Skin Melanoma Classification Using Deep Convolutional Neural Networks

Chapter · March 2020

DOI: 10.1201/9781351003827-11

3 authors:

Khalid Hosny
Zagazig University

Mohamed A. Kassem
Kafr El-Sheikh University

M. M. Fouad
Alexandria Higher Institute of Engineering & Technology

Deep Learning in Computer Vision

Digital Imaging and Computer Vision Series

Series Editor

Rastislav Lukac

Foveon, Inc./Sigma Corporation San Jose, California, U.S.A.

Dermoscopy Image Analysis
by M. Emre Celebi, Teresa Mendonça, and Jorge S. Marques

Semantic Multimedia Analysis and Processing
by Evaggelos Spyrou, Dimitris Iakovidis, and Phivos Mylonas

Microarray Image and Data Analysis: Theory and Practice
by Luis Rueda

Perceptual Digital Imaging: Methods and Applications
by Rastislav Lukac

Image Restoration: Fundamentals and Advances
by Bahadir Kursat Gunturk and Xin Li

Image Processing and Analysis with Graphs: Theory and Practice
by Olivier Lézoray and Leo Grady

Visual Cryptography and Secret Image Sharing
by Stelvio Cimato and Ching-Nung Yang

Digital Imaging for Cultural Heritage Preservation: Analysis, Restoration, and Reconstruction of Ancient Artworks
by Filippo Stanco, Sebastiano Battiato, and Giovanni Gallo

Computational Photography: Methods and Applications
by Rastislav Lukac

Super-Resolution Imaging
by Peyman Milanfar

Deep Learning in Computer Vision

Principles and Applications

Edited by
Mahmoud Hassaballah and Ali Ismail Awad

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper

International Standard Book Number-13: 978-1-138-54442-0 (Hardback)

Library of Congress Cataloging-in-Publication Data

Names: Hassaballah, Mahmoud, editor. | Awad, Ali Ismail, editor.
Title: Deep learning in computer vision : principles and applications / edited by M. Hassaballah and Ali Ismail Awad.
Description: First edition. | Boca Raton, FL : CRC Press/Taylor and Francis, 2020. | Series: Digital imaging and computer vision | Includes bibliographical references and index.
Identifiers: LCCN 2019057832 (print) | LCCN 2019057833 (ebook) | ISBN 9781138544420 (hardback ; acid-free paper) | ISBN 9781351003827 (ebook)
Subjects: LCSH: Computer vision. | Machine learning.
Classification: LCC TA1634 .D437 2020 (print) | LCC TA1634 (ebook) | DDC 006.3/7--dc23
LC record available at https://lccn.loc.gov/2019057832
LC ebook record available at https://lccn.loc.gov/2019057833

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Foreword

Deep learning, while it has multiple definitions in the literature, can be defined as "inference of model parameters for decision making in a process mimicking the understanding process in the human brain"; or, in short: "brain-like model identification". We can say that deep learning is a way of data inference in machine learning, and the two together are among the main tools of modern artificial intelligence. Novel technologies away from traditional academic research have fueled R&D in convolutional neural networks (CNNs); companies like Google, Microsoft, and Facebook ignited the "art" of data manipulation, and the term "deep learning" became almost synonymous with decision making.

Various CNN structures have been introduced and invoked in many computer vision-related applications, with greatest success in face recognition, autonomous driving, and text processing. The reality is: deep learning is an art, not a science. This state of affairs will remain until its developers develop the theory behind its functionality, which would lead to "cracking its code" and explaining why it works, and how it can be structured as a function of the information gained with data. In fact, with deep learning, there is good and bad news. The good news is that the industry—not necessarily academia—has adopted it and is pushing its envelope. The bad news is that the industry does not share its secrets. Indeed, industries are never interested in procedural and textbook-style descriptions of knowledge.

This book, *Deep Learning in Computer Vision: Principles and Applications*—as a journey in the progress made through deep learning by academia—confines itself to deep learning for computer vision, a domain that studies sensory information used by computers for decision making, and has had its impacts and drawbacks for nearly 60 years. Computer vision has been and continues to be a system: sensors, computer, analysis, decision making, and action. This system takes various forms and the flow of information within its components, not necessarily in tandem. The linkages between computer vision and machine learning, and between it and artificial intelligence, are very fuzzy, as is the linkage between computer vision and deep learning. Computer vision has moved forward, showing amazing progress in its short history. During the sixties and seventies, computer vision dealt mainly with capturing and interpreting optical data. In the eighties and nineties, geometric computer vision added science (geometry plus algorithms) to computer vision. During the first decade of the new millennium, modern computing contributed to the evolution of object modeling using multimodality and multiple imaging. By the end of that decade, a lot of data became available, and so the term "deep learning" crept into computer vision, as it did into machine learning, artificial intelligence, and other domains.

This book shows that traditional applications in computer vision can be solved through invoking deep learning. The applications addressed and described in the eleven different chapters have been selected in order to demonstrate the capabilities of deep learning algorithms to solve various issues in computer vision. The content of this book has been organized such that each chapter can be read independently of the others. Chapters of the book cover the following topics: accelerating the CNN inference on field-programmable gate arrays, fire detection in surveillance applications, face recognition, action and activity recognition, semantic segmentation for autonomous driving, aerial imagery registration, robot vision, tumor detection, and skin lesion segmentation as well as skin melanoma classification.

From the assortment of approaches and applications in the eleven chapters, the common thread is that deep learning for identification of CNN provides accuracy over traditional approaches. This accuracy is attributed to the flexibility of CNN and the availability of large data to enable identification through the deep learning strategy. I would expect that the content of this book to be welcomed worldwide by graduate and postgraduate students and workers in computer vision, including practitioners in academia and industry. Additionally, professionals who want to explore the advances in concepts and implementation of deep learning algorithms applied to computer vision may find in this book an excellent guide for such purpose. Finally, I hope that readers would find the presented chapters in the book interesting and inspiring to future research, from both theoretical and practical viewpoints, to spur further advances in discovering the secrets of deep learning.

Prof Aly Farag, PhD, Life Fellow, IEEE, Fellow, IAPR
Professor of Electrical and Computer Engineering
University of Louisville, Kentucky

Preface

Simply put, computer vision is an interdisciplinary field of artificial intelligence that aims to guide computers and machines toward understanding the contents of digital data (i.e., images or video). According to computer vision achievements, the future generation of computers may understand human actions, behaviors, and languages similarly to humans, carry out some missions on their behalf, or even communicate with them in an intelligent manner. One aspect of computer vision that makes it such an interesting topic of study and active research field is the amazing diversity of daily-life applications such as pedestrian protection systems, autonomous driving, biometric systems, the movie industry, driver assistance systems, video surveillance, and robotics as well as medical diagnostics and other healthcare applications. For instance, in healthcare, computer vision algorithms may assist healthcare professionals to precisely classify illnesses and cases; this can potentially save patients' lives through excluding inaccurate medical diagnoses and avoiding erroneous treatment. With this wide variety of applications, there is a significant overlap between computer vision and other fields such as machine vision and image processing. Scarcely a month passes where we do not hear from the research and industry communities with an announcement of some new technological breakthrough in the areas of intelligent systems related to the computer vision field.

With the recent rapid progress on deep convolutional neural networks, deep learning has achieved remarkable performance in various fields. In particular, it has brought a revolution to the computer vision community, introducing non-traditional and efficient solutions to several problems that had long remained unsolved. Due to this promising performance, it is gaining more and more attention and is being applied widely in computer vision for several tasks such as object detection and recognition, object segmentation, pedestrian detection, aerial imagery registration, video processing, scene classification, autonomous driving, and robot localization as well as medical image-related applications. If the phrase "deep learning for computer vision" is searched in Google, millions of search results will be obtained. Under these circumstances, a book entitled *Deep Learning in Computer Vision* that covers recent progress and achievements in utilizing deep learning for computer vision tasks will be extremely useful.

The purpose of this contributed volume is to fill the existing gap in the literature for the applications of deep learning in computer vision and to provide a bird's eye view of recent state-of-the-art models designed for practical problems in computer vision. The book presents a collection of eleven high-quality chapters written by renowned experts in the field. Each chapter provides the principles and fundamentals of a specific topic, introduces reviews of up-to-date techniques, presents outcomes, and points out challenges and future directions. In each chapter, figures, tables, and examples are used to improve the presentation and analysis of covered topics. Furthermore, bibliographic references are included in each chapter, providing a good starting point for deeper research and further exploration of the topics considered in this book. Further, this book is structured such that each chapter can be read independently from the others as follows:

Chapter 1 presents a state-of-the-art of CNN inference accelerators over FPGAs. Computational workloads, parallelism opportunities, and the involved memory accesses are analyzed. At the level of neurons, optimizations of the convolutional and fully connected layers are explained and the performances of the different methods compared, while at the network level, approximate computing and data-path optimization methods are covered and state-of-the-art approaches compared. The methods and tools investigated in this chapter represent the recent trends in FPGA CNN inference accelerators and will fuel future advances in efficient hardware deep learning.

Chapter 2 concentrates on object detection problem using deep CNN (DCNN): the recent developments of several classical CNN-based object detectors are discussed. These detectors significantly improve detection performance either through employing new architectures or through solving practical issues like degradation, gradient vanishing, and class imbalance. Detailed background information is provided to show the progress and improvements of different models. Some evaluation results and comparisons are reported on three datasets with distinctive characteristics.

Chapter 3 proposes three methods for fire detection using CNNs. The first method focuses on early fire detection with an adaptive prioritization mechanism for surveillance cameras. The second CNN-assisted method improves fire detection accuracy with a main focus on reducing false alarms. The third method uses an efficient deep CNN for fire detection. For localization of fire regions, a feature map selection algorithm that intelligently selects appropriate feature maps sensitive to fire areas is proposed.

Chapter 4 presents an accurate and real-time multi-biometric system for identifying a person's identity using a combination of two discriminative deep learning approaches to address the problem of unconstrained face recognition: CNN and deep belief network (DBN). The proposed system is tested on four large-scale challenging datasets with high diversity in the facial expressions—SDUMLA-HMT, FRGC V 2.0, UFI, and LFW—and new state-of-the-art recognition rates on all the employed datasets are achieved.

Chapter 5 introduces a study of the concept of sequence learning using RNN, LSTM, and its variants such as multilayer LSTM and bidirectional LSTM for action and activity recognition problems. The chapter concludes with major issues of sequence learning for action and activity recognition and highlights recommendations for future research.

Chapter 6 discuses semantic segmentation in autonomous driving applications, where it focuses on constructing efficient and simple architectures to demonstrate the benefit of flow and depth augmentation to CNN-based semantic segmentation networks. The impact of both motion and depth information on semantic segmentation is experimentally studied using four simple network architectures. Results of experiments on two public datasets—Virtual-KITTI and CityScapes—show reasonable improvement in overall accuracy.

Chapter 7 presents a method based on deep learning for geolocalizing drones using only onboard cameras. A pipeline has been implemented that makes use of the availability of satellite imagery and traditional computer vision feature detectors and descriptors, along with renowned deep learning methods (semantic segmentation), to be able to locate the aerial image captured from the drone within the satellite imagery. The method enables the drone to be autonomously aware of its surroundings and navigate without using GPS.

Chapter 8 is intended to be a guide for the developers of robot vision systems, focusing on the practical aspects of the use of deep neural networks rather than on theoretical issues.

The last three chapters are devoted to deep learning in medical applications. Chapter 9 covers basic information about CNNs in medical applications. CNN developments are discussed from different perspectives, specifically, CNN design, activation function, loss function, regularization, optimization, normalization, and network depth. Also, a deep convolutional neural network (DCNN) is designed for brain tumor detection using MRI images. The proposed DCNN architecture is evaluated on the RIDER dataset, achieving accurate detection accuracy within a time of 0.24 seconds per MRI image.

Chapter 10 discusses automatic segmentation of skin lesion boundaries from surrounding tissue and presents a novel deep learning segmentation methodology via full-resolution convolutional network (FrCN). Experimental results show the great promise of the FrCN method compared to state-of-the-art deep learning segmentation approaches such as fully convolutional networks (FCN), U-Net, and SegNet with overall segmentation.

Chapter 11 is about the automatic classification of color skin images, where a highly accurate method is proposed for skin melanoma classification utilizing two modified deep convolutional neural networks and consisting of three main steps. The proposed method is tested using the well-known MED-NODE and DermIS & DermQuest datasets.

It is very necessary to mention here that the book is a small piece in the puzzle of computer vision and its applications. We hope that our readers find the presented chapters in the book interesting and that the chapters will inspire future research both from theoretical and practical viewpoints to spur further advances in the computer vision field.

The editors would like to take this opportunity to express their sincere gratitude to the contributors for extending their wholehearted support in sharing some of their latest results and findings. Without their significant contribution, this book could not have fulfilled its mission. The reviewers deserve our thanks for their constructive and timely input. Special profound thanks go to Prof Aly Farag, Professor of Electrical and Computer Engineering, University of Louisville, Kentucky for writing the Foreword for this book. Finally, the editors acknowledge the efforts of the CRC Press Taylor & Francis for giving us the opportunity to edit a book on deep learning for computer vision. In particular, we would like to thank Dr Rastislav Lukac, the editor of the Digital Imaging and Computer Vision book series, and Nora Konopka for initiating this project. Really, the editorial staff at CRC Press has done a meticulous job, and working with them was a pleasant experience.

Mahmoud Hassaballah
Qena, Egypt

Ali Ismail Awad
Luleå, Sweden

Editors Bio

Mahmoud Hassaballah was born in 1974, Qena, Egypt. He received his BSc degree in Mathematics in 1997 and his MSc degree in Computer Science in 2003, both from South Valley University, Egypt, and his Doctor of Engineering (D Eng) in computer science from Ehime University, Japan in 2011. He was a visiting scholar with the department of computer & communication science, Wakayama University, Japan in 2013 and GREAH laboratory, Le Havre Normandie University, France in 2019. He is currently an associate professor of computer science at the faculty of computers and information, South Valley University, Egypt. He served as a reviewer for several journals such as *IEEE Transactions on Image Processing, IEEE Transactions on Fuzzy Systems, Pattern Recognition, Pattern Recognition Letters, IET Image Processing, IET Computer Vision, IET Biometrics, Journal of Real-Time Image Processing, and Journal of Electronic Imaging*. He has published over 50 research papers in refereed international journals and conferences. His research interests include feature extraction, object detection/recognition, artificial intelligence, biometrics, image processing, computer vision, machine learning, and data hiding.

Ali Ismail Awad (SMIEEE, PhD, PhD, MSc, BSc) is currently an Associate Professor (Docent) with the Department of Computer Science, Electrical, and Space Engineering, Luleå University of Technology, Luleå, Sweden, where he also serves as a Coordinator of the Master Programme in Information Security. He is a Visiting Researcher with the University of Plymouth, United Kingdom. He is also an Associate Professor with the Electrical Engineering Department, Faculty of Engineering, Al-Azhar University at Qena, Qena, Egypt. His research interests include information security, Internet-of-Things security, image analysis with applications in biometrics and medical imaging, and network security. He has edited or co-edited five books and authored or co-authored several journal articles and conference papers in these areas. He is an Editorial Board Member of the following journals: *Future Generation Computer Systems*, *Computers & Security*, *Internet of Things: Engineering Cyber Physical Human Systems*, and *Health Information Science and Systems*. Dr Awad is currently an IEEE senior member.

Contributors

Ahmad El Sallab
Valeo Company
Cairo, Egypt

Ahmed Nassar
IRISA Institute
Rennes, France

Alaa S. Al-Waisy
University of Bradford
Bradford, UK

Ali Ismail Awad
Luleå University of Technology
Luleå, Sweden
and
Al-Azhar University
Qena, Egypt

Amin Ullah
Sejong University
Seoul, South Korea

Ashraf A. M. Khalaf
Minia University
Minia, Egypt

François Berry
University Clermont Auvergne
Clermont-Ferrand, France

Guanghui Wang
University of Kansas
Kansas City, Kansas

Hazem Rashed
Valeo Company
Cairo, Egypt

Hesham F.A. Hamed
Egyptian Russian University
Cairo, Egypt
and
Minia University
Minia, Egypt

Javier Ruiz-Del-Solar
University of Chile
Santiago, Chile

Kaidong Li
University of Kansas
Kansas City, Kansas

Kamel Abdelouahab
Clermont Auvergne University
Clermont-Ferrand, France

Khalid M. Hosny
Zagazig University
Zagazig, Egypt

Khan Muhammad
Sejong University
Seoul, South Korea

Mahmoud Hassaballah
South Valley University
Qena, Egypt

Mahmoud Khaled Abd-Ellah
Al-Madina Higher Institute for Engineering and Technology
Giza, Egypt

Maxime Pelcat
University of Rennes
Rennes, France

Miyoung Lee
Sejong University
Seoul, South Korea

Mohamed A. Kassem
Kafr El Sheikh University
Kafr El Sheikh, Egypt

Mohamed Elhelw
Nile University
Giza, Egypt

Mohamed M. Foaud
Zagazig University
Zagazig, Egypt

Mohammed A. Al-Masni
Kyung Hee University
Seoul, South Korea
and
Yonsei University
Seoul, South Korea

Mugahed A. Al-Antari
Kyung Hee University
Seoul, South Korea
and
Sana'a Community College
Sana'a, Republic of Yemen

Patricio Loncomilla
University of Chile
Santiago, Chile

Rami Qahwaji
University of Bradford
Bradford, UK

Salman Khan
Sejong University
Seoul, South Korea

Senthil Yogamani
Valeo Company
Galway, Ireland

Shumoos Al-Fahdawi
University of Bradford
Bradford, UK

Sung Wook Baik
Sejong University
Seoul, South Korea

Tae-Seong Kim
Kyung Hee University
Seoul, South Korea

Tanveer Hussain
Sejong University
Seoul, South Korea

Usman Sajid
University of Kansas
Kansas City, Kansas

Wenchi Ma
University of Kansas
Kansas City, Kansas

Yuanwei Wu
University of Kansas
Kansas City, Kansas

1 Accelerating the CNN Inference on FPGAs

Kamel Abdelouahab, Maxime Pelcat, and François Berry

CONTENTS

1.1 INTRODUCTION

The exponential growth of big data during the last decade motivates for innovative methods to extract high semantic information from raw sensor data such as videos, images, and speech sequences. Among the proposed methods, convolutional neural networks (CNNs) [1] have become the de facto standard by delivering near-human accuracy in many applications related to machine vision (e.g., classification [2], detection [3], segmentation [4]) and speech recognition [5].

This performance comes at the price of a large computational cost as CNNs require up to 38 GOPs to classify a single frame [6]. As a result, dedicated hardware is required to accelerate their execution. Graphics processing units GPUs are the most widely used platform to implement CNNs as they offer the best performance in terms of pure computational throughput, reaching up 11 TFLOPs [7]. Nevertheless, in terms of power consumption, field-programmable gate array (FPGA) solutions are known to be more energy efficient (vs. GPU). While GPU implementations have demonstrated state-of-the-art computational performance, CNN acceleration will soon be moving towards FPGAs for two reasons. First, recent improvements in FPGA technology put FPGA performance within striking distance of GPUs with a reported performance of 9.2 TFLOPs for the latter [8]. Second, recent trends in CNN development increase the sparsity of CNNs and use extremely compact data types. These trends favor FPGA devices, which are designed to handle irregular parallelism and custom data types. As a result, next-generation CNN accelerators are expected to deliver up to $5.4\times$ better computational throughput than GPUs [7].

As an inflection point in the development of CNN accelerators might be near, we conduct a survey on FPGA-based CNN accelerators. While a similar survey can be found in [9], we focus in this chapter on the recent techniques that were not covered in the previous works. In addition to this chapter, we refer the reader to the works of Venieris et al. [10], which review the toolflows automating the CNN mapping process, and to the works of Sze et al., which focus on ASICs for deep learning acceleration.

The amount and diversity of research on the subject of CNN FPGA acceleration within the last 3 years demonstrate the tremendous industrial and academic interest. This chapter presents a state-of-the-art review of CNN inference accelerators over FPGAs. The computational workloads, their parallelism, and the involved memory accesses are analyzed. At the level of neurons, optimizations of the convolutional and fully connected (FC) layers are explained and the performances of the different methods compared. At the network level, approximate computing and data-path optimization methods are covered and state-of-the-art approaches compared. The methods and tools investigated in this survey represent the recent trends in FPGA CNN inference accelerators and will fuel the future advances on efficient hardware deep learning.

1.2 BACKGROUND ON CNNS AND THEIR COMPUTATIONAL WORKLOAD

In this first section, we overview the main features of CNNs, mainly focusing on the computations and parallelism patterns involved during their inference.

1.2.1 General Overview

Deep* CNNs are feed-forward†, sparsely connected‡ neural networks. A typical CNN structure consists of a pipeline of layers. Each layer inputs a set of data, known as a feature map (FM), and produces a new set of FMs with *higher-level semantics*.

1.2.2 Inference versus Training

As typical machine learning algorithms, CNNs are deployed in two phases. First, the *training* stage works on a known set of annotated data samples to create a model with a *modeling* power (which semantics extrapolates to natural data outside the training set). This phase implements the *back-propagation* algorithm [11], which iteratively updates CNN parameters such as convolution weights to improve the predictive power of the model. A special case of CNN training is *fine-tuning*. When *fine-tuning* a model, weights of a previously trained network are used to initialize the parameters of a new training. These weights are then adjusted for a new constraint, such as a different dataset or a reduced precision.

The second phase, known as *inference*, uses the learned model to classify new data samples (i.e., inputs that were not previously seen by the model). In a typical setup, CNNs are trained/fine-tuned only once, on large clusters of GPUs. By contrast, the inference is implemented each time a new data sample has to be classified. As a consequence, the literature mostly focuses on accelerating the inference phase. As a result, our discussion overviews the main methods employed to accelerate the inference. Moreover, since most of the CNN accelerators benchmark their performance on models trained for image classification, we focus our chapter on this application. Nonetheless, the methods detailed in this survey can be employed to accelerate CNNs for other applications such object detection, image segmentation, and speech recognition.

1.2.3 Inference, Layers, and CNN Models

CNN inference refers to the *feed-forward* propagation of B input images across L layers. This section details the computations involved in the major types of these layers. A common practice is to manipulate layers, parameters, and FMs as multidimensional arrays, as listed in Table 1.1. Note that when it will be relevant, the type of the layer will be denoted with superscript, and the position of the layer will be denoted with subscript.

---

* Includes a large number of layer, typically above three.

† The information flows from the neurons of a layer ℓ towards the neurons of a layer. $\ell + 1$

‡ CNNs implement the weight sharing technique, applying a small number of weights across all the input pixels (i.e., image convolution).

TABLE 1.1
Tensors Involved in the Inference of a Given Layer ℓ with Their Dimensions

X	Input FMs	$B \times C \times H \times W$	B	Batch size (Number of input frames)
Y	Output FMs	$B \times N \times V \times U$	$W/H/C$	Width/Height/Depth of Input FMs
Θ	Learned Filters	$N \times C \times J \times K$	$U/V/N$	Width/Height/Depth of Output FMs
β	Learned biases	N	K/J	Horizontal/Vertical Kernel size

A convolutional layer (*conv*) carries out the feature extraction process by applying – as illustrated in Figure 1.1 – a set of three-dimensional convolution filters Θ^{conv} to a set of B input volumes $\mathbf{X}^{\text{conv}}$. Each input volume has a depth C and can be a color image (in the case of the first *conv* layer), or an output generated by previous layers in the network. Applying a three-dimensional filter to three-dimensional input results in a 2D *(FM)*. Thus, applying N three-dimensional filters in a layer results in a three-dimensional output with a depth N.

In some CNN models, a learned offset β^{conv} – called a *bias* – is added to processed feature maps. However, this practice has been discarded in recent models [6]. The computations involved in feed-forward propagation of *conv* layers are detailed in Equation 1.1.

$$\forall \{b,n,u,v\} \in [1,B] \times [1,N] \times [1,V] \times [1,U]$$

$$\mathbf{Y}^{\text{conv}}[b,n,v,u] = \beta^{\text{conv}}[n] + \sum_{c=1}^{C}\sum_{j=1}^{J}\sum_{k=1}^{K} \mathbf{X}^{\text{conv}}[b,c,v+j,u+k] \cdot \Theta^{\text{conv}}[n,c,j,k] \qquad (1.1)$$

One may note that applying a depth convolution to a 3D input boils down to applying a mainstream 2D convolution to each of the 2D channels of the input, then, at each point, summing the results across all the channels, as shown in Equation 1.2.

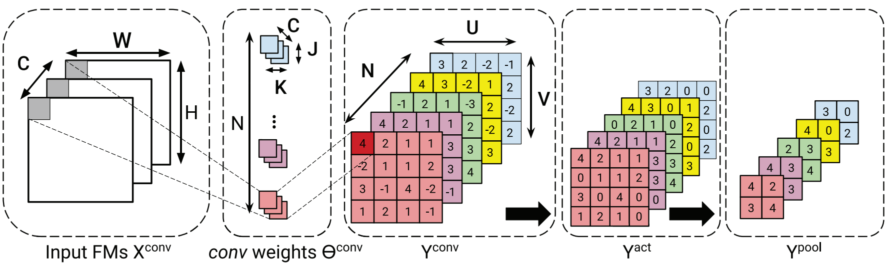


FIGURE 1.1 Feed-forward propagation in *conv*, *act*, and *pool* layers (batch size $B = 1$, bias β omitted).

$$\forall n \in [1, N]$$

$$\boldsymbol{Y}[\boldsymbol{n}]^{\text{conv}} = \beta^{\text{conv}}[n] + \sum_{c=1}^{C} \text{conv2D}\left(\mathbf{X}[\boldsymbol{c}]^{\text{conv}}, \Theta[\boldsymbol{c}]^{\text{conv}}\right) \tag{1.2}$$

Each *conv* layer of a CNN is usually followed by an activation layer that applies a *nonlinear* function to all the values of FMs. Early CNNs were trained with TanH or Sigmoid functions, but recent models employ the rectified linear unit (ReLU) function, which grants faster training times and less computational complexity, as highlighted in Krizhevsky et al. [12].

$$\forall \{b, n, u, v\} \in [1, B] \times [1, N] \times [1, V] \times [1, U]$$

$$\boldsymbol{Y}^{\text{act}}[b, n, h, w] = \text{act}(\boldsymbol{X}^{\text{act}}[b, n, h, w]) \mid \text{act:=TanH, Sigmoid, ReLU} \ldots \tag{1.3}$$

The convolutional and activation parts of a CNN are directly inspired by the cells of visual cortex in neuroscience [13]. This is also the case with *pooling* layers, which are periodically inserted in between successive *conv* layers. As shown in Equation 1.4, *pooling* sub-samples each channel of the input FM by selecting either the *average*, or, more commonly, the *maximum* of a given neighborhood $\boldsymbol{K}$. As a result, the dimensionality of an FM is reduced, as illustrated in Figure 1.1.

$$\forall \{b, n, u, v\} \in [1, B] \times [1, N] \times [1, V] \times [1, U]$$

$$\boldsymbol{Y}^{\text{pool}}[b, n, v, u] = \max_{p, q \in [1:K]} \left(\boldsymbol{X}^{\text{pool}}[b, n, v + p, u + q]\right) \tag{1.4}$$

When deployed for classification purposes, the CNN pipeline is often terminated by FC layers. In contrast with convolutional layers, FC layers do not implement weight sharing and involve as much weight as input data (i.e., $W = K$, $H = J, U = V = 1$). Moreover, in a similar way as *conv* layers, a nonlinear function is applied to the outputs of FC layers.

$$\forall \{b, n\} \in [1, B] \times [1, N]$$

$$\boldsymbol{Y}^{\text{fc}}[b, n] = \beta^{\text{fc}}[n] + \sum_{c=1}^{C} \sum_{h=1}^{H} \sum_{w=1}^{W} \boldsymbol{X}^{\text{fc}}[b, c, h, w] \cdot \Theta^{\text{fc}}[n, c, h, w] \tag{1.5}$$

The Softmax function is a generalization of the Sigmoid function, and "squashes" a N-dimensional vector $\boldsymbol{X}$ to $Sigmoid(\boldsymbol{X})$ where each output is in the range [0,1]. The Softmax function is used in various multi-class classification methods, especially in CNNs. In this case, the Softmax layer is placed at the end of the network and the dimension of vector it operates on (i.e., N) represents the number of classes in the considered dataset. Thus, the input of the Softmax is the data generated by the last fully connected layer, and the output is the probability predicted for each class.

$$\forall \{b,n\} \in [1,B] \times [1,N]$$

$$\text{Softmax}(\mathbf{X}[b,n]) = \frac{\exp(\mathbf{X}[b,n])}{\sum_{c=1}^{N} \exp(\mathbf{X}[b,c])} \tag{1.6}$$

Batch normalization was introduced [14] to speed up training by linearly shifting and scaling the distribution of a given batch of inputs B to have zero mean and unit variance. These layers find also their interest when implementing binary neural networks (BNNs) as they reduce the quantization error compared to an arbitrary input distribution, as highlighted in Hubara et al. [15]. Equation 1.7 details the processing of *batch norm* layers, where the mean μ and the variance σ are statistics collected during the training, α and γ are parameters learned during the training, and ϵ is a hyper-parameter set empirically for numerical stability purposes (i.e., avoiding division by zero).

$$\forall \{b,n,u,v\} \in [1,B] \times [1,N] \times [1,V] \times [1,U]$$

$$\mathbf{Y}^{\text{BN}}[b,n,v,u] = \frac{\mathbf{X}^{\text{BN}}[b,n,u,v] - \mu}{\sqrt{\sigma^2 + \epsilon}} \gamma + a \tag{1.7}$$

1.2.4 Workloads and Computations

The accuracy of CNN models has been increasing since their breakthrough in 2012 [12]. However, this accuracy comes at a high computational cost. The main challenge that faces CNN developers is to improve classification accuracy while maintaining a tolerable computational workload. As shown in Table 1.2, this challenge was successfully addressed by Inception [16] and ResNet models [17], with their use of bottleneck 1×1 convolutions that reduce both model size and computations while increasing depth and accuracy.

1.2.4.1 Computational Workload

As shown in Equations 1.1 and 1.5, the processing of CNN involves an intensive use of Multiply Accumulate (MAC) operation. All these MAC operations take place at *conv* and FC layers, while the remaining parts of network are element-wise transformations that can be generally implemented with low-complexity computational requirements.

TABLE 1.2
Popular CNN Models with Their Computational Workload*

Model	AlexNet [12]	GoogleNet [16]	VGG16 [6]	VGG19 [6]	ResNet101 [17]	ResNet-152 [17]
Top1 err (%)	42.9%	31.3%	28.1%	27.3%	23.6% %	23.0%
Top5 err (%)	19.80%	10.07%	9.90%	9.00%	7.1%	6.7%
L_c	5	57	13	16	104	155
$\sum_{\ell=1}^{L_c} \mathcal{C}_\ell^{\text{conv}}$	666 M	1.58 G	15.3 G	19.5 G	7.57 G	11.3 G
$\sum_{\ell=1}^{L_c} \mathcal{W}_\ell^{\text{conv}}$	2.33 M	5.97 M	14.7 M	20 M	42.4 M	58 M
Act	ReLU					
Pool	3	14	5	5	2	2
L_f	3	1	3	3	1	1
$\sum_{\ell=1}^{L_f} \mathcal{C}_\ell^{\text{fc}}$	58.6 M	1.02 M	124 M	124 M	2.05 M	2.05 M
$\sum_{\ell=1}^{L_f} \mathcal{W}_\ell^{fc}$	58.6 M	1.02 M	124 M	124 M	2.05 M	2.05 M
$\mathcal{C}$	724 M	1.58 G	15.5 G	19.6 G	7.57 G	11.3 G
$\mathcal{W}$	61 M	6.99 M	138 M	144 M	44.4 M	60 M

* Accuracy Measured on Single-Crops of ImageNet Test-Set

In this chapter, the computational workload $\mathcal{C}$ of a given CNN corresponds to the number of MACs it involves during inference*. The number of these MACs mainly depends on the topology of the network, and more particularly on the number of *conv* and FC layers and their dimensions. Thus, the computational workload can be expressed as in Equation 1.8, where L_c is the number of *conv* (fully connected) layers, and $\mathcal{C}_\ell^{\text{conv}}$ ($\mathcal{C}_\ell^{\text{fc}}$) is the number of MACs occurring on a given convolution (fully connected) layer ℓ.

$$\mathcal{C} = \sum_{\ell=1}^{L_c} \mathcal{C}_\ell^{\text{conv}} + \sum_{\ell=1}^{L_f} \mathcal{C}_\ell^{\text{fc}} \tag{1.8}$$

$$\mathcal{C}_\ell^{\text{conv}} = N_\ell \times C_\ell \times J_\ell \times K_\ell \times U_\ell \times V_\ell \tag{1.9}$$

$$\mathcal{C}_\ell^{\text{fc}} = N_\ell \times C_\ell \times W_\ell \times H_\ell \tag{1.10}$$

* Batch size is set to 1 for clarity purposes.

In a similar way, the number of weights, and consequently the size of a given CNN model, can be expressed as follows:

$$\mathcal{W} = \sum_{\ell=1}^{L_c} \mathcal{W}_\ell^{\text{conv}} + \sum_{\ell=1}^{L_f} \mathcal{W}_\ell^{\text{fc}} \tag{1.11}$$

$$\mathcal{W}_\ell^{\text{conv}} = N_\ell \times C_\ell \times J_\ell \times K_\ell \tag{1.12}$$

$$\mathcal{W}_\ell^{\text{fc}} = N_\ell \times C_\ell \times W_\ell \times H_\ell \tag{1.13}$$

For state-of-the-art CNN models, L_c, N_ℓ, and C_ℓ can be quite large. This makes CNNs *computationally and memory intensive*, where for instance, the classification of a single frame using the VGG19 network requires 19.5 billion MAC operations.

It can be observed in the same table that most of the MACs occur on the convolution parts, and consequently, 90% of the execution time of a typical inference is spent on *conv* layers [18]. By contrast, FC layers marginalize most of the weights and thus the size of a given CNN model.

1.2.4.2 Parallelism in CNNs

The high computational workload of CNNs makes their inference a challenging task, especially on low-energy embedded devices. The key solution to this challenge is to leverage on the extensive concurrency they exhibit. These parallelism opportunities can be formalized as follows:

- **Batch Parallelism:** CNN implementations can simultaneously classify multiple frames grouped as a *batch B* in order to reuse the filters in each layer, minimizing the number the memory accesses. However, and as shown in [10], batch parallelism quickly reaches its limits. This is due to the fact that most of the memory transactions result from storing intermediate results and not loading CNN parameters. Consequently, reusing the filters only slightly impacts the overall processing time per image.
- **Inter-layer Pipeline Parallelism:** CNNs have a feed-forward hierarchical structure consisting of a succession of data-dependent layers. These layers can be executed in a pipelined fashion by launching layer (ℓ) before ending the execution of layer ($\ell - 1$). This pipelining costs latency but increases throughput.

Moreover, the execution of the most computationally intensive parts (i.e., *conv* layers), exhibits the four following types of concurrency:

- **Inter-FM Parallelism:** Each two-dimensional plane of an FM can be processed separately from the others, meaning that P_N elements of $\boldsymbol{Y}^{\text{conv}}$ can be computed in parallel ($0 < P_N < N$).

- **Intra-FM Parallelism:** In a similar way, pixels of a single output FM plane are data-independent and can thus be processed concurrently by evaluating $P_V \times P_U$ values of $\boldsymbol{Y}^{\text{conv}}[n]$ $(0 < P_V \times P_U < V \times U)$.
- **Inter-convolution Parallelism:** Depth convolutions occurring in *conv* layers can be expressed as a sum of 2D convolutions, as shown in Equation 1.2. These 2D convolutions can be evaluated simultaneously by computing concurrently P_c elements $(0 < P_c < C)$.
- **Intra-convolution Parallelism:** The 2D convolutions involved in the processing of *conv* layers can be implemented in a pipelined fashion such as in [76]. In this case $P_J \times P_K$ multiplications are implemented concurrently $(0 < P_J \times P_K < J \times K)$.

1.2.4.3 Memory Accesses

As a consequence of the previous discussion, the inference of a CNN shows large vectorization opportunities that can be exploited by allocating multiple computational resources to concurrently process multiple features. However, this parallelization can not accelerate the execution of a CNN if no datacaching strategy is implemented. In fact, memory bandwidth is often the bottleneck when processing CNNs.

In *FC* parts, the execution can be memory-bounded because of the high number of weights that these layers contain, and consequently, the high number of memory reads required.

This is expressed in Equation 1.14, where $\mathcal{M}_\ell^{\text{fc}}$ refers to the number of memory accesses occurring in an FC layer ℓ. This number can be written as the sum of memory accesses reading the inputs $\mathbf{X}_\ell^{\text{fc}}$, the memory accesses reading the weights $(\theta_\ell^{\text{fc}})$, and the number of memory accesses writing the results $(\mathbf{Y}_\ell^{\text{fc}})$.

$$\mathcal{M}_\ell^{\text{fc}} = \text{MemRd}(\mathbf{X}_\ell^{\text{fc}}) + \text{MemRd}(\theta_\ell^{\text{fc}}) + \text{MemWr}(\mathbf{Y}_\ell^{\text{fc}}) \tag{1.14}$$

$$= C_\ell H_\ell W_\ell + N_\ell C_\ell H_\ell W_\ell + N_\ell \tag{1.15}$$

$$\sim N_\ell C_\ell H_\ell W_\ell \tag{1.16}$$

Note that the fully connected parts of state-of-the-art models involve large values of N_ℓ and C_ℓ, making the memory reading of weights the most impacting factor, as formulated in Equation 1.16. In this context, batch parallelism can significantly accelerate the execution of CNNs with a large number of FC layers.

In the *conv* parts, the high number of MAC operations results in a high number of memory accesses, as each MAC requires at least 2 memory reads and 1 memory write*. This number of memory accesses accumulates with the high dimensions of data manipulated by *conv* layers, as shown in Equation 1.18. If all these accesses are towards external memory (for instance, DRAM), throughput and energy consumption

* This is the best-case scenario of a fully pipelined MAC, where intermediate results do not need to be loaded.

will be highly impacted, because DRAM access engenders high latency and energy consumption, even more than the computation itself [21].

$$\mathcal{M}_\ell^{\text{conv}} = \text{MemRd}(\mathbf{X}_\ell^{\text{conv}}) + \text{MemRd}(\theta_\ell^{\text{conv}}) + \text{MemWr}(\mathbf{Y}_\ell^{\text{conv}}) \tag{1.17}$$

$$= C_\ell H_\ell W_\ell + N_\ell C_\ell J_\ell K_\ell + N_\ell U_\ell V_\ell \tag{1.18}$$

The number of these DRAM accesses, and thus latency and energy consumption, can be reduced by implementing a memory-caching hierarchy using on-chip memories. As discussed in the next sections, state-of-the-art CNN accelerators employ register files as well as several levels of caches. The former, being the fastest, is implemented at the nearest of the computational capabilities. The latency and energy consumption resulting from these caches is lower by several orders of magnitude than external memory accesses, as pointed out in Sze et al. [22].

1.2.4.4 Hardware, Libraries, and Frameworks

In order to catch the parallelism of CNNs, dedicated hardware accelerators are developed. Most of them are based on GPUs, which are known to perform well on regular parallelism patterns thanks to simd and simt execution models, a dense collection of floating-point computing elements that peak at 12 TFLOPs, and high capacity/bandwidth on/off-chip memories [23]. To support these hardware accelerators, specialized libraries for deep learning are developed to provide the necessary programming abstraction, such as CudNN on Nvidia GPU [24]. Built upon these libraries, dedicated frameworks for deep learning are proposed to improve productivity of conceiving, training, and deploying CNNs, such as Caffe [25] and TensorFlow [26].

Beside GPU implementations, numerous FPGA accelerators for CNNs have been proposed. FPGAs are fine-grained programmable devices that can catch the CNN parallelism patterns with no memory bottleneck, thanks to the following:

1. A high density of hard-wired digital signal processor (DSP) blocks that are able to achieve up to 20 (8 TFLOPs) TMACs [8].
2. A collection of in situ on-chip memories, located next to DSPs, that can be exploited to significantly reduce the number of external memory accesses.

As a consequence, CNNs can benefit from a significant acceleration when running on reconfigurable hardware. This has caused numerous research efforts to study FPGA-based CNN acceleration, targeting both high performance computing (HPC) applications [27] and embedded devices [28].

In the remaining parts of this chapter, we conduct a survey on methods and hardware architectures to accelerate the execution of CNN on FPGA. The next section lists the evaluation metrics used, then Sections 1.4 and 1.5 respectively study the computational transforms and the data-path optimization involved in recent CNN accelerators. Finally, the last section of this chapter details how approximate computing is a key in FPGA-based deep learning, and overviews the main contributions implementing these techniques.

1.3 FPGA-BASED DEEP LEARNING

Accelerating a CNN on an FPGA-powered platform can be seen as an optimization effort that focuses on one or several of the following criteria:

- *Computational Throughput* ($\mathcal{T}$): A large number of the works studied in this chapter focus on reducing the CNN execution times on the FPGA (i.e., the computation latency), by improving the computational throughput of the accelerator. This throughput is usually expressed as the number of MACs an accelerator performs per second. While this metric is relevant in the case of HPC workloads, we prefer to report the throughput as the number of frames an accelerator processes per second (fps), which better suits the embedded vision context. The two metrics can be directly related using Equation 1.19, where $\mathcal{C}$ is defined in Equation 1.8, and refers to the number of computations a CNN involve in order to process a single frame:

$$\mathcal{T}_{(\text{FPS})} = \frac{\mathcal{T}_{(\text{MACS})}}{\mathcal{C}_{(\text{MAC})}} \tag{1.19}$$

- *Classification/Detection Perf.* ($\mathcal{A}$): Another way to reduce CNN execution times is to trade some of their modeling performance in favor of faster execution timings. For this reason, the classification and detection metrics are reported, especially when dealing with *approximate computing* methods. Classification performance is usually reported as top-1 and top-5 accuracies, and detection performance is reported using the mAP50 and mAP75 metrics.
- *Energy and Power Consumption* ($\mathcal{P}$): Numerous FPGA-based acceleration methods can be categorized as either latency-driven or energy-driven. While the former focus on improving the computational throughput, the latter considers the power consumption of the accelerator, reported in watts. Alternatively, numerous latency-driven accelerators can be ported to low-power-range FPGAs and perform well under strict power consumption requirements.
- *Resource Utilization* ($\mathcal{R}$): When it comes to FPGA acceleration, the utilization of the available resources (lut, DSP blocks, sram blocks) is always considered. Note that the resource utilization can be correlated to the power consumption*, but improving the ratio between the two is a technological problem that clearly exceeds the scope of this chapter. For this reason, both power consumption and resources utilization metrics will be reported when available.

An FPGA implementation of a CNN has to satisfy to the former requirements. In this perspective, the literature provides three main approaches to address the problem

* At a similar number of memory accesses. These accesses typically play the most dominant role in the power consumption of an accelerator.

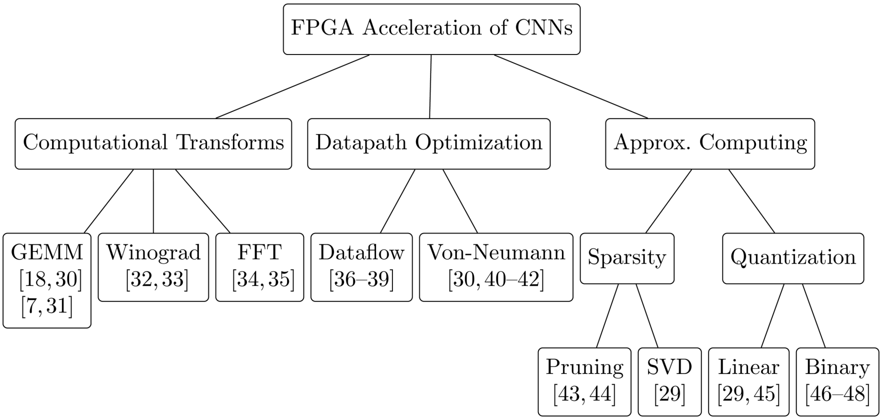


FIGURE 1.2 Main approaches to accelerate CNN inference on FPGAs.

of FPGA-based deep learning. These approaches mainly consists of computational transforms, data-path optimization, and approximate computing techniques, as illustrated in Figure 1.2.

1.4 COMPUTATIONAL TRANSFORMS

In order to accelerate the execution of *conv* and FC layers, numerous implementations rely on computational transforms. These transforms, which operate on the FM and weight arrays, aim at vectorizing the implementations and reducing the number of operations occurring during inference.

Three main transforms can be distinguished. The *im2col* method reshapes the feature and weight arrays in a way to transform depth convolutions into matrix multiplications. The *FFT* method operates on the frequency domain, transforming convolutions into multiplications. Finally, in *Winograd* filtering, convolutions boil down to element-wise matrix multiplications thanks to a tiling and a linear transformation of data.

These computational transforms mainly appear in temporal architectures and are implemented by means of variety of *linear algebra* libraries such OpenBLAS for CPUs* or cuBLAS for GPUs†. Besides this, various implementations make use of these transforms to efficiently map CNNs on FPGAs.

This section discusses the three former methods, highlighting their use-cases and computational improvements. For a better understanding, we recall that for each layer ℓ:

- The input feature map is represented as four-dimensional array **X**, in which the dimensions $B \times C \times H \times W$ respectively refer to the batch size, the number of input channels, the height, and the width.

* https://www.openblas.net/
† https://developer.nvidia.com/cublas

- The weights are represented as four-dimensional array Θ, in which the dimensions $N \times C \times J \times K$ respectively refer to the depth of the output feature map, the depth of the input feature map, the vertical, and the horizontal kernel size.

1.4.1 The im2col Transformation

In CPUs and GPUs, a common way to process CNNs is to map *conv* and FC layers as general matrix multiplications (GEMMs). A number of studies generalize this approach to FPGA-based implementations.

For FC layers, in which the processing boils down to a matrix-vector multiplication problem, the GEMM-based implementations find their interest when processing a *batch* of FMs. As mentioned in Section 1.2.4.1, most of the weights of CNNs are employed in the FC parts. Instead of loading these weights multiple times to classify multiple inputs, features extracted from a batch of inputs are concatenated onto a $CHW \times B$ matrix. In this case, the weights are loaded only one time per batch, as depicted in Figure 1.3a. As a consequence, the former Equation 1.16 – which expressed the number of memory accesses occurring on FC layers – becomes the following:

$$\mathcal{M}_\ell^{\text{fc}} = \text{MemRd}(\theta_\ell^{\text{fc}}) + \text{MemRd}(\mathbf{X}_\ell^{\text{fc}}) + \text{MemWr}(\mathbf{Y}_\ell^{\text{fc}}) \tag{1.20}$$

$$= N_\ell C_\ell W_\ell H_\ell + BC_\ell H_\ell W_\ell + BN_\ell \tag{1.21}$$

$$\sim N_\ell C_\ell H_\ell W_\ell \tag{1.22}$$

As detailed in Section 1.2.4.2, the vectorization of FC layers is often employed in GPU implementations to increase the computational throughput while maintaining a constant memory bandwidth utilization. The same concept holds true for FPGA implementations [31, 48, 49], which batch the FC layers to map them as GEMMs.

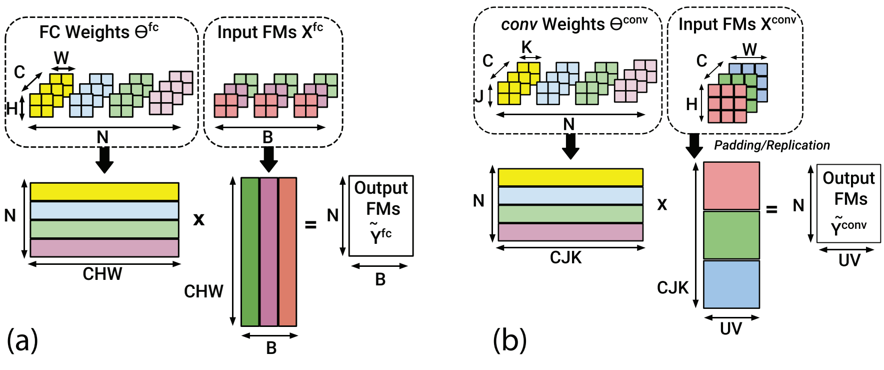


FIGURE 1.3 GEMM-based processing of FC layers (a) and conv layers (b).

3D convolutions can also be mapped as GEMMs using the so-called *im2col* method introduced in [30]. First, this method flattens all the weights of a given *conv* layer onto an $N \times CKJ$ matrix $\tilde{\Theta}$. Second, it rearranges the input feature maps onto a $CKJ \times UV$ matrix $\tilde{X}$, squashing each feature map to a column*. With these reshaped data, the output feature maps $\tilde{Y}$ are computed by multiplying of two former matrices, as illustrated in Figure 1.3b.

$$\tilde{\mathbf{Y}}^{\text{conv}} = \tilde{\Theta}^{\text{conv}} \times \tilde{\mathbf{X}}^{\text{conv}} \tag{1.23}$$

Suda et al. [29] and more recently, Zhang et al. [50] and Guan et al. [51] leverage on *im2col* to derive OpenCL-based FPGA accelerators for CNN. However, this method introduces redundant data in the input FM matrix, which can lead to either inefficiency in storage or complex memory access patterns. As a result, and as pointed out in [22], other strategies to map convolutions have to be considered.

1.4.2 Winograd Transform

Winograd minimal filtering algorithm, introduced in [52], is a computational transform that can be applied to process convolutions with a stride of 1, which is very common in CNN topologies.

This algorithm is particularly efficient when processing small convolutions (where $K \leq 3$), as advocated in [53]. In this work, authors outperformed the throughput of the conventional *im2col* method by a factor of ×7.2 when executing VGG16 on a TitanX GPU.

In Winograd filtering (Figure 1.4), data is processed by blocks, referred to as *tiles*, as follows:

1. An input FM tile x of size $(u \times u)$ is pre-processed: $\tilde{x} = \mathbf{A}^T x \mathbf{A}$
2. In a similar way, θ, the filter tile of size $(k \times k)$, is transformed into $\tilde{\theta}$: $\tilde{\theta} = \mathbf{B}^T x \mathbf{B}$

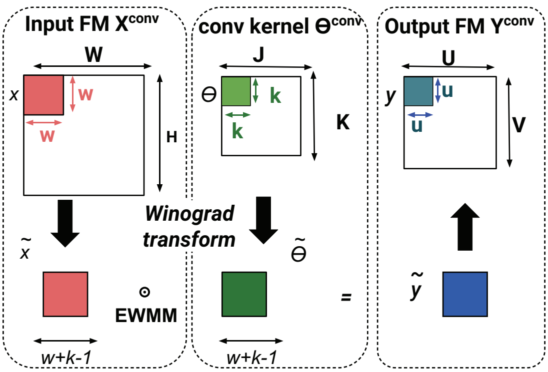


FIGURE 1.4 Winograd filtering $F(u \times u, k \times k)$.

---

* That's what the im2col name refers to: flattening an image to a column.

3. Winograd filtering algorithm, denoted $F(u \times u, k \times k)$, outputs a tile y of size $(u \times u)$ that is computed according to Equation 1.24

$$y = \boldsymbol{C}^T \left[\tilde{\theta} \odot \tilde{x} \right] \boldsymbol{C} \tag{1.24}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ are transformation matrices defined in the Winograd algorithm [52] and $\odot$ denotes the Hadamard product also known as EWMM.

While a standard filtering requires $u^2 \times k^2$ multiplications, Winograd algorithm, denoted $F(u \times u, k \times k)$, requires $(u + k - 1)^2$ multiplications [52]. In the case of tiles of a size $u = 2$ and kernels of size $k = 3$, this corresponds to an arithmetic complexity reduction of $\times 2.25$ [53], and in this case, transform matrices can be written as follows:

$$\mathbf{A}^T = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & -1 \end{bmatrix}; \quad \mathbf{B}^T = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 1/2 \\ 1/2 & -1/2 & 1/2 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.25}$$

Beside this complexity reduction, implementing Winograd filtering in FPGA-based CNN accelerators has two advantages. First, transformation matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ can be evaluated offline once u and k are determined. As a result, these transforms become multiplications with the constants that can be implemented by means of lut and shift registers, as proposed in [54].

Second, Winograd filtering can employ the loop optimization techniques discussed in Section 1.5.2 to vectorize the implementation. On one hand, the computational throughput is increased when *unrolling* the computation of the ewmm parts over multiple DSP blocks. On the other hand, memory bandwidth is optimized using loop *tiling* to determine the size of the FM tiles and filter buffers.

First, utilization of Winograd filtering in FPGA-based CNN accelerators is investigated in [32] and delivers a computational throughput of 46 GOPs when executing AlexNet convolutional layers. This performance is significantly improved by a factor of ×42 in [31] when optimizing the data path to support Winograd convolutions (by employing loop unrolling and tiling strategies), and storing the intermediate FM in on-chip buffers (cf Section 1.4).

The same method is employed in [54] to derive a CNN accelerator on a Xilinx ZCU102 device that delivers a throughput of 2.94 TOPs on VGG convolutional layers. The reported throughput corresponds to half of the performance of a TitanX device, with 5.7× less power consumption [23]*.

* Implementation in the TitanX GPU employs Winograd algorithm and 32-bit floating point arithmetic.

1.4.3 Fast Fourier Transform

Fast Fourier Transform (FFT) is a well known algorithm to transform the 2D convolutions into ewmm in the frequency domain, as shown in Equation 1.26:

$$\mathbf{conv2D}(X[c], \Theta[n,c]) = \mathrm{IFFT}(\mathrm{FFT}(X[c]) \odot \mathrm{FFT}(\Theta[n,c])) \tag{1.26}$$

Using FFT to process 2D convolutions reduces the complexity from $O(W^2 \times K^2)$ to $O(W^2 log_2(W))$, which is exploited to derive FPGA-based accelerators and to infer CNN [34]. When compared to standard filtering and Winograd algorithm, FFT finds its interest in convolutions with large kernel size ($K > 5$), as demonstrated in [53, 55]. The computational complexity of FFT convolutions can be further reduced to $O(Wlog_2(K))$ using the overlap-and-add method [56], which can be applied when the signal size is much larger than the filter size, which is typically the case in *conv* layers ($W >> K$). Works in [33, 57] leverage on the overlap-and-add to implement frequency domain acceleration for *conv* layers on FPGA, which results in a computational throughput of 83 GOPs for AlexNet (Table 1.3).

1.5 DATA-PATH OPTIMIZATIONS

As highlighted in Section 2.4.2, the execution of CNN exhibits numerous sources of parallelism. However, due to the resource limitations of FPGA devices, it might be impossible to fully exploit all the concurrency patterns, especially with the sheer volume of operations involved in deep topologies. In other words, the execution of recent CNN models cannot fully be unrolled sometimes, not even for a single *conv* layer.

To address this problem, the general approach, advocated in state-of-the-art implementations, is to map a limited number of processing elements (PEs) on the FPGA. These PEs are then reused by temporally iterating data through them.

1.5.1 Systolic Arrays

Early FPGA-based accelerators for CNN implemented systolic arrays to accelerate the 2D filtering in convolutions layers [58—61]. As illustrated in Figure 1.5a, systolic arrays employ *a static collection* of PE, typically arranged in a 2-dimensional grid. These PE operate as a co-processor under the control of a central processing unit. The configuration of systolic arrays is *agnostic* to the CNN model, making them inefficient to process large-scale networks for the following three reasons:

First, the static collection of PE can support convolutions only up to a given filter size K_m, where typical values of K_m range from 7 in [59] to 10 in [61]. Therefore, in convolutional layer (ℓ), $K_\ell > K_m$ is not supported by the accelerator. Second, systolic arrays suffer from under-utilization when processing layers in which the kernel size K_ℓ is much smaller then K_m. This is for instance the case in [61], where the processing of 3×3 convolutions uses only 9% of DSP blocks, while the processing of these layers can be further parallelized and thus accelerated. Third and finally, PE in systolic arrays do not usually include memory caches and have to fetch their inputs from

TABLE 1.3
Accelerators Employing Computational Transforms

Method	Entry	Network	Comp (GOP)	Params (M)	Bit-width	Desc.	Device	Freq (MHz)	Through (GOPs)	Power (W)	LUT (K)	DSP	Memory (MB)
Winograd	[33]	AlexNet-C	1.3	2.3	Float 32	OpenCL	Virtex7 VX690T	200	46	–	505	3683	56.3
	[32]	AlexNet-C	1.3	2.3	Float16	OpenCL	Arria10 GX1150	303	1382	44.3	246	1576	49.7
	[55]	VGG16-C	30.7	14.7	Fixed 16	HLS	Zynq ZU9EG	200	3045	23.6	600	2520	32.8
	[55]	AlexNet-C	1.3	2.3	Fixed 16	HLS	Zynq ZU9EG	200	855	23.6	600	2520	32.8
FFT	[34]	AlexNet-C	1.3	2.3	Float 32	–	Stratix5 QPI	200	83	13.2	201	224	4.0
	[34]	VGG19-C	30.6	14.7	Float 32	–	Stratix5 QPI	200	123	13.2	201	224	4.0
GEMM	[30]	AlexNet-C	1.3	2.3	Fixed 16	OpenCL	Stratix5 GXA7	194	66	33.9	228	256	37.9
	[50]	VGG16-F	31.1	138.0	Fixed 16	HLS	Kintex KU060	200	365	25.0	150	1058	14.1
	[50]	VGG16-F	31.1	138.0	Fixed 16	HLS	Virtex7 VX960T	150	354	26.0	351	2833	22.5
	[51]	VGG16-F	31.1	138.0	Fixed 16	OpenCL	Arria10 GX1150	370	866	41.7	437	1320	25.0
	[51]	VGG16-F	31.1	138.0	Float 32	OpenCL	Arria10 GX1150	385	1790	37.5	–	2756	29.0

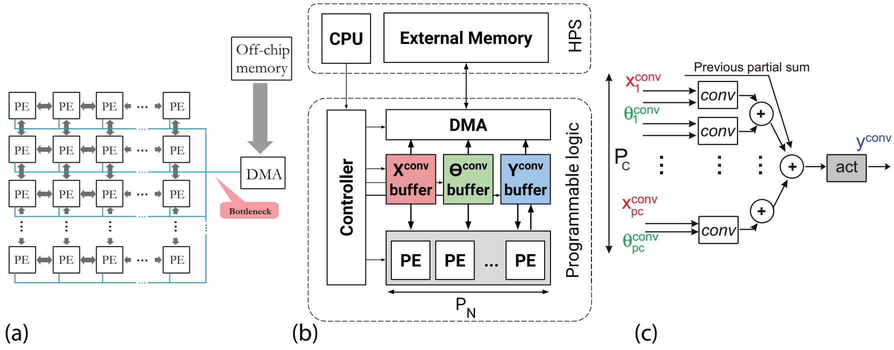


FIGURE 1.5 Generic data paths of FPGA-based CNN accelerators: (a) Static systolic array. (b) Dedicated SIMD accelerator. (c) Dedicated processing element.

an off-chip memory. As a result, the performance of systolic arrays can rapidly be bounded by memory bandwidth of the device.

1.5.2 Loop Optimization in Spatial Architectures

Due to the inefficiency of systolic arrays, flexible and dedicated spatial architectures for CNN were mapped on FPGA. The general computation flow in these accelerators is illustrated in Figure 1.5b.

First, FMs and weights are fetched from DRAM to on-chip buffers, and are then *streamed* into the PE. At the end of the PE computation, results are transferred back to on-chip buffers and, if necessary, to the external memory in order to be fetched in their turn to process the next layers. Each PE – as depicted in Figure 1.5c – is configurable and has its own *computational* capabilities by means of DSP blocks, and its own data *caching* capabilities by means of on-chip registers. With this paradigm, the problem of CNN mapping consists of finding the optimal architectural and temporal configuration of PE: in other words, the best number of DSP blocks per PE, the optimal temporal scheduling of data that maximizes the computational throughput.

For convolutional layers, in which the processing is described in Listing 1.1, finding the optimal PE configuration comes down to a loop optimization problem [28, 29, 39, 40, 62–64].

Listing 1.1: Nested Loops

```
// Lb : Batch
for (int b =0;b<B,l++) {
// Ll: Layer
for (int l =0;l<L,l++) {
// Ln: Y Depth
for (int n =0;n<N;n++) {
// Lv: Y Columns
for (int v =0;v<V,v++) {
// Lu: Y Raws
```

```
for (int u =0;u<U,u++) {
// Lc: X Depth
for (int c =0;n<C;c++) {
// Lj: Theta Columns
for (int j =0;j<J,j++) {
// Lk: Theta Raws
for (int k =0;k<K,k++) {
Y[b,l,n,v,u] +=
X[b,l,c,v+j,u+k] *
Theta [l,n,c,j,k]
}}}}}}}
```

Listing 1.2: Loop Tiling in conv layers

```
for (int b =0;b<B,l++){
for (int n =0;n<N;n+= Tn){
for (int v =0;v<V,v+= Tv){
for (int u =0;u<U,u+= Tu){
for (int c =0;n<C;c+= Tc){
// DRAM : Load in on - chip
buffers the tiles :
// X[l,c:c+Tc ,v:v+Tv ,u:u+Tu]
// Theta [l,n:n+Tn ,c:c+Tc ,j,k]
for (int tn =0; tn <Tn;tn ++){
for (int tv =0; tv <Tv ,tv ++){
for (int tu =0; tu <Tu ,tu ++){
for (int tc =0; tn <Tc;tc ++){
for (int j =0;j<J,j++){
for (int k =0;k<K,k++){
Y[l,tn ,tv ,tu] +=
X[l,tc ,tv+j,tu+k] *
Theta [l,tn ,tc ,j,k];
}}}}}} // DRAM : Store output
}}}}
```

This problem is addressed by applying loop optimization techniques such *loop unrolling*, *loop tiling*, or *loop interchange* to the 7 nested loops of Listing 1.1. In this case, the unroll and tiling factors (respectively P_i and T_i) determine the number of PEs, the computational resources, and the on-chip memory allocated to each PE.

Loop Unrolling

Unrolling a loop L_i with an unrolling factor P_i ($P_i \leq i, i \in \{L,V,U,N,C,J,K\}$) accelerates its execution by allocating multiple computational resources. Each of the parallelism patterns listed in Section 1.2.4.2 can be implemented by unrolling one of the loops of Listing 1.1, as summarized in Table 1.4. For the configuration given in Figure 1.5c, the unrolling factor P_N sets the number of PEs. The remaining factors – P_C, P_K, P_J – determine the number of multipliers, as well as the size of buffer contained in each PE (Figure 1.6).

TABLE 1.4
Loop Optimization Parameters P_i and T_i

Parallelism	Intra layer	Inter FM	Intra FM		Inter conv.	Intra conv.	
Loop	L_L	L_N	L_V	L_U	L_c	L_J	L_K
Unroll Factor	P_L	P_N	P_V	P_U	P_c	P_J	P_K
Tiling Factor	T_L	T_N	T_U	T_U	T_C	T_J	T_K

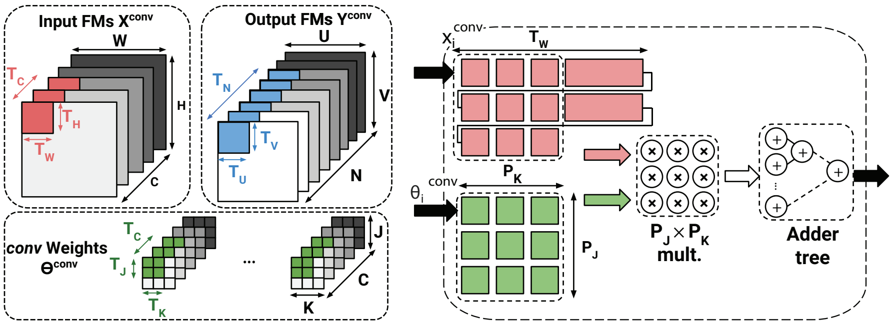


FIGURE 1.6 Loop tiling and unrolling in convolutional layers.

Loop Tiling

In general, the capacity of on-chip memory in current FPGA is not large enough to store the weights and intermediate FM of all CNN layers*. For example, AlexNet's convolutional layers resort to 18.6 Mbits of weights, and generate a total 70.7 Mbits of intermediate feature maps†. In contrast, the highest-end Stratix V FPGA provides a maximum of 52 Mbits of on-chip ram.

As a consequence, FPGA-based accelerators resort to external DRAM to store these data. As mentioned in Section 1.2.4.3, DRAM accesses are costly in terms of energy and latency, and data caches must be implemented by means of on-chip buffers and local registers. The challenge is thus to build a data path in a way that every data transferred from DRAM is reused as much as possible.

For *conv* layers, this challenge can be addressed by *tiling* the nested loops of Listing 1.1. *Loop tiling* [66] divides the FM and weights of each layer into multiple groups that can fit into the on-chip buffers. For the configuration given in Figure 1.5c, the size of the buffers containing input FM, weights, and output FM is set according to the tiling factors listed in Table 1.4.

$$\mathcal{B}_X^{\text{conv}} = T_C \times T_H \times T_W \tag{1.27}$$

* Exception can be made for [6666], where a large cluster of FPGAs is interconnected and resorts only to on-chip memory to store CNN weights and intermediate data.
† Estimated by summing the number of outputs for each convolution layer.

$$\mathcal{B}_{\Theta}^{\text{conv}} = T_N \times T_C \times T_J \times T_K \tag{1.28}$$

$$\mathcal{B}_{Y}^{\text{conv}} = T_N \times T_V \times T_U \tag{1.29}$$

With these buffers, the memory accesses occurring in the *conv* layer (cf Equation 1.18) are respectively divided by $\mathcal{B}_X^{\text{conv}}$, B_{Θ}^{conv} and B_Y^{conv}, as expressed in Equation 1.30.

$$\mathcal{M}_{\ell}^{\text{conv}} = \frac{C_\ell H_\ell W_\ell}{T_C T_H T_W} + \frac{N_\ell C_\ell J_\ell K_\ell}{T_N T_C T_J T_K} + \frac{N_\ell U_\ell V_\ell}{T_N T_V T_U} \tag{1.30}$$

Since the same hardware is reused to accelerate the execution of multiple conv layers with different workloads, the tiling factors are agnostic to the workload of a specific layer, as can be noticed in the denominator of Equation 1.30. As a result, the value of the tiling factors is generally set to optimize the overall performance of a CNN execution.

1.5.3 Design Space Exploration

Finding the optimal unrolling and tiling factors for a specific device is a complex problem that is generally solved using brute-force design space exploration [29, 39, 40, 48, 67, 68]. This exploration is driven by an analytical model, in which the inputs are loop factors P_i, T_i and outputs are theoretical predictions of the computational throughput ($\mathcal{T}$), the size of buffers ($\mathcal{B}$), and the number of external memory accesses ($\mathcal{M}$). This model is parametrized by the available resources of a given FPGA platform and the workload of the considered CNN. To select feasible solutions for this optimization problem, most literature approaches rely on the *Roofline* method [69] to accept or reject design solutions that do not match with the maximum computational throughput or the maximum memory bandwidth of a given device (Figures 1.7).

A typical design space exploration driven by the roofline model is illustrated in Figure 1.8. In this graph, each point represents the performance of an explored solution (P_i, T_i). For a given FPGA platform, the attainable bandwidth and computational throughput are respectively reported by the diagonal and horizontal lines. Point A is an invalid solution, as it is above the bandwidth roof, while point A' is feasible but delivers mediocre computational throughput. Acceptable solutions are represented by points C and D, the latter being better than the former since it has lower bandwidth requirements.

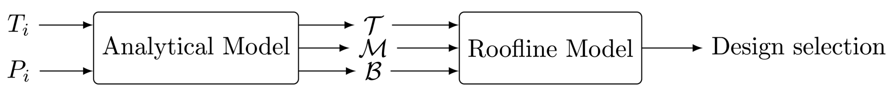


FIGURE 1.7 Design space exploration methodology.

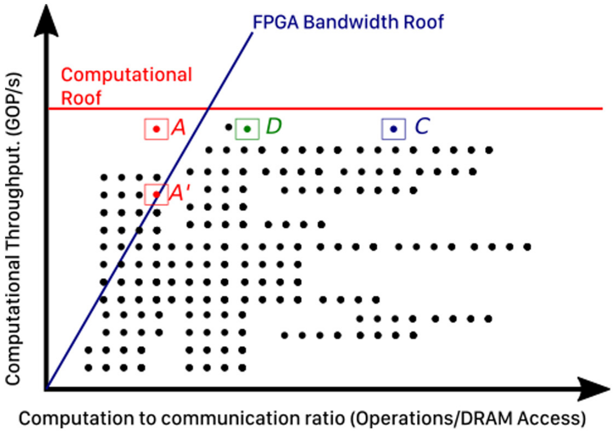


FIGURE 1.8 Example of a design selection driven by the roofline model.

1.5.4 FPGA Implementations

Employing loop optimizations to derive FPGA-based CNN accelerator was first investigated in [39]. In this work, Zhang et al. report a computational throughput of 61.62 GOPs in the execution of AlexNet convolutional layers by unrolling loops L_C and L_N. This accelerator, described with Vivado HLS tools, relies on 32-bit floating-point arithmetic. Works in [68] follow the same unrolling scheme and feature a 16-bit fixed-point arithmetic, resulting in a ×2.2 improvement in terms of computational throughput. Finally, the same unrolling and tiling scheme is employed in recent work [48], where authors report a ×13.4 improvement, thanks to a deeply pipelined FPGA cluster of four Virtex7-XV960t devices.

In all these implementations, loops L_J and L_K are not unrolled because J and K are usually small, especially in recent topologies. Works of Motamedi et al. [40] study the impact of unrolling these loops in AlexNet, where the first convolutional layers use large 11×11 and 5×5 filters. Expanding loop unrolling and tiling to loops L_J and L_K results in a 1.36× improvement in computational throughput vs. [39] on the same VX485T device when using 32-bit floating-point arithmetic. Nevertheless, and as pointed out in [63], unrolling these loops is ineffective for recent CNN models that employ small convolution kernels.

The values of U, V, N can be very large in CNN models. Consequently, unrolling and tiling loops L_U, L_V, L_N can be efficient only for devices with high computational capabilities (i.e., DSP blocks). This is demonstrated in works of Rahman et al. [67] that report an improvement of ×1.22 over [39] when enlarging the design space exploration to loops L_U, L_V, L_N, which comes at the price of very long exploration timing. In order to keep data in on-chip buffer after the execution of a given layer, works of Alwani et al. [62] advocate the use of *fused-layer* accelerators by tiling across layer L_L. As a result, authors are able to remove 95% of DRAM accesses at the cost of 362 KB of extra on-chip memory.

In all these approaches, loops L_N, L_C, L_J, L_K are unrolled in a similar way they are tiled (i.e., $T_i = P_i$). By contrast, the works of Ma et al. [63, 70] fully explore all the design variables searching for optimal loop unroll and tiling factors. More particularly, the authors demonstrate that the input FM and weights are optimally reused when unrolling only computations within a single input FM (i.e., when $P_C = P_J = P_K = 1$). Tiling factors are set in such a way that all the data required to compute an element of Y are fully buffered (i.e., $T_C = C$, $T_K = K$, $T_J = J$). The remaining design parameters are derived after a brute-force design exploration. The same authors leverage on these loop optimizations to build an RTL compiler for CNNs in [71]. To the best of our knowledge, this accelerator outperforms all the previous implementations that are based on loop optimization in terms of computational throughput (Tables 1.5 through 1.7).

1.6 APPROXIMATE COMPUTING OF CNN MODELS

Besides the computational transforms and data-path optimization, the CNN execution can be accelerated when employing approximate computing, which is known to perform efficiently on FPGAs [73].

In the methods detailed in this section, a minimal amount of the CNN accuracy is traded to improve the computational throughput or energy efficiency of the accelerator. Two main strategies are employed. The first implements approximate *arithmetic* to process the CNN layers with a reduced precision. The second aims at reducing the number of operations occurring in CNN models without critically affecting the modeling performance. Note that both approaches can resort to *fine-tuning* in order to compensate the accuracy loss introduced by approximate computing.

1.6.1 Approximate Arithmetic for CNNs

Several studies have demonstrated that the precision of both operations and operands in CNN, and more generally in neural networks, can be reduced without critically affecting their predictive performance. This reduction can be achieved by *quantizing* either or both of the CNN inputs, weights, and/or FM using a fixed-point numerical representation.

1.6.1.1 Fixed-Point Arithmetic

In a general way, CNN models are deployed in CPU and GPU using the same numerical precision they were trained with, relying on the *single-precision floating-point* representation. This format employs 32 bits, arranged according to the IEEE754 standard. As current FPGAs support floating operations, various implementations [39, 62, 67] employ such data representation.

Nonetheless, numerous studies such [74–76] demonstrate that the inference of CNNs can be achieved with a reduced precision of operands. More particularly, works in [77, 78] demonstrate the applicability of fixed-point (F×P) arithmetic to *train* and *infer* CNNs. The F×P representation encodes numbers with a given bit-width b, using i bits for the *integer* part, and f bits for the *fractional* part ($b = i + f$). Note that the value of i is selected according the desired *numerical range*, and the value of f is selected according to the desired numerical *precision*.

TABLE 1.5
Accelerators Employing Loop Optimization

Entry	Network	Comp (GOP)	Params (M)	Bit-width	Desc.	Device	Freq (MHz)	Through (GOPs)	Power (W)	LUT (K)	DSP	Memory (MB)
[40]	AlexNet-C	1.3	2.3	Float 32	HLS	Virtex7 VX485T	100	61.62	18.61	186	2240	18.4
[29]	VGG16SVD-F	30.8	50.2	Fixed 16	RTL	Zynq Z7045	150	136.97	9.63	183	780	17.5
[30]	AlexNet-C	1.3	2.3	Fixed 16	OpenCL	Stratix5 GSD8	120	187.24	33.93	138	635	18.2
[30]	AlexNet-F	1.4	61.0	Fixed 16	OpenCL	Stratix5 GSD8	120	71.64	33.93	272	752	30.1
[30]	VGG16-F	31.1	138.0	Fixed 16	OpenCL	Stratix5 GSD8	120	117.9	33.93	524	1963	51.4
[68]	AlexNet-C	1.3	2.3	Float 32	HLS	Virtex7 VX485T	100	75.16	33.93	28	2695	19.5
[49]	AlexNet-F	1.4	61.0	Fixed 16	HLS	Virtex7 VX690T	150	825.6	126.00	N.R	14400	N.R
[49]	VGG16-F	31.1	138.0	Fixed 16	HLS	Virtex7 VX690T	150	1280.3	160.00	N.R	21600	N.R
[69]	NIN-F	2.2	61.0	Fixed 16	RTL	Stratix5 GXA7	100	114.5	19.50	224	256	46.6
[69]	AlexNet-F	1.5	7.6	Fixed 16	RTL	Stratix5 GXA7	100	134.1	19.10	242	256	31.0
[38]	AlexNet-F	1.4	61.0	Fixed 16	RTL	Virtex7 VX690T	156	565.94	30.20	274	2144	34.8
[63]	AlexNet-C	1.3	2.3	Float 32	HLS	Virtex7 VX690T	100	61.62	30.20	273	2401	20.2
[64]	VGG16-F	31.1	138.0	Fixed 16	RTL	Arria10 GX1150	150	645.25	50.00	322	1518	38.0
[42]	AlexNet-C	1.3	2.3	Fixed 16	RTL	Cyclone5 SEM	100	12.11	N.R	22	28	0.2
[42]	AlexNet-C	1.3	2.3	Fixed 16	RTL	Virtex7 VX485T	100	445	N.R	22	2800	N.R
[72]	NiN	20.2	7.6	Fixed 16	RTL	Stratix5 GXA7	150	282.67	N.R	453	256	30.2
[72]	VGG16-F	31.1	138.0	Fixed 16	RTL	Stratix5 GXA7	150	352.24	N.R	424	256	44.0
[72]	ResNet-50	7.8	25.5	Fixed 16	RTL	Stratix5 GXA7	150	250.75	N.R	347	256	39.3
[72]	NiN	20.2	7.6	Fixed 16	RTL	Arria10 GX1150	200	587.63	N.R	320	1518	30.4
[72]	VGG16-F	31.1	138.0	Fixed 16	RTL	Arria10 GX1150	200	720.15	N.R	263	1518	44.5
[72]	ResNet-50	7.8	25.5	Fixed 16	RTL	Arria10 GX1150	200	619.13	N.R	437	1518	38.5
[73]	AlexNet-F	1.5	7.6	Float 32	N.R	Virtex7 VX690T	100	445.6	24.80	207	2872	37
[73]	VGG16SVD-F	30.8	50.2	Float 32	N.R	Virtex7 VX690T	100	473.4	25.60	224	2950	47

TABLE 1.6
Accelerators Employing Approximate Arithmetic

A×C	Entry	Dataset	Comp (GOP)	Params (M)	Bit-width In/Out	FMs	θ^{conv}	θ^{FC}	Acc (%)	Device	Freq (MHz)	Through. (GOPs)	Power (W)	LUT (K)	DSP	Memory (MB)
FP32	[51]	ImageNet	30.8	138.0	32	32	32	32	90.1	Arria10 GX1150	370	866	41.7	437	1320	25.0
FP16	[32]	ImageNet	30.8	61.0	16	16	16	16	79.2	Arria10 GX1150	303	1382	44.3	246	1576	49.7
	[64]	ImageNet	30.8	138.0	16	16	8	8	88.1	Arria10 GX1150	150	645	N.R	322	1518	38.0
DFP	[72]	ImageNet	30.8	138.0	16	16	16	16	N.R	Arria10 GX1150	200	720	N.R	132	1518	44.5
	[51]	ImageNet	30.8	138.0	16	16	16	16	N.R	Arria10 GX1150	370	1790	N.R	437	2756	29.0
	[91]	Cifar10	1.2	13.4	20	2	1	1	87.7	Zynq Z7020	143	208	4.7	47	3	N.R
	[46]	Cifar10	0.3	5.6	20/16	2	1	1	80.1	Zynq Z7045	200	2465	11.7	83	N.R	7.1
BNN	[93]	MNIST	0.0	9.6	8	2	1	1	98.2	Stratix5 GSD8	150	5905	26.2	364	20	44.2
	[93]	Cifar10	1.2	13.4	8	8	1	1	86.3	Stratix5 GSD8	150	9396	26.2	438	20	44.2
	[93]	ImageNet	2.3	87.1	8	32	1a	1	66.8	Stratix5 GSD8	150	1964	26.2	462	384	44.2
	[94]	Cifar10	1.2	13.4	8	2	2	2	89.4	Xilinx7 VX690T	250	10962	13.6	275	N.R	39.4
TNN	[94]	SVHN	0.3	5.6	8	2	2	2	97.6	Xilinx7 VX690T	250	86124	7.1	155	N.R	12.2
	[94]	GTSRB	0.3	5.6	8	2	2	2	99.0	Xilinx7 VX690T	250	86124	6.6	155	N.R	12.2

TABLE 1.7
Accelerators Employing Pruning and Low Rank Approximation

Reduc.	Entry	Dataset	Comp (GOP)	Params (M)	Removed Param. (%)	Bit-width	Acc (%)	Device	Freq (MHz)	Through. (GOPs)	Power (W)	LUT (K)	DSP	Memory (MB)
SVD	[29]	ImageNet	30.8	50.2	63.6	16 Fixed	88.0	Zynq 7Z045	150	137.0	9.6	183	780	17.50
Pruning	[44]	Cifar10	0.3	13.9	89.3	8 Fixed	91.5	Kintex 7K325T	100	8620.7	7.0	17	145	15.12
	[7]	ImageNet	1.5	9.2	85.0	32 Float	79.7	Stratix 10	500	12000.0	141.2	N.R	N.R	N.R

In the simplest version of fixed-point arithmetic, all the numbers are encoded with the *same* fractional and integer bit-widths. This means that the position of the radix point is similar for all the represented numbers. In this chapter, we refer to this representation as *static* F×P.

When compared to floating point, F×P is known to be more efficient in terms of hardware utilization and power consumption. This is especially true in FPGAs [79], where – for instance – a single DSP block in Intel devices can either implement *one* 32-bit floating-point multiplication or *three* concurrent F×P multiplications of 9 bits [8]. This motivated early FPGA implementations such as [61, 80] to employ fixed-point arithmetic in deriving CNN accelerators. These implementations mainly use a 16-bit Q8.8 format, where 8 bits are allocated to the integer parts, and 8 bits to the fractional part. Note that the same Q8.8 format is used for representing the features and the weights of all the layers.

In order to prevent overflow, the former implementations also *expand* the bit-width when computing weighted sums of convolutions. Equation 1.31 explains how the bit-width is expanded; if b_X bits are used to quantize the input FM and b_Θ bits are used to quantize the weights, an accumulator of b_{acc} bits is required to represent a weighted sum of $C_\ell K_\ell^2$ elements, where:

$$b_{\text{acc}} = b_x + b_\Theta + \max_\ell \left[\log_2 \left(C_\ell K_\ell^2 \right) \right] \tag{1.31}$$

In practice, most FPGA accelerators use 48-bit accumulators, such as in [59, 60] (Figure 1.9).

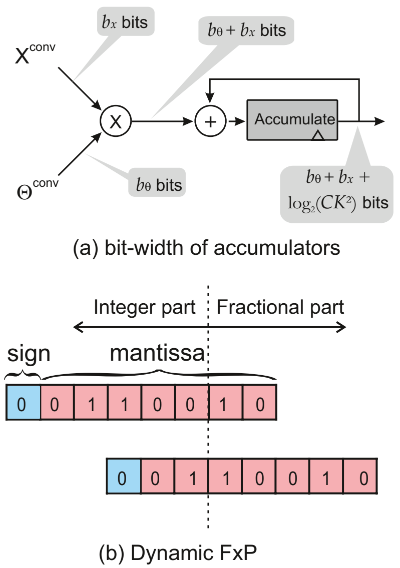


FIGURE 1.9 Fixed-point arithmetic for CNN accelerators.

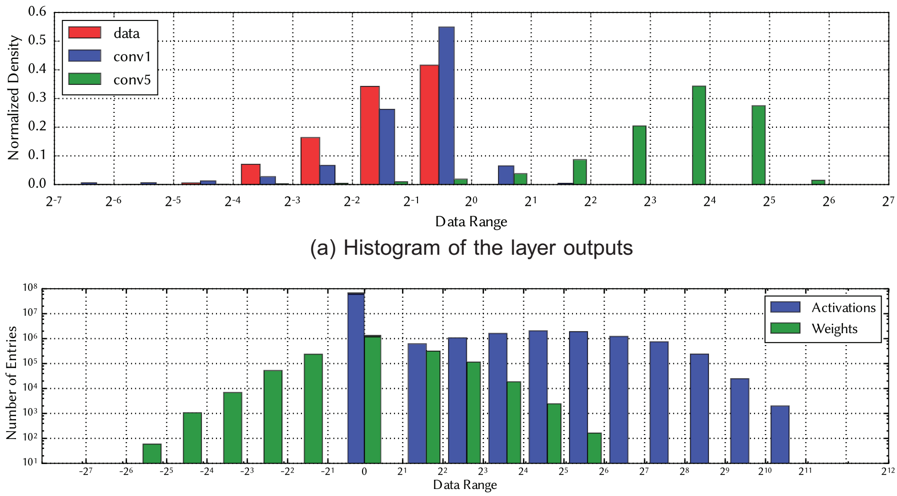
1.6.1.2 Dynamic Fixed Point for CNNs

In deep topologies, it can be observed that distinct parts of a network can have significantly different ranges of data. In particular, the features of the deep layers tend to have a much larger numerical range when compared to the features of the first CNN layers.

The histograms of Figure 1.10a depict this phenomenon for AlexNet convolutional layers*. While the CNN inputs (data column) are normalized take their values between 0 and 1, the outputs of the first convolutional layer (conv1 column) have a *wider* numerical range, between 2^{-7} and 2^{2}. This is even more salient for the fifth convolutional layer, where most of the outputs take their values between 2^{-1} and 2^{6}. The same problem appears when comparing the numericals of the CNN weights, and CNN activations. In this case, the weights are numerically much *smaller* when compared to the activations, as illustrated in Figure 1.10b†.

As a consequence, large bit-widths have to be allocated to the integer fractional parts in order to keep a uniform precision across the network while preventing overflow. This expansion badly increases the resource requirements of a given FPGA mapping. As a result, static F×P, with its unique shared fixed exponent, is ill-suited to deep learning, as pointed out in. [81]

To address this problem, works in [77, 81, 82] advocate the use of *dynamic* F×P [83]‡. In dynamic F×P, different scaling factors are used to process different parts of the network. In other words, the position of the radix point varies from one layer



(a) Histogram of the layer outputs

(b) Histogram of weights and activations. Inputs and weights encoded in 8 bits

FIGURE 1.10 Distribution of AlexNet activations and weights.

---

* Code made available at github.com/KamelAbdelouahab/CNN-Data-Distribution.

† This figure deliberately multiplies the weights and activations by a scale factor of $2^7 - 1$ to emulate an 8-bit quantization.

‡ Another approach to address this problem is to use custom floating point representations, as detailed in [31].

to another. More particularly, weights, weighted sums, and outputs of each layer are assigned distinct integer and fractional bit-widths.

The optimal values of these bit-widths (i.e., the ones that deliver the best trade-off between accuracy loss and computational load) for each layer can be derived after a profiling process performed by dedicated frameworks that support F×P. Among these frameworks, Ristretto [81] and FixCaffe [84] are compatible with Caffe, while TensorFlow natively supports 8-bit computations. Most of these tools can *fine-tune* a given CNN model to improve the accuracy of the quantized network.

In particular, the works in [85] demonstrate the efficiency dynamic of F×P, pointing out how the inference of AlexNet is possible using 6 bits in dynamic F×P instead of 16 bits with a conventional fixed-point format.

1.6.1.3 FPGA Implementations

The FPGA-based CNN accelerator proposed in [29] is built upon this quantization scheme and employs different precisions to represent the FM, convolution kernels, and FC weights with 16, 8, and 10 bits, respectively. Without fine-tuning, the authors report a drop of 1% in the classification accuracy of AlexNet. In a similar way, Qiu et al. employ F×P to quantize the VGG network with respectively 8 bits for the weights, 8 bits for activations, and 4 bits for FC layers, resulting in an accuracy drop of 2%. In all these accelerators, dynamic quantization is supported by means of data shift modules [28, 82]. Finally, the accelerator in [41] relies on the Ristretto framework [81] to derive an AlexNet model wherein the data is quantized in 16 bits with distinct integer bit-widths per layer*.

1.6.1.4 Extreme Quantization and Binary Networks

Training and inferring CNNs with *extremely compact data representations* is an area that has recently gained a lot of research interest. Early works of Courbariaux et al. in BinaryConnect [86] demonstrate the feasibility of training neural networks using *binary* weights, i.e., weights with either a value of $-\theta$ or θ encoded in 1 bit. BinaryConnect lowers the bandwidth requirements of a network by a factor of ×32 at the price of an accuracy loss, evaluated at 19.2% on ImageNet†. The same authors go further in their investigations in [15] and propose BNNs that represent both feature maps and weights with only 1 bit. In these networks, negative values are represented as 0, while positive values are represented as 1. BNNs greatly simplify the processing of convolutions, boiling down the computations of MACs into bitwise XNOR operations followed by a pop-count (see Figure 1.11b). Moreover, the authors use the *sign* function as activation and apply batch normalization before applying the activation, which reduces the information lost during binarization (see Figure 1.11a). In turn, a higher drop in classification accuracy occurs when using BNNs, evaluated at 29.8% for ImageNet. This accuracy drop is then lowered to 11% by Rastegari et al., using different scale factors for binary weights (i.e., $-_{\theta 1}$ or $+_{\theta 2}$).

* Since the same PEs are reused to process different layers, the same bit width is used with a variable radix point for each layer.

† When compared to an exact 32-bit implementation of AlexNet.

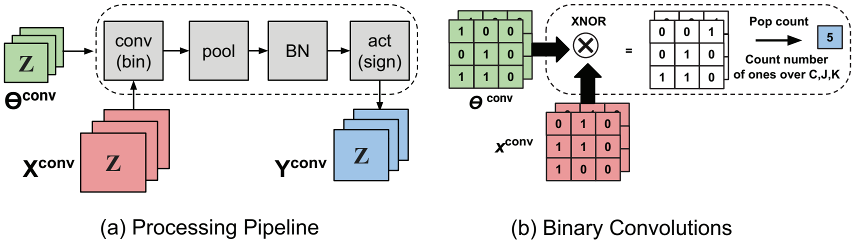


FIGURE 1.11 Binary neural networks.

Beside BNNs, *pseudo-binary networks*, such as DoReFa-Net [87] and quantized neural networks (QNNs) [88], reduce the accuracy drop to 6.5% when employing a slightly expanded bit-width (2 bits) to represent the intermediate FM. Similarly, in TTQ [89], weights are constrained to three values (2 bits) $-_{\theta 1}$, 0, $-_{\theta 2}$, but FMs are represented in a 32-bit float scheme. As a consequence, the efficiency gain of TTQ is not as high as in BNNs. In turn, TTQ achieves comparable accuracy on ImageNet, within 0.7% of full-precision.

In FPGAs, BNNs benefit from a significant acceleration, as the processing of "binary" convolutions can be mapped on XNOR gates followed by a pop-count operation, as depicted in Figure 1.11b. Furthermore, and as suggested in [7], a pop-count operation can be implemented using lookup tables in a way that convolutions are processed only with logical elements. Thus, the DSP blocks can be used to process the batch norm calculation (Equation 1.7, which can be formulated as a linear transform in order to reduce the number of operations). This approach is followed in the implementation of [90] to derive an FPGA-based accelerator for BNNs that achieves 207.8 GOPs while only consuming 4.7 W and 3 DSP blocks to classify the Cifar10 dataset.

For the same task, works in [45, 91] use a smaller network configuration* and reach a throughput of 2.4 TOPs when using a larger Zynq 7Z045 device with 11W power consumption.

For ImageNet classification, binary net implementation of [92] delivers an overall throughput of 1.9 TOPs on a Stratix V GSD device. In all these works, the first layer is not binarized to achieve better classification accuracy. As pointed out in [92], the performance in this layer can be improved when using a higher number of DSP blocks. Finally, an accelerator for TTQ proposed in [93] achieves a peak performance of 8.36 TMACs when classifying the Cifar10 dataset with a 2-bit precision.

1.6.2 Reduced Computations

In addition to approximate arithmetic, several studies attempt to reduce the number of operations involved in CNNs. For FPGA-based implementations, two main strategies are investigated: *weight pruning*, which increases the *sparsity* of the model, and *low-rank approximation* of filters, which reduces the number of multiplications occurring in the inference.

* The network topology used in this work involves 90% fewer computations and achieves 7% less classification accuracy on Cifar10.

1.6.2.1 Weight Pruning

As highlighted in [94], CNNs as overparametrized networks and a large amount of the weights can be removed – or *pruned* – without critically affecting the classification accuracy. In its simplest form, pruning is performed according to the magnitude, such that the lowest values of the weights are truncated to zero [95]. In a more recent approach, weight removal is driven by energy consumption of a given node of the graph, which is 1.74× more efficient than magnitude-based approaches [96]. In both cases, pruning is followed by a fine-tuning of the remaining weights in order to improve the classification accuracy. This is for instance the case in [97], where pruning removes respectively 53% and 85% of the weights in AlexNet *conv* and FC layers for less then 0.5% accuracy loss (Figure 1.12).

1.6.2.2 Low Rank Approximation

Another way to reduce the computations occurring in CNNs is to maximize the number of *separable filters*. A 2D-separable filter, denoted θ^{sep}, has a unitary rank*, and can be expressed as two successive 1D filters ($\theta^{sep}_{J\times 1}$ then $\theta^{sep}_{1\times K}$). Filter decomposition reduces the number of multiplications from $J\times K$ to $J+K$. This is illustrated in Figure 1.13, where the 3×3 averaging filter is separable, and can thus be decomposed into two successive one-dimensional convolutions.

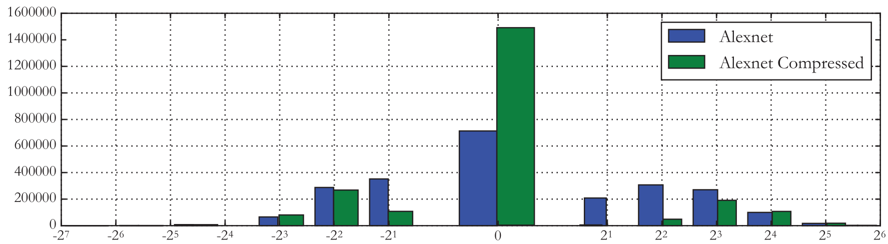


FIGURE 1.12 Histogram of conv weights in a compressed AlexNet model†.

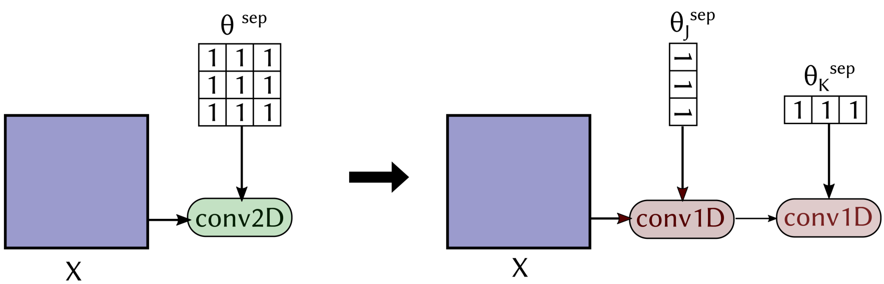


FIGURE 1.13 Example of a separable filter.

* Meaning that rank(θ^{sep}) = 1.

† Pruned filters treated as zero-valued weights.

The same concept expands to depth convolutions, where a separable filter requires $C+J+K$ multiplications instead of $C \times J \times K$ multiplications.

Nonetheless, only a small proportion of CNN filters are separable. To increase this proportion, a first approach is to force the convolution kernels to be separable by penalizing *high-rank filters* when training the network [98]. Alternatively, and after the training, the weights Θ of a given layer can be approximated into a small set of r *low-rank filters*. In this case, $r \times (C+J+K)$ multiplications are required to process a single depth convolution.

Finally, CNN computations can be reduced further by decomposing the weight matrix $\tilde{\Theta}$ through single-value decomposition (SVD). As demonstrated in the early works of [99], SVD greatly reduces the resource utilization of a given 2D-filter implantation. Moreover, SVD also finds its interest when processing FC layers and convolutions that employ the *im2col* method (cf Section 1.4.1). In a similar way to pruning, low rank approximation or SVD is followed by a fine-tuning in order to counterbalance the drop in classification accuracy.

1.6.2.3 FPGA Implementations

In FPGA implementations, SVD is applied on FC layer to significantly reduce the number of weights, such as in [28], where the authors derive a VGG16-SVD model that achieves 87.96% accuracy on ImageNet with 63% fewer parameters.

Alternatively, one can take advantage of the numerous research efforts given to accelerate Sparse GEMM on FPGA [100]. In this case, the challenge is to determine the optimal format of matrices that maximizes the chance to detect and skip zero computations, such compressed sparse column (CSC) or compressed sparse row (CSR) formats*. Based on this, Sze et al. [22] advocate the use of the CRC to process CNN. Indeed, this format requires lower memory bandwidths when the output matrix is smaller then the input, which is typically the case in CNNs where $N < CJK$, as in Figure 1.3b.

However, this efficiency of CRC format is valid only for extremely sparse matrices (typically with ≤1% of non-zeros), while in practice, pruned CNN matrices are not that sparse (typically, ≤4% – 80% of non-zeros). Therefore, works in [7] propose a *zero skip scheduler* that identifies zero elements and skips them in the scheduling of the MAC processing. As a consequence, the number of cycles required to compute the sparse GEMM is reduced. For AlexNet layers, the zero skip scheduler results in a 4× speedup. The same authors project a throughput of 12 TOPs for pruned CNN in the next Intel Stratix10 FPGAs, which outperforms the computational throughput of state-of-the-art GPU implementations by 10%.

1.7 CONCLUSIONS

In this chapter, a number of methods and tools have been compared that aim at porting convolutional neural networks onto FPGAs. At the network level, approximate computing and data-path optimization methods have been covered, while at

* This format represents a matrix by three one-dimensional arrays, that respectively contain nonzero values, row indices, and column indices.

the neuron level, the optimizations of convolutional and fully connected layers have been detailed and compared. All the different degrees of freedom offered by FPGAs (custom data types, local data streams, dedicated processors, etc.) are exploited by the presented methods. Moreover, algorithmic and data-path optimizations can and should be jointly implemented, resulting in additive hardware performance gains.

CNNs are by nature overparameterized and support particularly well approximate computing techniques such as weight pruning and fixed-point computation. Approximate computing already constitutes a key to CNN acceleration over hardware and will certainly continue driving the performance gains in the years to come.

BIBLIOGRAPHY

1. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
2. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, September 2014.
3. R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision - ICCV '15*, 2015, pp. 1440–1448.
4. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR '15*, 2015, pp. 3431–3440.
5. Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," arXiv preprint, vol. arXiv:1701, 2017.
6. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint, vol. arXiv:1409, pp. 1–14, 2014.
7. E. Nurvitadhi, S. Subhaschandra, G. Boudoukh, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. OngGeeHock, Y. T. Liew, K. Srivatsan, and D. Moss, "Can FPGAs beat GPUs in accelerating next-generation deep neural networks?" in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '17*, 2017, pp. 5–14.
8. Intel FPGA, "Intel Stratix 10 variable precision DSP blocks user guide," pp. 4–5, 2017.
9. G. Lacey, G. Taylor, and S. Areibi, "Deep learning on FPGAs: Past, present, and future," arXiv e-print, 2016.
10. S. I. Venieris, A. Kouris, and C.-S. Bouganis, "Toolflows for mapping convolutional neural networks on FPGAs," *ACM Computing Surveys*, vol. 51, no. 3, pp. 1–39, June 2018.
11. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
12. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems - NIPS'12*, 2012, p. 19.
13. D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.
14. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning - ICML '15*, F. Bach and D. Blei, Eds., vol. 37, 2015, pp. 448–456.

15. I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, “Binarized neural networks,” in *Advances in Neural Information Processing Systems – NIPS ’16*, February 2016, pp. 4107–4115.
16. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR ’15*, pp. 1–9, 2015.
17. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR ’16*, 2016, pp. 770–778.
18. J. Cong and B. Xiao, “Minimizing computation in convolutional neural networks,” in *Proceedings of the International Conference on Artificial Neural Networks - ICANN ’14*. Springer, 2014, pp. 281–290.
19. A. Canziani, A. Paszke, and E. Culurciello, “An analysis of deep neural network models for practical applications,” arXiv e-print, May 2016.
20. R. G. Shoup, “Parameterized convolution filtering in a field programmable gate array,” in *Proceedings of the International Workshop on Field Programmable Logic and Applications on More FPGAs*, 1994, pp. 274–280.
21. M. Horowitz, “Computing’s energy problem (and what we can do about it),” in *Proceedings of the IEEE International Solid-State Circuits - ISSCC ’14*. IEEE, February 2014, pp. 10–14.
22. V. Sze, Y.-H. Chen, T.-J. Yang, and J. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, December 2017.
23. Nvidia, “GPU-based deep learning inference: A performance and power analysis,” White Paper, 2015. https://www.nvidia.com/
24. S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, “cuDNN: Efficient primitives for deep learning,” arXiv e-print, 2014.
25. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the ACM International Conference on Multimedia – MM ’14*, 2014, pp. 675–678.
26. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, and G. Brain, “TensorFlow: A system for large-scale machine learning,” in *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation - OSDI ’16*, 2016, pp. 265–284.
27. K. Ovtcharov, O. Ruwase, J.-y. Kim, J. Fowers, K. Strauss, and E. Chung, “Accelerating deep convolutional neural networks using specialized hardware,” White Paper, pp. 3–6, February 2015.
28. J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, T. Tang, N. Xu, S. Song, Y. Wang, and H. Yang, “Going deeper with embedded FPGA platform for convolutional neural network,” in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA ’16*. ACM, 2016, pp. 26–35.
29. N. Suda, V. Chandra, G. Dasika, A. Mohanty, Y. Ma, S. Vrudhula, J.-s. Seo, and Y. Cao, “Throughput-optimized openCL-based FPGA accelerator for large-scale convolutional neural networks,” in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA ’16*, 2016, pp. 16–25.
30. K. Chellapilla, S. Puri, and P. Simard, “High performance convolutional neural networks for document processing,” in *Proceedings of the International Workshop on Frontiers in Handwriting Recognition – FHR ’06*. Suvisoft, October 2006.

31. U. Aydonat, S. O'Connell, D. Capalija, A. C. Ling, and G. R. Chiu, "An openCL(TM) deep learning accelerator on arria 10," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '17*, ACM, Ed. ACM, 2017, pp. 55–64.
32. R. DiCecco, G. Lacey, J. Vasiljevic, P. Chow, G. Taylor, and S. Areibi, "Caffeinated FPGAs: FPGA framework for convolutional neural networks," in *Proceedings of the International Conference on Field- Programmable Technology - FPT '16*, pp. 265–268, 2016.
33. C. Zhang and V. Prasanna, "Frequency domain acceleration of convolutional neural networks on CPU-FPGA shared memory system," in *Proceedings of the ACM/SIGDA International Symposium on Field- Programmable Gate Arrays - FPGA '17*, 2017, pp. 35–44.
34. J. H. Ko, B. A. Mudassar, T. Na, and S. Mukhopadhyay, "Design of an energy-efficient accelerator for training of convolutional neural networks using frequency-domain computation," in *Proceedings of the Annual Conference on Design Automation - DAC '17*, 2017.
35. S. Venieris and C. Bouganis, "FpgaConvNet: A framework for mapping convolutional neural networks on FPGAs," in *Proceedings of the IEEE Annual International Symposium on Field-Programmable Custom Computing Machines - FCCM '16*, 2016, pp. 40–47.
36. H. Sharma, J. Park, D. Mahajan, E. Amaro, J. K. Kim, C. Shao, A. Mishra, and H. Esmaeilzadeh, "From high-level deep neural models to FPGAs," in *Proceedings of the International Symposium on Microarchitecture - MICRO '16*, 2016, pp. 1–12.
37. H. Li, X. Fan, L. Jiao, W. Cao, X. Zhou, and L. Wang, "A high performance FPGA-based accelerator for large-scale convolutional neural networks," in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL '16*. IEEE, August 2016, pp. 1–9.
38. K. Abdelouahab, M. Pelcat, J. Serot, C. Bourrasset, and F. Berry, "Tactics to directly map CNN graphs on embedded FPGAs," *IEEE Embedded Systems Letters*, vol. 9, no. 4, pp. 113–116, December 2017.
39. C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '15*, ser. FPGA, 2015, pp. 161–170.
40. M. Motamedi, P. Gysel, V. Akella, and S. Ghiasi, "Design space exploration of FPGA-based deep convolutional neural networks," in *Proceedings of the Asia and South Pacific Design Automation Conference – ASPDAC '16*, January 2016, pp. 575–580.
41. M. Motamedi, P. Gysel, and S. Ghiasi, "PLACID: A platform for FPGA-based accelerator creation for DCNNs," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 13, no. 4, pp. 62:1–62:21, September 2017.
42. P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient learning," arXiv preprint, 2017.
43. T. Fujii, S. Sato, H. Nakahara, and M. Motomura, "An FPGA realization of a deep convolutional neural network using a threshold neuron pruning," in *Proceedings of the International Symposium on Applied Reconfigurable Computing – ARC '16*, vol. 9625, 2017, pp. 268–280.
44. S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFaNet: Training low bitwidth convolutional neural networks with low bitwidth gradients," arXiv e-print, 2016.
45. Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "FINN: A framework for fast, scalable binarized neural network inference," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '17*, 2017, pp. 65–74.
46. R. Andri, L. Cavigelli, D. Rossi, and L. Benini, "YodaNN: An ultralow power convolutional neural network accelerator based on binary weights," in *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 236–241, July 2016.

47. R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multicontext deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR '15*, 2015, pp. 1265–1274.
48. C. Zhang, D. Wu, J. Sun, G. Sun, G. Luo, and J. Cong, "Energy-efficient CNN implementation on a deeply pipelined FPGA cluster," in *Proceedings of the International Symposium on Low Power Electronics and Design - ISLPED '16*, 2016, pp. 326–331.
49. C. Zhang, Z. Fang, P. Zhou, P. Pan, and J. Cong, "Caffeine: Towards uniformed representation and acceleration for deep convolutional neural networks," in *Proceedings of the International Conference on Computer-Aided Design - ICCAD '16*. ACM, 2016, pp. 1–8.
50. J. Zhang and J. Li, "Improving the performance of openCL-based FPGA accelerator for convolutional neural network," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '17*, 2017, pp. 25–34.
51. Y. Guan, H. Liang, N. Xu, W. Wang, S. Shi, X. Chen, G. Sun, W. Zhang, and J. Cong, "FP-DNN: An automated framework for mapping deep neural networks onto FPGAs with RTL-HLS hybrid templates," in *Proceedings of the IEEE Annual International Symposium on Field- Programmable Custom Computing Machines - FCCM '17*. IEEE, 2017, pp. 152–159.
52. S. Winograd, *Arithmetic Complexity of Computations*. SIAM, 1980, vol. 33.
53. A. Lavin and S. Gray, "Fast algorithms for convolutional neural networks," arXiv e-print, vol. arXiv: 150, September 2015.
54. L. Lu, Y. Liang, Q. Xiao, and S. Yan, "Evaluating fast algorithms for convolutional neural networks on FPGAs," in *Proceedings of the IEEE Annual International Symposium on Field-Programmable Custom Computing Machines - FCCM '17*, 2017, pp. 101–108.
55. J. Bottleson, S. Kim, J. Andrews, P. Bindu, D. N. Murthy, and J. Jin, "ClCaffe: OpenCL accelerated caffe for convolutional neural networks," in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium – IPDPS '16*, 2016, pp. 50–57.
56. T. Highlander and A. Rodriguez, "Very efficient training of convolutional neural networks using fast fourier transform and overlap-and- add," arXiv preprint, pp. 1–9, 2016.
57. H. Zeng, R. Chen, C. Zhang, and V. Prasanna, "A framework for generating high throughput CNN implementations on FPGAs," in *Proceedings of the ACM/SIGDA International Symposium on Field- Programmable Gate Arrays - FPGA '18*. ACM Press, 2018, pp. 117–126.
58. M. Sankaradas, V. Jakkula, S. Cadambi, S. Chakradhar, I. Durdanovic, E. Cosatto, and H. P. Graf, "A massively parallel coprocessor for convolutional neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '17*. IEEE, July 2009, pp. 53–60.
59. C. Farabet, C. Poulet, J. Y. Han, Y. LeCun, D. R. Tobergte, and S. Curtis, "CNP: An FPGA-based processor for convolutional networks," in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL '09*, pp. 32–37, 2009.
60. S. Chakradhar, M. Sankaradas, V. Jakkula, and S. Cadambi, "A dynamically configurable coprocessor for convolutional neural networks," *ACM SIGARCH Computer Architecture News*, vol. 38, no. 3, pp. 247–257, June 2010.
61. V. Gokhale, J. Jin, A. Dundar, B. Martini, and E. Culurciello, "A 240 G-ops/s mobile coprocessor for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR '14*, June 2014, pp. 696–701.
62. M. Alwani, H. Chen, M. Ferdman, and P. Milder, "Fused-layer CNN accelerators," in *Proceedings of the Annual International Symposium on Microarchitecture - MICRO '16,* vol. 2016, December 2016.
63. Y. Ma, Y. Cao, S. Vrudhula, and J.-s. Seo, "Optimizing loop operation and dataflow in FPGA acceleration of deep convolutional neural networks," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '17*, 2017, pp. 45–54.

64. V. Gokhale, A. Zaidy, A. Chang, and E. Culurciello, "Snowflake: An efficient hardware accelerator for convolutional neural networks," in *Proceedings of the IEEE International Symposium on Circuits and Systems - ISCAS '17*. IEEE, May 2017, pp. 1–4.
65. Microsoft, "Microsoft unveils Project Brainwave for real-time AI," 2017. https://www.microsoft.com/en-us/research/blog/microsoft-unveils-project-brainwave/
66. S. Derrien and S. Rajopadhye, "Loop tiling for reconfigurable accelerators," in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL '01*. Springer, 2001, pp. 398–408.
67. R. Atul, L. Jongeun, and C. Kiyoung, "Efficient FPGA acceleration of convolutional neural networks using logical-3D compute array," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition - DATE '16*. IEEE, 2016, pp. 1393–1398.
68. Y. Ma, N. Suda, Y. Cao, J. S. Seo, and S. Vrudhula, "Scalable and modularized RTL compilation of Convolutional Neural Networks onto FPGA," in *Proceedings of the 26th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1–8. IEEE, 2016.
69. S. Williams, A. Waterman, and D. Patterson, "Roofline: An insightful visual performance model for multicore architectures," *Communications of the ACM*, vol. 52, no. 4, p. 65, April 2009.
70. Y. Ma, M. Kim, Y. Cao, S. Vrudhula, and J.-s. Seo, "End-to-end scalable FPGA accelerator for deep residual networks," in *Proceedings of the IEEE International Symposium on Circuits and Systems - ISCAS '17*. IEEE, May 2017, pp. 1–4.
71. Y. Ma, Y. Cao, S. Vrudhula, and J.-s. Seo, "An automatic RTL compiler for high-throughput FPGA implementation of diverse deep convolutional neural networks," in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL '17*. IEEE, September 2017, pp. 1–8.
72. Z. Liu, Y. Dou, J. Jiang, J. Xu, S. Li, Y. Zhou, and Y. Xu, "Throughput-optimized FPGA accelerator for deep convolutional neural networks," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 10, no. 3, pp. 1–23, 2017.
73. S. Mittal, "A survey of techniques for approximate computing," *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–33, March 2016.
74. S. Anwar, K. Hwang, and W. Sung, "Fixed point optimization of deep convolutional neural networks for object recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, April 2015.
75. S. Gupta, A. Agrawal, P. Narayanan, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proceedings of the International Conference on Machine Learning - ICML '15*, 2015, pp. 1737–1746.
76. D. Lin, S. Talathi, and V. Annapureddy, "Fixed point quantization of deep convolutional networks," in *Proceedings of the International Conference on Machine Learning - ICML '16*, 2016, pp. 2849–2858.
77. M. Courbariaux, Y. Bengio, and J.-P. David, "Training deep neural networks with low precision multiplications," arXiv e-print, December 2014.
78. S. Zhou, Y. Wang, H. Wen, Q. He, and Y. Zou, "Balanced quantization: An effective and efficient approach to quantized neural networks," *Journal of Computer Science and Technology*, vol. 32, pp. 667–682, 2017.
79. J.-P. David, K. Kalach, and N. Tittley, "Hardware complexity of modular multiplication and exponentiation," *IEEE Transactions on Computers*, vol. 56, no. 10, pp. 1308–1319, October 2007.
80. C. Farabet, B. Martini, B. Corda, P. Akselrod, E. Culurciello, and Y. LeCun, "NeuFlow: A runtime reconfigurable dataflow processor for vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR '11*. IEEE, June 2011, pp. 109–116.

81. P. Gysel, M. Motamedi, and S. Ghiasi, "Hardware-oriented approximation of convolutional neural networks," arXiv preprint, 2016, p. 8.
82. A. Kouris, S. I. Venieris, and C.-S. Bouganis, "CascadeCNN: Pushing the performance limits of quantisation in convolutional neural networks," in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL '18*, pp. 155–1557, July 2018.
83. D. Williamson, "Dynamically scaled fixed point arithmetic," in *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing Conference*. IEEE, 1991, pp. 315–318.
84. S. Guo, L. Wang, B. Chen, Q. Dou, Y. Tang, and Z. Li, "FixCaffe: Training CNN with low precision arithmetic operations by fixed point caffe," in *Proceedings of the International Workshop on Advanced Parallel Processing Technologies - APPT '17.* Springe, August 2017, pp. 38–50.
85. P. Gysel, J. Pimentel, M. Motamedi, and S. Ghiasi, "Ristretto: A framework for empirical study of resource-efficient inference in convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 1–6, 2018.
86. M. Courbariaux, Y. Bengio, and J.-P. David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in *Advances in Neural Information Processing Systems – NIPS '15*, 2015, pp. 3123–3131.
87. H. Nakahara, T. Fujii, and S. Sato, "A fully connected layer elimination for a binarizec convolutional neural network on an FPGA," in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL '17.* IEEE, September 2017, pp. 1–4.
88. I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *Journal of Machine Learning Research*, vol. 18, pp. 187:1–187:30, September 2018.
89. C. Zhu, S. Han, H. Mao, and W. J. Dally, "Trained ternary quantization," in *Proceedings of the International Conference on Learning Representations – ICLR '17*, December 2017.
90. R. Zhao, W. Song, W. Zhang, T. Xing, J.-H. Lin, M. Srivastava, R. Gupta, and Z. Zhang, "Accelerating binarized convolutional neural networks with software-programmable FPGAs," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '17*, 2017.
91. N. J. Fraser, Y. Umuroglu, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "Scaling binarized neural networks on reconfigurable logic," in *Proceedings of the Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures and Design Tools and Architectures for Multicore Embedded Computing Platforms - PARMA-DITAM '17.* ACM, 2017, pp. 25–30.
92. S. Liang, S. Yin, L. Liu, W. Luk, and S. Wei, "FP-BNN: Binarized neural network on FPGA," *Neurocomputing*, vol. 275, pp. 1072–1086, January 2018.
93. A. ProstBoucle, A. Bourge, F. Ptrot, H. Alemdar, N. Caldwell, and V. Leroy, "Scalable high-performance architecture for convolutional ternary neural networks on FPGA," in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL '17*, pp. 1–7, July 2017.
94. B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky, "Sparse convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR '15*, 2015, pp. 806–814.
95. S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems – NIPS '15*, 2015, pp. 1135–1143.

96. T.-J. Yang, Y.-H. Chen, and V. Sze, “Designing energy-efficient convolutional neural networks using energy-aware pruning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR ’17*, pp. 5687–5695, 2017.
97. S. Han, H. Mao, and W. J. Dally, “Deep compression - compressing deep neural networks with pruning, trained quantization and huffman coding,” in *Proceedings of the International Conference on Learning Representations – ICLR ’16*, 2016, pp. 1–13.
98. A. Sironi, B. Tekin, R. Rigamonti, V. Lepetit, and P. Fua, “Learning separable filters,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 94–106, 2015.
99. C. Bouganis, G. Constantinides, and P. Cheung, “A novel 2D filter design methodology for heterogeneous devices,” in *Proceedings of the Annual IEEE Symposium on Field-Programmable Custom Computing Machines - FCCM ’05*. IEEE, 2005, pp. 13–22.
100. R. Dorrance, F. Ren, and D. Markovi, “A scalable sparse matrix-vector multiplication kernel for energy-efficient sparse-blas on FPGAs,” in *Proceedings of the ACM/SIGDA International Symposium on Field- Programmable Gate Arrays - FPGA ’14*. ACM, 2014, pp. 161–170.

References

1. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
2. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, September 2014.
3. R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision - ICCV '15*, 2015, pp. 1440–1448.
4. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR '15*, 2015, pp. 3431–3440.
5. Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," arXiv preprint, vol. arXiv:1701, 2017.
6. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint, vol. arXiv:1409, pp. 1–14, 2014.
7. E. Nurvitadhi, S. Subhaschandra, G. Boudoukh, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. OngGeeHock, Y. T. Liew, K. Srivatsan, and D. Moss, "Can FPGAs beat GPUs in accelerating next-generation deep neural networks?" in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '17*, 2017, pp. 5–14.
8. Intel FPGA, "Intel Stratix 10 variable precision DSP blocks user guide," pp. 4–5, 2017.
9. G. Lacey, G. Taylor, and S. Areibi, "Deep learning on FPGAs: Past, present, and future," arXiv e-print, 2016.
10. S. I. Venieris, A. Kouris, and C.-S. Bouganis, "Toolflows for mapping convolutional neural networks on FPGAs," *ACM Computing Surveys*, vol. 51, no. 3, pp. 1–39, June 2018.
11. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
12. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems - NIPS'12*, 2012, p. 19.
13. D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.
14. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning - ICML '15*, F. Bach and D. Blei, Eds., vol. 37, 2015, pp. 448–456.
15. I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in Neural Information Processing Systems – NIPS '16*, February 2016, pp. 4107–4115.
16. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR '15*, pp. 1–9, 2015.

17. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR '16*, 2016, pp. 770–778.
18. J. Cong and B. Xiao, "Minimizing computation in convolutional neural networks," in *Proceedings of the International Conference on Artificial Neural Networks - ICANN '14*. Springer, 2014, pp. 281–290.
19. A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," arXiv e-print, May 2016.
20. R. G. Shoup, "Parameterized convolution filtering in a field programmable gate array," in *Proceedings of the International Workshop on Field Programmable Logic and Applications on More FPGAs*, 1994, pp. 274–280.
21. M. Horowitz, "Computing's energy problem (and what we can do about it)," in *Proceedings of the IEEE International Solid-State Circuits - ISSCC '14*. IEEE, February 2014, pp. 10–14.
22. V. Sze, Y.-H. Chen, T.-J. Yang, and J. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, December 2017.
23. Nvidia, "GPU-based deep learning inference: A performance and power analysis," White Paper, 2015. https://www.nvidia.com/
24. S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "cuDNN: Efficient primitives for deep learning," arXiv e-print, 2014.
25. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia – MM '14*, 2014, pp. 675–678.
26. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, and G. Brain, "TensorFlow: A system for large-scale machine learning," in *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation - OSDI '16*, 2016, pp. 265–284.
27. K. Ovtcharov, O. Ruwase, J.-y. Kim, J. Fowers, K. Strauss, and E. Chung, "Accelerating deep convolutional neural networks using specialized hardware," White Paper, pp. 3–6, February 2015.
28. J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, T. Tang, N. Xu, S. Song, Y. Wang, and H. Yang, "Going deeper with embedded FPGA platform for convolutional neural network," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '16*. ACM, 2016, pp. 26–35.
29. N. Suda, V. Chandra, G. Dasika, A. Mohanty, Y. Ma, S. Vrudhula, J.-s. Seo, and Y. Cao, "Throughput-optimized openCL-based FPGA accelerator for large-scale convolutional neural networks," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '16*, 2016, pp. 16–25.
30. K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for document processing," in *Proceedings of the International Workshop on Frontiers in Handwriting Recognition – FHR '06*. Suvisoft, October 2006.
31. U. Aydonat, S. O'Connell, D. Capalija, A. C. Ling, and G. R. Chiu, "An openCL(TM) deep learning accelerator on arria 10," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '17*, ACM, Ed. ACM, 2017, pp. 55–64.
32. R. DiCecco, G. Lacey, J. Vasiljevic, P. Chow, G. Taylor, and S. Areibi, "Caffeinated FPGAs: FPGA framework for convolutional neural networks," in *Proceedings of the International Conference on Field- Programmable Technology - FPT '16*, pp. 265–268, 2016.

33. C. Zhang and V. Prasanna, "Frequency domain acceleration of convolutional neural networks on CPU-FPGA shared memory system," in *Proceedings of the ACM/SIGDA International Symposium on Field- Programmable Gate Arrays - FPGA '17*, 2017, pp. 35–44.
34. J. H. Ko, B. A. Mudassar, T. Na, and S. Mukhopadhyay, "Design of an energy-efficient accelerator for training of convolutional neural networks using frequency-domain computation," in *Proceedings of the Annual Conference on Design Automation - DAC '17*, 2017.
35. S. Venieris and C. Bouganis, "FpgaConvNet: A framework for mapping convolutional neural networks on FPGAs," in *Proceedings of the IEEE Annual International Symposium on Field-Programmable Custom Computing Machines - FCCM '16*, 2016, pp. 40–47.
36. H. Sharma, J. Park, D. Mahajan, E. Amaro, J. K. Kim, C. Shao, A. Mishra, and H. Esmaeilzadeh, "From high-level deep neural models to FPGAs," in *Proceedings of the International Symposium on Microarchitecture - MICRO '16*, 2016, pp. 1–12.
37. H. Li, X. Fan, L. Jiao, W. Cao, X. Zhou, and L. Wang, "A high performance FPGA-based accelerator for large-scale convolutional neural networks," in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL '16*. IEEE, August 2016, pp. 1–9.
38. K. Abdelouahab, M. Pelcat, J. Serot, C. Bourrasset, and F. Berry, "Tactics to directly map CNN graphs on embedded FPGAs," *IEEE Embedded Systems Letters*, vol. 9, no. 4, pp. 113–116, December 2017.
39. C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '15*, ser. FPGA, 2015, pp. 161–170.
40. M. Motamedi, P. Gysel, V. Akella, and S. Ghiasi, "Design space exploration of FPGA-based deep convolutional neural networks," in *Proceedings of the Asia and South Pacific Design Automation Conference – ASPDAC '16*, January 2016, pp. 575–580.
41. M. Motamedi, P. Gysel, and S. Ghiasi, "PLACID: A platform for FPGA-based accelerator creation for DCNNs," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 13, no. 4, pp. 62:1–62:21, September 2017.
42. P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient learning," arXiv preprint, 2017.
43. T. Fujii, S. Sato, H. Nakahara, and M. Motomura, "An FPGA realization of a deep convolutional neural network using a threshold neuron pruning," in *Proceedings of the International Symposium on Applied Reconfigurable Computing – ARC '16*, vol. 9625, 2017, pp. 268–280.
44. S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFaNet: Training low bitwidth convolutional neural networks with low bitwidth gradients," arXiv e-print, 2016.
45. Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "FINN: A framework for fast, scalable binarized neural network inference," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '17*, 2017, pp. 65–74.
46. R. Andri, L. Cavigelli, D. Rossi, and L. Benini, "YodaNN: An ultralow power convolutional neural network accelerator based on binary weights," in *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 236–241, July 2016.
47. R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multicontext deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR '15*, 2015, pp. 1265–1274.

48. C. Zhang, D. Wu, J. Sun, G. Sun, G. Luo, and J. Cong, "Energy-efficient CNN implementation on a deeply pipelined FPGA cluster," in *Proceedings of the International Symposium on Low Power Electronics and Design - ISLPED '16*, 2016, pp. 326–331.
49. C. Zhang, Z. Fang, P. Zhou, P. Pan, and J. Cong, "Caffeine: Towards uniformed representation and acceleration for deep convolutional neural networks," in *Proceedings of the International Conference on Computer-Aided Design - ICCAD '16*. ACM, 2016, pp. 1–8.
50. J. Zhang and J. Li, "Improving the performance of openCL-based FPGA accelerator for convolutional neural network," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '17*, 2017, pp. 25–34.
51. Y. Guan, H. Liang, N. Xu, W. Wang, S. Shi, X. Chen, G. Sun, W. Zhang, and J. Cong, "FP-DNN: An automated framework for mapping deep neural networks onto FPGAs with RTL-HLS hybrid templates," in *Proceedings of the IEEE Annual International Symposium on Field- Programmable Custom Computing Machines - FCCM '17*. IEEE, 2017, pp. 152–159.
52. S. Winograd, *Arithmetic Complexity of Computations*. SIAM, 1980, vol. 33.
53. A. Lavin and S. Gray, "Fast algorithms for convolutional neural networks," arXiv e-print, vol. arXiv: 150, September 2015.
54. L. Lu, Y. Liang, Q. Xiao, and S. Yan, "Evaluating fast algorithms for convolutional neural networks on FPGAs," in *Proceedings of the IEEE Annual International Symposium on Field-Programmable Custom Computing Machines - FCCM '17*, 2017, pp. 101–108.
55. J. Bottleson, S. Kim, J. Andrews, P. Bindu, D. N. Murthy, and J. Jin, "ClCaffe: OpenCL accelerated caffe for convolutional neural networks," in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium – IPDPS '16*, 2016, pp. 50–57.
56. T. Highlander and A. Rodriguez, "Very efficient training of convolutional neural networks using fast fourier transform and overlap-and- add," arXiv preprint, pp. 1–9, 2016.
57. H. Zeng, R. Chen, C. Zhang, and V. Prasanna, "A framework for generating high throughput CNN implementations on FPGAs," in *Proceedings of the ACM/SIGDA International Symposium on Field- Programmable Gate Arrays - FPGA '18*. ACM Press, 2018, pp. 117–126.
58. M. Sankaradas, V. Jakkula, S. Cadambi, S. Chakradhar, I. Durdanovic, E. Cosatto, and H. P. Graf, "A massively parallel coprocessor for convolutional neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '17*. IEEE, July 2009, pp. 53–60.
59. C. Farabet, C. Poulet, J. Y. Han, Y. LeCun, D. R. Tobergte, and S. Curtis, "CNP: An FPGA-based processor for convolutional networks," in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL '09*, pp. 32–37, 2009.
60. S. Chakradhar, M. Sankaradas, V. Jakkula, and S. Cadambi, "A dynamically configurable coprocessor for convolutional neural networks," *ACM SIGARCH Computer Architecture News*, vol. 38, no. 3, pp. 247–257, June 2010.
61. V. Gokhale, J. Jin, A. Dundar, B. Martini, and E. Culurciello, "A 240 G-ops/s mobile coprocessor for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR '14*, June 2014, pp. 696–701.
62. M. Alwani, H. Chen, M. Ferdman, and P. Milder, "Fused-layer CNN accelerators," in *Proceedings of the Annual International Symposium on Microarchitecture - MICRO '16,* vol. 2016, December 2016.

63. Y. Ma, Y. Cao, S. Vrudhula, and J.-s. Seo, “Optimizing loop operation and dataflow in FPGA acceleration of deep convolutional neural networks,” in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA ’17*, 2017, pp. 45–54.
64. V. Gokhale, A. Zaidy, A. Chang, and E. Culurciello, “Snowflake: An efficient hardware accelerator for convolutional neural networks,” in *Proceedings of the IEEE International Symposium on Circuits and Systems - ISCAS ’17*. IEEE, May 2017, pp. 1–4.
65. Microsoft, “Microsoft unveils Project Brainwave for real-time AI,” 2017. https://www.microsoft.com/en-us/research/blog/microsoft-unveils-project-brainwave/
66. S. Derrien and S. Rajopadhye, “Loop tiling for reconfigurable accelerators,” in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL ’01*. Springer, 2001, pp. 398–408.
67. R. Atul, L. Jongeun, and C. Kiyoung, “Efficient FPGA acceleration of convolutional neural networks using logical-3D compute array,” in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition - DATE ’16*. IEEE, 2016, pp. 1393–1398.
68. Y. Ma, N. Suda, Y. Cao, J. S. Seo, and S. Vrudhula, “Scalable and modularized RTL compilation of Convolutional Neural Networks onto FPGA,” in *Proceedings of the 26th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1–8. IEEE, 2016.
69. S. Williams, A. Waterman, and D. Patterson, “Roofline: An insightful visual performance model for multicore architectures,” *Communications of the ACM*, vol. 52, no. 4, p. 65, April 2009.
70. Y. Ma, M. Kim, Y. Cao, S. Vrudhula, and J.-s. Seo, “End-to-end scalable FPGA accelerator for deep residual networks,” in *Proceedings of the IEEE International Symposium on Circuits and Systems - ISCAS ’17*. IEEE, May 2017, pp. 1–4.
71. Y. Ma, Y. Cao, S. Vrudhula, and J.-s. Seo, “An automatic RTL compiler for high-throughput FPGA implementation of diverse deep convolutional neural networks,” in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL ’17*. IEEE, September 2017, pp. 1–8.
72. Z. Liu, Y. Dou, J. Jiang, J. Xu, S. Li, Y. Zhou, and Y. Xu, “Throughput-optimized FPGA accelerator for deep convolutional neural networks,” *ACM Transactions on Reconfigurable Technology and Systems*, vol. 10, no. 3, pp. 1–23, 2017.
73. S. Mittal, “A survey of techniques for approximate computing,” *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–33, March 2016.
74. S. Anwar, K. Hwang, and W. Sung, “Fixed point optimization of deep convolutional neural networks for object recognition,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, April 2015.
75. S. Gupta, A. Agrawal, P. Narayanan, K. Gopalakrishnan, and P. Narayanan, “Deep learning with limited numerical precision,” in *Proceedings of the International Conference on Machine Learning - ICML ’15*, 2015, pp. 1737–1746.
76. D. Lin, S. Talathi, and V. Annapureddy, “Fixed point quantization of deep convolutional networks,” in *Proceedings of the International Conference on Machine Learning - ICML ’16*, 2016, pp. 2849–2858.
77. M. Courbariaux, Y. Bengio, and J.-P. David, “Training deep neural networks with low precision multiplications,” arXiv e-print, December 2014.
78. S. Zhou, Y. Wang, H. Wen, Q. He, and Y. Zou, “Balanced quantization: An effective and efficient approach to quantized neural networks,” *Journal of Computer Science and Technology*, vol. 32, pp. 667–682, 2017.

79. J.-P. David, K. Kalach, and N. Tittley, “Hardware complexity of modular multiplication and exponentiation,” *IEEE Transactions on Computers*, vol. 56, no. 10, pp. 1308–1319, October 2007.
80. C. Farabet, B. Martini, B. Corda, P. Akselrod, E. Culurciello, and Y. LeCun, “NeuFlow: A runtime reconfigurable dataflow processor for vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR ’11*. IEEE, June 2011, pp. 109–116.
81. P. Gysel, M. Motamedi, and S. Ghiasi, “Hardware-oriented approximation of convolutional neural networks,” arXiv preprint, 2016, p. 8.
82. A. Kouris, S. I. Venieris, and C.-S. Bouganis, “CascadeCNN: Pushing the performance limits of quantisation in convolutional neural networks,” in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL ’18*, pp. 155–1557, July 2018.
83. D. Williamson, “Dynamically scaled fixed point arithmetic,” in *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing Conference*. IEEE, 1991, pp. 315–318.
84. S. Guo, L. Wang, B. Chen, Q. Dou, Y. Tang, and Z. Li, “FixCaffe: Training CNN with low precision arithmetic operations by fixed point caffe,” in *Proceedings of the International Workshop on Advanced Parallel Processing Technologies - APPT ’17*. Springe, August 2017, pp. 38–50.
85. P. Gysel, J. Pimentel, M. Motamedi, and S. Ghiasi, “Ristretto: A framework for empirical study of resource-efficient inference in convolutional neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 1–6, 2018.
86. M. Courbariaux, Y. Bengio, and J.-P. David, “BinaryConnect: Training deep neural networks with binary weights during propagations,” in *Advances in Neural Information Processing Systems – NIPS ’15*, 2015, pp. 3123–3131.
87. H. Nakahara, T. Fujii, and S. Sato, “A fully connected layer elimination for a binarizec convolutional neural network on an FPGA,” in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL ’17*. IEEE, September 2017, pp. 1–4.
88. I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, “Quantized neural networks: Training neural networks with low precision weights and activations,” *Journal of Machine Learning Research*, vol. 18, pp. 187:1–187:30, September 2018.
89. C. Zhu, S. Han, H. Mao, and W. J. Dally, “Trained ternary quantization,” in *Proceedings of the International Conference on Learning Representations – ICLR ’17*, December 2017.
90. R. Zhao, W. Song, W. Zhang, T. Xing, J.-H. Lin, M. Srivastava, R. Gupta, and Z. Zhang, “Accelerating binarized convolutional neural networks with software-programmable FPGAs,” in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA ’17*, 2017.
91. N. J. Fraser, Y. Umuroglu, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, “Scaling binarized neural networks on reconfigurable logic,” in *Proceedings of the Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures and Design Tools and Architectures for Multicore Embedded Computing Platforms - PARMA-DITAM ’17*. ACM, 2017, pp. 25–30.
92. S. Liang, S. Yin, L. Liu, W. Luk, and S. Wei, “FP-BNN: Binarized neural network on FPGA,” *Neurocomputing*, vol. 275, pp. 1072–1086, January 2018.
93. A. ProstBoucle, A. Bourge, F. Ptrot, H. Alemdar, N. Caldwell, and V. Leroy, “Scalable high-performance architecture for convolutional ternary neural networks on FPGA,” in *Proceedings of the International Conference on Field Programmable Logic and Applications - FPL ’17*, pp. 1–7, July 2017.

94. B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky, "Sparse convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR '15*, 2015, pp. 806–814.
95. S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems – NIPS '15*, 2015, pp. 1135–1143.
96. T.-J. Yang, Y.-H. Chen, and V. Sze, "Designing energy-efficient convolutional neural networks using energy-aware pruning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR '17*, pp. 5687–5695, 2017.
97. S. Han, H. Mao, and W. J. Dally, "Deep compression - compressing deep neural networks with pruning, trained quantization and huffman coding," in *Proceedings of the International Conference on Learning Representations – ICLR '16*, 2016, pp. 1–13.
98. A. Sironi, B. Tekin, R. Rigamonti, V. Lepetit, and P. Fua, "Learning separable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 94–106, 2015.
99. C. Bouganis, G. Constantinides, and P. Cheung, "A novel 2D filter design methodology for heterogeneous devices," in *Proceedings of the Annual IEEE Symposium on Field-Programmable Custom Computing Machines - FCCM '05*. IEEE, 2005, pp. 13–22.
100. R. Dorrance, F. Ren, and D. Markovi, "A scalable sparse matrix-vector multiplication kernel for energy-efficient sparse-blas on FPGAs," in *Proceedings of the ACM/SIGDA International Symposium on Field- Programmable Gate Arrays - FPGA '14*. ACM, 2014, pp. 161–170.

1. G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
2. Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
3. I. Arel, D. C. Rose, T. P. Karnowski, et al., "Deep machine learning-a new frontier in artificial intelligence research," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, 2010.
4. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
5. G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
6. T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8614–8618, IEEE, 2013.
7. G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.
8. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.
9. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, pp. 3104–3112, 2014.
10. R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 129–136, 2011.

11. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.
12. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.
13. L. He, G. Wang, and Z. Hu, “Learning depth from single images with deep neural network embedding focal length,” *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4676–4689, 2018.
14. J. Gao, J. Yang, G. Wang, and M. Li, “A novel feature extraction method for scene recognition based on centered convolutional restricted boltzmann machines,” *Neurocomputing*, vol. 214, pp. 708–717, 2016.
15. M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, “Deep learning applications and challenges in big data analytics,” *Journal of Big Data*, vol. 2, no. 1, p. 1, 2015.
16. G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, “A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.
17. X. Mo, K. Tao, Q. Wang, and G. Wang, “An efficient approach for polyps detection in endoscopic videos based on faster R-CNN,” in *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 3929–3934, IEEE, 2018.
18. A. Elgammal, B. Liu, M. Elhoseiny, and M. Mazzone, “CAN: Creative adversarial networks, generating “art” by learning about styles and deviating from style norms,” arXiv preprint arXiv:1706.07068, 2017.
19. W. Xu, S. Keshmiri, and G. Wang, “Adversarially approximated autoencoder for image generation and manipulation,” *IEEE Transactions on Multimedia*, doi:10.1109/TMM.2019.2898777, 2019.
20. W. Xu, S. Keshmiri, and G. Wang, “Toward learning a unified many-to-many mapping for diverse image translation,” *Pattern Recognition*, doi:10.1016/j.patcog.2019.05.017, 2019.
21. W. Ma, Y. Wu, Z. Wang, and G. Wang, “MDCN: Multi-scale, deep inception convolutional neural networks for efficient object detection,” in *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 2510–2515, IEEE, 2018.
22. Z. Zhang, Y. Wu, and G. Wang, “Bpgrad: Towards global optimality in deep learning via branch and pruning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3301–3309, 2018.
23. L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, “Deep learning for generic object detection: A survey,” arXiv preprint arXiv:1809.02165, 2018.
24. F. Cen and G. Wang, “Dictionary representation of deep features for occlusion-robust face recognition,” *IEEE Access*, vol. 7, pp. 26595–26605, 2019.
25. M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
26. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European Conference on Computer Vision*, pp. 740–755, Springer, 2014.
27. S. P. Bharati, S. Nandi, Y. Wu, Y. Sui, and G. Wang, “Fast and robust object tracking with adaptive detection,” in *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 706–713, IEEE, 2016.
28. Y. Wu, Y. Sui, and G. Wang, “Vision-based real-time aerial object localization and tracking for UAV sensing system,” *IEEE Access*, vol. 5, pp. 23969–23978, 2017.

29. S. P. Bharati, Y. Wu, Y. Sui, C. Padgett, and G. Wang, "Real-time obstacle detection and tracking for sense-and-avoid mechanism in UAVs," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 185–197, 2018.
30. Y. Wei, X. Pan, H. Qin, W. Ouyang, and J. Yan, "Quantization mimic: Towards very tiny CNN for object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 267–283, 2018.
31. K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, et al., "T-CNN: Tubelets with convolutional neural networks for object detection from videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2896–2907, 2018.
32. W. Chu and D. Cai, "Deep feature based contextual model for object detection," *Neurocomputing*, vol. 275, pp. 1035–1042, 2018.
33. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.
34. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.
35. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
36. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2018.
37. Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2det: A single-shot object detector based on multi-level feature pyramid network," *CoRR*, vol. abs/1811.04533, 2019.
38. P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.
39. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," arXiv preprint arXiv:1312.6229, 2013.
40. J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.
41. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.
42. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*, pp. 21–37, Springer, 2016.
43. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.
44. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
45. X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.
46. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

47. Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, “Dual path networks,” in *Advances in Neural Information Processing Systems*, pp. 4467–4475, 2017.
48. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017.
49. D. G. Lowe, “Object recognition from local scale-invariant features,” in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 99, no. 2, pp. 1150–1157, IEEE, 1999.
50. R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, 2015.
51. J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in *Advances in Neural Information Processing Systems*, pp. 379–387, 2016.
52. Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving into high quality object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, 2018.
53. J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271, 2017.
54. S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 448–456, 2015.
55. C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, “DSSD: Deconvolutional single shot detector,” arXiv preprint arXiv:1701.06659, 2017.
56. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.
57. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” arXiv preprint arXiv:1902.09630, 2019.
58. P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu, “Scale-transferrable object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 528–537, 2018.
59. S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, “Single-shot refinement neural network for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4203–4212, 2018.
60. B. Singh and L. S. Davis, “An analysis of scale invariance in object detection snip,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3578–3587, 2018.
61. P. Zhu, L. Wen, X. Bian, L. Haibin, and Q. Hu, “Vision meets drones: A challenge,” arXiv preprint arXiv:1804.07437, 2018.
62. P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, Q. Nie, H. Cheng, C. Liu, X. Liu, et al., “VisDrone-DET2018: The vision meets drone object detection in image challenge results,” In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 437–468, 2018.
63. J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” in *Advances in Neural Information Processing Systems*, pp. 2546–2554, 2011.
64. B. Wu and R. Nevatia, “Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors,” in *Tenth IEEE International Conference on Computer Vision*, vol. 1, pp. 90–97, 2005.

65. Bradley, David M. “Learning in modular systems,” No. CMU-RI-TR-09-26. CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 2010.
1. B. C. Ko, K.-H. Cheong, and J.-Y. Nam, “Fire detection based on vision sensor and support vector machines,” *Fire Safety Journal*, vol. 44, pp. 322–329, 2009.
2. V. D. Nguyen, H. Van Nguyen, D. T. Tran, S. J. Lee, and J. W. Jeon, “Learning framework for robust obstacle detection, recognition, and tracking,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 1633–1646, 2017.
3. I. Mehmood, M. Sajjad, and S. W. Baik, “Mobile-cloud assisted video summarization framework for efficient management of remote sensing data generated by wireless capsule sensors,” *Sensors*, vol. 14, pp. 17112–17145, 2014.
4. K. Muhammad, M. Sajjad, M. Y. Lee, and S. W. Baik, “Efficient visual attention driven framework for key frames extraction from hysteroscopy videos,” *Biomedical Signal Processing and Control*, vol. 33, pp. 161–168, 2017.
5. R. Hamza, K. Muhammad, Z. Lv, and F. Titouna, “Secure video summarization framework for personalized wireless capsule endoscopy,” *Pervasive and Mobile Computing*, vol. 41, pp. 436–450, 2017/10/01/ 2017.
6. K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H. H. G. Wang, and S. W. Baik, “Secure surveillance framework for IoT systems using probabilistic image encryption,” *IEEE Transactions on Industrial Informatics*, vol. 14(8), pp. 3679–3689, 2018.
7. K. Muhammad, J. Ahmad, and S. W. Baik, “Early fire detection using convolutional neural networks during surveillance for effective disaster management,” *Neurocomputing*, vol. 288, pp. 30–42, 2018.
8. C. Thou-Ho, W. Ping-Hsueh, and C. Yung-Chuen, “An early fire-detection method based on image processing,” in *2004 International Conference on Image Processing, 2004. ICIP ‘04*, vol. 3, 2004, pp. 1707–1710.
9. http://www.bbc.com/news/world-asia-42828023 (Visited 31 January, 2018, 9 AM).
10. P. Foggia, A. Saggese, and M. Vento, “Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25(9), pp. 1545–1556, 2015.
11. B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay, and A. E. Cetin, “Computer vision based method for real-time fire and flame detection,” *Pattern Recognition Letters*, vol. 27(1), pp. 49–58, 2006.
12. D. Han, and B. Lee, “Development of early tunnel fire detection algorithm using the image processing,” in *International Symposium on Visual Computing*, 2006, pp. 39–48.
13. G. Marbach, M. Loepfe, and T. Brupbacher, “An image processing technique for fire detection in video images,” *Fire Safety Journal*, vol. 41(4), pp. 285–289, 2006.
14. T. Celik, and H. Demirel, “Fire detection in video sequences using a generic color model,” *Fire Safety Journal*, vol. 44(2), pp. 147–158, 2009.
15. A. Rafiee, R. Dianat, M. Jamshidi, R. Tavakoli, and S. Abbaspour, “Fire and smoke detection using wavelet analysis and disorder characteristics,” in *2011 3rd International Conference on Computer Research and Development*, 2011, pp. 262–265.
16. Y. H. Habiboğlu, O. Günay, and A. E. Çetin, “Covariance matrix-based fire and flame detection method in video,” *Machine Vision and Applications*, vol. 23(6), pp. 1103–1113, 2012.
17. A. Sorbara, E. Zereik, M. Bibuli, G. Bruzzone, and M. Caccia, “Low cost optronic obstacle detection sensor for unmanned surface vehicles,” in *2015 IEEE Sensors Applications Symposium (SAS)*, 2015, pp. 1–6.
18. I. Kolesov, P. Karasev, A. Tannenbaum, and E. Haber, “Fire and smoke detection in video with optimal mass transport based optical flow and neural networks,” in *2010 IEEE International Conference on Image Processing*, 2010, pp. 761–764.

19. H. J. G. Haynes, "Fire loss in the United States during 2015," http://www.nfpa.org/, 2016.
20. K. Muhammad, T. Hussain, and S. W. Baik, "Efficient CNN based summarization of surveillance videos for resource-constrained devices," *Pattern Recognition Letters*, 2018.
21. M. Sajjad, S. Khan, T. Hussain, K. Muhammad, A. K. Sangaiah, A. Castiglione, et al., "CNN-based anti-spoofing two-tier multi-factor authentication system," *Pattern Recognition Letters*, 126(2019): 123–131.
22. M. Hassaballah, A. A. Abdelmgeid, and H. A. Alshazly, "Image features detection, description and matching," in *Image Feature Detectors and Descriptors*, Awad, A. I., Hassaballah, M., (Eds.): Springer, 2016, pp. 11–45.
23. A. I. Awad, and M. Hassaballah, *Image Feature Detectors and Descriptors*. Studies in Computational Intelligence. Springer International Publishing, Cham, 2016.
24. F. U. M. Ullah, A. Ullah, K. Muhammad, I. U. Haq, and S. W. Baik, "Violence detection using spatiotemporal features with 3D Convolutional Neural Network," *Sensors*, vol. 19(11), p. 2472, 2019.
25. M. Sajjad, S. Khan, Z. Jan, K. Muhammad, H. Moon, J. T. Kwak, et al., "Leukocytes classification and segmentation in microscopic blood smear: A resource-aware healthcare service in smart cities," *IEEE Access*, vol. 5, pp. 3475–3489, 2017.
26. I. U. Haq, K. Muhammad, A. Ullah, and S. W. Baik, "DeepStar: Detecting starring characters in movies," *IEEE Access*, 7(2019): 9265–9272.
27. M. Hassaballah, H. A. Alshazly, and A. A. Ali, "Ear recognition using local binary patterns: A comparative experimental study," *Expert Systems with Applications*, vol. 118, pp. 182–200, 2019.
28. A. I. Awad, and K. Baba, "Singular point detection for efficient fingerprint classification," *International Journal on New Computer Architectures and Their Applications (IJNCAA)*, vol. 2, pp. 1–7, 2012.
29. A. Ullah, K. Muhammad, J. D. Ser, S. W. Baik, and V. Albuquerque, "Activity recognition using temporal optical flow convolutional features and multi-layer LSTM," *IEEE Transactions on Industrial Electronics*, vol. 66(12), pp. 9692–9702, 2019.
30. A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep Bi-directional LSTM with CNN features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.
31. M. Sajjad, S. Khan, K. Muhammad, W. Wu, A. Ullah, and S. W. Baik, "Multi-grade brain tumor classification using deep CNN with extensive data augmentation," *Journal of Computational Science*, vol. 30, pp. 174–182, 2019.
32. J. Ahmad, K. Muhammad, J. Lloret, and S. W. Baik, "Efficient conversion of deep features to compact binary codes using Fourier decomposition for multimedia big data," *IEEE Transactions on Industrial Informatics*, vol. 14(7), pp. 3205–3215, 2018.
33. J. Ahmad, K. Muhammad, S. Bakshi, and S. W. Baik, "Object-oriented convolutional features for fine-grained image retrieval in large surveillance datasets," *Future Generation Computer Systems*, vol. 81, pp. 314–330, 2018.
34. S. Frizzi, R. Kaabi, M. Bouchouicha, J. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 877–882.
35. J. Sharma, O.-C. Granmo, M. Goodwin, and J. T. Fidje, "Deep convolutional neural networks for fire detection in images," in *International Conference on Engineering Applications of Neural Networks*, 2017, pp. 183–193.
36. Z. Zhong, M. Wang, Y. Shi, and W. Gao, "A convolutional neural network-based flame detection method in video sequence," *Signal, Image and Video Processing*, vol. 12(8), pp. 1619–1627, 2018.

37. K. Muhammad, S. Khan, M. Elhoseny, S. H. Ahmed, and S. W. Baik, "Efficient fire detection for uncertain surveillance environment," *IEEE Transactions on Industrial Informatics*, vol. 15(5), pp. 3113–3122, 2019.
38. S. Khan, K. Muhammad, S. Mumtaz, S. W. Baik, and V. H. C. d. Albuquerque, "Energy-efficient deep CNN for smoke detection in foggy IoT environment," *IEEE Internet of Things Journal*, vol. 6(6), pp. 9237–9245, 2019.
39. K. Muhammad, S. Khan, V. Palade, I. Mehmood, and V. H. C. D. Albuquerque, "Edge intelligence-assisted smoke detection in foggy surveillance environments," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.
40. K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," *IEEE Access*, vol. 6, pp. 18174–18183, 2018.
41. K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49(7), pp. 1419–1434, 2018.
42. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.
43. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.
44. F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 1MB model size," arXiv preprint arXiv:1602.07360, 2016.
45. K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49(7), pp. 1419–1434, 2018.
46. K. Wattanachote, and T. K. Shih, "Automatic dynamic texture transformation based on a new motion coherence metric," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26(10), pp. 1805–1820, 2015.
47. D. Y. Chino, L. P. Avalhais, J. F. Rodrigues, and A. J. Traina, "BoWFire: Detection of fire in still images by integrating pixel color and texture analysis," in *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, 2015, pp. 95–102.
48. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, et al., "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014, pp. 675–678.
49. K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49(7), pp. 1419–1434, 2018.
50. K. Muhammad, J. Ahmad, M. Sajjad, and S. W. Baik, "Visual saliency models for summarization of diagnostic hysteroscopy videos in healthcare systems," *SpringerPlus*, vol. 5(1), p. 1495, 2016.
51. R. Di Lascio, A. Greco, A. Saggese, and M. Vento, "Improving fire detection reliability by a combination of videoanalytics," in *International Conference Image Analysis and Recognition*, 2014, pp. 477–484.
52. S. J. Pan, and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22(10), pp. 1345–1359, 2010.

53. S. Rudz, K. Chetehouna, A. Hafiane, H. Laurent, and O. Séro-Guillaume, "Investigation of a novel image segmentation method dedicated to forest fire applicationsThis paper is dedicated to the memory of Dr Olivier Séro-Guillaume (1950–2013), CNRS Research Director," *Measurement Science and Technology*, vol. 24(7), p. 075403, 2013.
54. L. Rossi, M. Akhloufi, and Y. Tison, "On the use of stereovision to develop a novel instrumentation system to extract geometric fire fronts characteristics," *Fire Safety Journal*, vol. 46(1–2), pp. 9–20, 2011.
1. K. Nandakumar, A. Ross, and A. K. Jain, "Introduction to multibiometrics," in *Appeared in Proc of the 15th European Signal Processing Conference (EUSIPCO) (Poznan Poland)*, 2007, pp. 271–292.
2. A. S. Al-Waisy, "Ear identification system based on multi-model approach," *International Journal of Electronics Communication and Computer Engineering*, vol. 3, no. 5, pp. 2278–4209, 2012.
3. M. S. Al-ani and A. S. Al-Waisy, "Milti-view face datection based on kernel principal component analysis and kernel," *International Journal on Soft Computing (IJSC)*, vol. 2, no. 2, pp. 1–13, 2011.
4. A. S. Al-Waisy, R. Qahwaji, S. Ipson, and S. Al-Fahdawi, "A fast and accurate Iris localization technique for healthcare security system," in *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 1028–1034.
5. R. Jafri and H. R. Arabnia, "A survey of face recognition techniques," *Journal of Information Processing Systems*, vol. 5, no. 2, pp. 41–68, 2009.
6. C. Ding, J. Choi, D. Tao, and L. S. Davis, "Multi-directional multi-level dual-cross patterns for robust face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 518–531, 2016.
7. A. S. Al-Waisy, R. Qahwaji, S. Ipson, and S. Al-Fahdawi, "A multimodal deep learning framework using local feature representations for face recognition," *Machine Vision and Applications*, vol. 29, no. 1, pp. 35–54, 2017.
8. A. S. Al-Waisy, R. Qahwaji, S. Ipson, and S. Al-fahdawi, "A multimodal biometric system for personal identification based on deep learning approaches," in *Seventh International Conference on Emerging Security Technologies (EST)*, 2017, pp. 163–168.
9. A. Ross, K. Nandakumar, and J. K. Anil, "Handbook of multibiometrics," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1689–1699, 2006.
10. A. Ross and A. K. Jain, "Multimodal biometrics : An overview," in *12th European Signal Processing Conference IEEE*, 2004, pp. 1221–1224.
11. A. Lumini and L. Nanni, "Overview of the combination of biometric matchers," *Information Fusion*, vol. 33, pp. 71–85, 2017.
12. L. Deng and D. Yu, "Deep learning methods and applications," *Signal Processing*, vol. 28, no. 3, pp. 198–387, 2013.
13. J. Liu, C. Fang, and C. Wu, "A fusion face recognition approach based on 7-layer deep learning neural network," *Journal of Electrical and Computer Engineering*, vol. 2016, pp. 1–7, 2016.
14. P. Fousek, S. Rennie, P. Dognin, and V. Goel, "Direct product based deep belief networks for automatic speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2013, pp. 3148–3152.
15. H. Lee, P. Pham, Y. Largman, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in Neural Information Processing Systems Conference*, 2009, pp. 1096–1104.
16. R. Sarikaya, G. E. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," *IEEE Transactions on Audio, Speech and Language Process.*, vol. 22, no. 4, pp. 778–784, 2014.

17. C. Ding and D. Tao, "A comprehensive survey on pose-invariant face recognition," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 7, no. 3, pp. 1–40, 2016.
18. S. Biswas, G. Aggarwal, P. J. Flynn, and K. W. Bowyer, "Pose-robust recognition of low-resolution face images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 3037–3049, 2013.
19. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
20. L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces.," *Journal of the Optical Society of America. A, Optics and Image Science*, vol. 4, no. 3. pp. 519–524, 1987.
21. K. Meethongjan and D. Mohamad, "A summary of literature review : Face recognition," *Review Literature and Arts of the Americas*, vol. 2007, no. July, pp. 1–12, 2007.
22. M. Fischer, H. K. Ekenel, and R. Stiefelhagen, "Analysis of partial least squares for pose-invariant face recognition," *in IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2012, pp. 331–338.
23. T. Berg and P. Belhumeur, "Tom-vs-Pete classifiers and identity-preserving alignment for face verification," in *Procedings of the British Machine Vision Conference*, 2012, pp. 129.1–129.11.
24. A. Li, S. Shan, and W. Gao, "Coupled bias – variance tradeoff for cross-pose face recognition," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 305–315, 2012.
25. Z. Zhu, P. Luo, X. Wang, and X. Tang, "Multi-view perceptron : A deep model for learning face identity and view representations," *Advances in Neural Information Processing Systems*, pp. 1–9, 2014.
26. A. S. Al-Waisy, "Detection and recognition of human faces based on hybrid techniques," *International Journal of Applied Computing (IJAC)*, vol. 5, no. 2, pp. 115–126, 2012.
27. M. S. Al-Ani and A. S. Al-waisy, "Face recognition approach based on wavelet-curvelet technique," *Signal & Image Processing: An International Journal (SIPIJ)*, vol. 3, no. 2, pp. 21–31, 2012.
28. Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," *Advances in Neural Information Processing Systems*, pp. 1988–1996, 2014.
29. L. Deng, "Three classes of deep learning architectures and their applications: A tutorial survey," *APSIPA Transactions on Signal and Information Processing*, 2013.
30. Y. Sun, X. Wang, and X. Tang, "Hybrid deep learning for face verification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1489–1496.
31. Y. Taigman, M. A. Ranzato, T. Aviv, and M. Park, "DeepFace: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.
32. C. Ding and D. Tao, "Robust face recognition via multimodal deep face representation," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2049–2058, 2015.
33. J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *in Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 209–216.
34. J. Hu, J. Lu, and Y. P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1875–1882.

35. Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen, "Fusing robust face region descriptors via multiple metric learning for face recognition in the wild," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3554–3561.
36. M. Guillaumin, J. Verbeek, C. Schmid, M. Guillaumin, J. Verbeek, C. Schmid, M. Guillaumin, J. Verbeek, C. Schmid, and L. J. Kuntzmann, "Is that you ? Metric learning approaches for face identification To cite this version," in *IEEE 12th International Conference on Computer Vision*, 2009, pp. 498–505.
37. M. Kostinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2288–2295.
38. H. V Nguyen and L. Bai, "Cosine Similarity Metric Learning for Face Verification," in *Asian Conference on Computer Vision*. Springer Berlin Heidelberg, 2011, pp. 1–12.
39. C. Peng, X. Gao, N. Wang, and J. Li, "Graphical representation for heterogeneous face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 301–312, 2017.
40. C. Peng, N. Wang, J. Li, and X. Gao, "DLFace: Deep local descriptor for cross-modality face recognition," *Pattern Recognition*, vol. 90, pp. 161–171, 2019.
41. C. Peng, X. Gao, S. Member, N. Wang, and D. Tao, "Multiple representations-based face sketch – photo synthesis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 11, pp. 2201–2215, 2016.
42. Y. Gao, J. Ma, and A. L. Yuille, "Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2545–2560, 2017.
43. M. Hassaballah and S. Aly, "Face recognition: Challenges, achievements and future directions," *IET Computer Vision*, vol. 9, no. 4, pp. 614–626, 2015.
44. R. Lopes and N. Betrouni, "Fractal and multifractal analysis: A review, *Medical Image Analysis*, vol. 13, no. 4, pp. 634–649, 2009.
45. B. Mandelbrot, *The Fractal Geometry of Nature*. Library of Congress Cataloging in Publication Data, United States of America, 1983.
46. B. Mandelbrot, "Self-affinity and fractal dimension," *Physica Scripta*, vol. 32, 1985, pp. 257–260.
47. K. Lin, K. Lam, and W. Siu, "Locating the human eye using fractal dimensions," in *Proceedings of International Conference on Image Processing*, 2001, pp. 1079–1082.
48. M. H. Farhan, L. E. George, and A. T. Hussein, "Fingerprint identification using fractal geometry," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 1, pp. 52–61, 2014.
49. G. Hinton, "A practical guide to training restricted boltzmann machines a practical guide to training restricted boltzmann machines," *Neural Networks: Tricks of the Trade*. Springer, Berlin, Heidelberg, 2010, pp. 599–619.
50. H. Larochelle and Y. Bengio, "Classification using discriminative restricted Boltzmann machines," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 536–543.
51. B. Abibullaev, J. An, S. H. Jin, S. H. Lee, and J. Il Moon, "Deep machine learning a new frontier in artificial intelligence research," *Medical Engineering and Physics*, vol. 35, no. 12, pp. 1811–1818, 2013.
52. G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
53. H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, "Face recognition using convolutional neural network and simple logistic classifier," *Soft Computing in Industrial Applications*. Springer International Publishing, pp. 197–207, 2014.

54. Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends® in Machine Learning,* vol. 2, no. 1, pp. 1–127, 2009.
55. A. S. Al-Waisy, R. Qahwaji, S. Ipson, S. Al-Fahdawi, and T. A. M. Nagem, "A multi-biometric iris recognition system based on a deep learning approach," *Pattern Analysis and Applications*, vol. 21, no. 3, pp. 783–802, 2018.
56. P. Viola, O. M. Way, and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.
57. A. R. Chowdhury, T.-Y. Lin, S. Maji, and E. Learned-Miller, "One-to-many face recognition with bilinear CNNs," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–9.
58. D. Wang, C. Otto, and A. K. Jain, "Face search at scale: 80 million gallery," arXiv Prepr. arXiv1507.07242, pp. 1–14, 2015.
59. G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
60. Y. Yin, L. Liu, and X. Sun, "SDUMLA-HMT: A multimodal biometric database," in *Chinese Conference Biometric Recognition*, Springer-Verlag Berlin Heidelberg, pp. 260–268, 2011.
61. P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, 2005, vol. I, pp. 947–954.
62. L. Lenc and P. Král, "Unconstrained facial images: Database for face recognition under real-world conditions," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9414, pp. 349–361, 2015.
63. G. B. Huang, M. Mattar, T. Berg, and E. Learned-miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," in *Technical Report 07-49, University of Massachusetts, Amherst*, 2007, pp. 1–14.
64. A. S. Al-Waisy, R. Qahwaji, S. Ipson, and S. Al-Fahdawi, "A robust face recognition system based on curvelet and fractal dimension transforms," in *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 548–555.
65. W. R. Schwartz, H. Guo, and L. S. Davis, "A robust and scalable approach to face identification," in *European Conference on Computer Vision*, 2010, pp. 476–489.
66. M. Y. Shams, A. S. Tolba, and S. H. Sarhan, "A vision system for multi-view face recognition," *International Journal of Circuits, Systems and Signal Processing*, vol. 10, pp. 455–461, 2016.
67. J. Li, T. Qiu, C. Wen, K. Xie, and F.-Q. Wen, "Robust face recognition using the deep C2D-CNN model based on decision-level fusion," *Sensors*, vol. 18, no. 7, pp. 1–27, 2018.
68. J. Holappa, T. Ahonen, and M. Pietikäinen, "An optimized illumination normalization method for face recognition," in *IEEE Second International Conference on Biometrics: Theory, Applications and Systems*, 2008, pp. 1–6.
69. P. Král, L. Lenc, and A. Vrba, "Enhanced local binary patterns for automatic face recognition," arXiv Prepr. arXiv1702.03349, 2017.
70. J. Lin and C.-T. Chiu, "Lbp edge-mapped descriptor using mgm interest points for face recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 1183–1187.
71. L. Lenc, "Genetic algorithm for weight optimization in descriptor based face recognition methods," in *Proceedings of the 8th International Conference on Agents and Artificial Intelligence*. SCITEPRESS-Science and Technology Publications, 2016, pp. 330–336.

72. J. Gaston, J. Ming, and D. Crookes, "Unconstrained face identification with multi-scale block-based correlation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 1477–1481.
73. G. B. Huang and E. Learned-miller, "Labeled faces in the wild : Updates and new reporting procedures," *Department of Computer Science, University of Massachusetts Amherst*, Amherst, MA, USA, Technical Report, pp. 1–14, 2014.
74. X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li, "High-fidelity pose and expression normalization for face recognition in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 787–796.
75. A. Ouamane, M. Bengherabi, A. Hadid, and M. Cheriet, "Side-information based exponential discriminant analysis for face verification in the wild," in *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2015, pp. 1–6.
76. Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1891–1898.
77. Y. Sun, X. Wang, and X. Tang, "Hybrid deep learning for face verification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 1997–2009, 2016.
78. G. B. Huang, H. Lee, and E. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2518–2525.
79. O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015, pp. 1–12.
80. O. Barkan, J. Weill, L. Wolf, and H. Aronowitz, "Fast high dimensional vector multiplication face recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1960–1967.
81. T. Hassner, S. Harel, E. Paz, and R. Enbar, "Effective face frontalization in unconstrained images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4295–4304.

1. A. Ullah, K. Muhammad, J. Del Ser, S. W. Baik, and V. Albuquerque, "Activity recognition using temporal optical flow convolutional features and multi-layer LSTM," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9692–9702, 2019.
2. I. U. Haq, K. Muhammad, A. Ullah, and S. W. Baik, "DeepStar: Detecting starring characters in movies," *IEEE Access*, vol. 7, pp. 9265–9272, 2019.
3. Y. Liu, L. Nie, L. Han, L. Zhang, and D. S. Rosenblum, "Action2Activity: Recognizing complex activities from sensor data," in *IJCAI*, 2015, pp. 1617–1623.
4. Y. Liu, L. Nie, L. Liu, and D. S. Rosenblum, "From action to activity: Sensor-based activity recognition," *Neurocomputing*, vol. 181, pp. 108–115, 2016.
5. A. Ullah, K. Muhammad, I. U. Haq, and S. W. Baik, "Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments," *Future Generation Computer Systems*, vol. 96, pp. 386–397, 2019.
6. M.-C. Roh, H.-K. Shin, and S.-W. Lee, "View-independent human action recognition with volume motion template on single stereo camera," *Pattern Recognition Letters*, vol. 31, pp. 639–647, 2010.
7. M. Xin, H. Zhang, H. Wang, M. Sun, and D. Yuan, "Arch: Adaptive recurrent-convolutional hybrid networks for long-term action recognition," *Neurocomputing*, vol. 178, pp. 87–102, 2016.
8. D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, pp. 249–257, 2006.

9. M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, “Action classification in soccer videos with long short-term memory recurrent neural networks,” in *International Conference on Artificial Neural Networks*, 2010, pp. 154–159.
10. A. Kovashka, and K. Grauman, “Learning a hierarchy of discriminative space-time neighborhood features for human action recognition,” in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2046–2053.
11. M. Sekma, M. Mejdoub, and C. B. Amar, “Human action recognition based on multi-layer fisher vector encoding method,” *Pattern Recognition Letters*, vol. 65, pp. 37–43, 2015.
12. J. Hou, X. Wu, Y. Sun, and Y. Jia, “Content-attention representation by factorized action-scene network for action recognition,” *IEEE Transactions on Multimedia*, vol. 20, pp. 1537–1547, 2018.
13. F. U. M. Ullah, A. Ullah, K. Muhammad, I. U. Haq, and S. W. Baik, “Violence detection using spatiotemporal features with 3D convolutional neural network,” *Sensors*, vol. 19, p. 2472, 2019.
14. K.-i. Funahashi, and Y. Nakamura, “Approximation of dynamical systems by continuous time recurrent neural networks,” *Neural Networks*, vol. 6, pp. 801–806, 1993.
15. S. Hochreiter, and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, pp. 1735–1780, 1997.
16. H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
17. M. Sajjad, S. Khan, K. Muhammad, W. Wu, A. Ullah, and S. W. Baik, “Multi-grade brain tumor classification using deep CNN with extensive data augmentation,” *Journal of Computational Science*, vol. 30, pp. 174–182, 2019.
18. J. Ahmad, K. Muhammad, S. Bakshi, and S. W. Baik, “Object-oriented convolutional features for fine-grained image retrieval in large surveillance datasets,” *Future Generation Computer Systems*, vol. 81, pp. 314–330, 2018/04/01/ 2018.
19. K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, “Efficient deep CNN- based fire detection and localization in video surveillance applications,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1419–1434, 2019.
20. K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H. H. G. Wang, and S. W. Baik, “Secure surveillance framework for IoT systems using probabilistic image encryption,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3679–3689, 2018.
21. M. Sajjad, A. Ullah, J. Ahmad, N. Abbas, S. Rho, and S. W. Baik, “Integrating salient colors with rotational invariant texture features for image representation in retrieval systems,” *Multimedia Tools and Applications*, vol. 77, pp. 4769–4789, 2018.
22. A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, “Action recognition in video sequences using deep Bi-directional LSTM with CNN features,” *IEEE Access*, vol. 6, pp. 1155–1166, 2018.
23. A. Ogawa, and T. Hori, “Error detection and accuracy estimation in automatic speech recognition using deep bidirectional recurrent neural networks,” *Speech Communication*, vol. 89, pp. 70–83, 2017.
24. J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” arXiv preprint arXiv:1412.3555, 2014.
25. W. Li, W. Nie, and Y. Su, “Human action recognition based on selected spatio-temporal features via bidirectional LSTM,” *IEEE Access*, vol. 6, pp. 44211–44220, 2018.

26. M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1971–1980.
27. Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek, "VideoLSTM convolves, attends and flows for action recognition," *Computer Vision and Image Understanding*, vol. 166, pp. 41–50, 2018.
28. C.-Y. Ma, M.-H. Chen, Z. Kira, and G. AlRegib, "TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition," *Signal Processing: Image Communication*, vol. 71, pp. 76–87, 2019.
29. Z. Chen, B. Ramachandra, T. Wu, and R. R. Vatsavai, "Relational long short-term memory for video action recognition," arXiv preprint arXiv:1811.07059, 2018.
30. W. Du, Y. Wang, and Y. Qiao, "Recurrent spatial-temporal attention network for action recognition in videos," *IEEE Transactions on Image Processing*, vol. 27, pp. 1347–1360, 2018.
31. J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, et al., "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.
32. Y. Huang, X. Cao, Q. Wang, B. Zhang, X. Zhen, and X. Li, "Long-short term features for dynamic scene classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 1038–1047, 2019.
33. S. Ma, L. Sigal, and S. Sclaroff, "Learning activity progression in lstms for activity detection and early detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1942–1950.
34. L. Sun, K. Jia, K. Chen, D.-Y. Yeung, B. E. Shi, and S. Savarese, "Lattice long short-term memory for human action recognition," in *ICCV*, 2017, pp. 2166–2175.
35. V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4041–4049.
36. H. Yang, J. Zhang, S. Li, and T. Luo, "Bi-direction hierarchical LSTM with spatial-temporal attention for action recognition," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 1, pp. 775–786, 2019.
37. K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," arXiv preprint arXiv:1212.0402, 2012.
38. H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2556–2563.
39. C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004*, 2004, pp. 32–36.
40. M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, 2009, pp. 2929–2936.
41. F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.
42. W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, et al., "The kinetics human action video dataset," arXiv preprint arXiv:1705.06950, 2017.
43. C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, et al., "AVA: A video dataset of spatio-temporally localized atomic visual actions," arXiv preprint arXiv:1705.08421, vol. 3, p. 6, 2017.
44. S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, et al., "Youtube-8m: A large-scale video classification benchmark," arXiv preprint arXiv:1609.08675, 2016.

45. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.
46. M. Monfort, B. Zhou, S. A. Bargal, A. Andonian, T. Yan, K. Ramakrishnan, et al., "Moments in time dataset: One million videos for event understanding," arXiv preprint arXiv:1801.03150, 2018.
47. S. Khan, K. Muhammad, S. Mumtaz, S. W. Baik, and V. H. C. de Albuquerque, "Energy-efficient deep CNN for smoke detection in foggy IoT environment," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9237–9245, 2019.
48. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.
49. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size," arXiv preprint arXiv:1602.07360, 2016.
50. N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," arXiv preprint arXiv:1807.11164, vol. 5, 2018.
51. M. Sajjad, M. Nasir, F. U. M. Ullah, K. Muhammad, A. K. Sangaiah, and S. W. Baik, "Raspberry Pi assisted facial expression recognition framework for smart security in law-enforcement services," *Information Sciences*, vol. 479, pp. 416–431, 2019.
52. M. Sajjad, S. Khan, T. Hussain, K. Muhammad, A. K. Sangaiah, A. Castiglione, et al., "CNN-based anti-spoofing two-tier multi-factor authentication system," *Pattern Recognition Letters*, vol 126, pp. 123–131, 2019.
53. J. Ahmad, K. Muhammad, J. Lloret, and S. W. Baik, "Efficient conversion of deep features to compact binary codes using Fourier decomposition for multimedia big data," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 3205–3215, July 2018.
54. K. Muhammad, T. Hussain, and S. W. Baik, "Efficient CNN based summarization of surveillance videos for resource-constrained devices," *Pattern Recognition Letters*, 2018.
55. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
56. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.
1. Richard S. Wallace, Anthony Stentz, Charles E. Thorpe, Hans P. Moravec, William Whittaker, and Takeo Kanade. First results in robot road-following. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 1089–1095. Citeseer, 1985.
2. Sebastian Thrun. Toward robotic cars. *Communications of the ACM*, 53(4): 99–106, 2010.
3. Matthew A. Turk, David G. Morgenthaler, Keith D. Gremban, and Martin Marra. VITS-A vision system for autonomous land vehicle navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3): 342–361, 1988.
4. Kichun Jo, Junsoo Kim, Dongchul Kim, Chulhoon Jang, and Myoungho Sunwoo. Development of autonomous car-Part II: A case study on the implementation of an autonomous driving system based on distributed architecture. *IEEE Transactions on Industrial Electronics*, 62(8): 5119–5132, 2015.
5. Ernst Dieter Dickmanns. *Dynamic Vision for Perception and Control of Motion*. Springer-Verlag, London, 2007.
6. Mahdi Rezaei and Reinhard Klette. *Computer Vision for Driver Assistance*. Springer, Cham Switzerland, 2017.

7. Mahmoud Hassaballah and Khalid M. Hosny. *Recent Advances in Computer Vision: Theories and Applications*, volume 804. Springer International Publishing, 2019.
8. David Michael Stavens. *Learning to Drive: Perception for Autonomous Cars*. Stanford University, 2011.
9. Yuna Ro and Youngwook Ha. A factor analysis of consumer expectations for autonomous cars. *Journal of Computer Information Systems*, 59(1): 52–60, 2019.
10. Adil Hashim, Tanya Saini, Hemant Bhardwaj, Adityan Jothi, and Ammannagari Vinay Kumar. Application of swarm intelligence in autonomous cars for obstacle avoidance. In *Integrated Intelligent Computing, Communication and Security*, pages 393–404. Springer, 2019.
11. Angelos Amanatiadis, Evangelos Karakasis, Loukas Bampis, Stylianos Ploumpis, and Antonios Gasteratos. ViPED: On-road vehicle passenger detection for autonomous vehicles. *Robotics and Autonomous Systems*, 112: 282–290, 2019.
12. Yusuf Artan, Orhan Bulan, Robert P. Loce, and Peter Paul. Passenger compartment violation detection in HOV/HOT lanes. *IEEE Transactions on Intelligent Transportation Systems*, 17(2): 395–405, 2016.
13. Dorsa Sadigh, S Shankar Sastry, and Sanjit A. Seshia. Verifying robustness of human-aware autonomous cars. *IFAC-PapersOnLine*, 51(34): 131–138, 2019.
14. Bhargava Reddy, Ye-Hoon Kim, Sojung Yun, Chanwon Seo, and Junik Jang. Real-time driver drowsiness detection for embedded system using model compression of deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 121–128, 2017.
15. Theocharis Kyriacou, Guido Bugmann, and Stanislao Lauria. Vision-based urban navigation procedures for verbally instructed robots. *Robotics and Autonomous Systems*, 51(1): 69–80, 2005.
16. SAE International Committee. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. Technical Report J3016–201401, SAE International, 2014. http://doi.org/ 10.4271/J3016_201806.
17. Jonathan Horgan, Ciaran Hughes, John McDonald, and Senthil Yogamani. Vision-based driver assistance systems: Survey, taxonomy and advances. In *International Conference on Intelligent Transportation Systems (ITSC)*, pages 2032–2039. IEEE, 2015.
18. Zhentao Hu, Tianxiang Chen, Quanbo Ge, and Hebin Wang. Observable degree analysis for multi-sensor fusion system. *Sensors*, 18(12): 4197, 2018.
19. Shan Luo, Joao Bimbo, Ravinder Dahiya, and Hongbin Liu. Robotic tactile perception of object properties: A review. *Mechatronics*, 48: 5467, 2017.
20. Guilherme N. DeSouza and Avinash C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2): 237–267, 2002.
21. Oliver Pink, Jan Becker, and Soren Kammel. Automated driving on public roads: Experiences in real traffic. *IT-Information Technology*, 57(4): 223–230, 2015.
22. Thomas Braunl. *Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems*. Springer Science & Business Media, 2008.
23. Gregory Dudek and Michael Jenkin. *Computational Principles of Mobile Robotics*. Cambridge University Press, 2010.
24. Uwe Handmann, Thomas Kalinke, Christos Tzomakas, Martin Werner, and Werner von Seelen. Computer vision for driver assistance systems. In *Enhanced and Synthetic Vision 1998*, volume 3364, pages 136–148. International Society for Optics and Photonics, 1998.
25. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436, 2015.

26. Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *International Conference on Machine Learning (ICML-11)*, pages 689–696, 2011.
27. Li Deng and Dong Yu. Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4): 197–387, 2014.
28. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.
29. Josh Patterson and Adam Gibson. *Deep Learning: A Practitioner's Approach*. O'Reilly Media, Inc., 2017.
30. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
31. Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.
32. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.
33. Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Daniel Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan, Nigel Duffy, et al. Evolving deep neural networks. In *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pages 293–312. Elsevier, 2019.
34. Michael Hauser, Sean Gunn, Samer Saab Jr, and Asok Ray. State-space representations of deep neural networks. *Neural Computation*, 31(3): 538–554, 2019.
35. Abhinav Valada, Gabriel L. Oliveira, Thomas Brox, and Wolfram Burgard. Deep multispectral semantic scene understanding of forested environments using multimodal fusion. In *International Symposium on Experimental Robotics*, pages 465–477. Springer, 2016.
36. Taigo M. Bonanni, Andrea Pennisi, D.D Bloisi, Luca Iocchi, and Daniele Nardi. Human-robot collaboration for semantic labeling of the environment. In *3rd Workshop on Semantic Perception, Mapping and Exploration*, pp. 1–6, 2013.
37. Abhijit Kundu, Yin Li, Frank Dellaert, Fuxin Li, and James M. Rehg. Joint semantic segmentation and 3D reconstruction from monocular video. In *European Conference on Computer Vision*, pages 703–718. Springer, 2014.
38. Ondrej Miksik, Vibhav Vineet, Morten Lidegaard, Ram Prasaath, Matthias Nießner, Stuart Golodetz, Stephen L. Hicks, Patrick Perez, Shahram Izadi, and Philip H.S. Torr. The semantic paintbrush: Interactive 3d mapping and recognition in large outdoor spaces. In *33rd Annual ACM Conference on Human Factors in Computing Systems,* pages 3317–3326. ACM, 2015.
39. Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
40. Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2): 88–97, 2009.
41. Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8): 1915–1929, 2013.
42. Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

43. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E . Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, 25, pages 1097–1105. Curran Associates, Inc., 2012.
44. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
45. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
46. Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.
47. Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12): 2481–2495, 2017.
48. Jun Mao, Xiaoping Hu, Xiaofeng He, Lilian Zhang, Liao Wu, and Michael J. Milford. Learning to fuse multiscale features for visual place recognition. *IEEE Access*, 7: 5723–5735, 2019.
49. Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122, 2015.
50. Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *EEE Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2016.
51. Boyu Chen, Peixia Li, Chong Sun, Dong Wang, Gang Yang, and Huchuan Lu. Multi attention module for visual tracking. *Pattern Recognition*, 87: 80–93, 2019.
52. Liuyuan Deng, Ming Yang, Hao Li, Tianyi Li, Bing Hu, and Chunxiang Wang. Restricted deformable convolution based road scene semantic segmentation using surround view cameras. arXiv preprint arXiv:1801.00708, 2018.
53. Varun Ravi Kumar, Stefan Milz, Christian Witt, Martin Simon, Karl Amende, Johannes Petzold, Senthil Yogamani, and Timo Pech. Monocular fisheye camera depth estimation using sparse lidar supervision. In *International Conference on Intelligent Transportation Systems (ITSC)*, pages 2853–2858. IEEE, 2018.
54. Marius Cordts, Timo Rehfeld, Lukas Schneider, David Pfeiffer, Markus Enzweiler, Stefan Roth, Marc Pollefeys, and Uwe Franke. The stixel world: A medium-level representation of traffic scenes. *Image and Vision Computing*, 68: 40–52, 2017.
55. Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision*, pages 44–57. Springer, 2008.
56. Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.
57. Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. arXiv preprint arXiv:1604.01685, 2016.
58. Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 807–814. IEEE, 2005.
59. Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *IEEE International Conference on Computer Vision,* pages 4990–4999, 2017.

60. Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 954–960, 2018.
61. Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2016.
62. German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016.
63. NuTonomy. Nuscenes dataset. 2012. https://www.nuscenes.org/.
64. Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. arXiv preprint arXiv:1810.08705, 2018.
65. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4): 834–848, 2018.
66. Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10): 2024–2039, 2016.
67. Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11): 3174–3182, 2018.
68. Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.
69. Clement Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE Computer Vision and Pattern Recognition*, volume 2, page 7, 2017.
70. Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian Conference on Computer Vision*, pages 213–228. Springer, 2016.
71. Lingni Ma, Jorg Stückler, Christian Kerl, and Daniel Cremers. Multi-view deep learning for consistent semantic mapping with RGB-d cameras. arXiv preprint arXiv:1703.08866, 2017.
72. Di Lin, Guangyong Chen, Daniel Cohen-Or, Pheng-Ann Heng, and Hui Huang. Cascaded feature network for semantic segmentation of rgb-d images. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1320–1328. IEEE, 2017.
73. Yuanzhouhan Cao, Chunhua Shen, and Heng Tao Shen. Exploiting depth from single monocular images for object detection and semantic segmentation. *IEEE Transactions on Image Processing*, 26(2): 836–846, 2017.
74. Min Bai, Wenjie Luo, Kaustav Kundu, and Raquel Urtasun. Exploiting semantic information and deep matching for optical flow. In *European Conference on Computer Vision*, pages 154–170. Springer, 2016.
75. Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Imaging Understanding Workshop*, pages 121–130. Vancouver, British Columbia, 1981.
76. Gunnar Farneback. Two-frame motion estimation based on polynomial expansion. In *Scandinavian Conference on Image Analysis*, pages 363–370. Springer, 2003.

77. Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.
78. Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017.
79. Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1495–1501, 2017.
80. Junhwa Hur and Stefan Roth. Joint optical flow and temporally consistent semantic segmentation. In *European Conference on Computer Vision*, pages 163–177. Springer, 2016.
81. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
82. Mennatullah Siam, Heba Mahgoub, Mohamed Zahran, Senthil Yogamani, Martin Jagersand, and Ahmad El-Sallab. Modnet: Moving object detection network with motion and appearance for autonomous driving. arXiv preprint arXiv:1709.04821, 2017.
83. Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. arXiv preprint arXiv:1701.05384, 2017.
84. Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.
85. Mennatullah Siam and Mohammed Elhelw. Enhanced target tracking in uav imagery with pn learning and structural constraints. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 586–593, 2013.
86. Ahmed Salaheldin, Sara Maher, and Mohamed Helw. Robust real-time tracking with diverse ensembles and random projections. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 112–120, 2013.

1. K. D. Atherton, "Senate hearing: Drones are "basically flying smart-phones"." https://www.popsci.com/technology/article/2013–03/ how-drone-smartphone, March 2013.
2. B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS-Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and More*. Springer Science & Business Media, 2007.
3. D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, et al., "Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in gps-denied environments," *IEEE Robotics & Automation Magazine*, vol. 21, no. 3, pp. 26–40, 2014.
4. G. Chowdhary, E. N. Johnson, D. Magree, A. Wu, and A. Shein, "Gps-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft," *Journal of Field Robotics*, vol. 30, no. 3, pp. 415–438, 2013.
5. S. Rady, A. Kandil, and E. Badreddin, "A hybrid localization approach for uav in GPS denied areas," in *IEEE/SICE International Symposium on System Integration*, pp. 1269–1274, IEEE, 2011.
6. A. Masselli, R. Hanten, and A. Zell, "Localization of unmanned aerial vehicles using terrain classification from aerial images," in *Intelligent Autonomous Systems*, 13, pp. 831–842, Springer, 2016.

7. M. Mantelli, D. Pittol, R. Neuland, A. Ribacki, R. Maffei, V. Jorge, E. Prestes, and M. Kolberg, "A novel measurement model based on abbrief for global localization of a UAV over satellite images," *Robotics and Autonomous Systems*, vol. 112, pp. 304–319, 2019.
8. V. Sowmya, K. Soman, and M. Hassaballah, "Hyperspectral image: Fundamentals and advances," in *Recent Advances in Computer Vision*, pp. 401–424, Springer, 2019.
9. B. Fan, Y. Du, L. Zhu, and Y. Tang, "The registration of UAV down-looking aerial images to satellite images with image entropy and edges," in *Intelligent Robotics and Applications*, pp. 609–617, 2010.
10. D.-G. Sim, R.-H. Park, R.-C. Kim, S. U. Lee, and I.-C. Kim, "Integrated position estimation using aerial image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 1–18, 2002.
11. M. Hassaballah and A. I. Awad, "Detection and description of image features: An introduction," in *Image Feature Detectors and Descriptors*, pp. 1–8, Springer, 2016.
12. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91–110, 2004.
13. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
14. T. Koch, P. d'Angelo, F. Kurz, F. Fraundorfer, P. Reinartz, and M. Korner, "The tum-dlr multimodal earth observation evaluation benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.
15. P. Shukla, S. Goel, P. Singh, and B. Lohani, "Automatic geolocation of targets tracked by aerial imaging platforms using satellite imagery," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 1, p. 381, 2014.
16. G. Conte and P. Doherty, "Vision-based unmanned aerial vehicle navigation using geo-referenced information," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, p. 10, 2009.
17. A. Yol, B. Delabarre, A. Dame, J.-E. Dartois, and E. Marchand, "Vision-based absolute localization for unmanned aerial vehicles," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3429–3434, IEEE, 2014.
18. N. Audebert, B. Le Saux, and S. Lefvre, "Semantic segmentation of earth observation data using multimodal and multi-scale deep networks," in *Asian Computer Vision Conference*, pp. 180–196, Springer, Cham, November 2016.
19. D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla, "Semantic segmentation of aerial images with an ensemble of CNNs," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3, p. 473, 2016.
20. F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf, "The ISPRS benchmark on urban object classification and 3d building reconstruction," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1, no. 3, pp. 293–298, 2012.
21. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602, 2013.
22. S. Lefvre, D. Tuia, J. D. Wegner, T. Produit, and A. S. Nassar, "Toward seamless multiview scene analysis from satellite to street level," *Proceedings of the IEEE*, vol. 105, pp. 1884–1899, October 2017.
23. S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 539–546, IEEE, 2005.
24. J. Bromley, I. Guyon, Y. LeCun, E. Sackinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Advances in Neural Information Processing Systems*, pp. 737–744, 1994.

25. A. Nassar, K. Amer, R. ElHakim, and M. ElHelw, "A deep CNN-based framework for enhanced aerial imagery registration with applications to uav geolocalization," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1513–1523, 2018.
26. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," arXiv preprint arXm:1505.0Ą597, 2015.
27. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXm:1Ą09.1556, 2014.
28. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
29. D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
30. T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, no. 2, pp. 26–31, 2012.
31. T. Dozat, "Incorporating nesterov momentum into adam," in *International Conference on Learning Representations*, 2016.
32. T. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons," *Biol. Skr.*, vol. 5, pp. 1–34, 1948.
33. P. Jaccard, "Le coefficient generique et le coefficient de communaute dans la flore marocaine." Mémoires de la Société Vaudoise des Sciences Naturelles, 2: 385–403, 1926.
34. D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.
35. M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.
36. R. Hundt, "Loop recognition in C++/Java/Go/Scala," *Proceedings of Scala Days*, p. 38, 2011.
37. A. FPOHOB, "Above varosha: One of the most famous ghost cities." https://youtu.be/AYKC0dsrh4U, February 2016.
38. ZFTurbo, "ZF UNET 224 pretrained model." https://github.com/ZFTurbo/ZF_UNET_224_Pretrained_Model, 2017.
39. Y. Liu, B. Fan, and C. Pan, "Method description for potsdam: 2D labelling challenge." http://www2.isprs.org/commissions/comm3/wg4/ 2d-sem-label-potsdam.html.
40. P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, "Learning aerial image segmentation from online maps," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 11, pp. 6054–6068, 2017.
1. Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187, 27–48, 2016.
2. Soren Goyal and PaulBenjamin. Object recognition using deep neural networks: A survey. http://arxiv.org/abs/1412.3684, 2014.
3. Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354–377, 2018.
4. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521, 436–444, 28 May 2015, doi:10.1038/nature14539,
5. Li Deng. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing*, 3, 1–29, 2014.

6. Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117, 2015.
7. Suraj Srinivas, Ravi Kiran Sarvadevabhatla, Konda Reddy Mopuri, Nikita Prabhu, Srinivas S. S. Kruthiventi, and R. Venkatesh Babu. A taxonomy of deep convolutional neural nets for computer vision. *Frontiers in Robotics and AI*, 2, 2016.
8. Xiaowei Zhou, Emanuele Rodola, Jonathan Masci, Pierre Vandergheynst, Sanja Fidler, and Kostas Daniilidis. Workshop geometry meets deep learning, ECCV 2016, 2016.
9. Yi Li, Yezhou Yang, Michael James, Danil Prokhorov. Deep Learning for Autonomous Robots, Workshop at RSS 2016, 2016.
10. Awad, Ali Ismail, and Mahmoud Hassaballah. Image feature detectors and descriptors. *Studies in Computational Intelligence*. Springer International Publishing, Cham, 2016.
11. Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, et al. Tensorflow: A system for large-scale machine learning. 12th Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283, 2016.
12. Patricio Loncomilla, Javier Ruiz-del-Solar, and Luz Martínez, Object recognition using local invariant features for robotic applications: A survey, *Pattern Recognition*, 60, 499–514, 2016.
13. Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 541–551, 1989.
14. R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 580–587, 2014.
15. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 6, 1137–1149, 2017.
16. Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517–6525, 2017.
17. Max Schwarz, Hannes Schulz, and Sven Behnke. RGB-D object recognition and pose estimation based on Pre-trained convolutional neural network features. *IEEE International Conference on Robotics and Automation (ICRA)*, 1329–1335, 2005.
18. Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from RGB-D images for object detection and segmentation. *European Conference on Computer Vision (ECCV)*, 345–360, 2014.
19. Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust RGB-D object recognition. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 681–687, 2015.
20. Joel Schlosser, Christopher K. Chow, and Zsolt Kira. Fusing LIDAR and images for pedestrian detection using convolutional neural networks. *IEEE International Conference on Robotics and Automation (ICRA)*, 2198–2205, 2016.
21. G. Pasquale, C. Ciliberto, F. Odone, L. Rosasco, and L. Natale. Teaching iCub to recognize objects using deep convolutional neural networks. *4th International Conference on Machine Learning for Interactive Systems (MLIS'15)*, 43, 21–25, 2015.
22. Dario Albani, Ali Youssef, Vincenzo Suriani, Daniele Nardi, and Domenico Daniele Bloisi. A deep learning approach for object recognition with NAO soccer robots. *RoboCup International Symposium*, 392–403, 2016.
23. Denis Tomè, Federico Monti, Luca Baroffio, Luca Bondi, Marco Tagliasacchi, and Stefano Tubaro. Deep convolutional neural networks for pedestrian detection. *Signal Processing: Image Communication*, 47, C, 482–489, 2016.

24. Hasan F. M. Zaki, Faisal Shafait, and Ajmal Mian. Convolutional hypercube pyramid for accurate RGB-D object category and instance recognition. *IEEE International Conference on Robotics and Automation (ICRA), 2016*, 1685–1692, 2016.
25. Judy Hoffman, Saurabh Gupta, Jian Leong, Sergio Guadarrama, and Trevor Darrell. Cross-modal adaptation for RGB-D detection. *IEEE International Conference on Robotics and Automation (ICRA)*, 5032–5039, 2016.
26. Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3D lidar using fully convolutional network. *Proceedings of Robotics: Science and System Proceedings*, 2016.
27. Nvidia: GPU-Based deep learning inference: A performance and power analysis. Whitepaper, 2015.
28. Jan Hosang, Mohamed Omran, Rodrigo Benenson, and Bernt Schiele. Taking a deeper look at pedestrians. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4073–4082, 2015.
29. Rudy Bunel, Franck Davoine, and Philippe Xu. Detection of pedestrians at far distance. *IEEE International Conference on Robotics and Automation (ICRA)*, 2326–2331, 2016.
30. Jimmy Ren, Xiaohao Chen, Jianbo Liu, Wenxiu Sun, Jiahao Pang, Qiong Yan, Yu-Wing Tai, and Li Xu. Accurate single stage detector using recurrent rolling convolution. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 752–760, 2017.
31. Zhaowei Cai, Quanfu Fan, Rogerio S. Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. *European Conference on Computer Vision (ECCV)*, 354–370, 2016.
32. Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Subcategory-aware convolutional neural networks for object proposals and detection. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 924–933, 2017.
33. Yousong Zhu, Jinqiao Wang, Chaoyang Zhao, Haiyun Guo, and Hanqing Lu. Scale-adaptive deconvolutional regression network for pedestrian detection. *Asian Conference on Computer Vision*, 416–430, 2016.
34. Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2129–2137, 2016.
35. Piotr Dollar, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, 8, 1532–1545, 2014.
36. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097–1105, 2012.
37. Gabriel L. Oliveira, Abhinav Valada, Claas Bollen, Wolfram Burgard, and Thomas Brox. Deep learning for human part discovery in images. *IEEE International Conference on Robotics and Automation (ICRA)*, 1634–1641, 2016.
38. Nicolás Cruz, Kenzo Lobos-Tsunekawa, and Javier Ruiz-del-Solar. Using convolutional neural networks in robots with limited computational resources: Detecting NAO robots while playing soccer. RoboCup 2017: Robot World Cup XXI, 19–30, 2017.
39. Daniel Speck, Pablo Barros, Cornelius Weber, and Stefan Wermter. Ball localization for robocup soccer using convolutional neural networks. *RoboCup International Symposium*, 19–30, 2016.
40. Francisco Leiva, Nicolas Cruz, Ignacio Bugueño, and Javier Ruiz-del-Solar. Playing soccer without colors in the SPL: A convolutional neural network approach. *RoboCup Symposium*, 2018 (in press).

41. Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34, 4–5, 705–724, 2015.
42. Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. *IEEE International Conference on Robotics and Automation (ICRA)*, 1316–1322, 2015.
43. Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours. *IEEE International Conference on Robotics and Automation (ICRA)*, 3406–3413, 2016.
44. Di Guo, Tao Kong, Fuchun Sun, and Huaping Liu. Object discovery and grasp detection with a shared convolutional neural network. *IEEE International Conference on Robotics and Automation (ICRA)*, 2038–2043, 2016.
45. Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37, 4–5, 421–436, 2018.
46. Jaeyong Sung, Seok Hyun Jin, and Ashutosh Saxena. Robobarista: Object part based transfer of manipulation trajectories from crowd-sourcing in 3D pointclouds. *Robotics Research*, 3, 701–720, 2018.
47. Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. *IEEE International Conference on Robotics and Automation (ICRA)*, 512–519, 2016.
48. Aseem Saxena, Harit Pandya, Gourav Kumar, Ayush Gaud, and K. Madhava Krishna. Exploring convolutional networks for end-to-end visual servoing. *IEEE International Conference on Robotics and Automation (ICRA)*, 3817–3823, 2017.
49. Yang Gao, Lisa Anne Hendricks, Katherine J. Kuchenbecker, and Trevor Darrell. Deep learning for tactile understanding from visual and haptic data. *IEEE International Conference on Robotics and Automation (ICRA)*, 536–543, 2016.
50. Manuel Lopez-Antequera, Ruben Gomez-Ojeda, Nicolai Petkov, and Javier Gonzalez-Jimenez. Appearance-invariant place recognition by discriminatively training a convolutional neural network. *Pattern Recognition Letters*, 92, 89–95, 2017.
51. Yi Hou, Hong Zhang, and Shilin Zhou. Convolutional neural network-based image representation for visual loop closure detection. *IEEE International Conference on Information and Automation (ICRA)*, 2238–2245, 2015.
52. Niko Sunderhauf, Feras Dayoub, Sean McMahon, Ben Talbot, Ruth Schulz, Peter Corke, Gordon Wyeth, Ben Upcroft, and Michael Milford. Place categorization and semantic mapping on a mobile robot. *IEEE International Conference on Robotics and Automation (ICRA)*, 5729–5736, 2006.
53. Niko Sunderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Robotics: Science and System Proceedings*, 2015.
54. Yiyi Liao, Sarath Kodagoda, Yue Wang, Lei Shi, and Yong Liu. Understand Scene Categories by objects: A semantic regularized scene classifier using convolutional neural networks. *IEEE International Conference on Robotics and Automation (ICRA)*, 2318–2325, 2016.
55. Peter Uršic, Rok Mandeljc, Aleš Leonardis, and Matej Kristan. Part-based room categorization for household service robots. *IEEE International Conference on Robotics and Automation (ICRA)*, 2287–2294, 2016.
56. Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. *IEEE Conference on Computer Vision and Pattern recognition (CVPR)*, 3485–3492, 2010.

57. Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40, 6, 1452–1464, 2018.
58. Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *Advances in Neural Information Processing Systems* 27 (NIPS), 487–495, 2014.
59. ILSVRC 2016 scene classification, 2016. http://image-net.org/challenges/LSVRC/2016/results.
60. Places Challenge 2016. http://places2.csail.mit.edu/results2016.html.
61. Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. *IEEE Conference on Computer Vision and Pattern recognition (CVPR)*, 3485–3492, 2010.
62. Caio Cesar Teodoro Mendes, Vincent Fremont, and Denis Fernando Wolf. Exploiting fully convolutional neural networks for fast road detection. *IEEE International Conference on Robotics and Automation (ICRA)*, 3174–3179, 2016.
63. Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 567–576, 2015.
64. Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. Technical Report. arXiv preprint arXiv:1706.05587, 2017.
65. Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6230–6239, 2017.
66. Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90, 119–133, 2019.
67. Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1451–1460, 2018.
68. Rui Zhang, Sheng Tang, Yongdong Zhang, Jintao Li, and Shuicheng Yan. Scale-adaptive convolutions for scene parsing. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2031–2039, 2017.
69. Ping Luo, Guangrun Wang, Liang Lin, and Xiaogang Wang. Deep dual learning for semantic image segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2718–2726, 2017.
70. Jun Fu, Jing Liu, Yuhang Wang, and Hanqing Lu. Stacked deconvolutional network for semantic segmentation. *IEEE Transactions on Image Processing*, 99, 2017.
71. Cesar Cadena, Anthony Dick, and Ian D. Reid. Multi-modal auto-encoders as joint estimators for robotics scene understanding. *Proceedings of Robotics: Science and System Proceedings*, 2016.
72. Farzad Husain, Hannes Schulz, Babette Dellen, Carme Torras, and Sven Behnke. Combining semantic and geometric features for object class segmentation of indoor scenes. *IEEE Robotics and Automation Letters*, 2 (1), 49–55, 2016.
73. Guangrun Wang, Ping Luo, Liang Lin, and Xiaogang Wang. Learning object interactions and descriptions for semantic image segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5859–5867, 2017.
74. Alessandro Giusti, Jérôme Guzzi, Dan C. Ciresan, Fang-Lin He, Juan P. Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen Schmidhuber, Gianni Di Caro, Davide Scaramuzza, and Luca M. Gambardella. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1, 2, 661–667, 2016.

75. Dan C. Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3642–3649, 2012.
76. Lei Tai and Ming Liu. A robot exploration strategy based on q-learning network. *IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 57–62, 2016.
77. Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. *IEEE International Conference on Robotics and Automation (ICRA)*, 3357–3364, 2017.
78. Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J. Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, Dharshan Kumaran, and Raia Hadsell. Learning to navigate in complex environments. *International Conference on Learning Representations (ICLR)*, 2017.
79. Kenzo Lobos-Tsunekawa, Francisco Leiva, and Javier Ruiz-del-Solar. Visual navigation for biped humanoid robots using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 3, 4, 3247–3254, 2018.
80. Gabriele Costante, Michele Mancini, Paolo Valigi, and Thomas A. Ciarfuglia. Exploring representation learning with CNNs for frame to frame ego-motion estimation. *IEEE Robotics and Automation Letters*, 1, 1, 18–25, 2016.
81. Shichao Yang, Daniel Maturana, and Sebastian Scherer. Real-time 3D scene layout from a single image using convolutional neural networks. *IEEE International Conference on Robotics and Automation (ICRA)*, 2183–2189, 2016.
82. Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. *IEEE International Conference on Robotics and Automation (ICRA)*, 4762–4769, 2016.
83. Chengxi Ye, Yezhou Yang, Cornelia Fermuller, and Yiannis Aloimonos. What can i do around here? Deep functional scene understanding for cognitive robots. *IEEE International Conference on Robotics and Automation (ICRA)*, 4604–4611, 2017.
84. Ivan Bogun, Anelia Angelova, and Navdeep Jaitly. Object recognition from short videos for robotic perception. arXiv:1509.01602v1 [cs.CV], 4 September 2015.
85. Jimmy Ren, Xiaohao Chen, Jianbo Liu, Wenxiu Sun, Jiahao Pang, Qiong Yan, Yu-Wing Tai, and Li Xu. Accurate single stage detector using recurrent rolling convolution. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 752–760, 2017.
86. Guy Lev, Gil Sadeh, Benjamin Klein, and Lior Wolf. RNN fisher vectors for action recognition and image annotation. *European Conference on Computer Vision (ECCV)*, 833–850, 2016.
87. Farzad Husain, Babette Dellen, and Carme Torra. Action recognition based on efficient deep feature learning in the spatio-temporal domain. *IEEE Robotics and Automation Letters*, 1, 2, 984–991, 2016.
88. Xiaochuan Yin and Qijun Chen. Deep metric learning autoencoder for nonlinear temporal alignment of human motion. *IEEE International Conference on Robotics and Automation (ICRA)*, 2160–2166, 2016.
89. Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, and Ashutosh Saxena. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. *IEEE International Conference on Robotics and Automation (ICRA)*, 3118–3125, 2016.
90. Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. *IEEE International Conference on Computer Vision (ICCV)*, 4489–4497, 2015.

91. Huiwen Guo, Xinyu Wu, and Wei Fengab. Multi-stream deep networks for human action classification with sequential tensor decomposition. *Signal Processing*, 140, 198–206, 2017.
92. Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4694–4702, 2015.
93. Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, and Xiangyang Xue. Multi-stream multi-class fusion of deep networks for video classification. *ACM Multimedia*, 791–800, 2016.
94. Adithyavairavan Murali, Animesh Garg, Sanjay Krishnan, Florian T. Pokorny, Pieter Abbeel, Trevor Darrell, and Ken Goldberg. TSC-DL: Unsupervised trajectory segmentation of multi-modal surgical demonstrations with deep learning. *IEEE International Conference on Robotics and Automation (ICRA)*, 4150–4157, 2016.
95. Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations (ICLR'16)*, 2016.
96. Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A. Horowitz, and William J. Dally. EIE: Efficient inference engine on compressed deep neural network. *International Conference on Computer Architecture (ISCA)*, 243–254, 2016.
97. Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *Advances in Neural Information Processing Systems*, 28, 2017–2025, 2015.
98. Ankur Handa, Michael Bloesch, Viorica Patraucean, Simon Stent, John McCormac, and Andrew Davison. gvnn: Neural network library for geometric computer vision. *Computer Vision – ECCV 2016 Workshops*, 67–82, 2016.
99. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518, 529–533, 2015.
100. David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529, 7587, 484–489, 2016.
101. Rodrigo Verschae and Javier Ruiz-del-Solar. Object detection: Current and future directions. *Frontiers in Robotics and AI*, 29, 2, 2015.
102. Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. *International Conference on Unsupervised and Transfer Learning Workshop*, 27, 17–37, 2012.
103. Marc-Andr Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning. *Pattern Recognition*, 77, 329–353, 2018.
104. Jia Deng, Alexander C. Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? *European Conference on Computer Vision: Part V (ECCV'10)*, 71–84, 2010.
105. Stefan Leutenegger, Thomas Whelan, Richard A. Newcombe, and Andrew J. Davison. Workshop the future of real-time SLAM: Sensors, processors, representations, and algorithms, *ICCV*, 2015.
106. Ken Goldberg. Deep grasping: Can large datasets and reinforcement learning bridge the dexterity gap? Keynote Talk at ICRA, 2016.

1. Marra, F., Poggi, G., Sansone, C., Verdoliva, L.: A deep learning approach for iris sensor model identification. *Pattern Recognition Letters* 113, 4653 (2018), integrating Biometrics and Forensics.
2. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief Nets. *Neural Computation* 18(7), 1527–1554 (2006), https://doi.org/10.1162/neco.2006.18.7.1527, pMID: 16764513.
3. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11, 3371–3408 (Dec 2010).
4. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. *Neural Computation* 14(8), 1771–1800 (2002), https://doi.org/10.1162/089976602760128018.
5. Mosavi, A., Varkonyi-Koczy, A.R.: Integration of machine learning and optimization for robot learning. In: Jabłoński, R., Szewczyk, R. (eds.) *Recent Global Research and Education: Technological Challenges*, pp. 349–355. Springer International Publishing, Cham (2017).
6. Bengio, Y.: Learning deep architectures for AI. *Foundations and Trends in Machine Learning* 2(1), 1–127 (2009), http://dx.doi.org/10.1561/2200000006.
7. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)* 115(3), 211–252 (2015), https://doi.org/10.1007/s11263–015–0816-y.
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems* 25, (NIPS 2012), pp. 1097–1105. Curran Associates, Inc. (2012).
9. Hassaballah, M., Awad, A.I.: Detection and description of image features: An introduction. In: Awad, A.I., Hassaballah, M. (eds.) *Image Feature Detectors and Descriptors: Foundations and Applications, Studies in Computational Intelligence*, Vol. 630, pp. 1–8. Springer International Publishing, Cham (2016).
10. Lowe, D.G.: Distinctive image features from scale-invariant key-points. *International Journal of Computer Vision* 60(2), 91–110 (2004), https://doi.Org/10.1023/B:VISI.0000029664.99615.94.
11. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).* Vol. 1, pp. 886–893. IEEE (2005), https://doi.org/10.1109/CVPR.2005.177.
12. Yang, J., Jiang, Y.G., Hauptmann, A.G., Ngo, C.W.: Evaluating bag- of-visual-words representations in scene classification. In: *Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval*. pp. 197–206. ACM, New York, NY, USA (2007), https://doi.org/10.1145/1290082.1290111.
13. Awad, A.I., Hassaballah, M.: *Image Feature Detectors and Descriptors: Foundations and Applications, Studies in Computational Intelligence*, Vol. 630. Springer International Publishing, Cham, 1st edn. (2016).
14. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11), 2278–2324 (1998), https://doi.org/10.1109/5–726791.
15. Bishop, C.: *Pattern Recognition and Machine Learning.* Springer-Verlag New York, 1 edn. (2006).
16. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology* 160(1), 106–154 (1962).

17. Abd-Ellah, M.K., Awad, A.I., Khalaf, A.A.M., Hamed, H.F.A.: Two-phase multi-model automatic brain tumour diagnosis system from magnetic resonance images using convolutional neural networks. *EURASIP Journal on Image and Video Processing* 97(1), 1–10 (2018).
18. Srinivas, S., Sarvadevabhatla, R.K., Mopuri, K.R., Prabhu, N., Kruthiventi, S.S., Babu, R.V.: Chapter 2—An introduction to deep convolutional neural nets for computer vision. In: Zhou, S.K., Greenspan, H., Shen, D. (eds.) *Deep Learning for Medical Image Analysis*, pp. 25–52. Academic Press (2017).
19. Zeiler, M., Fergus, R.: Stochastic pooling for regularization of deep convolutional neural networks. In: *Proceedings of the International Conference on Learning Representation (ICLR)*. pp. 1–9 (2013).
20. Srivastava, R.K., Masci, J., Kazerounian, S., Gomez, F., Schmidhuber, J.: Compete to compute. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems* 26, (NIPS 2013), pp. 2310–2318. Curran Associates, Inc. (2013).
21. Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: *2017 International Conference on Engineering and Technology (ICET)*. pp. 1–6 (Aug 2017), https://doi.org/10.1109/ICEngTechnol.2017.8308186.
22. Maas, A.L.: Rectifier nonlinearities improve neural network acoustic models. In: Dasgupta, S., McAllester, D. (eds.) *Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research*, Vol. 28. PMLR, Atlanta, Georgia, USA (17–19 June 2013).
23. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. pp. 1026–1034. IEEE, Santiago, Chile (2015), https://doi.org/10.1109/ICCV.2015.123.
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 770–778. IEEE, Las Vegas, NV, USA (June 2016), https://doi.org/10.1109/CVPR.2016.90.
25. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Nature* 323, 333–336 (1986), https://doi.org/10.1038/323533a0.
26. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Lechevallier, Y., Saporta, G. (eds.) *Proceedings of COMP-STAT'2010*. pp. 177–186. Physica-Verlag HD, Heidelberg, Paris, France (2010), https://doi.org/10.1007/978–3-7908–2604–3_16.
27. B.T.Polyak: Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics* 4(5), 1–17 (1964), https://doi.org/10.1016/0041–5553(64)90137–5.
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *CoRR* abs/1412.6980 (2014), http://arxiv.org/abs/1412.6980.
29. Nesterov, Y.: A method of solving a convex programming problem with convergence rate O(1/sqr(k)). *Soviet Mathematics Doklady* 27(2), 372–376 (1983).
30. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, 2121–2159 (2011).
31. Zeiler, M.D.: ADADELTA: An adaptive learning rate method. *CoRR* abs/1212.5701 (2012), http://arxiv.org/abs/1212.5701.
32. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: Dasgupta, S., McAllester, D. (eds.) *Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research*, Vol. 28, pp. 1139–1147. PMLR, Atlanta, Georgia, USA (17–19 June 2013).

33. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* abs/1207.0580 (2012), http://arxiv.org/abs/1207.0580.
34. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using DropConnect. In: Dasgupta, S., McAllester, D. (eds.) *Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research*, Vol. 28, pp. 1058–1066. PMLR, Atlanta, Georgia, USA (17–19 June 2013).
35. Wang, S., Manning, C.: Fast dropout training. In: Dasgupta, S., McAllester, D. (eds.) *Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research*, Vol. 28, pp. 118–126. PMLR, Atlanta, Georgia, USA (17–19 June 2013).
36. Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. In: Dasgupta, S., McAllester, D. (eds.) *Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research*, Vol. 28, pp. 1319–1327. PMLR, Atlanta, Georgia, USA (17–19 June 2013).
37. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, Vol. 37. pp. 448–456. ICML'15, JMLR.org (2015).
38. Hornik, K.: Approximation capabilities of multilayer feedforward networks. *Neural Networks* 4(2), 251–257 (1991), https://doi.org/10.1016/0893–6080(91)90009-T.
39. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556 (2014), http://arxiv.org/abs/1409.1556.
40. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 1–9. Boston, MA, USA (2015), https://doi.org/10.1109/CVPR.2015.7298594.
41. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Vol. 2. pp. 2377–2385. NIPS'15, MIT Press, Montreal, Canada (2015).
42. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems* 27, pp. 3320–3328. Curran Associates, Inc. (2014).
43. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: DeCAF: A deep convolutional activation feature for generic visual recognition. In: Xing, E.P., Jebara, T. (eds.) *Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research*, Vol. 32, pp. 647–655. PMLR, Bejing, China (22–24 June 2014).
44. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *Computer Vision—ECCV 2014*. pp. 584–599. Springer International Publishing, Cham (2014), https://doi.org/10.1007/978–3-319–10590–1_38.
45. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: An astounding baseline for recognition. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. pp. 512–519 (June 2014), https://doi.org/10.1109/CVPRW.2014.131.
46. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Gordon, G., Dunson, D., Dudík, M. (eds.) *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, Vol. 15, pp. 315–323. PMLR, Fort Lauderdale, FL, USA (11–13 Apr 2011).

47. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (eds.) *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, Vol. 9, pp. 249–256. PMLR, Chia Laguna Resort, Sardinia, Italy (13–15 May 2010).
48. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1929–1958 (2014).
49. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11, 3371–3408 (2010).
50. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional Nets. In: Valstar, M., French, A., Pridmore, T. (eds.) *Proceedings of the British Machine Vision Conference*. BMVA Press (2014), http://dx.doi.org/10.5244/C.28.6.
51. Lasagne: https://lasagne.readthedocs.io/en/latest/, Accessed: September 01, 2019.
52. Keras: https://keras.io/, Accessed: September 01, 2019.
53. Neidinger, R.: Introduction to automatic differentiation and MAT-LAB object-oriented programming. *SIAM Review* 52(3), 545–563 (2010), https://doi.org/10.1137/080743627.
54. Hosang, J., Benenson, R., Dollar, P., Schiele, B.: What makes for effective detection proposals? *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(4), 814–830 (April 2016), https://doi.org/10.1109/TPAMI.2015.2465908.
55. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 580–587. IEEE, Columbus, OH, USA (June 2014), https://doi.org/10.1109/CVPR.2014.81.
56. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 3431–3440. IEEE, Boston, MA, USA (June 2015), https://doi.org/10.1109/CVPR.2015.7298965.
57. Girshick, R.: Fast R-CNN. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. pp. 1440–1448. IEEE, Santiago, Chile (Dec 2015), https://doi.org/10.1109/ICCV.2015.169.
58. Brody, H.: Medical imaging. *Nature* 502, S81 (10/30 2013).
59. Shen, D., Wu, G., Suk, H.I.: Deep learning in medical image analysis. *Annual review of biomedical engineering* 19, 221–248 (June 21 2017).
60. Schmidhuber, J.: Deep learning in neural networks: An overview. *Neural Networks* 61, 85–117 (2015).
61. Wu, G., Kim, M., Wang, Q., Munsell, B.C., Shen, D.: Scalable high-performance image registration framework by unsupervised deep feature representations learning. *IEEE Transactions on Biomedical Engineering* 63(7), 1505–1516 (July 2016), https://doi.org/10.1109/TBME.2015.2496253.
62. Wu, G., Kim, M., Wang, Q., Gao, Y., Liao, S., Shen, D.: Unsupervised deep feature learning for deformable registration of MR brain images. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2013*. Lecture Notes in Computer Science, Vol. 8150, pp. 649–656. Springer Berlin Heidelberg, Berlin, Heidelberg (2013).

63. Liao, S., Gao, Y., Oto, A., Shen, D.: Representation learning: A unified deep learning framework for automatic prostate MR segmentation. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2013.* Lecture Notes in Computer Science, Vol. 8150, pp. 254–261. Springer Berlin Heidelberg, Berlin, Heidelberg (2013).
64. Guo, Y., Gao, Y., Shen, D.: Deformable MR prostate segmentation via deep feature learning and sparse patch matching. *IEEE Transactions on Medical Imaging* 35(4), 1077–1089 (April 2016), https://doi.org/10.1109/TMI.2015.2508280.
65. Kim, M., Wu, G., Shen, D.: Unsupervised deep learning for hippocampus segmentation in 7.0 Tesla MR images. In: Wu, G., Zhang, D., Shen, D., Yan, P., Suzuki, K., Wang, F. (eds.) *Machine Learning in Medical Imaging.* Lecture Notes in Computer Science, Vol. 8184, pp. 1–8. Springer International Publishing, Cham (2013), https://doi.org/10.1007/978–3-319–02267–3_1.
66. Abd-Ellah, M.K., Awad, A.I., Khalaf, A.A.M., Hamed, H.F.A.: Classification of brain tumor MRIs using a kernel support vector machine. In: Li, H., Nykanen, P., Suomi, R., Wickramasinghe, N., Widen, G., Zhan, M. (eds.) *Building Sustainable Health Ecosystems, WIS 2016. Communications in Computer and Information Science*, Vol. 636. Springer, Cham, Tampere, Finland (2016).
67. Abd-Ellah, M.K., Awad, A.I., Khalaf, A.A.M., Hamed, H.F.A.: Design and implementation of a computer-aided diagnosis system for brain tumor classification. In: *2016 28th International Conference on Microelectronics (ICM).* pp. 73–76. IEEE, Giza, Egypt (2016).
68. Roth, H.R., Lu, L., Liu, J., Yao, J., Seff, A., Cherry, K., Kim, L., Summers, R.M.: Improving computer-aided detection using convolutional neural networks and random view aggregation. *IEEE Transactions on Medical Imaging* 35(5), 1170–1181 (May 2016), https://doi.org/10.1109/TMI.2015.2482920.
69. Ciompi, F., de Hoop, B., van Riel, S.J., Chung, K., Scholten, E.T., Oudkerk, M., de Jong, P.A., Prokop, M., van Ginneken, B.: Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. *Medical Image Analysis* 26(1), 195–202 (September 2015), https://doi.org/10.1016/j.media.2015.08.001.
70. Gao, M., Bagci, U., Lu, L., Wu, A., Buty, M., Shin, H.C., Roth, H., Papadakis, G.Z., Depeursinge, A., Summers, R.M., Xu, Z., Mollura, D.J.: Holistic classification of CT attenuation patterns for interstitial lung diseases via deep convolutional neural networks. *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization* 6(1), 1–6 (2016), https://doi.org/10.1080/21681163.2015.1124249.
71. Brosch, T., Tang, L.Y.W., Yoo, Y., Li, D.K.B., Traboulsee, A., Tam, R.: Deep 3D convolutional encoder networks with shortcuts for multi-scale feature integration applied to multiple sclerosis lesion segmentation. *IEEE Transactions on Medical Imaging* 35(5), 1229–1239 (May 2016), https://doi.org/10.1109/TMI.2016.2528821.
72. Dou, Q., Chen, H., Yu, L., Zhao, L., Qin, J., Wang, D., Mok, V.C., Shi, L., Heng, P.: Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks. *IEEE Transactions on Medical Imaging* 35(5), 1182–1195 (May 2016), 10.1109/TMI.2016.2528129.
73. Abd-Ellah, M.K., Awad, A.I., Khalaf, A.A.M., Hamed, H.F.A.: A review on brain tumor diagnosis from MRI images: Practical implications, key achievements, and lessons learned. *Magnetic Resonance Imaging* 61, 300–318 (2019), https://doi.org/10.1016/j.mri.2019.05.028.

74. Ciresan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Mitosis detection in breast cancer histology images with deep neural networks. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2013*. pp. 411–418. Springer International Publishing, Berlin, Heidelberg (2013), https://doi.org/10.1007/978–3-642–40763–5_51.
75. Kleesiek, J., Urban, G., Hubert, A., Schwarz, D., Maier-Hein, K., Bendszus, M., Biller, A.: Deep MRI brain extraction: A 3D convolutional neural network for skull stripping. *NeuroImage* 129, 460–469 (2016).
76. Moeskops, P., Viergever, M.A., Mendrik, A.M., de Vries, L.S., Benders, M.J.N.L., Isgum, I.: Automatic segmentation of MR brain images with a convolutional neural network. *IEEE Transactions on Medical Imaging* 35(5), 1252–1261 May 2016).
77. Weisenfeld, N.I., Warfield, S.K.: Automatic segmentation of newborn brain MRI. *NeuroImage* 47(2), 564–572 (2009).
78. Zhang, W., Li, R., Deng, H., Wang, L., Lin, W., Ji, S., Shen, D.: Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. *NeuroImage* 108, 214–224 (2015).
79. Nie, D., Wang, L., Gao, Y., Shen, D.: Fully convolutional networks for multi-modality isointense infant brain image segmentation. In: *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. pp. 1342–1345 (April 2016).
80. Zhao, X., Wu, Y., Song, G., Li, Z., Fan, Y., Zhang, Y.: Brain tumor segmentation using a fully convolutional neural network with conditional random fields. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Winzeck, S., Handels, H. (eds.) *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Lecture Notes in Computer Science, Vol. 10154, pp. 75–87. Springer International Publishing, Cham (2016).
81. Casamitjana, A., Puch, S., Aduriz, A., Vilaplana, V.: 3D convolutional neural networks for brain tumor segmentation: A comparison of multi-resolution architectures. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Winzeck, S., Handels, H. (eds.) *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Lecture Notes in Computer Science, Vol. 10154, pp. 150–161. Springer International Publishing, Cham (2016).
82. Pereira, S., Oliveira, A., Alves, V., Silva, C.A.: On hierarchical brain tumor segmentation in MRI using fully convolutional neural networks: A preliminary study. In: *2017 IEEE 5th Portuguese Meeting on Bioengineering (ENBENG)*. pp. 1–4 (Feb 2017).
83. Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.M., Larochelle, H.: Brain tumor segmentation with deep neural networks. *Medical Image Analysis* 35, 18–31 (2017), https://doi.Org/10.1016/j.media.2016.05.004.
84. Wang, G., Li, W., Ourselin, S., Vercauteren, T.: Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation. In: Crimi, A., Bakas, S., Kuijf, H., Keyvan, F., Reyes, M., van Walsum, T. (eds.) *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. pp. 61–72. Springer International Publishing, Cham (2019), https://doi.org/10.1007/978–3-030–11726–9_6.
85. Abd-Ellah, M.K., Khalaf, A.A.M., Awad, A.I., Hamed, H.F.A.: TPUAR-Net: Two parallel U-Net with asymmetric residual-based deep convolutional neural network for brain tumor segmentation. In: Karray, F., Campilho, A., Yu, A. (eds.) *Image Analysis and Recognition. ICIAR 2019*. Lecture Notes in Computer Science, Vol. 11663, pp. 106–116. Springer International Publishing, Cham (2019).
86. The Cancer Imaging Archive: RIDER NEURO MRI database. https://wiki.cancerimagingarchive.net/display/Public/RIDER+NEURO+MRI (2016), Accessed: February 5th, 2019.
87. Tharwat, A.: Classification assessment methods. *Applied Computing and Informatics* (2018), https://doi.org/10.1016/j.aci.2018.08.003.

88. Mohsen, H., El-Dahshan, E.S.A., El-Horbaty, E.S.M., Salem, A.B.M.: Classification using deep learning neural networks for brain tumors. *Future Computing and Informatics Journal* 3, 68–71 (2018).
1. American Joint Committee on Cancer, "Melanoma of the skin," *Cancer Staging Manual*, pp. 209–220, Springer, New York, NY, 2002.
2. M. E. Celebi, H. A. Kingravi, B. Uddin, H. Lyatornid, Y. A. Aslandogan, W. V. Stoecker, and R. H. Moss, "A methodological approach to the classification of dermoscopy images," *Computerized Medical Imaging and Graphics*, vol. 31, no. 6, pp. 362–373, September, 2007.
3. R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics," *CA Cancer Journal for Clinicians*, vol. 68, no. 1, pp. 7–30, 2018.
4. American Cancer Society, "Cancer facts & figures 2018, American Cancer Society, Atlanta, 2018," Accessed [September 3, 2018]; https://www.cancer.org/cancer/melanoma-skin-cancer.html.
5. A. M. Noone, N. Howlader, M. Krapcho, D. Miller, A. Brest, M. Yu, J. Ruhl, Z. Tatalovich, A. Mariotto, D. R. Lewis, H. S. Chen, E. J. Feuer, and K. A. Cronin, "SEER cancer statistics review, 1975–2015," National Cancer Institute, 2018.
6. M. E. Vestergaard, P. Macaskill, P. E. Holt, and S. W. Menzies, "Dermoscopy compared with naked eye examination for the diagnosis of primary melanoma: A meta-analysis of studies performed in a clinical setting," *British Journal of Dermatology*, vol. 159, no. 3, pp. 669–676, 2008.
7. M. Silveira, J. C. Nascimento, J. S. Marques, A. R. Marçal, T. Mendonça, S. Yamauchi, J. Maeda, and J. Rozeira, "Comparison of segmentation methods for melanoma diagnosis in dermoscopy images," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 1, pp. 35–45, 2009.
8. M. E. Celebi, H. Iyatomi, G. Schaefer, and W. V. Stoecker, "Lesion border detection in dermoscopy images," *Computerized Medical Imaging and Graphics*, vol. 33, no. 2, pp. 148–53, March, 2009.
9. H. Ganster, A. Pinz, R. Rohrer, E. Wildling, M. Binder, and H. Kittler, "Automated melanoma recognition," *IEEE Transactions on Medical Imaging*, vol. 20, no. 3, pp. 233–239, March, 2001.
10. E. Meskini, M. S. Helfroush, K. Kazemi, and M. Sepaskhah, "A new algorithm for skin lesion border detection in dermoscopy images," *Journal of Biomedical Physics and Engineering*, vol. 8, no. 1, pp. 117–126, 2018.
11. G. Schaefer, B. Krawczyk, M. E. Celebi, and H. Iyatomi, "An ensemble classification approach for melanoma diagnosis," *Memetic Computing*, vol. 6, no. 4, pp. 233–240, December, 2014.
12. M. A. Al-Masni, M. A. Al-antari, M. T. Choi, S. M. Han, and T. S. Kim, "Skin lesion segmentation in dermoscopy images via deep full resolution convolutional networks," *Computer Methods and Programs in Biomedicine*, vol. 162, pp. 221–231, August, 2018.
13. M. E. Yuksel, and M. Borlu, "Accurate segmentation of dermoscopic images by image thresholding based on type-2 fuzzy logic," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 976–982, August, 2009.
14. K. Mollersen, H. M. Kirchesch, T. G. Schopf, and F. Godtliebsen, "Unsupervised segmentation for digital dermoscopic images," *Skin Research and Technology*, vol. 16, no. 4, pp. 401–407, November, 2010.
15. M. E. Celebi, Q. Wen, S. Hwang, H. Iyatomi, and G. Schaefer, "Lesion border detection in dermoscopy images using ensembles of thresholding methods," *Skin Research and Technology*, vol. 19, no. 1, pp. E252–E258, 2013.

16. F. Peruch, F. Bogo, M. Bonazza, V. M. Cappelleri, and E. Peserico, "Simpler, faster, more accurate melanocytic lesion segmentation through MEDS," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 2, pp. 557–565, February, 2014.
17. H. Y. Zhou, G. Schaefer, A. H. Sadka, and M. E. Celebi, "Anisotropic mean shift based fuzzy c-means segmentation of dermoscopy images," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 1, pp. 26–34, February, 2009.
18. S. Kockara, M. Mete, V. Yip, B. Lee, and K. Aydin, "A soft kinetic data structure for lesion border detection," *Bioinformatics*, vol. 26, no. 12, pp. i21–i28, June 15, 2010.
19. S. Suer, S. Kockara, and M. Mete, "An improved border detection in dermoscopy images for density based clustering," *BMC Bioinformatics*, vol. 12, no. 10, p. S12.
20. F. Y. Xie, and A. C. Bovik, "Automatic segmentation of dermoscopy images using self-generating neural networks seeded by genetic algorithm," *Pattern Recognition*, vol. 46, no. 3, pp. 1012–1019, March, 2013.
21. M. E. Celebi, H. A. Kingravi, H. Iyatomi, Y. A. Aslandogan, W. V. Stoecker, R. H. Moss, J. M. Malters, J. M. Grichnik, A. A. Marghoob, H. S. Rabinovitz, and S. W. Menzies, "Border detection in dermoscopy images using statistical region merging," *Skin Research and Technology*, vol. 14, no. 3, pp. 347–353, August, 2008.
22. Q. Abbas, M. E. Celebi, and I. F. Garcia, "Skin tumor area extraction using an improved dynamic programming approach," *Skin Research and Technology*, vol. 18, no. 2, pp. 133–142, May, 2012.
23. Q. Abbas, M. E. Celebi, I. F. Garcia, and M. Rashid, "Lesion border detection in dermoscopy images using dynamic programming," *Skin Research and Technology*, vol. 17, no. 1, pp. 91–100, February, 2011.
24. B. Erkol, R. H. Moss, R. J. Stanley, W. V. Stoecker, and E. Hvatum, "Automatic lesion boundary detection in dermoscopy images using gradient vector flow snakes," *Skin Research and Technology*, vol. 11, no. 1, pp. 17–26, February, 2005.
25. M. Mete, and N. M. Sirakov, "Lesion detection in demoscopy images with novel density-based and active contour approaches," *BMC Bioinformatics*, vol. 11, no. 6, p. S23, 2010.
26. A. R. Sadri, M. Zekri, S. Sadri, N. Gheissari, M. Mokhtari, and F. Kolahdouzan, "Segmentation of dermoscopy images using wavelet networks," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 1134–1141, 2013.
27. K. Korotkov, and R. Garcia, "Computerized analysis of pigmented skin lesions: A review," *Artificial Intelligence in Medicine*, vol. 56, no. 2, pp. 69–90, October, 2012.
28. M. E. Celebi, Q. Wen, H. Iyatomi, K. Shimizu, H. Zhou, and G. Schaefer, "A state-of-the-art survey on lesion border detection in dermoscopy images," *Dermoscopy Image Analysis*, pp. 97–129, CRC Press, 2015.
29. R. B. Oliveira, J. P. Papa, A. S. Pereira, and J. M. R. S. Tavares, "Computational methods for pigmented skin lesion classification in images: Review and future trends," *Neural Computing & Applications*, vol. 29, no. 3, pp. 613–636, February, 2018.
30. S. Pathan, K. G. Prabhu, and P. C. Siddalingaswamy, "Techniques and algorithms for computer aided diagnosis of pigmented skin lesions-A review," *Biomedical Signal Processing and Control*, vol. 39, pp. 237–262, 2018.
31. M. A. Al-Masni, M. A. Al-Antari, J. M. Park, G. Gi, T. Y. Kim, P. Rivera, E. Valarezo, M. T. Choi, S. M. Han, and T. S. Kim, "Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system," *Computer Methods and Programs in Biomedicine*, vol. 157, pp. 85–94, 2018.
32. G. Carneiro, J. Nascimento, and A. P. Bradley, "Automated analysis of unregistered multi-view mammograms with deep learning," *IEEE Transactions on Medical Imaging*, vol. 36, no. 11, pp. 2355–2365, 2017.

33. N. Dhungel, G. Carneiro, and A. P. Bradley, "A deep learning approach for the analysis of masses in mammograms with minimal user intervention," *Medical Image Analysis*, vol. 37, pp. 114–128, April, 2017.
34. G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. van der Laak, B. van Ginneken, and C. I. Sanchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, December, 2017.
35. X. Zhao, Y. Wu, G. Song, Z. Li, Y. Zhang, and Y. Fan, "A deep learning model integrating FCNNs and CRFs for brain tumor segmentation," *Medical Image Analysis*, vol. 43, pp. 98–111, January, 2018.
36. E. Gibson, W. Li, C. Sudre, L. Fidon, D. I. Shakir, G. Wang, Z. Eaton-Rosen, R. Gray, T. Doel, Y. Hu, T. Whyntie, P. Nachev, M. Modat, D. C. Barratt, S. Ourselin, M. J. Cardoso, and T. Vercauteren, "NiftyNet: A deep-learning platform for medical imaging," *Computer Methods and Programs in Biomedicine*, vol. 158, pp. 113–122, May, 2018.
37. M. A. Al-Antari, M. A. Al-Masni, M. T. Choi, S. M. Han, and T. S. Kim, "A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification," *International Journal of Medical Informatics*, vol. 117, pp. 44–54, September, 2018.
38. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–44, May 28, 2015.
39. M. A. Al-Masni, M. A. Al-Antari, J. M. Park, G. Gi, T. Y. Kim, P. Rivera, E. Valarezo, S. M. Han, and T. S. Kim, "Detection and classification of the breast abnormalities in digital mammograms via regional Convolutional Neural Network," in *Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jeju Island, Republic of Korea, 2017, pp. 1230–1233.
40. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, June, 2017.
41. D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in Neural Information Processing Systems*, 2012, pp. 2843–2851.
42. C. Cernazanu-Glavan, and S. Holban, "Segmentation of bone structure in X-ray images using convolutional neural network," *Advances in Electrical and Computer Engineering*, vol. 13, no. 1, pp. 87–94, 2013.
43. M. Melinščak, P. Prentašić, and S. Lončarić, "Retinal vessel segmentation using deep neural networks," in *10th International Conference on Computer Vision Theory and Applications (VISAPP 2015)*, pp. 577–582, 2015.
44. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.
45. E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, April, 2017.
46. A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," arXiv preprint arXiv:1704.06857, 2017.
47. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

48. V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, December, 2017.
49. V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” arXiv preprint arXiv:1511.00561, 2015.
50. H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.
51. L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, April, 2018.
52. L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” CoRR abs/1606.00915, 2016.
53. Y. X. Li, and L. L. Shen, “Skin lesion analysis towards melanoma detection using deep learning network,” *Sensors*, vol. 18, no. 2, February, 2018.
54. L. Bi, J. Kim, E. Ahn, A. Kumar, M. Fulham, and D. Feng, “Dermoscopic image segmentation via multistage fully convolutional networks,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2065–2074, 2017.
55. L. Q. Yu, H. Chen, Q. Dou, J. Qin, and P. A. Heng, “Automated melanoma recognition in dermoscopy images via very deep residual networks,” *IEEE Transactions on Medical Imaging*, vol. 36, no. 4, pp. 994–1004, 2017.
56. Y. D. Yuan, M. Chao, and Y. C. Lo, “Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance,” *IEEE Transactions on Medical Imaging*, vol. 36, no. 9, pp. 1876–1886, 2017.
57. M. Goyal, and M. H. Yap, “Multi-class semantic segmentation of skin lesions via fully convolutional networks,” arXiv preprint arXiv:1711.10449, 2017.
58. B. S. Lin, K. Michael, S. Kalra, and H. R. Tizhoosh, “Skin lesion segmentation: U-Nets versus clustering,” in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–7.
59. Y. Yuan, “Automatic skin lesion segmentation with fully convolutional-deconvolutional networks,” arXiv preprint arXiv:1703.05165, 2017.
60. N. C. F. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, and A. Halpern, “Skin lesion analysis toward melanoma detection: A challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), hosted by the International Skin Imaging Collaboration (ISIC),” in *15th IEEE International Symposium on Biomedical Imaging (ISBI 2018)*, 2018, pp. 68–172.
61. International Skin Imaging Collaboration, “ISIC 2017: Skin lesion analysis towards melanoma detection,” Accessed [October 19, 2018]; https://challenge.kitware.com/#challenges.
62. Z. Jiao, X. Gao, Y. Wang, and J. Li, “A deep feature based framework for breast masses classification,” *Neurocomputing* vol. 197, pp. 221–231, 2016.
63. T. Kooi, G. Litjens, B. van Ginneken, A. Gubern-Merida, C. I. Sancheza, R. Mann, A. den Heeten, and N. Karssemeijer, “Large scale deep learning for computer aided detection of mammographic lesions,” *Medical Image Analysis*, vol. 35, pp. 303–312, January, 2017.

64. H. R. Roth, L. Lu, J. M. Liu, J. H. Yao, A. Seff, K. Cherry, L. Kim, and R. M. Summers, "Improving computer-aided detection using convolutional neural networks and random view aggregation," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1170–1181, May, 2016.
65. M. Lin, Q. Chen, and S. Yan, "Network in network," in rXiv preprint arXiv:1312.4400, pp. 1–10, 2013.
66. D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Artificial Neural Networks–ICANN 2010*, 2010, pp. 92–101.
67. K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
68. Q. Guo, F. L. Wang, J. Lei, D. Tu, and G. H. Li, "Convolutional feature learning and Hybrid CNN-HMM for scene number recognition," *Neurocomputing*, vol. 184, pp. 78–90, April 5, 2016.
69. A. L. Maas, A. Y. Hannun, and A. Y. Ng., "Rectifier nonlinearities improve neural network acoustic models," in *Proceeding of 30th International Conference on Machine Learning (ICML)*, 2013, p. 3.
70. X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.
71. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
72. S. Hoo-Chang, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.
73. Y. Bar, I. Diamant, L. Wolf, S. Lieberman, E. Konen, and H. Greenspan, "Chest pathology identification using deep feature selection with non-medical training," *Computer Methods in Biomechanics and Biomedical Engineering-Imaging and Visualization*, vol. 6, no. 3, pp. 259–263, 2018.
74. R. K. Samala, H. P. Chan, L. Hadjiiski, M. A. Helvie, J. Wei, and K. Cha, "Mass detection in digital breast tomosynthesis: Deep convolutional neural network with transfer learning from mammography," *Medical Physics*, vol. 43, no. 12, pp. 6654–6666, December, 2016.
75. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.
76. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. H. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, December, 2015.
77. E. Szymanska, E. Saccenti, A. K. Smilde, and J. A. Westerhuis, "Double-check: validation of diagnostic statistics for PLS-DA models in metabolomics studies," *Metabolomics*, vol. 8, no. 1, pp. S3–S16, June, 2012.
78. S. Smit, M. J. van Breemen, H. C. J. Hoefsloot, A. K. Smilde, J. M. F. G. Aerts, and C. G. de Koster, "Assessing the statistical validity of proteomics based biomarkers," *Analytica Chimica Acta*, vol. 592, no. 2, pp. 210–217, June 5, 2007.
79. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning; Data Mining, Inference and Prediction*, Second ed., Springer, New York, 2008.
80. S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in Bioinformatics*, vol. 18, no. 5, pp. 851–869, September, 2017.

81. Image Segmentation Keras, "Implementation of Segnet, FCN, UNet and other models in Keras," Accessed [October 23, 2018]; https://github.com/divamgupta/image-segmentation-keras.
82. Keras, "Keras: The python deep learning library," Accessed [January 8, 2019]; https://keras.io/.
83. M. Dong, X. Lu, Y. Ma, Y. Guo, Y. Ma, and K. Wang, "An efficient approach for automated mass segmentation and classification in mammograms," *Journal of Digital Imaging*, vol. 28, no. 5, pp. 613–25, 2015.
84. M. A. Al-antari, M. A. Al-masni, S. U. Park, J. Park, M. K. Metwally, Y. M. Kadah, S. M. Han, and T. S. Kim, "An automatic computer-aided diagnosis system for breast cancer in digital mammograms via deep belief network," *Journal of Medical and Biological Engineering*, vol. 38, no. 3, pp. 443–456, June, 2018.
85. D. Zikic, Y. Ioannou, M. Brown, and A. Criminisi, "Segmentation of brain tumor tissues with convolutional neural networks," in *MICCAI-BRATS*, 2014, pp. 36–39.
86. S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in MRI images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1240–1251, May, 2016.
87. D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.
88. M. D. Zeiler, "ADADELTA: An adaptive learning rate method," arXiv preprint arXiv:1212.5701, 2012.
89. D. P. Kingma, and J. L. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, arXiv preprint arXiv:1412.6980, 2015.
90. T. Tieleman, and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," 4, http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf, 2012.
91. M. Berseth, "ISIC 2017-skin lesion analysis towards melanoma detection," arXiv preprint arXiv:1703.00523v1, 2017.
92. L. Bi, J. Kim, E. Ahn, and D. Feng, "Automatic skin lesion analysis using large-scale dermoscopy images and deep residual networks," arXiv preprint arXiv:1703.04197, 2017.
93. A. Menegola, J. Tavares, M. Fornaciali, L. T. Li, S. Avila, and E. Valle, "RECOD titans at ISIC challenge 2017," arXiv preprint arXiv:1703.04819, 2017.
1. I. Maglogiannis, and C. N. Doukas, "Overview of advanced computer vision systems for skin lesions characterization", *Transactions on Information Technology in Biomedicine*, vol. 13, no. 5, pp. 721–733, 2009.
2. M. S. Arifin, M. G. Kibria, A. Firoze, M. A. Amini, and H. Yan, "Dermatological disease diagnosis using color-skin images", *International Conference on Machine Learning and Cybernetics*, vol. 5, pp. 1675–1680, 2012.
3. J. M. Gálvez, D. Castillo, L. J. Herrera, B. S. Román, O. Valenzuela, F. M. Ortuño, and I. Rojas, "Multiclass classification for skin cancer profiling based on the integration of heterogeneous gene expression series", *PLoS ONE*, vol. 13, no. 5, pp. 1–26, 2018.
4. A. Masood, A. A. Al-Jumaily, and T. Adnan, "Development of automated diagnostic system for skin cancer: Performance analysis of neural network learning algorithms for classification", *International Conference on Artificial Neural Networks*, pp. 837–844, 2014.
5. P. G. Cavalcanti, and J. Scharcanski, "Macroscopic pigmented skin lesion segmentation and its influence on lesion classification and diagnosis", *Color Medical Image Analysis*, vol. 6, pp. 15–39, 2013.

6. M. J. M. Vasconcelos, and L. Rosado, "No-reference blur assessment of dermatological images acquired via mobile devices", *Image and Signal Processing*, vol. 8509, pp. 350–357, 2014.
7. K. Bunte, M. Biehl, M. F. Jonkman, and N. Petkov, "Learning effective color features for content-based image retrieval in dermatology", *Pattern Recognition*, vol. 44, pp. 1892–1902, 2011.
8. S. V. Patwardhan, A. P. Dhawan, and P. A. Relue, "Classification of melanoma using tree structured wavelet transforms", *Computer Methods and Programs in Biomedicine*, vol. 72, no. 3, pp. 223–239, 2003.
9. W. Y. Chang, A. Huang, C. Y. Yang, C. H. Lee, Y. C. Chen, T. Y. Wu, and G. S. Chen, "Computer-aided diagnosis of skin lesions using conventional digital photography: A reliability and feasibility study", *PLoS ONE*, vol. 8, no. 11, pp. 1–9, 2013.
10. S. Kundu, N. Das, and M. Nasipuri, "Automatic detection of ringworm using Local Binary Pattern (LBP)", 2011, https://arxiv.org/abs/1103.0120.
11. R. Amelard, A. Wong, and D. A. Clausi, "Extracting morphological high-level intuitive features (HLIF) for enhancing skin lesion classification", *International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4458–4461, 2012.
12. E. M. Karabulut, and T. Ibrikci, "Texture analysis of melanoma images for computer-aided diagnosis", *International Conference on Intelligent Computing, Computer Science and Information Systems (ICCSIS)*, vol. 2, pp. 26–29, 2016.
13. J. A. Almaraz-Damian, V. Ponomaryov, and E. R. Gonzalez, "Melanoma CADe based on ABCD rule and Haralick texture features", *International Kharkiv Symposium on Physics and Engineering of Microwaves, Millimeter and Submillimeter Waves (MSMW)*, pp. 1–4, 2016.
14. I. Giotis, N. Molders, S. Land, M. Biehl, M. F. Jonkman, and N. Petkov, "MED-NODE: A computer-assisted melanoma diagnosis system using non-dermoscopic images", *Expert Systems with Applications*, vol. 42, no. 19, pp. 6578–6585, 2015.
15. M. H. Jafari, S. Samavi, N. Karimi, S. M. R. Soroushmehr, K. Ward, and K. Najarian, "Automatic detection of melanoma using broad extraction of features from digital images", *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1357–1360, 2016.
16. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
17. P. Dubal, S. Bhatt, C. Joglekar, and S. Patil, "Skin cancer detection and classification", *International Conference on Electrical Engineering and Informatics (ICEEI)*, pp. 1–6, 2017.
18. D. Kumar, M. J. Shafiee, A. Chung, F. Khalvati, M. Haider, and A. Wong, "Discovery radiomics for computed tomography cancer detection", 2015, https://arxiv.org/abs/1509.00117.
19. E. Nasr-Esfahan, S. Samavi, N. Karimi, S. M. R. Soroushmehr, M. H. Jafari, K. Ward, and K. Najarian, "Melanoma detection by analysis of clinical images using convolutional neural network", *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1373–1376, 2016.
20. J. Premaladha, and K. S. Ravichandran, "Novel approaches for diagnosing melanoma skin lesions through supervised and deep learning algorithms", *Journal of Medical Systems*, vol. 40, no. 96, pp. 1–12, 2016.
21. S. A. Kostopoulos et al., "Adaptable pattern recognition system for discriminating Melanocytic Nevi from Malignant Melanomas using plain photography images from different image databases", *International Journal of Medical Informatics*, vol. 105, pp. 1–10, 2017.
22. A. Esteva, B. Kuprel, H. M. Blau, S. M. Swetter, J. Ko, R. A. Novoa, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks", *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

23. S. Jain, V. Jagtap, and N. Pise, "Computer aided melanoma skin cancer detection using image processing", *Procedia Computer Science*, vol. 48, pp. 735–740, 2015.
24. D. Gautam, and M. Ahmed, "Melanoma detection and classification using SVM based decision support system", *IEEE India Conference (INDICON)*, pp. 1–6, 2015.
25. Convolutional Neural Networks (CNNs / ConvNets), the Stanford CS class notes, Assignments, Spring 2017, http://cs231n.github.io/convolutional-networks/, Accessed: 18 August 2017.
26. R. D. Azulay, D. R. Azulay, and L. Azulay-Abulafia, *Dermatologia*. 6th edition. Rio de Janeiro: Guanabara Koogan, 2013.
27. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 101–118, 1998.
28. S. Srinivas, R. K. Sarvadevabhatl, K. R. Mopur, N. Prabhu, S. S. S. Kruthiventi, and R. V. Babu, "A taxonomy of deep convolutional neural nets for computer vision", *Frontiers in Robotics and AI*, vol. 2, pp. 1–18, 2016.
29. A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks", *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 1097–1105, 2012.
30. O. Russakovsky et al., "ImageNet large scale visual recognition challenge", *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
31. J. D. Prusa, and T. M. Khoshgoftaar, "Improving deep neural network design with new text data representations", *Journal of Big Data*, Springer, vol. 4, pp. 1–16, 2017.
32. T. A. Martin, A. J. Sanders, L. Ye, J. Lane, and W. G. jiang, "Cancer invasion and metastasis: Molecular and cellular perspective", Landes Bioscience, pp. 135–168, 2013.
33. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems*, vol. 2, pp. 3320–3328, 2014.
34. S. J. Pan, and Q. Yang, "A survey on transfer learning", *Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
35. K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning", *Journal of Big Data*, Springer, vol. 3, no 9, pp. 1–15, 2016.
36. Dermatology Information System, 2012, http://www.dermis.net, Accessed: 16 August 2017.
37. DermQuest, 2012, http://www.dermquest.com, Accessed: 16 August 2017.
38. C. Chih-Chung, and L. Chih-Jen, "LIBSVM: A library for support vector machines", *Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, 2013.
39. B. Basavaprasad, and R. S. Hegad, "Color image segmentation using adaptive Growcut method", *Procedia Computer Science*, vol. 45, pp. 328–335, 2015.
40. M. Stojanovi et al., "Understanding sensitivity, specificity and predictive values", *Vojnosanitetski pregled*, vol. 71, no. 11, pp. 1062–1065, 2014.
41. T. Fawcett, "An introduction to ROC analysis", *Pattern Recognition Letter*, vol. 27, no. 8, pp. 861–874, 2006.
42. K. M. Hosny, M. A. Kassem, and M. M. Foaud, "Classification of skin lesions using transfer learning and augmentation with Alex-net", *PLoS ONE*, vol. 14, no. 5, pp. 1–17, 2019.
43. K. M. Hosny, M. A. Kassem, and M. M. Foaud, "Skin cancer classification using deep learning and transfer learning", *Cairo International Biomedical Engineering Conference (CIBEC)*, pp. 90–93, 2018.
44. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.