 September 2013
Available online xxxx

Keywords:
Open-universe face recognition
Large-scale classification
Uncontrolled datasets
Sparse representations

ABSTRACT

With millions of users and billions of photos, web-scale face recognition is a challenging task that demands speed, accuracy, and scalability. Most current approaches do not address and do not scale well to Internet-sized scenarios such as tagging friends or finding celebrities. Focusing on web-scale face identification, we gather an 800,000 face dataset from the Facebook social network that models real-world situations where specific faces must be recognized and unknown identities rejected. We propose a novel Linearly Approximated Sparse Representation-based Classification (LASRC) algorithm that uses linear regression to perform sample selection for ℓ^1-minimization, thus harnessing the speed of least-squares and the robustness of sparse solutions such as SRC. Our efficient LASRC algorithm achieves comparable performance to SRC with a 100–250 times speedup and exhibits similar recall to SVMs with much faster training. Extensive tests demonstrate our proposed approach is competitive on pair-matching verification tasks and outperforms current state-of-the-art algorithms on open-universe identification in uncontrolled, web-scale scenarios.

1. Introduction

Face recognition is a well-researched field with a history that can be viewed as a journey of increasing scope, realism, and applicability to real-world facial analysis problems. Perhaps this journey is described best by the many datasets introduced over the years that addressed key challenges at the time of collection. Early datasets such as AT&T (ORL) [1], AR [2], Yale [3], FERET [4], and PIE [5] were collected in the laboratory to control and explore solutions for illumination, expression, age, pose, and disguise. In such tightly controlled environments, machine learning can match or surpass humans [6] and performance is often very good at the risk of overfitting to overly structured situations. As face recognition grew beyond the confines of laboratory settings, evaluations such as FRVT [7], FRGC [8], and MBE [9] applied face recognition to real problems like mugshot and passport scanning, high resolution imagery, 3D facial scans, and outdoor scenarios. Lately, face recognition research has shifted towards realistic faces captured in more uncontrolled conditions. In particular, consumer and Internet face recognition tasks have increased in popularity with "in-the-wild" datasets such as LFW [10], PubFig [11], and various private Facebook galleries [12–14]. This has spurred the development of more robust algorithms, although humans still outperform the best approaches [11].

With the increasing pervasiveness of digital cameras, the Internet, and social networking, there is a growing need to catalog and analyze large collections of photos. Because photo interest is largely determined by who appears in the picture, labeling photos with identities is particularly important. In fact, popular social networks such as Facebook allow users to place tags on photos to label people, encouraging collaboratively shared photo albums. Imagine millions of Internet users tagging their photos: such web-scale labeling problems present a real challenge and fascinating opportunity for automation by face recognition.

In such consumer-driven and Internet applications, there are many unique challenges in applying face recognition: the massive-scale nature of dozens or hundreds of faces each for hundreds or thousands of people, the uncontrolled nature of illumination, age, pose, expression, a high variance in image quality, and noisy data due to human mislabeling. Although there are several large-scale evaluations like FRVT [7], FRGC [8], and MBE [9] and verification datasets such as GBU [15] and LFW [10], open-universe face identification remains a little-studied problem in the research community at large, especially with respect to large-scale web and consumer related photo tagging tasks. For instance, in a social network context, only friends should be tagged while ignoring all others (Fig. 1(b)); however, in a local newspaper publication context, a public figure is more noteworthy (Fig. 1(a)). Thus as Fig. 1 depicts, depending on the context, real-world face recognition

☆ This paper has been recommended for acceptance by Martin David Levine.

* Corresponding author.
E-mail addresses: eortiz@cs.ucf.edu (E.G. Ortiz), brianbecker@cmu.edu (B.C. Becker).
URLs: http://enriquegortiz.com (E.G. Ortiz), http://briancbecker.com (B.C. Becker).

1077-3142/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.cviu.2013.09.004

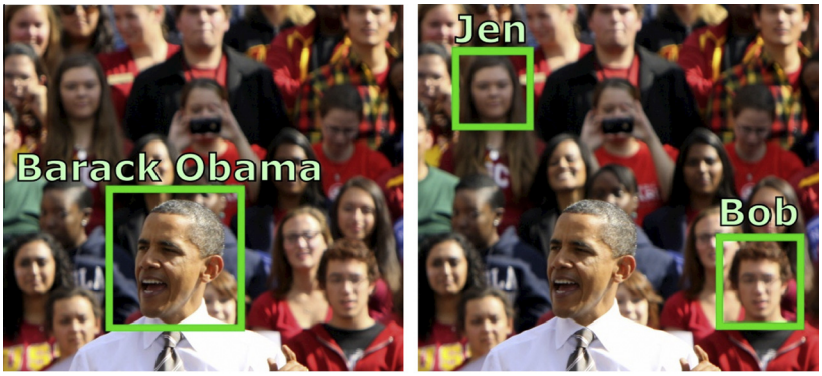


Fig. 1. In open-universe face identification, ignoring distractors is vital. In a news article scenario (a), only public figures are relevant. If the photo is uploaded to Facebook (b), the user only tags friends. All other faces are distractors. Photo credit to Neon Tommy.

must identify specific people reliably while rejecting all others as distractors.

To address these insufficiencies when scaling face identification to web-scale applications in the real-world, we construct a very large dataset from Facebook, propose a novel and efficient algorithm named Linearly Approximated Sparse Representation-based Classification (LASRC), and perform extensive performance evaluations. Inspired by robust sparse methods [16,17] that scale poorly as the number of training images increases (often taking seconds or even minutes using the fastest algorithms on a gallery of 100,000 faces), we investigate how to reduce the high computation times of ℓ^1-minimization techniques used to recover coefficient vectors relating a test face to those in a dictionary. Starting with least-squares solutions, we find the interesting result that imposing brute-force sparsity by thresholding low-magnitude coefficients can markedly improve accuracy in large-scale datasets. We establish the key insight that there exists a correlation between the high-magnitude components of ℓ^2 solutions and coefficients chosen by sparse ℓ^1-minimization. Our method LASRC exploits the speed of ℓ^2 to quickly initialize a sparse solution and serve as an approximation to ℓ^1-minimization, which accurately refines the solution. Furthermore, we show LASRC classifies 100–250 times faster than SRC with similar performance, is comparable to SVMs with almost no training required, and outperforms realtime, state-of-the-art algorithms in web-scale face recognition. We present five contributions:

1. The exploration of large-scale face identification, focusing on realistic open-universe scenarios (Section 2.2).
2. The release of feature descriptors for a new Facebook dataset and a Facebook downloader tool for analysis of large face datasets (Section 3).
3. The development of a novel algorithm, LASRC, for realtime, accurate, and web-scale face identification (Section 4).
4. The evaluation of local features, sparsity, and locality with large-scale datasets in an open-universe scenario (Sections 5 and 6).
5. The comparison of LASRC to many state-of-the-art algorithms with real-world datasets (Sections 7 and 8).

Finally, Section 9 concludes with a discussion and future work.

2. Background

Face recognition is a broad and diverse field [18]. To motivate our paper, we begin by describing a taxonomy of face recognition tasks, emphasizing the importance of open-universe face identification and describing related work. We summarize a relevant subset of face recognition algorithms. Finally, we also review popular controlled and web-gathered datasets with respect to their strengths and weaknesses in the task of facilitating the development of web-scale face recognition.

2.1. Taxonomy of face recognition

As summarized in Fig. 2, face recognition tasks can be categorized as: closed-universe face identification, open-universe face verification, or open-universe face identification.

- **Closed-Universe Face Identification:** Given a set of labeled training faces, what is the identity of a new face? This task is closed-universe because no new faces will be unknown; thus, results are reported as accuracy or error rates. This is the most common form of face recognition with controlled datasets such as Extended Yale B, AR, MultiPIE, or FERET [17,19–25,12–14,26–31].
- **Open-Universe Face Verification:** Given a pair of faces are they the "same" or "not same"? In other words, is an input face's claimed identity correct? Because people can claim any identity, the verification task is open-universe. Just as popular datasets like LFW [10], GBU [15], BANCA [32], XM2VTS [33], and PubFig [11], the task is referred to as pair-matching. Face verification performance is reported with a ROC curve [25,11,26].
- **Open-Universe Face Identification:** Given a labeled training gallery, (1) what is the probability that a new test face is known and (2) what is the most probable identity? Since new face identities are not restricted, the task is referred to as open-universe. Despite being the most realistic face recognition scenario, it is one of the least-studied. Results are reported using ROC or PR curves [16].

2.2. Open-universe face identification

Real-world tasks such as identifying famous people or labeling friends fall under open-universe face identification, the most realistic application domain for face recognition on the web, where the system must determine if the query face exists in the known gallery, and, if so, the most probable identity. As Fig. 1 shows, the ability to reject distractors in an open-universe way is critical to the success of face recognition in realistic scenarios. Thus, it is uncertain how the excellent results reported under closed-universe

Please cite this article in press as: E.G. Ortiz, B.C. Becker, Face recognition for web-scale datasets, Comput. Vis. Image Understand. (2013), http://dx.doi.org/10.1016/j.cviu.2013.09.004

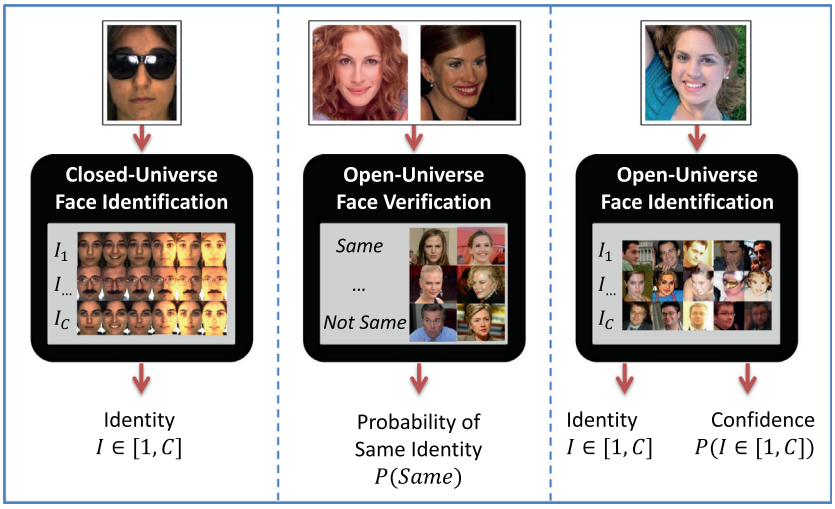


Fig. 2. Three common face recognition tasks.

assumptions [14,17,20,23,25,34] perform in open-universe scenarios. Likewise, verification tasks are popular and have progressed significantly [10,11,35], although verification algorithms have rarely been evaluated in identification tasks. Grother and Phillips [36] provide good insights by exploring the relationship between verification and identification tasks, however they use several simplifying assumptions that may not not be very applicable to web-scale face recognition: identity predictions are independent per individual and the distribution of predictions can be approximated via Monte-Carlo sampling. Thus it is unclear how and to what effectiveness verification algorithms can be efficiently adapted to web-scale face identification; in fact, a recent National Institute of Standards and Technology (NIST) report on face recognition [9] asserts identification-specific algorithms can offer more accurate predictions and better scalability to large populations than performing many verifications.

Historically, NIST has run a series of face recognition evaluations since the 1990s, including explorations of open-universe face identification. Phillips et al. [4] first evaluate the controlled FERET [4] dataset on open-universe identification with a greater than 90% correct identification of known individuals with little variance as the false accept rate of unknown individuals increased. Subsequently, the Face Recognition Vendor Test (FRVT) 2002 [7] evaluated the open-universe, watch-list task on a mixture of visa images and a quasi-controlled collection, where the gallery of known individuals is very small out of a large population of individuals. Finally, the Multi-Biometric Evaluation (MBE) 2010 [9] expands previous evaluations to a much larger scale evaluating both open-universe verification and identification. Although the image data is from mugshots, passports, driver's licenses, a much different image source than most consumer and web faces, the results provide valuable insights, confirming FRVT 2002 results that the identification rate decreases as the population size increases.

Li and Weschler in [37] examine open-set face recognition using Transduction Confidence Machines (TCM) with nearest neighbor on two small datasets (450 and 750 images) with controlled, frontal face images. Both [38,39] use a multi-verification system for open-set identification, where a verifier or 1-vs-all SVM classifier is trained for each identity. Given the responses from each verifier, a test face is labeled unknown if all verifiers give a negative response and the most likely candidate is given a positive response. Our use of SVMs is similar, however we employ a looser rejection criterion where we reject based on a threshold. Most recently, Scheirer et al. [40] explored the open-universe scenario in the object recognition community. They modify SVM margins by introducing two metrics: (1) generalization to separate the planes to handle data beyond the training data and (2) specialization to bring planes closer where an open-set risk measures the tradeoff; however they test on small datasets so scalability to the large scale problems we are addressing is uncertain.

2.3. Algorithmic related work

Since the scope of face recognition research is vast, we cover some recent advances in face identification shown hierarchically in Fig. 3, focusing on least-squares and sparse representations as these methods have demonstrated remarkable success in controlled datasets (other notable methods such as those based on attributes and similes [11] or V1-inspired features [14] do not fit into the subset in Fig. 3 and are not considered).

When considering face identification algorithms suitable for large-scale deployment on a social network or other realtime system with user interaction, several real-world requirements become evident. (1) Algorithms must scale with low training times because any training taking over a few minutes will feel unresponsive to end users, who expect new, added photos and identities to be rapidly processed. (2) Fast classification rates of at least a few Hz are necessary for realtime performance, otherwise users will be able to label faces faster than the system. (3) Identification performance must be high while reliably rejecting unknown identities otherwise users may feel the system is too unreliable. Many existing, popular face recognition, research algorithms suffer in one or more of these areas when applied to web-scale scenarios. We evaluate the subsequent related work with these requirements in mind.

Support Vector Machines: SVMs have fast classification and are very popular in recognition tasks [25,41,42]. Wolf et al. [25] showed good performance on a small subset of LFW with multi-feature SVMs. However, training one-vs-all SVMs with hundreds of classes and tens of thousands of examples takes hours, even with large-scale algorithms such as LIBLINEAR [43] with the dense data patch for speed [41]. Furthermore, limiting the training examples or tuning convergence parameters reduces classification rates too low to be competitive. Lin et al. [42] introduced an Averaged Stochastic Gradient Descent (ASGD) method to train huge SVMs rapidly, but it requires more than 30 min for our large datasets and yields accuracy well below LIBLINEAR. Thus, many current

Please cite this article in press as: E.G. Ortiz, B.C. Becker, Face recognition for web-scale datasets, Comput. Vis. Image Understand. (2013), http://dx.doi.org/10.1016/j.cviu.2013.09.004

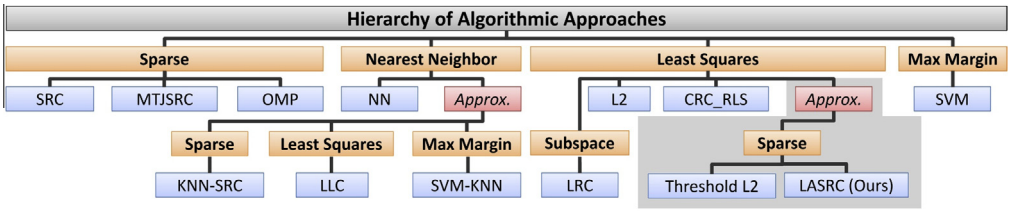


Fig. 3. A hierarchy of face identification algorithms discussed in this paper, grouped by broad categories. Slow performing algorithms such as SRC or SVMs do not scale well, but can employ fast approximations to make an initial guess that can be refined. Highlighted in gray, we propose a novel linear regression approximation for SRC, named LASRC.

SVM approaches train too slowly to be well-suited for dynamic, large-scale face recognition on the Internet where new photos are constantly uploaded and users expect rapid training of new faces and identities for improved recognition.

Sparse Representation Classification (SRC): In the pioneering work, Sparse Representation-based Classification (SRC), Wright et al. [16] presented the principle that a given test image can be represented by a linear combination of images from a large dictionary of faces. The key concept was that the test image can be represented by a small subset of the large dictionary; therefore, the corresponding coefficient vector is sparse, or has only a few non-zero elements obtained with ℓ^1-minimization. Their experiments showed SRC performed well on standard datasets with simple pixel representations and is robust to varying degrees of pixel corruption, block occlusion, and certain disguises. However, SRC required perfectly aligned faces and classification was slow, needing seconds per face.

A large breadth of research in the area of ℓ^1-minimization exists. Early work cast the problem as a linear program [44] and later accounted for small noise with a second-order cone program (SOCP) [45]. Interestingly, both methods are initialized by the ℓ^2 solution. Several faster algorithms have been developed: Gradient Projection for Sparse Representation (GPSR) [46], Homotopy [47], and Augmented Lagrange Multiplier (ALM) [48], amongst others. GPSR finds the solution by following the gradient direction via quadratic programming, Homotopy updates its active set of candidate non-zero coefficients based on a decision criterion from the ℓ^2 solution, and ALM casts the ℓ^1 problem as a Lagrange multiplier method in which infeasible points are given a high cost and thus ignored. Other methods focus on greedy approximations like Orthogonal Matching Pursuit (OMP) [49], which selects one new basis, or coefficient, at each iteration and approximates the sparse solution faster than full ℓ^1-minimization, although the correct solution is not guaranteed.

Improving SRC: Wagner et al. [17] furthered the SRC method by simultaneously aligning and classifying a test image with respect to a pre-aligned training gallery, thus handling pose variations in test images. Unfortunately, it is hard to find a well-aligned training set in real-world scenarios. To rectify this, Peng et al. in [50] combined low-rank and ℓ^1-minimization to perform batch alignment of images. However, this low-rank optimization takes a long time with large datasets even with recent optimizations for video [51]. Patel et al. [52] rectifies lighting and pose via estimation and learns a person specific dictionary via K-SVD an approximation technique used in OMP. They outperform standard SRC under varying illumination, pose, and occlusions. We assume fast funneling [53] or eye-based alignment adequately addresses the variations in pose.

Yang and Zhang [20] found that holistic features like PCA and LDA used in [16] cannot handle variations in illumination, expression, pose, and local deformations. Moreover, the occlusion matrix introduced in [16] makes the ℓ^1-minimization problem computationally prohibitive. They introduced a Gabor wavelet feature as well as a Gabor occlusion dictionary into SRC and showed their method, GSRC, performs better on standard datasets with large degrees of pose and occlusion variations. Also noting the usefulness of features, Chan and Kittler [30] used the Local Binary Pattern (LBP) [54] histogram descriptor, finding local features provided more robustness to misalignments than SRC on raw pixels. Likewise, Yuan and Yan [34] introduced a multi-task joint sparse representation named MTJSRC that fuses multiple local features.

Speeding up SRC: While the convex, ℓ^1-minimization problem can be easily solved by linear programming and other classical methods, the complexity remains too high for large, high-dimensional dictionaries [20]. Observing that the ℓ^1-optimization procedure of SRC is very slow, researchers have focused on speeding-up the process while maintaining robustness. Shi et al. [22] combined an explicit hashing function to reduce data dimensionality while preserving important structure information for ℓ^1-minimization via OMP. Differently, Nan and Jian [29] and Li et al. [28] used a fast K nearest neighbor method (KNN) to select training samples local to the test image for input to the ℓ^1-solver. They showed this KNN-SRC method performs well with a considerable speedup. Likewise, new correlation-based screening pre-processing rules such as the SAFE rule [55] or the Sphere Test 3 [56] have been proposed to safely and rapidly eliminate training samples before ℓ^1-minimization for increased speed.

Least-Squares Solutions: Instead of optimizing or approximating ℓ^1-minimization, other researchers loosened sparsity constraints by imposing an ℓ^2-norm rather than an ℓ^1-norm. Bypassing ℓ^1-optimization completely, very fast least-squares approaches can be used in coefficient vector recovery. In [27], Naseem et al. proposed a nearest-subspace least-squares method named LRC that can be extended with block-based recognition to handle occlusion. Similarly, Shi et al. [23] questioned whether face recognition is really a compressive sensing problem and demonstrated least-squares is comparable to SRC on controlled datasets. Zhang et al. [24] presented a regularized ℓ^2-minimization (CRC_RLS) that placed an additional constraint on the coefficient vector, adding robustness to occlusion. Furthermore, Wang et al. [19] asserted that locality is more important than sparsity and discovers a coefficient vector from a weighted least-squares solution, or Locally-constrained Linear Coding (LLC), performed on an image's K nearest neighbors. Moreover, Xu et al. [57] propounded that there is a tradeoff between sparsity and stability in linear solutions. Although studies have cast doubt on the advantages of sparsity for recognition, we show that pure ℓ^2-based methods struggle when presented with open-universe, real-world data from Labeled Faces in the Wild (LFW) [10], PubFig [11], and Facebook [12–14].

In summary, we have presented that SRC methods for face recognition perform well with high robustness with the drawbacks that they are (1) sensitive to pose variations and (2) slow to recover coefficient vectors. Least-squares methods address the speed issue by removing the ℓ^1 constraint on the coefficient vector, however exhibit increased sensitivity to variations in the data as we show later in Section 8.3. Although ℓ^1 methods are slow, they exhibit robustness in discovering the correct identity of test faces. Our

Please cite this article in press as: E.G. Ortiz, B.C. Becker, Face recognition for web-scale datasets, Comput. Vis. Image Understand. (2013), http://dx.doi.org/10.1016/j.cviu.2013.09.004

method combines the speed of least-squares to discover a subset of the initial dictionary to feed into ℓ^1-minimization to discover the final identity of a given test face. In our experimentation, we address pose sensitivity through the use of three popular features (LBP, HOG, and Gabor). Furthermore, we demonstrate least-squares works well for ℓ^1-approximation. Our combination of local features with ℓ^2 and subsequent ℓ^1-minimization provides the speed and robustness necessary to deal with real-world data.

2.4. Existing face datasets

Traditionally, face recognition operates on faces captured in artificial environments where conditions are carefully controlled or labeled (AR [2], Yale [3], and FERET [4]). More recently, web-gathered LFW [10] and PubFig [11] datasets have gained popularity with face verification tasks with an increased focus on large-scale evaluations such as GBU [15] and MBE [9]. We summarize existing datasets in Table 1.

2.4.1. Controlled datasets

Faces in highly controlled datasets such as Ext. Yale B [3] and the AR Face Database [2] are very popular choices for face recognition evaluation. The Ext. Yale B [3] dataset contains 38 subjects under 64 lighting conditions (Fig. 4(a)). The AR Face Database [2] contain 50 male and 50 female subjects with images taken two weeks apart for each (Fig. 4(b)). The FERET dataset [4] (Fig. 4(c)) explores variations in pose, expression, and even time. Although testing on such datasets provides a good baseline for proof-of-concept, excellent results do not necessarily ensure success on uncontrolled, real-world scenarios. Private datasets such as those used in FRVT [7], FRGC [8], and MBE [9] are less controlled and much larger and realistic, being pulled from law enforcement and visa sources.

2.4.2. Verification datasets

Two datasets designed for face verification have become popular: the Good, the Bad, and the Ugly (GBU) [15] and Labeled Faces in the Wild (LFW) [10]. Unlike identification tasks that explicitly determine the identity of a face, in verification tasks, pairs of images are compared for similarity to determine if the identity of the two people are the same or not. GBU has 6.5k photos of 437 identities divided into three partitions: easy (good), hard (bad), and very difficult (ugly) faces to match. The division of faces into three partitions is particularly useful to evaluate algorithmic performance at different difficulty levels. The LFW dataset has 13.2k faces of over five thousand celebrities and public figures, and has inspired an interest in face recognition applied to real-world, "in-the-wild" photos.

2.4.3. Web-gathered datasets

Seeking more realistic faces, two new datasets gathered from Internet images using keyword searches of famous people have been introduced: the 13.2k image Labeled Faces in the Wild (LFW) [10] dataset (Fig. 4)) and the 58.8k image Public Figures (PubFig) [11] dataset (Fig. 4(e)). Researchers have also used social network faces [12–14], but these datasets have not been released. The predominant use of LFW and PubFig is face verification [10,11,35], although small subsets have been used for closed-universe face identification [25,14]. To adapt these datasets for testing open-universe face identification tasks, we first aligned all faces with the LFW standard, funneling method of Huang et al. [53]. We created five datasets from the 200 identities of PubFig with a random 75%/25% train/test split. To incorporate the open-universe aspect, all aligned LFW faces were added as distractors (except 138 overlapping identities). This mimics a web-scale face recognition scenario of finding specific celebrities while ignoring all other faces.

3. Facebook dataset

Our interest is in large-scale, realistic face identification scenarios for personal photo collections where diversity is naturally-captured. Several works have explored face identification with photos from Facebook [12–14], but only in the closed-universe scenario. None have addressed the more important open-universe scenario where the algorithm will encounter many background faces that should be rejected as non-friends (Fig. 1). Focusing on the scenario of automatically tagging friends in open-universe social networks, we created a new 800,000 face dataset (Fig. 4(f)) collected from tagged Facebook photos. Feature descriptors for this new dataset and our downloader tool for Facebook photos, tags, face detection, matching, and alignment are available at http://www.enriquegortiz.com/fbfaces.

Table 1
A brief summary of a subset of popular and Internet-based face recognition datasets, listing whether or not they are publicly available for download, the photographic source of the images (captured in a lab, taken from law enforcement visas/mugshots, or the Internet), whether or not the images were controlled (i.e. if the subjects were captured in a specific setting or in the wild), for what task most papers use the dataset (closed universe identification, face verification, or open universe identification), approximately how many faces per known identity there are, the number of known identities in the dataset, the number of total faces, and the number of unknown identities.

Dataset name	Public	Source	Controlled	Main task	Faces/ID	Known IDs	# Faces	Unknown IDs
DOS/Natural [9]	No	Visas	Yes	Open ID	1	520k	625k	50k
DOS/HCINT [9]	No	Visas	Yes	Verification	3	37.4k	121k	30k
LEO [9]	No	Mugshots	Yes	Open ID	1	1.6M	2.4M	200k
SANDIA [9]	No	Lab	Yes	Verification	50	263	13.9k	–
FERET [4]	Yes	Lab	Yes	Closed ID	12	1.2k	14k	–
ATT (ORL) [1]	Yes	Lab	Yes	Closed ID	10	40	400	–
Ext. Yale B [3]	Yes	Lab	Yes	Closed ID	576	28	16.1k	–
AR [2]	Yes	Lab	Yes	Closed ID	30	126	4k	–
GBU [15]	Yes	Lab	Semi†	Verification	15	437	6.5k	–
LFW [10]	Yes	Web	No	Verification	3	5.7k	13.2k	–
MultiPIE [31]	Yes	Lab	Yes	Closed ID	2k	337	750k	–
PubFig [11]	Yes	Web	No	Verification	300	200	58.8k	–
Facebook [12]	No	Web	No	Closed ID	25	15.8k	439k	–
Facebook [13]	No	Web	No	Closed ID	65	946	61.7k	–
Facebook [14]	No	Web	No	Closed ID	100	100	10k	–
PubFig + LFW (Ours)	Yes	Web	No	Open ID	175	200	58k	11k
Facebook (Ours)	Semi*	Web	No	Open ID	112	6.1k	803k	110k

* Raw images not available for privacy reasons, but feature descriptors are available.
† Some photos are taken outdoors in natural lighting.

Please cite this article in press as: E.G. Ortiz, B.C. Becker, Face recognition for web-scale datasets, Comput. Vis. Image Understand. (2013), http://dx.doi.org/10.1016/j.cviu.2013.09.004

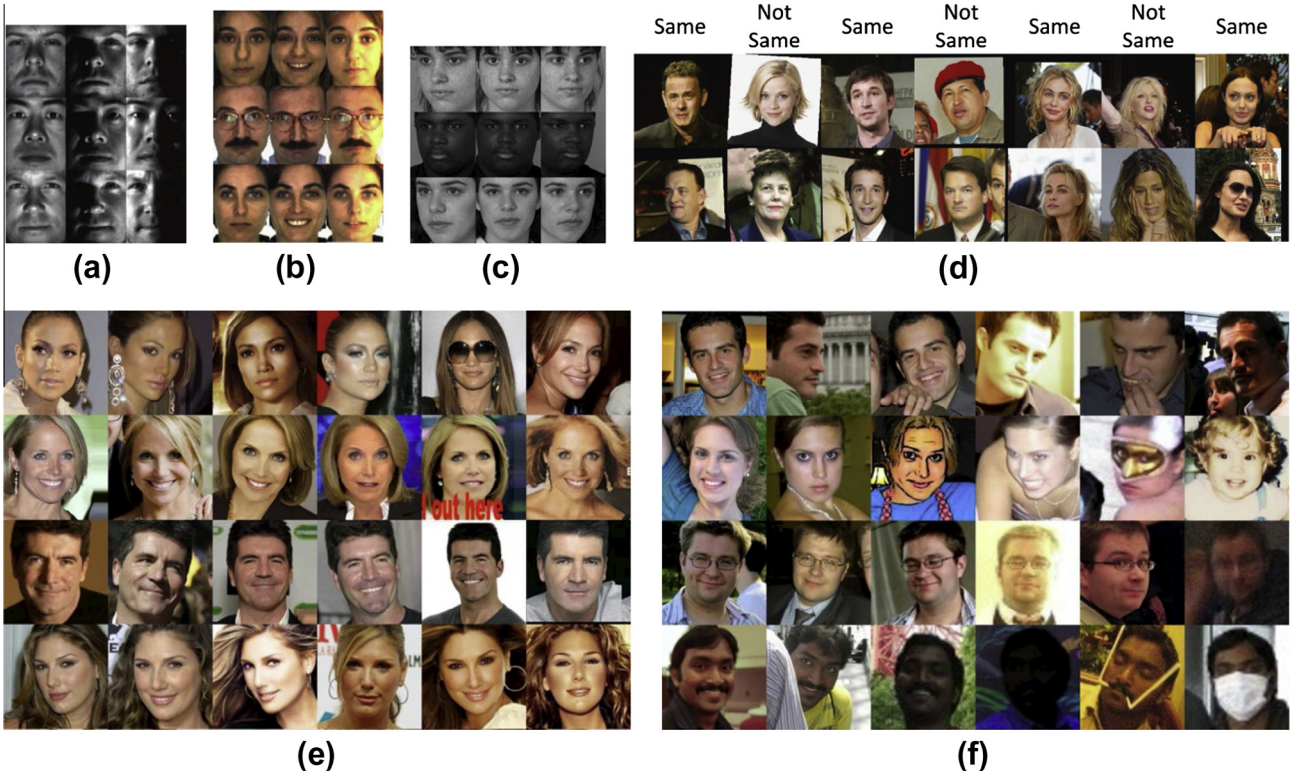


Fig. 4. Example from controlled datasets (a–c) and web-gathered datasets (d–f). (a) Ext. Yale B [3] concentrates on illumination, (b) AR [2] on disguises, and (c) FERET [4] on pose. (d) LFW [10] focuses on pair matching between famous faces and (f) PubFig [11] between celebrity photos. (f) Our challenging, realistic Facebook dataset is naturally diverse in pose, illumination, occlusion, and age. Publishing consent was obtained.

3.1. Dataset construction

Using our provided tools, researchers can build very similar, yet customizable datasets from Facebook.

Face Collection: We collected 24.6 million photos with a total 29.2 million tags, representing 2.9 million unique people from a total of 83,000 Facebook users. The high-performance SHORE face detection system [58,59] was used to detect 48.3 million frontal faces with a rotation range of approximately ±35° at a rate of 20 Hz. From 3000 ground-truth face and tag matches, we modeled the probability that a tag represents a nearby face based on distance and orientation. Using a false alarm rate (FAR) of 1%, 17.4 million face matches were extracted and aligned by a similarity transform based on SHORE-reported eye positions.

Including Distractors: For many photos, distractor (unknown) faces exist in the background. For each test face, we collected tagged, non-friend faces also in the photo and labeled them as distractors. As listed in Table 2, there are similar numbers of test and distractor faces. Thus, our dataset exactly models the real-life scenario and allows evaluation of the face identification algorithms' ability to reject unknown faces under the open-universe scenario.

Table 2
Facebook (FB) and PubFig + LFW (PF) datasets detailing the training identities per dataset and the number of dataset repetitions. Reported training, test, and distractor faces per dataset are averaged.

Name	Ids	Reps	Train	Test	Distractor
FB256	256	8	22.0k	7.2k	4.5k
FB512	512	4	42.4k	13.9k	9.0k
FB1024	1024	2	88.6k	29.0k	18.8k
PF + LFW	200	5	35.5k	11.6k	11.7k

Dataset Statistics: To best mimic real-world usage, we randomly placed Facebook users into groups of 256, 512, and 1024 identities to simulate users with varying numbers of friends. For thorough evaluation, we sample multiple repetitions of each group with no overlap amongst any identities or photos. Only users with at least 20 photos were kept as they are more likely to be tagged and represent more than 75% of the collected faces. We collected all the photos a user had been tagged in and used the oldest 75% faces as training and the remaining most recent 25% photos as testing, which most closely models the real-world.

3.2. Evaluation criterion

The standard metrics for open-universe face identification are ROC curves based on the detect and identify rate, which reports the number of knowns correctly classified at a given threshold, and the false accept rate, which shows the number of unknowns falsely labeled at a given threshold [4]. In addition, we propose using precision, which encodes the ratio of correct identifications to the number of returned identifications, and recall, which is a ratio of coverage over the known test data [60]. Intuitively, where the ROC curves tell us the tradeoff between correctly labeling data of interest vs. labeling the data of disinterest, the PR curves, as defined, tell us at a given threshold how much data of interest do we label and how well we do on that data. From a social-networking standpoint, it is advantageous to provide the user fewer labels with high precision; therefore, we feel that recall at 95% precision better reflects real-world performance as this corresponds to the percentage of detected faces of interest that can be labeled with only one mistake in 20 predictions. Finally, since fast classification and training are necessary in such dynamic, real-world situations, we report train and test times.

Please cite this article in press as: E.G. Ortiz, B.C. Becker, Face recognition for web-scale datasets, Comput. Vis. Image Understand. (2013), http://dx.doi.org/10.1016/j.cviu.2013.09.004

3.3. Dataset bias

Torralba and Efros [61] emphasized the importance of minimizing the selection, capture, and negative set biases of new datasets. Unlike LFW and PubFig images, our Facebook dataset does not suffer from a keyword-based selection bias as we automatically extracted faces from crowd-annotated personal photos. However, selection is biased towards younger people given social network demographics. In contrast to the professional photographer bias of LFW and PubFig, Facebook's capture bias is predominantly skewed towards everyday, consumer quality photos. Traditionally, classification is handled as a binary problem where you must label a positive class of interest amidst a negative class consisting of a very large range of classes it is not, where coverage of all classes is very difficult. The negative set bias in our scenario is minimized due to the large sampling range offered by data collection via Facebook. More importantly, the Facebook dataset has a large negative set in the form of a realistic set of distractors from non-friend background faces.

4. Linearly approximated SRC

Our problem is the classic face recognition scenario where we want to classify a test image $\boldsymbol{y} \in \mathbb{R}^m$ given a database of C known subjects (classes). Assume the n_j faces of subject $j \in [1, \ldots, C]$ are stacked into a matrix $A_j = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{n_j}]$ as column vectors, therefore matrix A is composed of all of the faces for all subjects $A = [A_1, \ldots, A_C] \in \mathbb{R}^{m \times n}$, where m is the length of the feature vector and $n = n_1 + \cdots + n_j$ is the total number of images. Assuming that test image $\boldsymbol{y}$ can be represented as a linear combination of images of itself within the training set, we can represent the problem as $\boldsymbol{y} = A\boldsymbol{x}$, where $\boldsymbol{x}$ is a coefficient vector encoding the relationship of $\boldsymbol{y}$ to the columns of A.

4.1. Least-squares solution

A typical solution is to use the traditional method for error minimization, least-squares, to find an estimate of $\boldsymbol{x}$, which casts the minimization as:

$$\hat{\boldsymbol{x}}_{\ell_2} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - A\boldsymbol{x}\|_2^2, \tag{1}$$

and is computed by the psuedoinverse as follows:

$$\hat{\boldsymbol{x}}_{\ell_2} = (A^T A)^{-1} A^T \boldsymbol{y}. \tag{2}$$

The ℓ^2 solution is convenient as it is very fast to evaluate and the pseudoinverse can be precomputed with Singular Value Decomposition (SVD) and cached. In the case of an underdetermined system, we can use the least-norm solution, which is also calculated with SVD. Wright et al. [16] stated that $\hat{\boldsymbol{x}}_2$ is dense, as seen in the ℓ^2 coefficients in Fig. 9(a), and therefore is not very informative. However, recent studies [23,24] show that ℓ^2 works well for common datasets even though the measurements are noisy.

4.2. Sparse representation-based classification

Compressive sensing has been shown to outperform least-squares using only a subset of available data [16]. Given test image $\boldsymbol{y}$ and training set A, we know that the images of the same class to which $\boldsymbol{y}$ should match is a small subset of A. Therefore, the coefficient vector $\boldsymbol{x}$ should only have non-zero entries for those few images from the same class and zeros for the rest. Imposing this sparsity constraint upon the coefficient vector $\boldsymbol{x}$ with small dense error ϵ to handle noise/occlusion results in the following formulation:

$$\hat{\boldsymbol{x}}_{\ell_1} = \min_{x,\epsilon} \|\boldsymbol{x}\|_1 + \|\epsilon\|_2 \text{ s.t. } \boldsymbol{y} = A\boldsymbol{x} + \epsilon, \tag{3}$$

where the ℓ^1-norm enforces a sparse solution by minimizing the absolute sum of the coefficients. The result of the sparsity constraint is seen in the ℓ^1 coefficients in Fig. 9(a), where the largest non-zero values are concentrated on the matching training images corresponding to the correct class.

Wright et al. [16] identifies the test image $\boldsymbol{y}$ by determining the class of training samples that best reconstructs the face from the recovered coefficients:

$$I(\boldsymbol{y}) = \min_j r_j(\boldsymbol{y}) = \min \|\boldsymbol{y} - A_j \boldsymbol{x}_j\|_2, \tag{4}$$

where the label $I(\boldsymbol{y})$ of the test image $\boldsymbol{y}$ is the minimal residual or reconstruction error $r_j(\boldsymbol{y})$ and $\boldsymbol{x}_j$ is the recovered coefficients from the global solution $\hat{\boldsymbol{x}}_{\ell_1}$ that belong to class j. Confidence in the determined identity is obtained using the Sparsity Concentration Index (SCI) proposed by [16]. SCI is a measure of how distributed the residuals are across classes:

$$\text{SCI} = \frac{C \cdot \max_j \|\boldsymbol{x}_j\|_1 / \|\hat{\boldsymbol{x}}_{\ell_1}\|_1 - 1}{C - 1} \in [0, 1]. \tag{5}$$

SCI ranges from zero (the test face is represented equally by all classes) to one (the test face is fully represented by one class). Wright et al. [16] show that SCI is a better metric than the minimum residual for rejecting distractor faces, which is particularly important in open-universe, real-world environments.

4.3. Approximating SRC

A large drawback to SRC is the computational complexity required by ℓ^1-minimization, which requires several seconds per image [16,17] even on datasets with only a few hundred or thousand training samples. Compared to least-squares which takes less than 100 ms for the largest Facebook datasets, the fastest ℓ^1-solver, Homotopy [47], takes at least 5 s while more accurate solvers take over a minute. Therefore, we developed a way to approximate ℓ^1-minimization.

The objective function $v(\boldsymbol{x})$ of the Lagrangian formulation of the ℓ^1-minimization (3) specified as a sequence of vector operations is as follows:

$$v(\boldsymbol{x}) = \|\boldsymbol{y} - \sum_{i=1}^{n} \boldsymbol{a}_i x_i\|_2 + \lambda \sum_{i=1}^{n} |x_i|, \tag{6}$$

in which we denote $\boldsymbol{a}_i \in \mathbb{R}^m$ as the ith column of A, x_i as the ith element of coefficient vector $\boldsymbol{x}$, and λ as the sparsity controlling parameter. Assuming K sparsity where at most K values are non-zero, for any i for which $x_i = 0$ in (6), then $\|\boldsymbol{a}_i x_i\|_2 = 0$, $|x_i| = 0$, and $\boldsymbol{a}_i$ do not contribute to $v(\boldsymbol{x})$. Based on this observation, we rewrite the objective function as:

$$v(\boldsymbol{\alpha}) = \|\boldsymbol{y} - \sum_{i=1}^{K} \boldsymbol{\omega}_i \alpha_i\|_2 + \lambda \sum_{i=1}^{K} |\, \alpha_i \,|, \tag{7}$$

where $\boldsymbol{\omega}_i$ represents a column from a matrix Ω containing only columns contributing to the error and α its corresponding coefficient values. Since the error estimation is not dependent on the zero entries of $\boldsymbol{x}$, $v(\boldsymbol{x}) = v(\boldsymbol{\alpha})$. With the new dictionary Ω and coefficient vector $\boldsymbol{\alpha}$, we can reformulate the ℓ^1-minimization as:

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{y} - \Omega\boldsymbol{\alpha}\|_2 + \lambda \|\boldsymbol{\alpha}\|_1. \tag{8}$$

The new objective function $v(\boldsymbol{\alpha})$ is analytically identical to $v(\boldsymbol{x})$, yet much faster to evaluate for $K \ll n$. Since the ℓ^1 solution produced by the GPSR ℓ^1-solver [46] with $\tau = 0.01$ is 97.6% sparse, significant speed-ups are possible. However, ℓ^1-minimization is an iterative optimization with a finite step-size so some difference in solution is expected. We measure the difference to be 4% on randomly generated data, but only 1.6% using 10,000 images from Facebook.

Please cite this article in press as: E.G. Ortiz, B.C. Becker, Face recognition for web-scale datasets, Comput. Vis. Image Understand. (2013), http://dx.doi.org/10.1016/j.cviu.2013.09.004

This formulation depends on knowing which coefficients of $\boldsymbol{x}$ will be non-zero in order to form Ω, or equivalently, which training samples will be included in the sparse minimization. Finding the exact contributing samples is no easier than ℓ^1-minimization, but we claim it is easier to approximate. As discussed in Section 4.1, ℓ^2-minimization is very fast, convenient, and has proven to be adequate for standard face recognition datasets. Furthermore, it is evident in Fig. 9(a) that although the ℓ^2 solution is dense, the highest peaks are similar to the ℓ^1 solution and correspond to the training images that match the identity of the test image. Moreover, as previously noted the ℓ^2 solution is used to initialize several ℓ^1 solvers. We conclude that despite ℓ^2 being noisier, it has a similar shape to ℓ^1 and is likely to serve as a good approximation. In Section 6.2.1, we show that high-magnitude coefficients of least-squares have a high probability of corresponding to non-zero coefficients in ℓ^1 solutions. This correlation is largely related to the fact that both obtain global solutions on similar error functions with different norm constraints.

Algorithm 1. Linearly Approximated SRC (LASRC)

1. **Input:** Training gallery $A \in \mathbb{R}^{m\times n}$, test face $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$, and sparsity controlling parameter λ.
2. Normalize the columns of A to have unit ℓ^2-norm
3. Compute linear regression using the pre-calculated pseudoinverse $\hat{\boldsymbol{x}}_{\ell_2} = (A^T A)^{-1} A^T \boldsymbol{y}$
4. Select K samples from A corresponding to the largest coefficients in $|\hat{\boldsymbol{x}}_{\ell_2}|$, yielding subset Ω
5. Solve the ℓ^1-minimation problem with approximated subset dictionary $\Omega \in \mathbb{R}^{m\times K}$

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{y} - \Omega\boldsymbol{\alpha}\|_2 + \lambda\|\boldsymbol{\alpha}\|_1$$

6. Compute residual errors for each class $j \in [1, C]$

$$r_j(\boldsymbol{y}) = \|\boldsymbol{y} - \Omega_j\boldsymbol{\alpha}_j\|_2$$

7. Compute SCI

$$\mathrm{SCI} = \frac{C \cdot \max_j \|\boldsymbol{\alpha}_j\|_1 / \|\hat{\boldsymbol{\alpha}}\|_1 - 1}{C - 1}$$

7. **Output**: identity $I(\boldsymbol{y}) = \arg\min_j r_j(\boldsymbol{y})$, confidence $P(I \in [1, C]|\boldsymbol{y}) = \mathrm{SCI}$

4.4. Linearly approximated SRC

Our proposed algorithm, Linearly Approximated SRC (LASRC), uses ℓ^2 solutions to approximate ℓ^1-minimization to gain the speed of least-squares and the robustness of SRC. In Fig. 5, we show our complete system for face recognition. We focus on the classification stage, where we perform linear regression approximation and SRC. We first rapidly compute the coefficient vector $\hat{\boldsymbol{x}}_{\ell_2}$ with linear regression (2) using the pre-calculated pseudo-inverse $(A^T A)^{-1}A^T$. Next, we select the top K training samples from A corresponding to the largest magnitude coefficients $|\hat{\boldsymbol{x}}_{\ell_2}|$ and create the approximated matrix $\Omega = \boldsymbol{a}_s$. We then use the smaller dictionary Ω as input to the ℓ^1-solver to compute a new sparse vector $\boldsymbol{\alpha}$ shown in (8). The most probable identity is found using the minimal residual error $r_j(\boldsymbol{y}) = \|\boldsymbol{y} - \Omega_j\boldsymbol{\alpha}_j\|_2$. Finally, we compute SCI as in (5) for the probability that the given test image identity exists in the training database. In the hierarchy shown in Fig. 3, our method is sparse using a least-squares approximation.

5. Feature representations

Using local features to augment classification is a widely used technique [25,54,62]. However, due to underlying assumptions of pixel-wise linearity, least-squares and sparse methods have primarily focused on raw pixels [16,17,23,24]. On the other hand, Chan and Kittler [30] and Yang and Zhang [20] reported that using features increased accuracy by 20–40% when misalignments or pose variations were present. Furthermore, there is evidence that multi-feature sparse methods can be successful with object recognition [34].

5.1. Feature selection and extraction

Because real-world datasets contain pose variations even after alignment, we use three fast and popular local features: Gabor wavelets [62], Local Binary Patterns (LBP) [54], and Histogram of Oriented Gradients (HOG) [63]. Inclusion of more features aids recognition slightly, but at much higher computational costs.

Before feature extraction, all images are first normalized by subtracting the mean, removing the first order brightness gradient, and performing histogram equalization. Gabor wavelets were extracted with one scale $\lambda = 4$ at four orientations $\theta = \{0°, 45°, 90°, 135°\}$ with a tight face crop at a resolution of 25×30 pixels. A null Gabor filter includes the raw pixel image (also 25×30) in the descriptor. In agreement with [26], we found looser crops work better for histogram-based features. The standard $LBP^{U2}_{8,2}$ and HOG descriptors are extracted from 72×80 resolution loosely cropped images with a histogram size of 59 and 32 over 9×10 and 8×8 pixel patches, respectively. All descriptors were scaled to unit norm, dimensionality reduced with PCA, and zero-meaned.

5.2. Performance

For reporting results, we use both controlled datasets (Section 2.4.1) and the Facebook datasets (Section 3). Times are from a 2.3 GHz machine (single-threaded).

5.2.1. Controlled datasets

To better understand feature performance, we present results on controlled datasets (Section 2.4.1), including both the originally reported accuracies and our results when running the same algorithms on a 1995 length vector concatenated from Gabor, LBP, and HOG. For Ext. Yale B, we randomly selected 32 images per subject for training, leaving 32 for testing. This random selection is repeated 10 times. For the AR Face Database, we selected seven images from Session 1 for training and seven images from Session 2 two weeks later for testing. Using standard experimental protocols and the same database setups as [16,20–22,28,34], our results are directly comparable to previously reported accuracies. Table 3 clearly illustrates two important conclusions. First, higher-dimensional local features powerfully aid all algorithms. Secondly, since most algorithms achieve a 99.5% or higher accuracy with features, we conclude face recognition on small, same day, and moderately controlled illumination datasets is largely a solved problem. Finally, to explore robustness against pose, 1400 faces from 198 identities from the FERET dataset [4] with pose variations of $\theta = \{-25°, -15°, 0°, 15°, 25°\}$ were used in the same manner as [20]. Fig. 6(a) uses the FERET pose dataset (Section 2.4.1) to compare SRC [16] with raw pixels, GSRC [20] with Gabor features, and LASRC with local features. A single feature aids recognition

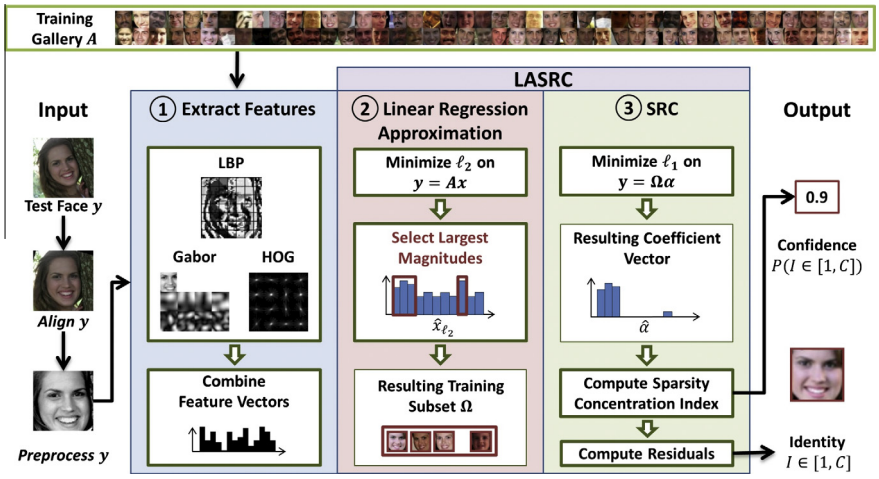
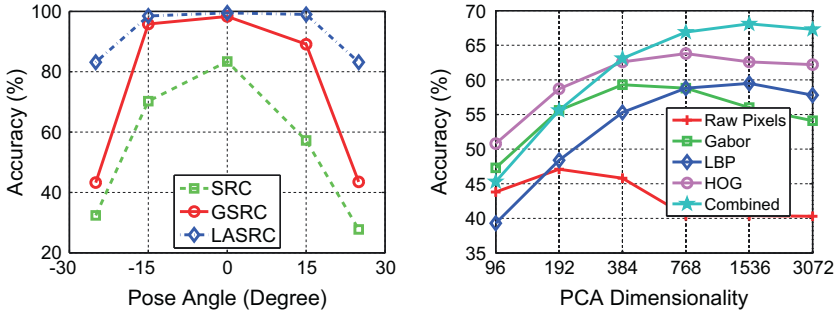


Fig. 5. System flowchart depicting how LASRC classifies a new test face y given a set of training faces A. After alignment and preprocessing, local features are extracted and concatenated, linear regression is performed to select representative training samples Ω, and ℓ^1-minimization is performed to calculate the most probable identity and confidence.

Table 3
Accuracy on controlled datasets as reported originally vs. a three feature representation (Gabor, HOG, LBP). Most algorithms achieve >99.5% with features.

Algorithm	Extended yale B		AR face dataset	
	Reported acc (%)	Feature acc (%)	Reported acc (%)	Feature acc (%)
NN[a]	90.7	92.1	89.7	98.7
SVM[a] [25]	97.7	**99.8**	95.7	99.6
SVM-KNN [64]	–	99.7	–	98.1
SRC [16]	98.1	99.7	94.7	99.9
MTJSRC[b,c] [34]	**99.5**	99.7	–	99.7
LLC [19]	–	99.7	–	99.9
OMP [22]	96.4	99.6	**96.9**	**100.0**
KNN-SRC [29]	88.0	99.7	–	99.9
LRC [27]	–	98.7	–	98.9
L2 [23]	98.9	**99.8**	95.9	99.9
CRC_RLS [24]	97.9	**99.8**	93.7	**100.0**
LASRC (Ours)	–	99.7	–	99.9

[a] Reported from [16].
[b] Accuracy interpolated from graph.
[c] Not using a raw pixel representation.



(a) FERET Pose Accuracy (b) Facebook Accuracy

Fig. 6. Performance of LASRC with features. (a) Three features improve accuracy on the FERET pose dataset (Section 2.4.1) by as much as 55%. (b) Accuracy on Facebook dataset with various features and varying dimensionality.

by 20%, but multiple features with LASRC boosts accuracy up to 50% compared to raw pixels.

5.2.2. Facebook dataset

Repeating similar experiments with Gabor, LBP, and HOG local features on our large-scale, real-world Facebook datasets, we investigate in Fig. 6(b) the individual contributions of each feature to LASRC as dimensionality is varied from 96 to 3072. Because linear approximation is so efficient and a small sample selection K greatly speeds ℓ^1-minimization, LASRC classifies in under 150 ms even on the largest Facebook dataset with 3072 dimensions. Raw pixels plateau first at 47% with 200 dimensions while features such

as LBP, Gabor, and HOG peak at 59% between 400 and 800 dimensions. Finally, a representation of multiple features combined achieves peak accuracy of 67% at 1536 dimensions (512 from Gabor, HOG, and LBP each), which is 20% over raw pixels. We see a significant increase in open-universe performance with more features, similar to the closed-universe accuracy in Fig. 6.

5.3. Effect of occlusion in real-life

One of the well known advantages of linear representations such as SRC is their ability to robustly handle occlusions, noise, and disguise via the creation of an occlusion dictionary [16,23]. Since occlusions are clearly evident in real-world faces, we resized Facebook images to 15×13 and used a 195×195 identity matrix as an occlusion dictionary. Compared to SRC on raw pixels, SRC with an occlusion dictionary yields an improvement of 0.5% in accuracy and 1.1% increase in recall at 95% precision. We conclude that an occlusion dictionary helps performance, but much less than features. This is unsurprising as [16,23] used all unoccluded faces for training and all occluded faces for testing, which is rarely the case in real-world scenarios. Furthermore, occlusion dictionaries assume raw pixel representations or linear Gabor filters [20], so a general solution for histogram features such as LBP and HOG is still an open research problem. Because features increase accuracy by 15–25% (Fig. 6(b)) while occlusion dictionaries only help by 0.5%, we choose to focus on multi-feature representations.

5.4. Effect of dataset size in real-life

Although our proposed approach targets very large, web-scale datasets in environments where users of social media upload and share many photos, it is worthwhile to investigate performance on casual users who only infrequently upload photos. To simulate scenarios where individuals may have only a few photos for training, we randomly subsampled each user's photo collection in the Facebook dataset by 50%, 25%, and 10%. Fig. 7. shows the performance in terms of recall at high precision as the dataset size is varied across a selection of algorithms; notice LASRC remains competitive to existing methods, even in scenarios where some users have only 3 training faces available.

6. Sparsity and locality analysis

Lately there has been controversy between the relative effectiveness of least-squares [23,24,27,57] vs. sparse [16,17,34] solutions. Furthermore, some works advocate the use of locality [19,29] for approximation. Since LASRC uses ℓ^2 solutions to approximate ℓ^1 sparse solutions, we explore how these algorithms perform in large-scale, open-universe scenarios with respect to sparsity and locality.

6.1. Sparsity

By selecting only a small pool of K training samples for ℓ^1-minimization, LASRC yields an extremely sparse solution. Typical sparsity for GPSR ℓ^1-minimization with $\lambda = 0.01$ is about 97%; whereas LASRC is 99.7–99.9% sparse with $K = 64$. However, [23,24] claim that sparsity is not needed in face recognition, prompting us to ask important questions:

- What ℓ^1-solver should LASRC use?
- How do non-sparse, least-squares solutions perform in realistic, open-universe scenarios?
- Is ℓ^1-minimization necessary for LASRC?
- How fast are ℓ^1, ℓ^2, and LASRC algorithms?

6.1.1. Algorithms for ℓ^1-minimization

To answer the first question, a variety of ℓ^1-minimization techniques could be used [65]. Table 4 evaluates popular approaches to ℓ^1-minimization within LASRC, which seeks a sparse representation between relatively few samples in a high dimensional space. All algorithms were run with $\lambda = 0.01$, $tol = 10^{-6}$, and all other parameters set to their defaults. While several algorithms perform similarly, we selected GPSR [46] as a good compromise.

6.1.2. Least-squares performance

On controlled datasets, [23,24,27] used least-squares to achieve results comparable to SRC with orders of magnitude speed benefits. However, they operate with completely balanced datasets with an equal number of training samples per class. Since ℓ^2 solutions are dense with all training images contributing to the residual error computation, least-squares methods are more sensitive to imbalances in image distribution. Realistic datasets such as LFW, PubFig, and Facebook are naturally unbalanced, so least-squares approaches yield poor accuracy and even poorer precision and recall performance (Table 4). Existing works [23,24,27] fail to address this issue, so we attempted to give least-squares algorithms a competitive edge by balancing the datasets. As shown in Table 4, least-squares balanced to a max of 100 randomly-selected training images per identity increases accuracy by 10% and recall at 95% precision by 12%. However, it still underperforms LASRC.

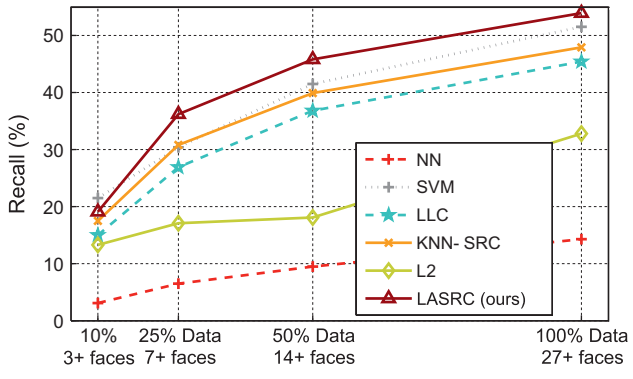


Fig. 7. Effect of recall at 95% precision by varying the size of the dataset (mean number of minimum training faces for all Facebook datasets) across multiple algorithms.

Table 4
Evaluation of least-squares and ℓ^1-solvers with LASRC ($K = 64$). Results reported on Facebook datasets with mean accuracy, mean recall at 95% precision, and mean classification time per test face.

Algorithm	Recall (%)	Accuracy (%)	Time (ms/face)
L2[a] [23]	22.4	49.3	55.3
L2 (balanced, max 100)[a] [23]	34.5	59.2	52.7
Thresholded L2	41.9	63.3	21.2
LLC[a] [19]	46.1	61.5	38.1
KNN-SRC[a] [29]	48.5	63.3	31.6
LRC[a] [27]	28.4	57.2	43.4
LASRC (Homotopy[a] [47])	50.5	65.1	61.1
LASRC (l1magic [66])	44.6	63.3	**29.3**
LASRC (L1_LS [67])	53.4	**66.6**	79.1
LASRC (GPSR [46])	**54.5**	66.5	31.7
LASRC (ALM [48])	54.4	66.5	35.2

[a] Confidence calculated from residuals instead of SCI.

6.1.3. Imposing sparsity on ℓ^2 solutions

Although balancing the dataset for maximum accuracy significantly improves performance, it is perplexing that least-squares seemingly contradicts the findings of [23,24] with 7% less accuracy and 20% lower recall than LASRC. Are LASRC's performance benefits coming from simple sparsity or ℓ^1-minimization? To investigate, we propose a hypothetical Thresholded L2 algorithm that imposes sparsity on ℓ^2 solutions by thresholding low magnitude coefficients to zero. Thresholded L2 is identical to LASRC's approximation step except it bypasses the second ℓ^1-minimization step to isolate the effect of sparsity.

For analysis, we varied sparsity from 0% to 99.9% and the balancedness of the Facebook dataset from unbalanced (all images with variable faces per person) to completely balanced (25 training faces per person). The results graphed in Fig. 8 provide several key insights. First, simple sparsity does not appreciably increase recall and in fact decreases accuracy when datasets are completely balanced, which agrees with [23,24]. Second, what is surprising is that even the crude, brute-force imposition of sparsity by Thresholded L2 can increase performance of both accuracy and recall significantly in the unbalanced cases. The results in Fig. 8 suggest that least-squares [23,24] with local features are not ideal for naturally unbalanced, open-universe data such as Facebook as even very simple sparse methods can better take advantage of extra user photos available for training to provide superior performance. Sophisticated ℓ^1-minimization methods for imposing sparsity can further increase recall to outperform least-squares by 12–32% (Table 4).

6.1.4. LASRC vs. Least-squares speed

A puzzling result from Table 4 is that LASRC (GPSR) classifies faster than least-squares (L2) even though LASRC includes the same ℓ^2 step in addition to ℓ^1-minimization. The reason for this discrepancy is that least-squares calculates residuals (4) for all classes whereas LASRC only calculates residuals for classes represented by the $K = 64$ selected training samples. In fact, the difference between L2 and Thresholded L2 shows that calculating residuals takes over half of the classification time. Thus with a fast ℓ^1-solver, LASRC can be 2 times faster than least-squares on our largest FB dataset with 1024 identities.

6.2. Locality

Recognizing the value of sparsity, but unable to accept the slow performance of even the fastest ℓ^1-solvers [65], Nan and Jian [29] and Li et al. [28] both proposed locality approximations to SRC. KNN-SRC [29], selects a small subset of nearby training samples for ℓ^1-minimization to greatly speed up SRC. LLC [19] replaces the ℓ^1-minimization step with a weighted least-squares emphasizing locality. Similarly to KNN-SRC, SVM-KNN [64] trains a local SVM to classify each test sample. Refer to Fig. 3 for a hierarchy of algorithms. The screening rules of [55,56] are based on the correlation of the test sample with the training samples, which has an equivalence to Euclidean distance when samples are normalized and thus performs within 0.1% of KNN-SRC (proof omitted for brevity).

The goal of approximating SRC is to select a small set of training samples for ℓ^1-minimization so that classification time is greatly reduced while maintaining performance similar to SRC. KNN-SRC [28,29] proposes nearest neighbor approximation based on the assumption that a Euclidean distance metric will select faces of the same class as the test face. However, we claim samples in ℓ^1-sparse solutions are not necessarily local under this metric; therefore it is better to select training samples that would be chosen by ℓ^1-minimization, which can be approximated with linear regression (least squares). To evaluate this claim, we examine recovered coefficients for a typical test image from an FB512 dataset in Fig. 9. All methods exhibit a peak at the correct class, so Fig. 9(b) shows a zoomed in view of the correct class. Notice LASRC with linear regression weighs samples more similarly to SRC (ℓ^1) than KNN-SRC or ℓ^2.

6.2.1. KNN vs. Linear regression approximation

For a quantitative evaluation of the best metric of locality to approximate ℓ^1-minimization, we created dictionaries of randomly generated synthetic samples with the same parameters as Yang et al. [65]. For 10,000 test samples (randomly generated from the dictionary with noise), we calculated the energy or overlap of samples selected by nearest neighbor and linear regression with the full sparse solution found by ℓ^1-minimization as we varied K. Fig. 10(a) shows that linear regression captures the energy of the ℓ^1-minimization solution with much fewer samples than nearest neighbor. Repeating the same experiment with 10,000 samples from real Facebook data confirms that linear regression approximates ℓ^1-minimization better than nearest neighbor (Fig. 10(b)).

6.2.2. Locality speed optimizations

To ensure fair speed comparisons between locality metrics, both KNN and linear regression were optimized. Linear regression was optimized as a single multiplication A^+y of the test sample y with the pre-calculated pseudoinverse A^+. Performing KNN na-

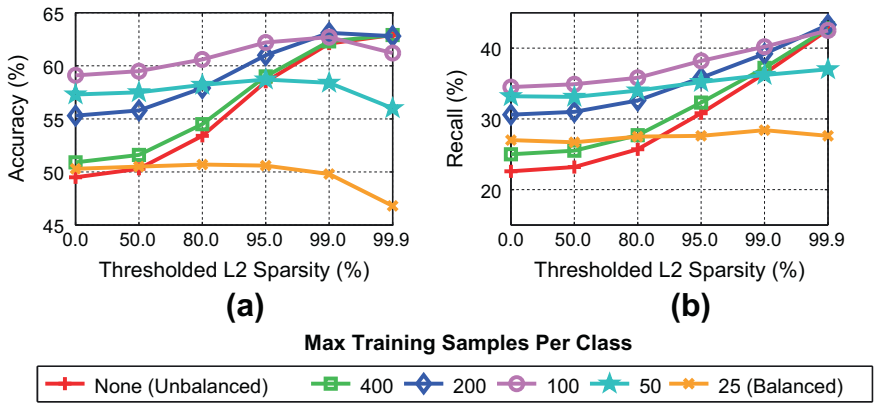


Fig. 8. Thresholded L2 performance on Facebook as sparsity and balancedness is varied. (a) Accuracy increases with sparsity for unbalanced datasets (b) Sparsity increases recall at 95% precision for all but the completely balanced case.

Please cite this article in press as: E.G. Ortiz, B.C. Becker, Face recognition for web-scale datasets, Comput. Vis. Image Understand. (2013), http://dx.doi.org/10.1016/j.cviu.2013.09.004

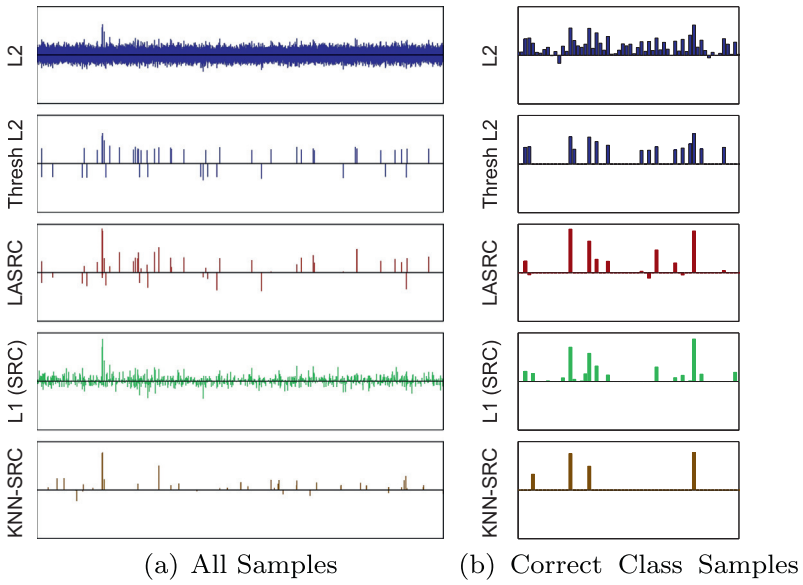


Fig. 9. Recovered coefficients from an example Facebook test face for (a) all training samples and (b) zoomed in only on the training samples from the correct class (also corresponding to the peak in (a)).

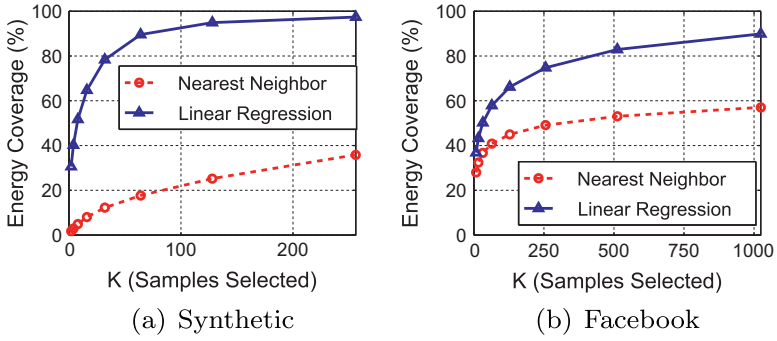


Fig. 10. Percent of ℓ^1-solution selected by approximation algorithms (weighted by coefficient magnitude) from 10,000 test samples drawn from (a) random synthetic data and (b) a Facebook dataset.

ively is slow, but we optimized it by omitting the square root, expanding the term $\|(A_i - y)\|^2$ into $\|A_i\|^2 + \|y\|^2 - 2\|A_i^T y\|^2$, vectorizing the n dot products $A_i^T y$ into a single matrix multiplication $A^T y$, and pre-calculating $\|A_i\|^2$. For further speedups, B test samples denoted as $Y = [y_1, \ldots, y_B]$ can be batch multiplied as $A^+ Y$ or $2\|A^T Y\|$ to take advantage of memory caching. Because many photos are often uploaded at once as an album, we feel processing several test samples simultaneously is reasonable. We used a batch size of $B = 16$, which yielded a 4–5X speedup for both algorithms as seen in Fig. 11(a).

6.2.3. Locality performance on Facebook

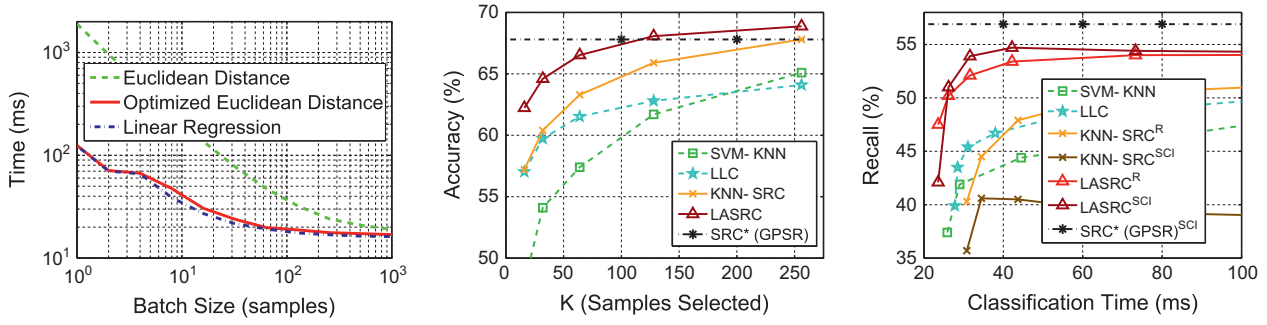
We evaluated locality approximating methods of SVM-KNN, KNN-SRC, LLC, and LASRC on Facebook data as K was varied (we omit OMP because it is too slow). In a closed-universe scenario reported in Fig. 11(b), LASRC achieves the best accuracy. As expected, KNN-SRC begins to converge with LASRC as K approaches the total number of faces n, when both become SRC. Although accuracy is informative, Fig. 11(c) shows classification time vs. recall at 95% precision in an open-universe scenario for a more realistic comparison. We also investigated using SCI vs. residuals for the probability of a distractor and concluded that SCI aids LASRC while degrading KNN-SRC's performance. In all cases, LASRC performs faster and with higher recall than all other locality-approximating methods.

7. Comparison on face verification

Although SRC methods are designed to exploit information from many different faces of a particular subject and is thus best suited for identification tasks, we can adapt it to the verification task. Given a dictionary, ℓ^1-minimization is performed for both images in the face pair to recover their respective coefficients similar to [68]; instead of calculating residuals per class, calculating the cosine distance between the faces' coefficient vectors yields a similarity metric that is surprisingly powerful. SRC for face verification requires no class information at all and is thus completely unsupervised. To avoid the intractability of using a large dictionary, we propose employing LASRC's dictionary approximation via least squares regression to select candidate pools of images for ℓ^1-minimization. In this section, we evaluate the applicability of SRC and LASRC to the task of face verification using two popular face verification datasets: Labeled Faces in the Wild (LFW) [10] and the Good, the Bad, and the Ugly (GBU) [15].

7.1. Labeled faces in the wild results

Labeled Faces in the Wild (LFW), previously described as a popular face verification dataset, challenges algorithms to identify if a pair of faces captured in uncontrolled conditions represent the same person. For this task, we must further adapt LASRC, where



(a) NN vs. Linear Regression Time (b) Facebook Accuracy (c) Facebook Recall vs. Time

Fig. 11. Analysis of locality approximating algorithms. (a) Both nearest neighbor and linear regression see speed benefits from batch calculations because of caching effects. (b) Accuracy on Facebook as K increases. (c) Recall at 95% precision vs. classification time as K increases. For LASRC and KNN-SRC, confidence calculated with SCI and residuals ^{R} are shown. SRC is shown as a straight line for reference (actual K or classification time are too high to show on the graphs). *SRC tuned for max recall rather than accuracy with $\lambda = 0.05$ so LASRC is able to achieve higher accuracy in (b) (SRC with $\lambda = 0.01$ yields max accuracy, but is too computationally expensive).

unlike face identification in which the dictionary subset is formed from the ℓ^2 coefficients from a single image, we instead take the absolute value of the ℓ^2 coefficients from each image in the pair, add them together, and select the highest resulting summed coefficients. This method selects faces that are correlated with both images, resulting in a more representative dictionary on which to perform ℓ^1-minimization and calculate a similarity from the cosine distance of the ℓ^1 coefficients. We use $K = 400$ for the dictionary size and a combination of HOG, LBP, and Gabor features and average the resulting similarities. When applied to verification, SRC and LASRC do not at any time use any ground truth information and are thus completely unsupervised algorithms as they do not require any class labels for any of the pairs.

Fig. 12 shows the ROC curve for SRC, LASRC, and several other unsupervised LFW methods. Following [68], we select a small dictionary of randomly chosen faces on which we apply SRC. LASRC outperforms a number of existing algorithms and boosts SRC performance by selecting a dictionary more correlated than randomly chosen faces. Table 5 lists existing accuracies with standard error for state-of-the-art algorithms, showing LASRC increases accuracy over SRC by ~3%. Even though LASRC is designed primarily as a face identification algorithm that excels at exploiting information from multiple images of individuals, it does well against many face verification algorithms on the LFW dataset.

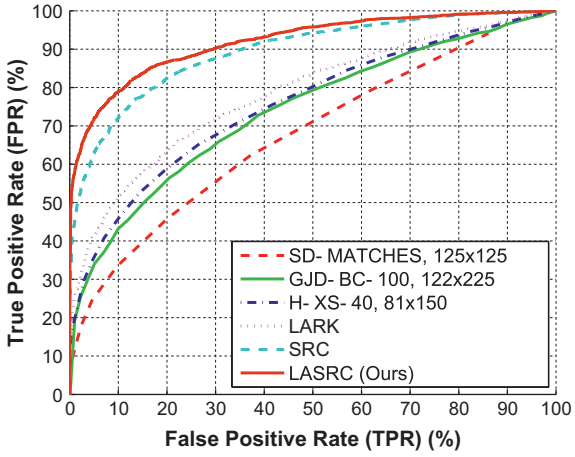


Fig. 12. ROC curves on Labeled Faces in the Wild (LFW) dataset for unsupervised algorithms from http://vis-www.cs.umass.edu/lfw/results.html (some algorithms did not provide ROC curve data, please reference papers cited in Table 5). Note SRC is our implementation of [68] and the addition of LASRC boosts performance.

7.2. The good, the bad, and the ugly results

The Good, the Bad, and the Ugly dataset, as described in Section 2.4.2, evaluates verification methods on three partitions of data varying from least to greatest difficulty. Since our method is not originally intended for face identification and the dataset requires about ~38 million comparisons, LASRC requires several algorithmic modifications. The first is in the selection of the approximated dictionary. We use the 13k image LFW dataset as a source from which we approximate a small dictionary with 1000 elements using linear regression against all other LFW images. The second modification, as previously described, requires the computation of a coefficient vector for each image in the GBU dataset followed by a cosine distance computed between each pair. We found that fusing the similarity scores amongst HOG, LBP, and Gabor features with a $\lambda = 0.001$ for the ℓ^1-minimization resulted in the best results. Given the final similarity matrices for each data split (good, bad, and ugly), we compute ROC curves as specified in [15].

As seen in Fig. 13 and Table 6, LASRC performs competitively on the good partition, but lags largely in the bad and ugly sets. Because LASRC is completely unsupervised while the leading two methods use models built from class labels, LASRC is at the top of unsupervised methods on the good partition with V1-like [74].

7.3. LASRC verification summary

We attribute the degradation of the performance on the bad and ugly sets of GBU and the flattening of the ROC curve on LFW to the fact that the linear representation assumption struggles recon-

Table 5
Unsupervised results on Labeled Faces in the Wild (LFW). For SRC-based verification (second half of table), note that our SRC baseline is very similar to that of [68], and that our LASRC approach boosts accuracy by ~3% over SRC. Despite not being developed for face verification, LASRC performs reasonably well compared to state-of-the-art and improves results over SRC verification. Note the HybridSparse algorithm uses a dissimilarity score that we omit.

Algorithm	Accuracy ± SE (%)
SD-MATCHES, 125 × 125 [69]	64.10 ± 0.62
GJD-BC-100, 122 × 225 [69]	68.47 ± 0.65
H-XS-40, 81 × 150 [69]	69.45 ± 0.48
LARK [70]	72.23 ± 0.49
LHS [71]	73.40 ± 0.00
HybridSparse [68]	84.70 ± 0.47
I-LPQ* [72]	86.20 ± 0.46
SRC (Ours)	81.14 ± 0.24
LASRC (Ours)	84.13 ± 0.36

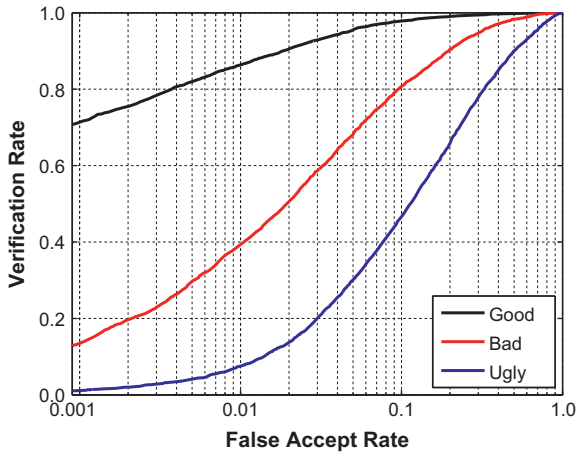


Fig. 13. GBU ROC. Verification rate vs. false accept rate for the good, bad, and ugly division of the GBU dataset. Good does well with performance degrading as the difficulty increases.

Table 6
Verification rate at false accept rate of 0.1% for the good, bad, and ugly partitions. LASRC performs competitively on the good partition, but does poorly on the bad and ugly partitions.

Method	Good (%)	Bad (%)	Ugly (%)	Training set
FRVT Fusion* [15]	98.0	80.0	15.0	Proprietary
CohortLDA* [73]	83.8	48.2	11.4	GBUx8
V1-like [74]	73.0	24.1	5.8	GBUx8
Kernel GDA [75]	69.1	28.5	5.1	GBUx8
LRPCA [15]	64.0	24.0	7.0	GBUx8
EBGM [76]	50.0	8.1	1.9	FERET
LBP [54]	51.4	5.0	1.9	None
LASRC (Ours)	71.3	13.5	1.1	LFW

* Supervised algorithms, which depend on class models built using identities of faces.

structing the test images as the parameters like pose or blurriness vary. It is important to note that performance would benefit from selecting an approximated dictionary between each pair of the GBU dataset as was done with LFW; however, due to the large quantity of comparisons necessary and its computational cost, the only option is a global approximated dictionary. Furthermore, Fig. 7 shows a marked increase in algorithmic performance as more faces of the same person become available, leading us to wonder if comparing only two images at a time is a limiting factor in using verification in web scenarios. Overall, LASRC performs reasonably well and competitively for an unsupervised algorithm under easier verification tasks, but struggles as the data becomes more limited and challenging. The question is how do these results translate to the much more difficult task of identification?

8. Comparison to state-of-the-art identification

To evaluate the holistic performance of LASRC against current state-of-the-art algorithms on a large scale, we used realistic PubFig + LFW (Section 2.4.3) and Facebook (Section 3) datasets. We differentiate between non-realtime algorithms, which are often higher performing, but too slow to be useful in real-world scenarios (either during training or classification), and realtime algorithms, which are much faster but often not as accurate. Refer to Fig. 3 for a hierarchy of tested algorithms.

8.1. Non-realtime algorithms

Four algorithms from Table 3 suffer from slow training or classification times: SVMs, SRC, OMP, and MTJSRC. We omit algorithms like GSRC [20] because they cannot use multiple features. For the baseline SRC algorithm, we test with two ℓ^1-solvers: Homotopy [47] and GPSR [46]. We tuned Homotopy for speed with a lower tolerance $tol = 10^{-3}$. We optimized GPSR for $B = 16$ batched operation (Section 6.2.2) and tuned for maximum recall with $\lambda = 0.05$ ($\lambda = 0.01$ yields higher accuracy, but lower recall with slower classification times). To validate the applicability of SRC in real-world situations, we also compare against the popular SVM approach using the large-scale, one-vs-all LIBLINEAR [43] algorithm optimized with dense data support for faster training [41] and a slack value of $c = 1$. Wolf et al. [25] demonstrated a One-Shot Similarity Score (OSS) kernel boosts accuracy with few training images; however, we find a linear SVM works just as well for large datasets. MTJSRC [34], a late fusion, multi-feature SRC approach, was tuned for two iterations for best performance. OMP was performed with $K = 64$ and batch optimized with $B = 16$ (same as LASRC, KNN-SRC, and LLC).

8.2. Realtime algorithms

The remaining eight algorithms from Table 3 are more suited to realtime operation: NN, SVM-KNN [64], LLC [19], KNN-SRC [29], LRC [27], L2 [23], CRC_RLS [24], and LASRC (Ours). Except for SVM-KNN, all realtime algorithms classify multiple test samples at once with a batch parameter of $B = 16$ (Section 6.2.2). SVM-KNN uses the LibSVM library [77] to train a probabilistic, one-vs-all SVM with a pre-computed linear kernel for maximum speed. The locality approximating value $K = 64$ is used for SVM-KNN, LLC, KNN-SRC, and LASRC. For better performance with LRC, L2, and CRC_RLS, we balanced the datasets by random selection to a maximum of 100 and 200 training faces per identity for Facebook and PubFig + LFW, respectively. KNN-SRC and LASRC both use $\lambda = 0.01$ for the GPSR [46] ℓ^1-minimization algorithm, although we use the minimum residual as confidence for KNN-SRC and SCI to reject distractors for LASRC.

8.3. PubFig + LFW and Facebook performance

Using the real-world datasets from Sections 2.4.3 and 3, we compare LASRC performance to other algorithms in both closed-universe and open-universe scenarios.

Table 7
PubFig + LFW (200 classes). Recall at 95% precision (open-universe), Accuracy (closed-universe), and classification time per test face (two significant figures only) for PubFig + LFW. All standard deviations are below 3%. Italicized entries indicate non-realtime times.

Algorithm	Recall (%)	Accuracy (%)	Time (ms)
Non-realtime			
SVM (Liblinear [43]) ‡ [25]	58.5	80.2	**1**
SRC (Homotopy [47])† [16]	72.2	72.2	*1800*
SRC (GPSR [46])*[16]	**73.9**	**81.8**	*4300*
OMP [49]	63.9	79.3	*1500*
MTJSRC [34]	44.3	70.1	*1300*
Realtime			
NN	38.2	65.8	**16**
SVM-KNN [64]	62.5	73.2	31
LLC [19]	66.0	77.8	22
KNN-SRC [29]	67.9	78.8	35
LRC [27]	48.3	70.9	30
L2 [23]	58.0	76.8	21
CRC-RLS [24]	54.9	73.5	23
LASRC (Ours)	**72.6**	**81.3**	27

* Tuned for maximum recall with $\lambda = 0.05$.
† Tuned for speed with $\lambda = 0.01$, $tol = 10^{-3}$.
‡ Tuned for maximum precision and recall without downsampling.

8.3.1. Closed-universe accuracy

As reported in Table 3, almost all algorithms achieved 99.5% or higher accuracy in small, controlled datasets. Although not our focus, we repeat a similar closed-universe comparison with large-scale, realistic datasets. Tables 7 and 8 show mean accuracy for PubFig (LFW is only used in open-universe scenarios) and Facebook (with 256, 512, and 1024 friend datasets). It is interesting to note that accuracies are significantly more varied and much lower, reaching a maximum of only 67–82%. On Facebook, SVMs achieve best accuracy with SRC (GPSR) trailing by 2.0–2.4%. On PubFig, SRC surpasses SVMs by 1.6%, likely because SRC can better exploit the many more training samples per identity. Among the realtime algorithms, LASRC takes the lead by 2.0–4.4%. Additionally, LASRC achieves similar performance to SRC with only a 0.5–1.3% difference. We conclude that SRC is competitive with SVMs and LASRC best approximates SRC in closed-universe scenarios.

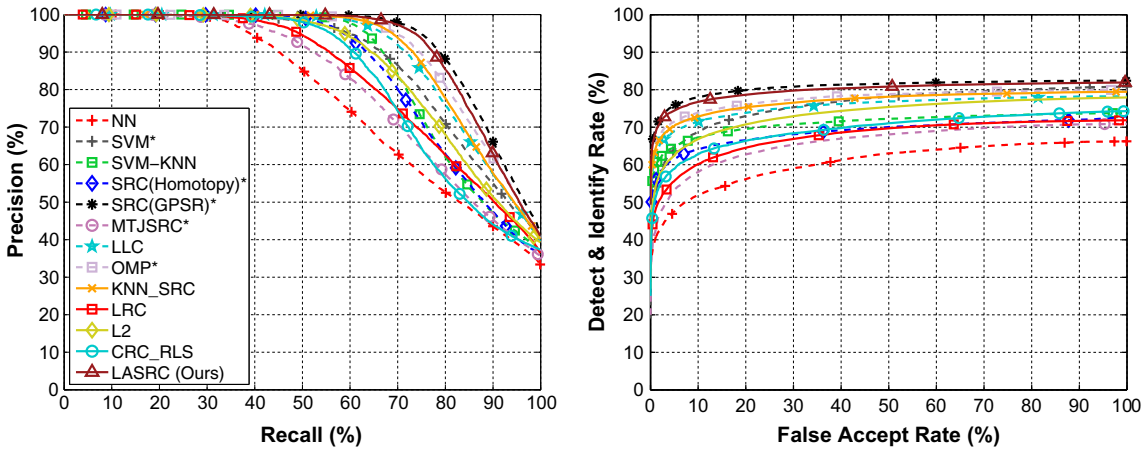
8.3.2. Open-universe precision and recall

Since face recognition algorithms must reject unknown identities in real-world environments, accuracy in a closed-universe is a poor metric for performance. We present more representative results in the form of open-universe PR and ROC curves and recall at 95% precision as described in Section 3.2 for PubFig + LFW (Fig. 14 and Table 7) and Facebook (Fig. 15 and Table 8) datasets. Overall, SRC exceeds all other non-realtime algorithms at high precision, besting even non-realtime SVMs by 5.1–15.4% and demonstrating sparse approaches can perform very well in real-world situations. Sparsity-enforcing KNN-SRC, LLC, and LASRC algorithms surpass the dense, least-squares approaches of LRC, L2, and CRC_RLS by >10%, confirming the usefulness of sparsity in open-universe scenarios. LASRC again surpasses all other realtime algorithms by 4.8–6.5%. LASRC's excellent performance is especially evident in Figs. 14 and 15 where it is the only realtime method to achieve a PR and ROC

Table 8
Facebook (256, 512, and 1024 classes). Recall at 95% precision (open-universe), Accuracy (closed-universe), and classification time per test face (two significant figures only) for three sizes of Facebook datasets. Italicized entries indicate non-realtime times. All standard deviations are below 3%.

	Facebook (256 classes)			Facebook (512 classes)			Facebook (1024 classes)			All
Algorithm	Recall (%)	Acc. (%)	Time (ms)	Recall (%)	Acc. (%)	Time (ms)	Recall (%)	Acc. (%)	Time (ms)	Max train time (min)
Non-realtime										
SVM (Liblinear [43])‡ [25]	54.1	**73.1**	**1**	50.9	**69.5**	**3**	50.0	**67.4**	**6**	*124.7*
SRC (Homotopy [47])† [16]	41.4	59.7	*1300*	36.9	54.3	*2600*	**34.8**	**50.8**	*5400*	**0.0**
SRC (GPSR [46])* [16]	**59.2**	71.1	*2400*	**56.4**	67.3	*5400*	**55.2**	65.0	*11000*	**0.0**
OMP [49]	51.3	68.3	*890*	49.5	63.1	*1600*	48.7	59.8	*2800*	**0.0**
MTJSRC [34]	30.5	58.9	*840*	23.9	51.2	*1800*	17.7	44.9	*4300*	**0.5**
Realtime										
NN	17.9	51.8	**11**	14.1	46.4	**21**	12.7	43.4	**44**	**0.0**
SVM-KNN [64]	50.5	62.6	31	45.1	56.8	42	42.0	52.6	61	**0.0**
LLC [19]	49.4	66.1	24	45.1	60.9	34	43.7	57.6	56	**0.0**
KNN-SRC [29]	51.7	67.8	55	47.8	62.8	67	46.0	59.3	90	**0.0**
LRC [27]	31.3	60.8	19	27.9	56.6	38	25.9	54.3	72	0.2
L2 [23]	41.5	65.3	23	34.0	58.8	44	27.9	53.	91	1.2
CRC-RLS [24]	45.0	63.9	24	36.2	57.4	46	30.6	52.5	95	2.0
LASRC (Ours)	**57.7**	**69.8**	22	**54.3**	**66.1**	29	**51.6**	**63.7**	44	1.3

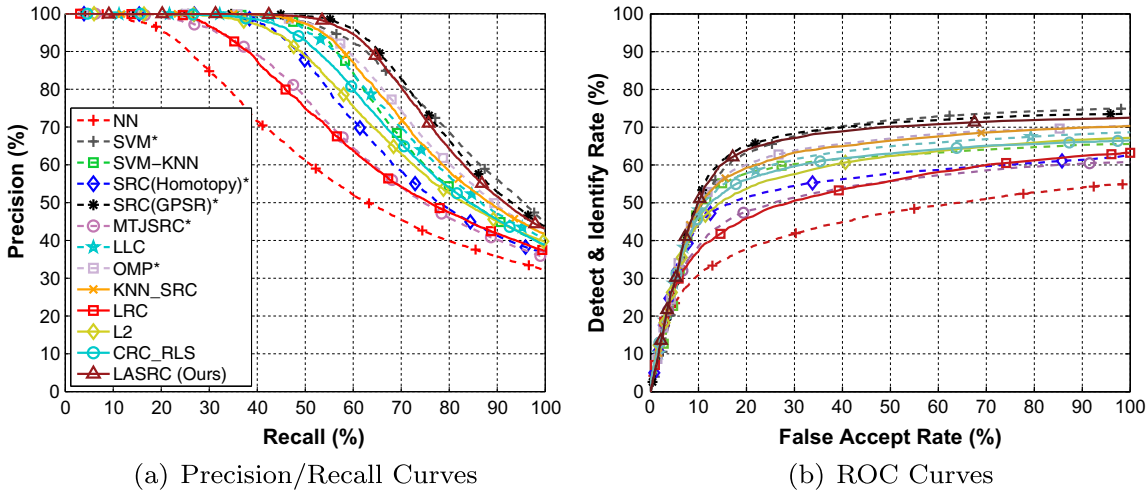
* Tuned for maximum recall with $\lambda = 0.05$.
† Tuned for speed with $\lambda = 0.01$, $tol = 10^{-3}$.
‡ Tuned for maximum precision and recall without downsampling.



(a) Precision/Recall Curves (b) ROC Curves

Fig. 14. Precision/Recall and ROC curves for PF + LFW. Of all the realtime algorithms, only LASRC achieves comparable performance to non-realtime methods such as SRC and SVMs.

Please cite this article in press as: E.G. Ortiz, B.C. Becker, Face recognition for web-scale datasets, Comput. Vis. Image Understand. (2013), http://dx.doi.org/10.1016/j.cviu.2013.09.004



(a) Precision/Recall Curves (b) ROC Curves

Fig. 15. Precision/Recall and ROC curves for PubFig + LFW. Of all the realtime algorithms, only LASRC achieves comparable performance to non-realtime methods such as SRC and SVMs.

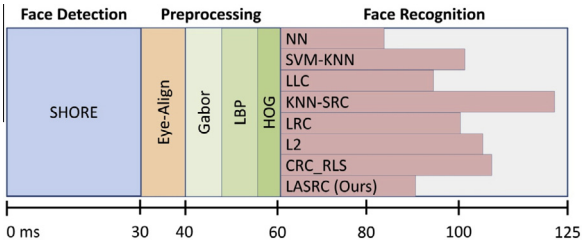


Fig. 16. Timeline of all steps in the entire face recognition system. All times reported with a single core of a 2.27 GHz machine.

curves similar to non-realtime algorithms, such as SRC and SVMs. More precisely, LASRC can classify over half of all seen faces with 95% precision, a recall rate that exceeds SVMs by 1.6–14.1%. Further, we completely outperform the non-realtime algorithms of OMP, MTJSRC, and Homotopy.

8.3.3. Training and classification times

One of the greatest advantages of LASRC is its scalability to large datasets while maintaining rapid classification at a mean rate of 30 Hz over all PubFig + LFW and Facebook datasets. On the largest Facebook dataset with over 90k training faces, LASRC classifies faster than all other realtime methods except NN. Furthermore, training time is under a minute except for the FB1024 datasets where it peaks at 2.1 min. While SVM classification is extremely fast, LASRC can train 95 times faster while still achieving similar or better recall at 95% precision. It is important to note that SVM training time can be reduced by limiting the maximum number of iterations; however by doing this, we found precision and recall dropped steeply while training time remained much higher than LASRC. Likewise, using 10,000 randomly subsampled negative examples for each class in the one-vs-all SVM reduced training by 4 times, but also significantly reduced recall by 9–16%. Even with these speedups, LASRC still trains 25 times faster than SVMs. Therefore, we present results with LIBLINEAR's default maximum number of iterations and without any subsampling. While LASRC only approximates SRC's performance, we feel a 2.1% mean drop in recall at 95% precision is worth reducing classification from 4–11 s to 22–44 ms, a 100–250 times speedup. Fig. 16 depicts the timeline for realtime methods.

9. Conclusions

In this paper, we present a novel Linearly Approximated SRC (LASRC) algorithm that excels at large-scale, realistic face identification tasks in open-universe scenarios where unknown and distractor faces must be rejected. Combining the speed of least-squares with the robustness of sparse representations, LASRC improves upon SRC with only one extra, easily-tunable parameter K. By selecting a small pool of K training samples for ℓ^1-minimization with a linear regression approximation, classification time is greatly reduced with only a small loss in recall. We extensively evaluate traditional, sparse, and least-squares algorithms with respect to sparsity and locality under real-world scenarios on two very large and diverse face datasets: (1) a combination of PubFig and LFW and (2) a new Facebook dataset. Our results show linearly approximated sparse representations with local features are very much applicable to real-world face identification tasks. While popular algorithms may be less-suited to dynamic, web-scale scenarios because of slow training times (SVMs) or slow classification (SRC), LASRC represents a good compromise that both trains and classifies rapidly while retaining good recall and precision. LASRC exhibits the advantages of SRC with at least 100X faster classification and achieves better performance than other fast sparse methods. Furthermore, our approach compares well to SVMs while training orders of magnitude more rapidly, even against state-of-the-art algorithms designed for speed and tuned for fast, approximate training. Finally, our approach outperforms many recent realtime algorithms in speed, accuracy, and recall.

In the future, better sample selection for the training set, a more sophisticated method of rejecting distractors, and tighter integration with ℓ^1-minimization algorithms could benefit LASRC. For faster performance, one could reduce dimensionality during the linear regression step and reduce ℓ^1-minimization iterations for speed without significantly impacting performance. Similarly, multi-threading or GPU acceleration would likely speed up LASRC by several times. For better accuracy, new feature representations could be explored. In situations where many training faces per subject or frontal faces are not available, more evaluation is needed. Performance could be boosted with expectation–maximization, where candidate samples are proposed and ℓ^1-minimization evaluates them.

While our presented approach is a promising step towards fast, web-scale face recognition, there is much room for improvement. We hope that by releasing descriptors for our datasets, a utility to download and create datasets from Facebook, and a MATLAB

Please cite this article in press as: E.G. Ortiz, B.C. Becker, Face recognition for web-scale datasets, Comput. Vis. Image Understand. (2013), http://dx.doi.org/10.1016/j.cviu.2013.09.004

toolkit for face recognition, future researchers will be able to more easily develop and evaluate new algorithms for realistic, open-universe face recognition scenarios.

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship and the Florida Education Fund McKnight Doctoral Fellowship. Special thanks to those that provided proofreading and all of those who volunteered their time to help collect data from Facebook.

References

[1] A.T.L. Cambridge, The Database of Faces. <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>.
[2] A.R. Martinez, R. Benavente, The AR Face Database, Tech. Rep., Computer Vision Center (CVC), 1998.
[3] A. Georghiades, D. Kriegman, P.N. Belhumeur, From few to many: generative models for recognition under variable pose and illumination, TPAMI 23 (6) (2001) 643–660.
[4] P.J. Phillips, H. Wechsler, J. Huang, P.J. Rauss, The FERET database and evaluation procedure for face-recognition algorithms, IVC 16 (5) (1998) 295–306.
[5] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression database, TPAMI 25 (2003) 1615–1618.
[6] A. O'Toole, P. Phillips, F. Jiang, J. Ayyad, N. Pénard, H. Abdi, Face recognition algorithms surpass humans matching faces over changes in illumination, TPAMI 29 (9) (2007) 1642–1646.
[7] P. Phillips, P. Grother, R. Micheals, D. BlackBurn, E. Tabassi, M. Bone, Nat'l Inst. of Standards and Technology Interagency/Internal Report (NISTIR) 6965.
[8] P. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, in: CVPR, 2005, pp. 947–954.
[9] P.J. Grother, G.W. Quinn, P.J. Phillips, Report on the Evaluation of 2d Still-Image Face Recognition Algorithms, Nat'l Inst. of Standards and Technology Interagency/Internal Report (NISTIR) 7709.
[10] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Tech. rep., University of Massachusetts, Amherst, 2007.
[11] N. Kumar, A. Berg, P. Belhumeur, S. Nayar, Describable visual attributes for face verification and image search, TPAMI 33 (10) (2011) 1962–1977.
[12] Z. Stone, T. Zickler, T. Darrell, Autotagging Facebook: social network context improves photo annotation, in: CVPR Workshop, IEEE, 2008, pp. 1–8.
[13] B. Becker, E. Ortiz, Evaluation of face recognition techniques for application to Facebook, in: FG, IEEE, 2008, pp. 1–6.
[14] N. Pinto, Z. Stone, T. Zickler, D. Cox, Scaling up biologically-inspired computer vision: a case study in unconstrained face recognition on facebook, in: CVPR, IEEE, 2011, pp. 35–42.
[15] P.J. Phillips, J.R. Beveridge, B.A. Draper, G. Givens, A.J. O'Toole, D.S. Bolme, J. Dunlop, Y.M. Lui, H. Sahibzada, S. Weimer, An introduction to the good, the bad, & the ugly face recognition challenge problem, in: FG, IEEE, 2011, pp. 346–353.
[16] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, TPAMI 31 (2) (2009) 210–227.
[17] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, Y. Ma, Towards a practical face recognition system: robust alignment and illumination by sparse representation, TPAMI (34) (2011) 372–386.
[18] W. Zhao, R. Chellappa, P. Phillips, A. Rosenfeld, Face recognition in still and video images: a literature survey, CSUR (2003) 399–458.
[19] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: CVPR, IEEE, 2010.
[20] M. Yang, L. Zhang, Gabor feature based sparse representation for face recognition with gabor occlusion dictionary, in: ECCV, IEEE, 2010, pp. 448–461.
[21] J. Huang, M. Yang, Fast sparse representation with prototypes, in: CVPR, IEEE, 2010, pp. 3618–4625.
[22] Q. Shi, C. Shen, H. Li, Rapid face recognition using hashing, in: CVPR, IEEE, 2010, pp. 2753–2760.
[23] Q. Shi, A. Eriksson, A. van den Hengel, C. Shen, Is face recognition really a compressive sensing problem? in: CVPR, 2011, pp. 553–560.
[24] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: which helps face recognition? in: ICCV, 2011.
[25] L. Wolf, T. Hassner, Y. Taigman, Effective unconstrained face recognition by combining multiple descriptors and learned background statistics, TPAMI 33 (10) (2011) 1978–1990.
[26] J.R. del Solar, R. Verschae, M. Correa, Recognition of faces in unconstrained environments: a comparative study, EURASIP JASP (2009) 1:1–1:19.
[27] I. Naseem, R. Togneri, M. Bennamoun, Linear regression for face recognition, TPAMI 32 (2010) 2106–2112.
[28] C. Li, J. Guo, H. Zhang, Local sparse representation based classification, in: ICPR, IEEE, 2010, pp. 649–652.
[29] Z. Nan, Y. Jian, K nearest neighbor based local sparse representation classifier, in: CCPR, IEEE, 2010, pp. 1–5.
[30] C. Chan, J. Kittler, Sparse representation of (multiscale) histograms for face recognition robust to registration and illumination problems, in: ICIP, IEEE, 2010, pp. 2441–2444.
[31] R. Gross, I. Matthews, J. Cohn, T. Kanade, S. Baker, Multi-pie, Image Vision Comput. 28 (5) (2010) 807–813.
[32] E. Bailly-Baillire, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Marithoz, J. Matas, K. Messer, V. Popovici, F. Pore, B. Ruiz, J.-P. Thiran, The BANCA database and evaluation protocol, in: AVBPA, Lecture Notes in Computer Science, vol. 2688, 2003, pp. 625–638.
[33] K. Messer, J. Kittler, M. Sadeghi, S. Marcel, C. Marcel, S. Bengio, F. Cardinaux, C. Sanderson, J. Czyz, L. Vandendorpe, S. Srisuk, M. Petrou, W. Kurutach, A. Kadyrov, R. Paredes, B. Kepenekci, F. Tek, G. Akar, F. Deravi, N. Mavity, Face verification competition on the XM2VTS database, in: AVBPA, Lecture Notes in Computer Science, vol. 2688, 2003, pp. 964–974.
[34] X. Yuan, S. Yan, Visual classification with multi-task joint sparse representation, in: CVPR, IEEE, 2010, pp. 3493–3500.
[35] Q. Yin, X. Tang, J. Sun, An associate-predict model for face recognition, in: CVPR, IEEE, 2011, pp. 497–504.
[36] P. Grother, P. Phillips, Models of large population recognition performance, in: CVPR, 2004.
[37] F. Li, H. Wechsler, Open set face recognition using transduction, IEEE Trans. Pattern Anal. Machine Intel. 27 (11) (2005) 1686–1697.
[38] H.K. Ekenel, L. Szasz-Toth, R. Stiefelhagen, Open-set face recognition-based visitor interface system, ICVS (2009) 43–52.
[39] H. Gao, H.K. Ekenel, R. Stiefelhagen, Robust open-set face recognition for small-scale convenience applications, DAGM (2010) 393–402.
[40] W. Scheirer, A. Rocha, A. Sapkota, T. Boult, Towards open set recognition, TPAMI (99) (2012) 1.
[41] S. Maji, J. Malik, Fast and Accurate Digit Classification, Tech. Rep., EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-159, 2009.
[42] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, T. Huang, Large-scale image classification: fast feature extraction and SVM training, in: CVPR, IEEE, 2011, pp. 1689–1696.
[43] R. Fan, K. Chang, C. Hsieh, X. Wang, C. Lin, Liblinear: a library for large linear classification, J. Machine Learn. Res. 9 (2008) 1871–1874.
[44] S.S. Chen, D.L. Donoho, Michael, A. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1998) 33–61.
[45] E.J. Cands, J.K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, CPAM 59 (8) (2006) 1207–1223.
[46] M. Figueiredo, R. Nowak, S. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, STSP, 2007.
[47] D. Malioutov, M. Cetin, A. Willsky, Homotopy continuation for sparse signal representation, ICASSP, vol. 5, IEEE, 2005, pp. 733–736.
[48] J. Yang, Y. Zhang, Alternating Direction Algorithms for l1-Problems in Compressive Sensing, Tech. Rep., Rice University, 2010.
[49] J.A. Tropp, Greed is good: algorithmic results for sparse approximation, IEEE Trans. Infor. Theor. 50 (2004) 2231–2242.
[50] Y. Peng, A. Ganesh, J. Wright, W. Xu, Y. Ma, RASL: robust alignment by sparse and low-rank decomposition for linearly correlated images, TPAMI, 2010.
[51] K. Wu, L. Wang, F. Soong, Y. Yam, A sparse and low-rank approach to efficient face alignment for photo-real talking head synthesis, in: ICASSP, IEEE, 2011, pp. 1397–1400.
[52] V.M. Patel, T. Wu, S. Biswas, P.J. Phillips, R. Chellappa, Dictionary-based face recognition under variable lighting and pose, Trans. Inform. Forensics Secur., 2012.
[53] G. Huang, V. Jain, E. Learned-Miller, Unsupervised joint alignment of complex images, in: ICCV, IEEE, 2007, pp. 1–8.
[54] T. Ahonen, A. Hadid, M. Pietikäinen, Face description with local binary patterns: application to face recognition, TPAMI, 2006.
[55] L.E. Ghaoui, V. Viallon, T. Rabbani, Safe Feature Elimination in Sparse Supervised Learning, arXiv 1009.4219.
[56] Z. Xiang, H. Xu, P. Ramadge, Learning sparse representations of high dimensional data on large scale dictionaries, in: NIPS, 2011.
[57] H. Xu, C. Caramanis, S. Mannor, Sparse algorithms are not stable: a no-free-lunch theorem, TPAMI 34 (1) (2012) 187–193.
[58] F. IIS, Shore, 2010. <http://www.iis.fraunhofer.de/EN/bf/bv/kognitiv/biom/dd.jsp>.
[59] C. Kueblbeck, A. Ernst, Face detection and tracking in video sequences using the modified census transformation, Image Vision Comput. 24 (6) (2006) 564–572.
[60] M. Everingham, J. Sivic, A. Zisserman, Hello! My Name is... Buffy–Automatic Naming of Characters in TV Video, 2006.
[61] A. Torralba, A. Efros, Unbiased look at dataset bias, in: CVPR, IEEE, 2011, pp. 1521–1528.
[62] C. Liu, H. Wechsler, Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition, TIP 11 (4) (2002) 467–476.
[63] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, vol. 1, 2005, pp. 886–893.
[64] H. Zhang, A. Berg, M. Maire, J. Malik, SVM-kNN: discriminative nearest neighbor classification for visual category recognition, in: CVPR, IEEE, 2006, pp. 2126–2136.
[65] A. Yang, A. Ganesh, Z. Zhou, S. Sastry, Y. Ma, Fast l1-minimization algorithms and an application in robust face recognition: a review, in: ICIP, IEEE, 2010, pp. 1849–1852.

Please cite this article in press as: E.G. Ortiz, B.C. Becker, Face recognition for web-scale datasets, Comput. Vis. Image Understand. (2013), http://dx.doi.org/10.1016/j.cviu.2013.09.004

[66] E. Candes, J. Romberg, l1-Magic: A Collection of Matlab Routines for Solving the Convex Optimization Programs Central to Compressive Sampling. <http://www.acm.caltech.edu/l1magic/>.
[67] S. Kim, K. Koh, M. Lustig, S. Boyd, An efficient method for compressed sensing, in: ICIP, IEEE, 2007, pp. 117–120.
[68] H. Guo, R. Wang, J. Choi, L.S. Davis, Face verification using sparse representationsin, in: CVPRW, IEEE, 2012.
[69] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: ICML, 2010.
[70] H. Seo, P. Milanfar, Face verification using the lark representation, Trans. Inform. Forensics Secur. 6 (4) (2011) 1275–1286.
[71] G. Sharma, S. ul Hussain, F. Jurie, Local higher-order statistics (LHS) for texture categorization and facial analysis, in: ECCV, 2012.
[72] S. ul Hussain, T. Napolon, F. Jurie, Face recognition using local quantized patterns, BMVC (2012).
[73] Y.M. Lui, D. Bolme, P. Phillips, J. Beveridge, B. Draper, Preliminary studies on the good, the bad, and the ugly face recognition challenge problem, CVPRW (2012) 9–16.
[74] N. Pinto, J. DiCarlo, D. Cox, How far can you get with a modern face recognition test set using only simple features?, in: CVPR, CVPR, 2009, pp.2591–2598.
[75] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Comput. 12 (10) (2000) 2385–2404.
[76] J. Beveridge, D. Bolme, B. Draper, M. Teixeira, The CSU Face Identification Evaluation System, MVA.
[77] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, TIST 2 (2011) 27:1–27:27. <http://www.csie.ntu.edu.tw/cjlin/libsvm>.

Enrique G. Ortiz received a B.S. and M.S. degree in computer engineering from the University of Central Florida in 2007 and 2009 respectively. He has been a Ph.D. student in the Computer Vision Lab at the University of Central Florida since 2007, with interests primarily in human action and facial recognition.

Brian C. Becker received a B.S. degree in computer engineering from the University of Central Florida in 2007 and an M.S. and Ph.D. degree in robotics from Carnegie Mellon University in 2010 and 2012, respectively. Since 2007, he has researched medical robotics in the Robotics Institute at Carnegie Mellon. He is currently employed at Carnegie Mellon's National Robotics Engineering Center where he specializes in robotic perception.

Please cite this article in press as: E.G. Ortiz, B.C. Becker, Face recognition for web-scale datasets, Comput. Vis. Image Understand. (2013), http://dx.doi.org/10.1016/j.cviu.2013.09.004