J. Vis. Commun. Image R. 36 (2016) 142–148

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Rate-energy-accuracy optimization of convolutional architectures for face recognition ☆

L. Bondi [a], L. Baroffio [a,*], M. Cesana [a], M. Tagliasacchi [a], G. Chiachia [b], A. Rocha [b]

[a] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo Da Vinci, 32, 20133 Milan, Italy
[b] Reasoning for Complex Data (RECOD) Lab., Institute of Computing, University of Campinas (UNICAMP), Campinas, SP, Brazil

ARTICLE INFO

Article history:
Received 30 January 2015
Accepted 22 December 2015
Available online 3 February 2016

Keywords:
Convolutional architectures
Convolutional Neural Networks (CNNs)
Optimization
Coding
Face recognition
Analyze-then-Compress (ATC)
Deep learning
Deep neural networks

ABSTRACT

Face recognition systems based on Convolutional Neural Networks (CNNs) or convolutional architectures currently represent the state of the art, achieving an accuracy comparable to that of humans. Nonetheless, there are two issues that might hinder their adoption on distributed battery-operated devices (e.g., visual sensor nodes, smartphones, and wearable devices). First, convolutional architectures are usually computationally demanding, especially when the depth of the network is increased to maximize accuracy. Second, transmitting the output features produced by a CNN might require a bitrate higher than the one needed for coding the input image. Therefore, in this paper we address the problem of optimizing the energy-rate-accuracy characteristics of a convolutional architecture for face recognition. We carefully profile a CNN implementation on a Raspberry Pi device and optimize the structure of the neural network, achieving a 17-fold speedup without significantly affecting recognition accuracy. Moreover, we propose a coding architecture custom-tailored to features extracted by such model.

1. Introduction

Humans are able to identify and recognize a face, automatically assigning it to a given person, with just a few glances. Our brain processes visual stimuli and stores a concise representation of a face in our memory. It is able to correctly distinguish between up to tens of thousands of different faces, although it might not be able to recall the name of the person they correspond to [22]. Furthermore, our brain is capable of recognizing instances of the same face acquired under different conditions (e.g. point of view, illumination, aging).

The problem of automatically detecting and recognizing faces has received significant attention in the past fifty years, with the goal of matching the capabilities of human vision. The first attempts date back to the end of the 1960s, with the models proposed by Bledsoe [3] and Kanade [13]. Such early efforts strived for modeling a face in terms of fiducial points and their relationships, and are quite fragile to changes in imaging conditions. More recently, such approaches have been outperformed by more effective models that achieve better recognition accuracy while requiring low computational resources. Sirovich and Kirby propose Eigenfaces [23], a compact yet effective representation of human faces. Turk and Pentland extended upon such a model to build a computationally efficient face detection and recognition system [25]. Face recognition and verification systems based on aligned images acquired in controlled environments have made great strides in the last twenty years, being able to reduce the error rate by three orders of magnitude [18]. Most current approaches achieve state-of-the-art performance by exploring rich representations of the underlying visual content that consists of up to tens of thousands handcrafted features [1,4].

In the last few years, following a trend also present in several image classification tasks, Convolutional Neural Networks (CNNs) and Deep Learning techniques have been applied to large-scale face verification and recognition systems as well [9,19]. Recently, Taigman et al. proposed DeepFace [24], a deep CNN that has been proven to perform on a par with humans in terms of face recognition accuracy. Such a system exploits a large amount of heterogeneous training data to learn discriminative low-dimensional representations of faces. Likewise, Chiachia et al. [5] have used CNNs to address the problem of familiar face recognition by explicitly learning enhanced person-specific face representations from large amounts of experience with the appearance of individuals.

☆ This paper has been recommended for acceptance by Yehoshua Zeevi.

* Corresponding author.
E-mail addresses: luca.bondi@polimi.it (L. Bondi), luca.baroffio@polimi.it (L. Baroffio), matteo.cesana@polimi.it (M. Cesana), marco.taglisacchi@polimi.it (M. Tagliasacchi), chiachia@ic.unicamp.br (G. Chiachia), rocha@ic.unicamp.br (A. Rocha).

http://dx.doi.org/10.1016/j.jvcir.2015.12.015
1047-3203/

CNNs usually require a large amount of training data and computational resources to be effectively trained. Nonetheless, convolutional architectures based on random weights have been shown to provide good results in terms of accuracy for the task of face recognition and matching.

Effective technologies for automatic face recognition enable a large number of services, such as automated surveillance systems, person authentication, image tagging, and advanced human–computer interaction. Such tasks are often performed in a distributed fashion on battery-operated low-power devices such as mobile phones, smart cameras or nodes of a Visual Sensor Network (VSN). The problem of performing distributed visual analysis tasks on low-power devices has been extensively addressed. The traditional approach to such kind of tasks, hereinafter denoted as *Compress-Then-Analyze* (CTA) [21], consists in the acquisition of visual content on a low-power node, its compression by means of image (e.g., JPEG) or video (e.g., H.264/AVC) coding primitives, and its transmission to a central node, where processing takes place.

Recently, the novel *Analyze-Then-Compress* (ATC) paradigm has been proposed [21,20,2,17]. In this paradigm, low-power nodes acquire visual content and extract higher-level information from it by means of efficient feature extraction algorithms. Such content is then compressed resorting to ad hoc coding primitives and transmitted to a central node, that performs visual analysis. Such approach has been proven to achieve good results in terms of rate-energy-accuracy performance for a number of tasks including content-based retrieval [21,20], object tracking [2] and mobile augmented reality [17]. Moving part of the computational burden, that is, detecting faces and computing compact yet representative features, from a centralized node to sensing devices produces several positive effects:

1. it fairly distributes computational complexity over network nodes, without requiring power-eager central units to serve a high number of clients;
2. it requires less bandwidth than transmitting images or video sequences; and
3. it reduces the risk of privacy violations, avoiding the transmission of the pixel-level visual content.

In this paper, we aim at investigating the adoption of convolutional architectures in distributed battery-operated devices, with the objective of enabling the *Analyze-Then-Compress* paradigm for face identification/recognition.

According to [10–12,16], there are two main problems in biometrics: verification and identification. Depending on the application context, a biometric system may operate either in verification mode or in identification/recognition mode.

In the verification mode, we have a biometric system which validates a person's identity by comparing the acquired biometric sample with his own biometric template previously stored in the system database. In this setup, an individual who wants to be recognized claims an identity (e.g., "Bob"), normally using a PIN, a user name, or even a smart card, and the system retrieves Bob's biometric sample from the database and conducts a 1:1 comparison of the retrieved model to the one just acquired from the person claiming to be "Bob" to determine whether the claim is true or not (e.g., "Is this sample really from Bob?").

In turn, in the identification/recognition mode, we have a system which recognizes an individual by searching the templates of all the users in the database for a match. Therefore, in this setup, we have a 1:*N* comparison to establish an individual's identity (or fail to do so if the subject is not enrolled in the system database) without the subject having to claim any identity (e.g., "Whose biometric data is this?").

In this work, therefore, we focus our efforts on the identification/recognition mode in which we have a face sample acquired by a distributed battery-operated device and need to compare it to several other instances (faces) in a server when trying to recognize a given person. In this operational mode, it is a pre-requisite to have some individuals enrolled into the system, otherwise it is not possible to perform any recognition. Posing this problem in the context of machine learning classification, the database or gallery of enrolled individuals represents the training data to which we need to compare an unseen test sample when performing recognition. Here, if "Bob" is enrolled in the system, it means that the system has been fed with some of his face samples. When he steps in for recognition, the system collects a new face sample and compares it to all existing user face samples in the system database when performing recognition.

The adoption of complex convolutional models in the context of face recognition requires to address two issues, i.e., (i) the high demand of computational resources; and (ii) the apparent data expansion introduced by such architectures. In particular, we empirically evaluate the computational requirements of the feature extraction process, investigating the impact of all hyperparameters. We perform the experiments on a low-power ARM-based Raspberry Pi computer taking the convolutional architecture proposed by Pinto et al. [19] as a reference. Furthermore, we optimize the CNN considering two main aspects: first, we propose an energy-efficient yet effective convolutional network, custom-tailored to the computational resources of the available hardware; second, we introduce ad hoc feature coding primitives aimed at significantly reducing the output bitrate of such a model.

We organized the remaining of this paper into four sections. Section 2 gives an overview of Convolutional Neural Networks. Section 3 introduces an optimized convolutional architecture that dramatically speeds up feature extraction. Section 4 describes the proposed architecture and a rate-accuracy comparison between ATC vs. CTA. Finally, Section 5 presents the conclusions and future research directions.

2. Background on Convolutional Neural Networks (CNNs) and convolutional architectures

With the "big data" revolution, many applications including computer vision, speech/audio recognition, social networks, among others, are dealing with larger and larger amounts of data. At the same time, the advent of powerful, parallel and scalable computing architectures is enabling more complex and effective statistical and computational models. In this context, large neural networks are getting constantly increasing attention and achieve outstanding results in terms of task accuracy for a number of heterogeneous applications [14,24,7].

In particular, in the context of computer vision, Convolutional Neural Networks (CNNs) have been proposed as effective variants of MultiLayer Perceptrons, inspired by human visual cortex [15] with very heterogeneous applications [14,24]. They combine three main features to achieve invariance with respect to imaging conditions: local receptive fields, shared weights, and spatial pooling. Fig. 1 depicts a block diagram of the 3-layer architecture proposed by Pinto et al. [19] as an example. Each layer of such model comprises sublayers of neurons that perform filtering, thresholding, spatial pooling, and normalization. These linear and non-linear operations generate increasingly more complex feature maps, ultimately outputting a feature-based representation that is fed to a classifier for prediction. Traditional CNNs require a computationally expensive training stage to be performed on a large amount of training data in order to learn the filter weights. Instead, the peculiarity of the convolutional architecture proposed by Pinto