Lecture Notes in Electrical Engineering 722

T. Laxminidhi
Jyoti Singhai
Sreehari Rao Patri
V. V. Mani *Editors*

Advances in Communications, Signal Processing, and VLSI

Select Proceedings of IC2SV 2019

Springer

Lecture Notes in Electrical Engineering

Volume 722

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering - quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Editorial Director (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada:

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries:

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** This series is indexed by EI Compendex and Scopus databases. ****

More information about this series at http://www.springer.com/series/7818

T. Laxminidhi · Jyoti Singhai · Sreehari Rao Patri · V. V. Mani
Editors

Advances in Communications, Signal Processing, and VLSI

Select Proceedings of IC2SV 2019

Springer

Editors
T. Laxminidhi
National Institute of Technology Karnataka
Mangalore, India

Sreehari Rao Patri
National Institute of Technology Warangal
Warangal, Telangana, India

Jyoti Singhai
Department of Electronics
and Communication Engineering
Maulana Azad National Institute
of Technology
Bhopal, Madhya Pradesh, India

V. V. Mani
Department of Electronics
and Communication Engineering
National Institute of Technology Warangal
Warangal, India

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-33-4057-2 ISBN 978-981-33-4058-9 (eBook)
https://doi.org/10.1007/978-981-33-4058-9

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Organization

Program Chairs

Dr. Venkata Mani Vakamulla, NIT, Warangal, India
Dr. Sreehari Rao Patri, NIT, Warangal, India

Preface

This book presents the advances in communications, signal processing, and VLSI taken from the excerpts of the proceedings of the International Conference on Communications, Signal Processing and VLSI held by the ECE Department, National Institute of Technology Warangal during October 23–24, 2019. This conference offered an excellent forum for exchange of ideas among interested researchers, students, peers, and practitioners in the upcoming areas of signal processing, 5G communications, chip design, image processing, and machine learning. It focused on the congregation of the deft designers across academy and industry, aiming to improve the living standards of mankind.

The selected proceedings of IC2SV 2019 included in this book were peer reviewed and thoroughly checked for plagiarism using Turnitin software. We thank the administration, NIT Warangal, for their support and all the authors for sharing their research outcomes that helped to publish this book.

Mangalore, India — T. Laxminidhi
Bhopal, India — Jyoti Singhai
Warangal, India — Sreehari Rao Patri
Warangal, India — V. V. Mani

Contents

About the Editors

Dr. T. Laxminidhi is currently working as a Professor at the National Institute of Technology, (Surathkal) Karnataka India. He has over 20 years of teaching and research experience in his specific areas of interests, which include analog and mixed signal IC design and signal processing. He has published several reputed international journal papers and international conference papers. He received his B.Tech degree from Mangalore University and ME degrees in Industrial Electronics from NIT Karnataka. He obtained his PhD from IIT Madras.

Dr. Jyoti Singhai is currently a Professor at the department of Electronics and Communication Engineering, Maulana Azad National Institute of Technology Bhopal. She obtained her B.E. (Electronics), M.Tech (Digital Communication) and PhD from MANIT Bhopal. Her major areas of research interests include Image and video processing, wireless communication and routing protocols. She has published 50 papers in respected international journals. She received the DST BOYSCAST Fellowship in 2010, MP Young Scientist award and AICTE Career Award to Young Teachers.

Dr. Sreehari Rao Patri is currently working as an Associate Professor at the National Institute of Technology in Warangal, India. He has over 20 years of teaching and research experience in his specific areas of interests, which include analog and mixed signal IC design, power management integrated circuit design and communications. He has published 19 international journal papers and 18 international conference papers. He received his B.Tech degree from Nagarjuna University and ME degrees in communication systems from IIT Roorkee. He obtained his PhD from NIT Warangal.

Dr. V. V. Mani is currently working as an Associate Professor at the National Institute of Technology in Warangal, India. She has over 15 years of teaching and research experience in her specific areas of interest, which include signal processing for communication and smart antenna design. She has published 20 international journal papers and 20 international conference papers. She received her BE and ME degrees in electronics and communications engineering from Andhra University in Vishakhapatnam and earned her PhD from the Indian Institute of Technology Delhi, India.

Gibbs-Shannon Entropy and Related Measures: Tsallis Entropy

G. Ramamurthy and T. Jagannadha Swamy

Abstract In this research paper, it is proved that an approximation to the Gibbs-Shannon entropy measure naturally leads to the Tsallis entropy for the real parameter q = 2. Several interesting measures based on the input as well as the output of a discrete memoryless channel are provided and some of the properties of those measures are discussed. It is expected that these results will be of utility in Information Theoretic research.

Keywords Entropy · Approximation · Memoryless channel · Entropy-like measures

1 Introduction

From the considerations of statistical physics, J. Willard Gibbs proposed an interesting entropy measure. Independently, motivated by the desire to capture the uncertainity associated with a random variable, C. E. Shannon proposed an entropy measure. It is later realized that Gibbs and Shannon entropy measures are very closely related. From the considerations of the statistical communication theory, defining mutual information (based on the definition of conditional entropy measure), Shannon successfully proved the channel coding Theorem (which defined the limits of reliable communication from a source to destination over a noisy channel) [1]. Thus, from the point of view of information theory, Shannon entropy measure became very important and useful.

Also, in recent years, Tsallis proposed an entropy measure generalizing the Boltzmann-Gibbs entropy measure. The authors became interested in the relationship between the Tsallis entropy and the Shannon entropy. Under some conditions,

G. Ramamurthy (✉)
Department of CSE, MECHYD, Hyderabad, India
e-mail: rama.murthy@mechyd.ac.in; rammurthy@iiit.ac.in

T. Jagannadha Swamy
Department of ECE, GRIET, Hyderabad, India
e-mail: tatajagan@gmail.com

T. Laxminidhi et al. (eds.), *Advances in Communications, Signal Processing, and VLSI*, Lecture Notes in Electrical Engineering 722,
https://doi.org/10.1007/978-981-33-4058-9_1

the authors proved that the Shannon entropy leads to the Tsallis entropy. As a natural generalization, the authors proposed interesting measures defined on probability mass functions and the channel matrix of a Discrete Memoryless Channel (DMC).

This research paper is organized as follows. In Sect. 2, an approximation to the Shannon entropy is discussed and the relationship to the Tsallis entropy is proved. In Sect. 3, interesting measures on probability mass functions are proposed. Finally, conclusions are reported in Sect. 4.

2 Approximation to Gibbs-Shannon Entropy Measure: Tsallis Entropy

It is well known that the Gibbs-Shannon entropy measure associated with a discrete random variable (specified by the probability mass function $\{\{p_i\}$ for $1 \leq i \leq M\}$ is given by

$$H(X) = -\sum_{i=1}^{M} p_i \log_2 p_i \tag{1}$$

Also, in recent years, Tsallis proposed another entropy measure which in the case of a discrete random variable is given by

$$S_q(p) = \frac{1}{q-1}\left(1 - \sum\nolimits_x p(x)\right)^q \tag{2}$$

where S denotes the entropy, p (.) is the probability mass function of interest and "q" is a real parameter. In the limit as q approaches 1, the normal Boltzmann—Gibbs entropy is recovered. The parameter "q" is a measure of the non-extensivity of the system of interest. The authors became interested in knowing whether there is any relationship between the Gibbs-Shannon entropy measure and the Tsallis entropy under some conditions. This question naturally led to a discovery which is summarized in the following Lemma.

Lemma 1 Consider a discrete random variable X with finite support for the probability mass function. Under reasonable assumptions, we have that

$$H(X) \approx \left(1 - \sum\nolimits_{i=1}^{M} p_i^2\right) \log_2^e = S_2(p) \log_2^e \tag{3}$$

Proof From the basic theory of infinite series [2], we have the following: for $|x| < 1$, we have that

$$\log_e(1-x) = -x + \frac{x^2}{2} - \frac{x^2}{3} + \cdots + (-1)^{n+1}\frac{(-x)^n}{n} + \cdots$$

Let $p_i = (1 - q_i)$ with $0 < p_i < 1$. Then we have $0 < q_i < 1$.
Thus, we have

$$\log_e(1-q_i) = -q_i + \frac{q_i^2}{2} - \frac{q_i^3}{3} + \cdots \tag{4}$$

Now let us consider the entropy H(X) of a discrete random variable, where X assumes finitely many values. We have that

$$\begin{aligned} H(x) &= -\sum\nolimits_{i=1}^{M} p_i \log_2^{p_i} \quad bits \\ &= \quad -\sum\nolimits_{i=1}^{M} (1-q_i)\log_2(1-q_i) \quad bits \\ &= -\sum\nolimits_{i=1}^{M} (1-q_i)\log_e(1-q_i)\log_2 e \end{aligned} \tag{5}$$

Now using the above infinite series and neglecting the terms $\frac{q_i^2}{2}$, $\frac{q_i^3}{3}$, and so on, we have that

$$\begin{aligned} H(x) \quad &\approx \quad -\sum\nolimits_{i=1}^{M} (1-q_i)(-q_i)\log_2^e \\ &\approx \left(\sum\nolimits_{i=1}^{M} p_i\right)(1-p_i)\log_2^e \\ &\approx \left(1 - \sum\nolimits_{I=1}^{M} p_i^2\right)\log_2^e \end{aligned} \tag{Q.E.D.}$$

Remark Thus, the square of the $L^2 - norm$ of the vector corresponding to the probability mass function (of a discrete random variable) is utilized to approximate the entropy of the discrete random variable. In summary, we have that

$$H(x) \approx f(p_1, p_2, \ldots p_M) = (1 - \sum\nolimits_{i-1}^{M} p_i^2)\log_2^e \tag{6}$$

Thus, an approximation to the Gibbs-Shannon entropy naturally leads to the scaled Tsallis entropy for the real parameter q = 2. The quantity H(X) with the above approximation is rounded-off to the nearest integer. For the continuous case, i.e. for probability density functions associated with continuous random variables, similar result can easily be derived and is avoided for brevity.

Note If the logarithm is taken to a different base, a scaling constant should be included.

We would like to study the properties satisfied by the function f (.,.,…,) approximating the entropy. The following claim can easily be proved.

Lemma 2 The maximum value of $f(p_1, p_2, \ldots, p_M)$ is attained when $\{p_i\}_{i=1}^{M}$ are all equal, i.e. $p_i = \frac{1}{M}$ for $1 \leq i \leq M$ Q.E.D.

Proof The proof follows by the application of the Lagrange Multipliers method. Detailed proof is avoided for brevity Q.E.D.

Thus, the maximum value of approximation to the entropy of a discrete random variable assuming "M" values is $\left(1 - \frac{1}{M}\right) \log_2^e$.

It is easy to see that this approximation to the Gibbs-Shannon entropy satisfies only two (out of four) axioms satisfied by the Shannon entropy functional.

Remark As in the proof of the above Lemma, it is possible to provide higher order approximations to the Gibbs-Shannon entropy measure. Also, in the spirit of Lemma 1, Renyi and other types of entropies can easily be approximated. The details are avoided for brevity.

3 Novel Measures on Probability Distributions

Shannon's entropy of a discrete random variable constitutes an important scalar-valued measure defined on the class of probability mass functions (of the discrete random variables). In contrast to the moments of discrete random variables, the entropy does not depend on the actual values assumed by the discrete random variable. Thus, one is naturally led to the definition of other measures associated with discrete random variables which depend only on the probability mass function (and not the values assumed by it).

3.1 L^q-Norm of Probability Vectors: Tsallis Entropy

- We first treat the probability mass function of a discrete random variable as a vector of probabilities. It should be kept in mind the M-dimensional probability vector (corresponding to "M" values assumed by the discrete random variable) lies on a hyperplane in the "positive orthant" (of the M-dimensional Euclidean space) only. Also, as a natural generalization, we can also conceptualize an "infinite-dimensional" probability vector corresponding to a discrete random variable which assumes infinitely many values.
- Consider a "probability vector" (corresponding to the associated probability mass function—finite or infinite dimensional) and define the L^q-norm of the vector (in the same manner as done in pure mathematics). Let

$$M_q(\overline{p}) = \left[\sum_{j=1}^{\infty} [p_x(j)]^q\right]^{\frac{1}{q}} \text{ for } q \geq 1 \tag{7}$$

As discussed in [3], some interesting properties are satisfied by M_q. Also, all the results associated with L^q-norm (in pure mathematics) such as the Holder and Minkowski inequalities can be readily invoked with the measure M_q.

It is elementary to see that such a measure can easily be related to the Tsallis entropy. Specifically, we have that

$$M_q^q = 1 - (q-1)S_q(p) \tag{8}$$

Now let us define the following function naturally associated with the L^q-norm, i.e. M_q.

$$g_q(\overline{p}) = 1 - M_q(\overline{p})$$

It is easy to see that for any two real numbers q_1, q_2 such that $q_1 > q_2$, we have that

$$g_{q_1}(\overline{p}) > g_{q_2}(\overline{p})$$

In view of this, it is easy to reason that $\lim_{q\to\infty} g_q(\overline{p}) = 1$. In a similar spirit, it is possible to derive an inequality associated with $(q-1)S_q(p)$, i.e. scaled Tsallis entropy (for different values of the real parameter "q").

- Based on the properties of M_q, it is easy to see that the probability mass function-based infinite-dimensional probability vectors always belong to discrete Hilbert space.
- Let us first consider the case where the support of the probability mass function is finite. The L^2-norm of the associated probability vector is

$$M_2 = \left[\sum_{j=1}^{M} [p_X(j)]^2\right]^{\frac{1}{2}} \tag{9}$$

We reasoned in the previous section that such a measure naturally arises in approximating the Gibbs-Shannon entropy functional/measure (to a good degree of accuracy).

- Using a similar approach, the conditional entropy can be approximated. Also, using the approximation for H(X/Y)/H(Y/X), H(X)/H(Y), the mutual information between the input and output of a Discrete Memoryless Channel (DMC) can be approximated. The details are avoided for brevity.

3.2 Quadratic Forms Associated with Probability Mass Functions

- Clearly, the expression in (3) is an interesting measure defined over the Vector, $\overline{p_X}$ representing the probability mass function. Thus, one is naturally led to the definition of a quadratic form defined over the vector $\overline{p_X}$. Specifically, let us define quadratic forms associated with the channel matrix, Q (of a DMC),
  i.e. $\overline{p_X^T} Q \overline{p_X}$.
  Since $\overline{p_X^T} Q = \overline{p_{Y,}}$, we readily have that

$$\overline{p_X^T} Q \overline{p_X} \; = \; \overline{p_Y^T}\,\overline{p_X} = \langle \overline{p_Y}, \overline{p_X} \rangle \tag{10}$$

Claim Thus, the quadratic form associated with the channel matrix of a DMC represents the inner producy between the probability vectors $\overline{p_Y} and \overline{p_X}$.

- It readily follows that in the case of a "noiseless channel", we have that Q = I and thus the quadratic form becomes the "square of the Euclidean length" (L^2-norm) of the probability vector. It is thus always positive.

- In view of the relationship of the Tsallis entropy to the Gibbs-Shannon entropy measure, we define the following measure associated with the stochastic matrix W and the probability vector $\overline{p_X}$, i.e.

$$\overline{S_2} = 1 - \overline{p_X^T} W \overline{p_X}.$$

If W is the channel matrix of a discrete memoryless channel, the above measure has an interesting interpretation (discussed previously). In this case,

$$\overline{S_2} = 1 - \overline{p_X^T} W \overline{p_X} = 1 - \overline{p_X^T}\,\overline{p_Y}.$$

It is easy to reason that this measure is non-negative. Also using Lemma 2, the above entropy-type measure can be bounded.

It is interesting to see its interpretation when W is the state transition matrix of a homogeneous Markov chain. In this case, the state of the dynamical system captured through an associated probability vector evolves through the associated Markov chain. (We can capture the idea of initial entropy, transient entropy and equilibrium entropy of the associated Markov chain modeling the physical phenomena.)

Furthermore, W could be a doubly stochastic matrix. It is immediate that when W happens to be an identity matrix, i.e. W = I, then the above measure is the Tsallis entropy for parameter q = 2 (i.e. an approximation to the Gibbs-Shannon entropy measure).

- Hence, we would like to study the properties of the quadratic form using the Inner product between two probability vectors (namely the input and output probability vectors of a DMC). In that effort, we would like to address the following question:

Q: How does the inner product of two probability vectors summarize the "similarity/dissimilarity" of probability mass functions?

In this effort, we invoke the Cauchy-Schwartz inequality associated with bounding the inner product between two vectors:

$$\left[\overline{p_Y^T}\,\overline{p_Y}\right]^2 \leq \left[\sum\nolimits_{i=1}^{M} p_X^2(i)\right]\left[\sum\nolimits_{i=1}^{M} p_Y^2(i)\right] \tag{11}$$

- It is easily seen that the following holds true:

$$\sum\nolimits_{i=1}^{M} p_X^2(i) = \begin{cases} \{1 \text{ if } \overline{p_X} \text{ is degenerate} \\ \{< 1 \text{ if } \overline{p_X} \text{ is non - degenerate} \end{cases} \tag{12}$$

- Furthermore, the minimum possible value of $\sum\nolimits_{i=1}^{M} p_X^2(i)$ (i.e. value of 1/M) occurs when $p_X(i) = 1/M$ for all $1 \leq i \leq M$.
- Also, it should be noted that the inequality in (11) reduces to equality only when $p_X(i) = p_y(i)$ for all $1 \leq i \leq M$.

That is, the inner product between probability vectors $\overline{p_X}$ and $\overline{p_Y}$ attains the "maximum" value when they are both the same (equal).

- Suppose $\overline{p_X}$ is the invariant probability distribution (also called the steady-state probability distribution) of the homogeneous Discrete Time Markov Chain (DTMC) associated with the Channel matrix Q (a stochastic matrix). In this case, we have that

$$\overline{p_X^T} Q = \overline{p_X^T} \tag{13}$$

Then the quadratic form associated with $\overline{p_X}$ becomes

$$\overline{p_X^T} Q \overline{P_X} = \overline{p_X^T}\,\overline{p_X} > 0$$

Thus, the quadratic form attains the maximum value. Equivalently, we have that in this case, the value of the quadratic form is the same as that in the case of a noiseless channel.

- In the same spirit of the definition of mutual information, let us define the following scalar "measure" between the input and output of a Discrete Memoryless Channel (DMC).

$$\text{J(X; Y)} = \overline{p_X^T} Q \overline{p_Y} = \overline{p_Y^T} \overline{p_Y} \tag{14}$$

where $\overline{p_X}$ corresponds to the input probability vector (i.e. the input probability mass function) and $\overline{p_Y}$ corresponds to the output probability vector. Let us investigate some of the properties of the scalar measure J(X; Y).

(i) Since $\overline{p_X^T} Q = \overline{P_Y^T}$, we have that $\overline{p_Y^T}\overline{p_Y} > 0$.

Also, we have that J(X; X) = = $\overline{p_X^T} Q \overline{p_X} = \overline{p_X} \geq 0$.
That is, J(X; X) is zero when the probability vectors $\overline{p_X}, \overline{p_Y}$ are orthogonal vectors (as in the case of vector spaces).

(ii) J(Y; X) = $\overline{p_Y^T} Q \overline{p_X}$. Now substituting $\overline{p_Y^T} = \overline{p_X^T} Q$,
we have that J (Y; X) = $\overline{p_Y^T}\overline{p_Y}$ = J (X; Y). Thus the scalar measure is symmetric.

Now we check whether the scalar-valued measure satisfies the triangular inequality. (The random variable X is the input to a discrete memoryless channel whose output is Y. Y is in turn the input to another discrete memoryless channel whose output is Z.)

$$J(X; Y) = \overline{p_X^T} Q \overline{p_Y} = \overline{p_Y^T} \overline{p_Y},$$

$$J(Y; Z) = \overline{p_Y^T} Q \overline{p_Z} = \overline{p_Z^T} \overline{p_Z},$$

Hence, we necessarily have that

$$J(X; Y) + J(Y; Z) = \overline{p_Y^T} \overline{p_Y} + \overline{p_Z^T} \overline{p_Z}$$

But by definition $J(X; Z) = \overline{p_X^T} Q \overline{P_Z} = \overline{P_Z^T} \overline{P_Z}$
Thus $J(X; Y) + J(Y; Z) \geq J(X; Z)$
Hence the triangular inequality is satisfied. Thus, the following Lemma is established.

Lemma 3 The scalar-valued measure

$$J(X; Y) = \overline{p_X^T} Q \overline{p_Y} = \overline{p_Y^T} \overline{p_Y} \tag{15}$$

between the probability vectors (corresponding to the probability mass functions of the random variables X, Y) is a "Pseudo-Metric" on the space of probability vectors (where the random variable Y is the output of a Discrete Memoryless Channel whose input is X).

In the spirit of the definition of the Tsallis entropy, we can define an Interesting entropy-type measure 1- J(X; Y). It is precisely the Tsallis entropy of the probability mass function of the output of a discrete memoryless channel.

Remark It is well known that higher degree forms (multivariate polynomials) are captured through Tensors. Thus using tensors, new measures can be defined generalizing the above ideas. The details are not provided for brevity.

3.3 Now We Summarize the Results Discussed so Far in the Following

J(X) $= \left[\sum_{j=1}^{\infty} [p_X(j)]^2\right]^{\frac{1}{2}}$ is like the "Euclidean Length" of a probability vector.

In Lemma 1, it was shown that $1 - [J(X)]^2$ approximates the Gibbs-Shannon entropy of the random variable X.

- J(X; Y) is a scalar-valued measure on the input X and output Y of a discrete memoryless channel.

4 Conclusions

In this research paper, the relationship between the Gibbs-Shannon entropy measure and the Tsallis entropy (for q = 2) is demonstrated. Based on this result, various interesting measures associated with probability mass functions (defined at the input and output of a Discrete Memoryless Channel) are defined. It is expected that these results will be of utility in Information Theoretic Investigations.

References

1. Ash RB, Information theory. Dover Publications, Inc, New York
2. Knopp K, Theory and applications of infinite series. Dover Publications, Inc, New York
3. Rama Murthy G, Weakly short memory stochastic processes: signal processing perspectives. In: Proceedings of international conference on frontiers of interface between statistics and sciences, December 20, 2009 to January 02, 2010

Recognition of Natural and Computer-Generated Images Using Convolutional Neural Network

K. Rajasekhar and Gopisetti Indra Sai Kumar

Abstract Recognizing Natural Images (NI) and Computer-Generated Images (CGI) by a human is difficult due to the use of new-age computer graphics tools for designing more photorealistic CGI images. Identifying whether an image was captured naturally or if it is a computer generated image is a fundamental research problem. For this problem, we design and implement a new Convolutional Neural Network (ConvNet) architecture along with data augmentation techniques. Experimental results show that our method outperforms existing methods by 2.09 percentage for recognizing NI and CGI images.

Keywords Convolutional neural network · Computer-generated image · Natural image · MATLAB · Image processing · Image forensics

1 Introduction

Identification of Natural Images (NI) and Computer-Generated Images (CGI) has become an important research problem. Computer graphics now a days has evolved into the same photorealism as natural images due to various graphics designing tools. However, recognizing the differences between the images is difficult. The methods that exist now are based on the statistical and intrinsic properties of images, i.e. to design a category-distinctive feature with a proper threshold to separate the

K. Rajasekhar
Department of Electronics and Communication Engineering, University College of Engineering Kakinada (A), Jawaharlal Nehru Technological University Kakinada, Kakinada, Andhra Pradesh, India
e-mail: rajakarumuri87@gmail.com

G. I. S. Kumar (✉)
Computers and Communications (M.Tech), Department of Electronics and Communication Engineering, University College of Engineering Kakinada(A), Jawaharlal Nehru Technological University Kakinada, Kakinada, Andhra Pradesh, India
e-mail: gopisettiindrasaikumar@gmail.com

T. Laxminidhi et al. (eds.), *Advances in Communications, Signal Processing, and VLSI*,
Lecture Notes in Electrical Engineering 722,
https://doi.org/10.1007/978-981-33-4058-9_2

Fig. 1 Computer-generated image

Fig. 2 Naturally captured image

two classes. This method performs well for simple datasets and poorly on complex datasets, for example, Columbia dataset, comprising images of different origins.

Today's methods based on Convolutional Neural Network (ConvNet) have gained more popularity in analyzing visual imagery and the reason for this is to learn automatically multiple levels of representation for a given task in an "end-to-end" manner [1]. This makes ConvNet more suitable for complex datasets in image recognition tasks. Inspired by the recent success of ConvNet, we and several other researchers chose this approach for CGI and NI recognition tasks. Our method gives good performance for the complex dataset of Google Image Search (Google) images versus Photo-Realistic Computer Graphics (PRCG) images from different origins and close to real-world applications.

Image forensics is an active field where images are analyzed. But in recent years, computers are able to generate photorealistic images which have become more challenging for forensics to determine whether the image is real or fake. In multimedia security, a number of approaches use ConvNet for steganalysis and image forensics. From Figs. 1 and 2, we get an impression that the first image is naturally generated and the second image is computer generated which is in contrast to reality.

2 Related Work

ConvNet for Recognition of Computer Generated Image and Natural Images. For the recognition of CGI and NI using ConvNet, there are two different works where one is presented in IEEE WIFS by Rahmouni et al. [2]. In this work, they followed a three-step procedure: filtering, statistical feature extraction, and classification. They considered a relatively simple dataset (Raise versus Level-Design) with homogeneous NIs and CGIs from the green channel. The second work is done by Quan et al. by using deeper and particularly designed cascaded convolutional layer with the Columbia dataset mainly Google versus PRCG and using all three channels of an image, i.e. red, blue, and green [3]. Qian et al. proposed a model on ConvNet for steganalysis and reported promising results [4]. Later, Pibre et al. studied the "shape" of ConvNet and identified the best ConvNet model after numerous experiments [5].

3 Dataset Used for Implementation

We conducted our experiment on Columbia photographic images and photorealistic computer graphics dataset [6, 7] considering

- 800 PRCG images from 40 3D graphic websites (prcg_images)
- 800 natural images from Google image search. (google_images)
- 800 natural images from authors' personal collection (personal images)

We considered all three channels of an image, i.e. Red, Blue, and Green. These images are collected from different sites and are of different origins.

4 Framework Proposed

Recognition of NIs and CGI images can be treated as a binary classification problem because we give an image as input and obtain a binary label as output.

Before giving input to the network, we augment the data using imageDataAugmenter so that the data will be according to the input size of the network and also it provides preprocessing image augmentation options like resizing, rotation, reflection, etc. for the data. These preprocessed images are fed to the ConvNet network architecture which is arranged as shown in Fig. 3 so a trained model (.mat file) of required accuracy is obtained after training the network with the data. The model generated will be used to recognize the NI and CGI images. Here, we split the data into two sets: one is for training and the other is for validation; this is mainly for the purpose of obtaining the generalized model for recognition.

4.1 Network Architecture

Our network architecture consists of four convolution layers, three fully connected layers, one SoftMax layer, and a classification layer. An image is taken as an input using the image input layer. The input size of the image for the network is (233 $\times$ 233 $\times$ 3) Normalization as zerocenter. In our network, a convolutional 2D layer

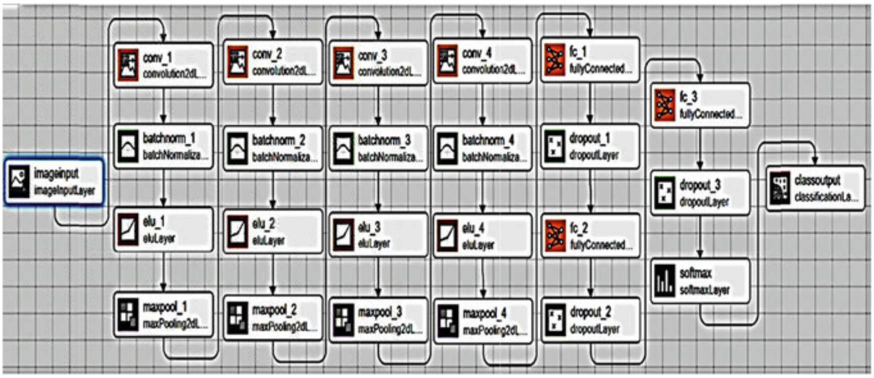

Fig. 3 Network architecture

(Conv_1), batch normalization layer (batchnorm_1), ELU layer (elu_1), and max-pooling 2D layer (maxpool_1) are treated as a single layer as there are 4 layers [8, 9]. All the max-pooling 2D layers have the same pool size of 3 × 3 and a stride of 2,2, and zero padding. The fully connected layers are followed by the Dropout layer each [10]. The dropout probability value is kept default. The SoftMax layer gives a probability vector of class labels. Therefore, the dimension of its output is equal to the number of classes, and the sum of its output is 1. Finally, classification layer is used to produce the output label. Our network layer description is shown in Fig. 4.

4.2 Optimizer

The stochastic gradient descent with momentum (SGDM) is the optimizer used in the network for training [11]. The training options are chosen as follows: minibatch size of 36, initial learn rate as 10^{-9}, validation frequency as 9, and the number of epochs to train is based on the accuracy that is needed to get. We actually trained for 300 epochs in order to obtain the required accuracy for the model. We also shuffled the data at every epoch so that the model will be able to generalize the data outside the datasets used. We can also choose the learn rate schedule to be piecewise and the drop factor and the drop period to be any random values. But at the initial stage, we choose the learn rate schedule to be constant and later we manually decreased the learn rate.

4.3 Training

We actually split the data into two sets with 75% of data used to train the network and the remaining 25% data used to validate the trained model. Minibatch size is made

```
25x1 Layer array with layers:

 1  'imageinput'   Image Input             233x233x3 images with 'zerocenter' normalization
 2  'conv_1'       Convolution             32 7x7x3 convolutions with stride [1  1] and padding [0  0  0  0]
 3  'batchnorm_1'  Batch Normalization     Batch normalization with 32 channels
 4  'elu_1'        ELU                     ELU with Alpha 1
 5  'maxpool_1'    Max Pooling             3x3 max pooling with stride [2  2] and padding [0  0  0  0]
 6  'conv_2'       Convolution             64 7x7x32 convolutions with stride [1  1] and padding 'same'
 7  'batchnorm_2'  Batch Normalization     Batch normalization with 64 channels
 8  'elu_2'        ELU                     ELU with Alpha 1
 9  'maxpool_2'    Max Pooling             3x3 max pooling with stride [2  2] and padding [0  0  0  0]
10  'conv_3'       Convolution             48 5x5x64 convolutions with stride [1  1] and padding 'same'
11  'batchnorm_3'  Batch Normalization     Batch normalization with 48 channels
12  'elu_3'        ELU                     ELU with Alpha 1
13  'maxpool_3'    Max Pooling             3x3 max pooling with stride [2  2] and padding [0  0  0  0]
14  'conv_4'       Convolution             64 3x3x48 convolutions with stride [1  1] and padding 'same'
15  'batchnorm_4'  Batch Normalization     Batch normalization with 64 channels
16  'elu_4'        ELU                     ELU with Alpha 1
17  'maxpool_4'    Max Pooling             3x3 max pooling with stride [2  2] and padding [0  0  0  0]
18  'fc_1'         Fully Connected         4096 fully connected layer
19  'dropout_1'    Dropout                 50% dropout
20  'fc_2'         Fully Connected         4096 fully connected layer
21  'dropout_2'    Dropout                 50% dropout
22  'fc_3'         Fully Connected         2 fully connected layer
23  'dropout_3'    Dropout                 50% dropout
24  'softmax'      Softmax                 softmax
25  'classoutput'  Classification Output   crossentropyex with classes 'google_images' and 'prcg_images'
```

Fig. 4 Layer description

to be 36 and the number of epochs is chosen to be 200. So the network is trained for 6600 iterations at a constant learn rate of 0.001 got by a validation accuracy of 79 percent for this run as shown in Fig. 5. Though we can choose the learn rate to be piecewise, we actually choosen learn rate manually after obtaining the model with 79 accuracy and varied the learn rate within a range from 10^{-3} to 10^{-6} for 100 epochs so that our model achieved good accuracy. Here, the validation frequency is maintained at 8; this is mainly used to calculate the validation accuracy and loss of the model to generalize the model to other data. For every epoch, it took around 5 min and for every iteration it took 0.135 min; here in MATLAB, there is an option to do it parallelly so it uses cores of the CPU to work simultaneously.

One point to be noted here is we split the data according to the label of the folder name, i.e. folder containing natural images are named google_images and the other folder is named as prcg_images. So the classification layer gets the names for their binary labels from these folder names.

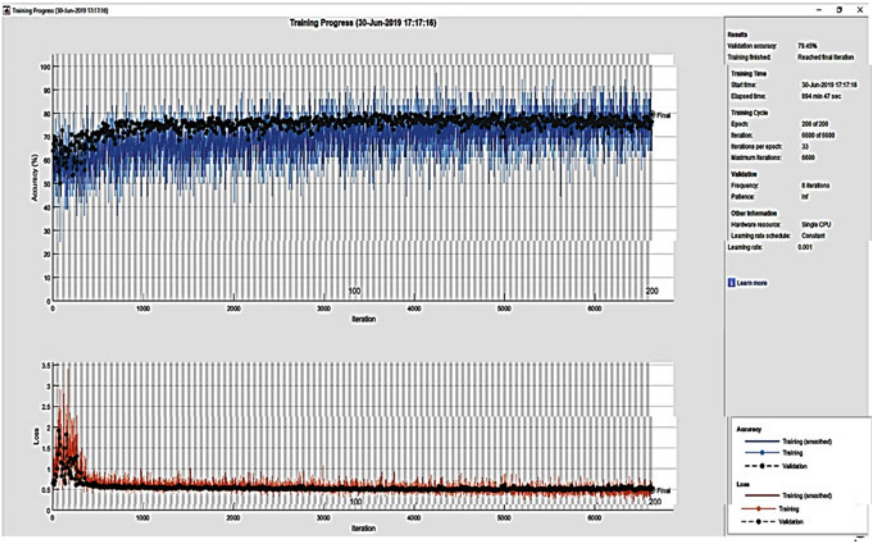

Fig. 5 Training plot for accuracy versus ıteration and loss versus ıteration

5 Results

5.1 Implementation Details

MATLAB R2019a. All of the experiment is done in the MATLAB R2019a Deep Learning package [10, 11] using Deep Learning Toolbox and the Deep Network Designer application. We have designed our network architecture using the Deep Network Designer app and generated code for the network architecture. We used image augmentation techniques to the data before applying them to the network. We split the data into training and validation in a 3:1 ratio.

In Training options, we choose sgdm optimizer and related parameters for this experiment and trained the network. After training, in order to improve the accuracy of the model, we save the checkpoints and retrain the network. Figure 6 shows the classified images by the trained model, and Fig. 7 gives the probabilities for the classification of the images as prcg_image and google_image.

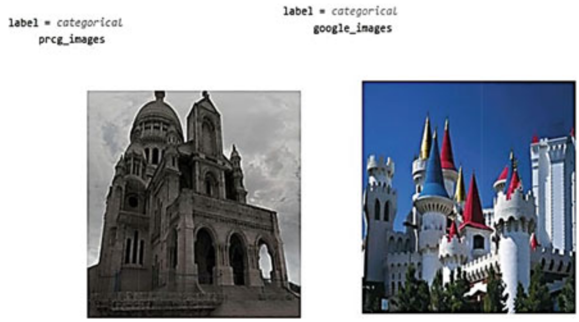

Fig. 6 Classified ımages

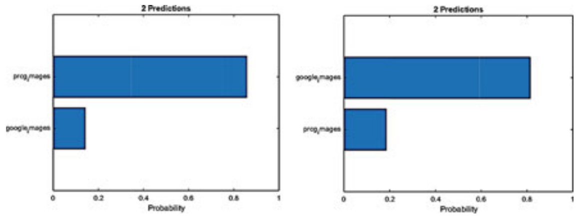

Fig. 7 Classification probabilities

5.2 *Visualization of ConvNet Layers*

After the model with the required accuracy is obtained, we visualize the network layers for the data it gets trained in. For this purpose, in MATLAB the DeepDreamImage option will be available. Figure 8 gives details about the patterns that are grasped by the hidden layers in the ConvNet.

Figure 9 shows the table for activation strength and the classification layer through this layer visualization; we can say that Natural Images have more complex patterns

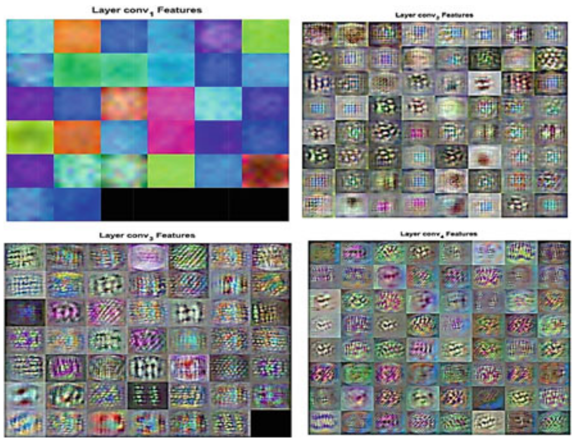

Fig. 8 Visualization of hidden ConvNet layers

Iteration	Activation Strength	Pyramid Level
1	0.58	1
2	5.11	1
3	7.53	1
4	10.16	1
5	12.43	1
6	14.05	1
7	14.58	1
8	15.90	1
9	16.54	1
10	17.90	1

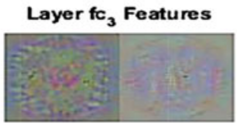

Fig. 9 Classification layer activation strength and visualization

and Computer Generated Images have fewer complex patterns and are mostly simple patterns.

Through this, we can say that CGI images nowadays are getting more photorealistic so the pattern complexity in CGI is increasing. Maybe in the future, their differentiation will be more difficult compared to now.

Visualization of Activations for Test Images. For the testing purpose, we give some Natural and Computer Generated images and visualize the activations of the layers. Figure 10 shows the activations of the layers for the NI and CGI images. Figure 11 shows the strong activation of the layer to recognize the image. These are actually the normalized images which are gray; otherwise, they are black and white in color. After that, we apply this model to the video to detect the frames in the video whether

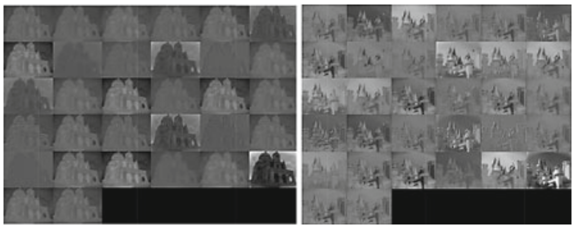

Fig. 10 Activations of ConvNet layer 1 by CGI and NI

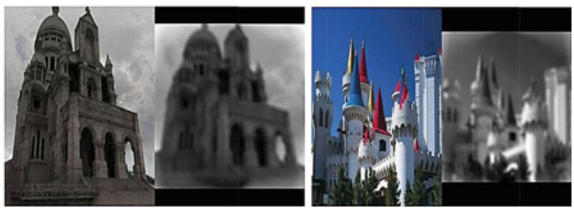

Fig. 11 Strong activations of ConvNet layer 1