TAMING WILD FACES: WEB-SCALE, OPEN-UNIVERSE FACE IDENTIFICATION IN STILL AND VIDEO IMAGERY

by

ENRIQUE G. ORTIZ
B.S. University of Central Florida, 2007
M.S. University of Central Florida, 2009

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2014

Major Professor: Mubarak Shah

ABSTRACT

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 and videos. In this dissertation, we explore unconstrained still-image and video-based face recognition in real-world scenarios, e.g. social photo sharing and movie trailers, where people of interest are recognized and all others are ignored. In such a scenario, we must obtain high precision in recognizing the known identities, while accurately rejecting those of no interest.

Recent advancements in face recognition research has seen Sparse Representation-based Classification (SRC) advance to the forefront of competing methods. However, its drawbacks, slow speed and sensitivity to variations in pose, illumination, and occlusion, have hindered its wide-spread applicability. The contributions of this dissertation are three-fold:

1. For still-image data, we propose a novel Linearly Approximated Sparse Representation-based Classification (LASRC) algorithm that uses linear regression to perform sample selection for l1-minimization, thus harnessing the speed of least-squares and the robustness of SRC. On our large dataset collected from Facebook, LASRC performs equally to standard SRC with a speedup of 100-250x.

2. For video, applying the popular l1-minimization for face recognition on a frame-by-frame basis is prohibitively expensive computationally, so we propose a new algorithm Mean Sequence SRC (MSSRC) that performs video face recognition using a joint optimization leveraging all of the available video data and employing the knowledge that the face track frames belong to the same individual. Employing MSSRC results in a speedup of 5x on average over SRC on a frame-by-frame basis.

3. Finally, we make the observation that MSSRC sometimes assigns inconsistent identities to the same individual in a scene that could be corrected based on their visual similarity. There-

fore, we construct a probabilistic affinity graph combining appearance and co-occurrence similarities to model the relationship between face tracks in a video. Using this relationship graph, we employ random walk analysis to propagate strong class predictions among similar face tracks, while dampening weak predictions. Our method results in a performance gain of 15.8% in average precision over using MSSRC alone.

To God.

ACKNOWLEDGMENTS

I would like to thank my advisor, Mubarak Shah, and committee for their guidance throughout the research process. I would like to also acknowledge my family, friends, and labmates throughout the years that have accompanied me on this journey of self-discovery.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

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 and videos. Popular social networks, such as Facebook, allow users to place tags on photos to label people, encouraging collaboratively organized photo albums amongst friends, a simple, yet tedious task for humans. It is approximated that 350 million photos are uploaded to Facebook daily [1] and 100 hours of video uploaded to YouTube every minute [2], in addition to the large catalog of movies available on services like Apple iTunes, Google Play, and Amazon Instant, which easily translates to billions of faces to tag. Because visual interest is largely determined by who appears in the image, labeling identities is particularly important. Imagine millions of social network users needing to tag their photos; further imagine watching a movie, home video, or YouTube video and wanting to find all of the scenes with a particular person of interest. Such web-scale labeling problems present a real challenge and fascinating opportunity for automation by face recognition.

Face recognition's long history could be described best by its many datasets introduced over the years that addressed key challenges at the time of collection. Early datasets such as AT&T (ORL) [3], AR [4], Yale [5], FERET [6], and PIE [7] 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 [8] 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 [9], FRGC [10], and MBE [11] 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 [12], PubFig [13], and various private Facebook galleries [14–16]. This has spurred the

development of more robust algorithms, although humans still outperform the best approaches [13]. On controlled scenarios, face identification performance is excellent (~99.5%), as shown in Sec. 3.3.2.1. Further, on more realistic data like LFW performance is high at 95.1%. However, in both scenarios its good performance does not translate to web-scale recognition tasks. With the expanding capability to share photos online, it is imperative to break from the reliance on standard benchmark face datasets to more complicated, realistic datasets.

All face recognition tasks addressed by the aforementioned datasets fall into one of three categories (Fig. 1.1): closed-universe face identification, open-universe face verification, and open-universe face identification.

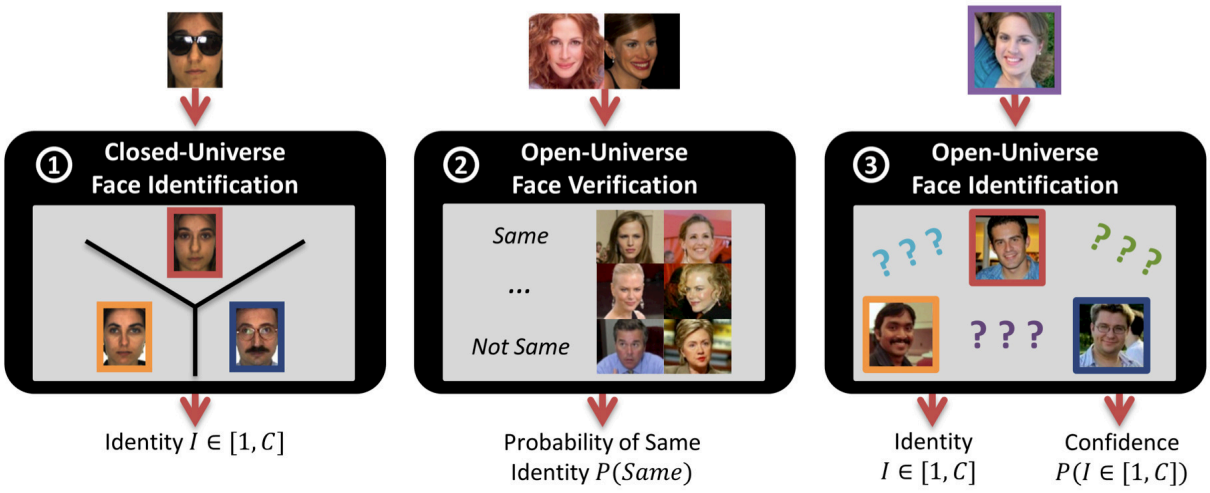
1. **Closed-Universe Face Identification**: Face recognition research, whether still-image or video-based, generally works in a closed identification framework where it is assumed that the classifier will only receive test images from subjects in the training model (Fig. 1.1(a)). In other words, given 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 setting is the most common form of face recognition with controlled datasets such as Extended Yale B, AR, MultiPIE, or FERET [14–30].

2. **Open-Universe Face Verification**: On the other hand, face verification techniques present a more open framework that returns a prediction that a pair of images is the same or not (Fig. 1.1(b)). In other words, is an input face's claimed identity correct? Because people can claim any identity, the verification task is open-universe. As popular datasets like LFW [12], PubFig [13], GBU [31], BANCA [32], and XM2VTS [33], the task is referred to as pair-matching.

3. **Open-Universe Face Identification**: However, more realistically, a complete open-universe, face identification scenario is necessary for most system deployments (Fig. 1.1(c)). This paradigm posits, 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, open-universe face identification is one of the least-studied tasks.



(a) Closed-Universe Face Identification

(b) Open-Universe Verification

(c) Open-Universe Face Identification

Figure 1.1: Three common face recognition tasks. Closed-universe face identification assumes all input face images are from a known class. Open-universe face verification simply assumes a pair of input images are the same or not the same. Open-universe face identification assumes the input test sample can be unknown.

Generally, web-scale tasks fall under open-universe face identification. For example, in a social network context, only friends should be tagged within a photo while other faces should be ignored or in the context of a movie, only known cast members or public figures should be tagged, while all others should be annotated as background actors. Moreover, imagine a task often referred to as the watch-list problem requires a security system to watch for a small, specific set of people of interest while ignoring all others. As the Multiple Biometrics Evaluation 2010 [11] concluded, "In practice, the open-set identification task is more difficult for biometric systems (and presumably

for human operators) than the verification task." The study further stated that in the watch-list task, which is a subproblem of face identification, classification becomes increasingly difficult as more identities are added to the list, similar to scaling from a specific set of actors in a movie to the large number users in a social-network scenario. Most recently, there has been interest in the object recognition community on open-universe recognition [34], however this development has yet to reach wide-spread interest in the face recognition community. Increasing attention in the research community at large and the difficulty of the problem highlight the necessity of evaluating real-world, open-universe facial identification.

Existing studies in open-universe face identification are either small-scale [6] or private, controlled, and tailored to specific application domains (mugshot and visa images) [9, 11]. Therefore, in contrast to existing face recognition studies, this dissertation pursues the real-world, open-universe face identification task, in which unknown identities must be rejected with high precision. To address this task we collect two datasets one for still-image face recognition using Facebook and another for video face recognition from YouTube. While exploring these datasets, we develop three methods for open-universe face identification for still-image and video application domains:

1. We present Linearly Approximated Sparse Representation-based Classification (LASRC) for fast classification or rejection of individuals in large-scale image databases.

2. We extend the Sparse Representation-based Classification (SRC) paradigm to the task of video face recognition using Mean Sequence Sparse Representation-based Classification (MSSRC) to accurately label known actors, while rejecting background actors.

3. We propose an Affinity-based Propagation technique that temporally and visually relates face tracks in a video to further improve performance in recognition.

In the subsequent sections, we motivate these three methods and their strengths with respect to their application domains.

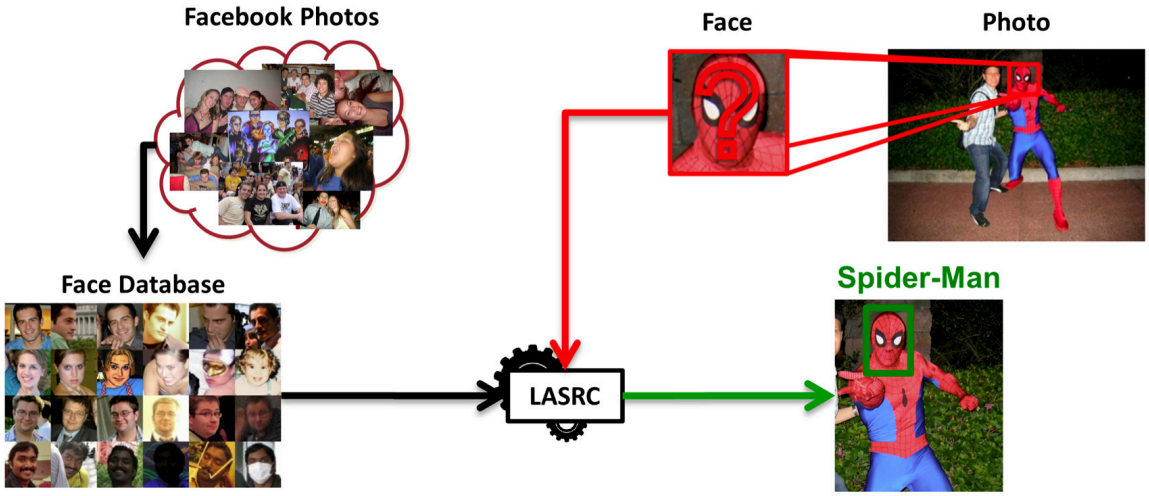


Figure 1.2: Still-Image Face Identification. We address the difficult problem of identifying a face from an unconstrained image with a dictionary of faces from many people, while rejecting unknown individuals.

1.1 Still-Image, Open-Universe Face Identification

In consumer-driven and Internet applications as depicted in Fig. 1.2, 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 [9], FRGC [10], and MBE [11] and verification datasets such as GBU [31] and LFW [12], 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, where we must identify specific people reliably while rejecting all others as distractors.

With the expanding capability to share photos online, face identification becomes crucial for the sharing and organization of images of interest. With existing research and datasets fo-

cusing on closed-universe face identification and open-universe face verification, it is imperative to break from the reliance on standard benchmark face datasets to more complicated, realistic datasets. Therefore, unlike past face identification studies, we generated large-scale, real-world datasets from Facebook emulating real photo-albums with user annotated face images for easy benchmarking (http://face.enriquegortiz.com).

Furthermore, classification is a crucial stage of any face recognition algorithm, where the goal is to match a query to its correct identity. Most recently, the ground-breaking work, Sparse Representation-based Classification (SRC) [35], showed that the assumption that a novel test image can be represented by a linear approximation of the training set can be used for classification. Although SRC has maintained high accuracies, its need for well-aligned, normalized data and computational complexity has received criticism, therefore its realistic application has been limited.

To address existing insufficiencies with SRC methods when scaling face identification to web-scale applications, we propose a novel and efficient algorithm named Linearly Approximated Sparse Representation-based Classification (LASRC). Inspired by these sparse methods [17, 35] that scale poorly as the number of training images increase (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.

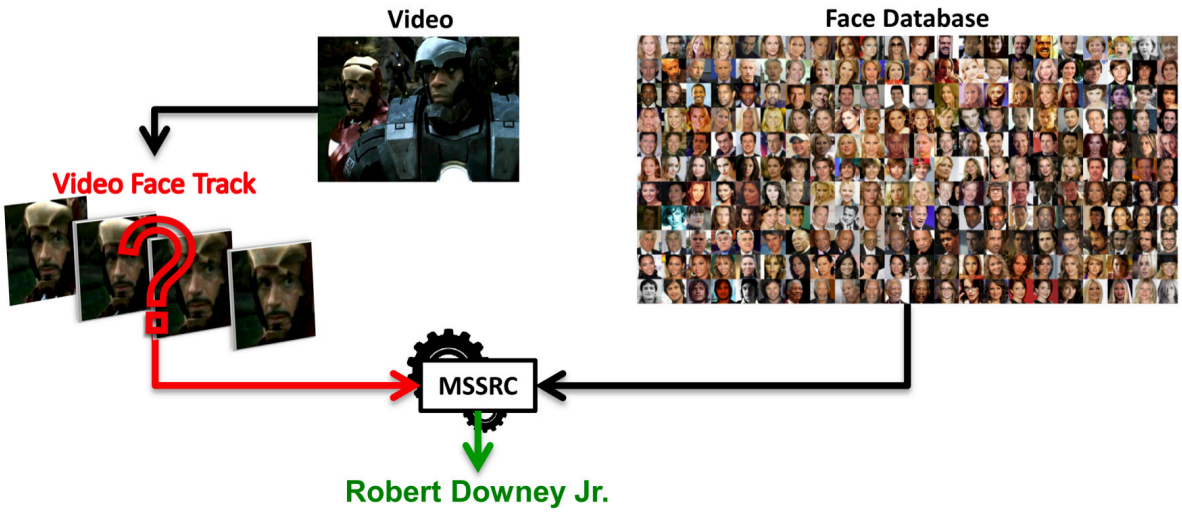


Figure 1.3: Video Based Face Identification. We address the difficult problem of identifying a video face track with a dictionary of still face images of many people, while rejecting unknown individuals.

1.2 Video-based, Open-Universe Face Identification

Face Recognition has received widespread attention for the past three decades due to its wide-applicability ranging from surveillance to photo album annotation. Only recently has this interest spread into the domain of video, where the problem becomes more challenging due to the person's motion and changes in both illumination and occlusions. However, it also has the benefit of providing many samples of the same person, thus providing the opportunity to convert many weak examples into a strong prediction of the identity.

As video search sites like YouTube have grown, video content-based search has become increasingly necessary. For example, a capable retrieval system should return all videos containing specific actors upon a user's request. On sites like YouTube, where a cast list or script may not be available, the visual content is the key to accomplishing this retrieval accurately. In this dissertation, we explore the often little-studied, open-universe scenario in which it is important to

recognize and reject unknown identities, i.e. we identify famous actors appearing in movie trailers while rejecting background faces that represent unknown extras.

The main drawback of video-based face recognition is the availability of annotated video face tracks. With the advent of social networking and photo-sharing, computer vision tasks on the Internet have become increasingly intriguing and viable. This avenue is one little exploited by video face recognition. Although large collections of annotated individuals in videos are not freely available, collecting data of annotated still images is easily doable, as witnessed by datasets like Labeled Faces in the Wild (LFW) [12] and Public Figures (PubFig) [13]. Due to wide availability, we employ large databases of still images to recognize individuals in videos, as depicted in Fig. 1.3.

Existing video face recognition methods tend to perform classification on a frame-by-frame basis and later combine those predictions using an appropriate metric. A straight-forward application of ℓ^1-minimization in this fashion is very computationally expensive. In contrast, we propose a novel method, Mean Sequence Sparse Representation-based Classification (MSSRC), that performs a joint optimization over all faces in the track at once. Though this seems expensive, we show that this optimization reduces to a single ℓ^1-minimization over the mean face track, thus reducing a many classification problem to one with inherent computational and practical benefits.

Our proposed method aims to perform video face recognition across domains, leveraging thousands of labeled, still images gathered from the Internet, specifically the PubFig and LFW datasets, to perform face recognition on real-world, unconstrained videos. To do this we collected 101 movie trailers from YouTube and automatically extracted and tracked faces in the video to create a dataset for video face recognition (`http://vfr.enriquegortiz.com`). We show our method outperforms existing methods in precision and recall, exhibiting the ability to better reject unknown or uncertain identities.

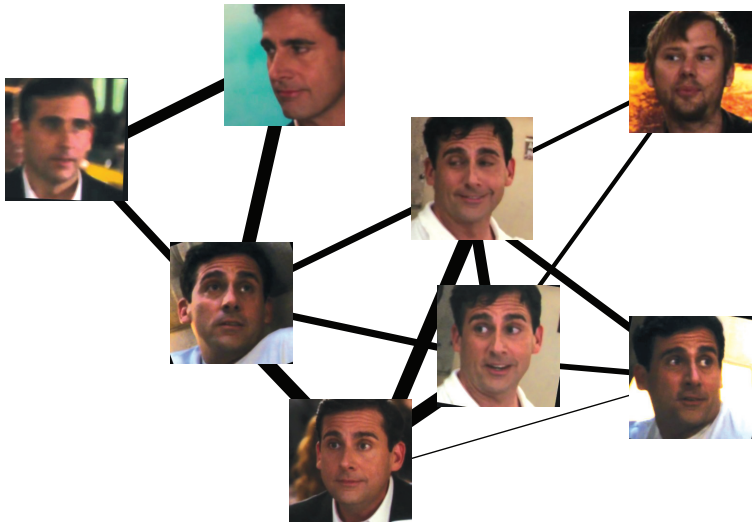


Figure 1.4: Subgraph from the movie Date Night for the actor Steve Carrel, where each node represents a face track. The edge weights are generated using our affinity metrics showing a strong relationship between tracks of the same person (Steve Carrel) and a weak relationship with the unknown actor (top right).

1.3 Affinity-based Video Face Recognition

In the last few years, there has been increased interest in face recognition in sitcoms [36,37]. These methods have focused on using additional context such as script text, audio, and clothing; however, the employed face identification methods have not been very accurate. Moreover, their end goal is person-identification, which in one sense is a more difficult task than standard face recognition because it requires continued recognition even when there is no visible face data. On the other hand, it is an easier task because the consistent contextual cues, i.e. hair style and clothing, compensate for inaccurate face recognition performance. Instead of focusing on fusing different contextual inputs, we focus on the difficult task of developing a highly precise method for unconstrained video face recognition.

Most video-based face recognition methods, like our method MSSRC, if they retain any temporal information, only consider the relationship between frames, thus ignoring any temporal

or visual affinity between individual face tracks in the same video. In any given sitcom or movie scene, many face tracks are produced for present actors. This result is sometimes due to poor tracking, shot changes, or pose variations. Due to these same reasons, face predictions may be noisy, where a face track may be classified correctly as one individual and a later track of the same person identified incorrectly. Within these scenes, there is a reasonable assumption that the people of interest do not change facial appearance much, therefore a strong relationship can be associated between face tracks of the same person as shown in Fig. 1.4.

Given the key insight that algorithms tend to misclassify face tracks visually similar to those correctly labeled, we propose an affinity-based method to share classification knowledge throughout an entire video. To do this we first build an affinity graph relating every face track to every other face track in a given video. Then we use random walks to propagate correct labels and demote wrongly labeled face tracks to improve prediction results over the entire movie. We construct the probabilistic affinity graph using the appearance and co-occurrence of predicted labels, to smooth the label predictions of closely related face tracks via random walk analysis. In the resolution of this dissertation, we show our method increases average precision and accuracy on our unconstrained Movie Trailer Face Dataset and The Big Bang Theory Dataset.

1.4 Contributions

The purpose of this dissertation is to analyze the problem of open-universe face identification in both video and still-imagery, a relevant, but little-studied problem in face recognition research. We propose two novel algorithms Linearly Approximated Sparse Representation-based Classification (LASRC) for still-image recognition and Mean Sequence Sparse Representation-based Classification (MSSRC) for video-based recognition. Both methods have strengths in their respective domains, but both perform exceptionally well in the task of rejecting unknown identities. Finally, we propose an Affinity-based Propagation scheme to correct noisy misclassifications

to better identify known actors and reject unknown, background actors. Our contributions are enumerated as follows:

1. Develop a novel algorithm, LASRC, for realtime, accurate, and web-scale, still-image face identification.

2. Introduce a new algorithm, MSSRC, that performs video face recognition using a joint optimization leveraging all of the available video data and employing the knowledge that face track frames belong to the same individual.

3. Propose an affinity-based propagation scheme for the accurate identification of known individuals and the rejection of unknowns in video via Random Walks.

4. Release two large, real-world face recognition datasets:

    i. Facebook Face Dataset: consisting of feature descriptors for a new Facebook Face Dataset from 800,000 faces images and a Facebook downloader tool for analysis of large face datasets.

    ii. Movie Trailer Face Dataset: consisting of 101 movie trailers from YouTube and 4,485 video face tracks.

Each of these points will be discussed in greater detail throughout the dissertation.

1.5 Organization of Dissertation

This dissertation is organized as follows: Chapter 2 describes the taxonomy of face recognition research and related work from both still-image and video-based face recognition. Next, Chapter 3 derives our efficient algorithm LASRC for open-universe face identification and presents a comparison of LASRC to many state-of-the-art of algorithms with large-scale, real-world datasets collected from PubFig, LFW, and Facebook. Subsequently, Chapter 4 introduces a complete

pipeline for video face recognition from tracking to recognition using our novel method MSSRC for real-world, open-universe video face identification, which we test on difficult movie trailers collected from YouTube. Next, Chapter 5 describes our method to smooth label predictions across a movie using the affinity between face tracks. Finally, Chapter 6 summarizes the contributions and findings of this dissertation followed by a discussion of future directions to explore.

CHAPTER 2: BACKGROUND

Over the course of several decades, face recognition research has amassed a large breadth of studies. Given our area of focus is open-universe face identification, we begin with an overview of existing work in open-universe identification. Next we discuss the relevant datasets and works first for still-image face recognition and then video face recognition. For a more general overview of face recognition research, we refer the readers to [38, 39] for still-image and [38, 40] for video-based face recognition.

2.1 Open-Universe 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. Thus, it is uncertain how the excellent results reported under closed-universe assumptions [16, 17, 19, 22, 24, 41] perform in open-universe scenarios. Likewise, verification tasks are popular and have progressed significantly [12, 13, 42], although verification algorithms have rarely been evaluated in identification tasks. Grother and Phillips [43] 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 [11] 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 90s, including explorations of open-universe face identification. Phillips *et al*. [6] first evaluate the controlled FERET [6] 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 [9] 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 [11] 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 [44] 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 [45] and [46] 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*. [34] 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 trade-off; however they test on small datasets so scalability to the large scale problems we are addressing is uncertain.

2.2 Still-Image Face Recognition

Still-image face recognition has a long history of research with several datasets exploring many parameters. In this section, we detail several datasets controlled and uncontrolled ("wild") as well as the most relevant algorithmic works to our method.

2.2.1 Datasets

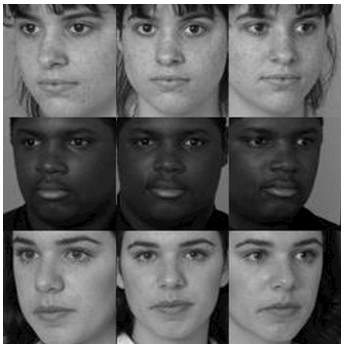
Traditionally, face recognition operates on faces captured in artificial environments where conditions are carefully controlled or labeled (AR [4], Yale [5], and FERET [6]). More recently, web-gathered LFW [12] and PubFig [13] datasets have gained popularity with face verification tasks with an increased focus on large-scale evaluations such as GBU [31] and MBE [11]. We summarize existing datasets in Tab. 2.1.

2.2.1.1 Controlled Datasets

Faces in highly controlled datasets such as Ext. Yale B [5] and the AR Face Database [4] are very popular choices for face recognition evaluation. The Extended Yale B [5] dataset contains 38 subjects under 64 lighting conditions (Fig. 2.1(a)). The AR Face Database [4] contain 50 male and 50 female subjects with images taken two weeks apart for each (Fig. 2.1(b)). The FERET dataset [6] (Fig. 2.1(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 [9], FRGC [10], and MBE [11] are less controlled and much larger and realistic, being pulled from law enforcement and visa sources.

(a) Ext. YaleB

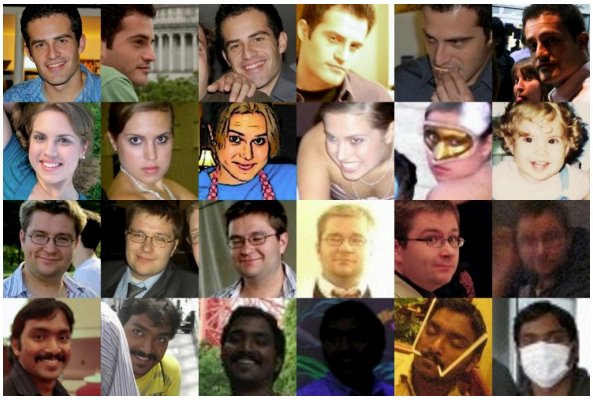
(b) AR



(c) FERET

(d) Labeled Faces in the Wild

(e) PubFig



(f) Facebook

Figure 2.1: Example faces highlighting the emergence of realism from controlled datasets (a-c) to web-gathered datasets (d-f). (a) Extended Yale B [5] concentrates on illumination, (b) AR [4] on disguises, and (c) FERET [47] on pose. (d) LFW [12] focuses on pair matching between famous faces while (f) PubFig [13] has gathered many celebrity photos. (f) Our challenging yet realistic Facebook dataset is naturally diverse in pose, illumination, occlusion, age, and even drawings. Publishing consent was obtained.

Table 2.1: 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. †Some photos are taken outdoors in natural lighting. *Raw images not available for privacy reasons, but feature descriptors are available.

Dataset Name	Public	Source	Controlled	Main Task	Faces/ ID	Known IDs	# Faces	Unknown IDs
DOS/Natural [11]	No	Visas	Yes	Open ID	1	520k	625k	50k
DOS/HCINT [11]	No	Visas	Yes	Verification	3	37.4k	121k	30k
LEO [11]	No	Mugshots	Yes	Open ID	1	1.6M	2.4M	200k
SANDIA [11]	No	Lab	Yes	Verification	50	263	13.9k	-
FERET [6]	Yes	Lab	Yes	Closed ID	12	1.2k	14k	-
ATT (ORL) [3]	Yes	Lab	Yes	Closed ID	10	40	400	-
Ext. Yale B [5]	Yes	Lab	Yes	Closed ID	576	28	16.1k	-
AR [4]	Yes	Lab	Yes	Closed ID	30	126	4k	-
GBU [31]	Yes	Lab	Semi†	Verification	15	437	6.5k	-
LFW [12]	Yes	Web	No	Verification	3	5.7k	13.2k	-
MultiPIE [30]	Yes	Lab	Yes	Closed ID	2k	337	750k	-
PubFig [13]	Yes	Web	No	Verification	300	200	58.8k	-
Facebook [14]	No	Web	No	Closed ID	25	15.8k	439k	-
Facebook [15]	No	Web	No	Closed ID	65	946	61.7k	-
Facebook [16]	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	85	6.1k	803k	110k

2.2.1.2 Verification Datasets

Two datasets designed for face verification have become popular: the Good, the Bad, and the Ugly (GBU) [31] and Labeled Faces in the Wild (LFW) [12]. 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 65,000 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 13,200 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.

2.2.1.3 Web-Gathered Datasets

As previously mentioned, 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) [12] dataset (Fig. 2.1(d)) and the 58.8k image Public Figures (PubFig) [13] dataset (Fig. 2.1(e)). Researchers have also used social network faces [14–16], but these datasets have not been released. The predominant use of LFW and PubFig is face verification [12, 13, 42], although small subsets have been used for closed-universe face identification [16, 24]. 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.* [48]. 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 setup mimics a web-scale scenario of finding specific celebrities while ignoring all others faces.

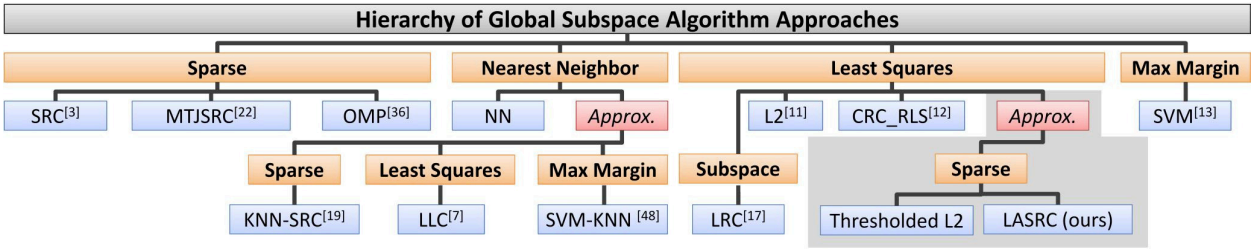


Figure 2.2: A hierarchy of face identification algorithms discussed in this paper, grouped by broad categories. Slow performing algorithms such as SRC or SVMs do not scale well, but can employ fast approximations to make an initial guess that can be refined. Highlighted in gray, we propose a novel linear regression approximation for SRC, named LASRC.

2.2.2 *Related Work*

Since the scope of face recognition research is vast, we cover some recent advances in face identification shown hierarchically in Fig. 2.2, 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 [13] or V1-inspired features [16] do not fit into the subset in Fig. 2.2 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 [24, 49, 50]. Wolf *et al*. [24] 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 [51], which is a highly optimized version of linear SVMs, and the dense data patch for speed [49]. Furthermore, limiting the training examples or tuning convergence parameters reduces classification rates too low to be competitive. Lin *et al*. [50] introduced an Averaged Stochastic Gradient Descent (ASGD) method to train huge SVMs rapidly, but it requires more than 30 minutes for our large datasets and yields accuracy well below LIBLINEAR. Thus, many current 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-based Classification (SRC): In the pioneering work on SRC, Wright *et al*. [35] 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 [52] and later accounted for small noise with a second-order cone program (SOCP) [53]. Interestingly, both methods are initialized by the ℓ^2 solution. Several faster algorithms have been developed: Gradient Projection for Sparse Representation (GPSR) [54], Homotopy [55], and Augmented Lagrange Multiplier (ALM) [56], 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) [57], 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 [58] 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 [59]. Patel *et al*. [60] 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 [48] or eye-based alignment adequately addresses the variations in pose.

Yang and Zhang [19] found that holistic features like PCA and LDA used in [35] cannot handle variations in illumination, expression, pose, and local deformations. Moreover, the occlusion matrix introduced in [35] 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 [29] used the Local Binary Pattern (LBP) [61] histogram descriptor, finding local features provided more robustness to misalignments than SRC on raw pixels. Likewise, Yuan and Yan [41] 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 [19]. 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*. [21] combined an explicit hashing function to reduce data dimensionality while preserving important structure information for ℓ^1-minimization via OMP. Differently, Nan and Jian [28] and Li *et al*. [27] 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 [62] or the Sphere Test 3 [63] 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 [26], 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*. [22] questioned whether face recognition is really a compressive sensing problem and demonstrated least-squares is comparable to SRC on controlled datasets. Zhang *et al*. [23] 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*. [18] 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*. [64] 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 ℓ^2-based methods struggle when presented with open-universe, real-world data from Labeled Faces in the Wild (LFW) [12], PubFig [13], and Facebook [14–16].

In summary, 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 Sec. 3.5.3. Although ℓ^1 methods are slow, they exhibit robustness in discovering the correct identity of test faces. Our 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 minor variations in pose and illumination 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 $\ell^2$and subsequent ℓ^1-minimization provides the speed and robustness necessary to deal with real-world data.

Table 2.2: A brief summary of a subset of popular and Internet-based video face recognition datasets, listing whether or not they are publicly available for download, the photographic source of the images (captured in a lab, sitcom (TV or Movie), 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. †Some photos are taken outdoors in natural lighting.

Dataset Name	Public	Source	Controlled	Main Task	Faces/ ID	Known IDs	# Faces	Unknown IDs
MBGC/FOCS [65]	Yes	Lab	Yes†	Closed ID	3	61	197	-
Honda/UCSD [66]	Yes	Lab	Yes	Closed ID	2	35	75	-
Buffy [67]	Yes	Sitcom	No	Open ID	80	8	639	156
YouTube Celebrities [68]	Yes	Web	No	Closed ID	41	47	1910	-
YouTube Faces [24]	Yes	Web	No	Verification	0	3425	1595	-
Big Bang Theory [69]	Yes	Sitcom	No	Open ID	304	11	3344	415
MTFD (Ours)	Yes	Web	No	Open ID	7	210	1552	2933

2.3 Video-Based Face Recognition

In this section, we further explore datasets and the most related work as they relate to video-based face recognition.

2.3.1 Datasets

Most controlled video datasets [70–72] have fallen out of use, with the exception of a few. We summarize existing datasets in Tab. 2.2 with a special focus on web-gathered datasets. With the existence of such a large video sharing website, gathering unconstrained videos from YouTube has become very popular and easy. By searching for famous people, the YouTube Celebrities [68] and YouTube Faces Datasets [24] were created. The YouTube Celebrities Dataset (Fig. 2.3(b)) consists of 1,910 video clips of 47 actors and politicians for face identification and due to its novelty, it has received much attention. The YouTube Face Dataset (Fig. 2.3(c)) on the other hand focuses on the face verification task with 3,425 videos of 1,595 different people with an average of 2.15

videos per person. Although, this dataset is large and from a sizable number of people, its low number of videos per person makes it difficult to adapt for the face identification task. Following the pioneering work of Everingham *et al*. [36], where the goal was to label all characters in the TV show Buffy, several authors have begun to attack the same problem. Recently, Cinbis *et al*. [67] released a new subset of the Buffy dataset (Fig. 2.3(a)) from episodes 9, 21, and 45 for a total of 639 manually annotated face tracks. Finally, Baüml *et al*. [69] released the Big Bang Theory Dataset (Fig. 2.3(d)) for identification within a sitcom. The Big Bang Theory dataset provides the largest Faces to ID ratio, however our new Movie Trailer Face Dataset (MTFD) provides the largest open-universe analysis by including the most unknowns and a larger number known IDs, 210 vs. 11 to be exact. Further, other studies have considered the unknowns as an actual class, which underperforms the alternative of outright rejecting the unknowns.

2.3.2 Related Work

For a complete survey of video-based face recognition refer to [40]; here we focus on an overview of the most related methods. Current video face recognition techniques fall into one of four categories: key-frame based, temporal model based, image-set matching based, and context based.

Key-frame based methods generally perform a prediction on the identity of each key-frame in a face track followed by a probabilistic fusion or majority voting to select the best match. Due to the large variations in the data, key-frame selection is claimed to be crucial in this paradigm [73]. Zhao *et al*.'s [74] work is most similar to us in that they use a database with still images collected from the Internet. They learn a model over this dictionary by learning key faces via clustering. These cluster centers are compared to test frames using a nearest-neighbor search followed by majority, probabilistic voting to make a final prediction. Chen *et al*. [75] present a dictionary based method most similar to ours, however they focus on dictionary learning done on a per face track basis, whereas we focus on the classification using a still-image gallery. Finally,

Baüml *et al*. [69] do not use key-frames, but similarly perform probabilistic voting over all frames in a track using a classifier trained via Maximum Likelihood Regression (MLR). We, on the other hand, use a classification scheme that enhances robustness by finding an agreement amongst the individual frames in a single optimization.

Temporal model based methods learn the temporal, facial dynamics of the face throughout a video. Several methods employ Hidden Markov Models (HMM) for this end [68]. Most related to us, Hadid *et al*. [76] use a still image training library by imposing motion information upon it to train an HMM and Zhou *et al*. [77] probabilistically generalize a still-image library to do video-to-video matching. Generally training these models is prohibitively expensive, especially when the dataset size is large.

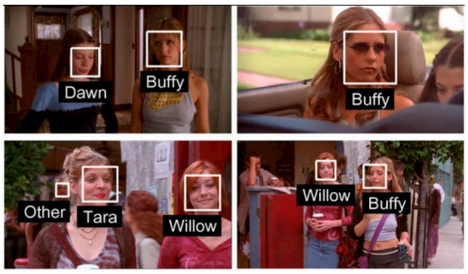
Image-set matching based methods allow the modeling of a face track as an image-set. Many methods, like [78, 79], perform a mutual subspace distance where each face track is modeled in their own subspace from which a distance is computed between each. They are effective with clean data, but these methods are very sensitive to the variations inherent in video face tracks. Lee and Kriegman [79] attempt to address this by learning a subspace for each pose within a face track. Other methods take a more statistical approach, like [67], which used Logistic Discriminant-based Metric Learning (LDML) to learn a relationship between images in face tracks, where the inter-class distances are maximized. LDML is very computationally expensive and focuses more on learning relationships within the data, whereas we directly relate the test track to the training data.

Context based methods have been very popular due to their application to movies and sitcoms. Several works [36, 37, 80] perform person identification, where they use all available information, *e.g*. clothing appearance and audio, to identify the cast rather than the facial information alone. Authors in [81] used a small user selected sample of characters in the given movie to compute a pixel-wise Euclidean distance to handle occlusion. While others, *e.g*. [82], use a manifold for known characters which successfully clusters input frames. These methods have focused on simple face recognition techniques, supplemented by context, but on the other hand we focus on

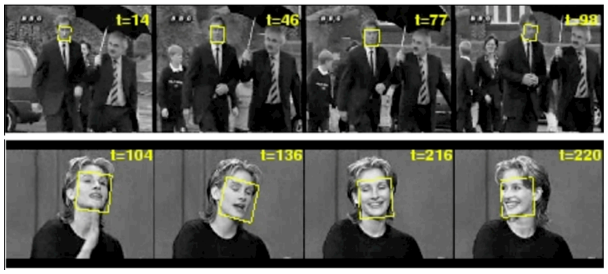
improving the precision of the face recognizer. Moreover, our framework can easily be extended to handle context in the second stage.

2.4 Affinity-based Face Recognition

Over the years, researchers have realized the benefit of using context whether it be the co-occurrence of images or the temporal distance between face tracks to further improve recognition performance. Several graph-based methods employ Markov models in an active-learning paradigm in which a few samples are selected to be labeled by the user, then used to label the rest of the data. Gallagher and Chen [83] create a Markov network where similarity edges are formed between faces in different photos and dissimilarity edges between the others, with an edge weight defined by appearance. This graph is then used in Loopy Belief Propagation to label all unlabeled test samples. Kapoor *et al*. [84] combines Gaussian Processes to enforce label smoothness with Markov Random Fields to encode the match and non-match structures, where matches are images of the same individual (faces within a track) and non-matches are faces in the same shot. More recently, Lin *et al*. [85] create a probabilistic, Markov framework using multiple contexts (faces, events, and location) to improve recognition. The strengths of these methods lie in that they are iterative methods that allow feedback from users and thus label the unlabeled data with few samples. However, our aim is to label many face tracks, therefore we develop a technique that smooths the initial predictions across all tracks in one optimization. Also, [84] is the only one that uses video and they do this by creating edges between frames of the same track, whereas our framework allows us to create a single node per track therefore reducing the size of the graph and thus computational complexity.

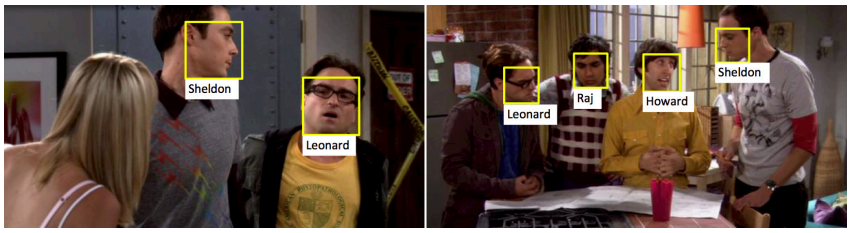


(a) Buffy

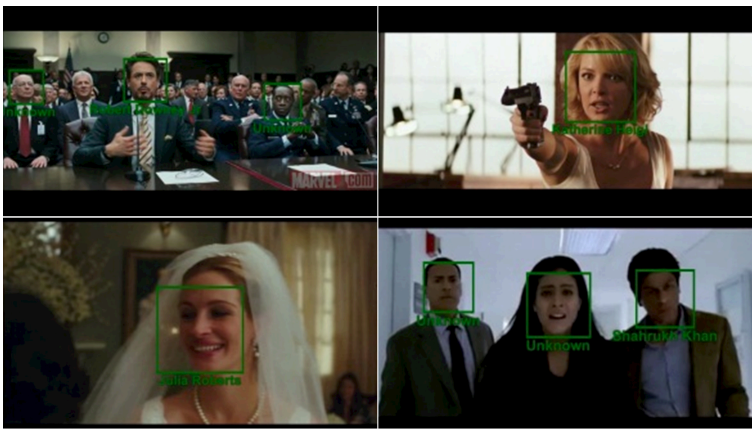


(b) YouTube Celebrities

(c) YouTube Faces



(d) Big Bang Theory



(e) Movie Trailers

Figure 2.3: Example faces from existing, realistic video face datasets. (a) Buffy is collected from several episodes of the TV show "Buffy the Vampire Slayer", (b) YouTube Celebrities is collected from videos of celebrities on YouTube, (c) YouTube Faces is also consists of celebrities from YouTube, but has many more clips and focuses on the problem of face verification, (d) Big Bang Theory consists of 6 episodes from the TV show, and (e) Movie Trailers, our new dataset, consists of 113 movie trailers.

CHAPTER 3: STILL-IMAGE, OPEN-UNIVERSE FACE IDENTIFICATION

Sparse Representation-based Classification is currently very popular due to its high accuracy, but its large computational complexity makes it ill-suited for large-scale face identification. Therefore, in this chapter we explore how we can increase the speed of SRC for application to the web-scale task of automatically tagging faces in photos. We show that by combining least-squares for approximation with the robustness of ℓ^1-methods to find the final solution, as shown in Fig. 3.1, we can obtain high performance without sacrificing too much speed. We further explore the difficult task of rejecting unknown identities, desirable in real-world applications like tagging photos on social networks, and evaluate several state-of-the-art algorithms with our new real-world datasets collected from Facebook.

3.1 Linearly Approximated SRC for Face Identification

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 $\boldsymbol{B}_j = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{n_j}]$ as column vectors, therefore matrix $\boldsymbol{B}$ is composed of all of the faces for all subjects $B = [\boldsymbol{B}_1, \ldots, \boldsymbol{B}_C] \in \mathbb{R}^{m \times n}$, 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 $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x}$, where $\boldsymbol{x}$ is a coefficient vector encoding the relationship of $\boldsymbol{y}$ to the columns of $\boldsymbol{B}$.

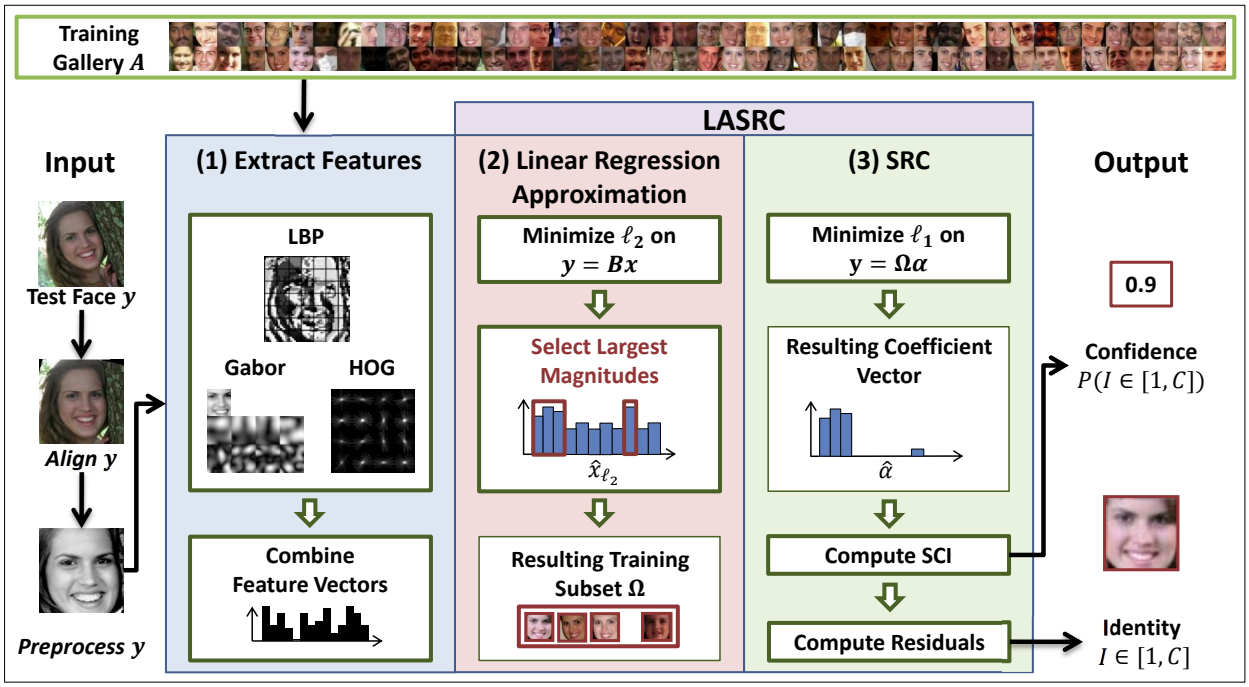


Figure 3.1: System pipeline depicting how LASRC classifies a new test face y given a set of training faces $\boldsymbol{B}$. After alignment and preprocessing, local features are extracted and concatenated, linear regression is performed to select a pool of representative training samples Ω, and SRC with ℓ^1-minimization is performed to calculate the most probable identity and confidence.

3.1.1 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:

$$\hat{\boldsymbol{x}}_{\ell_2} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{x}\|_2^2, \tag{3.1}$$

and is computed by the psuedoinverse as follows:

$$\hat{\boldsymbol{x}}_{\ell_2} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{y}. \tag{3.2}$$

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. If the system is under-determined, a least-norm formulation is used and has a similar pseudoinverse. Wright *et al*. [35] stated that $\hat{\boldsymbol{x}}_2$ is dense and therefore is not very informative. However, recent studies [22, 23] show that ℓ^2 works well for common datasets even though the measurements are noisy.

3.1.2 Sparse Representation-based Classification

Compressive sensing has been shown to outperform least-squares using only a subset of available data [35]. Given test image $\boldsymbol{y}$ and training set $\boldsymbol{B}$, we know that the images of the same class to which $\boldsymbol{y}$ should match is a small subset of $\boldsymbol{B}$. 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:

$$\hat{\boldsymbol{x}}_{\ell_1} = \min_{x,\epsilon} \|\boldsymbol{x}\|_1 + \|\epsilon\|_2 \;\; \text{s.t.} \;\; \boldsymbol{y} = \boldsymbol{B}\boldsymbol{x} + \epsilon, \tag{3.3}$$

where the ℓ^1-norm enforces a sparse solution by minimizing the absolute sum of the coefficients. The sparsity constraint results in the largest non-zero values being concentrated on the matching training images corresponding to the correct class.

Wright *et al*. [35] identifies the test image $\boldsymbol{y}$ by determining the class of training samples that best reconstructs the face from the recovered coefficients:

$$I(\boldsymbol{y}) = \min_j r_j(\boldsymbol{y}) = \min \|\boldsymbol{y} - \boldsymbol{B}_j \boldsymbol{x}_j\|_2, \tag{3.4}$$

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 [35].

SCI is a measure of how distributed the residuals are across classes:

$$SCI = \frac{C \cdot \max_j \|\boldsymbol{x}_j\|_1 / \|\hat{\boldsymbol{x}}_{\ell_1}\|_1 - 1}{C - 1} \in [0, 1]. \tag{3.5}$$

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.* [35] 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.

3.1.3 Approximating SRC

A large drawback to SRC is the computational complexity required by ℓ^1-minimization, which requires several seconds per image [17, 35] 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 [55], takes at least 5 seconds 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 (3.3) specified as a sequence of vector operations is as follows:

$$v(\boldsymbol{x}) = \|\boldsymbol{y} - \sum_{i=1}^{n} \boldsymbol{a}_i x_i\|_2 + \lambda \sum_{i=1}^{n} |x_i|, \tag{3.6}$$

in which we denote $\boldsymbol{b}_i \in \mathbb{R}^m$ as the i-th column of $\boldsymbol{B}$, x_i as the i-th 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 (3.6), then $\|\boldsymbol{b}_i x_i\|_2 = 0$, $|x_i| = 0$, and $\boldsymbol{b}_i$ do not contribute to $v(\boldsymbol{x})$. Based on this observation, we rewrite the objective function as:

$$v(\boldsymbol{\alpha}) = \|\boldsymbol{y} - \sum_{i=1}^{K} \boldsymbol{\omega}_i \alpha_i\|_2 + \lambda \sum_{i=1}^{K} |\alpha_i|, \tag{3.7}$$

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 above 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:

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{y} - \Omega\boldsymbol{\alpha}\|_2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{3.8}$$

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 [54] 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.

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 Sec. 3.1.1, ℓ^2-minimization is very fast, convenient, and has proven to be adequate for standard face recognition datasets. Furthermore, it is evident 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, as we will show in Section 3.4. 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 Sec. 3.4.2.1, 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.

3.1.4 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. 3.1, 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 (3.2) using the pre-calculated pseudo-inverse $(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T$. Next, we select the top K training samples from $\boldsymbol{B}$ 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 (3.8). 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 (4.11) for the probability that the given test image identity exists in the training database. In the hierarchy shown in Fig. 2.2, our method is sparse using a least-squares approximation.

Algorithm 1 Linearly Approximated SRC (LASRC)

1. **Input:** Training gallery $\boldsymbol{B} \in \mathbb{R}^{m\times n}$, test face $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$, and sparsity controlling parameter λ.
2. Normalize the columns of $\boldsymbol{B}$ to have unit ℓ^2-norm
3. Compute linear regression using the pre-calculated pseudoinverse $\hat{\boldsymbol{x}}_{\ell_2} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{y}$
4. Select K samples from $\boldsymbol{B}$ 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

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

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

3.2 Facebook Dataset

As discussed in Chatper 2, traditionally, face recognition operates on faces captured in artificial environments where conditions are carefully controlled or labeled (AR [4], Yale [5], and FERET [47]). More recently, web-gathered LFW [12] and PubFig [13] datasets have gained popularly with face verification tasks. 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 [14–16], 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. Focusing on the scenario of automatically tagging friends in open-universe social networks, we created a new 800,000 face dataset (Fig. 2.1(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://face.enriquegortiz.com`.

3.2.1 Dataset Construction

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

Face Collection: Similar to Stone *et al*. [14] and Becker and Ortiz [15], 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 [86, 87] was used to detect 48.3 million frontal faces with a rotation range of approximately $\pm 35^{\circ}$ at a rate of 20 Hz. From 3,000 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.

Table 3.1: 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	200	5	35.5k	11.6k	11.7k

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 Tab. 3.1, 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.

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.

3.2.2 Evaluation Criterion

For photo-tagging algorithms in social networks, we evaluate using precision and recall curves, recall at 95% precision, and computational cost. Because accuracy is not particularly informative in an open-universe scenario, where there are distractors, 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 [88]. Intuitively, the PR curves tell us at a given threshold how much data of interest do we label and how well we do on that data. Often Average Precision (AP) or F-scores are used to summarize PR curves, but we feel that recall at 95% precision better reflects real-world performance as this corresponds to the percentage of detected faces that can be labeled with only one mistake in 20 predictions. Since fast classification and training times are necessary in such dynamic, real-world situations, it is important to report train and test times.

3.2.3 Dataset Bias

Torralba and Efros [89] 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, our dataset has a large negative set in the form of a realistic set of distractors from non-friend background faces.

3.3 Feature Representations

Using local features to augment classification is a widely used technique [24, 61, 90]. However, due to underlying assumptions of pixel-wise linearity, least-squares and sparse methods have

primarily focused on raw pixels [17, 22, 23, 35]. On the other hand, Chan and Kittler [29] and Yang and Zhang [19] 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 in object recognition [41].

3.3.1 Feature Selection and Extraction

Due to real-world pose variations, even after alignment, we use three popular features: Gabor wavelets [90], Local Binary Patterns (LBP) [61], and Histogram of Oriented Gradients (HOG) [91]. Inclusion of more features aids recognition slightly, but with loss in time.

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 25x30 pixels. A null Gabor filter includes the raw pixel image (also 25x30) in the descriptor. In agreement with [25], we found looser crops work better for histogram-based features. The standard $\text{LBP}^{U2}_{8,2}$ (uniform circular patterns of 8 pixels and a radius of 2) and HOG descriptors are extracted from 72x80 loosely cropped images. Each features has a histogram size of 59 and 32 over 9x10 and 8x8 pixel patches, respectively. All descriptors were scaled to unit norm, dimensionality reduced with PCA to 512 dimensions each, and zero-meaned.

3.3.2 Performance

For reporting results, we use both controlled datasets (Sec. 2.2.1.1) and the Facebook datasets (Sec. 3.2). Times are from a 2.3 GHz machine (single-threaded).

3.3.2.1 Controlled Datasets

To better understand feature performance, we present results on controlled datasets (Sec. 2.2.1.1), including both the originally reported accuracies and our results when running the same

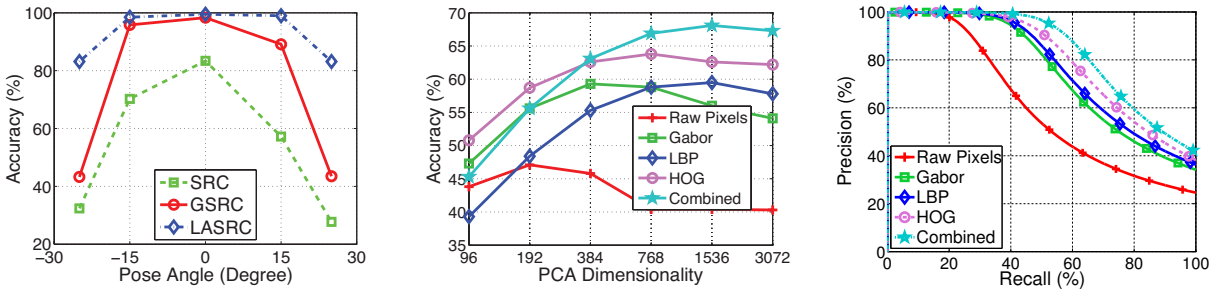
algorithms on a 1995 length vector concatenated from Gabor, LBP, and HOG. For Ext. Yale B, we randomly selected 32 images per subject for training, leaving 32 for testing. This random selection is repeated 10 times. For the AR Face Database we selected seven images from Session 1 for training and seven images from Session 2 two weeks later for testing. Using standard experimental protocols and the same database setups as [19–21, 27, 35, 41], our results are directly comparable to previously reported accuracies. Tab. 3.2 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 [6] with pose variations of $\theta = \{-25°, -15°, 0°, 15°, 25°\}$ were used in the same manner as [19]. Fig. 3.2(a) uses the FERET pose dataset (Sec. 2.2.1.1) to compare SRC [35] with raw pixels, GSRC [19] with Gabor features, and LASRC with local features. A single feature aids recognition by 20%, but multiple features with LASRC boosts accuracy up to 50% compared to raw pixels.

3.3.2.2 Facebook Dataset

Repeating similar experiments with Gabor, LBP, and HOG features on our large-scale, real-world Facebook datasets, we investigate in Fig. 3.2(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 400-800 dimensions. Finally, a representation of multiple features combined achieves peak accuracy of 67% at 1536 dimensions (512 from each feature), 20% over raw pixels. Similar to the closed-universe accuracy in Fig. 3.2, we see a large increase in open-universe performance with more features.

Table 3.2: Accuracy on controlled datasets as originally published vs. performance using our three feature representation (Gabor, HOG, LBP). Most algorithms achieve >99.5% with features. [a]Reported from [35]. [b]Accuracy interpolated from graph. [c]Not using a raw pixel representation.

	Extended Yale B		**AR Face Dataset**	
Algorithm	Reported Acc (%)	Feature Acc (%)	Reported Acc (%)	Feature Acc (%)
NN [a]	90.7	92.1±0.7	89.7	98.7
SVM[a] [24]	97.7	**99.8±0.1**	95.7	99.6
SVM-KNN [92]	-	99.7±0.1	-	98.1
SRC [35]	98.1	99.7±0.1	94.7	99.9
MTJSRC[b,c] [41]	**99.5**	99.7±0.1	-	99.7
LLC [18]	-	99.7±0.1	-	99.9
OMP [21]	96.4	99.6±0.1	**96.9**	**100.0**
KNN-SRC [28]	88.0	99.7±0.1	-	99.9
LRC [26]	-	98.7±0.4	-	98.9
L2 [22]	98.9	**99.8±0.1**	95.9	99.9
CRC_RLS [23]	97.9	**99.8±0.1**	93.7	**100.0**
LASRC (Ours)	-	99.7±0.1	-	99.9



(a) FERET Pose Accuracy (b) Facebook Accuracy (c) Facebook PR Curves

Figure 3.2: Performance of LASRC with Features. (a) Performance on FERET pose dataset (b) Accuracy on Facebook dataset with various features and varying dimensionality. (c) Precision and recall curves on Facebook for feature representations with $m = 1536$ dimensionality.

3.3.3 *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 [22, 35]. Since occlusions are clearly evident in real-world faces, we resized Facebook images to 15x13 and used a 195x195 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 [22, 35] 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 [19], 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. 3.2(b)) while occlusion dictionaries only help by 0.5%, we choose to focus on multi-feature representations.

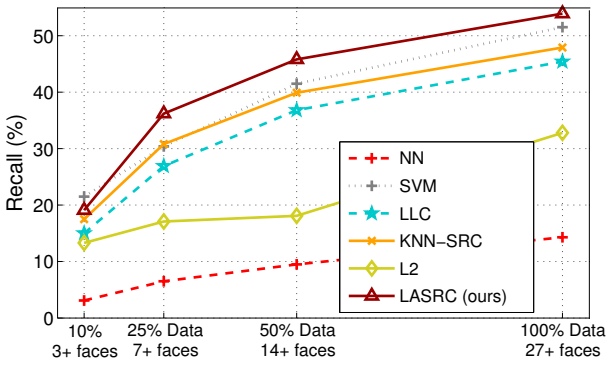


Figure 3.3: Effect of recall at 95% precision by varying the size of the dataset (mean number of minimum training faces for all Facebook datasets) across multiple algorithms.

3.3.4 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. 3.3. shows the performance as 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.

3.4 Sparsity and Locality Analysis

Lately there has been controversy between the relative effectiveness of least-squares [22, 23, 26, 64] vs. sparse [17, 35, 41] solutions. Furthermore, some works advocate the use of locality [18, 18, 28] 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.

3.4.1 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, [22, 23] 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?

3.4.1.1 Algorithms for ℓ^1-minimization

To answer the first question, a variety of ℓ^1-minimization techniques could be used [93]. Tab. 3.3 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 [54] as a good compromise.

3.4.1.2 Least-Squares Performance

On controlled datasets, [22,23,26] 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 (Tab. 3.3). Existing works [22, 23, 26] fail to address this issue, so we attempted to give least-squares algorithms a competitive edge by balancing the datasets. As shown in Tab. 3.3, 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.

3.4.1.3 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 [22,23] with 7% less accuracy and 20% lower recall than LASRC. Are LASRC's performance benefits coming from 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. 3.4 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 [22, 23]. 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. 3.4 suggest that least-squares [22, 23] 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.

In short, our results suggest that least-squares [22, 23] with local features are not ideal for naturally unbalanced, open-universe data such as Facebook. In fact, even very simple sparse methods like Thresholded L2 are superior. Sophisticated ℓ^1-minimization methods of imposing sparsity can further increase recall to outperform least-squares by 12-32% (Tab. 3.3).

3.4.1.4 LASRC vs. Least-Squares Speed

A puzzling result from Tab. 3.3 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 (4.10) 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.

3.4.2 Locality

Recognizing the value of sparsity, but unable to accept the slow performance of even the fastest ℓ^1-solvers [93], Nan and Jian [28] and Li *et al.* [27] both proposed locality approximations to SRC. KNN-SRC [28], selects a small subset of nearby training samples for ℓ^1-minimization to greatly speed up SRC. LLC [18] replaces the ℓ^1-minimization step with a weighted least-squares emphasizing locality. Similarly to KNN-SRC, SVM-KNN [92] trains a local SVM to classify each test sample. Refer to Fig. 2.2 for a hierarchy of algorithms. Screening rules of [62, 63] are based on correlation of the test sample with training samples, which has an equivalence to Euclidean distance when samples are normalized and thus performs within 0.1% of KNN-SRC.

Table 3.3: 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. [a]Confidence calculated from residuals instead of SCI.

Algorithm	Recall (%)	Accuracy (%)	Time (ms/face)
L2[a] [22]	22.4	49.3	55.3
L2 (balanced, max 100)[a] [22]	34.5	59.2	52.7
Thresholded L2	41.9	63.3	21.2
LLC[a] [18]	46.1	61.5	38.1
KNN-SRC[a] [28]	48.5	63.3	31.6
LRC[a] [26]	28.4	57.2	43.4
LASRC (Homotopy[a] [55])	50.5	65.1	61.1
LASRC (l1magic [94])	44.6	63.3	**29.3**
LASRC (L1_LS [95])	53.4	**66.6**	79.1
LASRC (GPSR [54])	**54.5**	66.5	31.7
LASRC (ALM [56])	54.4	66.5	35.2

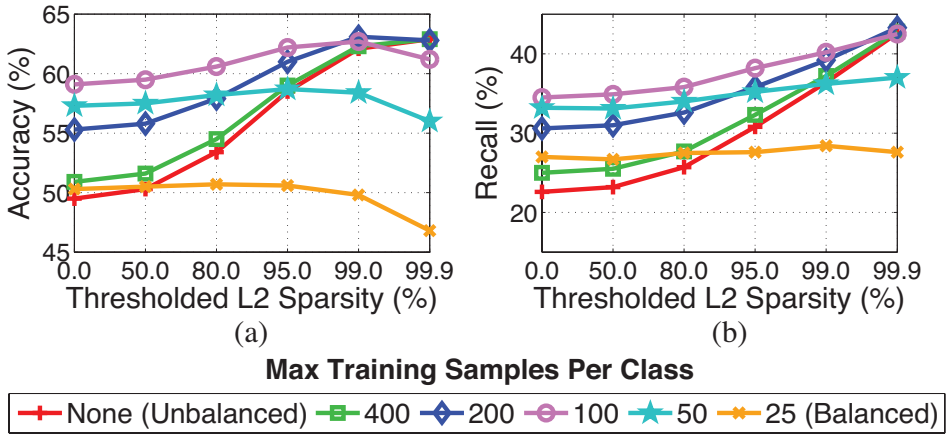


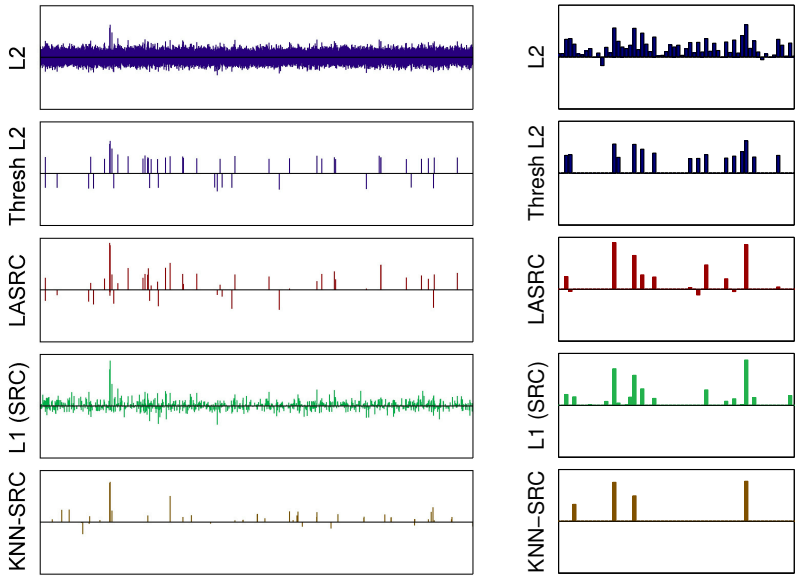
Figure 3.4: Thresholded L2 performance on Facebook as sparsity and balancedness is varied. (a) Accuracy increases with sparsity for unbalanced datasets (b) Sparsity increases recall at 95% precision for all but the completely balanced case.

The goal of approximating SRC is to select a small subset of training samples for ℓ^1-minimization so that classification time is greatly reduced while maintaining performance similar to SRC. KNN-SRC [27,28] 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. 3.5. All methods exhibit a peak at the correct class, so Fig. 3.5(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.

3.4.2.1 KNN vs. Linear Regression Approximation

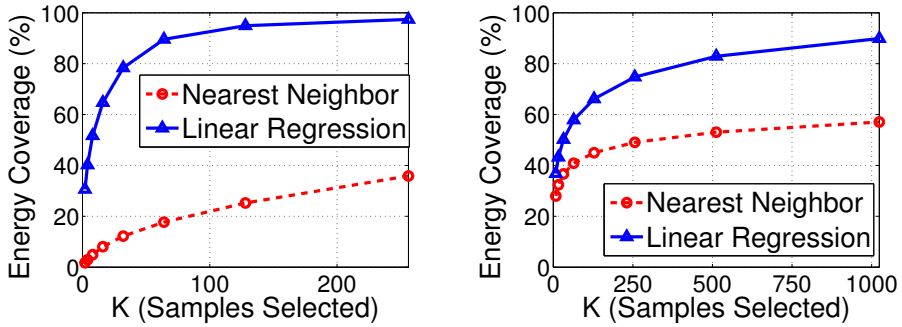
For a quantitative evaluation of the best metric of locality to approximate ℓ^1-minimization, we created dictionaries of randomly generated synthetic samples with the same parameters as

Yang *et al*. [93]. For 10,000 test samples (randomly generated from the dictionary with noise), we calculated the energy or overlap of samples selected by nearest neghbor and linear regression with the full sparse solution found by ℓ^1-minimization as we varied K. Fig. 3.6(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. 3.6(b)).



(a) All Samples (b) Correct Class Samples

Figure 3.5: Recovered coefficients from a Facebook test face for (a) all training samples and (b) zoomed in only on the training samples from the correct class (corresponding to the peak in (a)).



(a) Synthetic (b) Facebook

Figure 3.6: 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.

3.4.2.2 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 $\boldsymbol{B}^{+}\boldsymbol{y}$ of the test sample $\boldsymbol{y}$ with the pre-calculated pseudoinverse $\boldsymbol{B}^{+}$. Performing KNN naively is slow, but we optimized it by omitting the square root, expanding the term $\|(\boldsymbol{B}_i - \boldsymbol{y})\|^2$ into $\|\boldsymbol{B}_i\|^2 + \|\boldsymbol{y}\|^2 - 2\|\boldsymbol{B}_i^T\boldsymbol{y}\|^2$, vectorizing the n dot products $\boldsymbol{B}_i^T y$ into a single matrix multiplication $\boldsymbol{B}^T\boldsymbol{y}$, and pre-calculating $\|\boldsymbol{B}_i\|^2$. For further speedups, p test samples denoted as $\boldsymbol{Y} = [\boldsymbol{y}_1, ..., \boldsymbol{y}_p]$ can be batch multiplied as $\boldsymbol{B}^{+}\boldsymbol{Y}$ or $2\|\boldsymbol{B}^T\boldsymbol{Y}\|$ 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 $p = 16$, which yielded a 4-5X speedup for both algorithms as seen in Fig. 3.7(a).

3.4.2.3 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. 3.7(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. 3.7(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.

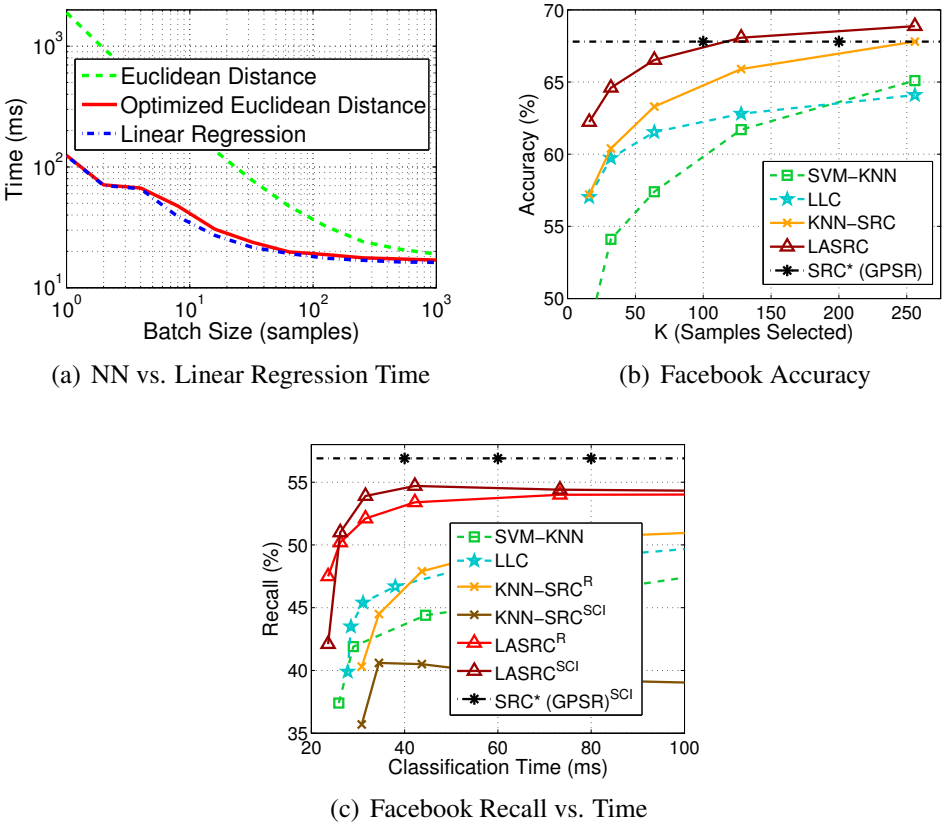
3.5 Comparison to State-of-the-Art

To evaluate the holistic performance of LASRC against current state-of-the-art algorithms on a large scale, we used realistic PubFig+LFW (Sec. 2.2.1.3) and Facebook (Sec. 3.2) 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. 2.2 for a hierarchy of tested algorithms.

3.5.1 Non-realtime Algorithms

Four algorithms from Tab. 3.2 suffer from slow training or classification times: SVMs, SRC, OMP, and MTJSRC. We omit algorithms such as GSRC [19] because they cannot use multiple features. For the baseline SRC algorithm, we test with two ℓ^1-solvers: Homotopy [55] and GPSR [54]. We tuned Homotopy for speed with a lower tolerance $tol = 10^{-3}$. We optimized GPSR for $B = 16$ batched operation (Sect. 3.4.2.2) 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 [51] algorithm optimized with dense data support for faster training [49] and a slack value of $c = 1$. Wolf *et al*. [24] 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 [41], 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 $p = 16$ (same as LASRC, KNN-SRC, and LLC).



(a) NN vs. Linear Regression Time

(b) Facebook Accuracy

(c) Facebook Recall vs. Time

Figure 3.7: 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).

3.5.2 Realtime Algorithms

The remaining eight algorithms from Tab. 3.2 are more suited to realtime operation: NN, SVM-KNN [92], LLC [18], KNN-SRC [28], LRC [26], L2 [22], CRC_RLS [23], and LASRC (Ours). Except for SVM-KNN, all realtime algorithms classify multiple test samples at once with a batch parameter of $B = 16$ (Sec. 3.4.2.2). SVM-KNN uses the LibSVM library [96] 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 [54] ℓ^1-minimization algorithm, although we use the minimum residual as confidence for KNN-SRC and SCI to reject distractors for LASRC.

3.5.3 PubFig+LFW and Facebook Performance

Using the real-world datasets from Sec. 2.2.1.3 and 3.2, we compare LASRC performance to other algorithms in both closed-universe and open-universe scenarios.

3.5.3.1 Closed-Universe Accuracy

As reported in Tab. 3.2, 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. Tab. 3.4 shows mean accuracy with standard deviations 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.

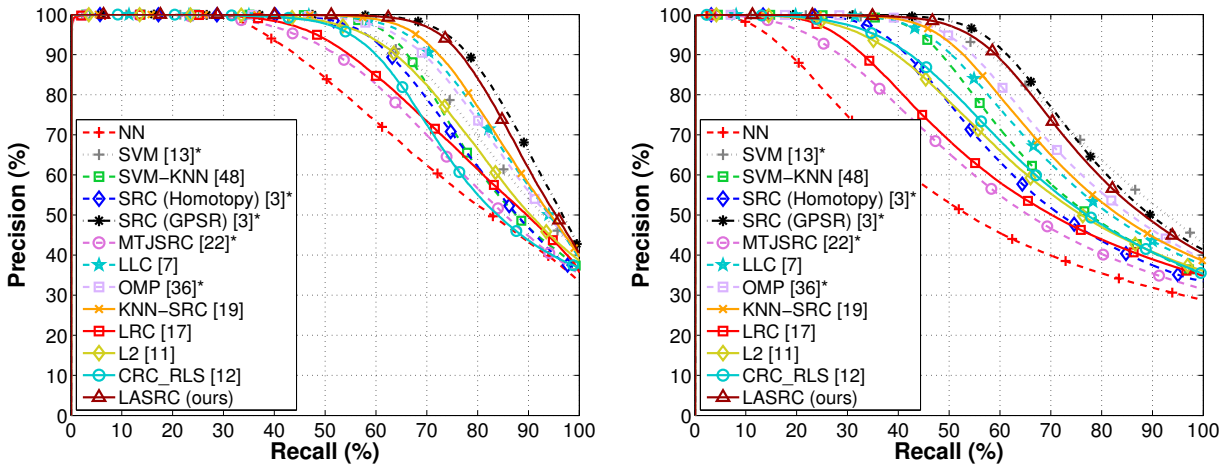
Table 3.4: PubFig+LFW (200 classes). Recall at 95% precision (open-universe), Accuracy (closed-universe), and classification time per test face (two significant figures only) for PubFig+LFW and three sizes of Facebook datasets. Red highlighted entries indicate non-realtime times. ‡ Tuned for maximum precision and recall without downsampling. †Tuned for speed with $\lambda = 0.01, tol = 10^{-3}$. *Tuned for maximum recall with $\lambda = 0.05$.

	Algorithm	Recall (%)	Accuracy (%)	Time (ms)
Non-Realtime	SVM (Liblinear [51])‡ [24]	58.5	80.2	**1**
	SRC (Homotopy [55])† [35]	72.2	72.2	1800
	SRC (GPSR [54])* [35]	**73.9**	**81.8**	4300
	OMP [57]	63.9	79.3	1500
	MTJSRC [41]	44.3	70.1	1300
Realtime	NN	38.2	65.8	**16**
	SVM-KNN [92]	62.5	73.2	31
	LLC [18]	66.0	77.8	22
	KNN-SRC [28]	67.9	78.8	35
	LRC [26]	48.3	70.9	30
	L2 [22]	58.0	76.8	21
	CRC-RLS [23]	54.9	73.5	23
	LASRC (Ours)	**72.6**	**81.3**	27

3.5.3.2 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 curves in Fig. 3.8 and recall at 95% precision in Tab. 3.4 for PubFig+LFW and Facebook datasets. 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 Fig. 3.8 where it is the only realtime method to achieve a PR curve 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.



(a) PubFig+LFW (b) Facebook

Figure 3.8: Precision and recall curves for (a) PubFig+LFW and (b) Facebook. Of all the realtime algorithms, only LASRC achieves comparable performance to non-realtime methods (denoted with *) such as SRC and SVMs.

3.5.3.3 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 minutes. 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. 3.9 depicts the timeline for realtime methods.

Table 3.5: 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 PubFig+LFW and three sizes of Facebook datasets. Red highlighted entries indicate non-realtime times. ‡ Tuned for maximum precision and recall without downsampling. †Tuned for speed with $\lambda = 0.01, tol = 10^{-3}$. *Tuned for maximum recall with $\lambda = 0.05$.

		Facebook (256 classes)			**Facebook (512 classes)**			**Facebook (1026 classes)**			**All**
	Algorithm	**Recall (%)**	**Acc. (%)**	**Time (ms)**	**Recall (%)**	**Acc. (%)**	**Time (ms)**	**Recall (%)**	**Acc. (%)**	**Time (ms)**	**Max Train Time (min)**
Non-Realtime	SVM (Liblinear [51])‡ [24]	54.1	**73.1**	**1**	50.9	**69.5**	**3**	50.0	**67.4**	**6**	124.7
	SRC (Homotopy [55])† [35]	41.4	59.7	1300	36.9	54.3	2600	**34.8**	**50.8**	5400	**0.0**
	SRC (GPSR [54])* [35]	**59.2**	71.1	2400	**56.4**	67.3	5400	**34.8**	**50.8**	5400	**0.0**
	OMP [57]	51.3	68.3	890	49.5	63.1	1600	**55.2**	65.0	11000	**0.0**
	MTJSRC [41]	30.5	58.9	840	23.9	51.2	1800	48.7	59.8	2800	**0.0**
Realtime	NN	17.9	51.8	**11**	14.1	46.4	**21**	12.7	43.4	**44**	**0.0**
	SVM-KNN [92]	50.5	62.6	31	45.1	56.8	42	42.0	52.6	61	**0.0**
	LLC [18]	49.4	66.1	24	45.1	80.2	24	43.7	57.6	56	**0.0**
	KNN-SRC [28]	51.7	67.8	55	47.8	62.8	67	46.0	59.3	90	**0.0**
	LRC [26]	31.3	60.8	19	27.9	56.6	38	25.9	54.3	72	0.2
	L2 [22]	41.5	65.3	23	34.0	58.8	44	27.9	53.	91	1.2
	CRC-RLS [23]	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

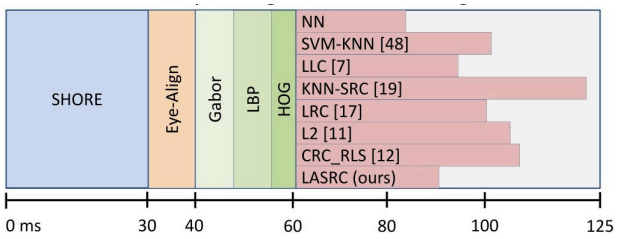


Figure 3.9: Timeline of all steps in the entire face recognition system. All times reported with a single core of a 2.27 GHz machine.

3.6 Summary

In this chapter, 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 via 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. 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 real-time algorithms in speed, accuracy, and recall.

CHAPTER 4: VIDEO-BASED, OPEN-UNIVERSE FACE IDENTIFICATION

Video face identification is an obvious, yet difficult, extension of still-image face recognition techniques. In this chapter, we present our complete system including face tracking, feature extraction, and identification for video face recognition. Most interestingly, we extend the Sparse Representation-based Classification (SRC) framework to the recognition of video face tracks and show that this seemingly difficult task reduces to a simple formulation. We show this result mathematically followed by experiments comparing several methods using existing datasets and our new Movie Trailer Face Dataset (MTFD) collected from YouTube.

4.1 Video Face Identification Pipeline

In this section, we describe our end-to-end video face recognition system as depicted in Fig. 4.1. First, we detail our algorithm for face tracking based on face detections from video. Next, we chronicle the features we use to describe the faces and handle variations in pose, lighting, and occlusion. Finally, we derive our optimization for video face recognition that classifies a video face track based on a dictionary of still images.

4.1.1 Face Tracking

Our method performs the difficult task of face tracking based on face detections extracted using the high-performance SHORE face detection system [87] and generates a face track based on two metrics. To associate a new detection to an existing track, our first metric determines the ratio of the maximum sized bounding box encompassing both face detections to the size of the

larger bounding box of the two detections. The formulation is as follows:

$$d_{spatial} = \frac{w * h}{\max(h_1 * w_1, h_2 * w_2)}, \tag{4.1}$$

where (x_1, y_1, w_1, h_1) and (x_2, y_2, w_2, h_2) are the (x, y) location and the width and height of the previous and current frames respectively. The overall width w and height h are computed as $w = \max(x_1 + w_1, x_2 + w_2) - \min(x_1, x_2)$ and $h = max(y_1 + h_1, y_2 + h_2) - \min(y_1, y_2)$. Intuitively, this metric encodes the dimensional similarity of the current and previous bounding boxes, intrinsically considering the spatial information.

The second tracking metric takes into account the appearance information via a local color histogram of the face. We compute the distance as a ratio of the histogram intersection of the RGB histograms with 30 bins per channel of the last face of a track and the current detection to the total summation of the histogram bins:

$$d_{appearance} = \sum_{i=1}^{n} \min(a_i, b_i) / \sum_{i=1}^{n} a_i + b_i, \tag{4.2}$$

where a and b are the histograms of the current and previous face. We compare each new face detection to existing tracks; if the location and appearance metric is similar, the face is added to the track, otherwise a new track is created. Finally, we use a global histogram for the entire frame, encoding scene information, to detect scene boundaries, in other words the end of a scene, and impose a lifespan of 20 frames of no detection to output existing tracks.

4.1.2 Feature Extraction

The features here are the same as those described in Section 3.3.1, however we reduce dimensionality using PCA to 1536 dimensions for each feature, as we found this resulted in better performance in the case of video face recognition.

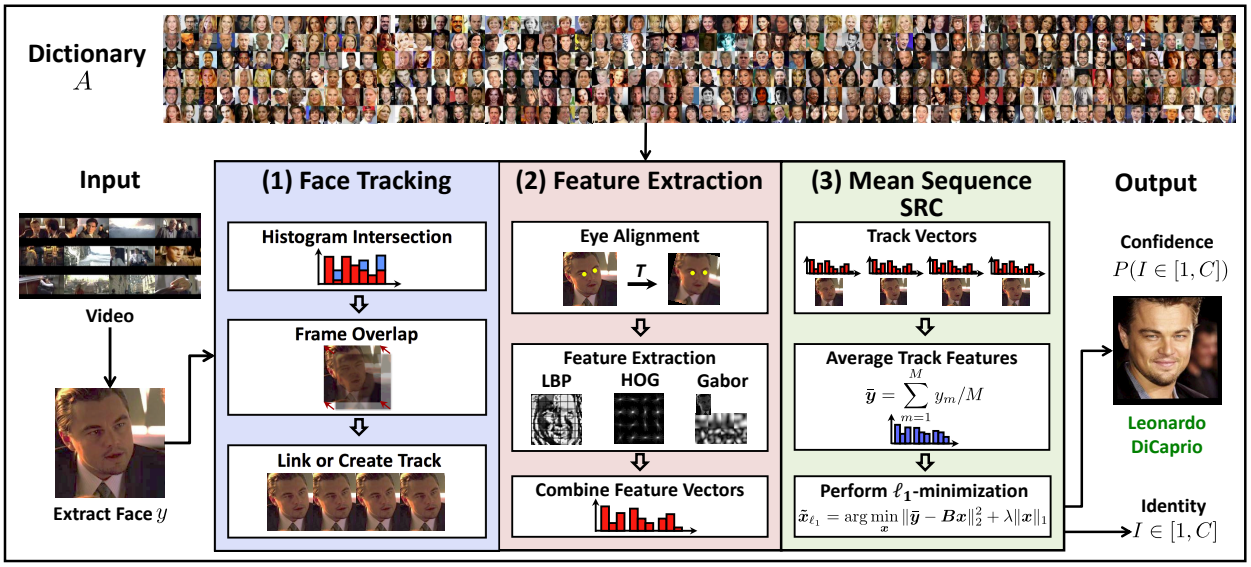


Figure 4.1: Video Face Identification Pipeline. With a video as input, we perform face detection and track a face throughout the video clip. Then we extract, PCA, and concatenate three features, Gabor, LBP, and HOG. Finally, we perform face recognition using our novel algorithm MSSRC with an input face track and dictionary of still images.

4.1.3 Mean Sequence Sparse Representation-based Classification (MSSRC)

Given a test image $\boldsymbol{y}$ and training set $\boldsymbol{B}$, from Chapter 3, we know that the images of the same class to which $\boldsymbol{y}$ should match is a small subset of $\boldsymbol{B}$ and their relationship is modeled by $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x}$, where $\boldsymbol{x}$ is the coefficient vector relating them. 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 $\boldsymbol{x}$ results in the following formulation:

$$\hat{\boldsymbol{x}}_{\ell_1} = \arg\min_x \|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \tag{4.3}$$

where the ℓ^1-norm enforces a sparse solution by minimizing the absolute sum of the coefficients and λ specifies how much weight is given to this norm.

The leading principle of our method is that all of the images $\boldsymbol{y}$ from the face track $Y =$

$[\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_M]$ belong to the same person. Because all images in a face track belong to the same person, one would expect a high degree of correlation amongst the sparse coefficient vectors $\boldsymbol{x}_j \forall j \in [1 \ldots M]$, where M is the length of the track. Therefore, we can look for an agreement on a single coefficient vector $\boldsymbol{x}$ determining the linear combination of training images $\boldsymbol{B}$ that make up the unidentified person. In fact, with sufficient similarity between the faces in a track, one might expect nearly the same coefficient vector to be recovered for each frame. This provides the intuition for our approach: we enforce a single coefficient vector for all frames. Mathematically, this means the sum squared residual error over the fames should be minimized. We enforce this constraint on the ℓ^1 solution of Eqn. 4.3 as follows:

$$\tilde{\boldsymbol{x}}_{\ell_1} = \arg\min_{\boldsymbol{x}} \sum_{m=1}^{M} \|\boldsymbol{y}_m - \boldsymbol{B}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \tag{4.4}$$

where we minimize the ℓ^2 error over the entire image sequence, while assuming the coefficient vector $\boldsymbol{x}$ is sparse and the same over all of the images.

Focusing on the first part of the equation, more specifically the ℓ^2 portion, we can rearrange it as follows:

$$\begin{aligned} \sum_{m=1}^{M} \|\boldsymbol{y}_m - \boldsymbol{B}\boldsymbol{x}\|_2^2 &= \sum_{m=1}^{M} \|\boldsymbol{y}_m - \bar{\boldsymbol{y}} + \bar{\boldsymbol{y}} - \boldsymbol{B}\boldsymbol{x}\|_2^2 \\ &= \sum_{m=1}^{M} (\|\boldsymbol{y}_m - \bar{\boldsymbol{y}}\|_2^2 + 2(\boldsymbol{y}_m - \bar{\boldsymbol{y}})^T(\bar{\boldsymbol{y}} - \boldsymbol{B}\boldsymbol{x}) + \|\bar{\boldsymbol{y}} - \boldsymbol{B}\boldsymbol{x}\|_2^2), \end{aligned} \tag{4.5}$$

where $\bar{\boldsymbol{y}} = \sum_{m=1}^{M} \boldsymbol{y}_m / M$. However,

$$\sum_{m=1}^{M} 2(\boldsymbol{y}_m - \bar{\boldsymbol{y}})^T(\bar{\boldsymbol{y}} - \boldsymbol{B}\boldsymbol{x}) = 2\left(\textstyle\sum_{m=1}^{M} \boldsymbol{y}_m - M\bar{\boldsymbol{y}}\right)^T (\bar{\boldsymbol{y}} - \boldsymbol{B}\boldsymbol{x})$$
$$= 0(\bar{\boldsymbol{y}} - \boldsymbol{B}\boldsymbol{x}) = 0. \tag{4.6}$$

Thus, Eq. 4.6 becomes:

$$\sum_{m=1}^{M} \|\boldsymbol{y}_m - \boldsymbol{B}\boldsymbol{x}\|_2^2 = \sum_{m=1}^{M} \|\boldsymbol{y}_m - \bar{\boldsymbol{y}}\|_2^2 + M\|\bar{\boldsymbol{y}} - \boldsymbol{B}\boldsymbol{x}\|_2^2, \tag{4.7}$$

where the first part of the sum is a constant. Therefore, we obtain the final simplification of our original minimization:

$$\begin{aligned}\tilde{\boldsymbol{x}}_{\ell_1} &= \arg\min_{\boldsymbol{x}} \sum_{m=1}^{M} \|\boldsymbol{y}_m - \boldsymbol{B}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1^2 \\ &= \arg\min_{\boldsymbol{x}} M\|\bar{\boldsymbol{y}} - \boldsymbol{B}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \\ &= \arg\min_{\boldsymbol{x}} \|\bar{\boldsymbol{y}} - \boldsymbol{B}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \end{aligned} \tag{4.8}$$

where M, by division, is absorbed by the constant weight λ. By this sequence, our optimization reduces to the ℓ^1-minimization of $\boldsymbol{x}$ for the mean face track $\bar{\boldsymbol{y}}$.

This conclusion, that enforcing a single, consistent coefficient vector $\boldsymbol{x}$ across all images in a face track $\boldsymbol{Y}$ is equivalent to a single ℓ^1-minimization over the average of all the frames in the face track, is key to keeping our approach robust yet fast. Instead of performing M individual ℓ^1-minimizations over each frame and classifying via some voting scheme, our approach performs a single ℓ^1-minimization on the mean of the face track, which is not only a significant speed up, but

theoretically sound. Furthermore, we empirically validate in subsequent sections that our approach outperforms other forms of temporal fusion and voting amongst individual frames.

Finally, we classify the average test track $\bar{\boldsymbol{y}}$ by determining the class of training samples that best reconstructs the face from the recovered coefficients similar to single image face recognition discussed in the previous chapter. First we compute the class probabilities:

$$p(l_c|\bar{\boldsymbol{y}}) = 1 - \frac{r_c(\bar{\boldsymbol{y}})}{\sum_c r_c(\bar{\boldsymbol{y}})}, \tag{4.9}$$

where $r_c = \|\bar{\boldsymbol{y}} - \boldsymbol{B}_c\boldsymbol{x}_c\|$ is the reconstruction error and $\boldsymbol{x}_c$ are the recovered coefficients from the global solution $\tilde{\boldsymbol{x}}_{\ell_1}$ that belong to class c. The most likely class is then the most probable class:

$$l = \max_c p(l_c|\bar{\boldsymbol{y}}). \tag{4.10}$$

Confidence in the determined identity is obtained using the Sparsity Concentration Index (SCI), which is a measure of how distributed the residuals are across classes:

$$\chi = \frac{C \cdot \max_j \|x_j\|_1 / \|\tilde{\boldsymbol{x}}\|_1 - 1}{C - 1} \in [0, 1], \tag{4.11}$$

ranging from 0 (equally represented all classes) to 1 (fully represented by one class).

4.2 Movie Trailer Face Dataset

Existing datasets do not capture the large-scale identification scope we wish to evaluate. The YouTube Celebrities Dataset [68] has unconstrained videos from YouTube, however they are very low quality and only contain 3 unique videos per person, which they segment. The YouTube Faces Dataset [24] and Buffy Dataset [67] also exhibit challenging scenarios, however YouTube Faces is geared towards face verification, same vs. not same, and Buffy only contains 8 actors; thus, both are ill-suited for the large-scale face identification of our proposed video retrieval framework.

Algorithm 2 Mean Sequence SRC (MSSRC)

1. **Input:** Training gallery $\boldsymbol{B}$, test face track $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_M]$, and sparsity weight parameter λ.
2. Normalize the columns of $\boldsymbol{B}$ to have unit ℓ^2-norm.
3. Compute mean of the track $\bar{\boldsymbol{y}} = \sum_{m=1}^{M} \boldsymbol{y}_m / M$ and normalize to unit ℓ^2-norm.
5. Solve the ℓ^1-minimation problem

$$\tilde{\boldsymbol{x}}_{\ell_1} = \arg\min_{\boldsymbol{x}} \|\bar{\boldsymbol{y}} - \boldsymbol{B}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \quad (4.3)$$

6. Compute class probabilities

$$p(l_c|\bar{\boldsymbol{y}}) = 1 - \frac{r_c(\bar{\boldsymbol{y}})}{\sum_c r_c(\bar{\boldsymbol{y}})} \quad (4.9)$$

7. **Output**: identity l and confidence χ

$$l = \max_c p(l_c|\bar{\boldsymbol{y}}) \quad (4.10)$$

$$\chi = \frac{C \cdot \max_j \|x_j\|_1 / \|\tilde{\boldsymbol{x}}\|_1 - 1}{C - 1} \quad (4.11)$$

4.2.1 Dataset Construction

Face Collection: We built our Movie Trailer Face Dataset using 101 movie trailers from YouTube from the 2010 release year that contained celebrities present in the supplemented PublicFig+10 dataset. These videos were then processed to generate face tracks using the method described above.

Including Distractors: Movies contain many background, unknown actors, therefore during tracking they are automatically captured. Including these distractors, allows us to evaluate how well algorithms perform in terms recognizing known individuals, while rejecting unknowns.

Dataset Statistics: The resulting dataset contains 4,485 face tracks, 65% consisting of unknown identities (not present in PubFig+10) and 35% known, a small sample is shown in Fig. 4.2.

The class distribution is shown in Fig. 4.3 with the number of face tracks per celebrity in the movie trailers ranging from 5 to 60 labeled samples. The fact that half of the public figures do not appear in any of the movie trailers presents an interesting test scenario in which the algorithm must be able to distinguish the subject of interest from within a large pool of potential identities.

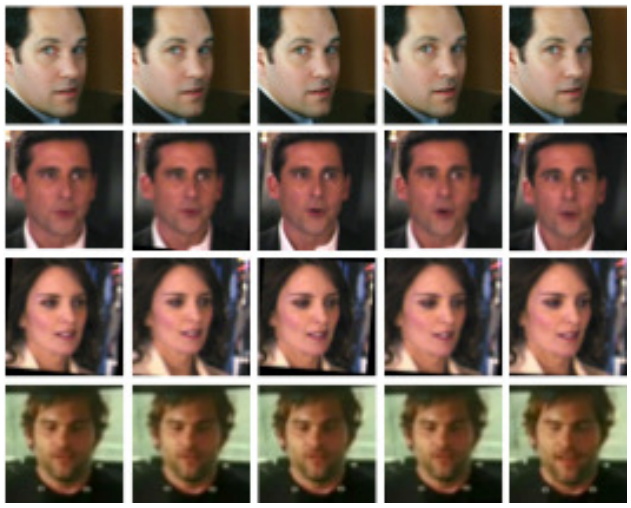


Figure 4.2: Face track samples from our Movie Trailer Face Dataset (MTFD). From top to bottom, Paul Rudd, Steve Carrell, Tina Fey, and Sean William Scott.

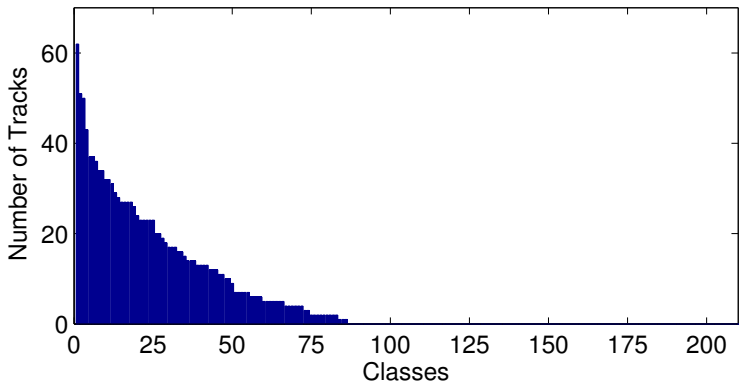


Figure 4.3: The distribution of face tracks across the identities in PubFig+10.

4.2.2 Evaluation Criterion

Just as with the still-image experiments, precision and recall present a good way to show the trade-off between labeling a portion of the data while maintaining high recognition, especially in the presence of unknown individuals.

4.2.3 Dataset Bias

Torralba and Efros [89] emphasized the importance of minimizing the selection, capture, and negative set biases of new datasets. Selection bias refers to the kinds of images or source of the images, *e.g*. nature, Internet search, *etc*. Similarly to the PubFig images, we suffer from a keyword-based selection bias since we only extract faces from 2010 trailers we searched for on YouTube, however it is ameliorated by the fact that we automatically extract faces from these videos keeping both those that are known in PubFig and not. The selection is also biased towards public figures for both PubFig and the Movie Trailer Face datasets that are professionally imaged in most instances. The capture bias references the tendency of photographs to take pictures the same way. For example, since our faces are from movie trailers, although directors may have stylistic differences, the shots taken may be consistent across different movies. The negative set bias refers to what is considered outside of focus or as Torralba and Efros put it, "the rest of the world". 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 is minimized due to the large sampling range offered by data collection via the internet. Furthermore, the Movie Trailer Face dataset has a large negative set of unknown actors that are simply not in PubFig or are unknown background actors.

4.3 Experiments

In this section, we first compare our tracking method to a standard method used in the literature. Then, we evaluate our video face recognition method on three existing datasets, YouTube Celebrities, YouTube Faces, Buffy, and our new Movie Trailer Face dataset. We also evaluate several algorithms, including MSSRC (ours), on our new Movie Trailer Face Dataset, showing the strengths and weaknesses of each and thus proving experimentally the validity of our algorithm.

Table 4.1: Face Tracking Results. Our method outperforms the KLT-based [36] method in terms of MOTA by 2%.

		Method	
Video		KLT [36]	Ours
'The Killer Inside'	MOTP	68.93	69.35
	MOTA	42.88	42.16
'My Name is Khan'	MOTP	65.63	65.77
	MOTA	44.26	48.24
'Biutiful'	MOTP	61.58	61.34
	MOTA	39.28	43.96
'Eat Pray Love'	MOTP	56.98	56.77
	MOTA	34.33	35.60
'The Dry Land'	MOTP	64.11	62.70
	MOTA	27.90	30.15
Average	MOTP	**63.46**	63.19
	MOTA	37.73	**40.02**

4.3.1 Tracking Results

To analyze the quality of our automatically generated face tracks, we ground-truthed five movie trailers from the dataset: 'The Killer Inside', 'My Name is Khan', 'Biutiful', 'Eat, Pray, Love', and 'The Dry Land'. Based on tracking literature [97], we use two CLEAR MOT metrics, Multiple Object Tracking Accuracy and Precision (MOTP and MOTA), for evaluation that better

consider issues faced by trackers than standard accuracy, precision, or recall. The MOTA tells us how well the tracker did overall in regards to all of the ground-truth labels, while the MOTP appraises how well the tracker performed on the detections that exist in the ground-truth.

Although our goal is not to solve the tracking problem, in Tab. 4.1 we show our results compared to a standard face tracking method. The first column shows a KLT-based method [36], where the face detections are associated based on a ratio of overlapping tracked features, and the second shows our method. Both methods are similarly precise, however our metrics have a larger coverage of total detections/tracks by 2% in MOTA with a 3.5x speedup. Results are available online.

Table 4.2: YouTube Faces Results. Results for top performing video face verification algorithm MBGS and our competitive method MSSRC. Note: MBGS results are different from those published, but they are the output of default settings in their system.

Method	Accuracy $\pm$ SE	AUC	EER
MBGS [24]	75.3 ± 2.5	82.0	26.0
MSSRC (Ours)	$\mathbf{75.3 \pm 2.2}$	**82.9**	**25.3**

4.3.2 YouTube Faces Dataset

Although face identification is the focus of this thesis, we evaluated our method on the YouTube Faces Dataset [24] for face verification (same/not same), to show that our method can also work in this context. To the best of our knowledge, there is only one paper [98], that has done face verification using SRC, however it was not in the context of video face recognition, but that of still images from LFW. The YouTube Faces Dataset consists of 5,000 video pairs, half same and half not. The videos are divided into 10 splits each with 500 pairs. The results are averaged over the ten splits, where for each split one is used for testing and the remaining nine for training. The final results are presented in terms of accuracy, area under the curve, and equal error rate. As seen

in Tab. 4.2, we obtain competitive results with the top performing method MBGS [24], within 1% in terms of accuracy, and MSSRC even surpasses it in terms of area under the curve (AUC) by just below 1% with a lower equal error rate by 0.7%. We perform all experiments with the same LBP data provided by [24] and a τ value of 0.0005.

Table 4.3: YouTube Celebrities Results. We outperform the best reported result by 6%.

Method	Accuracy (%)
HMM [68]	71.24
MDA [99]	67.20
SANP [100]	65.03
COV+PLS [101]	70.10
UISA [102]	74.60
MSSRC (Ours)	**80.75**

4.3.3 YouTube Celebrities Dataset

The YouTube Celebrities Dataset [68] consists of 47 celebrities (actors and politicians) in 1910 video clips downloaded from YouTube and manually segmented to the portions where the celebrity of interest appears. There are approximately 41 clips per person segmented from 3 unique videos per actor. The dataset is challenging due to pose, illumination, and expression variations, as well as high compression and low quality. Using our tracker, we successfully tracked 92% of the videos as compared to the 80% tracked in their paper [68]. The standard experimental setup selects 3 training clips, 1 from each unique video, and 6 test clips, 2 from each unique video, per person. In Tab. 4.3, we summarize reported results on YouTube Celebrities, where we outperform the state-of-the-art by at least 6%.

Table 4.4: Buffy Dataset. We obtain a slight gain in accuracy over the reported method.

Method	Accuracy (%)
LDML [67]	85.88
MSSRC (Ours)	**86.27**

4.3.4 Buffy Dataset

The Buffy Dataset consists of 639 manually annotated face tracks extracted from episodes 9, 21, and 45 from different seasons of the TV series "Buffy the Vampire Slayer". They generated tracks using the KLT-based method [36] (available on the author's website). For features, we compute SIFT descriptors at 9 fiducial points as described in [67] and use their experimental setup with 312 tracks for training and 327 testing. They present a Logistic Discriminant-based Metric Learning (LMDL) method that learns a subspace. In their supervised experiments, they tried several classifiers with each obtaining similar results. However, using our classifier, there is a slight improvement (Tab. 4.4).

Table 4.5: Movie Trailer Face Dataset. MSSRC outperforms all of the non-SRC methods by at least 8% in AP and 20% recall at 90% precision.

Method	AP (%)	Recall (%)
NN	9.53	0.00
SVM	50.06	9.69
LDML [67]	19.48	0.00
MLR [69]	45.98	4.62
L2	36.16	0.00
SRC (First Frame)	42.15	13.39
SRC (Voting)	54.88	23.47
MSSRC (Ours)	**58.70**	**30.23**

4.3.5 *Movie Trailer Face Dataset*

In this section, we present results on our unconstrained Movie Trailer Face Dataset that allows us to test larger scale face identification, as well as each algorithms ability to reject unknown identities. In our test scenario, we chose the Public Figures (PF) [13] dataset as our training gallery, supplemented by images collected of 10 actors and actresses from web searches for additional coverage of face tracks extracted from movie trailers. We also cap the maximum number of training images per person in the dataset to 200 for better performance due to the fact that predictions are otherwise skewed towards the people with the most examples. The distribution of face tracks across all of the identities in the PubFig+10 dataset are shown in Fig. 4.3. In total, PubFig+10 consists of 34,522 images and our Movie Trailer Face Dataset has 4,485 face tracks, which we use to conduct experiments on several algorithms.

4.3.5.1 *Algorithmic Comparison*

The tested methods include NN, LDML, SVM, MLR, L2, SRC, and our method MSSRC. For the experiments with NN, LDML, SVM, MLR, L2, and SRC, we test each individual frame of the face track and predict its final identity via probabilistic voting and its confidence is an average over the predicted distances or decision values. The confidence values are used to reject predictions to evaluate the precision and recall of the system. Note all MSSRC experiments are performed with a λ value of 0.01. We present results in terms of precision and recall as defined in [36].

Tab. 4.5 presents the results for the described methods on the Movie Trailer Face Dataset in terms of two measures, average precision and recall at 90% precision. NN performs very poorly in terms of both metrics, which explains why NN based methods have focused on finding "good" key-frames to test on. LDML struggles with the larger number of training classes vs. the Buffy experiment with only 19.48% average precision. The L2 method performs surprisingly well for a simple method. Similarly, MLR struggles at ignoring unknowns, but performs close to SVMs

in terms of average precision. We also tried Mean L2 with similar performance. The SVM and SRC based methods perform very closely at high recall, but not in terms of AP and recall at 90% precision with MSSRC outperforming SVM by 8% and 20% respectively. In Fig. 4.4, the SRC based methods reject unknown identities better than the others.

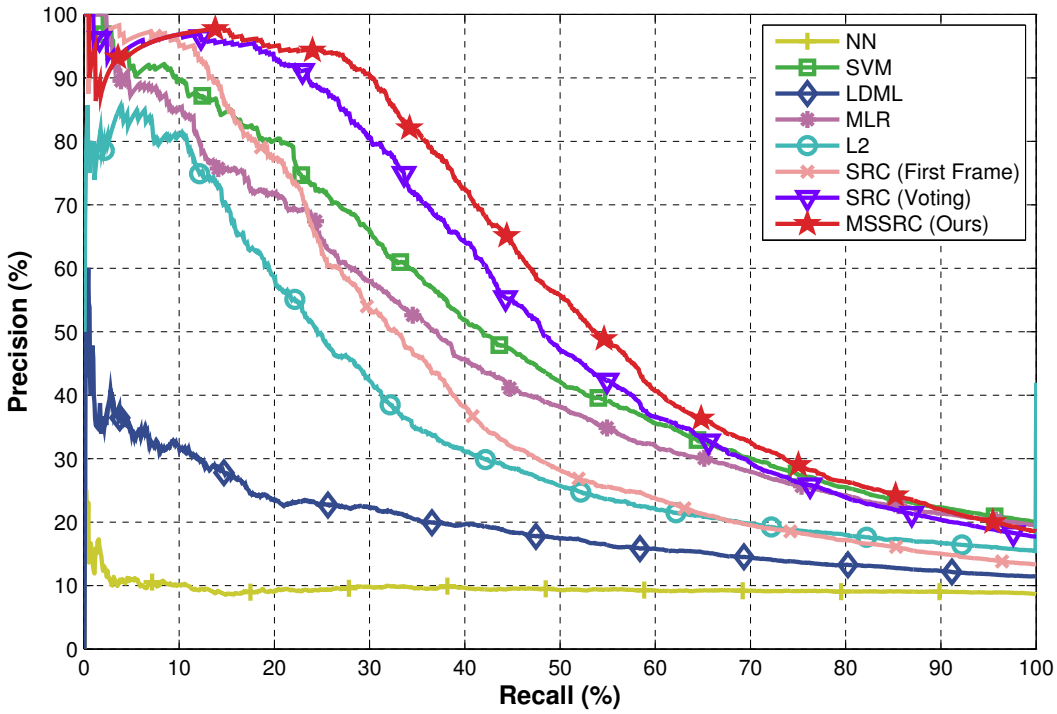
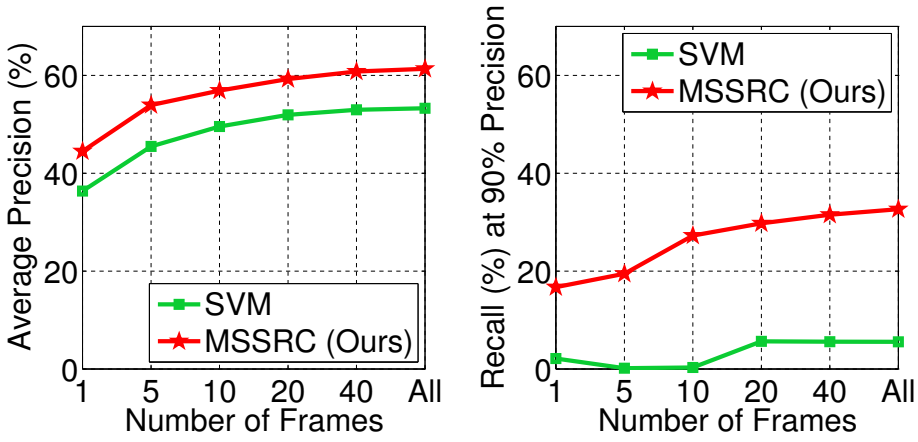


Figure 4.4: Precision vs. Recall for the Movie Trailer Face Dataset. MSSRC rejects unknowns or distractors better than all others.

The straightforward application of SRC on a frame-by-frame basis and our efficient method MSSRC perform within 4% of each other, thus experimentally validating that MSSRC is computationally equivalent to performing standard SRC on each individual frame. Instead of computing SRC on each frame, which takes approximately 45 minutes per track, we reduce a face track to a single feature vector for ℓ^1-minimization (1.5 min/track). Surprisingly, MSSRC obtains better recall at 90% precision by 7% and 4% in average precision. Instead of fusing results after classification, as done on the frame by frame methods, MSSRC benefits in better rejection of uncertain

predictions. In terms of timing, the preprocessing steps of tracking runs identically for SRC and MSSRC at 20fps and feature extraction runs at 30fps. For identification, MSSRC classifies at 20 milliseconds per frame, whereas SRC on a single frame takes 100 milliseconds. All other methods classify in less than 1ms, however with a steep drop in precision and recall.



(a) Average Precision (b) Recall at 90% Precision

Figure 4.5: Effect of Varying Track Length. We see that performance levels out at about 20 frames (close to the average track length). MSSRC outperforms SVM by 8% in average in terms of AP.

4.3.5.2 Effect of Varying Track Length

The question remains, do we really need all of the images? To answer this question we select the first m frames for each track and test the two best performing methods from the previous experiments: MSSRC and SVM. Fig. 4.5 shows that at just after 20 frames performance plateaus, which is close to the average track length of 22 frames. Most importantly, the results show that using multiple frames is beneficial since moving from using 1 frame to 20 frames results in a 5.57% and 16.03% increase in average precision and recall at 90% precision respectively for MSSRC. Furthermore, Fig. 4.5 shows that the SVM's performance also increases with more frames, although MSSRC outperforms the SVM method in its ability to reject unknown identities.

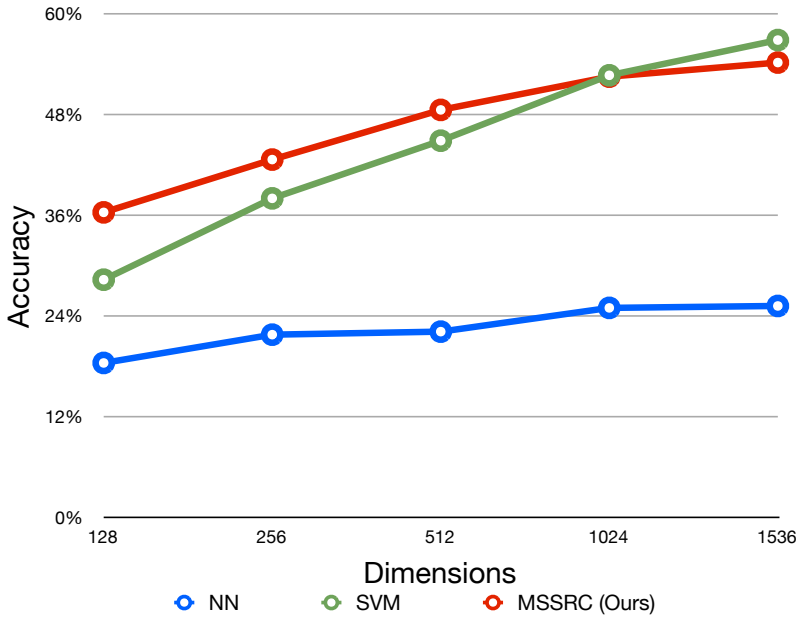


Figure 4.6: Classification as a function of PCA dimension. As the dimensionality increases, the accuracy begins to plateau at about 1024 for SVM and 512 for MSSRC.

4.3.5.3 Effect of Dimensionality Reduction

Fig. 4.3.5.2 shows the effect of dimensionality reduction on three algorithms, NN, SVM, and MSSRC. Increasing the number of dimensions benefits the SVM method the most, with all methods plateauing around 1536 dimensions for each feature. We cannot increase this any further since it is the maximum dimensionality of HOG's selected parameters.

4.3.6 Combining MSSRC with LASRC

An obvious extension of our method MSSRC is to combine our approximation method presented in the previous chapter to speedup computation. This extension is straightforward requiring adding the least-squares approximation and selecting the dictionary elements corresponding to the largest coefficient values. In this section, we explore the effect of varying the approximation value, which determines how many dictionary elements to keep to pass to the ℓ^1-approximation.

We vary the approximation value from 128 to all dictionary elements and record performance in terms of average precision, recall at 90% precision, and test time per track as shown in Tab. 4.6. With a small approximation value of 128, performance is very poor, however with an approximation value of 10,000 (1/3 of the data), there is a small performance loss of 1-2% in terms of average precision and recall at 90% precision with a 3x speedup.

Table 4.6: Effect of varying approximation value on speed after combining LASRC and MSSRC.

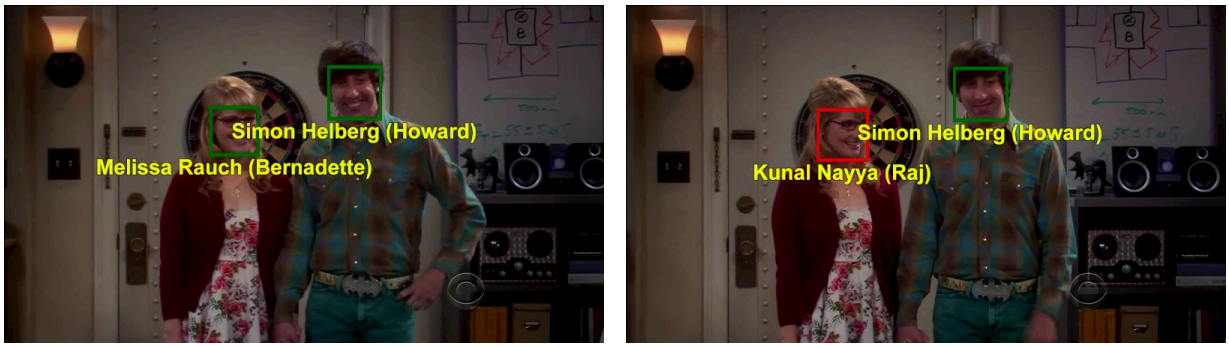
Approx. Value	AP (%)	Recall (%)	Test Time (s/track)
128	41.6	13.0	1.0
256	44.1	16.4	1.3
512	46.6	17.0	1.6
2056	50.8	21.2	4.8
8224	55.8	27.0	40.0
10000	57.0	28.6	69.0
ALL	58.7	30.2	211.0

4.4 Summary

In this chapter, we presented a fully automatic end-to-end system for video face recognition, which includes face tracking and identification leveraging information from both still images for the known dictionary and video for recognition. Our simple, yet efficient face tracking algorithm compares well to an existing popular method with a 3.5x speedup. We proposed a novel algorithm Mean Sequence SRC, MSSRC, that performs a joint optimization using all of the available image data to perform video face recognition. We finally showed that our method outperformed the state-of-the-art on real-world, unconstrained videos in our new Movie Trailer Face Dataset. Furthermore, we showed our method especially excels at rejecting unknown identities outperforming the next best method in terms of average precision by 8%. Video face recognition presents a very compelling area of research with difficulties unseen in still-image recognition.

CHAPTER 5: AFFINITY-BASED VIDEO FACE IDENTIFICATION

Although, MSSRC provides state-of-the-art performance, as shown in the previous chapter, several misclassifications still exist. As shown in Fig. 5.1, in a scene from the popular sitcom "The Big Bang Theory" the character "Bernadette" is classified correctly by MSSRC, but later in the scene the character is misclassified. Given global knowledge, *i.e*. the relationship between face tracks, a confident classification could help correct weak misclassifications. In this Chapter, we describe our two-stage method to perform more consistent recognition. Stage 1 performs classification as described in the previous chapter. Stage 2 encodes the visual relationship, classification similarity, and label co-occurrence, to describe the relationship between each face track. Given the affinity of the face tracks within a video sequence, we use Random Walks to smooth the initial predictions by propagating strong correct classifications and dampening weak misclassifications.



(a) Earlier in Scene (b) Later in Scene

Figure 5.1: The Big Bang Theory Labeling Error. In this scene of The Big Bang Theory, there is a correct labeling of Bernadette, followed by a jump cut and misclassification of a similar looking face track of Bernadette as Raj.

5.1 Affinity-based Propagation Method

In this section, we describe our affinity-based propagation method. This technique assumes initial class predictions and confidences provided by MSSRC shown in Eqn. 4.10 and Eqn. 4.11 respectively. As previously mentioned, our method first constructs an affinity graph relating every face track based on appearance, classification similarity, and label co-occurrence. Our method then propagates these predictions using an affinity graph and random walk analysis as shown in Fig. 5.2.

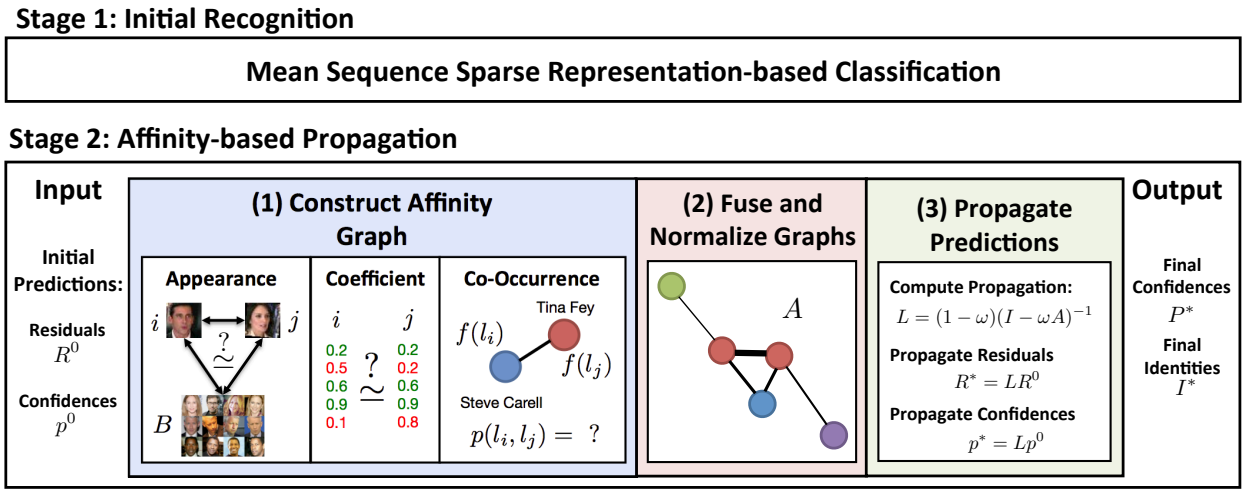


Figure 5.2: Affinity-based propagation takes initial predictions from our algorithm MSSRC and uses the affinity between face tracks in a video to smooth the initial predictions and converge on the final corrected labels.

5.1.1 Face Track Affinity

When creating a relationship among face tracks within a video, we must first consider what information the face tracks provide. The most obvious and powerful is the appearance information because we know that within a video the appearance of a person will remain more or less consistent. Next, we employ the coefficient vector denoting a face track's relationship to the images in the dictionary obtained via SRC, since similar face tracks should correspond to similar images in the

dictionary. Finally, the predictions computed by the face recognizer provide useful information in that we know the occurrence and co-occurrence of the assigned labels. Combining these three relationships provides a strong description of how information should be shared throughout a given video.

Appearance Affinity: For the appearance graph we use the Matched Background Similarity (MBGS) [24], which has been effective in the face verification task (same vs. not same). The MBGS metric computes a set-to-set distance between two face tracks $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ using a background set of images $\boldsymbol{B}$. First, the K nearest neighbors of $\boldsymbol{Y}_1$ to $\boldsymbol{B}$ are used as a negative set to train an SVM with $\boldsymbol{Y}_1$ as the positive set. Next, the trained model is used to classify the frames from track $\boldsymbol{Y}_2$ on which we compute the mean prediction score. We do the same for the second face track $\boldsymbol{Y}_2$ and then compute the final score as an average of the two. Using this metric we can compute the pairwise appearance similarity between each face track:

$$d_a(i,j) = MBGS(\boldsymbol{Y}_i, \boldsymbol{Y}_j, \boldsymbol{B}). \tag{5.1}$$

Intuitively, this metric answers the question do the face tracks look more like each other or the background set.

Coefficient Affinity: Given the output of the SRC-based method, if two face tracks are similar, we know that they should have a similar coefficient vector, i.e. they should be reconstructed by similar images in the training dictionary. Therefore, we employ the cosine distance between coefficient vectors to compute another pairwise similarity:

$$d_c(i,j) = \frac{\boldsymbol{x}_i \cdot \boldsymbol{x}_j}{\|\boldsymbol{x}_i\| \|\boldsymbol{x}_j\|} \tag{5.2}$$

where x_i and x_j are the coefficient vectors of face tracks i and j respectively computed by SRC.

Co-Occurrence Affinity: Finally, we consider the co-occurrence similarity of the labels of the two face tracks. Using the label predictions, we compute the frequency of each label in a

given trailer and use these to compute the Normalized Google Distance [103] as follows:

$$d_o(i,j) = \frac{\max(\log f(l_i), \log f(l_j)) - \log f(l_i, l_j)}{\log G - \min(\log f(l_i), \log f(l_j))}, \tag{5.3}$$

where $f(l_i)$ and $f(l_j)$ are the frequencies of the predicted labels from tracks i and j respectively, $f(l_i, l_j)$ is the frequency of the two labels occurring togethers, and G is the total number of predictions. We can handle $f(l_i, l_j)$ in two ways. 1) We assume it is zero, since no face track should receive two labels or 2) we can take the top k predictions from the classifier and compute frequencies and co-occurring frequencies based on these values. In our experimentation, both assumptions yielded near identical results, therefore we stick with (1) for simplicity, which reduces to a normalized co-occurrence.

5.1.2 Affinity Fusion

The computation of the aforementioned similarity metrics (Appearance, Coefficient, and Co-Occurrence) allows us to construct an affinity relationship between face tracks by fusing all three. We first convert our affinities into probabilities utilizing the standard sigmoid function and combine them using a weighted mean as follows:

$$\begin{aligned} d(i,j) = \alpha_a \exp\left(d_a(i,j)/\sigma_a\right) + \alpha_c \exp\left(d_c(i,j)/\sigma_c\right) \\ + \alpha_o \exp\left(d_o(i,j)/\sigma_o\right), \end{aligned} \tag{5.4}$$

where σ's and α's are the fusion weighing and smoothing parameters respectively, thus forming the elements of similarity matrix D.

5.1.3 *Random Walk Over Label Affinities*

Random walk techniques are popular in the retrieval domain [104] and semi-supervised learning [105] because of their wide applicability. Random walks can be understood as the likeliness of transitioning from node i to node j by means of a probabilistic walk between the most likely nodes in a graph. In our scenario, the nodes are the face tracks and the transition probability we wish to model is the likelihood that pairs of nodes are of the same person. We compute the transition probability between face tracks by normalizing the similarity matrix $\boldsymbol{D}$ by the row sum:

$$a(i,j) = \frac{d(i,j)}{\sum_k d(i,k)}, \tag{5.5}$$

where $d(i,j)$ is the similarity between nodes i and j as defined above in Eqn. 5.4, forming affinity matrix $\boldsymbol{A}$.

Given the transition probability matrix $\boldsymbol{A}$ obtained via normalization of similarity matrix $\boldsymbol{D}$, we can define the propagation of labels across the nodes resulting in a sharing of information with related face tracks (nodes). Such a propagation scheme requires that the label probabilities of highly related nodes be increased and inversely weak labels must be decreased. Let us first consider the binary case in which we have the probability of each node belonging to the known positive class. We can then formulate the label propagation as a weighted sum of the original prediction and the surrounding node probabilities based on their class probability and affinity to the node of interest. Therefore, we can iteratively propagate the class probabilities across the face tracks until an agreement is achieved. The resulting formulation is as follows:

$$p^t(i) = \omega \sum_j p^{t-1}(j) a(i,j) + (1-\omega) p^0(i), \tag{5.6}$$

where $p^{t-i}(j)$ is the predicted class probability from the previous iteration, $a(i,j)$ is the probability of transition between nodes i and j, $p^0(i)$ denotes the initial probability of the current node of

interest j, and ω specifies how much of the current and previous probabilities to keep. Given that we want to propagate the labels for every node, we can rewrite Eqn. 5.6 in matrix form:

$$\boldsymbol{p}^t = \omega \boldsymbol{A} \boldsymbol{p}^{t-1} + (1-\omega)\boldsymbol{p}^0, \tag{5.7}$$

where $\boldsymbol{p}^0$ is the initial class probability of the nodes provided by MSSRC.

It can be shown that the iterative method has a unique solution $\boldsymbol{p}^*$ following the derivation in [104, 105]. To do this, we evaluate the limit over Eqn. 5.7 given that the initial labeling is $\boldsymbol{p}^0$:

$$\boldsymbol{p}^* = \lim_{n\to\infty} (\omega \boldsymbol{A})^n \boldsymbol{p}^0 + (1-\omega) \sum_{i}^{t-1} (\omega \boldsymbol{A})^i \boldsymbol{p}^0 \tag{5.8}$$

Knowing that $p(i,j) \geq 0$ and $\sum_j p(i,j) = 1$, we can extrapolate from the Perron-Frobenius Theorem that the spectral radius of $\boldsymbol{A}$ is $\rho(\boldsymbol{A}) \leq 1$. Since $0 < \omega < 1$, we can say:

$$\lim_{t\to\infty} (\omega \boldsymbol{A})^{t-1} = 0.$$

Then knowing that the following limit is a geometric series, we obtain:

$$\lim_{t\to\infty} \sum_{i=0}^{t-1} (\omega \boldsymbol{A})^i = (\boldsymbol{I} - \omega \boldsymbol{A})^{-1},$$

where $\boldsymbol{I}$ is the identity matrix. Therefore, the sequence $\{\boldsymbol{p}^t\}$ converges to:

$$\boldsymbol{p}^* = (1-\omega)(\boldsymbol{I} - \omega \boldsymbol{A})^{-1} \boldsymbol{p}^0. \tag{5.9}$$

Given the solution $\boldsymbol{p}^*$, we can determine class association for each node.

For the multi-class scenario, we replace $\boldsymbol{p}^0$ with the matrix $\boldsymbol{P}^0$ containing the class proba-

bilities for each node, which results in:

$$\boldsymbol{P}^* = (1-\omega)(\boldsymbol{I} - \omega\boldsymbol{A})^{-1}\boldsymbol{P}^0. \tag{5.10}$$

Therefore, the labels for each class are determined as follows:

$$l_i = \max_{c \leq C} \boldsymbol{P}^*_{ic}, \tag{5.11}$$

where i is the node or face track of interest, c is the current class, and C is the number of classes.

Similarly, for the SRC-based method we propagate the confidence in the prediction χ for each node:

$$\boldsymbol{\chi}^* = (1-\omega)(\boldsymbol{I} - \omega\boldsymbol{A})^{-1}\boldsymbol{\chi}^0, \tag{5.12}$$

where the resulting values provide an accurate rejection criterion.

Table 5.1: The Big Bang Theory Dataset. MSSRC performs comparatively to the best reported results, but when combined with affinity-based propagation it outperforms the state-of-the-art by 4%.

Method	BBT-1	BBT-2	BBT-3	BBT-4	BBT-5	BBT-6	BBT Avg.
MLR+MRF (Reported)	95.18	**94.16**	77.81	79.35	79.93	75.85	83.71
MSSRC	94.47	89.56	82.84	81.58	81.05	84.37	85.65
MSSRC+Affinity	**95.19**	90.53	**86.00**	**84.21**	**83.11**	**85.91**	**87.49**

5.2 Experiments

In this section, we explore how well our affinity-based propagation method works on two difficult datasets, one from the TV sitcom "The Big Bang Theory" and the other from movie trailers. Exploring The Big Bang Theory dataset allows us to see how well our method labels the

known cast throughout entire episodes from the sitcom, which are much longer than the movie trailers we analyze. However, the Movie Trailer Face Dataset allows us to consider the scenario where there are many unknowns to reject, unlike the The Big Bang Theory. In both scenarios, recognition benefits from affinity-based propagation.

Algorithm 3 Affinity-based Propagation

1. **Input:** Face Tracks $[Y_1, \ldots, Y_N]$, Training gallery B, Initial Predictions P^0 and confidences $\boldsymbol{\chi}^0$.
2. For each face track pair compute affinities:

$$d_a(i,j) = MBGS(\boldsymbol{Y}_i, \boldsymbol{Y}_j, \boldsymbol{B}) \quad (5.1)$$

$$d_c(i,j) = \frac{x_i \cdot x_j}{\|x_i\|\|x_j\|} \quad (5.2)$$

$$d_o(i,j) = \frac{\max(\log f(l_i), \log f(l_j)) - \log f(l_i, l_j)}{\log G - \min(\log f(l_i), \log f(l_j))} \quad (5.3)$$

3. Fuse affinity metrics:

$$d(i,j) = \sum_{k=\{a,o,c\}} \alpha_k \exp\left(d_k(i,j)/\sigma_k\right) \quad (5.4)$$

5. Normalize affinity for random walk:

$$a(i,j) = \frac{d(i,j)}{\sum_k d(i,k)} \quad (5.5)$$

6. Propagate class prediction probabilities and confidences:

$$\boldsymbol{P}^* = (1-\omega)(\boldsymbol{I} - \omega \boldsymbol{A})^{-1} \boldsymbol{P}^0 \quad (5.10)$$

$$\boldsymbol{\chi}^* = (1-\omega)(\boldsymbol{I} - \omega \boldsymbol{A})^{-1} \boldsymbol{\chi}^0 \quad (5.12)$$

7. **Output**: confidences $\boldsymbol{\chi}$ and identities:

$$l_i = \max_{c \leq C} \boldsymbol{P}^*_{ic} \quad (5.11)$$

5.2.1 *The Big Bang Theory*

The Big Bang Theory dataset [69] consists of 3,759 face tracks across the first six episodes of the first season of the popular show. There are a total of 11 actors that are known and one additional "unknown" label. The training data is collected by using a weakly supervised technique matching a video's speaker with the name in the script. Here we evaluate performance using accuracy, where all of the unknown characters are considered as one class. The best reported method combines Maximum Likelihood Regression (MLR) and Markov Random Fields (MRF) for an average performance over all of the episodes of 83.7% as shown in Table 5.1. We also show MSSRC's performance, where we use the residual errors as a threshold to label unknowns. We use the weakly-labeled samples for each individual episode as the dictionary, except for the characters Raj and Howard where we use examples from all episodes to balance the dictionary. Using MSSRC, we are able to get a 2% increase and adding affinity-based propagation we get a 4% improvement over the state-of-the-art. We find that the increase due to affinity-based propagation is 2% over MSSRC since most misses are due to "unknowns" and characters that have very few examples in the dictionary.

5.2.2 *Movie Trailer Face Dataset*

We now explore our Move Trail Face Dataset following the same experimental setup described in the previous chapter. In Fig. 5.3, we show a graphical analysis of the movie "Date Night" before and after label propagation. In this sample it is evident that the graphs are divided into two distinct groups representing the two main characters in each movie. Furthermore, before propagation there is substantial confusion in the center of the graph where all of the "unknown" actors are concentrated with a few misclassifications within the two main character clusters. After label propagation, the misclassifications within the main character clusters are corrected, especially evident when zooming in on Tina Fey (Fig. 5.4). Moreover, confidence within the central region is

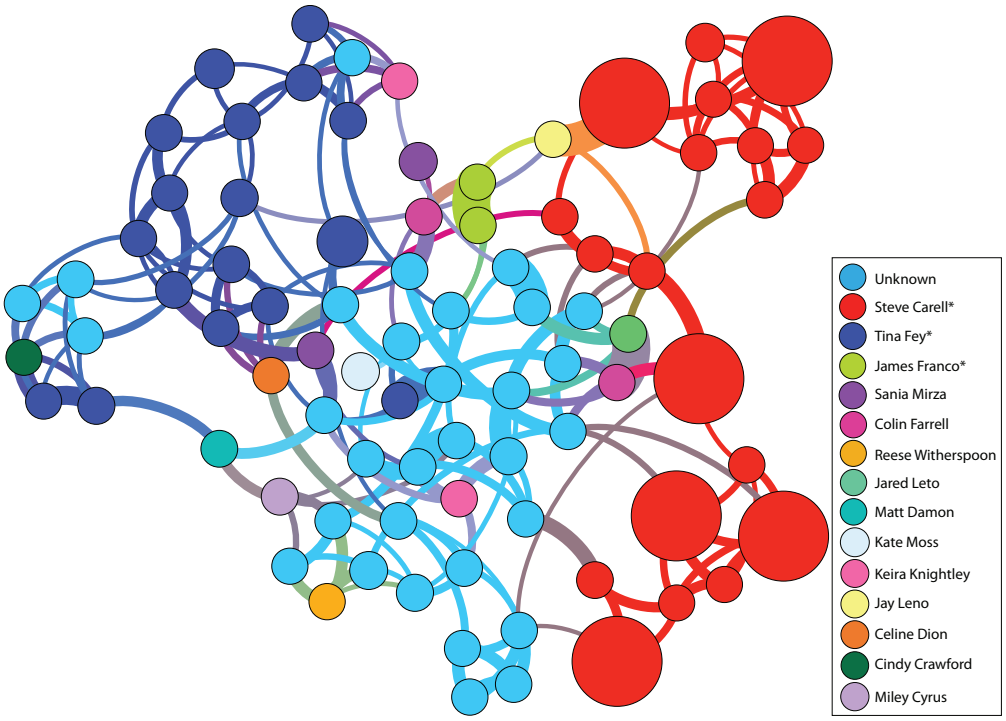
correctly weakened so that in the end there is less confusion.

For repeatability, the affinity propagation parameters for the different schemes are reported in Tab. 5.2. For all experiments, $\sigma_a = 1$, $\sigma_c = 1$, and $\sigma_o = 1$ as not much improvement was found by changing the smoothing parameter. All other parameters are obtained using a greedy parameter search, where the α's determine contribution from different affinities, K defines how many nearest neighbors each affinity graph uses, and the ω's defines how much the propagation scheme weighs the surrounding face track contribution versus the original class probabilities. For the fusion schemes, we optimize for accuracy, maximum average precision, and recall at high precision individually, emphasizing different goals. For example, if we are in a closed-universe scenario, accuracy over known individuals in the training dictionary is more important. However, if we are in an open-universe scenario in which we want to maximize rejection of unknowns with very accurate annotation, then maximizing precision is more important.

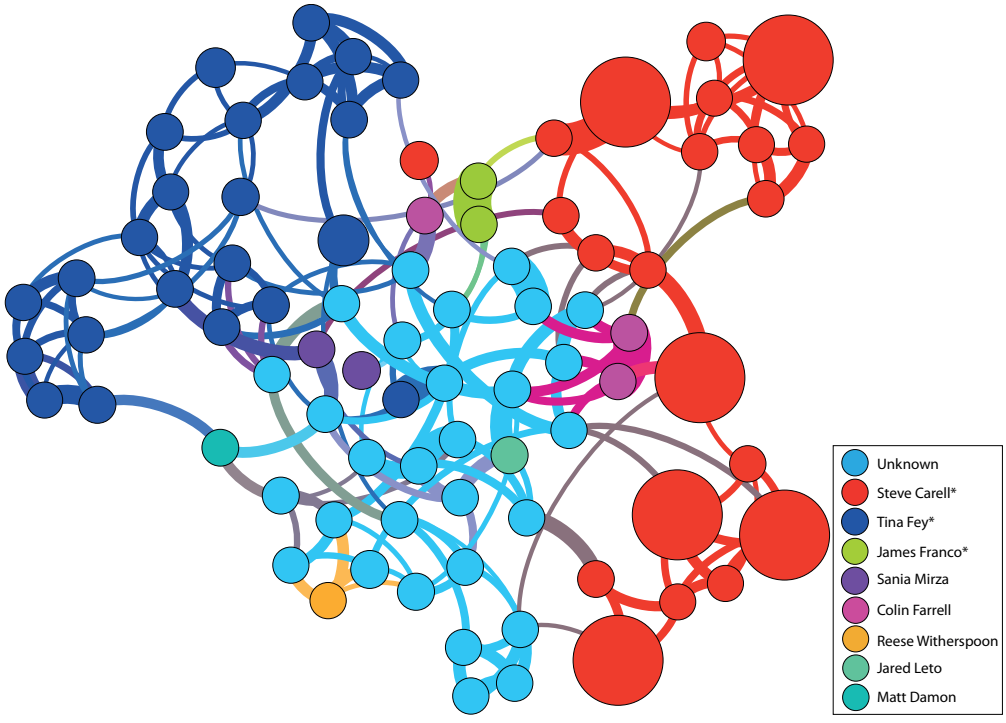
In Table 5.3, we show the baseline result for MSSRC followed by the result of applying affinity-based propagation using the individual similarity metrics and their fusion. Compelling results occur during the fusion of the different affinity metrics, we optimize the parameters for three different criterions: 1) Accuracy, 2) Average Precision, and 3) Recall at High Precision.

Accuracy: Maximum accuracy models a closed-universe where all of the face tracks are of known identities in the dictionary. Best results occur by propagating the initial predictions with an increase of about 34% accuracy. Optimizing for high accuracy, however, negatively impacts recall at high precision shown in Fig. 5.5 by 12.9% at 95% precision to 18.5% at 90% precision.

Average Precision: Pursuing maximum average precision models an open-universe, where we want a balance between classifying known identities well while rejecting unknowns with good precision. Optimizing for average precision using MSSRC results in an increase of 12.6%. Its benefit is evident by the teal line in Fig. 5.5 that shows an increase over the baseline (MSSRC) and shows it gives the best compromise in terms of average precision and accuracy compared to the other fusion schemes, outperforming all curves except at the lower recall values (below 40%).

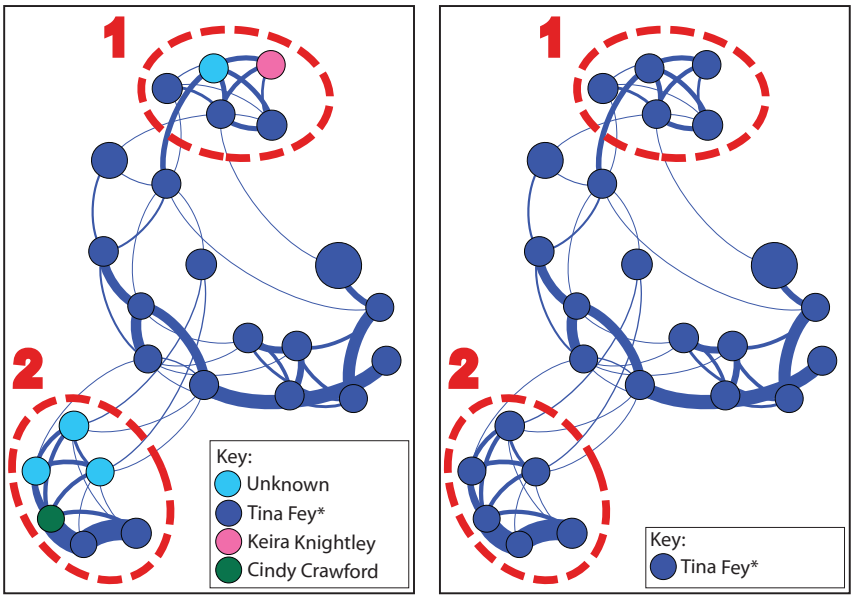


(a) Date Night Before Propagation



(b) Date Night After Propagation

Figure 5.3: Date Night Before and After Propagation. Accuracy increases from 73.6 to 88.7% and 88.6 to 94.3% in average precision. (*Cast Members)



(a) Before Propagation (b) After Propagation

Figure 5.4: Subgraph from the movie Date Night for the actress Tina Fey where each node is a face track. Within groups 1 and 2, denoted by red, dashed ellipticals in Fig. 5.4(a), there are errors in labeling nodes that should have been labeled Tina Fey. After affinity-based propagation, the errors are corrected as shown in Fig 5.4(b).

High Precision: Optimizing for recall at high precision as showcased by the red line in Fig. 5.5 provides a substantial increase over the baseline method, however does not outperform the Max AP scheme in overall precision. The results show that over 30% of the data can be labeled at greater than 95% precision and 37% of the data can be labeled at 90% precision via graph propagation, which is an increase of 11% and 7% respectively.

Tab. 5.2 also summarizes the relative contribution of each individual affinity metrics. The coefficient and appearance affinities attain similar results lagging behind max accuracy fusion by approximately 9% and max average precision fusion by less than 2%. Interestingly, if the goal is maximum AP, then using only the coefficient similarity is a viable option because it is quicker to compute than the appearance similarity and only results in about 2% drop over fusing all of the affinities. The coefficient affinity outperforms appearance by 9% recall at 95% precision. The

results for individual affinities are directly comparable to fusion while maximizing average precision. Fusion excels when maximizing accuracy and recall at high precision, where maximizing accuracy results in an increase of at least 9% over the individual affinities and 6% at high precision when maximizing recall at high precision.

Table 5.2: Affinity-Based Propagation Propagation Parameters. These are the resulting parameters after optimizing for different metrics: average precision, accuracy, and recall at high precision. The K parameter defines how many nearest-neighbors to use for graph construction, ω parameter defines how much to weighing surrounding node contribution versus its initial label, and the αs define how much to weigh the three different affinity metrics.

Parameters	K	ω	α_a	α_c	α_o
MSSRC	N/A	N/A	N/A	N/A	N/A
Appearance	5	0.7	1	0	0
Coefficient	10	0.7	0	1	0
Co-Occurrence	All	0.3	0	0	1
Fusion (Max Acc)	5	0.9	0.6	0.4	0
Fusion (Max AP)	5	0.8	0.6	0.3	0.1
Fusion (High Prec)	15	0.5	0.5	0.3	0.2

Table 5.3: Affinity-Based Propagation Results. We obtain a peak increase of 34.5% in accuracy and 12.6% in average precision.

Metrics	Acc.	AP	R@90P	R@95P
MSSRC	50.52	58.70	30.23	20.48
Appearance	75.36	70.02	31.14	16.45
Coefficient	75.62	69.40	31.34	25.42
Co-Occurrence	66.71	63.35	26.01	21.33
Fusion (Max Acc)	**84.98**	60.51	11.77	7.61
Fusion (Max AP)	77.18	**71.30**	31.14	16.51
Fusion (High Prec)	62.81	67.26	**37.52**	**31.79**

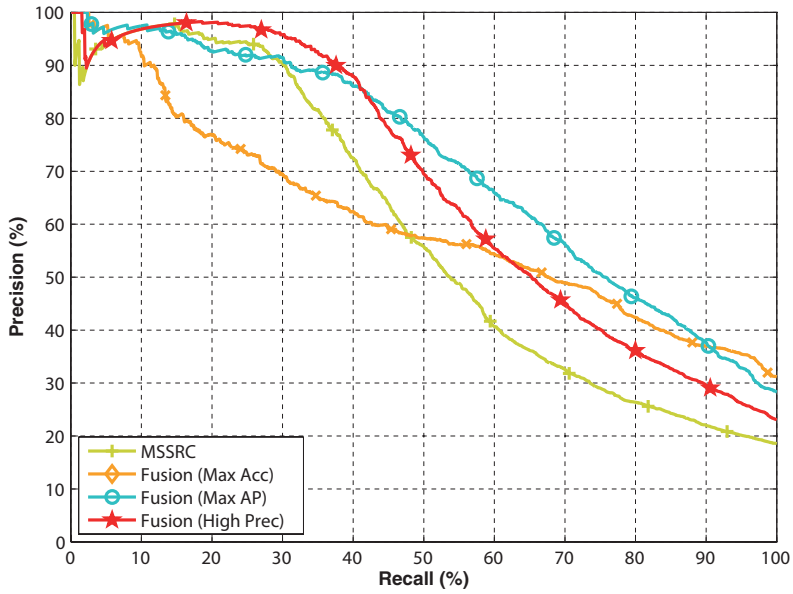


Figure 5.5: Affinity-based Propagation Precision and Recall Curves. Every affinity metric and fusion scheme provide different benefits over the baseline using no propagation. Appearance and coefficient affinities perform comparatively, where fusion optimized for high precision provides the best balance between all metrics.

5.3 Summary

In this chapter, we presented our method for affinity-based propagation. Observing that many misclassified face tracks in scenes were similar to other correctly classified tracks, we determined that having a global perspective would aid recognition performance. Our technique builds an affinity graph using the appearance and co-occurrence similarities to model the relationship between face tracks in a scene. Using this relationship graph, we employ random walk analysis to propagate strong class predictions among similar face tracks, while dampening weak predictions. In the experiments, we obtain state-of-the-art performance on the Big Bang Theory dataset and convincingly show that for our difficult Movie Trailer Face Dataset affinity-based propagation helps to more consistently label tracks correctly with increased performance in terms of accuracy and average precision.

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

In this dissertation, we explore the difficult task of open-universe face identification, where the goal is to not only recognize faces precisely, but also reject unknown individuals. This objective best describes web-scale applications like those found for auto-tagging photo albums and movies, where a specific set of people are of interest. In the case of social-sharing sites, a user may only be interested in images of his/her friends, while in a movie there are many background actors that are not well known and of little interest. For the task of open-universe face identification, we present several novel solutions and analyze the problem in depth.

Noting that the popular method Sparse Representation-based Classification (SRC) for still-image face identification is computationally expensive and thus ill-suited for web-scale deployment, we propose a new method Linearly Approximated Sparse Representation-based Classification (LASRC). We make the observation that knowing the zero coefficients ahead of time would greatly accelerate the search for the optimal, sparse coefficient vector. Observing that the ℓ^2-approximation produces a noisy version of the sparse coefficient vector with a similar structure, we combine the speed of least-squares to select a small subset of the training dictionary to pass along to the ℓ^1-approximation. In our experiments, we find that for large datasets collected from Facebook the approximated subset can be as small as 64 elements, with a speedup of at least 100x over standard SRC methods, and nearly identical performance in terms of accuracy, precision, and recall. Along with these combined benefits, we conclude that LASRC combined with local features results in a top contender for web-scale face identification.

Subsequently, we move from still-image to video face identification. We propose a complete end-to-end system from face tracking to video face identification. Our proposed face tracker, performs comparatively well to another popular detection based tracker with a 5x speedup. Next, we extend the Sparse Representation-based Classification (SRC) framework to video face recognition. A naïve application of SRC on a frame-by-frame basis for a video face track is too computa-

tionally expensive to be feasible, therefore we propose Mean Sequence SRC (MSSR) that performs a single joint optimization using the entire face track. This optimization interestingly reduces to the ℓ^1-minimization over the mean vector. Our method outperforms the next best method in terms of average precision by 8% and a 5x speedup over frame-by-frame SRC. Further, by combining LASRC and MSSRC we obtain a 3x speedup over MSSRC alone with a degradation of only 2% in average precision.

Finally, our method, MSSRC, treats each face track in a video independently, not sharing any information. In the resolution of this dissertation, we propose to augment MSSRC with a graph-based relationship to share information across the face tracks in a given video. This extension corrects misclassifications in which the classifier fails due to large pose or other variations by sharing the information with a closely related and correctly classified neighbor. Our affinity-based propagation method improved the state-of-the-art performance by ~4% on the Big Bang Theory dataset. On our dataset, Movie Trailer Face Dataset (MTFD), there is a substantial increase in performance with a peak increase of 34.5% in accuracy and 12.6% in average precision depending on the optimization metrics.

6.1 Future Work

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. Recent research in the object recognition community has benefited from dictionary selection [106, 107], where key photos are retained in the gallery, while others that are not useful are dropped. 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. GPU acceleration would be especially beneficial when batch labeling albums of photos. For better accuracy,

new feature representations could be explored. More specifically, features that are explicitly designed for spare representation could result in a substantial speedup and further increase in the recognition rate. In situations where many training faces per subject or frontal faces are not available, more evaluation is needed, *i.e*. dictionary learning techniques that only use a single example for training or methods to extrapolate a frontal face from profile views.

In regards to video face recognition, one possible future work would explore the effect of selecting key-frames, or less noisy frames, *i.e*. dropping noisy, occluded or poorly aligned, faces could boost performance. Instead of key-frames, another option would be to find representative means of the test data. In our current setup we use mostly frontal faces, however relaxation of the face detection parameters would capture non-frontal faces. Using multiple means would help in the case of extreme pose where it would be difficult to impose a single reconstruction from the training data. Possibly one could have multiple dictionaries, one for each pose type, ranging from frontal to profile, which would eliminate the need for perfect face alignment and allow the recognition across different views. Furthermore, there is a whole area of domain transfer [108, 109], which would be advantageous in discovering a relationship between the still-image training gallery and video face tracks. Basically, several unconstrained features, like the lighting and sensor type, are very different between the still-images and videos, therefore a good mapping between the two domains would be beneficial. Finally, future work should look at combining the ℓ^1-minimization and affinity-based propagation stages into a single optimization framework, as we believe they can aid each other in finding the optimal solution in one single step.

LIST OF REFERENCES

[1] D. Henschen, "Facebook On Big Data Analytics: An Insider's View," 2013.

[2] Google, "YouTube Statistics," 2013.

[3] A. T. L. Cambridge, "The Database of Faces."

[4] A. R. Martinez and R. Benavente, "The AR Face Database," tech. rep., Computer Vision Center (CVC), 1998.

[5] A. Georghiades, D. Kriegman, and P. N. Belhumeur, "From Few to Many: Generative Models for Recognition Under Variable Pose and Illumination," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[6] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET Database and Evaluation Procedure for Face Recognition Algorithms," *ELSEVIER Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.

[7] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1615–1618, 2003.

[8] A. O'Toole, P. Phillips, F. Jiang, J. Ayyad, N. Pénard, and H. Abdi, "Face Recognition Algorithms Surpass Humans Matching Faces Over Changes in Illumination," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1642–1646, 2007.

[9] P. Phillips, P. Grother, R. Micheals, D. BlackBurn, E. Tabassi, and M. Bone, "Face Recognition Vendor Test 2002," *NIST*, vol. 6965, 2003.

[10] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the Face Recognition Grand Challenge," in *Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 947–954, IEEE, 2005.

[11] P. J. Grother, G. W. Quinn, and P. J. Phillips, "Report on the Evaluation of 2D Still-Image Face Recognition Algorithms," *NIST*, vol. 7709, 2011.

[12] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," tech. rep., University of Massachusetts, Amherst, 2007.

[13] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar, "Describable Visual Attributes for Face Verification and Image Search," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1962–1977, 2011.

[14] Z. Stone, T. Zickler, and T. Darrell, "Autotagging Facebook: Social Network Context Improves Photo Annotation," in *Conf. on Computer Vision and Pattern Recognition Workshop*, pp. 1–8, IEEE, 2008.

[15] B. Becker and E. Ortiz, "Evaluation of face recognition techniques for application to Facebook," in *Automatic Face & Gesture Recognition*, pp. 1–6, IEEE, 2008.

[16] N. Pinto, Z. Stone, T. Zickler, and D. Cox, "Scaling Up Biologically-Inspired Computer Vision: A Case Study in Unconstrained Face Recognition on Facebook," in *Conf. on Computer Vision and Pattern Recognition*, pp. 35–42, IEEE, 2011.

[17] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma, "Towards a Practical Face Recognition System: Robust Alignment and Illumination by Sparse Representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, no. 34, pp. 372–386, 2011.

[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-Constrained Linear Coding for Image Classification," in *Conf. on Computer Vision and Pattern Recognition*, pp. 3360–3367, IEEE, 2010.

[19] M. Yang and L. Zhang, "Gabor Feature based Sparse Representation for Face Recognition with Gabor Occlusion Dictionary," in *European Conf. on Computer Vision*, pp. 448–461, 2010.

[20] J. Huang and M. Yang, "Fast Sparse Representation with Prototypes," in *Conf. on Computer Vision and Pattern Recognition*, pp. 3618–3625, IEEE, 2010.

[21] Q. Shi, C. Shen, and H. Li, "Rapid Face Recognition Using Hashing," in *Conf. on Computer Vision and Pattern Recognition*, pp. 2753–2760, IEEE, 2010.

[22] Q. Shi, A. Eriksson, A. van den Hengel, and C. Shen, "Is Face Recognition Really a Compressive Sensing Problem?," in *Conf. on Computer Vision and Pattern Recognition*, pp. 553–560, IEEE, 2011.

[23] L. Zhang, M. Yang, and X. Feng, "Sparse Representation or Collaborative Representation: Which Helps Face Recognition?," in *Int'l. Conf. on Computer Vision*, pp. 471–478, IEEE, 2011.

[24] L. Wolf, T. Hassner, and Y. Taigman, "Effective Unconstrained Face Recognition by Combining Multiple Descriptors and Learned Background Statistics," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1978–1990, 2011.

[25] J. R. del Solar, R. Verschae, and M. Correa, "Recognition of Faces in Unconstrained Environments: A Comparative Study," *Journal on Adv. in Signal Processing*, vol. 2009, pp. 1–19, 2009.

[26] I. Naseem, R. Togneri, and M. Bennamoun, "Linear Regression for Face Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 2106–2112, 2010.

[27] C. Li, J. Guo, and H. Zhang, "Local Sparse Representation Based Classification," in *Int'l. Conf. on Pattern Recognition.*, pp. 649–652, 2010.

[28] Z. Nan and Y. Jian, "K Nearest Neighbor Based Local Sparse Representation Classifier," in *Chinese Conf. on Pattern Recognition*, pp. 1–5, IEEE, 2010.

[29] C. Chan and J. Kittler, "Sparse Representation of (Multiscale) Histograms for Face Recognition Robust to Registration and Illumination Problems," in *Int'l. Conf. on Image Processing*, pp. 2441–2444, IEEE, 2010.

[30] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *ELSEVIER Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.

[31] 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, and S. Weimer, "An Introduction to the Good, the Bad, & the Ugly Face Recognition Challenge Problem," in *Automatic Face & Gesture Recognition*, pp. 346–353, IEEE, 2011.

[32] E. Bailly-Baillire, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Marithoz, J. Matas, K. Messer, V. Popovici, F. Pore, B. Ruiz, and J.-P. Thiran, "The BANCA Database and Evaluation Protocol," in *AVBPA*, vol. 2688 of *Lecture Notes in Computer Science*, pp. 625–638, Springer, 2003.

[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, and N. Mavity, "Face Verification Competition on the XM2VTS Database," in *AVBPA*, vol. 2688 of *Lecture Notes in Computer Science*, pp. 964–974, Springer, 2003.

[34] W. Scheirer, A. Rocha, A. Sapkota, and T. Boult, "Towards Open Set Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 99, pp. 1757–1772, 2012.

[35] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust Face Recognition via Sparse Representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[36] M. Everingham, J. Sivic, and A. Zisserman, "Taking the Bite Out of Automated Naming of Characters in TV Video," *ELSEVIER Image and Vision Computing*, vol. 27, no. 5, pp. 545–559, 2009.

[37] M. Tapaswi and M. Bäuml, "'Knock! Knock! Who is it?' Probabilistic Person Identification in TV-Series," in *Conf. on Computer Vision and Pattern Recognition*, pp. 2658–2665, IEEE, 2012.

[38] W. Zhao, R. Chellappa, J. Phillips, and A. Rosenfeld, "Face Recognition in Still and Video Images: A Literature Survey," *CSUR*, vol. 35, no. 4, pp. 399–458, 2003.

[39] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino, "2D and 3D Face Recognition: A Survey," *Pattern Recognition Letters*, vol. 28, no. 14, pp. 1885–1906, 2007.

[40] C. Shan, "Face Recognition and Retrieval in Video," in *Video Search and Mining*, vol. 287 of *Studies in Computational Intelligence*, pp. 235–260, Springer, 2010.

[41] X. Yuan and S. Yan, "Visual Classification with Multi-task Joint Sparse Representation," in *Conf. on Computer Vision and Pattern Recognition*, pp. 3493–3500, IEEE, 2010.

[42] Q. Yin, X. Tang, and J. Sun, "An associate-predict model for face recognition," in *Conf. on Computer Vision and Pattern Recognition*, pp. 497–504, IEEE, 2011.

[43] P. Grother and P. Phillips, "Models of Large Population Recognition Performance," in *Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 68–75, IEEE, 2004.

[44] F. Li and H. Wechsler, "Open Set Face Recognition Using Transduction," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1686–1697, 2005.

[45] H. Ekenel, L. Szasz-Toth, and R. Stiefelhagen, "Open-set face recognition-based visitor interface system," in *Computer Vision Systems*, vol. 5815 of *Lecture Notes in Computer Science*, pp. 43–52, Springer, 2009.

[46] H. Gao, H. Ekenel, and R. Stiefelhagen, "Robust Open-Set Face Recognition for Small-Scale Convenience Applications," in *Pattern Recognition*, vol. 6376 of *Lecture Notes in Computer Science*, pp. 393–402, Springer, 2010.

[47] P. J. Phillips, H. Moon, S. Rizvi, and P. J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[48] G. Huang, V. Jain, and E. Learned-Miller, "Unsupervised Joint Alignment of Complex Images," in *Int'l. Conf. on Computer Vision*, pp. 1–8, IEEE, 2007.

[49] S. Maji and J. Malik, "Fast and Accurate Digit Classification," tech. rep., EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-159, 2009.

[50] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-Scale Image Classification: Fast Feature Extraction and SVM Training," in *Conf. on Computer Vision and Pattern Recognition*, pp. 1689–1696, IEEE, 2011.

[51] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "LIBLINEAR: A Library for Large Linear Classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[52] S. S. Chen, D. L. Donoho, Michael, and A. Saunders, "Atomic Decomposition by Basis Pursuit," *Journal on Sci. Comp.*, vol. 20, no. 1, pp. 33–61, 1998.

[53] E. J. Candes, J. K. Romberg, and T. Tao, "Stable Signal Recovery from Incomplete and Inaccurate Measurements," *Wiley Comm. on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[54] M. Figueiredo, R. Nowak, and S. Wright, "Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2008.

[55] D. Malioutov, M. Cetin, and A. Willsky, "Homotopy Continuation for Sparse Signal Representation," in *Int'l Conf. on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 733–736, IEEE, 2005.

[56] J. Yang and Y. Zhang, "Alternating Direction Algorithms for l1-Problems in Compressive Sensing," *Tech. Rep., Rice University*, 2009.

[57] J. A. Tropp, "Greed is Good: Algorithmic Results for Sparse Approximation," *Trans. on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[58] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust Alignment by Sparse and Low-Rank Decomposition for Linearly Correlated Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, 2012.

[59] K. Wu, L. Wang, F. Soong, and Y. Yam, "A Sparse and Low-rank Approach to Efficient Face Alignment for Photo-Real Talking Head Synthesis," in *Int'l Conf. on Acoustics, Speech, and Signal Processing*, pp. 1397–1400, IEEE, 2011.

[60] V. M. Patel, T. Wu, S. Biswas, P. J. Phillips, and R. Chellappa, "Dictionary-Based Face Recognition Under Variable Lighting and Pose," *IEEE Trans. on Information Forensics and Security*, vol. 7, pp. 954–965, 2012.

[61] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face Description with Local Binary Patterns: Application to Face Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2006.

[62] L. E. Ghaoui, V. Viallon, and T. Rabbani, "Safe Feature Elimination in Sparse Supervised Learning," *arXiv*, vol. 1009.4219, 2010.

[63] Z. Xiang, H. Xu, and P. Ramadge, "Learning Sparse Representations of High dimensional Data on Large Scale Dictionaries," in *Neural Information Processing Systems*, 2011.

[64] H. Xu, C. Caramanis, and S. Mannor, "Sparse Algorithms are not Stable: A No-free-lunch Theorem," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 187–193, 2012.

[65] P. Phillips, P. Flynn, J. Beveridge, W. Scruggs, A. O?Toole, D. Bolme, K. Bowyer, B. Draper, G. Givens, Y. Lui, H. Sahibzada, I. Scallan, JosephA., and S. Weimer, "Overview of the Multiple Biometrics Grand Challenge," in *Advances in Biometrics*, vol. 5558 of *Lecture Notes in Computer Science*, pp. 705–714, Springer, 2009.

[66] K.-C. Lee and D. Kriegman, "Online Learning of Probabilistic Appearance Manifolds for Video-based Recognition and Tracking," in *Conf. on Computer Vision and Pattern Recognition*, pp. 852–859, IEEE, 2005.

[67] R. G. Cinbis, J. Verbeek, and C. Schmid, "Unsupervised Metric Learning for Face Identification in TV Video," in *Int'l. Conf. on Computer Vision*, pp. 1559–1566, IEEE, 2011.

[68] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley, "Face Tracking and Recognition with Visual Constraints in Real-World Videos," in *Conf. on Computer Vision and Pattern Recognition*, IEEE, 2008.

[69] M. Bäuml, M. Tapaswi, and R. Stiefelhagen, "Semi-Supervised Learning with Constraints for Person Identification in Multimedia Data," in *Conf. on Computer Vision and Pattern Recognition*, pp. 3602–3609, IEEE, 2013.

[70] R. Goh, L. Liu, X. Liu, and T. Chen, "The CMU Face In Action (FIA) Database," in *Analysis and Modelling of Faces and Gestures*, vol. 3723 of *Lecture Notes in Computer Science*, pp. 255–263, Springer, 2005.

[71] C. Sanderson and K. K. Paliwal, "Identity verification using speech and face information," *Digital Signal Processing*, vol. 14, no. 5, pp. 449–480, 2004.

[72] N. Troje and H. H. Blthoff, "Face Recognition Under Varying Poses: The Role of Texture and Shape," *Vision Research*, vol. 36, pp. 1761–1771, 1996.

[73] S. Berrani and C. Garcia, "Enhancing Face Recognition from Video Sequences Using Robust Statistics," in *Advanced Video and Signal Based Surveillance*, pp. 324–329, IEEE, 2005.

[74] M. Zhao, J. Yagnik, H. Adam, and D. Bau, "Large Scale Learning and Recognition of Faces in Web Videos," in *Automatic Face & Gesture Recognition*, pp. 1–7, IEEE, 2008.

[75] Y.-C. Chen, V. M. Patel, P. J. Phillips, and R. Chellappa, "Dictionary-based Face Recognition from Video," in *European Conf. on Computer Vision*, pp. 766–779, IEEE, 2012.

[76] A. Hadid and M. Pietikainen, "From Still Image to Video-based Face Recognition: An Experimental Analysis," *Automatic Face & Gesture Recognition*, pp. 813–818, 2004.

[77] S. Zhou, V. Krueger, and R. Chellappa, "Probabilistic Recognition of Human Faces from Video," *ELSEVIER Journal on Computer Vision and Image Understanding*, vol. 91, no. 1–2, pp. 214–245, 2003.

[78] O. Yamaguchi, K. Fukui, and K. Maeda, "Face Recognition Using Temporal Image Sequence," in *Automatic Face & Gesture Recognition*, pp. 318–323, IEEE, 1998.

[79] K. Lee, J. Ho, and D. Kriegman, "Acquiring Linear Subspaces for Face Recognition under Variable Lighting," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[80] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic, "Finding Actors and Actions in Movies," in *Int'l. Conf. on Computer Vision*, IEEE, 2013.

[81] O. Arandjelovic and A. Zisserman, "Automatic Face Recognition for Film Character Retrieval in Feature-Length Films," in *Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 860–867, IEEE, 2005.

[82] O. Arandjelovic and R. Cipolla, "Automatic Cast Listing in Feature-Length Films with Anisotropic Manifold Space," in *Conf. on Computer Vision and Pattern Recognition*, pp. 1513–1520, IEEE, 2006.

[83] A. C. Gallagher and T. Chen, "Using a Markov Network to Recognize People in Consumer Images," in *Int'l. Conf. on Image Processing*, vol. 4, pp. 489–492, IEEE, 2007.

[84] A. Kapoor, G. Hua, A. Akbarzadeh, and S. Baker, "Which Faces to Tag: Adding Prior Constraints into Active Learning," in *Int'l. Conf. on Computer Vision*, pp. 1058–1065, IEEE, 2009.

[85] D. Lin, A. Kapoor, G. Hua, and S. Baker, "Joint People, Event, and Location Recognition in Personal Photo Collections Using Cross-Domain Context," in *European Conf. on Computer Vision*, 2010.

[86] F. IIS, "SHORE," 2010.

[87] C. Kueblbeck and A. Ernst, "Face Detection and Tracking in Video Sequences Using the Modified Census Transformation," *ELSEVIER Image and Vision Computing*, vol. 24, no. 6, pp. 564–572, 2006.

[88] M. Everingham, J. Sivic, and A. Zisserman, "Hello! My Name Is... Buffy–Automatic Naming of Characters in TV Video," in *British Machine Vision Conference*, 2006.

[89] A. Torralba and A. Efros, "Unbiased Look at Dataset Bias," in *Conf. on Computer Vision and Pattern Recognition*, pp. 1521–1528, IEEE, 2011.

[90] C. Liu and H. Wechsler, "Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition," *IEEE Trans. on Image Processing*, vol. 11, no. 4, pp. 467–476, 2002.

[91] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, IEEE, 2005.

[92] H. Zhang, A. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition," in *Conf. on Computer Vision and Pattern Recognition*, pp. 2126–2136, IEEE, 2006.

[93] A. Yang, A. Ganesh, Z. Zhou, S. Sastry, and Y. Ma, "Fast l1-Minimization Algorithms and an Application in Robust Face Recognition: A Review," in *Int'l. Conf. on Image Processing*, pp. 1849–1852, IEEE, 2010.

[94] E. Candes and J. Romberg, "l1-magic: A Collection of MATLAB Routines for Solving the Convex Optimization Programs Central to Compressive Sampling," *www.acm.caltech.edu/l1magic/*, 2006.

[95] S. Kim, K. Koh, M. Lustig, and S. Boyd, "An Efficient Method for Compressed Sensing," in *Int'l. Conf. on Image Processing*, pp. 117–120, IEEE, 2007.

[96] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *IEEE Trans. on Intelligence Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[97] R. Kasturi, D. Goldgof, Padmanabhan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 319–336, 2009.

[98] H. Guo, R. Wang, J. Choi, and L. S. Davis, "Face verification using sparse representations," in *Conf. on Computer Vision and Pattern Recognition Workshop*, pp. 37–44, IEEE, 2012.

[99] R. Wang and X. Chen, "Manifold Discriminant Analysis," in *Conf. on Computer Vision and Pattern Recognition*, pp. 429–436, IEEE, 2009.

[100] Y. Hu, A. S. Mian, and R. Owens, "Sparse Approximated Nearest Points for Image Set Classification," in *Conf. on Computer Vision and Pattern Recognition*, pp. 121–128, IEEE, 2011.

[101] R. Wang, H. Guo, L. S. Davis, and Q. Dai, "Covariance Discriminative Learning: A Natural and Efficient Approach to Image Set Classification," in *Conf. on Computer Vision and Pattern Recognition*, pp. 2496–2503, IEEE, 2012.

[102] Z. Cui, S. Shan, H. Zhang, S. Lao, and X. Chen, "Image Sets Alignment for Video-Based Face Recognition," in *Conf. on Computer Vision and Pattern Recognition*, pp. 2626–2633, IEEE, 2012.

[103] R. L. Cilibrasi and P. M. B. Vitanyi, "The google similarity distance," *IEEE Trans. on Knowledge and Data Engineering*, 2007.

[104] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang, "Tag Ranking," in *Int'l Conf. on World Wide Web*, pp. 351–360, ACM, 2009.

[105] F. Wang and C. Zhang, "Label Propagation Through Linear Neighborhoods," *IEEE Trans. on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, 2008.

[106] Z. Jiang, Z. Lin, and L. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," in *Conf. on Computer Vision and Pattern Recognition*, pp. 1697–1704, IEEE, 2011.

[107] S. Kong and D. Wang, "A dictionary learning approach for classification: Separating the particularity and the commonality," in *European Conf. on Computer Vision*, vol. 7572 of *Lecture Notes in Computer Science*, pp. 186–199, Springer, 2012.

[108] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[109] L. Duan, D. Xu, and S.-F. Chang, "Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach," in *Conf. on Computer Vision and Pattern Recognition*, pp. 1338–1345, IEEE, 2012.